

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Využití simulačních metod pro vizuální analytiku
Diplomová práce

Autor: Bc. Martina Nosilová
Studijní obor: IM2-P

Vedoucí práce: Ing. Karel Mls, Ph.D.

Hradec Králové

listopad 2020

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 16.11.2020

A handwritten signature in blue ink, appearing to be 'Kerida', written over a horizontal dashed line.

vlastnoruční podpis

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Karlu Mlsovi, Ph.D. za metodické vedení práce, cenné rady a připomínky a především za trpělivost a ochotu, kterou mi během zpracování mé diplomové práce poskytl. Děkuji Mgr. Janu Draesslerovi, Ph.D. za pomoc při tvorbě hypotéz. Také děkuji Ing. Dominiku Bydžovskému za odbornou konzultaci při tvorbě diplomové práce. Dále děkuji své rodině, kamarádům a učitelům za podporu při studiu a rady při psaní diplomové práce.

Anotace

Diplomová práce se zabývá modelováním reálných situací. Je zaměřená na využití vizuální analytiky při jejich studiu a interpretaci. Diplomová práce se nejdříve zabývá teoretickými východisky, která jsou následně využita v praktické části této práce. Do praktické části byla vybrána, včetně stávajícího problematického stavu, dvě možná a diskutovaná řešení rekonstrukce křižovatky Mileta v Hradci Králové. Tyto tři varianty byly vymodelovány pomocí nástroje pro tvorbu multiagentových modelů - programu NetLogo. Následně byly vytvořeny simulace provozu v jednotlivých verzích křižovatky, včetně dynamické vizualizace toku vozidel a vytváření kolon. V praktické části jsou dále uvedeny výsledky dotazníkového šetření a ověřování pěti předem stanovených hypotéz. Cílem této práce je analýza a praktické ověření možností a přínosů vizualizačních technik v rámci simulačních přístupů, k popisu a zkoumání složitých dynamických systémů.

Klíčová slova:

Model, simulace, vizuální analýza, NetLogo.

Annotation

Title: The use of simulation methods for visual analytics

The diploma thesis deals with modeling of real situations, it is focused on the use of visual analytics in their study and interpretation. The diploma thesis first deals with the theoretical basis, which is then used in the practical part of this work. Two possible and discussed solutions for the reconstruction of the Mileta intersection in Hradec Králové were selected for the practical part, including the current problematic situation. These three variants were modeled using a tool for creating multi-agent models - the NetLogo software. Subsequently, traffic simulations were created in individual versions of the intersection, including dynamic visualization of the flow of vehicles and the formation of traffic jams. The practical part also presents the results of a questionnaire survey and verification of five predetermined hypotheses. The objective of the thesis is the analysis and practical verification of the possibilities and benefits of visualization techniques in the context of simulation approaches to the description and investigation of complex dynamic systems.

Key words:

Model, simulation, visual analysis, NetLogo.

Obsah

1	Úvod.....	1
1.1	Cíl práce a metodika zpracování	1
2	Modelování	3
2.1	Principy	4
2.2	Kritéria hodnocení.....	5
2.3	Přístupy	5
2.4	Typy modelů.....	5
3	Simulace.....	7
3.1	Charakteristika simulačního modelu.....	7
3.2	Využití simulace	8
3.3	Nástroje.....	10
4	Vodopádový model.....	14
4.1	Výhody a nevýhody	15
5	Vizuální analýza	16
5.1	Historie.....	17
5.2	Aplikace vizuální analýzy	17
5.3	Vizuální analýza v dopravě.....	19
6	Analýza simulačních přístupů	21
6.1	Stávající stav – varianta I.....	21
6.2	Nové řešení – varianta II.....	22
6.3	Nové řešení – varianta III.....	24
6.4	Modelovací nástroj – NetLogo.....	25
7	Hypotézy.....	36
7.1	Stanovení hypotéz.....	36
7.2	Vyhodnocení dotazníkového šetření	37

7.3	Ověření stanovených hypotéz	42
8	Shrnutí výsledků.....	48
9	Závěry a doporučení	51
10	Seznam použité literatury.....	53
11	Přílohy	56
11.1	Dotazník.....	56
11.2	Kód NetLogo – Varianta I.....	58
11.3	Kód NetLogo – Varianta II.....	74
11.4	Kód NetLogo – Varianta III.....	90

Seznam obrázků

Obrázek 1 – Oblasti zkoumání (Roques 2004).....	4
Obrázek 2 – Postup od objektu po model (Vlček a Chuchro 1999).....	4
Obrázek 3 – Sedm základních fází Vodopádového modelu (Royce 1970)	14
Obrázek 4 – Waterfall výhody a nevýhody (Dvořáček a Mareček 2017).....	15
Obrázek 5 – Visual Data Exploration (Keim 2010).....	18
Obrázek 6 – Stávající stav (varianta I.) (webkameryzive.cz).....	22
Obrázek 7 - Nový stav křižovatky Mileta – varianta s odbočovacími pruhy (Záleský 2020)	23
Obrázek 8 – Nový stav křižovatky Mileta – varianta III. (Volejníková 2013).....	24
Obrázek 9 – Parametry notebooku, na kterém byla provedena simulace (vlastní zpracování).....	25

Seznam zdrojových kódů

Zdrojový kód 1 - Agenti ve Variantě I. (vlastní zpracování).....	26
Zdrojový kód 2 - Deklarované proměnné v modelech se světelnou křižovatkou – Varianta I. a II. (vlastní zpracování)	27
Zdrojový kód 3 - Deklarované proměnné v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)	27
Zdrojový kód 4 - Procedura „SETUP“ v modelech se světelnou křižovatkou – Varianta I. a II. (vlastní zpracování).....	28
Zdrojový kód 5 - Procedura „SETUP“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování).....	28
Zdrojový kód 6 - Procedura „GO“ v modelech se světelnou křižovatkou – Varianta I. a II. (vlastní zpracování)	29
Zdrojový kód 7 - Procedura „GO“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování).....	29
Zdrojový kód 8 - Část procedury „set_turning_points“ v modelech se světelnou křižovatkou – Varianta I. (vlastní zpracování)	30
Zdrojový kód 9 - Část procedury „set_turning_points“ a procedura „set_invisible_points“ – Varianta III. (vlastní zpracování)	31

Zdrojový kód 10 - Procedura „set_preference_patches“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)	31
Zdrojový kód 11 - Procedura „set_preference_cars“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)	32
Zdrojový kód 12 - Část procedury „draw_background“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)	32
Zdrojový kód 13 - Procedura „traffic_roundabout“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)	33
Zdrojový kód 14 - Části procedury (pohyb v křižovatce) „move_cars“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)	34
Zdrojový kód 15 - ČástI procedury „move_cars“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)	35
Zdrojový kód 16 - Část procedury „traffic_lights“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování).....	35

Seznam tabulek

Tabulka 1 – Dílčí výpočty při testování hypotézy 1 (vlastní zpracování).....	42
Tabulka 2 – Dílčí výpočty při testování hypotézy 2 (vlastní zpracování).....	43
Tabulka 3 – Dílčí výpočty při testování hypotézy 3 (vlastní zpracování).....	44
Tabulka 4 – Dílčí výpočty při testování hypotézy 4 (vlastní zpracování).....	45
Tabulka 5 – Pearsonův koeficient pro hypotézu 4 (vlastní zpracování).....	46
Tabulka 6 – Dílčí výpočty při testování hypotézy 5 (vlastní zpracování).....	47
Tabulka 7 – Pearsonův koeficient pro hypotézu 5 (vlastní zpracování).....	47

Seznam grafů

Graf 1 – Věkové rozložení respondentů (vlastní zpracování)	37
Graf 2 – Pohlaví respondentů (vlastní zpracování).....	37
Graf 3 – Vztah ke křižovatce Mileta (vlastní zpracování).....	38
Graf 4 – Doba, po kterou respondenti vlastní řidičské oprávnění (vlastní zpracování)	38

Graf 5 – Stávající stav křižovatky Mileta je dle názorů respondentů (vlastní zpracování).....	39
Graf 6 – Vizuální porovnání všech tří variant křižovatek (vlastní zpracování).....	39
Graf 7 – Volba optimálního řešení respondenty (vlastní zpracování)	40
Graf 8 – Příznivost pro chodce/cyklisty (vlastní zpracování)	40
Graf 9 – Plynulost dopravy (vlastní zpracování)	41
Graf 10 – Sklony vytvářet kolony (vlastní zpracování).....	41

1 Úvod

Práce se zabývá problémovou křižovatkou Mileta v Hradci Králové. Možnosti přestavby křižovatky jsou diskutovány již mnoho let. Pro tuto diplomovou práci budou zvoleny dvě varianty vylepšení, se kterými bude dál pracováno v praktické části. Nejčastěji diskutovanými možnostmi, jak zlepšit situaci na křižovatce jsou dva návrhy – první se podobá stávajícímu stavu (varianta I.), tedy světelné křižovatce. Oproti stávajícímu stavu je navrženo přidání odbočovacích pruhů a podchodů pro chodce a cyklisty (varianta II.). Druhá možnost je víceúrovňový kruhový objezd, který by byl bez světelné signalizace a kde je také počítáno s podchody pro chodce (varianta III.).

Práci lze rozdělit na dvě části – teoretickou a praktickou. V teoretické části jsou popsána teoretická východiska, ze kterých je následně v praktické části vycházeno. Teoretická část je věnována modelování, simulaci, vodopádovému modelu a v neposlední řadě vizuální analýze. Každá kapitola obsahuje několik podkapitol, ve kterých jsou představeny dílčí oblasti dané problematiky.

Praktická část se zabývá především popisem možných variant křižovatky Mileta a tvorbou modelů ve vybraném programu. Na závěr praktické části jsou ověřovány hypotézy, které jsou vyhodnocovány na základě výsledků dotazníkového šetření a dat získaných z programu NetLogo.

1.1 Cíl práce a metodika zpracování

Cílem diplomové práce je analýza simulačních přístupů, nástrojů a jejich vizualizačních možností. Dále navrhnout model a ověřit jeho výstupy na konkrétní simulaci.

Jako oblast zkoumání byla zvolena křižovatka Mileta v Hradci Králové. Pro modely křižovatky bude vybrán program, ve kterém budou modely vypracovány.

Bylo stanoveno pět hypotéz, které se budou testovat na závěr práce. Testování proběhne na základě dotazníkového šetření a dat z modelů. Bude vybrán test pro ověření hypotéz a následně budou data v MS Excel zpracována.

Hypotéza 1

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. muži častěji volí variantu III. než ženy.

Hypotéza 2

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. starší respondenti častěji volí variantu III. než mladší respondenti.

Hypotéza 3

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. zkušenější řidiči častěji volí variantu III. než řidiči nezkušení.

(předpoklad pro zkušenost řidičů je doba, po kterou vlastní řidičský průkaz)

Hypotéza 4

Hypotéza ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 odpovídá seřazení modelů podle dat z NetLogo (počty projíždějících vozidel).

Hypotéza 5

Hypotéza ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 odpovídá seřazení modelů podle dat z NetLogo (časy na projetí křižovatkou).

2 Modelování

„Všechny modely jsou špatně, některé modely jsou užitečné.“ (Box 2013)

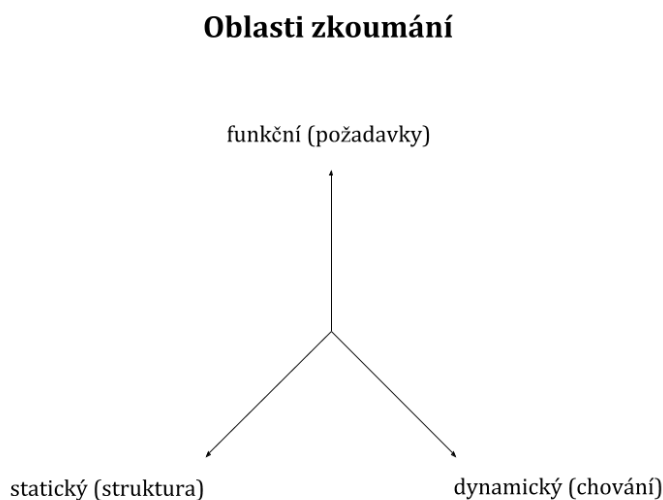
„Proces tvorby a aplikací modelu je nutno zasadit mezi inteligenci a zkušenost. Není-li tomu tak, pak výsledkem této náročné práce je samozřejmost nebo hloupost.“ (Habr 1976)

Modelování má několik funkcí, kterými jsou funkce poznávací, vysvětlovací a integrační. (Vlček a Chuchro 1999) Hlavním účelem modelování je zjednodušená reprezentace předem stanovené části reality. Model není dokonalým obrazem reality, protože ta je většinou složitější a obtížně se modelují všechny detaily, proto se při modelování uplatňuje tzv. systémový homomorfismus. Systémový homomorfismus popisuje dva systémy, kde při porovnání zjistíme, že prvek jednoho systému odpovídá prvku druhého systému, ale naopak toto pravidlo neplatí. Zpravidla systémový homomorfismus platí mezi originálem a modelem tohoto systému. Modelování umožňuje zkoumat realitu bez ohrožení reálného systému. Model také sestavujeme za účelem ověření či vyvrácení hypotézy. (HUBÁLOVSKÝ 2011; Bureš 2007)

Prostřednictvím modelovaného systému můžeme testovat chování reality při nastavení různých parametrů, sledovat dopady nebo odhadovat budoucí vývoj reality. Díky tomu můžeme předvídat kroky s negativním vlivem na realitu a předejít jim nebo snížit jejich negativní působení na realitu. Výhodou modelu je možnost sledovat vývoj a změny v realitě v čase. Procesy v modelu mohou probíhat rychleji než v realitě. (Láska 2017)

Při modelování si nejprve určíme způsob, jakým budeme zkoumat realitu, vymezení stěžejních pojmů, které se daného modelu týkají. Význačnou roli hraje správné zachycení různých aspektů zkoumané reality (systému), při kterém je podstatné zohlednění míry detailu. Zkoumání reality je rozděleno na úrovně, které můžeme zjednodušeně rozdělit do tří vrstev. První úroveň je úroveň konceptuální (analytická), která identifikuje problém, vymezí rozsah a návaznosti na okolí a poskytne základní pohled na strukturu i chování určitého systému. Specifikační (návrhová) úroveň rozvádí, zpřesňuje a formalizuje výstupy z konkrétní analýzy a zasazuje řešení do vybrané architektury na dané platformě. Třetí úroveň je

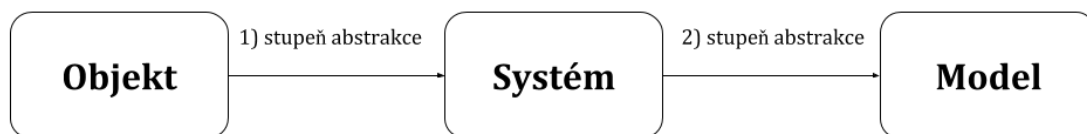
implementační, která řeší praktickou otázku „Čím to bude řešeno“. Při zkoumání předlohy k modelu zkoumáme tři oblasti funkční, dynamickou a statickou. Obrázek 1 tyto oblasti zobrazuje. (Roques 2004)



Obrázek 1 - Oblasti zkoumání (Roques 2004)

Při modelování je důležitý systémový přístup, při jehož uplatnění se vyžaduje nejprve vymezení objektu (existující reálný objekt nebo v mysli člověka), následuje definování systému na objektu a na závěr zobrazení systému modelem. Obrázek 2 ukazuje tento postup od objektu přes systém k modelu. (Vlček a Chuchro 1999)

2.1 Principy



Obrázek 2 - Postup od objektu po model (Vlček a Chuchro 1999)

Při modelování musíme dodržovat principy abstrakce, formalizace, jednoznačnosti a zamezení redundancím. Abstrakce umožňuje identifikovat podobnost ve zkoumané realitě a ve správné chvíli ji odstínit od nepodstatných rozdílů. Formalizace slouží především k efektivní komunikaci vývojového týmu i vzhledem k zákazníkovi. Díky formalizaci můžeme jasně identifikovat a popisovat všechny prvky modelu. Během modelování bychom měli zamezovat existenci (dvojího) navzájem sporného tvrzení. (Pelánek 2011)

2.2 Kritéria hodnocení

Každý model můžeme hodnotit podle předem daných kritérií, kterými jsou konzistentnost, kompetentnost, úplnost, jednoduchost, rozšiřitelnost, výstižnost, uživatelská přívětivost a v neposlední řadě musíme uvažovat náklady na vytvoření a údržbu. Konzistentnost zkoumá, zda model správně zobrazuje modelovaný jev. Kompetentnost hodnotí, jestli model slouží zamýšlenému účelu. Úplností rozumíme zahrnutí všech důležitých aspektů. Jednoduchost popisuje, zda model obsahuje jen relevantní prvky. Rozšiřitelností je myšleno, zda je model vytvořen, aby bylo možné ho dále rozvinout. Výstižnost posuzuje, zda pro zachycení modelovaného jevu byly využity vhodné výrazové prostředky. Uživatelská přívětivost řeší přijatelnost vzhledem ke schopnostem a znalostem tvůrců i uživatelů. (Roques 2004)

2.3 Přístupy

Při modelování se řídíme přístupem shora dolů nebo zdola nahoru. Pokud využijeme přístup shora dolů, pak je systém definován jako celek a postupně ho rozkládáme na menší části. Nevýhodou tohoto přístupu může být špatné odhadnutí vzájemného působení částí systému, které se mohou ukázat složitější, než si autor původně myslel, což prodlužuje vývoj.

Při přístupu zdola nahoru začínáme tvorbu modelu od nejnižší úrovně a postupně spojujeme jednotlivé části do subsystémů, které následně tvoří celý systém. Při využití tohoto přístupu je výhoda ve znovupoužitelnosti naprogramovaného kódu. (Láska 2017)

2.4 Typy modelů

Modely lze dělit na základě mnoha kritérií. Jedním ze způsobů rozlišení je dle času na *statický* a *dynamický*. Statickým modelem popisujeme složení reálného objektu v ustáleném stavu, kde model není závislý na čase. Často se jím popisuje reálný objekt v rovnovážném stavu. Naopak dynamický model závisí na běhu času. Dynamický model popisuje reálný objekt v průběhu času. (Kořínek 2013)

Dále je možné modely rozdělit na *diskrétní* a *spojité*. Diskrétní modely se často využívají pro modelování určitých operací. Pomocí diskrétních modelů můžeme například zharmonizovat potřeby několika pracovišť nebo snížit operační náklady.

Takové modely či jejich provedené simulace poskytnou podrobné statistické výsledky. Naproti tomu spojitě modely fungují na sdružené úrovni s celkovými stavy systémů a jejich změnami v čase. Tímto systémem může být například nástroj pro podporu strategického řízení ve firmě. (Mládková 2005)

Modely můžeme také dělit na *deterministické a stochastické*, toto rozdělení je z hlediska chování systémů. Deterministický model dokládá i po opakování pokusů za totožných výchozích podmínek shodné chování. U tohoto typu modelu je známa jeho struktura, která je většinou popsána algebraickou či diferenciální rovnicí, ale i hodnoty parametrů jsou nepopíratelné. Pokud by do modelu vstoupil prvek nejistoty neboli náhody, pak můžeme říci, že model je *stochastický*. Ten se liší od deterministického modelu tím, že při totožných výchozích podmínkách se budou výstupy lišit. Pokud chceme popsat stochastický model, popisujeme ho rovnicí (soustavou rovnic), která obsahuje náhodné veličiny, nenáhodné veličiny a parametry. (Vlček a Chuchro 1999)

Dalším rozdělením modelů je na *jednoduché, neorganizované* či *komplexní*. U jednoduchých je malý počet částí, mezi kterými jsou pravidelné interakce. Při zkoumání těchto modelů využíváme předpoklady s logickým odvozováním, k závěrům dojdeme dedukcí. Jednoduché modely lze popsat malým počtem rovnic s jasnými závěry. Příkladem takového systému může být například páka nebo kolo. Neorganizovaný model má velký počet částí, mezi kterými jsou náhodné a nahodilé interakce. Tento model zkoumáme pomocí statistických metod. Například to může být pohyb částic plynu. Komplexní systémy se skládají z velkého počtu částí a interakce mezi nimi je uspořádaná a organizovaná. Komplexní modely zkoumáme pomocí indukce či simulace. Typickým příkladem může být mraveniště nebo ekosystém. (Pelánek 2011)

Pokud hovoříme o shodě modelu s originálem, pak modely dělíme na *izomorfní a homomorfní*. Izomorfní modely jsou ekvivalentní vzhledem k jejich předloze, systém je tedy přesnou kopií modelované reality. V případě homomorfního modelu je model pouze podobný předloze a podobnost lze sledovat pouze jednostranně směrem od modelu k předloze. Tento způsob je nejčastěji využíván při modelování. (Kořínek 2013)

3 Simulace

S modelováním úzce souvisí simulace, díky které je možné modelovaný systém s jeho chováním reprezentovat v reálném čase. V dnešním světě se s tímto pojmem shledáváme často, proto může mít více významů. Simulace se využívá, pokud je řešená úloha příliš složitá a komplikovaná, což vylučuje použití analytických technik. Simulační metody jsou silný nástroj vhodný pro analýzu a vylepšování komplikovaných procesů a systémů. Jestliže hledíme na pojem simulovat v širším pojetí, pak to může znamenat napodobovat či předstírat. Hledíme-li na tento pojem v užším pojetí, pak simulací rozumíme simulaci pomocí počítačového softwaru, která nás bude zajímat v této práci.

Simulaci lze pochopit jako tvorbu logického modelu reálného systému, modelu procesu rozhodování a provádění velkého počtu experimentů s modelem. Účelem simulace je popsat systém a rozpoznat jeho strukturu a funkce, nalézt řešení daného problému nebo odhadnout budoucí chování modelu. Jestliže se najde řešení problému, přichází návrh a ověření funkčnosti nově zadané struktury modelu. (Prokůpek 2006)

3.1 Charakteristika simulačního modelu

Simulační model slouží pro napodobení reality či odhadnutí, kam se současný zkoumaný stav vyvíjí do budoucnosti. Simulační model je zvláštní formou modelu. Základní prvky, ze kterých se skládá simulační model, jsou entity (transakce), aktivity a zdroje. Entity se mění v průběhu času, lze o nich říci, že jsou dynamické objekty.

Entita vstoupí do systému, následně provede nějakou činnost, kdy dočasně obsadí anebo spotřebuje zdroje a na závěr opustí systém. Každá entita má svoje vlastnosti, které se nazývají atributy. Stejně jako může entita měnit během simulace svou formu, může měnit i své atributy.

Proces se skládá z dílčích aktivit neboli činností. Činnost je měřitelná jednotka práce, kde je výstupem změna vstupního prvku do předem formulovaného výstupu. V modelu jsou jednotlivé činnosti neboli aktivity propojeny spojnicemi, které naznačují logický sled aktivit a zároveň vedou pohyb entit.

V neposlední řadě jsou základním prvkem zdroje, které entity využívají nebo spotřebovávají. Kapacita je základní atribut zdroje. (Kořínek 2013)

3.2 Využití simulace

Simulace mohou sloužit k několika účelům. (Hartmann 2015)

- Technická funkce – slouží pro zkoumání dynamiky systému, která je řešena sadou rovnic popisujících komplexní interakci. Lze říci, že simulace není lepší než její předpoklady.
- Heuristický nástroj – pokud na řešení problému nemáme algoritmus nebo jinou přesnější metodu, pak je možné využít simulaci. V tomto případě jde z velké části o přibližné řešení problému, které je založené na odhadu, intuici, zkušenostech nebo zdravém rozumu.
- Experiment – pomocí simulace lze substituovat reálný experiment. Tato simulace nejčastěji napodobuje jevy, které není možné provést reálně, například v závislosti na časovou náročnost nebo etické problémy.
- Nástroj sloužící experimentátorům – tato simulace poslouží k jejich experimentům. Může se jednat o inspiraci či předběžné vybírání testovaného systému nebo tak lze analyzovat reálné experimenty.
- Pedagogický nástroj – slouží pro pochopení komplikovaného procesu či systému, kterým může být například dopravní zácpa či chování davu.

Simulace je v praxi často využívána. Příkladem může být článek, který se zabývá jednoduchou prostorovou simulací městského plánování v menších obcích. Z článku se dozvíme, že kvůli přesnosti dosažených výsledků většina prostorových simulací vyžaduje velké objemy dat, proto kvalitní simulace se v praxi obvykle nepoužívají pro malá města (menší než 25 000 obyvatel). Cílem citované studie je poskytnout snadno použitelný nástroj pro menší obce. Za posledních 70 let byly vytvořeny různé prostorové simulace v oblasti dynamiky systému a modelování agentů. První z nich pochází z 60. let minulého století, kdy ji založil Forrester na Massachusetts Institute of Technology. Součástí studie je webová aplikace, která vyžaduje minimální manuální vstup uživatele. Případová studie pro centrum města

Herdecke (v Německu) ukázala, že zavedení nové události v severní části studované oblasti bylo nejvhodnější pro zvýšení prostorové atraktivity. V městě Herdecke žije přibližně 25 000 obyvatel. Správu města zajišťuje 265 pracovníků, což je v závislosti na velikosti města a této oblasti adekvátní. Vzhledem k počtu zaměstnanců není umožněno komplexní zpracování dat, programování a validace simulací městských systémů v rámci každodenní rutiny. Společnost SimUSys byla založena ve spolupráci s projektanty města Herdecke. Měřítkem simulace je míra vnímání blaha občanů v rámci města, což zaměstnanci SimUSys označili jako prostorovou přitažlivost. Aplikace je snadno ovladatelná v každodenních administrativních procesech v menších městech. (Mueller et al. 2018)

Dalším příkladem z praxe může být inteligentní dopravní informační systém založený na integraci internetu a technologie agentů. Zde je řešena doprava ve městech, která jsou během posledních let stále více přeplněná v důsledku zvyšující se popularity osobních automobilů. Největším problémem jsou tvořící se dopravní zácpy a nehody, které způsobují ztrátu času i znečištění životního prostředí. Proto je potřeba tuto situaci řešit. Popis možného řešení problému je v tomto dokumentu popsán pomocí inteligentního systému, který by měl za pomoci internetu věci městům ulevit. Celý projekt stojí na internetu věcí, který je založen na internetu, síťovém bezdrátovém snímání, sledování, monitorování, a detekční technologii, která rozpozná dopravní objekt. Systém dokáže dávat nejnovější informace o dopravě v reálném čase, které pomáhají vybírat řidičům optimální trasy. V dokumentu je navržena architektura, která spojuje internet věcí s agentovými technologiemi do jediné platformy. Dále je použita aktivní radiofrekvenční identifikace (RFID), technologie bezdrátových senzorů, vytváření sítí a podobně. Model popsáný v tomto dokumentu představuje přehledný model simulace provozu v NetLogo. Řada vědců se zabývala problémem inteligentního sledování a řízení dopravy. Bylo vyvinuto několik přístupů. Například návrh mechanismu predikce dopravního toku založenému na modelu fuzzy neuronové sítě v chaotickém dopravním toku časové řady (Pang a Zhao 2008). Nebo popis fuzzy logiky založené na agentových technologiích pro řízení provozu v situacích zahrnujících více přístupů a pohybů vozidel (Bhadra et al. 2014). Hlavním úkolem systému je detekce mobilních objektů a jejich umístění, identifikace a přenášení získaných údajů na

monitorovací a řídicí centrum pro zpracování. Navrhovaný systém se skládá z velkého množství RFID a senzorů, které vysílají bezdrátově. To vyžaduje vyšší bezpečnost, aby bylo chráněno masivní množství dat a soukromí uživatelů. Další zkoumaná oblast zpracovává a analyzuje velké objemy odlišných údajů ze systému pro vytváření aplikací, které zlepšují tok vozidel po celém městě. (Omar 2015)

3.3 Nástroje

Informační technologie se rychle vyvíjejí a lze říci, že pracovní činnosti jsou často spjaty s informačními technologiemi. Vznikají různé simulace a modely, kvůli nimž vzniklo mnoho různých programů pro tvorbu modelů a simulací. (Láska 2017) Testované softwary byly vybrány dle zkušeností autorky, které získala během studia a jeden speciální software na modelování silničního provozu. Následně bude popsáno několik takových programů, jsou to:

- NetLogo,
- AnyLogic,
- PTV Vissim.

3.3.1 NetLogo

NetLogo je multiagentní systém (MAS), ve kterém dochází k interakci mezi agenty navzájem nebo mezi agentem a prostředím. V takovém systému se pohybuje více než jeden agent v prostředí. Agenti jsou aktivní prvky v systému, jejichž činností je přechod ze stávajícího stavu do cílového, který je cílem řešení. Mezi základní charakteristiky agentů patří (Popelka et al. 2010):

- Autonomnost – agenti mohou samostatně promýšlet a řešit specifické úlohy bez nezbytnosti komunikace s okolím (dalšími agenty či prostředím),
- Sociální chování – mezi agenty mohou být vytvářeny sociální vazby, jejichž účelem je dosažení společných cílů, důležitá je možnost přizpůsobení modelu pohledu na okolní svět dle modifikací vzájemných propojení,

- Reaktivita – agenti aktivují události, na které jsou schopni reagovat dle předem definovaných množin akcí,
- Intencionalita – agenti pracují s dlouhodobými cíli, které jsou rozkládány na „podproblémy“, jaký cíl agent plní závisí na jeho osobní motivaci pro dosažení určitého cíle.

Vazby mezi agenty jsou zpravidla volné a mění se či zanikají podle situace. Každý agent se v konkrétním čase nachází v lokálním stavu, ve kterém pojmá všechny jemu dostupné informace, jež označujeme jako znalost. Další součástí multiagentních systémů je prostředí, čímž chápeme vše okolo agentů. Agenti s prostředím přicházejí do styku či s ním bezprostředně interagují.

NetLogo je prostředí určené pro modelování multiagentních komplexních systémů, které je psané v Javě. Prostedí NetLogo vyvinul v roce 1999 Uri Wilensky a o jeho následný vývoj se stará Centrum e-learningu a počítačového modelování na Chicagské Northwestern Univerzity. V NetLogo se využívá programovací jazyk Logo.

NetLogo disponuje intuitivním prostředím s možností vizualizovat simulace. V programu najdeme nástroj pro import či export datových souborů, které jsou využívány v aplikacích geografických informačních systémů. Tento program je vhodný především pro komplexní systémy, které se během času vyvíjejí. Vývoj modelovaného systému plyne z chování mnoha agentů, které je definováno souborem pravidel. K tomuto programu je mnoho tutoriálů, dokumentace a široká knihovna modelů, které lze využívat. (Wilensky 2018)

3.3.2 AnyLogic

AnyLogic je simulační modelovací nástroj, který je vyvinut firmou AnyLogic Company, kterou byl poprvé prezentován v roce 2000. AnyLogic slouží pro vytváření diskrétních a dynamických modelů, ale i za pomoci agentů. Tento program umožňuje importovat CAD výkresy a vizualizovat modely. Tyto i jiné vizualizace dokáže simulovat ve 2D i 3D. AnyLogic disponuje nástrojem, který dokáže importovat soubory, se kterými se setkáváme v aplikacích geografických informačních systémů. Program AnyLogic využívá několik velkých firem například

Vodafone, Google nebo některé firmy zabývající se automobilovým průmyslem (např. Ford a Wolksvagen).

Simulace v AnyLogic se skládá ze čtyř stěžejních oblastí, kterými jsou Stock a Flow Diagrams, Statecharts, Action charts a Process flowcharts. Stock a Flow Diagrams čili stromové a průtokové diagramy, tyto diagramy jsou používány pro modelování systémové dynamiky. Statecharts se využívají často v modelování za pomoci agentů, kteří mají definované chování a používají se pro modelování diskrétních událostí jako je například selhání stroje. Action charts slouží pro definování algoritmů, ale mohou být využity při modelování diskrétních událostí, jako jsou například přesměrování hovorů, či modelování za pomoci agentů během rozhodování agentů. Process flowcharts slouží pro definování procesů v modelování událostí. AnyLogic má i své knihovny. Mezi standardní patří knihovna modelování procesů, chodců, železniční, fluidní, manipulace s materiálem nebo silniční dopravy. (Anylogic 2019)

3.3.3 PTV Vissim

PTV Vissim je softwarový balík, který je vyvíjen společností PVT Planung Transport Verkehr AG. Základ pro tento software byl vytvořen v roce 1974, kdy Rainer Wiedemann na Univerzitě Karlsruhe sestavil základní dopravní model, který zohledňuje fyzické i psychické aspekty řidičů. PTV Vissim je program pro modelování silniční dopravy, a jelikož je pro tyto modely přímo určen, je možno silniční dopravu simulovat přesně. Tento program nabízí dopravní analýzu a prognózu i optimalizaci signálů a dopravní odhady. Z modelu je několik výstupů, kterými jsou například sběr dat, cestovní čas, časové zdržení, délka kolon a záznamy o vozidlech. Tento program je diskrétní, stochastický a na časových krocích založený mikroskopický model. Využití aplikace může být např.:

- výzkumné projekty zaměřené zejména na kapacitu komunikací,
- optimalizování dopravně-projekčních návrhů křižovatek,
- mikroskopické simulace jízd vozidel MHD,
- a posouzení mezikřižovatkových úseků.

Prostřednictvím programu PTV Vissim jsou tvořeny tzv. mikrosimulace, ve kterých je každá entita reality modelována samostatně. Dále je možné v programu paralelně simulovat několik typů dopravy, které na sebe následně vzájemně působí. Jsou to například automobily, městská hromadná doprava, cyklisté, ale i rikši. (PTV Vissim 2019)

3.3.4 Výběr programu pro simulaci

Pro vytvoření modelů bylo vybráno prostředí NetLogo. Tento nástroj byl autorkou vybrán především pro dřívější zkušenosti z výuky a pro jeho uživatelský přívětivé prostředí. Dále pro jeho univerzálnost, dostupnost a aktuálnost. Nyní se tento modelovací nástroj znovu vrací do obliby a začal se využívat na různé projekty.

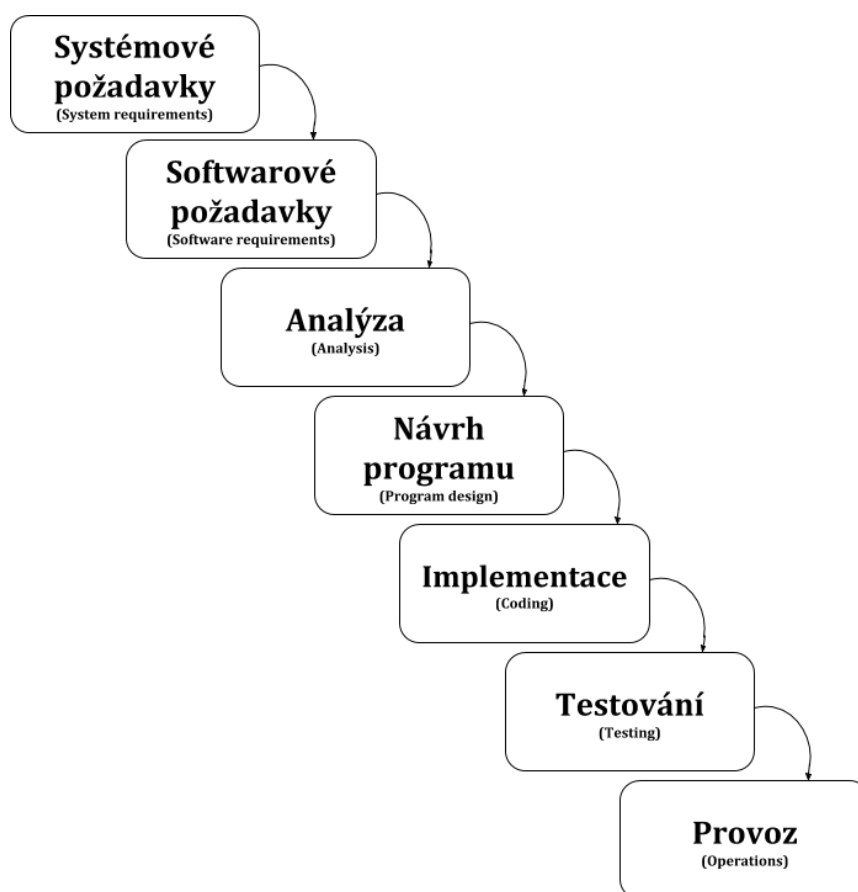
Například studie z roku 2020 „Model Study of Traffic Congestion Impacted by Incidents“ řeší dopravní situaci v hlavním městě Indonésie Jakarta, především zahlcení placených úseků kvůli dopravním nehodám a jejich negativních následků na plynulost dopravy. Autoři vycházeli z modelu “Traffic 2 lanes”, který vytvořil Uri Wilensky. V NetLogo vytvořili model, definovali jako agenty vozidla. V modelu vytvořili pravidla, kterými se provoz v modelu řídil. Následně provedli simulace, při kterých zjistili několik možností, jak zajistit plynulou dopravu na úsecích s mýtem. Jedním ze zlepšení bylo snížení aut na mýtných úsecích. Další možností je zvýšit počet jízdnic pruhů na placených úsecích. (Triastanto a Utama 2019)

Další studie, ve které je popsáno spojení modelovacích nástrojů NetLogo s nástrojem MODFLOW/SEAWAT, je „A coupled simulation architecture for agent-based/geohydrological modelling with NetLogo and MODFLOW“ z roku 2019. Zabývá se sociálně-ekonomickými systémy. Autoři se snaží reprezentovat propojení NetLogo a MODFLOW/SEAWAT a reprezentují to na Aquifer Thermal Energy Storage (ATES). (Jaxa-Rozen et al. 2019)

4 Vodopádový model

Vodopádový model (z angl. waterfall model) je součástí souboru tzv. rigorózních metodik neboli těžkých metodik, které jsou historicky největší skupinou metodik SW vývoje. Tyto metodiky předpokládají, že procesy při tvorbě ICT lze popisovat, plánovat, řídit a měřit. Mezi rigorózní metodiky se dále řadí například inkrementační, spirálový nebo prototypový model. (Buchalceková 2005)

Vodopádový model byl prvním zveřejněným modelem vývoje procesu softwaru. Byl vyvinut za účelem vyrovnat se rostoucí komplikovanosti produktů v leteckém průmyslu. Winston W. Royce jej poprvé popsal ve svém článku v roce 1970, který nesl název „Managing the Development of Large Software System“. V článku Royce popisuje tento model sedmi základními fázemi, které jsou vyobrazeny níže (Obrázek 3). (Royce 1970)

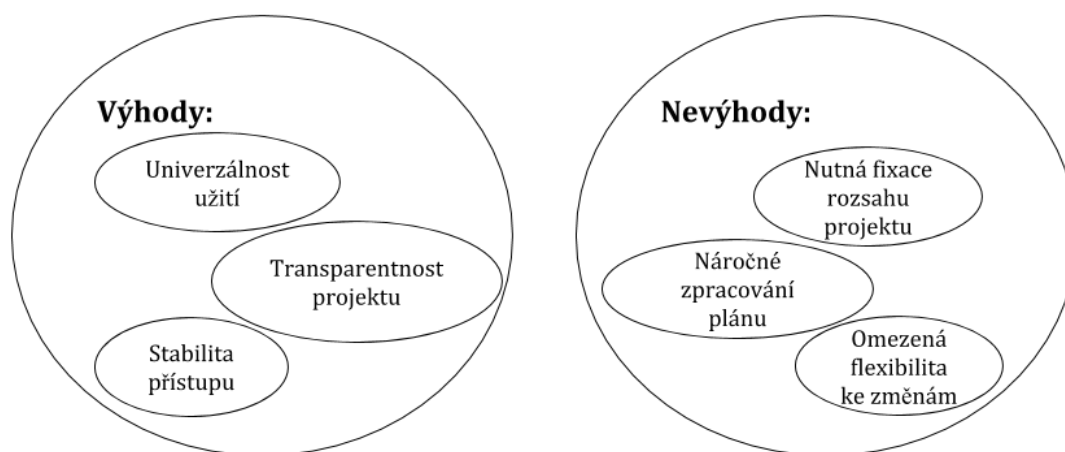


Obrázek 3 – Sedm základních fází Vodopádového modelu (Royce 1970)

Vodopádový model je hierarchický vývojový postup, kde můžeme sledovat ustavičně se svažující tok fázemi modelu (viz obrázek č. 3). Zpravidla se začíná specifikací požadavků na model a jejich následnou analýzou. Po těchto fázích přichází na řadu návrh modelu, ze které se plynule přechází k implementaci. Po implementaci probíhá testování a případná úprava modelu. Na závěr je model uveden do provozu a využit na určený záměr. (Sommerville 2013; Dvořák a Mareček 2017)

4.1 Výhody a nevýhody

Waterfall - výhody a nevýhody



Obrázek 4 – Waterfall výhody a nevýhody (Dvořáček a Mareček 2017)

Výhodou vodopádového modelu je snadná pochopitelnost stanoveného postupu vývoje. Za podmínky, že je dodržen postup vývoje, je zajištěna snadná integrace systému. Nevýhodou vodopádového modelu je umístění fáze testování za fázi vývoje, jestliže je nalezena chyba, je zpravidla složitější ji opravit, než pokud je prováděné průběžné testování. Pokud je projekt jasně specifikovaný a máme jistotu, že požadavky se v průběhu nebudou podstatně měnit, pak je vhodné použít vodopádový model. Pro jeho snadné nasazení je dost využívám, ovšem většinou v modifikovaných variantách. Jedna z variant modifikovaného vodopádového přístupu nazývaný „Rolling Wave“, ve kterém se plánuje po vlnách. Podstatou je plánovat detailně pouze etapy, které jsou v blízké budoucnosti a ty, které jsou vzdálenější, se neplánují detailně, ale pouze povrchně. (Sommerville 2013; Dvořák a Mareček 2017)

5 Vizuální analýza

Vizuální analýza je nezbytná v aplikačních oblastech, kde je třeba zpracovat a současně analyzovat velké množství informací. Vizuální analýza spojuje několik oblastí včetně informatiky, kognitivní vědy, grafického designu či společenských věd. Napomáhá při vývoji vědy a technologie při řešení problémů, které se zdají být neřešitelné. Při hledání výsledku využívá analytické myšlení, transformaci dat a reprezentaci výpočtu za pomoci vizualizace či simulace. (Kielman et al. 2009)

Vizuální analýzu můžeme považovat za vědu o analytickém myšlení rozšířenou o interaktivní vizuální rozhraní. Prostřednictvím vizuální analýzy jsou interpretovány nové výpočetní nástroje založené na interaktivních technikách a vizuální reprezentaci dat, které umožňují lidem lépe diskutovat problém. Na základě vizuální analýzy lze budovat strategické nebo taktické vizuální analytické technologie pro analýzu hrozeb, jejich prevenci či reakci na ně. Analytické uvažování je stěžejním bodem analytického úkolu aplikovat lidské úsudky při dosahování závěrů z kombinace důkazů a předpokladů. (Thomas 2019)

Vizuální analýza se snaží o spojení technik vizualizace informací a počítačové transformace a analýzy dat. Vizualizace informací představuje přímé rozhraní mezi uživatelem a strojem, což umocňuje lidské kognitivní schopnosti šesti základními způsoby (Thomas 2019):

- zvýšením kognitivních zdrojů, například pomocí vizuálního zdroje pro rozšíření lidské pracovní paměti,
- omezením vyhledávání, například reprezentací velkého množství dat v malém prostoru,
- zlepšením rozpoznávání vzorců, jako je to, když jsou informace organizovány v prostoru svými časovými vztahy,
- podporou snadného vnímání vztahů, které je jinak obtížnější vyvolat,
- vnímavým sledováním velkého počtu potenciálních událostí,

- poskytnutím manipulovatelného média, které na rozdíl od statických diagramů umožňuje zkoumání prostoru hodnot parametrů.

5.1 Historie

Metody analýzy, jako například statistika a dolování dat, se vyvíjely nezávisle na vizuálních a interakčních technikách. Jeden z nejdůležitějších kroků byla potřeba přejít od analýzy potvrzujících dat (tzn. pomocí grafů a jiných vizuálních reprezentací k současným výsledkům) k analýze průzkumných dat (tzn. interakce s daty, výsledky), která byla poprvé uvedena ve statistikách výzkumné komunity Johna W. Tukeye v jeho knize z roku 1997, která se jmenuje Průzkumná analýza dat. (Keim 2010)

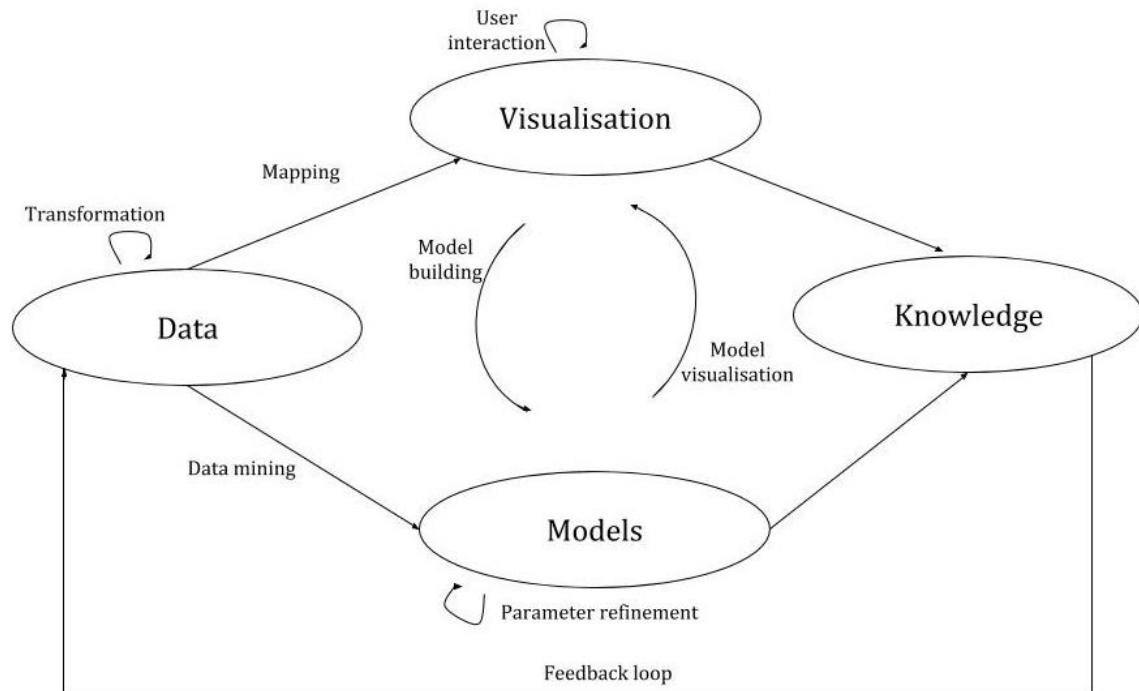
5.2 Aplikace vizuální analýzy

Do této kapitoly je čerpáno z (Keim 2010). Proces vizuální analýzy je zaměřen především na propojování analytických metod s vizuální reprezentací. Vizuální analýza nám představí nejpodstatnější aspekty dat. Vzhledem k masovým datovým souborům je vizuální analýza nástroj, který dokáže z takovýchto dat přehledně vyzdvihnout podstatné údaje, které se následně dají zkoumat podrobněji.

Oblastí, ve kterých se vizuální analýza využívá, je mnoho, nejčastěji jsou to fyzika, astronomie, lékařství, biologie, obchod nebo doprava. Například při monitorování počasí je nashromážděno velké množství dat prostřednictvím senzorů na zemi či na moři, ale také informace získané ze satelitů. Zjištěné informace je potřeba správně zpracovat a vhodně reprezentovat závěry. Při prezentaci dat je vhodný vizuální přístup, který pomáhá k interpretaci obrovského množství dat.

Prvním krokem při analýze je předzpracování a transformace dat pro tvorbu vizualizací a další zkoumání. Dále je nutné provést například čištění dat, normalizaci nebo integraci různých zdrojů dat. Celý proces kombinuje metody automatické a vizuální analýzy s úzkou vazbou na interakce člověka, což má za cíl získání znalostí z dat. Následující diagram zobrazuje abstraktní přehled fází (reprezentují oválné bubliny) a jejich přechodů (šipky) v procesu vizuální analýzy.

Visual Data Exploration



Obrázek 5 – Visual Data Exploration (Keim 2010)

V aplikačních scénářích se různorodé zdroje dat sjednocují ještě před použitím vizuálních a automatických metod analýzy. Na začátku je potřeba připravit data a transformovat je do potřebného formátu, aby bylo možné s nimi dále pracovat a zkoumat je, což nám v diagramu popisuje zpětná šipka „Transformation“, která se vztahuje k abstraktní fázi „Data“. Dále probíhá dolování dat za pomoci výpočetních metod pro automatické získání důležitých informací ze surových dat díky algoritmům automatické analýzy. Následuje tvorba modelu, po které je zapotřebí, aby analytik model vyhodnotil a následně ho vylepšil. Cílem celého procesu je vizualizace dat, jež pomáhá analytikům při úpravě parametrů a s výběrem dalších analytických algoritmů. Interakce uživatele s vizualizací napomáhá k optimalizaci modelu a odhalení nepatrných informací, kterými mohou být například bližší pohled na data z dané oblasti nebo další vizuální pohledy na data.

5.3 Vizuální analýza v dopravě

Očekává se, že Švýcarsko během příštích 35 let zvýší svou populaci o 25 % (dle údajů Švýcarského statistického úřadu). Jedním z hlavních cílů země je udržet status quo z hlediska ubytování, dostupnosti služeb a kvality měst. K řešení situace se analyzovaly možné budoucí scénáře a vyhodnocovaly se výsledky různých politik, aby byla poskytnuta odpovídající podpora budoucímu nárůstu počtu obyvatel. Změny by mohly zahrnovat například modernizaci infrastruktury a plánování veřejných služeb. V této studii byly vyhodnoceny dva scénáře, kde bylo cílem zhodnotit možné plánování měst, zaměřené především na využití půdy a mobilitu. Výsledkem bylo, že způsob využití půdy zaslouží velkou pozornost. Zvýšení populace vede k vyšší frekvenci dopravy, proto je důležité zvýšit využití veřejné dopravy, což lze udělat několika způsoby. První způsob je snadný, a to vyrovnat dobu navíc strávenou při cestování snížením nákladů na dopravu nebo zvýšit poplatky za soukromou dopravu. Tato možnost, ale nebyla shledána jako dobrá především proto, že by provoz veřejné dopravy byl nevýhodný, klesal by komfort při dopravě a hrozilo i zhoršení infrastruktury. Nehledě na to, že zvýšit náklady na soukromou dopravu by ve Švýcarsku nepřineslo ovoce, protože ve srovnání s ostatními náklady domácností jsou výdaje na dopravu vnímány jako nízké. Druhou možností je zlepšit infrastrukturu ve městě. Vzhledem k vysokým nákladům, ani tato možnost nepřipadala v úvahu. V neposlední řadě by mohlo pomoci zavedení nových technologií, které můžou snížit čas strávený cestováním ve veřejné dopravě v porovnání se soukromou dopravou, což by následně přilákalo více lidí, kteří by využili veřejnou hromadnou dopravu. Toto řešení bylo shledáno nejpřínosnějším, protože umožní rozvoj infrastruktury veřejné dopravy. Technologie omezí dobu cesty, ale i sníží problémy spojené s vlastnictvím vozidla (náklady na údržbu, pojištění). (Marini et al. 2018)

Studie se zaměřuje na snížení přetížení silnic, které může být vyřešeno za pomoci uzlů městské hromadné dopravy, které usnadňují dopravu ve městě. Pro názornou ukázkou slouží nástroje pro interaktivní vizualizaci (ITS), kde vytvoří interaktivní vizualizaci v reálném čase podpořenou pokročilou analýzou dat za pomoci algoritmů. V citované práci je využita interaktivní platforma pro vizuální

analýzu, která umožní průzkum historických dat a předpovídat budoucí provoz prostřednictvím interaktivního rozhraní. Platforma podporuje několik technik analýzy dat, například vizualizace či shlukování v cestách, případné anomálie nebo předpověď provozu. Přínosem této práce je vizuální analytický nástroj. (Kalamaras et al. 2018)

Samotný rámec poskytuje platformu tranSMART nový analytický engine, který je testovatelný, udržovatelný a rozšiřitelný. Poskytuje také řadu předem připravených vizuálně-analytických pracovních toků. Práce se v části zaměřuje na jeden z těchto pracovních postupů, konkrétně na interaktivní mapu tepla, jako příklad ilustrující interaktivní a dynamickou povahu platformy. „heat map“ SmartR poskytuje řešení zavedením „lazy-loading ap-proach“, kde se zpočátku zobrazuje pouze 100 nejvýznamnějších genů podle kritérií definovaných uživatelem. Toto snížení zobrazených dat nám umožňuje zacházet s jednotlivými poli „heat maps“ jako s pohyblivými dynamickými prvky, spíše než se statickým obrazem. To vede k několika užitečným funkcím, jako je například možnost změny klastrování v provozu, výběr různých barevných sad pro různé datové typy a přístupnost (barevná slepota) nebo třídění řádků a sloupců mapy tepla. Kromě interaktivní „heat maps“, je tu také několik dalších běžně používaných analýz např. korelační analýza, kde si uživatel může vybrat oblasti na srovnání pozemku a získat aktualizovanou analýzu nebo krabicové grafy. Během implementace nelze vyčerpat všechny možné analýzy. (Herzinger et al. 2017)

6 Analýza simulačních přístupů

Pro analýzu byla zvolena problémová křižovatka Mileta v Hradci Králové. Zkoumaná křižovatka se nachází na Gočárově okruhu, jedná se o hlavní tah ze směru Praha/Brno. Vzhledem k tomu, že je monitorován stále hustší a hustší provoz, převážně v ranních a odpoledních hodinách, dochází k dlouhým kolonám a dopravním zácpám po celém Hradci Králové.

Pro účely diplomové práce budou nastíněny a porovnány dvě varianty, jak celou problematiku vyřešit a zajistit tak efektivnější a plynulejší dopravu v Hradci Králové. První zkoumanou variantou je stávající řešení křižovatky (varianta I.), která není vhodná a je žádoucí toto řešení do budoucna vylepšit. Jednoduší na realizaci ze dvou návrhů řešení je varianta s přidáním odbočovacích pruhů, které nejsou ovlivněny světelnou signalizací (varianta II.). Dále je v této variantě počítáno s podchody pro chodce. Druhou možnou variantou zlepšení je vybudování úplně nové, moderní křižovatky Mileta. Kde je prvním krokem ke zvýšení plynulosti dopravy odstranění semaforu na hlavním tahu vybudováním podúrovňové silnice. Dalším faktorem je vytvoření podchodů pro chodce a zaručení jejich větší bezpečnosti v pohybu přes křižovatku. Pro účely odbočování se nabízí možnost vyvýšeného kruhového objezdu, který by neovlivňoval provoz po hlavním tahu v obou směrech (varianta III.). (Volejníková 2013) Analýza proběhne za pomoci softwaru (dále SW) NetLogo.

6.1 Stávající stav – varianta I.

Aktuálně je situace na křižovatce Mileta řešena semafor, které nejsou efektivní, co se týká plynulosti dopravy. Celé situaci nepomáhá ani fakt, že křižovatka se nachází nedaleko Fakultní nemocnice Hradec Králové. V případě potřeby rychlé záchranné pomoci a permanentním dopravním zácpám je velmi obtížné i uvolnění uličky pro potřeby záchranné služby a rychlému výjezdu sanitky či akutního převozu pacienta do nemocnice. Dále se na křižovatce nalézají 6 přechodů pro chodce, tedy 6 intervalů semaforů, které je potřeba důkladně naplánovat, tak aby nedocházelo k velkým prodlevám v plynulosti dopravy. Pro bezpečí chodců není situace nikterak příznivá, neboť v případě, že chtějí přejít přes celou křižovatku, jsou

nuceni zastavit se a dlouho vyčkávat na „ostrůvku“ mezi hlavními pruhy. V případě jakékoliv dopravní nehody, byť malé, může dojít k mnohem většímu riziku a ohrožení chodců. To, že interval na přejítí takto frekventované silnice je velmi dlouhý, nepomáhá ani fakt, že často lidé nestíhají autobusy městské hromadné dopravy (MHD). Obrázek 6 vyobrazuje stávající stav křižovatky. Pohled od severovýchodu.



Obrázek 6 – Stávající stav (varianta I.) (webkameryzive.cz)

Nynější stav ke křižovatce přivádí z každého směru tři pruhy, které jsou plně aut a tvoří se kolony, především ze směru Pardubice/Praha a Brno. Zároveň situaci nepomáhá, že ze směru od Brna a od FNHK slouží pruh na odbočení zároveň i pokračování v jízdě rovně, což zhoršuje plynulost dopravy. Ale ve špičce jsou vytvořeny kolony ve všech směrech a znemožňují plynulý průjezd Hradcem Králové. Vedení města se dlouhou dobu snaží situaci řešit, ale zatím se žádný z jejich návrhů nerealizuje.

6.2 Nové řešení – varianta II.

Obecně jsou známy dvě varianty zlepšení situace křižovatky Mileta. Jeden návrh zahrnuje rekonstrukci křižovatky vybudováním odbočovacích pruhů a podchodů (varianta II.). Na tomto návrhu je atraktivní především cena, která je třetinová oproti variantě s nadúrovňovým kruhovým objezdem, ale problém

světelné křižovatky řeší pouze z části. (Volejníková 2013) V roce 2013 byla odhadovaná cena 100 miliónů korun. V roce 2020 odhad ceny tohoto provedení křižovatky dosáhl 225 miliónů korun. (ZÁLESKÝ 2020) Provozu na této křižovatce pomůže vybudováním odbočovacích pruhů, které by byly vybudovány ze směru od Pardubic na FNHK, od UHK na Pardubice a od FNHK na silnici směr Futurum. V této variantě je uvažováno s třemi podchody, které budou řešit pohyb chodců a cyklistů v křižovatce, čím zvýší jejich bezpečnost a plynulost při pohybu v křižovatce. Tyto podchody budou pod silnici od UHK, od FNHK a pod silnici ze směru od Pardubic. Obrázek 7 ukazuje návrh varianty II. z jihozápadního pohledu.



Obrázek 7 - Nový stav křižovatky Mileta – varianta s odbočovacími pruhy (Záleský 2020)

6.3 Nové řešení – varianta III.

Varianta III. je výstavba nadúrovňového kruhového objezdu, pod kterým povede hlavní tah Praha-Ostrava. Tato varianta je přibližně dvakrát dražší než varianta II., předpokládaná cena v roce 2013 byla 300 milionů korun. Nyní by cena provedení byla vyšší. (Volejníková 2013) Jestli přestavba křižovatky pomůže Hradci Králové, může napovědět vizualizace. V modelech lze vyzkoušet různé varianty řešení a pomůžou k rozhodnutí se pro optimální z nich. Tento nový návrh křižovatky zabezpečí plynulý průjezd úsekem okruhu a zároveň umožní bezproblémový a plynulý průchod chodců, kteří se budou moci bezpečně přecházet z jedné strany na druhou za pomoci podchodů. Při návrhu byl střed kruhového objezdu využit jako park, který alespoň částečně zpříjemní „betonové“ prostředí křižovatky. Kruhový objezd je plánovaný nad úrovní nynější křižovatky a jejího okolí, proto budou podchody pod kruhovým objezdem v úrovni okolní krajiny, protože kruhový objezd bude postaven nad úrovní krajiny. Obrázek 8 představuje návrh nového řešení křižovatky s kruhovým objezdem z jihovýchodního pohledu.



Obrázek 8 – Nový stav křižovatky Mileta – varianta III. (Volejníková 2013)

6.4 Modelovací nástroj – NetLogo

Pro vizualizaci křižovatky Milety (a jejích obměn) byl zvolen nástroj NetLogo, pro jeho uživatelskou přívětivost, univerzálnost, dostupnost a aktuálnost.

Modely byly vytvořeny ve verzi NetLogo 6.1.1. NetLogo verze 6.1.1 běží na Windows Vista a novějších, Mac OS X 10.8.3 nebo novější nebo na standardních distribucích Linuxu založených na Debianu a Red Hat.

Simulace v NetLogo do videí na youtube pro potřeby dotazníku byla nahrána na přenosném počítači značky HP model ZBook 17 G4 s operačním systémem Microsoft Windows 10 Pro a typem systému x64-based PC. Na následujícím obrázku jsou některé parametry notebooku, na kterém byly spuštěny modely v NetLogo při natáčení

Procesor	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 2808 Mhz, jádra: 4, logické procesory: 8
Verze systému BIOS/Datum	HP P70 Ver. 01.15, 22.01.2018
Verze SMBIOS	3.0
Verze integrovaného řadiče	70.102
Režim systému BIOS	UEFI
Výrobce základní desky	HP
Základní deska	8270
Verze základní desky	KBC Version 46.66
Role platformy	Přenosný počítač
Stav zabezpečeného spouštění	Zapnuto
Konfigurace PCR7	Pro zobrazení je nutné zvýšení úrovně oprávnění
Adresář systému Windows	C:\Windows
Systémový adresář	C:\Windows\system32
Spouštěcí zařízení	\Device\HarddiskVolume3
Národní prostředí	Česko
Vrstva HAL (Hardware Abstraction Layer)	Verze = "10.0.18362.752"
Uživatelské jméno	DESKTOP-AP4E4QO\milan.nosil
Časové pásmo	Střední Evropa (běžný čas)
Nainstalovaná fyzická paměť (RAM)	16,0 GB

Obrázek 9 – Parametry notebooku, na kterém byla provedena simulace (vlastní zpracování)

V agentovém prostředí byly vytvořeny tři simulace možného vzhledu křižovatky Mileta. Aktuální světelná křižovatka (varianta I.), kde je vyobrazen pouze pohyb aut v křižovatce a chodci, cyklisti nejsou uvažováni, protože pro sledování pohybu aut je můžeme vynechat. Varianta II. křižovatky uvažuje odbočovací pruhy a částečné zachování křižovatky se semaforem. Varianta III. počítá s víceúrovňovým kruhovým objezdem. V následující části jsou uvedeny ukázky kódu, celé kódy jsou uvedeny v příloze.

6.4.1 Agenti a prostředí

V modelovacím nástroji NetLogo se pracuje především s interakcí mezi agenty a prostředím. Agenti do jisté míry jednají autonomně a snaží se na základě senzorů a aktuátorů dosáhnout efektivně svého cíle, projet křižovatkou bez kolize s ostatními agenty. V NetLogo jsou známy čtyři typy agentů, kterými jsou turtles (želvy, jediní pohybující se agenti), patches (políčka, mají každé pevnou pozici v modelu), links (spoje) a observer (pozorovatel). Prostředí v multiagentním systému (MAS) hraje důležitou roli. Prostředí může být využito pro správu agentů, kdy umožňuje činnost agentů v modelu.

Agenti v modelu jsou reaktivními, což znamená, že reagují na vnější vjemy v kombinaci s jejich vnitřním stavem. Všechny tři modely obsahují základní agenty v podobě „cars“, kteří představují auta pohybující se v modelu. Křižovatky se semaforey zahrnují další agenty pro účely fungování semaforů.

```
breed[cars car] ;agenti
breed[lightsFuturum lightFuturum] ;semafor
breed[lightsUHK lightUHK] ;semafor
breed[lightsHospital lightHospital] ;semafor
breed[lightsPardubice lightPardubice] ;semafor
```

Zdrojový kód 1 - Agenti ve Variantě I. (vlastní zpracování)

6.4.2 Proměnné

V NetLogo rozeznáváme tři typy proměnných – proměnné globální, proměnné želv a proměnné políček. Globální proměnná má pouze jednu hodnotu a může k ní přistupovat každý agent. Mimo globální proměnné můžeme v NetLogo využívat i lokální proměnné, které jsou vymezeny pouze v jedné proceduře, ve které také mohou být použity. Proměnné želv a políček mohou naproti tomu být jedinečné a jsou přidělovány každé želvě/políčku zvlášť. Mnoho základních proměnných želv a políček je v NetLogo již začleněno. Pokud se jedná o proměnnou přiřazenou políčku je před názvem uvedeno „p“ například souřadnice. Pokud se souřadnice týká želvy, značíme „xcor“, ale jestliže je to souřadnice políčka značíme „pxcor“. Dalšími proměnnými jsou například „color“ a „pcolor“ nebo „label“ a „plabel“. Uživatel si může sám definovat vlastní proměnné, ať už globální, lokální, želv nebo políček.

Agenti fungují se sadou deklarovaných proměnných, které určují jejich vlastnosti. Jejich hodnotu nastavujeme příkazem „set“. V modelech mají agenti křižovatek se semaforey stejnou sadu uživatelem definovaných proměnných.

```
cars-own[
  meaning-car      ;označení aut z určitého směru
  time-in-crossroad ;čas strávený na křižovatce - počítá se podle ticks
  preference-give  ;auta, která dávají přednost
  preference-has   ;auta, která dostávají přednost
  variable         ;pomocná proměnná při odbočování (otáčení auta do správného
                  ;směru)
]
```

Zdrojový kód 2 - Deklarované proměnné v modelech se světelnou křižovatkou – Varianta I. a II. (vlastní zpracování)

Model s kruhovým objezdem má jinou sadu uživatelem definovaných proměnných.

```
cars-own[
  meaning-car      ;označení aut z určitého směru
  time-in-crossroad ;čas strávený na křižovatce - počítá se podle ticks
  rotated          ;otočení aut o potřebné stupně při najíždění na kruhový objezd
  not-turning      ;proměná, která načítá, že auto nesjelo na sjezdu z
                  ;kruhového objezdu po určitém počtu auto musí sjet, aby
                  ;nejezdilo stále dokola)
  invisible        ;auta, která podjíždí pod kruhovým objezdem mají nastavené
                  ;hodnotu 1, ostatní mají nastavenou hodnotu 0
  original-color   ;uchovává barvu auta, je to pro potřeby "objevení" po podjetí
                  ;pod kruhovým objezdem
]
```

Zdrojový kód 3 - Deklarované proměnné v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)

6.4.3 Procedury

Procedury jsou příkazy a reportéry, které určují agentům, co mají dělat. Procedury jsou definované uživatelem a vždy začínají slovíčkem „to“ a končí slovíčkem „end“. Pokud je jednou procedura definovaná, lze ji dále bez omezení v programu používat. NetLogo má také mnoho začleněných příkazů a reportérů, kterým se říká primitiva. Příkladem primitiva je „create-turtle“ (lze použít v programu i zkratku „crt“), tento příkaz vytvoří určený počet nových želv.

Základní procedurou v modelech je procedura „SETUP“, která model dostane do základního nastavení. V této proceduře se používá funkce „clear-all“, která vše vymaže a začíná se tzv. s čistým štítem. Lze použít i zkratku „ca“. Dále se v ní nuluje počet kroků tzv. „ticks“ primitivem „reset-ticks“, která určuje, jak dlouho je model v čase puštěn. Dále jsou zde použity další dvě vestavěná funkce NetLogo „set“, který nastaví hodnotu proměnné a „import-drawing“, která vloží jako pozadí modelu vybraný obrázek. Ostatní jsou procedury definované uživatelem a pomáhají správně nastavit model v čase 0. K proceduře „setup“ je v Interface vytvořeno tlačítko, přes které se procedura spustí.

```

to setup
  ca
  reset-ticks
  draw_background      ;vykreslení pozadí
  draw_cars            ;první auta
  set_turning_points   ;nastavení turning-point
  traffic_lights       ;nastavení semaforů
  set_preference_patches ;preference - nastvení pro řízení v křižovatce
  set lights-change-tick 0 ;nastavení pomocné proměnné pro řízení semaforů
  set average_time_spent_crossroad 0 ;původní nastavení proměnné na 0
  import-drawing "Varianta_1_původní křižovatka.png" ;vložen obrázek jako
  ;pozadí modelu
end

```

Zdrojový kód 4 - Procedura „SETUP“ v modelech se světelnou křižovatkou - Varianta I. a II. (vlastní zpracování)

```

to setup
  ca
  reset-ticks
  draw_background      ;vykreslení pozadí
  draw_cars            ;první auta
  set_turning_points   ;nastavení turning-point
  set_invisible_points ;nastavení invisible-point
  traffic_roundabout   ;vykreslení kruhového objezdu
  set average_time_spent_crossroad 0 ;původní nastavení proměnné na 0
  import-drawing "Varianta_3_kruhový objezd.png" ;vložen obrázek jako
  ;pozadí modelu
end

```

Zdrojový kód 5 - Procedura „SETUP“ v modelu s kruhovým objezdem - Varianta III. (vlastní zpracování)

Další základní procedurou je procedura „GO“, která zahájí běh modelu. V této proceduře je určeno, které kroky budou provedeny během chodu modelu. V modelu křižovatek je zde okázáno na další procedury, které řeší pohyb aut a chod semaforů.

```

to go
  move_cars           ;pohyb aut v modelu
  creat_cars         ;tvorba aut v modelu během chodu
  traffic_light_set_orange ;procedura pro řízení semaforů
  set_lights_interval ;procedura pro řízení semaforů
  set_preference_cars ;preference - nastvení pro řízení v křižovatce
  ask cars [         ;čas strávený v křižovatce
    set time-in-crossroad (time-in-crossroad + 1)
    ;set label time-in-crossroad
  ]
  set average_time_spent_crossroad ((sum [time-in-crossroad] of
cars))/(count cars)
; nastavení proměnné - průměrný čas strávený na křižovatce
tick
end

```

Zdrojový kód 6 - Procedura „GO“ v modelech se světelnou křižovatkou - Varianta I. a II. (vlastní zpracování)

```

to go
  move_cars           ;pohyb aut v modelu
  creat_cars         ;tvorba aut v modelu během chodu
  set average_time_spent_crossroad ((sum [time-in-crossroad]
of cars))/(count cars)
; nastavení proměnné - průměrný čas strávený na křižovatce
  ask cars [
    set time-in-crossroad (time-in-crossroad + 1);nastavení proměnné – čas
;strávený v křižovatce
    ;set label time-in-crossroad ;štítek na auta - část strávený v křižovatce
  ]
  tick
end

```

Zdrojový kód 7 - Procedura „GO“ v modelu s kruhovým objezdem - Varianta III. (vlastní zpracování)

V části „PROPERTIES PATCHES“ jsou procedury, které nastavují vlastnosti patches. V modelech je v této části pouze procedura „set_turning_points“, která nastavuje body, kde auta v křižovatce odbočují.


```

to set_turning_points
;      -- auta od Pardubic --
  ask patches with [pxcor = -1 and pycor = -3][
    set turning-point 1
    set heading-car 270
  ]
  ask patches with [pxcor = 3 and pycor = -1][ ;možná odbočí
    set turning-point 1
    set heading-car 90
  ]
;      -- auta od UHK --
  ask patches with [pxcor = -2 and pycor = 3][
    set turning-point 2
    set heading-car 270
  ]
  ask patches with [pxcor = 0 and pycor = -1][
    set turning-point 2
    set heading-car 90
  ]
]
...

end

```

Zdrojový kód 8 - Část procedury „set_turning_points“ v modelech se světelnou křižovatkou – Varianta I. (vlastní zpracování)

V modelu kruhového objezdu jsou v této části dvě procedury „set_turning_points“ a „set_invisible_points“. První z procedur nastavuje patches, na kterých auta sjíždějí z kruhového objezdu. Druhá řeší podjezd aut pod kruhovým objezdem, kde určuje body, na kterých auta „mizí“ a následně se „objevují“.

```

to set_turning_points
;      -- sjezd na Futurum --
  ask patches with [pxcor = 4 and pycor = -3][
    set turning-point 1
    set heading-car 90
  ]
;      -- sjezd na Pardubice --
  ask patches with [pxcor = -4 and pycor = 3][
    set turning-point 1
    set heading-car 270
  ]
]
...

```

end

```
to set_invisible_points
  ask patches with [abs pxcor = 10 and pycor >= -2 and pycor <= 2][
    ;"podjezd" pod kruhovým objezdem
    set invisible-point 1
  ]
end
```

Zdrojový kód 9 - Část procedury „set_turning_points“ a procedura „set_invisible_points“ – Varianta III. (vlastní zpracování)

Část „PREFERENCE“ se zabývá určením předností a osahují ji varianty I. a II. V této části je řešen pohyb v křižovatce a dání správné přednosti v ní. Tato část obsahuje dvě procedury „set_preference_patches“ a „set_preference_cars“. Procedura „set_preference_patches“ nastavuje vlastnosti políčkům v křižovatce, na kterých auta mění směr nebo kde je potřeba dát přednost.

```
to set_preference_patches
  ask patches with [(pxcor = 2 and pycor = -4) or
    (pxcor = 1 and pycor = -4) or
    (pxcor = 0 and pycor = 4)][
    set give-preference-variable 1
  ]
  ask patches with [(pxcor = -1 and pycor = 2) or
    (pxcor = 3 and pycor = -1)][
    set give-preference 1
  ]
  ...
end
```

Zdrojový kód 10 - Procedura „set_preference_patches“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)

V proceduře „set_preference_cars“ přebírají auta vlastnosti od patches, přes které auta projedou. Tato procedura pomáhá při následném určování přednosti v proceduře „move_car“.

```
to set_preference_patches
  ask cars-on patches with [give-preference-variable = 1 ][
    set preference-give 1
  ]
```

```

ask cars-on patches with [has-preference-variable = 1][
  set preference-has 1
]
end

```

Zdrojový kód 11 - Procedura „set_preference_cars“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)

Část „DRAW BACKGROUND“ obsahují všechny tři varianty. Tato část souží pro vytvoření pozadí, ale jsou zde nastavovány proměnné, které jsou využity v dalších procedurách. Ve výsledných modelech je vzhled pozadí vyřešen primitivem „import-drawing“, které převezme vybraný obrázek a vloží ho na pozadí modelu. Toto pozadí bylo vytvořeno ve tril verzi Snagit editoru. V proceduře „draw_background“ jsou definovány proměnné patches „meaning“, která určuje, na jaké silnici je auto, a „traffic-light“, která určuje místo pro semaforey (ve variantě III. není obsaženo).

```

to draw_background
  ask patches [
;      -- silnice od UHK --
    if (pxcor <= 3 and pxcor >= -2) and pycor >= 4[
      ifelse ((pxcor >= 1 and pxcor <= 2))
        [set pcolor brown]
        [set pcolor black]
    ]
    if ((pxcor >= -2 and pxcor <= 0) and pycor >= 4)
      [set meaning "UHK"]
  ...

;      -- podklad pro semaforey --
    if (pxcor = 4 and (pycor >= 1 and pycor <= 3));Futurum
      [set pcolor yellow
set traffic-light 1]
  ...

End

```

Zdrojový kód 12 - Část procedury „draw_background“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)

Varianta III. zahrnuje oproti předchozím dvěma variantám další procedury, příkladem je procedura na vykreslení kruhového objezdu „traffic_roundabout“, která je důležitá pro chod modelu.

```

to traffic_roundabout ;vykreslení kruhového objezdu
  let cx 0 ;střed kruhu
  let cy 0 ;střed kruhu
  let r 5 ;poloměr kruhu
  let p2r ( 2 * pi * r ) ;obvod kruhu
  let step p2r / 360 ;jeden krok

  crt 1 [ ;tvorba 1 turtle
    setxy cx + r cy ;nastavit x y
    pd ;"vysunout" pero a nakreslit kruh
    set heading 0
    while [ p2r > 0 ][ ;kreslit dokud není hotový kruh
      lt 1 ;natočení
      fd step ;dopředu o jeden krok
      set p2r p2r - step ;proměná - počítá kdy je dokreslen kruh
      set color black
      set pcolor blue
      set meaning "Circle"
    ]
  ]
  die
]
end

```

Zdrojový kód 13 - Procedura „traffic_roundabout“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)

Další důležitou součástí modelů je oddíl „MOVE CARS“. Tato část obsahuje mimo jiné proceduru „move-cars“, ve které je řešen pohyb aut. Ve variantě I. a II. je řešen pohyb mimo křižovatku a pohyb v křižovatce, kde jde především o určení správné přednosti v jízdě. Vzhledem ke zjednodušení modelu bylo provedeno pár ústupků, co se týká střetu aut v křižovatce, především se týká míjejících se aut, která jedou od UHK a odbočují na směr Brno a zároveň auta jedoucí od FNHK a odbočují na Pardubice. Tyto auta se v reálném světě míjejí, ale v modelu přes sebe mohou přejíždět.

```
to move_cars
```

...

```

ask cars with [meaning = "crossroad"] [ ; pohyb v křižovatce
  ifelse (meaning-car = "Pardubice" and turning-point = 1) or
    (meaning-car = "UHK" and turning-point = 2) or

```

```

...
    turn_car
  ][
  ifelse (random 50 < prav_odbocovani) and
    ((meaning-car = "Hospital" and turning-point = 5) or ;auto může
    ;jet rovně nebo odbočit
    (meaning-car = "Futurum" and turning-point = 6))][
    turn_car
  ][
...

  if (meaning-car = "UHK")[
    ifelse (preference-give = 1) and [count cars-here with
      [ meaning-car = "UHK" ] ] of patch-ahead 1 = 0[
...

end

```

Zdrojový kód 14 - Části procedury (pohyb v křižovatce) „move_cars“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)

Ve variantě III. Je v proceduře „move_cars“ řešen pohyb aut v podjezdu, kdy je nastaven jejich zprůhlednění a následné „objevení“ pro názornější ukázkou pohybu aut. Dále je v proceduře řešen pohyb po kruhovém objezdu a sjíždění z něj.

```

to move_cars
...

ask cars with [invisible-point = 0 and invisible = 1][ ;pokud je auto
;v "podjezdu"
  move-to patch-ahead 1
]
...

ask cars with [meaning != "Circle" and not any? cars-on patch-ahead 2][
;nájezd na kruhový objezd
  if invisible-point = 1 and invisible = 0 [
    set invisible 1
    set color lput 80 extract-rgb color
    move-to patch-ahead 1
  ]

```

```

if invisible-point = 1 and invisible = 1 [
  set invisible 0
  set color original-color
  move-to patch-ahead 1
]

```

...

end

Zdrojový kód 15 - ČástI procedury „move_cars“ v modelu s kruhovým objezdem – Varianta III. (vlastní zpracování)

V modelech se světelnou křižovatkou (varianta I. a II.) je dalším důležitým oddílem „TRAFFIC LIGHTS“, ve kterém je řešeno nastavení semaforů. Intervaly semaforů ve variantě I. jsou počítány dle skutečně natočeného videa v odpoledních hodinách. Součástí tohoto oddílu je například procedura „traffic_lights“, která vytvoří výchozí nastavení semaforů. Při fungování semaforů je nastaveno, že pokud je na křižovatce auta z předchozího intervalu, pak se další auta rozjedou, až tyto křižovatku opustí.

```

to traffic_lights
  ask patches with [traffic-light = 1][
    sprout-lightsFuturum 1 [
      set color orange
      set shape "lights"
    ]
  ]
]

```

...

end

Zdrojový kód 16 - Část procedury „traffic_lights“ v modelu se světelnou křižovatkou – Varianta I. (vlastní zpracování)

7 Hypotézy

Je stanoveno pět hypotéz, které budou postupně ověřovány. Hypotézy budou ověřovány na základě výsledků z dotazníku. První tři hypotézy budou ověřovat výběr mezi druhou a třetí variantou křižovatky z různých pohledů. Nejdříve budou řešeny muži a ženy, poté starší a mladší respondenti a v neposlední řadě dle zkušenosti řidičů. Zkušenost řidičů určuje doba vlastnění řidičského průkazu. Poslední dvě hypotézy porovnávají data z dotazníku a z programu NetLogo.

7.1 Stanovení hypotéz

Hypotéza 1

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. muži častěji volí variantu III. než ženy.

Hypotéza 2

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. starší respondenti častěji volí variantu III. než mladší respondenti.

Hypotéza 3

Hypotéza ověřuje předpoklad, že mezi variantami II. a III. zkušenější řidiči častěji volí variantu III. než řidiči nezkušení.

(předpoklad pro zkušenost řidičů je doba, po kterou vlastní řidičský průkaz)

Hypotéza 4

Hypotéza ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 odpovídá seřazení modelů podle dat z NetLogo (počty projíždějících vozidel).

Hypotéza 5

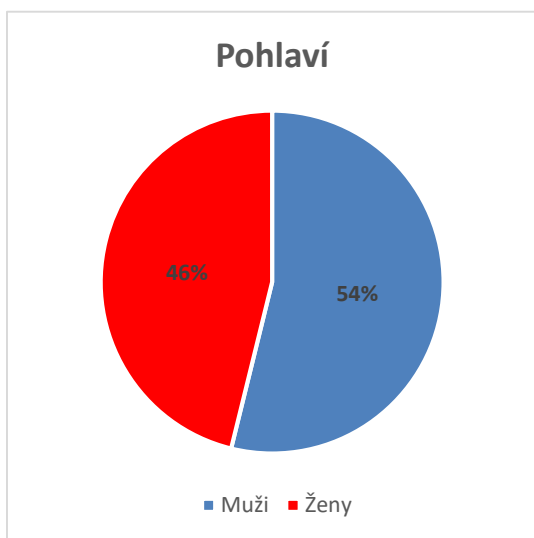
Hypotéza ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 odpovídá seřazení modelů podle dat z NetLogo (časy na projetí křižovatkou).

7.2 Vyhodnocení dotazníkového šetření

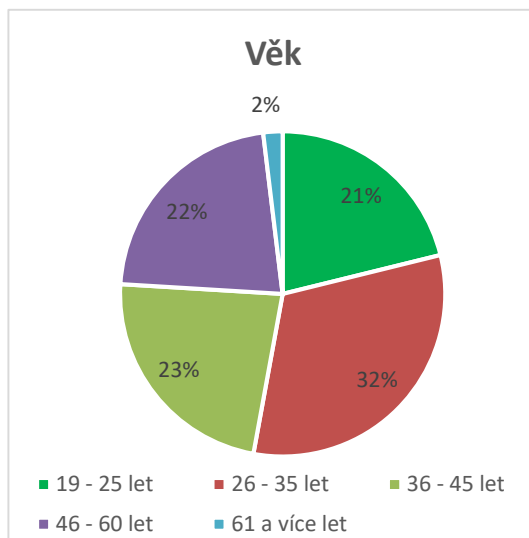
Dotazníkové šetření bylo provedeno online přes webovou stránku vyplnto.cz. K dotazníku bylo vytvořeno video v trial verzi softwaru Camtasia, které bylo následně umístěno na youtube.com a přes odkaz zpřístupněno respondentům. Video sloužilo jako podklad pro některé otázky v dotazníku. Celý zadaný dotazník je uveden v přílohách.

Dotazník se týkal křižovatky Milety v Hradci Králové, která se nachází na Gočárově okruhu u univerzitního kampusu Na Soutoku. Dotazník byl určen respondentům, kteří tímto místem projíždí nebo prochází. Součástí dotazníku bylo krátké video na <https://youtu.be/1JIgX9MAb1U>, podle kterého byly respondenty zodpovídaný otázky 4 - 10. Dotazník obsahoval deset otázek a byl anonymní.

Dotazníkové šetření se účastnilo 104 respondentů (48 žen a 56 mužů), všechny odpovědi byly ve vyhodnocení použity. Věkové rozpětí respondentů bylo široké, nejmladšímu respondentovi bylo 19 let a nejstaršímu 73 let v následujícím grafu jsou respondenti podle věku rozděleni do pěti skupin.

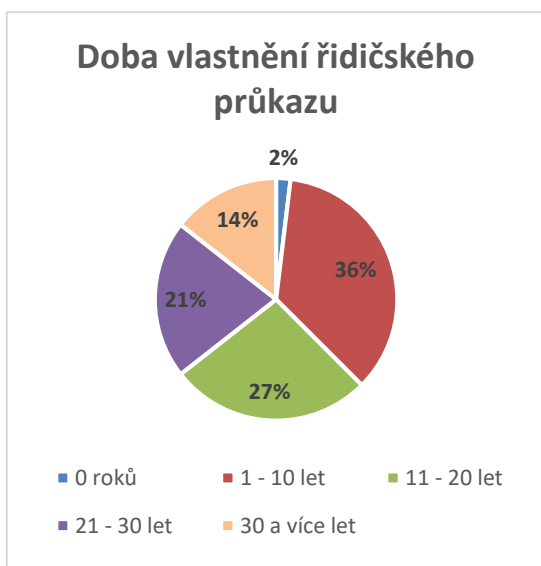


Graf 2 – Pohlaví respondentů (vlastní zpracování)



Graf 1 – Věkové rozložení respondentů (vlastní zpracování)

Dotazovaní vyplňovali, jak dlouho vlastní řidičské oprávnění. 102 respondentů odpovědělo, že jsou vlastníci řidičského průkazu. Dva respondenti řidičské oprávnění nevlastnili. Doba vlastnění řidičského oprávnění byla do Graf 4 rozčleněna do šesti skupin, kde první skupina zahrnuje respondenty bez řidičského oprávnění. Další skupiny jsou 1-10, 11-20, 21-30 a 30 a více let.



Graf 4 – Doba, po kterou respondenti vlastní řidičské oprávnění (vlastní zpracování)



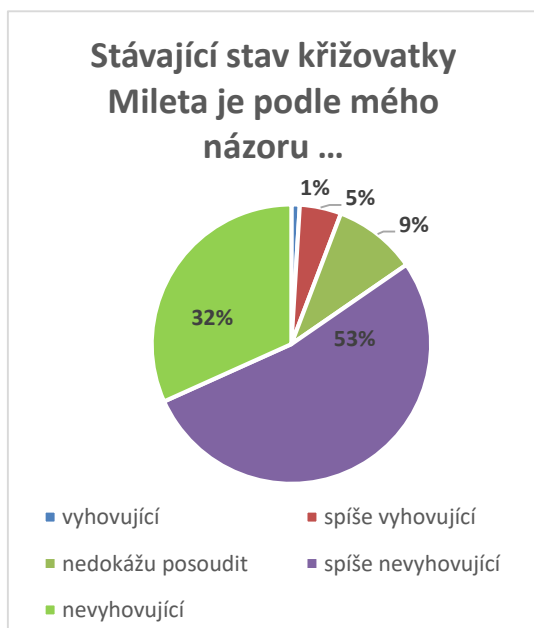
Graf 3 – Vztah ke křižovatce Mileta (vlastní zpracování)

V dotazníku se dále zkoumalo, jaký postoj mají respondenti ve vztahu ke křižovatce Mileta, Graf 3. Většina respondentů zvolila možnost řidič osobního automobilu (96 respondentů). Druhá nejpočetnější odpověď byla chodec, tuto možnost zvolilo 6 respondentů. V odpovědích se dále objevil po jednom řidič nákladního automobilu a spolujezdec.

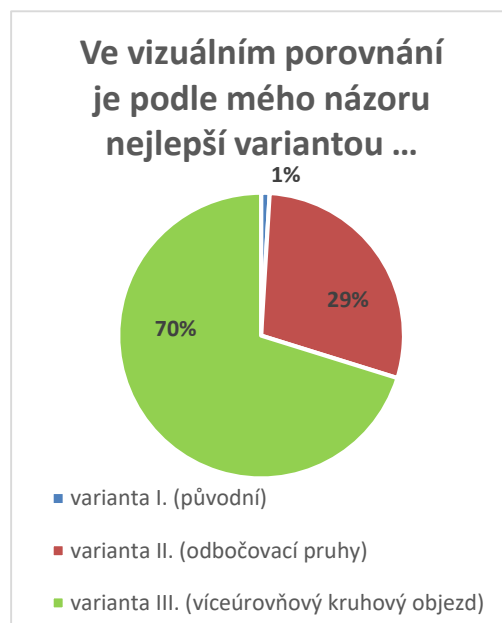
Dále respondenti hodnotili, jak vnímají aktuální stav křižovatk Mileta Graf 5. Pro hodnocení byla nastavena pětistupňová hodnotící škála:

- vyhovující,
- spíše vyhovující,
- nedokážu posoudit,
- spíše nevyhovující,
- nevyhovující.

Možnost, že křižovatka Mileta je vyhovující, zvolil jeden respondent. Spíše vyhovující se zdála pěti respondentům. Deset respondentů nedokáže posoudit, zda je stav vyhovující nebo nevyhovující. Nejvíce respondentů volilo možnost, že aktuální stav křižovatky je spíše nevyhovující, tuto možnost vybralo 55 respondentů. Možnost, že aktuální stav křižovatky Mileta je nevyhovující, zvolilo 33 respondentů.



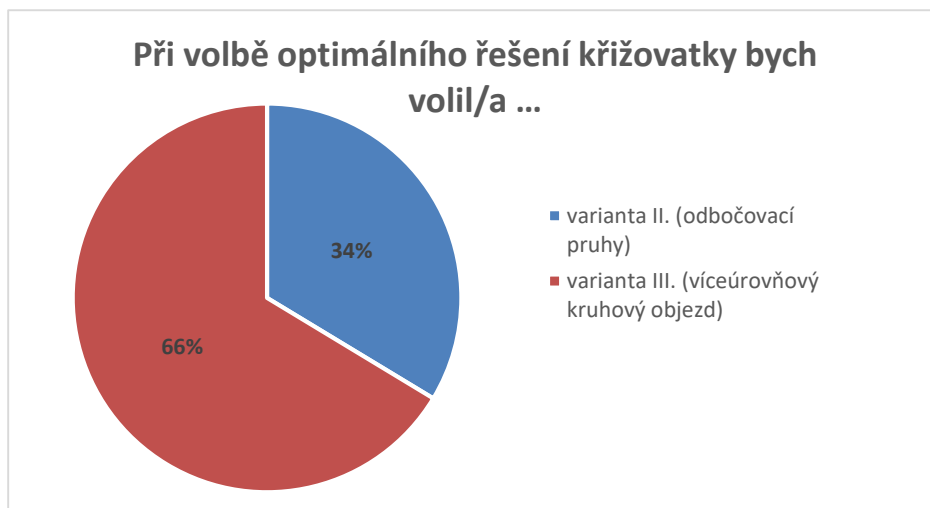
Graf 5 – Stávající stav křižovatky Mileta je dle názorů respondentů



Graf 6 – Vizuální porovnání všech tří variant křižovatek (vlastní zpracování)

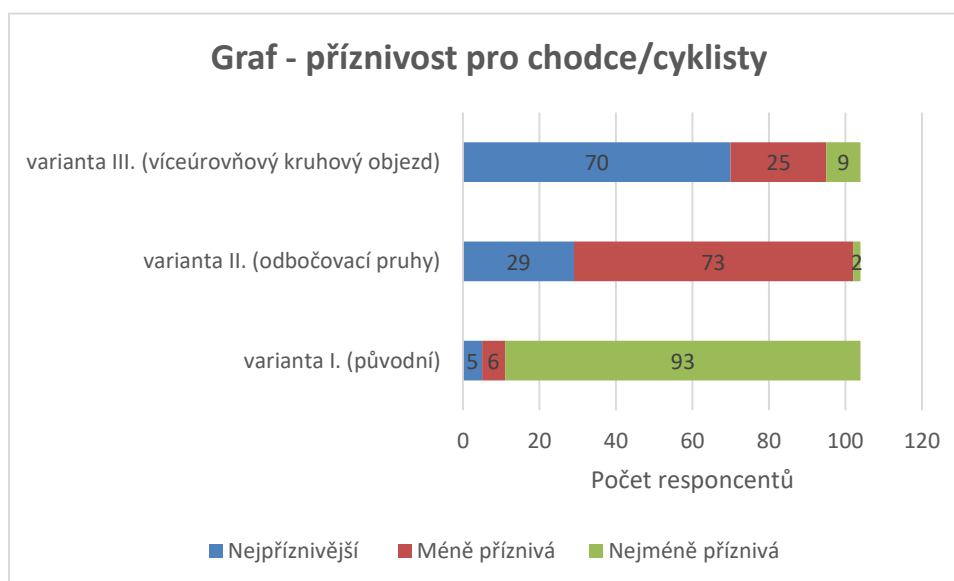
V dotazníku se dále řešilo vizuální porovnání na základě videa zveřejněného na youtube, Graf 6. V porovnávání simulací křižovatek jeden respondent vybral nejvhodnější variantou křižovatky variantu I. (původní). Většina respondentů volila mezi dalšími dvěma odpověďmi variantou II. (odbočovací pruhy) a variantou III. (víceúrovňový kruhový objezd). Variantu II. zvolilo 30 respondentů a variantu III. vybralo 73 respondentů.

Při volbě respondentů mezi možnými variantami realizace většina respondentů na základě videa volila variantu III. celkem 69 respondentů. Variantu II. vybralo 35 respondentů.



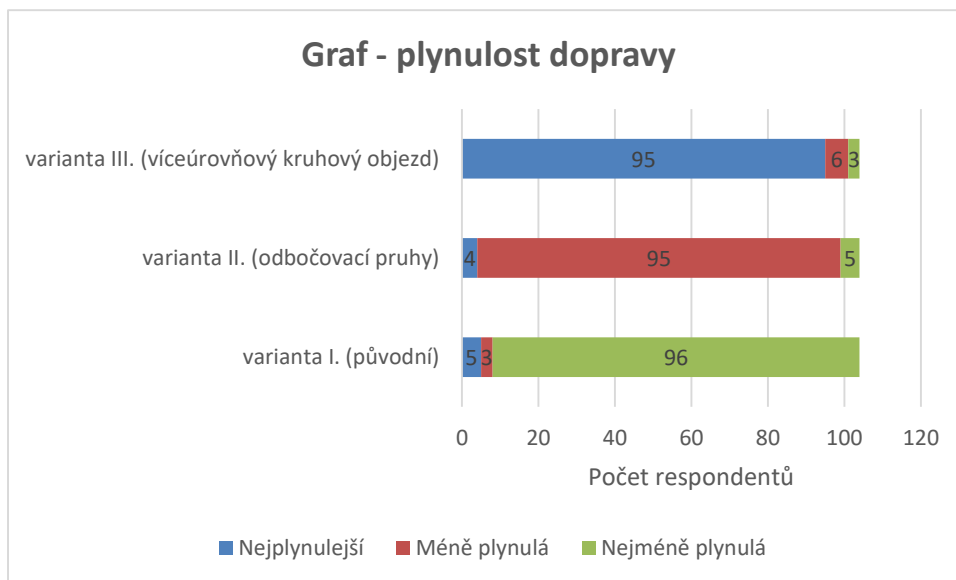
Graf 7 – Volba optimálního řešení respondenty (vlastní zpracování)

Respondenti dále rozhodovali o pořadí křižovatek vzhledem k tomu, jak je provedení vhodné pro chodce nebo cyklisty, což hodnotili na základě informací ve videu. V tomto porovnání nejhůře vyšla varianta I., kdy ji 93 respondentů hodnotilo jako nejméně příznivou pro chodce/cyklisty. Nejlépe hodnocenou byla varianta III., kterou za nejpříznivější pro chodce/cyklisty hodnotilo 70 respondentů.



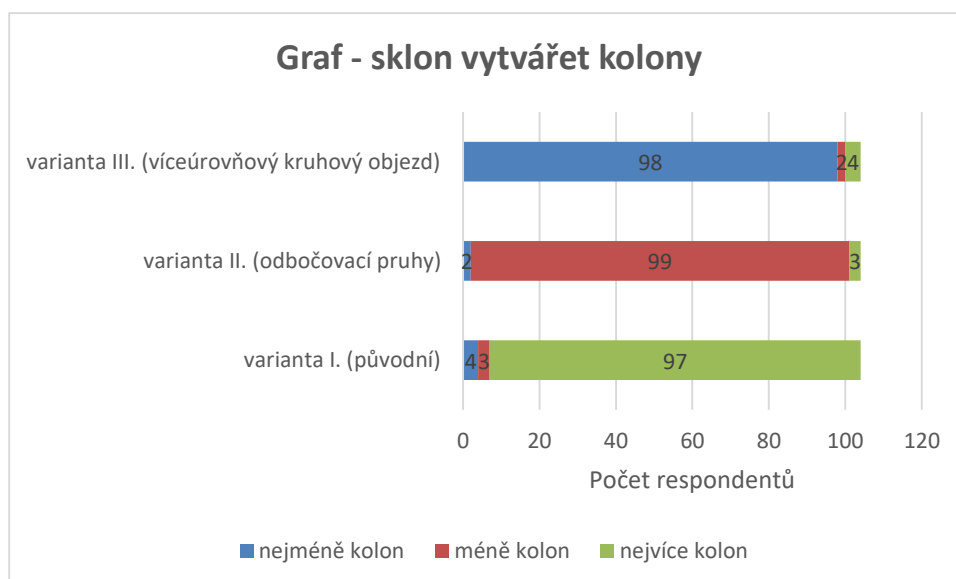
Graf 8 – Příznivost pro chodce/cyklisty (vlastní zpracování)

Dotazovaní dále křižovatky řadili podle plynulosti dopravy, kterou hodnotili podle videa. Za nejplynulejší variantu většinou respondenti hodnotili variantu III. (95 respondentů). Naopak za variantu, která má nejméně plynulou dopravu, respondenti volili variantu I. (96 respondentů).



Graf 9 – Plynulost dopravy (vlastní zpracování)

V neposlední řadě respondenti hodnotili tvorbu kolon ve variantách podle videa a řadili je podle jejich subjektivního názoru. 98 respondentů zvolilo variantu III. jako tu, kde se tvořilo nejméně kolon. Variantu I. majoritní část respondentů zvolila, jako tu kde se tvoří nejvíce kolon (97 respondentů).



Graf 10 – Sklony vytvářet kolony (vlastní zpracování)

7.3 Ověření stanovených hypotéz

Všechny výpočty byly provedeny v tabulkovém softwaru MS Excel. Kritická hodnota byla zjištěna prostřednictvím funkce CHIINV. P - hodnota byla vypočítána za pomoci funkce CHITEST. V tabulkách, které jsou u jednotlivých hypotéz uvedeny, zobrazují dílčí výsledky výpočtů v MS Excel. Při ověřování hypotéz byly sestaveny kontingenční tabulky získaných dat z dotazníkového šetření a dat získaných z NetLogo. Data z NetLogo pro vyhodnocení hypotéz 4 a 5 byly získány experimenty prostřednictvím BehaviorSpace.

Hypotéza 1 ověřuje předpoklad, že mezi variantami II. a III. muži častěji volí variantu III. než ženy.

Hypotéza byla ověřována prostřednictvím testu chí-kvadrát, testu nezávislosti v kontingenčních tabulkách. Test ověřil, zda volba varianty III. je u mužů a žen v populaci rozdílná. Hladina významnosti byla určena na 5 %.

Stanovení hypotéz:

$$H_0: n_{ij} = \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro všechna } i \in \{1, 2\}, j \in \{1, 2\}$$

$$H_A: n_{ij} \neq \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro nějaká } i, j$$

Testové kritérium s $df = (r - 1) * (s - 1)$

$$G = \sum_{i=1}^r \sum_{j=1}^s \frac{(n_{ij} - n'_{ij})^2}{n'_{ij}}$$

H₀: Muži a ženy volí stejně často variantu III.

H_A: Muži a ženy nevolí stejně často variantu III.

Tabulka 1 - Dílčí výpočty při testování hypotézy 1 (vlastní zpracování)

Rozsah vzorku (n)	104
Hladina významnosti (α)	0,05
Stupně volnosti	1
Kritická hodnota z tabulek (fce CHIINV)	3,841
Vypočítaná p-hodnota v Excelu (fce CHITEST)	0,443
Testové kritérium - Chí-kvadrát (χ^2)	0,59
Pearsonův koeficient kontingence (C)	0,075

Hypotézu H₀ nezamítáme

Spočítaná p-hodnota je vyšší než α (0,05) a vypočtená hodnota χ^2 (0,59) je nižší než kritická hodnota 3,841 (pro hladinu významnosti 0,05 a stupních volnosti 1). Můžeme říct, že na základě vypočítaných dat hypotézu H_0 nezamítáme. Nemůžeme tedy prokázat závislost mezi proměnnými, což nám potvrzuje i Pearsonův koeficient, který má hodnotu 0,075, tedy nám vykazuje zanedbatelnou závislost mezi proměnnými.

Hypotéza 2 ověřuje předpoklad, že mezi variantami II. a III. starší respondenti častěji volí variantu III. než mladší respondenti.

Hypotéza byla ověřována prostřednictvím testu chí-kvadrát, testu nezávislosti v kontingenčních tabulkách. Test ověřil, zda volba varianty III. je u starších a mladších respondentů v populaci rozdílná. Hladina významnosti byla určena na 5 %.

Stanovení hypotéz:

$$H_0: n_{ij} = \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro všechna } i \in \{1, 2\}, j \in \{1, 2\}$$

$$H_A: n_{ij} \neq \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro nějaká } i, j$$

Testové kritérium s $df = (r - 1) * (s - 1)$

$$G = \sum_{i=1}^r \sum_{j=1}^s \frac{(n_{ij} - n'_{ij})^2}{n'_{ij}}$$

H_0 : Při volbě varianty III. nezáleží na věku.

H_A : Při volbě varianty III. záleží na věku.

Tabulka 2 - Dílčí výpočty při testování hypotézy 2 (vlastní zpracování)

Rozsah vzorku (n)	104
Hladina významnosti (α)	0,05
Stupně volnosti	1
Kritická hodnota z tabulek (fce CHIINV)	3,841
Vypočítaná p-hodnota v Excelu (fce CHITEST)	0,04
Testové kritérium - Chí-kvadrát (χ^2)	4,253
Pearsonův koeficient kontingence (C)	0,198

Hypotézu H_0 zamítáme

P-hodnota je 0,04, tedy nižší než α (0,05). Vypočítaná hodnota χ^2 je 4,253, je vyšší než kritická hodnota 3,841 (pro hladinu významnosti 0,05 a stupních volnosti 1). Hypotézu H_0 zamítáme. Existuje závislost stáří respondenta při výběru varianty křižovatky. Pearsonův koeficient ukazuje slabou závislost proměnných.

Hypotéza 3 ověřuje předpoklad, že mezi variantami II. a III. zkušenější řidiči častěji volí variantu III. než řidiči nezkušení. Předpoklad pro zkušenost řidičů je doba, po kterou vlastní řidičský průkaz.

Hypotéza byla ověřována prostřednictvím testu chí-kvadrát, testu nezávislosti v kontingenčních tabulkách. Test ověřil, zda volba varianty III. je zkušenějších a méně zkušených řidičů v populaci rozdílná. Hladina významnosti byla určena na 5 %.

Stanovení hypotéz:

$$H_0: n_{ij} = \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro všechna } i \in \{1, 2\}, j \in \{1, 2\}$$

$$H_A: n_{ij} \neq \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro nějaká } i, j$$

Testové kritérium s $df = (r - 1) * (s - 1)$

$$G = \sum_{i=1}^r \sum_{j=1}^s \frac{(n_{ij} - n'_{ij})^2}{n'_{ij}}$$

H_0 : Při volbě varianty III. nezáleží na zkušenostech se řízením.

H_A : Při volbě varianty III. záleží na zkušenostech se řízením.

Tabulka 3 - Dílčí výpočty při testování hypotézy 3 (vlastní zpracování)

Rozsah vzorku (n)	104
Hladina významnosti (α)	0,05
Stupně volnosti	1
Kritická hodnota z tabulek (fce CHIINV)	3,841
Vypočítaná p-hodnota v Excelu (fce CHITEST)	0,369
Testové kritérium - Chí-kvadrát (χ^2)	0,808
Pearsonův koeficient kontingence (C)	0,088

Hypotézu H_0 nezamítáme

Spočítaná p-hodnota funkcí CHITEST je vyšší než hladina významnosti (0,05). Hodnota χ^2 je 0,808 a je tedy nižší než kritická hodnota 3,841 (pro hladinu

významnosti 0,05 a stupních volnosti 1). Hypotézu H0 nezamítáme. Nepovedla se prokázat závislost mezi proměnnými. Pearsonův koeficient je 0,088 a potvrzuje tak zanedbatelnou závislost mezi proměnnými.

Hypotéza 4 ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 odpovídá seřazení modelů podle dat z programu NetLogo (počty projíždějících vozidel).

Hypotéza byla ověřována prostřednictvím testu chí-kvadrát, testu nezávislosti v kontingenčních tabulkách. Test ověřil, zda názory respondentů nejsou v souladu s výsledky z programu NetLogo. Hladina významnosti byla určena na 5 %. Stanovení hypotéz:

$$H_0: n_{ij} = \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro všechna } i \in \{1, 2, 3\}, j \in \{1, 2, 3, 4, 5, 6\}$$

$$H_A: n_{ij} \neq \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro nějaká } i, j$$

Testové kritérium s $df = (r - 1) * (s - 1)$

$$G = \sum_{i=1}^r \sum_{j=1}^s \frac{(n_{ij} - n'_{ij})^2}{n'_{ij}}$$

H0: Mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 neodpovídá seřazení modelů podle dat z programu NetLogo.

HA: Mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 odpovídá seřazení modelů podle dat z programu NetLogo.

Tabulka 4 - Dílčí výpočty při testování hypotézy 4 (vlastní zpracování)

Rozsah vzorku (n)	624
Hladina významnosti (α)	0,05
Stupně volnosti	10
Kritická hodnota z tabulek (fce CHIINV)	18,306
Vypočítaná p-hodnota v Excelu (fce CHITEST)	0,001
Testové kritérium - Chí-kvadrát (χ^2)	1101,875
Pearsonův koeficient kontingence (C)	0,799

Hypotézu H0 zamítáme

P-hodnota je nižší než α (0,05). Vypočítaná hodnota χ^2 je vyšší než kritická hodnota 18,306 (pro hladinu významnosti 0,05 a stupních volnosti 10). Hypotézu

H0 zamítáme. Mezi seřazením podle subjektivních preferencí respondentů pro otázku 9 a seřazením modelů podle dat z programu NetLogo existuje závislost. Pearsonův koeficient u tohoto souboru vyšel 0,799, ukazuje na vysokou závislost, přestože p-hodnota vyšla nízká. Výsledek není relevantní. Je vhodné daný soubor rozdělit na tři podsoubory a v každém vypočítat Pearsonův koeficient kontingence zvlášť. Jak můžeme vidět v následující tabulce závislost proměnných je slabá.

Tabulka 5 – Pearsonův koeficient pro hypotézu 4 (vlastní zpracování)

Respondenti X NetLogo Varianta I.	0,196
Respondenti X NetLogo Varianta II.	0,208
Respondenti X NetLogo Varianta III.	0,208

Hypotéza 5 ověřuje předpoklad, že mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 odpovídá seřazení modelů podle dat z NetLogo (časy na projetí křižovatkou).

Hypotéza byla ověřována prostřednictvím testu chí-kvadrát, testu nezávislosti v kontingenčních tabulkách. Test ověřil, zda názory respondentů nejsou v souladu s výsledky z programu NetLogo. Hladina významnosti byla určena na 5 %. Stanovení hypotéz:

$$H_0: n_{ij} = \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro všechna } i \in \{1, 2, 3\}, j \in \{1, 2, 3, 4, 5, 6\}$$

$$H_A: n_{ij} \neq \frac{n_{i \cdot} \cdot n_{\cdot j}}{n} \text{ pro nějaká } i, j$$

$$\text{Testové kritérium s } df = (r - 1) * (s - 1)$$

$$G = \sum_{i=1}^r \sum_{j=1}^s \frac{(n_{ij} - n'_{ij})^2}{n'_{ij}}$$

H0: Mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 neodpovídá seřazení modelů podle dat z programu NetLogo.

HA: Mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 odpovídá seřazení modelů podle dat z programu NetLogo.

Tabulka 6 – Dílčí výpočty při testování hypotézy 5 (vlastní zpracování)

Rozsah vzorku (n)	624
Hladina významnosti (α)	0,05
Stupně volnosti	10
Kritická hodnota z tabulek (fce CHIINV)	18,306
Vypočítaná p-hodnota v Excelu (fce CHITEST)	0,001
Testové kritérium - Chí-kvadrát (χ^2)	1144,737
Pearsonův koeficient kontingence (C)	0,804

Hypotézu H0 zamítáme

Spočítaná hodnota χ^2 je vyšší než kritická hodnota 18,306 (pro hladinu významnosti 0,05 a stupních volnosti 10). P-hodnota je nižší než α (0,05). Hypotézu H0 zamítáme. Mezi seřazením podle subjektivních preferencí respondentů pro otázku 10 a seřazením modelů podle dat z programu NetLogo existuje závislost. Pearsonův koeficient u tohoto souboru indikuje vysokou závislost, ačkoli p-hodnota vyšla 0,001, tedy opravdu nízká. Tento výsledek není relevantní, proto bylo i v tomto případě vhodné daný soubor rozdělit na tři podsoubory a v každém vypočítat Pearsonův koeficient kontingence zvlášť. Jak můžeme vidět v následující tabulce závislost proměnných je slabá.

Tabulka 7 – Pearsonův koeficient pro hypotézu 5 (vlastní zpracování)

Respondenti X NetLogo Varianta I.	0,183
Respondenti X NetLogo Varianta II.	0,155
Respondenti X NetLogo Varianta III.	0,17

8 Shrnutí výsledků

V první polovině praktické části byly řešeny varianty křižovatky Mileta v Hradci Králové, které byly následně vymodelovány v programu NetLogo. Při zpracování modelů křižovatky byl využit multiagentní program NetLogo. Zvolený program využívá jazyk Logo, který je uživatelsky přívětivý. Svět v programu NetLogo je tvořen agenty a prostředím, kteří mezi sebou komunikují a interagují. Jsou zde využity proměnné, které mohou být proměnné agentů nebo globální proměnné. Proměnné agentů, určují vlastnost agentovi a každý agent může mít danou vlastnost s jiným parametrem. Oproti tomu globální proměnné nabývají v jeden okamžik jedné hodnotě a může k nim přistupovat jakýkoli agent. Neméně důležitou součástí jsou procedury, které představují příkazy a reportéry a určují, co mají agenti za daných podmínek vykonávat. V programu NetLogo jsou k dispozici primitiva, která představují implementované procedury, které jsou předem definované v programu Logo, tyto procedury usnadňují práci v programu NetLogo.

Byly vymodelovány tři varianty křižovatky Milety v Hradci Králové. Pokud se podíváme na rozsah modelů, tak kód v programu NetLogo ve variantě I. obsahuje 502 řádků, variantu II. tvoří 510 řádků a ve variantě III. je 333 řádků. Varianta I. představuje stávající stav křižovatky Mileta. Varianta II. a varianta III. představují navrhované možnosti nového uspořádání křižovatky (dle variant, které jsou veřejně diskutovány). Varianta II. má oproti variantě I. přidané odbočovací pruhy, které nebudou ovlivněny světelnou signalizací na křižovatce, a je v ní počítáno s podchody pro chodce. Varianta III. uvažuje hlavní tak Praha/Brno vyřešit podjezdem, nad kterým je počítáno s kruhovým objezdem. Stejně jako ve variantě II. je i v této počítáno pro podchody, které budou řešeny pod kruhovým objezdem, nad hlavním tahem Praha/Brno. Jak ukazují modely, obě varianty by zlepšili situaci na křižovatce a díky tomu by měli sanitky hladší průjezd křižovatkou, než je tomu nyní v dopravní špičce. Kruhový objezd by vyřešil kolony, které se tvoří od centra a od FNHK. Ve výběru varianty hraje určitě vysokou roli i cena výsledného provedení, která je u varianty II. třetinová oproti variantě III.

V druhé polovině praktické části bylo řešeno dotazníkové šetření a hypotézy. Na začátku bylo stanoveno pět hypotéz, které byly následně ověřovány na základě dat

z dotazníkového šetření a výstupů z programu NetLogo. Následně byl vytvořen dotazník, součástí kterého bylo video tvořené z fotek křižovatky a vizualizací možných provedení a zároveň ukázky modelů vytvořených v programu NetLogo. Toto video bylo zveřejněno na youtube.com.

Dotazníkového šetření se zúčastnilo 104 respondentů a rozložením 46 % ženy a 54 % muži. Respondenti měli širokou věkovou škálu. Na otázku, zda vlastní řidičské oprávnění pouze dva odpověděli, že řidičské oprávnění nevlastní, zbylých 102 respondentů v době vyplňování dotazníku řidičské oprávnění vlastnilo. Většina respondentů (92 %) se považuje vzhledem ke křižovatce Mileta převážně za řidiče osobního automobilu. Stávající stav křižovatky vnímá pouze 1 % respondentů za vyhovující, 5 % za spíše vyhovující, 9 % nedokázalo aktuální stav posoudit, 53 % si myslí, že stávající stav je spíše nevyhovující a 32 % považuje aktuální provedení křižovatky za nevyhovující. Při volbě optimálního řešení volilo 66 % variantu III. a 34 % variantu II. Když se podíváme na otázky, ve kterých respondenti měli radit křižovatky, ve všech třech otázkách většina respondentů považovala za nejlepší volbu variantu III. Tyto otázky se týkaly příznivosti pro chodce/cyklisty, plynulosti dopravy a tvorby kolon na křižovatce. Za vyhovující vnímal křižovatku pouze jeden respondent. Byla to 28 letá žena, která je řidička a podle všech svých odpovědí je se stávajícím stavem spokojená.

Hypotézy byly dokazovány na základě testu chí-kvadrát o nezávislosti znaků v kontingenční tabulce. Obecně u tohoto testu platí, že H_0 (nulová hypotéza) tvrdí, že v základním souboru neexistuje souvislost. Pro potřeby hypotéz, lze formulaci upravit. Například v první hypotéze můžeme říct, že muži i ženy volí variantu III. stejně často. U H_A (alternativní hypotézy) je naopak tvrzeno, že zkoumané veličiny spolu v základním souboru souvisí, což v případě naší hypotézy můžeme formulovat, že výběr varianty III. je rozdílný u mužů a žen. U ostatních hypotéz je formulace obdobná.

První tři hypotézy se týkaly upřednostnění varianty III. před variantou II., kdy bylo nejprve zkoumáno, zda rozhodnutí ovlivňuje pohlaví respondentů (hypotéza 1). Následná hypotéza se zaměřila na věk respondentů (hypotéza 2) a v neposlední řadě bylo ověřováno, zda čas, po který respondent vlastní řidičské oprávnění

(hypotéza 3) ovlivní jeho rozhodnutí. Bylo spočítáno, že u hypotézy 1 a hypotézy 3 nelze H_0 zamítnout. U hypotézy 2 H_0 zamítáme.

Hypotéza 4 a hypotéza 5 se zaměřily na porovnání dat z odpovědí respondentů a dat získaných z modelů v programu NetLogo. V obou případech byla H_0 zamítnuta a bylo prokázáno, že názory respondentů na otázky 9 a 10 z dotazníku jsou ve shodě s výsledky z modelů v programu NetLogo.

9 Závěry a doporučení

Tato diplomová práce se zabývala simulací a stěžejní částí bylo vytvořit modely v NetLogo a dále s nimi pracovat.

Cílem diplomové práce byla analýza simulačních přístupů, nástrojů a jejich vizualizačních možností. Dále navrhnout model a ověřit na konkrétní simulaci.

Čtyři kapitoly diplomové práce se věnují teorii zkoumané oblasti. V této části je čtenář seznámen se základními pojmy v oblasti modelování, kde se dozví principy modelování nebo kritéria hodnocení při posuzování modelů. Dále jsou v této části čtenáři představeny přístupy k modelování a v neposlední řadě je zde popsáno několik typů modelů. Další kapitola se věnuje simulaci, kde je možné se seznámit s charakteristikou simulačních modelů i jejich využitím a nástroji, které je možné využít při simulacích.

V této práci jsou zmíněné tři nástroje NetLogo, AnyLogic a PTV Vissim. Pro tvorbu modelu v praktické části byl vybrán program NetLogo pro jeho uživatelskou přívětivost, univerzálnost, dostupnost a aktuálnost.

V neposlední řadě je popsána vizuální analytika, její historie, ale také aplikace vizuální analýzy, kde se čtenář může dozvědět například obory, ve kterých je vizuální analýza nejčastěji využívána nebo je v této podkapitole nastíněn proces analýzy. Poslední části kapitoly o vizuální analýze je několik případových studií, které se týkaly dopravních problémů.

Dvě kapitoly věnované praktické části se zabývají modelováním v NetLogo a hypotézám na základě dotazníkového šetření. V části věnované modelování v NetLogo je na úvod čtenář seznámen s variantami křižovatek. Varianta I. představuje stávající stav křižovatky. Následující dvě varianty jsou v minulosti nejvíce diskutovanými variantami na optimalizaci křižovatky. Všechny tři varianty jsou popsány a je u nich obrázek (varianta I.) nebo vizualizace (varianta II. a varianta III.). Následuje stručný popis nástroje NetLogo, ve kterém jsou vytvořeny modely křižovatek. Na popis programu NetLogo navazují podkapitoly týkající se oblastí, které jsou podstatné při psaní kódu v jazyce Logo (programovací jazyk v nástroji NetLogo). Jsou to agenti a prostředí, proměnné a procedury. U každé oblasti jsou

uvedeny příklady z modelů, které byly vytvořeny. Celé kódy jsou k nahlédnutí v přílohách.

Na závěr je řešeno pět hypotéz, které byly ověřovány na základě odpovědí respondentů a dat získaných z programu NetLogo. Hypotézy 4 a 5 potvrzují, že věnovat pozornost vizuální analytice má smysl. Na základě videí, ve kterých byly simulace z NetLogo, respondenti řadili varianty křižovatky podle daných kritérií shodně, jako byly křižovatky seřazeny dle dat z NetLogo.

Vytvořené modely v programu NetLogo jsou vhodné pro další podrobnější zkoumání. Lze na nich aplikovat například tzv. chytré semaforey, které intervaly budou přizpůsobovat aktuální situaci na křižovatce a zajistí tak co nejplynulejší průjezd ze všech směrů. Bylo by možné upravit v modelech rychlost aut (jejich zpomalování a zrychlování). Další možností, by bylo zaměřit se při zkoumání modelů na chodce a cyklisty. Přidat je do modelů a sledovat, která varianta je pro ně nejvhodnější. Stejně tak by bylo možné dotazník zaměřit na chodce a cyklisty. Zajímavé by bylo do modelů přidat další agenty, kteří by ovlivnili dopravu, například

- kamiony a sledovat jak se změní chování modelů (především hustota dopravy),
- sanitky a dále optimalizovat modely (například přidáním speciálního pruhu pouze pro sanitky, kde by to bylo vhodné.

10 Seznam použité literatury

ANYLOGIC, 2019. *AnyLogic: Simulation Modeling Software Tools & Solutions for Business* [online] [vid. 2019-01-30]. Dostupné z: <https://www.anylogic.com/>

BHADRA, S., A. KUNDU a S. K. GUHA, 2014. An Agent based Efficient Traffic Framework using Fuzzy. *Fourth International Conference on Advanced Computing & Communication Technologies*.

BOX, George E. P., 2013. *An accidental statistician: the life and memories of George E.P. Box*. Hoboken, New Jersey: John Wiley and Sons, Inc. ISBN 978-1-118-51494-8.

BUCHALCEVOVÁ, Alena, 2005. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada. ISBN 978-80-247-1075-4.

BUREŠ, Vladimír, 2007. *Systémové myšlení a teorie systémů*. 2., rozš. vyd. Hradec Králové: Gaudeamus. ISBN 978-80-7041-537-5.

DVOŘÁK, Drahošlav a Martin MAREČEK, 2017. *Project Portfolio Management*. 1. vydání. Brno: Albatros Media a. s. ISBN 978-80-251-4893-8.

HABR, Jaroslav, 1976. *Prognostické modelování v hospodářské praxi*. Praha: SNTL - Státní nakladatelství technické literatury.

HARTMANN, Stephan, 2015. *The World as a Process: Simulations in the Natural and Social Sciences*. 22.

HERZINGER, Sascha, Wei GU, Venkata SATAGOPAM, Serge EIFES, Kavita REGE, Adriano BARBOSA-SILVA, Reinhard SCHNEIDER a ETRIKS CONSORTIUM, 2017. SmartR: an open-source platform for interactive visual analytics for translational research data. *Bioinformatics*. **33**(14), 2229–2231. ISSN 1367-4803, 1460-2059.

HUBÁLOVSKÝ, Štěpán, 2011. *Teorie systémů, modelování a simulace*. Hradec Králové: GAUDEAMUS. ISBN 978-80-7435-158-7.

JAXA-ROZEN, Marc, Jan H. KWAKKEL a Martin BLOEMENDAL, 2019. A coupled simulation architecture for agent-based/geohydrological modelling with NetLogo and MODFLOW. *Environmental Modelling & Software*. **115**, 19–37. ISSN 13648152.

KALAMARAS, Ilias, Alexandros ZAMICHOS, Athanasios SALAMANIS, Anastasios DROSOU, Dionysios D. KEHAGIAS, Georgios MARGARITIS, Stavros PAPADOPOULOS a Dimitrios TZOVARAS, 2018. An Interactive Visual Analytics Platform for Smart Intelligent Transportation Systems Management. *IEEE Transactions on Intelligent Transportation Systems*. **19**(2), 487–496.

KEIM, Daniel, ed., 2010. *Mastering the information age: solving problems with visual analytics*. Goslar: Eurographics Association. ISBN 978-3-905673-77-7.

KIELMAN, Joe, Jim THOMAS a Richard MAY, 2009. Foundations and Frontiers in Visual Analytics. *Information Visualization*. **8**(4), 239–246. ISSN 1473-8716.

KOŘÍNEK, Bc. Miloš, 2013. *Využití simulačních metod pro optimalizaci podnikových procesů*. Brno. Diplomová práce. Masarykova univerzita.

LÁSKA, Bc. Jakub, 2017. *Sociální simulace: nástroje a platformy*. Hradec Králové. Diplomová práce. Univerzita Hradec Králové.

MARINI, Marcello, Anna P GAWLIKOWSKA, Andrea ROSSI, Ndaona CHOKANI, Hubert KLUMPNER a Reza S ABHARI, 2018. The impact of future cities on commuting patterns: An agent-based approach. *Environment and Planning B: Urban Analytics and City Science*. 239980831775114. ISSN 2399-8083, 2399-8091.

MLÁDKOVÁ, Ludmila, 2005. *Moderní přístupy k managementu: tacitní znalost a jak ji řídit*. Praha: C.H. Beck. ISBN 978-80-7179-310-6.

MUELLER, Christian, Ulrike KLEIN a Angela HOF, 2018. An easy-to-use spatial simulation for urban planning in smaller municipalities. *Computers, Environment and Urban Systems*. **71**, 109–119. ISSN 01989715.

OMAR, Hasan, 2015. Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology. *International Journal of Advanced Computer Science and Applications* [online]. **6**(2) [vid. 2019-04-04]. ISSN 21565570, 2158107X. Dostupné z: <http://thesai.org/Publications/ViewPaper?Volume=6&Issue=2&Code=ijacsa&SerialNo=6>

PANG, M. a X. ZHAO, 2008. Traffic Flow Prediction of Chaos Time Series by Using Subtractive Clustering for Fuzzy Neural Network Modelling. *Proceedings 2nd International Symposium Information Technology Application*.

PELÁNEK, Radek, 2011. *Modelování a simulace komplexních systémů: jak lépe porozumět světu*. Brno: Masarykova univerzita. ISBN 978-80-210-5318-2.

POPELKA, Ondřej, Jiří ŠŤASTNÝ, Oldřich TRENZ a Jan ŽIŽKA, 2010. Multiagentní systémy (MAS). *Mendelova Univerzita v Brně* [online] [vid. 2019-04-04]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=32363

PROKŮPEK, Marek, 2006. *Simulační metody jako nástroj rozhodování - modelování pomocí programu witness*. Brno. Diplomová práce. Masarykova univerzita.

PTV VISSIM, 2019. *PTV VISSIM MOŽNOSTI A VYUŽITÍ*. Milan Koukol, FD Ústav dopravních systémů - PDF [online] [vid. 2019-04-04]. Dostupné z: <https://docplayer.cz/40724419-Ptv-vissim-moznosti-a-vyuziti-milan-koukol-fd-ustav-dopravnich-systemu.html>

ROQUES, Pascal, 2004. *UML in practice: the art of modeling software systems demonstrated through worked examples and solutions*. Hoboken, NJ: Wiley. ISBN 978-0-470-84831-9.

ROYCE, Dr Winston W, 1970. MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS. *IEEE WESCON* [online]. (26). Dostupné z: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>

SOMMERVILLE, Ian, 2013. *Softwarové inženýrství*. Brno: Computer Press. ISBN 978-80-251-3826-7.

THOMAS, J., 2019. Illuminating the Path: An R&D Agenda for Visual Analytics. *ResearchGate* [online] [vid. 2019-04-04]. Dostupné z: https://www.researchgate.net/publication/236993632_Illuminating_the_Path_An_RD_Agenda_for_Visual_Analytics

TRIASTANTO, Andreas Novian Dwi a Nugraha P. UTAMA, 2019. Model Study of Traffic Congestion Impacted by Incidents. In: *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA): 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*. Yogyakarta, Indonesia: IEEE, s. 1–6. ISBN 978-1-72813-452-9.

VLČEK, Dalibor a Jiří CHUCHRO, 1999. *Modely a modelování: (podpora strategických rozhodovacích procesů)*. Ostrava: Vysoká škola báňská - Technická univerzita, Ekonomická fakulta. ISBN 978-80-7078-621-5.

VOLEJNÍKOVÁ, Veronika, 2013. Mileta už nebude dopravní zátkou, opravdu to bude do tří let? | Hradec Králové.cz. *Hradec Králové, město na dlani* [online] [vid. 2020-11-09]. Dostupné z: <https://zpravy.hradeckralove.cz/mileta-uz-nejbude-dopravni-zatkou-opravdu-to-bude-do-tri-let-14571/>

WILENSKY, Uri, 2018. NetLogo. *NetLogo* [online] [vid. 2018-12-04]. Dostupné z: <https://ccl.northwestern.edu/netlogo/>

ZÁLESKÝ, Petr, 2020. Hradecká křižovatka Mileta by se mohla začít stavět koncem roku 2022. *iDNES.cz* [online] [vid. 2020-11-09]. Dostupné z: https://www.idnes.cz/hradec-kralove/zpravy/stavba-priprava-krizovatka-mileta-hradec-kralovehradecky.A200721_560355_hradec-zpravy_tuu

11 Přílohy

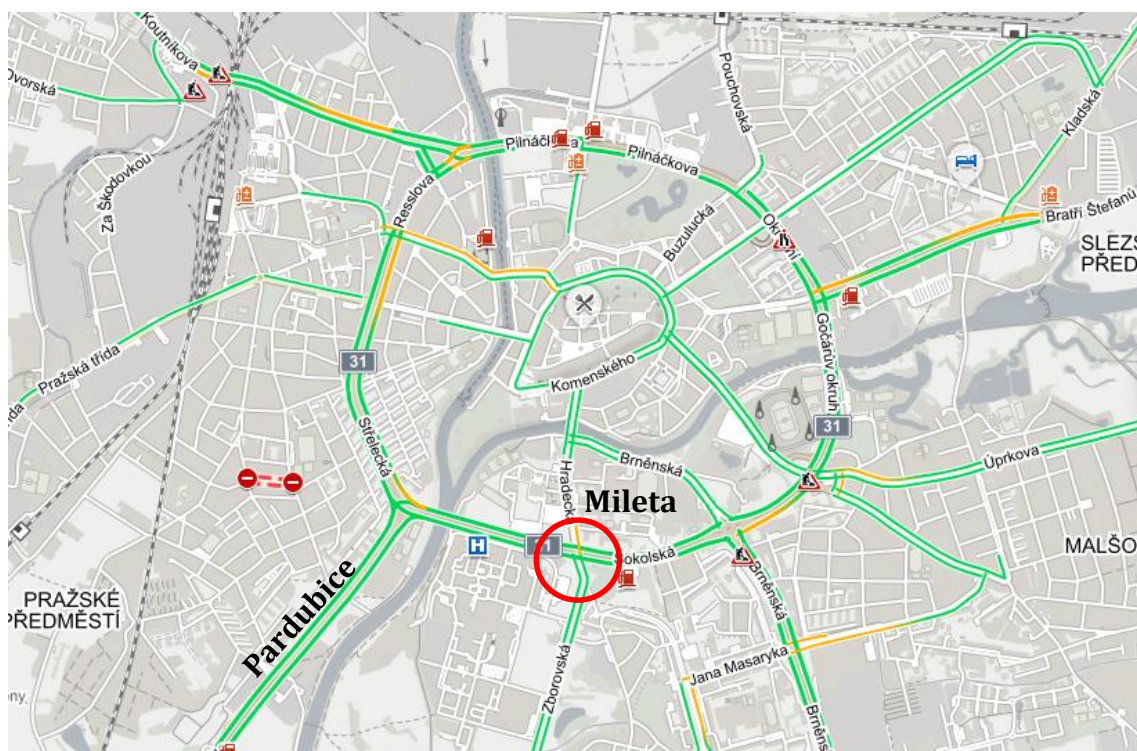
11.1 Dotazník

DOTAZNÍK K DIPLOMOVÉ PRÁCI – POROVNÁNÍ VARIANT PROVEDENÍ KŘIŽOVATKY MILETA

Dobrý den,

jmenuji se Martina Nosilová a studuji navazující magisterský obor Informační management na Fakultě informatiky a managementu Univerzity Hradec Králové. Ráda bych Vás tímto požádala o vyplnění krátkého dotazníku, na téma porovnání variant možného provedení křižovatky Mileta, který bude využit v praktické části mé diplomové práce na téma: **Využití simulačních metod pro vizuální analytiku**. Dotazník obsahuje 10 otázek a zabere Vám jen několik minut.

Dotazník se týká křižovatky v Hradci Králové tzv. Milety, která se nachází na Gočárově okruhu u univerzitního kampusu Na Soutoku. Dotazník je určen respondentům, kteří tímto místem příležitostně, opětovně projíždí nebo prochází. Součástí dotazníku je krátké video na <https://youtu.be/1JlgX9MAb1U>, na základě kterého Vámi budou zodpovídaný některé otázky v dotazníku. Dotazník je anonymní.



- 1) Jsem ...
 - a) žena
 - b) muž
 - c) ne-binární
- 2) Je mi let
- 3) Řidičské oprávnění vlastním roků/let

- 4) Ve vztahu k dané křižovatce se převážně považuji za ...
- a) řidič osobního automobilu
 - b) řidič nákladního automobilu
 - c) chodec
 - d) cyklista
 - e) jiné, uveďte prosím
- 5) Stávající stav křižovatky Mileta je podle mého názoru ... (první část videa na youtube)
- a) vyhovující
 - b) spíše vyhovující
 - c) nedokážu posoudit
 - d) spíše nevhovující
 - e) nevhovující
- 6) Seřadte podle příznivosti pro chodce/cyklisty ... (první část videa na youtube)
- a) varianta I. (původní)
 - b) varianta II. (odbočovací pruhy)
 - c) varianta III. (víceúrovňový kruhový objezd)
- 7) Ve vizuálním porovnání je podle mého názoru nejlepší variantou ... (druhá část videa na youtube)
- a) varianta I. (původní)
 - b) varianta II. (odbočovací pruhy)
 - c) varianta III. (víceúrovňový kruhový objezd)
- 8) Při volbě optimálního řešení křižovatky bych volil/a ... (druhá část videa na youtube)
- a) varianta II. (odbočovací pruhy)
 - b) varianta III. (víceúrovňový kruhový objezd)
- 9) Seřadte podle plynulosti dopravy... (druhá část videa na youtube)
- a) variantě I. (původní)
 - b) variantě II. (odbočovací pruhy)
 - c) variantě III. (víceúrovňový kruhový objezd)
- 10) Seřadte podle sklonu vytvářet kolony ... (druhá část videa na youtube)
- a) varianta I. (původní)
 - b) varianta II. (odbočovací pruhy)
 - c) varianta III. (víceúrovňový kruhový objezd)

Odkaz na video na youtube.com: <https://youtu.be/1JlgX9MAb1U>

11.2 Kód NetLogo – Varianta I.

```
breed[cars car]
breed[lightsFuturum lightFuturum]
breed[lightsUHK lightUHK]
breed[lightsHospital lightHospital]
breed[lightsPardubiceA lightPardubiceA]
breed[lightsPardubiceB lightPardubiceB]

globals [
  lights-change-tick      ;pomocná proměnná pro semaforey
  average_time_spent_crossroad  ;počítá průměrný čas strávený na křižovatce
]

cars-own [
  meaning-car            ;označení aut z určitého směru
  time-in-crossroad      ;čas strávený na křižovatce - počítá se podle ticks
  preference-give        ;auta, která dávají přednost
  preference-has        ;auta, která dostávají přednost
  variable               ;pomocná proměnná při odbočování (otáčení auta do správného
směru)
]

patches-own [
  meaning                ;označení určité části silnice v modelu
  traffic-light          ;patče se semaforey
  turning-point          ;body, na kterých se sjíždí z kruhového objezdu
  heading-car            ;natočení auta v různých turning-point
  give-preference-variable  ;dává políčku vlastnost, kterou následně předá autům - auta,
která dávají přednost
  give-preference        ;na tomto políčku je nutné dát přednost autu, které ji má
  has-preference         ;kontrola těchto políček, zda vůbec může auto zahájit odbočování
- sjezd z křižovatky
  has-preference-variable  ;dává políčku vlastnost, kterou následně předá autům - auta,
která dostávají přednost
]
```

```

to setup
  ca
  reset-ticks

  draw_background          ;vykreslení pozadí
  draw_cars                ;první auta
  set_turning_points       ;nastavení turning-point
  traffic_lights           ;nastavení semaforů
  set_preference_patches   ;preference - nastvení pro řízení v křižovatce

  set lights-change-tick 0 ;nastvení pomocné proměnné pro řízení semaforů
  set average_time_spent_crossroad 0 ;původní nastavení proměnné na 0

  import-drawing "Varianta_1_původní křižovatka.png" ;vložen obrázek jako pozadí modelu
end

to go
  move_cars                ;pohyb aut v modelu
  creat_cars               ;tvorba aut v modelu během chodu
  traffic_light_set_orange ;procedura pro řízení semaforů
  set_lights_interval      ;procedura pro řízení semaforů
  set_preference_cars      ;preference - nastvení pro řízení v křižovatce

  ask cars [               ;čas strávený v křižovatce
    set time-in-crossroad (time-in-crossroad + 1)
    ;set label time-in-crossroad
  ]

  set average_time_spent_crossroad ((sum [time-in-crossroad] of cars)/(count cars))
;nastavení proměnné - průměrný čas strávený na křižovatce

  tick
end

;;----- PROPERTIES PATCHES -----;;
to set_turning_points

```

```

;          -- auta od Pardubic --
ask patches with [pxcor = -1 and pycor = -3][
  set turning-point 1
  set heading-car 270
]
ask patches with [pxcor = 3 and pycor = -1][ ; možná odbočí
  set turning-point 1
  set heading-car 90
]
;          -- auta od UHK --
ask patches with [pxcor = -2 and pycor = 3][
  set turning-point 2
  set heading-car 270
]
ask patches with [pxcor = 0 and pycor = -1][
  set turning-point 2
  set heading-car 90
]
;          -- auta od Futura --
ask patches with [pxcor = -1 and pycor = 1][
  set turning-point 3
  set heading-car 90
]
ask patches with [pxcor = 3 and pycor = 3][ ; možná odbočí
  set turning-point 6
  set heading-car 270
]
;          -- auta od nemocnice --
ask patches with [(pxcor = 1 and pycor = 2) or (pxcor = 2 and pycor = 3)][
  set turning-point 4
  set heading-car 90
]
ask patches with [pxcor = 3 and pycor = -2][ ; možná odbočí
  set turning-point 5
  set heading-car 270
]
end

```

```

;; ----- PREFERENCE -----;;
to set_preference_patches
  ask patches with [(pxcor = 2 and pycor = -4) or
    (pxcor = 1 and pycor = -4) or
    (pxcor = 0 and pycor = 4)][
    set give-preference-variable 1
  ]

  ask patches with [(pxcor = -1 and pycor = 2) or
    (pxcor = 3 and pycor = -1)][
    set give-preference 1
  ]

  ask patches with [(pxcor = -1 and pycor = 3)][ ;kontroluji i další políčka
    set give-preference 2
  ]

  ask patches with [(pxcor = -1 and pycor = 4) or ;kontroluji i další políčka
    (pxcor = -2 and pycor = 5)][
    set has-preference 1
  ]

  ask patches with [(pxcor = 3 and pycor = -4) or
    (pxcor = -1 and pycor = 4) or
    (pxcor = -2 and pycor = 5)][
    set has-preference-variable 1
  ]
end

to set_preference_cars
  ask cars-on patches with [give-preference-variable = 1 ][
    set preference-give 1
  ]

  ask cars-on patches with [has-preference-variable = 1 ][
    set preference-has 1
  ]

```



```

]
end

;;----- DRAW BACKGROUND -----;;
to draw_background
  ask patches [
;      -- silnice od UHK --
    if (pxcor <= 3 and pxcor >= -2) and pycor >= 4[
      ifelse ((pxcor >= 1 and pxcor <= 2) )
        [set pcolor brown]
        [set pcolor black]
    ]
    if ((pxcor >= -2 and pxcor <= 0) and pycor >= 4)
      [set meaning "UHK"]

;      -- silnice od nemocnice --
    if (pxcor <= 3 and pxcor >= -1) and pycor <= -4[
      ifelse (pxcor <= 0 and pxcor >= 0)
        [set pcolor brown]
        [set pcolor black]
    ]
    if ((pxcor >= 1 and pxcor <= 3) and pycor <= -4)
      [set meaning "Hospital"]

;      -- silnice od Pardubic --
    if pxcor <= -2 and (pycor >= -3 and pycor <= 3) [
      ifelse (pycor >= 0 and pycor <= 1)
        [set pcolor brown]
        [set pcolor black]
    ]
    if ((pycor >= -3 and pycor <= -1) and pxcor <= -2)
      [set meaning "Pardubice"]

;      -- silnice od Futura --
    if pxcor >= 4 and (pycor >= -2 and pycor <= 3)[
      ifelse (pycor <= 0 and pycor >= 0)
        [set pcolor brown]

```

```

    [set pcolor black]
  ]
  if ((pycor >= 1 and pycor <= 3) and pxcor >= 4)
    [set meaning "Futurum"]

;      -- křižovatka --
  if (pxcor >= -2 and pxcor <= 3) and (pycor >= -3 and pycor <= 3)[
    set meaning "crossroad"
    set pcolor grey
  ]

;      -- podklad pro semaforey --
  if (pxcor = 4 and (pycor >= 1 and pycor <= 3));Futurum
    [set pcolor yellow
  set traffic-light 1]

  if ((pxcor >= -2 and pxcor <= 0) and pycor = 4);UHK
    [set pcolor yellow
  set traffic-light 2]

  if ((pxcor >= 1 and pxcor <= 3) and pycor = -4);Hospital
    [set pcolor yellow
  set traffic-light 3]

  if (pxcor = -3 and pycor = -3);PardubiceA
    [set pcolor yellow
  set traffic-light 4]

  if (pxcor = -3 and (pycor >= -2 and pycor <= -1));PardubiceB
    [set pcolor yellow
  set traffic-light 5]

;      -- okolí křižovatky --
  if (pxcor <= -3 and pycor >= 4)
    [set pcolor green]
  if (pxcor <= -2 and pycor <= -4)
    [set pcolor green]

```

```

if (pxcor >= 4 and pycor >= 4)
  [set pcolor green]
if (pxcor >= 4 and pycor <= -3)
  [set pcolor green]
]
end

;;----- DRAW CARS -----;;
to draw_cars
  ask n-of pocet_aut_UHK patches with [meaning = "UHK" and pycor <= 17][
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 180
        set meaning-car "UHK"
      ]
    ]
  ]

  ask n-of pocet_aut_Hospital patches with [meaning = "Hospital" and pycor >= -17] [
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 0
        set meaning-car "Hospital"
      ]
    ]
  ]

  ask n-of pocet_aut_Pardubice patches with [meaning = "Pardubice" and pxcor >= -17] [
    if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on patch
(pxcor - 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 90

```

```

    set meaning-car "Pardubice"
  ]
]
]

ask n-of pocet_aut_Futurum patches with [meaning = "Futurum" and pxcor <= 17][
  if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
(pxcor + 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
  sprout-cars 1 [
    set_car
    set heading 270
    set meaning-car "Futurum"
  ]
]
]
end

```

```

to set_car
  set size 1
  set shape "car"
  set time-in-crossroad 0
  ;set label time-in-crossroad
  set variable 0
end

```

```
;; ----- MOVE CARS -----;;
```

```

to move_cars
  ask cars with [ (xcor = 17 and ycor < 0) or (xcor > 0 and ycor = 17) or (xcor = -17 and ycor > 0)
or (xcor < 0 and ycor = -17)][
  die
]

```

```

ask cars with [meaning != "crossroad"] ; pohyb mimo křižovatku
  if not any? cars-on patch-ahead 1 [
    if not any? (lightsFuturum-on patch-ahead 1) with [color = red or color = orange][
      if not any? (lightsUHK-on patch-ahead 1) with [color = red or color = orange][
        if not any? (lightsHospital-on patch-ahead 1) with [color = red or color = orange][

```



```

]
]

if (meaning-car = "UHK")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "UHK" ] ] of
patch-ahead 1 = 0[
  move-to patch-ahead 1
  ][
  if not any? cars-on patch-ahead 1[
    move-to patch-ahead 1
  ]
]
]
]
if (meaning-car = "Hospital")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "Hospital" ] ] of
patch-ahead 1 = 0[
  move-to patch-ahead 1
  ][
  if not any? cars-on patch-ahead 1[
    move-to patch-ahead 1
  ]
]
]
]
]
]
]
]
]
]
end

```

```

to turn_car ;pravidla pro odbočování aut
if variable <= 0 [
  move-to patch-ahead 0 lt heading-car
  set variable variable + 1
]

```

```

if variable >= 1 [
  if (meaning-car = "Pardubice" or meaning-car = "Futurum") [
    if not any? cars-on patch-ahead 1 [
      move-to patch-ahead 1
    ]
  ]
]
if (meaning-car = "UHK") [
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "UHK" ] ] of patch-
ahead 1 = 0 [
    move-to patch-ahead 1
  ] [
    if not any? cars-on patch-ahead 1 [
      move-to patch-ahead 1
    ]
  ]
]
if (meaning-car = "Hospital") [
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "Hospital" ] ] of patch-
ahead 1 = 0 [
    move-to patch-ahead 1
  ] [
    if not any? cars-on patch-ahead 1 [
      move-to patch-ahead 1
    ]
  ]
]
end

```

```

to creat_cars
  if random 20 > hustota_provozu_UHK [
    ask n-of 1 patches with [meaning = "UHK" and pycor >= 17 and (pxcor >= -2 or pxcor <= 0) ] [
      if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
patch pxcor (pycor - 1) and not any? patches with [meaning = "crossroad" ] in-radius 2 [
        sprout-cars 1 [
          set_car
          set heading 180
        ]
      ]
    ]
  ]
end

```

```

    set meaning-car "UHK"
  ]
]
]
]

if random 20 > hustota_provozu_Hospital [
  ask n-of 1 patches with [meaning = "Hospital" and pycor <= -17] [
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
    patch pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 0
        set meaning-car "Hospital"
      ]
    ]
  ]
]

if random 10 > hustota_provozu_Pardubice [
  ask n-of 1 patches with [meaning = "Pardubice" and pxcor <= -17] [
    if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on
    patch (pxcor - 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 90
        set meaning-car "Pardubice"
      ]
    ]
  ]
]

if random 10 > hustota_provozu_Futurum [
  ask n-of 1 patches with [meaning = "Futurum" and pxcor >= 17] [
    if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
    (pxcor + 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [

```



```

    set_car
    set heading 270
    set meaning-car "Futurum"
  ]
]
]
]
end

```

```
;;----- TRAFFIC LIGHTS -----;;
```

```

to traffic_lights
  ask patches with [traffic-light = 1][
    sprout-lightsFuturum 1 [
      set color orange
      set shape "lights"
    ]
  ]
]

```

```

  ask patches with [traffic-light = 2][
    sprout-lightsUHK 1 [
      set color red
      set shape "lights"
    ]
  ]
]

```

```

  ask patches with [traffic-light = 3][
    sprout-lightsHospital 1 [
      set color red
      set shape "lights"
    ]
  ]
]

```

```

  ask patches with [traffic-light = 4][
    sprout-lightsPardubiceA 1 [
      set color red
      set shape "lights"
    ]
  ]
]

```

```

]

ask patches with [traffic-light = 5][
  sprout-lightsPardubiceB 1 [
    set color red
    set shape "lights"
  ]
]
end

```

```

to traffic_light_set_orange
  if (lights-change-tick = 0)[
    ask lightsFuturum [set color orange]
    ask lightsUHK [set color red]
    ask lightsHospital [set color red]
    ask lightsPardubiceA [set color red]
    ask lightsPardubiceB [set color red]
  ]

```

```

  if (lights-change-tick = 24)[
    ask lightsFuturum [set color red]
    ask lightsUHK [set color orange]
    ask lightsHospital [set color orange]
    ask lightsPardubiceA [set color red]
    ask lightsPardubiceB [set color red]
  ]

```

```

  if (lights-change-tick = 42)[
    ask lightsFuturum [set color red]
    ask lightsUHK [set color red]
    ask lightsHospital [set color green]
    ask lightsPardubiceA [set color orange]
    ask lightsPardubiceB [set color red]
  ]

```

```

  if (lights-change-tick = 53)[
    ask lightsFuturum [set color red]
  ]

```

```
ask lightsUHK [set color red]
ask lightsHospital [set color red]
ask lightsPardubiceA [set color green]
ask lightsPardubiceB [set color orange]
]
```

```
traffic_light_set_green
end
```

```
to traffic_light_set_green
ask lightsFuturum with [color = orange][
  if not any? cars-on patches with [meaning = "crossroad"] [
    set color green
  ]
]
```

```
ask lightsUHK with [color = orange][
  if not any? cars-on patches with [meaning = "crossroad"] [
    set color green
  ]
]
```

```
ask lightsHospital with [color = orange][
  if not any? cars-on patches with [meaning = "crossroad"] [
    set color green
  ]
]
```

```
ask lightsPardubiceA with [color = orange][
  set color green
]
```

```
ask lightsPardubiceB with [color = orange][
  any_cars_crossroad
]
end
```

```
to any_cars_crossroad
  if not any? cars-on patches with [meaning = "crossroad"]
    set color green
  ]
end
```

```
to set_lights_interval
  set lights-change-tick (lights-change-tick + 0.5)
```

```
  if lights-change-tick = 89[
    set lights-change-tick 0
  ]
end
```

11.3 Kód NetLogo – Varianta II.

```
breed[cars car]
breed[lightsFuturum lightFuturum]
breed[lightsUHK lightUHK]
breed[lightsHospital lightHospital]
breed[lightsPardubice lightPardubice]
```

```
globals [
  lights-change-tick      ;pomocná proměnná pro semaforey
  average_time_spent_crossroad  ;počítá průměrný čas strávený na křižovatce
]
```

```
cars-own [
  meaning-car            ;označení aut z určitého směru
  time-in-crossroad      ;čas strávený na křižovatce - počítá se podle ticks
  preference-give        ;auta, která dávají přednost
  preference-has         ;auta, která dostávají přednost
  variable               ;pomocná proměnná při odbočování (otáčení auta do správného
směru)
]
```

```
patches-own [
  meaning                ;označení určité části silnice v modelu
  traffic-light          ;patče se semaforey
  turning-point          ;body, na kterých se sjíždí z kruhového objezdu
  heading-car            ;natočení auta v různých turning-point
  give-preference-variable  ;dává políčku vlastnost, kterou následně předá autům - auta,
která dávají přednost
  give-preference        ;na tomto políčku je nutné dát přednost autu, které ji má
  has-preference         ;kontrola těchto políček, zda vůbec může auto zahájit odbočování
- sjezd z křižovatky
  has-preference-variable  ;dává políčku vlastnost, kterou následně předá autům - auta,
která dostávají přednost
]
```

```

to setup
  ca
  reset-ticks

  draw_background          ;vykreslení pozadí
  draw_cars                ;první auta
  set_turning_points       ;nastavení turning-point
  traffic_lights           ;nastavení semaforů
  set_preference_patches   ;preference - nastvení pro řízení v křižovatce

  set lights-change-tick 0 ;nastvení pomocné proměnné pro řízení semaforů
  set average_time_spent_crossroad 0 ;původní nastavení proměnné na 0

  import-drawing "Varianta_2_odbočovací pruh.png" ;vložen obrázek jako pozadí modelu
end

to go
  move_cars                ;pohyb aut v modelu
  creat_cars               ;tvorba aut v modelu během chodu
  traffic_light_set_orange ;procedura pro řízení semaforů
  set_lights_interval      ;procedura pro řízení semaforů
  set_preference_cars      ;preference - nastvení pro řízení v křižovatce

  ask cars [               ;čas strávený v křižovatce
    set time-in-crossroad (time-in-crossroad + 1)
    ;set label time-in-crossroad
  ]

  set average_time_spent_crossroad ((sum [time-in-crossroad] of cars)/(count cars))
;nastavení proměnné - průměrný čas strávený na křižovatce

  tick
end

;;----- PROPERTIES PATCHES -----;;
to set_turning_points
;      -- auta od Pardubic --

```

```

ask patches with [pxcor = -2 and pycor = -4][ ; odbočovací pruh
  set turning-point 1
  set heading-car 270
]
ask patches with [pxcor = 3 and pycor = -1][
  set turning-point 1
  set heading-car 90
]
;      -- auta od UHK --
ask patches with [(pxcor = -3 and pycor = 3)][ ; odbočovací pruh
  set turning-point 2
  set heading-car 270
]
ask patches with [pxcor = 0 and pycor = -2][
  set turning-point 2
  set heading-car 90
]
;      -- auta od Futura --
ask patches with [pxcor = -1 and pycor = 1][
  set turning-point 3
  set heading-car 90
]
ask patches with [pxcor = 3 and pycor = 3][ ; možná odbočí
  set turning-point 5
  set heading-car 270
]
;      -- auta od nemocnice --
ask patches with [(pxcor = 1 and pycor = 2) or (pxcor = 2 and pycor = 3)][
  set turning-point 4
  set heading-car 90
]
ask patches with [pxcor = 4 and pycor = -3][ ; odbočovací pruh
  set turning-point 4
  set heading-car 270
]
end

```

```

;; ----- PREFERENCE -----;;
to set_preference_patches
ask patches with [(pxcor = 2 and pycor = -4) or
  (pxcor = 1 and pycor = -4) or
  (pxcor = 0 and pycor = 4)][
  set give-preference-variable 1
]

ask patches with [(pxcor = -3 and pycor = 4) or ;odbočovací pruhy - dají přednos ostatním
autům
  (pxcor = -4 and pycor = 4) or
  (pxcor = 4 and pycor = 4)][
  set give-preference-variable 2
]

ask patches with [(pxcor = -1 and pycor = 2) or
  (pxcor = 3 and pycor = -2)][
  set give-preference 1
]

ask patches with [(pxcor = -1 and pycor = 3)][ ;kontroluji i další políčka
  set give-preference 2
]

ask patches with [(pxcor = -2 and pycor = 4)][ ;pomoc při odbočování od nemocnice na
Pardubice
  set give-preference 3
]

ask patches with [(pxcor = -3 and pycor = 3) or ;odbočovací pruhy - dají přednos ostatním
autům
  (pxcor = -2 and pycor = 4) or
  (pxcor = 4 and pycor = 5)][
  set give-preference 4
]

ask patches with [(pxcor = -1 and pycor = 4) or ;kontroluji i další políčka

```



```

        (pxcor = -2 and pycor = 5)][
set has-preference 1
]

ask patches with [(pxcor = 3 and pycor = -4) or
        (pxcor = -1 and pycor = 4) or
        (pxcor = -2 and pycor = 5)][
set has-preference-variable 1
]
end

to set_preference_cars
ask cars-on patches with [give-preference-variable = 1 ][
set preference-give 1
]

ask cars-on patches with [has-preference-variable = 1][
set preference-has 1
]

ask cars-on patches with [give-preference-variable = 2 ][
set preference-give 2
]
end

;; ----- DRAW BACKGROUND -----;;
to draw_background
ask patches [
;      -- silnice od UHK --
if (pxcor <= 3 and pxcor >= -3) and pycor >= 4[
ifelse ((pxcor >= 1 and pxcor <= 2) )
[set pcolor brown]
[set pcolor black]
]
if ((pxcor >= -3 and pxcor <= 0) and pycor >= 4)
[set meaning "UHK"]

```

```

;      -- silnice od nemocnice --
if (pxcor <= 4 and pxcor >= -2) and pycor <= -4[
  ifelse (pxcor <= 0 and pxcor >= 0)
    [set pcolor brown]
    [set pcolor black]
]
if ((pxcor >= 1 and pxcor <= 4) and pycor <= -4)
  [set meaning "Hospital"]

;      -- silnice od Pardubic --
if pxcor <= -2 and (pycor >= -4 and pycor <= 3) [
  ifelse (pycor >= 0 and pycor <= 1)
    [set pcolor brown]
    [set pcolor black]
]
if ((pycor >= -4 and pycor <= -1) and pxcor <= -2)
  [set meaning "Pardubice"]

;      -- silnice od Futura --
if pxcor >= 4 and (pycor >= -3 and pycor <= 3)[
  ifelse (pycor <= 0 and pycor >= -1)
    [set pcolor brown]
    [set pcolor black]
]
if ((pycor >= 1 and pycor <= 3) and pxcor >= 4)
  [set meaning "Futurum"]

;      -- křižovatka --
if (pxcor >= -2 and pxcor <= 3) and (pycor >= -3 and pycor <= 3)[
  set meaning "crossroad"
  set pcolor grey
]

;      -- podklad pro semaforey --
if (pxcor = 4 and (pycor >= 1 and pycor <= 3));Futurum
  [set pcolor yellow
  set traffic-light 1]

```

```

if ((pxcor >= -2 and pxcor <= 0) and pycor = 4);UHK
  [set pcolor yellow
set traffic-light 2]

if ((pxcor >= 1 and pxcor <= 3) and pycor = -4);Hospital
  [set pcolor yellow
set traffic-light 3]

if (pxcor = -3 and (pycor >= -3 and pycor <= -1));Pardubice
  [set pcolor yellow
set traffic-light 4]

;      -- okolí křižovatky --
if (pxcor <= -4 and pycor >= 4)
  [set pcolor green]
if (pxcor <= -3 and pycor <= -5)
  [set pcolor green]
if (pxcor >= 4 and pycor >= 4)
  [set pcolor green]
if (pxcor >= 5 and pycor <= -4)
  [set pcolor green]
]
end

;; ----- DRAW CARS ----- ;;
to draw_cars
  ask n-of pocet_aut_UHK patches with [meaning = "UHK" and pycor <= 17][
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 180
        set meaning-car "UHK"
      ]
    ]
  ]
]

```

```

ask n-of pocet_aut_Hospital patches with [meaning = "Hospital" and pycor >= -17] [
  if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
  pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
    sprout-cars 1 [
      set_car
      set heading 0
      set meaning-car "Hospital"
    ]
  ]
]

```

```

ask n-of pocet_aut_Pardubice patches with [meaning = "Pardubice" and pxcor >= -17] [
  if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on patch
  (pxcor - 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
    sprout-cars 1 [
      set_car
      set heading 90
      set meaning-car "Pardubice"
    ]
  ]
]

```

```

ask n-of pocet_aut_Futurum patches with [meaning = "Futurum" and pxcor <= 17] [
  if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
  (pxcor + 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
    sprout-cars 1 [
      set_car
      set heading 270
      set meaning-car "Futurum"
    ]
  ]
]
end

```

```

to set_car
  set size 1

```



```

ask cars with [meaning = "crossroad"] ; pohyb v křižovatce
ifelse (meaning-car = "Pardubice" and turning-point = 1) or
  (meaning-car = "UHK" and turning-point = 2) or
  (meaning-car = "Futurum" and turning-point = 3) or
  (meaning-car = "Hospital" and turning-point = 4)[
  turn_car
]
ifelse (random 50 < prav_odbocovani) and (meaning-car = "Futurum" and turning-point =
5)[
  turn_car
]
ifelse (preference-give = 1 and
  [give-preference] of patch-ahead 1 = 1 and
  any? neighbors with [ any? cars-here with [preference-has = 1]])[
  if meaning-car = "UHK" ;auta, která jedou od UHK a odbočují
směr Futurum
  move-to patch-ahead 0
]
if (meaning-car = "Hospital" and ;auta, která jedou od nemocnice
směrem na Pardubice v levém odbočovacím pruhu
  any? cars-on patches with [give-preference = 3])[
  move-to patch-ahead 0
]
]
ifelse (preference-give = 1 and ;auta, která jedou od nemocnice
směrem na Pardubice v pravém odbočovacím pruhu
  [give-preference] of patch-ahead 1 = 2 and
  any? cars-on patches with [has-preference = 1] and
  any? lightsUHK with [color = green])[
  move-to patch-ahead 0
]
]
ifelse (preference-give = 2 and ;pravidlo pro auta v odbočovacím
pruhu, aby pustilo auta jedoucí z křižovatky
  [give-preference] of patch-ahead 1 = 4)[
  move-to patch-ahead 0
]
]

```

```

if meaning-car = "Futurum" or meaning-car = "Pardubice"[
  if not any? cars-on patch-ahead 1[
    move-to patch-ahead 1
  ]
]
if (meaning-car = "UHK")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "UHK" ] ] of
patch-ahead 1 = 0[
  move-to patch-ahead 1
  ][
  if not any? cars-on patch-ahead 1[
    move-to patch-ahead 1
  ]
]
]
if (meaning-car = "Hospital")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "Hospital" ] ] of
patch-ahead 1 = 0[
  move-to patch-ahead 1
  ][
  if not any? cars-on patch-ahead 1[
    move-to patch-ahead 1
  ]
]
]
]
]
]
]
]
]
]
end

```

```

to turn_car
if variable <= 0 [
  move-to patch-ahead 0 lt heading-car
  set variable variable + 1
]

```

```

]

if variable >= 1 [
  if (meaning-car = "Pardubice" or meaning-car = "Futurum")[
    if not any? cars-on patch-ahead 1[
      move-to patch-ahead 1
    ]
  ]
]
if (meaning-car = "UHK")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "UHK" ] ] of patch-
ahead 1 = 0[
    move-to patch-ahead 1
  ][
    if not any? cars-on patch-ahead 1[
      move-to patch-ahead 1
    ]
  ]
]
if (meaning-car = "Hospital")[
  ifelse (preference-give = 1) and [count cars-here with [ meaning-car = "Hospital" ] ] of patch-
ahead 1 = 0[
    move-to patch-ahead 1
  ][
    if not any? cars-on patch-ahead 1[
      move-to patch-ahead 1
    ]
  ]
]
end

```

```

to creat_cars
  if random 20 > hustota_provozu_UHK [
    ask n-of 1 patches with [meaning = "UHK" and pycor >= 17 and (pxcor >= -2 or pxcor <= 0) ][
      if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
patch pxcor (pycor - 1) and not any? patches with [meaning = "crossroad" ] in-radius 2 [
        sprout-cars 1 [

```



```

    set_car
    set heading 180
    set meaning-car "UHK"
  ]
]
]
]

if random 20 > hustota_provozu_Hospital [
  ask n-of 1 patches with [meaning = "Hospital" and pycor <= -17] [
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
    patch pxcor (pycor - 1) and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 0
        set meaning-car "Hospital"
      ]
    ]
  ]
]

if random 10 > hustota_provozu_Pardubice [
  ask n-of 1 patches with [meaning = "Pardubice" and pxcor <= -17] [
    if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on
    patch (pxcor - 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 90
        set meaning-car "Pardubice"
      ]
    ]
  ]
]

if random 10 > hustota_provozu_Futurum [
  ask n-of 1 patches with [meaning = "Futurum" and pxcor >= 17] [

```

```

    if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
(pxcor + 1) pycor and not any? patches with [meaning = "crossroad"] in-radius 2 [
  sprout-cars 1 [
    set_car
    set heading 270
    set meaning-car "Futurum"
  ]
]
]
]
end

```

```
;;----- TRAFFIC LIGHTS -----;;
```

```

to traffic_lights
  ask patches with [traffic-light = 1][
    sprout-lightsFuturum 1 [
      set color orange
      set shape "lights"
    ]
  ]
]

```

```

ask patches with [traffic-light = 2][
  sprout-lightsUHK 1 [
    set color red
    set shape "lights"
  ]
]

```

```

ask patches with [traffic-light = 3][
  sprout-lightsHospital 1 [
    set color red
    set shape "lights"
  ]
]

```

```

ask patches with [traffic-light = 4][
  sprout-lightsPardubice 1 [

```

```
    set color red
    set shape "lights"
  ]
]
end
```

```
to traffic_light_set_orange
  if (lights-change-tick = 0)[
    ask lightsFuturum [set color orange]
    ask lightsUHK [set color red]
    ask lightsHospital [set color red]
    ask lightsPardubice [set color red]
  ]
```

```
  if (lights-change-tick = 17)[
    ask lightsFuturum [set color red]
    ask lightsUHK [set color orange]
    ask lightsHospital [set color orange]
    ask lightsPardubice [set color red]
  ]
```

```
  if (lights-change-tick = 34)[
    ask lightsFuturum [set color red]
    ask lightsUHK [set color red]
    ask lightsHospital [set color red]
    ask lightsPardubice [set color orange]
  ]
```

```
  traffic_light_set_green
end
```

```
to traffic_light_set_green
  ask lightsFuturum with [color = orange][
    any_cars_crossroad
  ]
```

```
  ask lightsUHK with [color = orange][
```

```

    any_cars_crossroad
]

ask lightsHospital with [color = orange][
    any_cars_crossroad
]

ask lightsPardubice with [color = orange][
    any_cars_crossroad
]
end

to any_cars_crossroad
    if not any? cars-on patches with [meaning = "crossroad"]
        set color green
    ]
end

to set_lights_interval
    set lights-change-tick (lights-change-tick + 0.5)
; set lights-change-tick (lights-change-tick)

    if lights-change-tick = 49[
        set lights-change-tick 0
    ]
end

```

11.4 Kód NetLogo – Varianta III.

```
breed[cars car]
```

```
globals[
```

```
  average_time_spent_crossroad ;počítá průměrný čas strávený na křižovatce
```

```
]
```

```
cars-own[
```

```
  meaning-car ;označení aut z určitého směru
```

```
  time-in-crossroad ;čas strávený na křižovatce - počítá se podle ticks
```

```
  rotated ;otočení aut o potřebné stupně při najíždění na kruhový objezd
```

```
  not-turning ;proměnná, která načítá, že auto nesjelo na sjezdu z kruhového objezdu  
;(po určitém počtu auto musí sjet, aby nejezdilo stále dokola)
```

```
  invisible ;auta, která podjíždí pod kruhovým objezdem mají nastavené hodnotu 1,  
;ostatní mají nastavenou hodnotu 0
```

```
  original-color ;uchovává barvu auta, je to pro potřeby "objevení" po podjetí pod  
kruhovým objezdem
```

```
]
```

```
patches-own[
```

```
  meaning ;označení určité části silnice v modelu
```

```
  turning-point ;body, na kterých se sjíždí z kruhového objezdu
```

```
  invisible-point ;body "zminzení" a "objevení" při podjíždění kruhového objezdu
```

```
  heading-car ;natočení auta v různých turning-point
```

```
]
```

```
to setup
```

```
  ca
```

```
  reset-ticks
```

```
  draw_background ;vykreslení pozadí
```

```
  draw_cars ;první auta
```

```
  set_turning_points ;nastavení turning-point
```

```
  set_invisible_points ;nastavení invisible-point
```

```
  traffic_roundabout ;vykreslení kruhového objezdu
```

```

set average_time_spent_crossroad 0 ;původní nastavení proměnné na 0

import-drawing "Varianta_3_kruhový objezd.png" ;vložen obrázek jako pozadí modelu
end

to go
  move_cars          ;pohyb aut v modelu
  creat_cars         ;tvorba aut v modelu během chodu

  set average_time_spent_crossroad ((sum [time-in-crossroad] of cars)/(count cars))
;nastavení proměnné - průměrný čas strávený na křižovatce

  ask cars [
    set time-in-crossroad (time-in-crossroad + 1) ;nastavení proměnné - čas strávený v
křižovatce
    ;set label time-in-crossroad ;štítek na auta - část strávený v křižovatce
  ]

  tick
end

;; ----- PROPERTIES PATCHES ----- ;;
to set_turning_points
;      -- sjezd na Futurum --
ask patches with [pxcor = 4 and pycor = -3][
  set turning-point 1
  set heading-car 90
]
;      -- sjezd na Pardubice --
ask patches with [pxcor = -4 and pycor = 3][
  set turning-point 1
  set heading-car 270
]
;      -- sjezd na UHK --
ask patches with [pxcor = 3 and pycor = 4][
  set turning-point 1
  set heading-car 0

```

```

]
;          -- sjezd na Hospital --
ask patches with [pxcor = -3 and pycor = -4][
  set turning-point 1
  set heading-car 180
]
end

to set_invisible_points
  ask patches with [abs pxcor = 10 and pycor >= -2 and pycor <= 2][;"podjezd" pod kruhovým
objezdem
  set invisible-point 1
]
end

;;----- DRAW BACKGROUND -----;;
to draw_background
;          -- od nemocnice --
ask patches [
  if (pxcor <= 3 and pxcor >= -3) and pycor <= -5 [
    road_UHK_Hospital
  ]
  if (pxcor = 3 and pycor <= -5)[
    set meaning "Hospital"
  ]
]
;          -- od UHK --
ask patches [
  if (pxcor <= 3 and pxcor >= -3) and pycor >= 5 [
    road_UHK_Hospital
  ]
  if (pxcor = -3 and pycor >= 5)[
    set meaning "UHK"
  ]
]
;          -- od Pardubice --
ask patches [

```

```

if (pxcor <= -9)[
  road_Pardubice_Futurum
]
if (pxcor >= -8 and pxcor <= -6 and abs pycor <= 2)[
  set pcolor blue
]
if (pxcor <= -5)[
  isle
]
if (pycor >= -3 and pycor <= 3 and pxcor <= -7)[
  set meaning "Pardubice"
]
]
;      -- od Futura --
ask patches [
  if (pxcor >= 9)[
    road_Pardubice_Futurum
  ]
  if (pxcor <= 8 and pxcor >= 6 and abs pycor <= 2)[
    set pcolor blue
  ]
  if (pxcor >= 5)[
    isle
  ]
  if (pycor >= -3 and pycor <= 3 and pxcor >= 7)[
    set meaning "Futurum"
  ]
]
end

```

```

to road_UHK_Hospital ;pomocná procedura pro vyreslení pozadí
  if (abs pycor >= 6 and abs pxcor <= 2)[
    set pcolor green
  ]
  if (abs pycor >= 5 and abs pxcor = 3)[
    set pcolor brown
  ]
end

```



```
]
end
```

to road_Pardubice_Futurum ;pomocná procedura pro vyreslení pozadí

```
ifelse (abs pycor >= 1 and abs pycor <= 2)[
  set pcolor black
][
  if (pycor = 0)[
    set pcolor green
  ]
]
end
```

to isle ;pomocná procedura pro vyreslení pozadí

```
if (abs pycor = 3)[
  set pcolor brown
]
end
```

to traffic_roundabout ;vykreslení kruhového objezdu

```
let cx 0          ;střed kruhu
let cy 0          ;střed kruhu
let r 5          ;poloměr kruhu
let p2r ( 2 * pi * r )    ;obvod kruhu
let step p2r / 360    ;jeden krok

crt 1 [          ;tvorba 1 turtle
  setxy cx + r cy    ;nastavit x y
  pd                ;"vysunout" pero a nakreslit kruh
  set heading 0
  while [ p2r > 0 ] [ ;kreslit dokud není hotový kruh
    lt 1            ;natočení
    fd step        ;dopředu o jeden krok
    set p2r p2r - step    ;proměná - počítá kdy je dokreslen kruh
    set color black
    set pcolor blue
    set meaning "Circle"
```

```

]
die
]
end

;;----- DRAW CARS -----;;
to draw_cars
  ask n-of pocet_aut_Hospital patches with [meaning = "Hospital" and pycor >= -17][
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
    pxcor (pycor - 1) and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 0
        set rotated 0
        set meaning-car "Hospital"
      ]
    ]
  ]

  ask n-of pocet_aut_UHK patches with [meaning = "UHK" and pycor <= 17][
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on patch
    pxcor (pycor - 1) and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 180
        set meaning-car "UHK"
      ]
    ]
  ]

  ask n-of pocet_aut_Pardubice patches with [meaning = "Pardubice" and pxcor >= -17 and
  pxcor <= -11 and pycor <= -1][
    if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on patch
    (pxcor - 1) pycor and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 90

```

```

    set meaning-car "Pardubice"
  ]
]
]

ask n-of pocet_aut_Futurum patches with [meaning = "Futurum" and pxcor <= 17 and pxcor
>= 11 and pycor >= 1] [
  if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
(pxcor + 1) pycor and not any? patches with [meaning = "crossing"] in-radius 2 [
  sprout-cars 1 [
    set _car
    set heading 270
    set meaning-car "Futurum"
  ]
]
]
end

to set_car
  set size 1
  set shape "car"
  set not-turning 0
  set invisible 0
  set original-color color

;set label extract-rgb color
;set label time-in-crossroad
end

;;----- MOVE CARS -----;;
to move_cars
  ask cars with [(xcor = 17 and ycor < 0) or (xcor > 0 and ycor = 17) or (xcor = -17 and ycor > 0)
or (xcor < 0 and ycor = -17)][
  die
]

ask cars with [invisible-point = 0 and invisible = 1][ ;pokud je auto v "podjezdu"

```

```
move-to patch-ahead 1  
]
```

ask cars with [meaning = "Circle" and turning-point = 1][;sjezd z kruhového objezdu, jestliže si minul 3 výjezdy nyní odboč

```
ifelse not-turning >= 4 [  
  turn_cars  
][  
ifelse random 10 > pravdepodobnost_sjeti [  
  turn_cars  
][  
  set not-turning not-turning + 1  
  move_around_circle  
]  
]  
]
```

ask cars with [meaning = "Circle" and invisible = 0 and turning-point != 1][;pohyb na kruhovém objezdu

```
move_around_circle  
]
```

ask cars with [meaning != "Circle" and not any? cars-on patch-ahead 2][;nájezd na kruhový objezd

```
if invisible-point = 1 and invisible = 0 [  
  set invisible 1  
  set color lput 80 extract-rgb color  
  move-to patch-ahead 1  
]
```

```
if invisible-point = 1 and invisible = 1 [  
  set invisible 0  
  set color original-color  
  move-to patch-ahead 1  
]
```

```
if not any? neighbors with [ any? cars-here with [invisible = 0 and meaning = "Circle"]][  
  if invisible = 0 [  
    move-to patch-ahead 1  
  ]  
]
```

```

    if meaning = "Circle"[
      set heading (atan pxcor pycor) - 90
    ]
  ]
]
]
end

```

```

to turn_cars ;sjíždění z kruhového objezdu
  set heading heading-car
  move-to patch-ahead 1
end

```

```

to move_around_circle ;pohyb po kruhovém objezdu, počítá se, že jeden "krok" trvá pět *
step
  let r sqrt (pxcor * pxcor + pycor * pycor) ;výpočet poloměru
  let p2r ( 2 * pi * r )
  let step p2r / 360
  let index 5
  while [ index > 0 ][
    lt 1
    fd step
    set index index - 1
  ]
end

```

```

to creat_cars
  if random 20 > hustota_provozu_UHK [
    ask n-of 1 patches with [meaning = "UHK"] [
      if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
      patch pxcor (pycor - 1) and not any? patches with [meaning = "crossing"] in-radius 2 [
        sprout-cars 1 [
          set_car
          set heading 180
          set meaning-car "UHK"
        ]
      ]
    ]
  ]
end

```

```
]
]
```

```
if random 20 > hustota_provozu_Hospital [
  ask n-of 1 patches with [meaning = "Hospital"] [
    if not any? cars-on patch pxcor (pycor + 1) and not any? cars-here and not any? cars-on
    patch pxcor (pycor - 1) and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 0
        set meaning-car "Hospital"
      ]
    ]
  ]
]
```

```
if random 10 > hustota_provozu_Pardubice [
  ask n-of 1 patches with [meaning = "Pardubice" and pxcor <= -17 and pycor <= -1] [
    if not any? cars-on patch (pxcor + 1) pycor and not any? cars-here and not any? cars-on
    patch (pxcor - 1) pycor and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 90
        set meaning-car "Pardubice"
      ]
    ]
  ]
]
```

```
if random 10 > hustota_provozu_Futurum [
  ask n-of 1 patches with [meaning = "Futurum" and pxcor >= 17 and pycor >= 1] [
    if not any? cars-on patch (pxcor - 1) pycor and not any? cars-here and not any? cars-on patch
    (pxcor + 1) pycor and not any? patches with [meaning = "crossing"] in-radius 2 [
      sprout-cars 1 [
        set_car
        set heading 270
        set meaning-car "Futurum"
      ]
    ]
  ]
]
```

```
    ]  
  ]  
] ]  
] ]  
end
```

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2018/2019

Studijní program: Systémové inženýrství a informatika
Forma: Prezenční
Obor/komb.: Informační management (im2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Nosilová Martina	Trpišov 113, Slatiňany - Trpišov	I1700681

TÉMA ČESKY:

Využití simulačních metod pro vizuální analytiku

TÉMA ANGLICKY:

The use of simulation methods for visual analytics

VEDOUcí PRÁCE:

Ing. Karel Mls, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je analýza simulačních přístupů, nástrojů a jejich vizualizačních možností. Navrhnout model a ověřit na konkrétní simulaci.

Obsah:

Úvod, rešerše tématu

Modelování (přístupy, metody, nástroje)

Simulace (základní pojmy, modely, využití simulace)

Vizuální analýza

Analýza simulačních přístupů

Závěr

SEZNAM DOPORUČENÉ LITERATURY:


Wilensky, U., Rand, W. An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo. 2015.

PELÁNEK, Radek. Modelování a simulace komplexních systémů. Jak lépe porozumět světu. 1. vyd. Brno: Masarykova univerzita, 2011. 236 s. mimo edice. ISBN 978-80-210-5318-2.

Fieguh, P. An Introduction to Complex Systems Society, Ecology, and Nonlinear Dynamics. 2016. ISBN 978-3319446059.

Podpis studenta: 

Datum: 10.10.2018

Podpis vedoucího práce: 

Datum: 10.10.2018