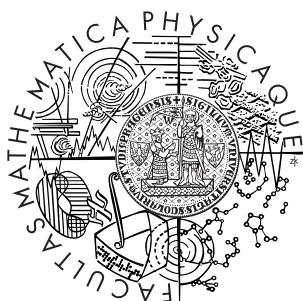


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



René Fischer

Simulace dopravní sítě

Katedra aplikované matematiky

Vedoucí bakalářské práce: RNDr. Ondřej Pangrác, Ph.D.

Studijní program: Informatika, Programování

2008

Na tomto mieste by som sa chcel podakovať vedúcemu mojej bakalárskej práce pánovi RNDr. Ondřejovi Pangrácoví, Ph.D., za jeho podnety, pripomienky, pomoc, ochotu a trpežlivosť po dobu jej spracovávania.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapújčováním práce a jejím zveřejňováním.

V Praze dne 30.5.2008

René Fischer

Obsah

1	Úvod	6
1.1	Motivácia	6
1.2	Ciel	7
1.3	Využitie	7
2	Návrh	8
2.1	Téma	8
2.2	Simulácia	8
2.3	Druhy simulácie	9
2.3.1	Spojité simulácia	9
2.3.2	Diskrétna simulácia	9
2.4	Špecifikácia a návrh programu	10
2.4.1	Požiadavky na funkčnosť programu	10
2.4.2	Platforma	10
3	Popis funkčnosti programu	12
3.1	Výber typu simulácie	12
3.2	Prvky siete	12
3.3	Čas v simulácii	14
3.3.1	Trvanie simulácie	14
3.4	Vstúpi auto do simulácie?	14
3.4.1	Vstúpi auto zo vstupu?	14
3.4.2	Vystúpi auto z parkoviska?	15
3.5	Určovanie ciela automobilom	15
3.5.1	Tvorba rozhodovacieho intervalu	16
3.6	Výber trasy v dopravnej sieti	16
3.6.1	Naplnenie štruktúr riadiacich rozhodovanie automobilov	16
3.6.2	Miera informovanosti vodičov	17

3.7	Chod simulácie	18
3.7.1	Prvý cyklus	18
3.7.2	Hlavný cyklus simulácie	18
3.7.3	Tretí cyklus	22
3.8	Optimalizácia	22
3.8.1	Zelená vlna	22
3.8.2	Vylepšenie križovatiek	23
4	Implementácia	25
4.1	Výber vývojových nástrojov	25
4.2	Objektový návrh programu	26
4.2.1	Popis a funkčnosť objektov	27
5	Užívateľská dokumentácia	31
5.1	Inštalácia	31
5.2	Hlavné menu	31
5.3	Vykreslovanie	32
5.4	Mód Editor	33
5.5	Mód Simulácia	36
5.5.1	Štatistiky a optimalizácia	37
5.5.2	Optimalizácia	37
5.6	Ukladanie-načítanie máp-priebehov	38
6	Záver	39
6.1	Praktické skúsenosti a poučenia	39
6.1.1	Odchýlky od špecifikácie	39
6.2	Ďalší vývoj	40
Literatúra		41

Název práce: Simulace dopravní sítě

Autor: René Fischer

Katedra (ústav): Katedra aplikované matematiky

Vedoucí bakalářské práce: RNDr. Ondřej Pangrác, Ph.D.

e-mail vedoucího: Ondrej.Pangrac@mff.cuni.cz

Abstrakt: Program dáva užívateľovi možnosť vytvorenia vlastnej a poupravenia už existujúcej cestnej siete s požadovanými parametrami a pravidlami. Následne bude môcť danú sieť nasimulovať a získať cenné informácie o možnom vývoji premávky v nej. Simulovanie by malo odhaliť vytažené miesta siete, na ktorých by mohli nastať v skutočnej premávke zápchy. Na základe pozorovania bude môcť poupraviť sieť spôsobom, ktorý by mal alebo predísť nechcené situácie. Program zahŕňa aj možnosť obmedzenej optimálizácie.

Klíčová slova: doprava, simulácia, dopravná siet, dopravná simulácia

Title: Traffic network simulation

Author: René Fischer

Department: Department of Applied Mathematics

Supervisor: RNDr. Ondřej Pangrác, Ph.D.

Supervisor's e-mail address: Ondrej.Pangrac@mff.cuni.cz

Abstract: The programme offers to the user the option to create his own traffic network or to adapt already existing traffic network, all with parameters and rules required by user. Afterwards the user can make a simulation of the network and thus get a valuable informations about possible development of the traffic. Simulation should reveal stressed places in the traffic, where traffic jams would be possible to occur in a reality. Under the observation the user is able to adapt the network in order to resolve unwanted situations. The programme also involves option of constricted simulation.

Keywords: traffic, simulation, traffic network, traffic simulation

Kapitola 1

Úvod

1.1 Motivácia

Časté čakania v autobusoch, električkách a pozorovanie vytažených komunikácií, na ktorých denne vznikali dopravné zápchy, boli inšpiráciou na vytvorenie program Simulácia dopravnej siete. Každé väčšie mesto sa s touto situáciou stretáva pravidelne. Týždenne, denne a v horších prípadoch aj niekolkokrát za deň sa na cestách premieta rovnaká scéna. Cesty preplnené často nervóznymi vodičmi. Tento problém prestáva byť problémom len väčších miest, ale stále častejšie sa s ním môžeme stretnúť aj v čoraz menších mestách. Svedkom tohto javu som sa stal aj v mojom rodnom, 30 tisícovom, meste. Tieto situácie vznikajú hlavne pri pravidelnom pohybe ľudí za prácou, nákupom, dovolenkou, ale aj pri zmenách v cestnej sieti, nesprávnym nastavením semaforov atď.. Dôvodov je nespocetne veľa. Tieto situácie sú riešené zmenami v pravidlach cestnej premávky, zmenami vo vlastnostiach komunikácií. Podkladmi na tieto zmeny môžu pochádzať z informácií zozbieraných na iných cestných sietach, ktoré môžu pochádzať z dlhodobého a prácneho pozorovania správania sa premávkou. Tieto riešenia však často nemusia byť pre miesto, v ktorom sú aplikované správne a môžu viest k ešte väčším problémom, s ktorými ma neskôr plno práce dopravná polícia.

1.2 Cieľ

Program by mal umožniť užívateľovi vymodelovať si svoju vlastnú cestnú sieť podľa svojej potreby, ktorá môže a nemusí odpovedať realite. Hlavným cieľom má byť možnosť spustiť si na užívateľom vytvorenom modely cestnej siete rôzne scenáre premávky a čo najvierohodnejšie nasimulovať správanie vodičov a vývoj dopravnej situácie v sieti. Počas behu simulácie bude užívateľ môcť vyzozorovať slabé, vytažené alebo miesta v sieti, na ktorých by sa v skutočnom svete mohli potenciálne tvoriť dopravné zápchy a narobiť vrásy nie jednému vodičovi. S programom by malo byť možné namodelovať nie len každodennú dopravnú špičku spôsobenú už spomínanou pravidelnou migráciou obyvateľov do práce a z práce, ale aj nepravidelné návaly motoristov stahujúcich sa za predĺženým víkendom na vidiek. Taktiež nepredvídateľné situácie ako dopravné nehody, preludy počasia, ktoré môžu danú komunikáciu na nejakú dobu úplne vyradiť z premávky. Všetko záleží na scenári a nastavených vlastnostiach modelu, ktoré umožňuje nasimulovať množstvo rôznorodých situácií. Do programu by som chcel zahrnúť aj optimalizáciu, ale keďže solídne optimalizačné úpravy sietí sú pomerne rozsiahla a netrieviálne problematika, bude funkcionálita programu zahŕňať iba obmedzenú a jednoduchšiu optimalizáciu. K možnostiam by mali patriť tzv. "Zelená vlna" na slede semaforov, určovanie/odporúčanie hlavných ciest a jednoduchších semaforových intervalov na križovatkách.

1.3 Využitie

Pred uvedením nových semaforov do premávky, začiatím výstavby alebo úpravy cestnej komunikácie by bolo vhodné nejakým spôsobom otestovať, či daný zásah do siete situáciu v premávke vylepší alebo naopak zhorší. Program simulácia dopravnej siete by mal slúžiť k práve spomínanému testovaniu. K prvoradému využitiu programu by malo patriť namodelovanie, nasimulovanie rozličných, rôznorodých situácií v dopravnej sieti, pozorovanie vývoja premávky a predídenie nechceným situáciám v skutočnej doprave. Zo širšieho hľadiska by mal program umožniť otestovať a pripraviť sa na každú situáciu v dopravnej sieti, bez nebezpečenstva vzniku nechcených situácií v skutočnej premávke.

Kapitola 2

Návrh

2.1 Téma

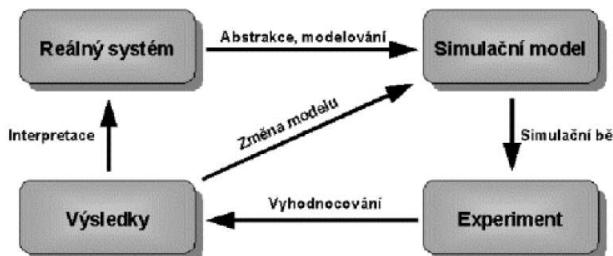
Téma simulácie je zaujímavá a z môjho pohľadu aj veľmi prospešná. Oblast jej využitia sa stále rozrastá. S jej výsledkami sa môžeme denne stretnúť, pocíťovať a vidieť. Podľa predpovede počasia sa človek ráno oblečie, podľa odolnosti a výdrže materiálu bude použitý materiál na stavbu budovy v ktorej pracujete, podľa predpokladaného vývoja na burze sa banka rozhodne investovať Vaše peniaze zarobené v práci. To sú len 3 príklady z množstva, v ktorých sa simulácia používa. V oblastiach ako meteorológia, stavebníctvo, ekonómia, biológia, demografia, chémia a fyzika ... a neposlednom rade doprava sa stala neoddeliteľnou súčasťou. Využitie v ekonómii je jedna z oblastí, ktorej by som sa v budúcnosti rád venoval. Kedže ekonomicke znalosti ešte len plánujem nadobudnúť, vybral som si ďalšiu z možností využitia simulácie, ktorá mi je blízka, a tou je doprava.

2.2 Simulácia

Simulácia je :

- napodobovanie činností systému v priebehu času, pričom vývoj chovania systému môže byť skúmaný pomocou simulačného modelu.[1]
- moderná metóda analýzy zložitých procesov, ktoré obsahujú prvky náhodného a dynamického chovania.[1]

- je proces tvorby modelu reálneho systému a prevádzanie experimentov s týmto modelom za účelom dosiahnutia lepšieho pochopenia chovania študovaného systému či za účelom posúdenia rôznych variant činnosti systému (Shannon).[1]



Obrázok 2.1: Simulácia

alebo empiricky z nameraných hodnôt.[2]

Pred tým, ako začneme študovať chovanie reálneho systému, musíme navrhnúť jeho model. Podstatou pojmu modelovania je náhrada skúmaného systému jeho modelom, presnejšie systémom, ktorý ho modeluje.[1] Model môže byť získaný buďto teoreticky zo základných fyzikálnych vlastností systému,

2.3 Druhy simulácie

Hlavným kritériom delenia simulácií je podľa toho ako prebiehajú v čase. Podľa tohto kritéria delíme simulácie na spojité a diskrétné.

2.3.1 Spojitá simulácia

Ak čas beží po malých ale rovnako veľkých časových intervaloch, nazývame takúto simuláciu spojitou simuláciou.[3] Dĺžka časového intervalu je vopred zvolená. Po každom prebehnutí časového intervalu sa zistuje stav modelu a zmeny, ktoré v modeli nastali. Faktor najviac ovplivňujúci presnosť získaných údajov je veľkosť časového intervalu. Presnosť údajov je nepriamo úmerná dĺžke intervalu. Pri voľbe intervalu je potrebné brať ohľad aj na vytáženie výpočtovej jednotky, v našom prípade CPU, ktoré môže byť pri veľmi krátkom intervale neúmerne vysoké.

2.3.2 Diskrétna simulácia

V diskrétnu simuláciu upravujeme model iba v takých časoch, keď sa niečo deje. To, čo sa deje, budeme označovať ako udalosť. Napríklad v simulácií

prepravy vlakom najprv dôjdete na vlakovú stanicu, krátko na to si kúpite lístok a nastúpite do vlaku. Potom dlho niekam idete a nakoniec vystúpite. To sú udalosti. Medzi týmito udalosťami sú rôzne dlhé časové intervaly a čo sa deje behom nich je pre simuláciu nepodstatné. (Riaditeľovi Českých drah, ktorý si takúto simuláciu objednal, je jedno či vo vlaku spíte alebo či si tam niečo čítate.)[3] Preformulované povedané, diskrétna simulácia je simulácia riadená udalosťami. Tieto udalosti nastávajú v nespojitéch časových okamžikoch. Chovanie modelu a získavanie údajov sa deje len v určitých okamžikoch, v ktorých sú vyvolané nejaké zmeny modelu. Do tzv. simulačného kalendára sa zaznamenávajú a podľa času výskytu zotriedujú udalosti, ktoré majú prebehnúť. Nastatie udalostí môže naplánovať ďalšie udalosti, ktoré budú podľa času výskytu zaradené do simulačného kalendára. V priebehu simulácie sa postupne spracúvajú udalosti zo simulačného kalendára, zvyčajne pokiaľ nie je kalendár prázdný.

2.4 Špecifikácia a návrh programu

2.4.1 Požiadavky na funkčnosť programu

Pred spustením simulácie, je nutné dopravnú sieť vytvoriť. K vytváraniu sietí je potrebné prostredie, ktoré to umožní. Prostredie by malo mať niekoľko hlavných vlastností. Malo by byť pohodlné, užívateľsky priateľské, jasné, aby aj pre užívateľa, ktorý sa s programom stretáva prvýkrát, bola práca s ním dostatočne intuitívna. Prostredie pre tvorbu sietí (dalej ako editor) by malo byť logicky oddelené od prostredia, v ktorom sa bude vykonávať vlastná simulácia (dalej ako simulátor). Z tohto dôvodu by mal mať program 2 módy, v ktorých by sa tieto funkcie vykonávali samostatne. Mód simulátora by sa mal vlastnosťami podobať módu editora a byť čo najviac užívateľsky priateľský, jednoduchý a ovládanie by malo byť opäť čo najintuitívnejšie ...

2.4.2 Platforma

Kedže nebola stanovená požiadavka prenositeľnosti programu na iný operačný systém ako systémy z rodiny MS Windows, bude program spustiteľný iba pod týmito operačnými systémami. Program využíva .Net Framework od firmy Microsoft, čo je modelom pre budovanie systémov na rodine operačných systémov Widnows.[4] Mnou používaná verzia .Net 2.0 je podporovaná operačnými systémami Windows 2000 SP3, 98, ME a vyššie[5], , pre-

to bude program spustiteľný na spomínaných systémoch s nainštalovaným balíčkom .Net Framework 2.0 a vyššie.

Kapitola 3

Popis funkčnosti programu

3.1 Výber typu simulácie

Z vyššie vymenovaných typov simulácie, presnejšie diskrétnej a spojitej, som pre program zvolil spojitú simuláciu. V sieti sa počas simulácie môže naraz nachádzať nesmierne množstvo automobilov. V riešení programu s použitím diskrétnej simulácie by tento pohyb nonstop zahľoval simulačný kalendár. Ten býva zväčša reprezentovaný ako prioritná fronta a jej spracovanie by bolo značne komplikované a predstavovalo veľké zaťaženie pre celý program. Pri použití spojitej simulácie sa spracovaniu veľkého množstva pohybu áut nevyhneme, ale ich spracovanie bude robené v každej iterácii hlavného cyklu simulácie postupné, prechádzaním každej cesty a spracovaní áut na nej. V programe sa vyhneme zaťaženiu spôsobenému spracovaním veľkého množstva udalostí v simulačnom kalendári.

3.2 Prvky siete

Sieť sa bude skladať z ciest a uzlov. Cesty sa druhovo deliť nebudú. Rozlišovať sa budú vo vlastnostiach, dĺžke, prejazdnosti, maximálnej povolenej rýchlosťi a počtu pruhov v každom smere. Prejazdnosť je vlastnosť cesty, ktorá bude ovplyvňovať rýchlosť automobilu, ktorou sa budú automobily pohybovať po komunikáciách. Zvyšné vlastnosti hovoria samé za seba. Uzly sa naproti tomu budú rozlišovať podľa významu a využitia. Budeme ich deliť do piatich hlavných kategórií :

Vstup/výstup je typ uzla, ktorým budú do siete vstupovať a vystupovať autá. V nastaveniach tohto typu uzla bude možné určiť množstvá áut, ktoré majú daným uzlom do siete vstúpiť a vystúpiť počas celého trvania simulácie. Množstvá áut bude možné upresniť samostatne pre každý časový interval(viz. 3.3). Dodržiavanie počtov si užívateľ bude môcť porovnávať so skutočnými hodnotami z behu simulácie(viz 5.5.1). Do tohto typu uzla bude môcť a musieť vstupovať presne 1 obojsmerná alebo jednosmerná cesta. Knižným príkladom tohto typu uzla by mohla byť D1 pri simulácií modelu Prahy.

Parkovisko bude slúžiť ako určitý druh oblasti, kam autá smerujú. Môže predstavovať nákupné centrum, sídlisko, celé mesto alebo obyčajné parkovisko. Ako každé parkovisko bude mať svoju kapacitu a maximálny počet prístupových ciest. Uzol nebude možne navštievovať s úmyslom prejazdu, skratky. Tak ako pri predchádzajúcim type uzla, aj parkovisku sa bude dať vo vlastnostiach nastaviť počet áut, ktoré majú počas určitého časového intervalu uzol navštíviť a priemerná hodnota zaplnenia parkoviska počas určitého intervalu. Aj v tomto prípade bude počas simulácie užívateľ mať možnosť nahliadnuť na skutočné hodnoty návštevnosti a zaplnení.

Križovatka je uzol, ktorý hovorí sám za seba. V nastaveniach bude možné zvoliť typ križovatky, svetelne alebo nesvetelne riadenú. Podľa typu križovatky bude následne, ak to bude mať zmysel, možné nastaviť hlavné cesty. V prípade svetelnej križovatky to bude bezkolízne nastavenie semaforov, trvanie jednotlivých fáz a čas začiatku prvej fázy. V tomto uzle bude ako v jedinom možné, v prípade, že to má zmysel, nastaviť smerovanie pruhov ciest vstupujúcich do križovatky.

Kruhový objazd bude nastaviteľný iba vo vlastnosti určujúcej počet príjazdov. Tomuto počtu bude odpovedať aj jeho veľkosť a zároveň minimálne trvanie času, ktorú auto strávi na okruhu pri jeho prejazde.

Pomocný uzol pomocný uzol bude mať 2 využitia. Prvým je zmena vlastností cesty, napríklad ak sa na ceste mení počet pruhov, maximálna rýchlosť atd.. Druhým vyžitím bude spojenie, respektíve rozdelenie dvoch jednosmerných ciest, do jednej obojsmernej a naopak.

3.3 Čas v simulácii

Kedže som pre tento program zvolil spojitú simuláciu, bolo potrebné určiť základnú časovú jednotku simulácie a čas, ktorý by v skutočnom svete znamenala. Po zvážení som zvolil, že 1 časový interval simulácie bude predstavovať 1 sekundu v skutočnom svete. Dĺžka simulácie by mohla niekedy trvať aj niekoľko hodín, preto som do vlastnosti simulácie pridal možnosť urýchliť chod simulácie, teda zmeniť časovú periódu v programe, predstavujúcu 1 sekundu v skutočnom svete. Z množstva operácií, veľkosti a zložitosti navrhutej siete a výkonu počítača sa nemusí táto vlastnosť plne prejavovať.

3.3.1 Trvanie simulácie

Trvanie simulácie sa bude nastavovať 2 parametrami. Prvý z nich predstavuje trvanie 1 intervalu v počte dĺžok základného časového intervalu simulácie. Druhý parameter predstavuje počet intervalov celej simulácie. Podľa parametra počtu intervalov sú v príslušných uzloch nastavované vytaženia a návštevnosť.

3.4 Vstúpi auto do simulácie?

K prvým rozhodovaniam, ktoré počas simulácie nastanú, je rozhodnutie, či v danom kroku, daným uzlom pojde automobil do simulácie. Autá budú môcť vstupovať iba z 2 typov uzlov, zo vstupov a parkovísk.

3.4.1 Vstúpi auto zo vstupu?

Z užívateľských údajov, ktoré sú nastavované vo vlastnostiach uzlov typu vstup v editori, o počte áut, ktoré majú počas daného časového intervalu vstupom do siete vstúpiť a z dĺžky intervalu sa vyráta percentuálna pravdepodobnosť vstupu auta do simulácie počas 1 časovej jednotky. Tento údaj bude kvôli dodržaniu užívateľských počtov ešte vynásobený koeficientom. Koeficient je vyrátavaný z porovnania počtov áut vystúpených počas daného intervalu s počtami, ktoré by mali v danú dobu intervalu počty teoreticky dosahovať. Podľa tohto porovnania môže koeficient pravdepodobnosť zväčšovať alebo zmenšovať. Následne je generátorom náhodných čísel vygenerované prirodzené číslo z intervalu (0,101). Ak je generované číslo menšie alebo rovné percentuálnej pravdepodobnosti vynásobenej koeficientom, je

pre vstup auta ”daná zelená”, v opačnom prípade bude vstup auta daným uzlom v danom kroku simulácie zakázaný.

3.4.2 Vystúpi auto z parkoviska?

Užívateľsky nastaviteľná vlastnosť, ktorá najviac ovplyvňuje rozhodovanie o výstupe auta z parkoviska je priemerné zaplnenia počas konkrétnych intervalov. Podľa vypočítaného rozdielu medzi aktuálnym a používateľom zadaným priemerným zaplnením je pevne daná hodnota pravdepodobnostného percenta podstatne zväčšená alebo zmenšená. Nasleduje generovanie náhodného čísla a jeho porovnanie s výslednou pravdepodobnosťou výstupu auta. V prípade, že hodnota generovaného čísla je menšia ako veľkosť nami vypočítanej konečnej pravdepodobnosti, bude auto z parkoviska vystupovať, v opačnom prípade, to bude v danom kroku na danom uzle zakázané.

3.5 Určovanie cieľa automobilom

Ciele pre automobily budú určované pre motoristov, ktorí vstupujú na cestnú komunikáciu siete iba z 2 typov uzlov. Ich cieľmi sa môžu stať, ako u vstupovaní do simulácie, znova iba 2 typy uzlov, vstupy a parkoviská. Určovanie cieľa áutomobilom je jedna z najdôležitejších častí programu, ktorá rozhoduje o čo najrealistickejšom správaní automobilov v sieti a teda o vierohodnosti simulovaného modelu. Kedže cieľmi sa môžu stať iba 2 typy uzlov, preto hlavným faktorom ovplyvňujúcim rozhodovanie o celi vstupujúceho auta budú vlastnosti všetkých uzlov týchto dvoch typov. Presnejšie ide o vlastnosť, ktorá určuje počet áut, ktoré majú daný uzol počas určitého intervalu navštíviť. Ďalšími faktormi, nepriamo podielajúcimi sa na výbere, sú aktuálne vzdielenosti potencionálnych cieľov od uzla, ktorým automobil vstupuje na cestu, aktuálny čas simulácie a počty automobilov nachádzajúcich v simulácii na komunikáciách. Zo spomínaných informácií sa vytvorí pomocný rozhodovací interval(viz 3.5.1). Každý podinterval, z ktorých sa skladá rozhodujúci interval, ”patrí” jednému z potencionálnych a dosažiteľných cieľov. Pomocou generátora náhodných čísel sa vygeneruje náhodné číslo patriace do intervalu. ”Vlastník” podintervalu, do ktorého vygenerované číslo padne, bude určený ako cieľ.

3.5.1 Tvorba rozhodovacieho intervalu

V úvode je veľkosť intervalu rovná 0. V iteráciách cyklu, ktorý prechádza každý dostupný uzol, vyrátvame veľkosť podintervalu, ktorého vlastníkom bude momentálne spracovaný uzol. V každej iterácii bude rozhodovací interval rozšírený o veľkosť práve vyrátaného podintervalu. Veľkosť podintervalu v pomere k celkovej výslednej veľkosti rozhodovacieho intervalu je rovná pravdepodobnosti, že vlastník daného podintervalu sa stane cieľom. Vzdialenosť uzlov v pomocných štruktúrach(viz 3.7) je uchovávaná ako čas potrebný k premiestneniu sa z jedného do druhého uzla. Časový interval simulácie, do ktorého by patril súčet aktuálneho času simulácie a času uchovávaného ako vzdialenosť oboch uzlov, označíme ako interval M. Určíme rozdiel medzi počtom áut smerujúcich do spracovávaného uzlu, ktorých predpokladaný príchod je v rámci intervalu M a počtu áut, ktoré by mali spracovaný uzol počas intervalu M navštíviť. Rozdiel vynásobíme koeficientom, ktorý je určený podľa veľkosti času, ktorý by mal teoreticky zostať autu po dorazení do cieľa, do uplynutia intervalu M. Veľkosť výsledného čísla sa rovná veľkosti podintervalu.

3.6 Výber trasy v dopravnej sieti

K orientácií vodičov v skutočnom svete sú na križovatkách umiestnené značky, ktoré ukazujú smer. Aby sa vedel automobil dostať do jemu zvoleného cieľa, je za potreby mať uložené informácie, podľa ktorých by sa v simulácii orientovali virtuálne automobily. Za týmto účelom boli zavedené pomocné štruktúry, v ktorých sú zaznamenané informácie, podľa ktorých sa pri prejazde uzlami automobily riadia. Informácie v štruktúrach sú dvojeho typu, všeobecného a aktualizovaného. Prvým typom informácií sú riadené neinformovaný motoristi, druhým logicky informovaný(viz 3.6.2).

3.6.1 Naplnenie štruktúr riadiacich rozhodovanie automobilov

Celú dopravnú sieť si môžeme predstaviť ako orientovaný ohodnotený graf (V, E). Kde V bude množina všetkých vrcholov a E množina všetkých hrán medzi vrcholmi z V . Kde vrcholy z V predstavujú naše uzly, a hrany z E naše jednosmerné cesty a obojsmerné cesty rozdelené na 2 jednosmerné. Orientácia hrán je v smere od uzla, z ktorého cesta vychádza do uzla, do

ktorého vchádzajú. Ohodnotenie hrán je určené funkciou, ktorá kombinuje dĺžku, maximálnu rýchlosť a prejazdnosť ciest. Ohodnotenie musí byť nezáporné! Na vyhľadávanie trás medzi jednotlivými vrcholmi grafu som použil kúsok pozmenený Dijkstrov algoritmus. Zmena v algoritme sa týka toho, že po nájdení trasy z počiatočného bodu, do vrcholu reprezentujúceho uzol typu parkovisko alebo vstup, nie je umožnené využívanie tohto vrcholu k ďalšiemu výpočtu. Ďalšie informácie o Dijkstrovom algoritme sú dostupné napríklad v knihe. [7]. Jedným prevedením Dijkstrovho algoritmu sa zisťia najkratšie cesty zo štartovného vrcholu do všetkých dostupných uzlov grafu. Aby sme poznali všetky potrebné informácie o najkratších cestách medzi každými dvoma uzlami v sieti, potrebujeme algoritmus pustiť presne toľko krát, kolko máme v sieti uzlov. Pri každom spustení algoritmu je za počiatočný vrchol zvolený uzol, ktorý ním ešte neboli. Po prevedení algoritmu si nezapisujeme informáciu o celej najkratšej trase medzi 2 uzlami, ale iba identifikátor cesty, na ktorú musíme v danom uzle prejsť, aby sme sa čo najkratšou cestou dostali do nášho cieľa. Uvedieme si príklad. Auto vstúpi do siete vstupom. Príde na prvú križovatku. Pozrie sa na smerovacie tabuľky (do našich pomocných štruktúr), a tie mu povedia, že ak chce ist do "Prahy", musí odbociť doprava. Odbočí a takto sa riadi, až kým nepríde do "Prahy". Správanie informovaných vodičov (viz 3.6.2) je riadené informáciami zo štruktúr, ktoré sú pravidelne každú minútu aktualizované. Takže vodič pri rozhodovaní nepozerá na smerovacie tabuľky, ale do svojho navigačného zariadenia (do aktualizovaných štruktúr).

3.6.2 Miera informovanosti vodičov

V súčasnosti sa u vodičov čoraz viac udomáčňujú navigačné zariadenia používajúce technológiu GPS. Tieto zariadenia pomáhajú vodičom v orientácii v cudzom prostredí a vo výhýbaní sa dopravným zápcham. Preto som do programu zaviedol možnosť nastaviť mieru informovanosti vodičov t.j. polymer vodičov s a bez takého zariadenia. Vstupujúcim vodičom, ktorý bol označený ako informovaný, je pridané navigačné zariadenie, ktoré riadi správanie automobilu v sieti. Navigačné zariadenia predstavujú v programme pravidelne aktualizované riadiace štruktúry (viz 3.6.1).

3.7 Chod simulácie

Celá simulácia pozostáva z pravidelného volania funkcie, v ktorej sa prevedie 1 krok simulácie, t.j. nasimuluje 1 sekunda skutočnej premávky. Vo tejto funkcií sa v 3 základných cykloch nasimuluje spomínané správanie premávky na všetkých uzloch a cestách siete. Volanie funkcie sa opakuje pravidelne po uplynutí momentálne nastaveného základného časového intervalu.

3.7.1 Prvý cyklus

V prvom cykle sa prevedie posunutie áut na pomocných kópiach každého pruhu siete. Na kópiach sa nasimuluje situácia, pri ktorej v danom kroku simulácie neopustí pruh žiadne auto. Priestor vzniknutý na konci pruhu sa zaznamená do pomocnej premennej . Táto premenná bude neskôr využívaná pri druhom najdôležitejšom cykle simulácie. Posun áut po pruhoch bude priblížený neskôr, pri vysvetľovaní tretieho cyklu.

3.7.2 Hlavný cyklus simulácie

Druhý, najpodstatnejší cyklus, spracováva správanie sa na každom uzle mapy počas 1 kroku simulácie. Podstatou tohto spracovania je vpúšťanie, vypúšťanie a prechod áut danými uzlami. Postupne bude vysvetlený postup spracovania každého typu uzlu.

Vstup : Pri kroku simulácie tohto typu uzla sa najprv spracúva cesta vstupujúca do uzla. Pokiaľ existuje v nejakom z pruhov vstupujúcej cesty auto, ktoré splňa podmienky na vstup do uzla, je dané auto odobraté z jeho aktuálneho pruhu a zapísané do štatistik uzla, cesty a pruhu(dalej už iba ako zápis do štatistik). Podmienky na vstup sú dostatočne veľká pozícia v pruhu vzhľadom na dĺžku, maximálnu rýchlosť a prejazdnosť cesty, ktoré prislúcha daný pruh(dalej iba ako podmienky na vstup). Pri spracovaní cesty vystupujúcej z uzlu sa najprv zo štatistik zistuje, či počet áut, ktoré vstúpili do simulácie počas aktuálneho intervalu daným uzlom už nedosahuje užívateľom zadaný predpokladaný počet. Nasleduje cyklus, ktorý určí, či má v danom kroku do simulácie uzlom vstúpiť auto a prípadne, ktorý uzol má byť jeho cieľ. V prípade vstupu auta nasleduje určenie jeho informovanosti a hľadanie pruhu, do ktorého sa vstupujúce auto zaradí. V prípade nájdenia pruhu vhodného na vstup nového auta do simulácie, je vytvorené auto s

vypočítanými vlastnosťami (ciel, informovanosť) a jeho vstup je zapísaný do štatistik a auto je vložené do nájdeného pruhu s prislúchajúcou pozíciou (viz ďalej). Cyklus je ukončený v prípade ak jedna z prípravných funkcií na vstup auta skončí neúspechom.

Parkovisko : Spracovanie kroku simulácie na uzloch tohto typu je dosť podobný spracovaniu uzlov typu vstup. V úvode sa v prípade splnenia časových podmienok zaznamená údaj o zaplnení parkoviska do štatistik. V prípade voľnej parkovacej kapacity uzla nasleduje cyklus, ktorý vpúšta na parkovisko autá splňujúce podmienky na vstup. Každé vstupujúce auto je zapísané do štatistik pruhu, cesty a uzla.. Výstup je, až na rozdiel v rozhodovaní o výstupe auta, riadení rovnakým spôsobom ako pri uzle typu vstup.

Pomocný uzol : Tento typ uzla podľa definície slúži iba na zmenu vlastností cesty a na spojenie alebo rozpojenie 2 jednosmerných ciest na 1 obojsmernú a naopak. Preto spracovanie 1 kroku simulácie pozostáva iba z prenesenia áut z vstupujúcich pruhov 1 cesty do vystupujúcich pruhov 2. cesty a naopak. Po úvodnej identifikácii ciest nasleduje vytvorenie zoznamu áut, ktoré splňajú podmienky vstupu (prechodu). Pole áut je nasledujúcim spôsobom spracované až do jeho vyprázdenia. Autu s najväčšiu pozíciu, čiže nachádzajúce sa najbližšie ku koncu cesty, je vyhľadaní vystupujúci pruh. Ak takýto pruh nebude nájdený, je dané auto, a všetky autá zo spracovávaného pola, ktoré stoja za nim v jednom pruhu, vyhodené z pola spracovávaných áut. V opačnom prípade je auto vyhodené z jeho aktuálneho pruhu a vložené do určeného vystupujúceho pruhu na prislúchajúcu pozíciu (viz ďalej). Nasleduje už iba zápis auta do štatistik.

Križovatka : Prevedenie kroku simulácie na križovatkách sa delí na 3 hlavné vetvy. Za prvé sa zistuje, či je daná križovatka v stave, v ktorom má zmysel určovať prednosti. Ak nie, je každá vstupujúca cesta spracovaná podobným spôsobom ako pri pomocných uzloch.

V prípade, že je na križovatke nutné určovať prednosti, bude v editore na uzle umožnené nastaviť okrem defaultných aj vlastné nastavenia hlavných ciest a semaforov. Takéto križovatky sú pri kroku simulácie riešené zvyšnými dvoma spomínanými vettami, kde prvá spracováva svetelne riadené a druhá nesvetelne riadené križovatky.

Pri spracovaní križovatiek riadených svetlenou signalizáciou sa hned v úvode určí aktuálne bežiaca fáza semaforu. Do pomocného poľa sa zaznamenajú smery, ktoré majú "zelenú". Následne sa prechodom cez každý pruh vstupujúcich ciest vytvorí pole áut, ktoré majú pre svoju cestu a smer zelenú a splňajú podmienky vstupu. Pokračujeme cyklom, ktorý vyberá auto s najväčšou pozíciou, k nemu bude vybraný pruh, do ktorého bude prehodené. Ak nie je pruh určený, je dané auto a všetky autá nachádzajúce sa v pruhu za ním vyhodené zo zoznamu spracovaných áut. Cyklus končí vyprázdením poľa áut.

Spracovanie križovatiek s hlavnými cestami je najzložitejšia vetva kroku simulácie pri tomto type uzla. Hned v úvode je vytvorené pole zoznamov(ďalej iba ako pole_aut), kde každý prvok poľa, teda zoznam, predstavuje 1 pruh. Do každého takého prvkmu pole_aut budú uložené autá jedného pruhu, ktoré splňajú podmienky vstupu. Cyklus, ktorý následne spracováva pole_aut, je ukončený spracovaním, prípadne vyhodení posledného auta z celého pole_aut. Ešte pred prvou iteráciou cyklu je vytvorené ďalšie pomocné pole(ďalej iba ako moment_autu) o dĺžke maximálneho počtu pruhov križovatky, v ktorom budeme ukladať autá s najväčšou pozíciou z vrcholov každého pruhu, t.j. z aktuálne prvej pozícii každého zoznamu z pole_aut. Vložením do moment_autu je auto automaticky vyhodené z pole_aut. Na začiatku každej iterácie je pole moment_autu aktualizované. Základným zmyslom cyklu je pri prechode cez celé pole moment_autu vybrať auto s najvyššou prioritou(prednosťou). Priorita medzi dvoma autami je určená podľa toho, či sa porovnávané autá nachádzajú na rovnakej ceste. Ak áno, nachádzajú sa na rôznych pruhoch 1 cesty. Kedže smerovania pruhov 1 cesty sú bezkonfliktné, rozhoduje sa preto podľa pozície áut, pripadne druhotne ešte podľa doby čakania strávenej na danej ceste. V prípade áut nachádzajúcich sa na rôznych cestách, je určenie priority zložitejšie, preto je tomu venovaná samostatná sekcia(viz ďalej). Pri prechode poľom moment_autu sa zároveň zistuje, či pre dané auto a jeho smer existuje pruh s dostatočným voľným miestom na začiatku. Ak nie, je auto odstránené z poľa moment_autu a zároveň je vyprázdený zoznam v pole_aut, prislúchajúce jeho pruhu. Dôvod je zrejmý, pretože ak nemôže opustiť pruh auto, ktoré sa nachádza úplne vpredu, nemôžu opustiť pruh ani auta stojace v pruhu za ním. Po ukončení výberu auta, nasleduje "filter", ktorý vyhodí z poľa moment_autu a pole_aut všetky automobily, ktoré by mu teoreticky mohli skrížiť cestu. Ak je také auto nájdené, sú z polí odstránené aj všetky autá stojace

za ním v pruhu. Na konci iterácie je prípadne vybrané auto presunuté, do cesty kam smerovalo, a zapísané do štatistik.

Kruhový objazd : V uzle tohto typu sa v simulačnom cykle najprv spracúvajú automobily, ktoré sa už v kruhovom objazde nachádzajú. Kedže aj v skutočnom svete strávi automobil určitú dobu prejazdom kruhového objazdu, tak aj v kruhovom objazde simulácie musí auto stráviť určitý čas, až potom mu je umožnené opustiť ho. Samozrejme pred opusteniu objazdu mu musí byť nájdený vhodný vystupujúci pruh. Až po spracovaní automobil s dostatočným časom stráveným v kruhovom objazde vzhľadom na veľkosť objazdu, sú spracované automobily vstupujúce do okruhu. Aj v tomto prípade majú pri spracovaní vstupu automobilov do uzla prednosť autá, ktoré sa nachádzajú najvyššie na ceste, prípadne s najväčším čakacím časom.

Vloženie auta do pruhu : Aby sa simulácia čo najviac priblížila skutočnosti, snažil som sa vkladanie áut do pruhov spraviť čo najrealistickejšie. Auta vstupujúce do siete v jednom kroku simulácie z parkovísk a vstupov sú postupne vkladané na 0 pozíciu v pruhu, 0 mínus priemerná dĺžka auta atď.. Záporné hodnoty nám nevadia, pretože sa z nich autá dostanú po spracovaní pohybu áut, ktoré sa prevádzka až po spracovaní uzlov. Pri vkladaní áut sa berie ohľad nato, aby vkladané autá po spracovaní pohybu áut na cestách mali nezáporné pozície. Autá, ktoré prechádzajú z 1 na 2 cestu, sú vkladané do ciest so zápornými hodnotami o veľkosti vzdialenosťi, ktorá im chýba k dosiahnutia konca cesty, ktorú opúšťajú. Pohyb prevádzzaný po spracovaní uzlov teda pripomína plynulý pohyb áut v skutočnej premávke.

Určenie priority(prednosti) medzi 2 autami na križovatke : V úvode sa identifikujú porovnávané automobily(hl_auto, mm_auto), zistia sa poradové čísla ciest, na ktorých sa porovnávané autá nachádzajú a cest kam teoreticky smerujú. O prednosti rozhoduje rozhodovací strom, ktorý podľa podmienok vetví rozhodovanie na príkladové situácie. Hodnota, ktorú rozhodovaním dostaneme je, či má hl_auto prednosť pred mm_auto. Vetvy sú postupne delené a rozhodované podľa toho, či sa hl_auto a mm_auto nachádzajú na hlavných/vedľajších cestách, smerujú na hlavnú/vedľajšiu cestu a pravidlom pravej ruky. Ak sa o prednosti nerozhodne a nastane zhoda v prioritách, rozhoduje ich pozícia na ceste, prípadne druhotne este dĺžka čakania na príslušných cestách.

3.7.3 Tretí cyklus

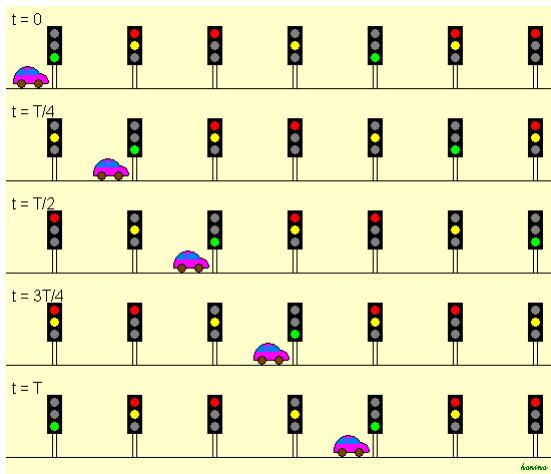
Posledný z trojice cyklov je skoro totožný s prvým z cyklov. Posun áut sa už ale neprevádzza na kópiach pruhov, ale na origináloch, ktoré sú už čiastočne upravené predchádzajúcim, druhým cyklom. Oproti kópiám pruhov, sú originálnym pruhom odobraté a pripojené autá. Cyklus prechádza a spracúva pruhy každej cesty simulácie. Každé auto (spracované od prvého v pruhu) je posunuté o maximálnu možnú dráhu, ktorú mu dovoľuje celková dĺžka cesty, pozícia auta pred ním a maximálna rýchlosť, ktorá je na danej ceste s danou prejazdnosťou možná.

3.8 Optimalizácia

K funkčnosti programu patrí aj obmedzená optimalizácia. Optimalizáciu môžeme rozdeliť do 3 hlavných oblastí. Vyhľadávanie obchádzok, zelená vlna a optimalizovanie križovatiek. Táto možnosť je umožnená užívateľovi až po prebehnutí celej simulácie. Optimalizačné výpočty sú robené z informácií nazbieraných počas simulovalia, preto sa dané optimalizačné návrhy vzťahujú k danej mape s danými nastaveniami uzlov. Každá z optimalizácií môže zmenou správania prvkov v sieti zasa ovplyvniť fungovanie iných prvkov, prípadne seba samých. Prvý spomenutý typ optimalizácie, vyhľadávanie obchádzok, sa vykonáva v rámci behu simulácie a informovanosti vodičov. Počas behu simulácie sú v pomocných štruktúrach pravidelne aktualizované informácie zaplnenosť a čakacích dôb automobilov. Tieto 2 vlastnosti ciest(pruhov) sú vzaté do úvahy pri prepočte prejazdností jednotlivých ciest(pruhov). Táto aktualizovaná prejazdnosť zasa ovplyvňuje ohodňovaciú funkciu v pravidelných výpočtoch najkratších ciest medzi uzlami, ktorých výsledky sa zapisujú do aktualizovaných riadiacich štruktúr. V konečnom dôsledku to pri výbere ciest pre automobily, ktorým bol nastavený príznak informovanosti, môže viest k zmene trasy, k zvoleniu najlepšej obchádzky.

3.8.1 Zelená vlna

Zelená vlna je pojem pre tak synchronizované fázy semaforov, aby vozidlo idúce doporučenou rýchlosťou zastihlo na všetkých semaforoch signál voľno. Zelená vlna sa používa najčastejšie na výpadovkách z veľkých miest.[9]



Obrázok 3.1: Zelená vlna

Pred výpočtami tohto typu optimalizácie je najprv užívateľom vybraný sled uzlov siete, ktorý musí spĺňať určité podmienky, aby mala vlna zmysel(viz 5.5.2). Následne sa postupným prechodom spracujú všetky uzly sledu. Na každých semaforoch sa upravujú pôvodné intervaly a prispôsobujú sa potrebám vlny. Fázy sa nastavujú spôsobom, aby celkové trvanie všetkých pravidelne sa striedajúcich fáz bolo na každom semafore rovnaké. Prechodom medzi uzlami sa zaznamenáva čas, ktorý by mal autu zabrátiť prechodu cestou medzi danými uzlami sledu.

Podľa neho sa v semaforoch nastavuje čas, kedy by mala teoreticky na semafore naskočiť prvá fáza. Táto synchronizácia by mala zabezpečiť, aby tesne pre príchodom automobilu na nasledujúce svetlá, bola prepnutá fáza umožňujúca mu prechod. Rovnaké celkové trvanie fáz zabezpečuje, aby táto situácia nastávala pri každom prechode striedajúcich sa fáz.

3.8.2 Vylepšenie križovatiek

Počas riešenia tejto problematiky a testovania rôznych riešení som dospel k záveru, že téma si zaslúží minimálne samostatnú prácu. Faktorov ovplyvňujúcich chod križovatky, prípadne úspechu jej optimalizačného návrhu je veľké množstvo. Čakacie doby automobilov na semaforoch-vedľajších cestách, množstvá áut prechádzajúcich križovatku v určitom smere, striedanie intenzity prichádzajúcich automobilov, jej rozdelenie do fáz dňa, počty a smerovania pruhov cest vstupujúcich do križovatky, dĺžka vstupujúcich-vystupujúcich cest, typ predchádzajúcich uzlov... Každá križovatka si zaslúží osobitý prístup a riešenie.

Pozorovanie : pri hľadaní vylepšovacích riešení je stále treba brať ohľad na viaceré kritérii. Každému z týchto kritérií je treba podľa situácie dávať rôznu prioritu. Mnou navrhnuté riešenia neboli univerzálné a skutočne optimalizačne riešili iba malé počty križovatiek. Na riešenie tejto témy by bolo vhodné použiť viackriteriálne rozhodovanie, čo je netriviálna úloha. Rozho-

dovaním pomocou metód viackriteriálnej optimalizácie sa zaoberá napríklad jeden z magisterských informatických oborov na MFF UK, Diskrétny modely a algoritmy.

Môj názor ku ktorému som došiel pri implementácii a testovaní tejto funkcie je, že najlepšie riešenie v rámci možností tohto programu, pre optimalizáciu križovatiek je vytvorenie modelu s križovatkou a prvkami, ktoré by mohli správanie automobilov ovplyvniť. Na takomto modeli zameranom na simulovanie vývoja premávky hlavne na danej križovatke, otestovať viaceré zmysluplné návrhy riešenia a za konečné riešenie zvoliť to s najpozitívnejšími výsledkami.

Kedže simulácia nemá možnosť sama otestovať rôzne varianty riešenia vyššie opísaným spôsobom, zvolil som hlavným kritériom optimalizácie(v rámci možností) počty automobilov prechádzajúce križovatkou v konkrétnych smeroch počas jednotlivých intervalov. Štatistické počty prechádzajúcich automobilov v jednotlivých smeroch sú usporiadane podľa veľkosti samostatne pre každý interval. Súčty poradí jednotlivých smerov v intervaloch budú pri optimalizácii smerodajné. Udávajú nám nápovedu, k určeniu hlavných, prípadne k nastaveniu trvania určitých fáz semaforov pre určité smery. Smerom, ktoré sa umiestnili na vrchole zoznamu bude venovaná najväčšia pozornosť(trvanie), priorita. V prípade svetelné riadených križovatiek sa vytvorí nové nastavenie semafora, v ktorom budú nastavené fázy pre každý možný smer. Podľa vypočítaných poradí budú fázam priradené trvania, od najdlhšej po najkratšiu. Na nesvetelných križovatkách budú hlavnými cestami zvolené tie, ktorými do križovatky vstupoval a vystupoval najvytaženejší smer.

Kapitola 4

Implementácia

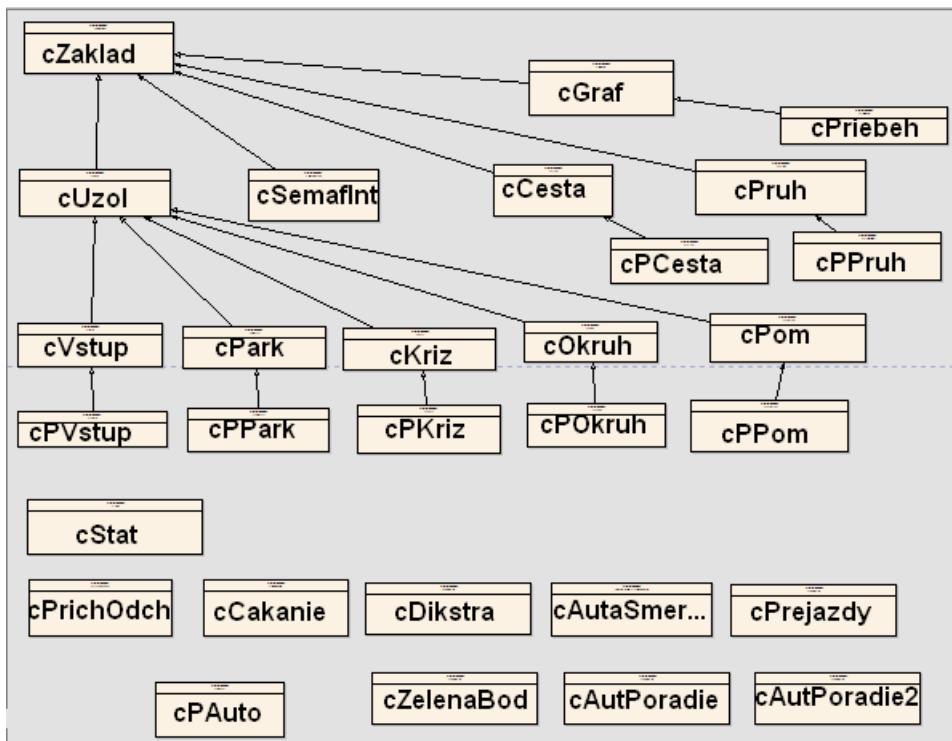
4.1 Výber vývojových nástrojov

Pri výbere programovacieho jazyka na vývoj programu som sa rozhodoval medzi jazykmi C++ a C#. Skúsenosti s programovaním v C# som nemal, preto som uprednostnil jazyk C++. Moja voľba jazyka, ako som neskôr zistil, nebola tá najlepšia. Jazyk C# je čisto riadený(managed), t.j. hlavným rozdielom je, že o správu pamäte sa nestaráte Vy, ale tzv. garbage collector. Kdežto v C++, je možná kombinácia čistého C++ a Visual C++(tiež známejho aj pod menami "Managed Extensions for C++", "C++/CLI"), kde Visual C++ je podobne ako C# riadený. Visual C++ je množina rozšírený pre C++ od firmy Microsoft, zahrňujúca gramatické a syntaktické rozšírenia, kľúčové slová a atributy, ktoré majú zaviesť syntax a jazyk jazyka C++ do .Net Framework.u[6]. Využitie kombinácie čistého a riadeného C++ sa spočiatku zdala výhodou, ale v neskorších fázach začala spôsobovať problémy, ktoré vyústili do časovo náročného prerábania programu do čisto riadeného C++. Veľkou výhodou použitia .Net Frameworku je funkcionality, ktorú poskytujú knižnice základných tried .Net. Okrem využitia .Net tried na prácu so zoznamami a súbormi(obdoba STL), bola nenahraditeľná knižnica Windows.Forms, ktorá obsahuje množstvo komponent na tvorbu užívateľsky prívetivého prostredia. Keďže nebola stanovená podmienka prenositeľnosti na iné platformy, je program spustiteľný pod rodinou operačných systémov MS Windows (2000 a vyššie). Firma Microsoft už vypracovala verzie .Net Framework.u, ktoré fungujú aj na niektorých distribúciách Unixu a Mac OX, takže na spomínaných systémoch s nainštalovaným príslušným .Net Frameworkom, by mal byť(mnou prakticky neodskúšané) program tiež

spustiteľný. Pre prácu na programe som si zvolil vývojové prostredie Microsoft Visual Studio 2005, ktoré poskytuje príjemné prostredie na pre vývoj aplikácií.

4.2 Objektový návrh programu

Objektovo orientované programovanie (ďalej iba OOP) je programovacia forma, vzor, prístup používajúci objekty a ich interakciu k návrhu aplikácií a počítačových programov. Táto programovacia technika obsahuje prvky ako dedičnosť, polymorfizmus a zapuzdrenosť. Od polovice 90tych rokov sa OOP stalo dominantnou programovacou metódou.[8] Jednou z vecí, ktorá by mala byť čo najviac premyslená a malo by jej byť venované čo najviac času a pozornosti, je návrh objektového modelu programu. Dobre navrhnutý model značne zjednoduší a pomôže pri implementácii programu a prípadných zmenách (typu dorábania funkčnosti).



Obrázok 4.1: Objektový model programu

4.2.1 Popis a funkčnosť objektov

cZaklad : je trieda, ktorá je predkom väčšiny tried programu. Obsahuje metódy a členy, ktoré využívajú a k fungovaniu potrebujú jeho potomkovia. Ide o členy, ktoré sú potrebné na vykreslovanie, na nastavenie a spracovanie času, na generovanie náhodných čísel... Z metód to sú napríklad vykresľovacia a metóda prevedenia kroku simulácie na danom uzle.

Triedy dediace od triedy cZaklad a určené pre potreby editora

cPruh : trieda, ktorá obsahuje položky, ktoré si zapamätávajú cesty do stupne z daného pruhu.

cCesta : obsahuje členy na zapamätanie si vlastností cesty ako dĺžka, počet pruhov, prejazdnosť, maximálnu povolenú rýchlosť. Referencie na obsahujúce pruhy, cestu opačného smeru a na uzol, do ktorého cesta vstupuje. K zaujímavým metódam, ktoré obsahuje väčšina tried predstavujúcich prvky v sieti, patria metódy na kopírovanie údajov, vykreslenie, zápis a načítanie zo súboru. Metóda na kopírovanie údajov slúži na vytvorenie kópie siete pre každou zmenou. Využitie tejto vlastnosti je popísané v užívateľskej dokumentácii.

cUzol : je potomkom triedy cZaklad, do ktorej boli vložené metódy a prvky spoločné pre všetky uzly. Patria k nim súradnice uzlu, zoznamy vstupujúcich a vystupujúcich ciest a metódy na prácu s nimi.

cVsup : v definícii tejto triedy pribudli zoznamy, do ktorých sa zaznamenávajú nastavenia vstupov ohľadom celkového počtu vstupujúcich/vystupujúcich áut a ich rozdelení do časových intervalov simulácie.

cPark : je veľmi podobná triede cVstup. Lísi sa iba v informáciach, ktoré sa do zoznamov ukladajú a vo vlastnosti kapacity parkoviska.

cSemaflInt : je trieda, v ktorej sa ukladajú kompletné informácie ohľadom semaforových fáz, ich trvania a smerov v daných fázach povolených.

cKriz : trieda obsahujúca referencie na hlavné cesty/semafor v prípade nesvetelne/svetelne riadenej križovatky. Najviac pravidel premávky sú pre riadenie križovatiek, preto má táto trieda funkcie nastavujúce defaultné nastavenia pre hlavné cesty/semafory a smerovanie pruhov cest vstupujúcich do križovatky, ktoré sú volané v prípade zmien na križovatke.

cOkruh, cPom : sa od seba a triedy cUzol podstatne nelíšia. Markantná zmena je iba v novom člene triedy cOkruh zaznamenávajúci veľkosť okruhu.

cGraf : je trieda, ktorá predstavuje a nesie informácie o celej dopravnej sieti v čase editovania a tvorby mapy. Obsahuje zoznamy všetkých uzlov, polí a časových informácií.

Triedy dediace od triedy cZaklad a určené pre potreby behu simulácie. Tieto triedy sú potomkami tried s názvom, ktorý si odvodíte vynechaním prvého písma P v ich nazve. Neplatí to iba o triede cPriebeh, ktorá je potomkom triedy cGraf. K spoločným vlastnostiam patrí nadefinovaná funkcia, prevádzajúca krok simulácie na danom prvku simulácie, funkcie na zaznamenávanie štatistik.

cPPruh : je potomok triedy cPruh, ktorému boli pridané členy slúžiace na ukladanie štatistik, zoznamu áut nachádzajúcich sa danom pruhu, farby pruhu, ktorou bol daný pruh pri poslednom kroku simulácie vykreslený. Na viacpruhovej ceste, na ktorej do určitého smeru odbočuje iba 1 pruh môže byť hustota premávky značne rozlišná, ako na ostatných pruhoch. Preto je v tejto triede pridaný atribút na zaznamenanie aktuálnej prejazdnosti pred daným pruhom, ktorý je využívaný pri aktualizačných výpočtoch.

cPCesta : je od svojho predka rozšírená o štatistické atribúty a určitú funkčnosť. Funkčnosť prevádzka krok simulácie na danom prvku simulácie, vyberá vhodný pruh pre auto vstupujúce na cestu atď..

cPVstup : je ako väčšina tried v tomto odseku rozšírená od svojich predchodcov o štatistiky, pomocné štruktúry ukladajúce informácie o najkratších cestách a autách smerujúcich do uzla. K zaujímavým funkciám patrí funkcia rozhodujúca o vstupe automobilu daným uzlom do siete.

cPark : je potomkom triedy cPark, rozšírené o podobné údaje ako cPVsup. Štatistiky sú oproti triede cPVsup obsiahlejšie. Taktiež tu je nadefinovaná funkcia rozhodujúca o vstupe automobilu z uzla typu parkovisko do siete.

cKriz : okrem rozšírení o štatistiky, štruktúry riadiace smerovanie áut prechádzajúcich križovatkou, sa v triede nachádzajú funkcie používané na zistenie predností medzi 2 autami, ich kríženia atď..

cPOkruh : - je od svojho predchodcu rozšírený okrem iných o zoznam áut, ktoré sa v danom kroku simulácie nachádzajú v objazde a o funkciu, ktorá určuje, či dané auto zotrvalo v kruhovom objazde dostatočne dlho na prechod ním, vzhľadom na jeho veľkosť.

cPPom : nie je od svojho predka ničím podstatným, čom sme pri ostatných triedach ešte nespomenuli, rozšírený.

cPPriebh : je trieda nesúca počas vlastnej simulácie celú cestnú sieť, kde prostriedky pre uchovanie prvkov siete zdedia od svojho predka, triedy cGraf. Preto sa rozšírenia dotkli hlavne funkcionality. Boli pridané funkcie na spustenie vlastnej simulácie, na spustenie výpočtov pre pomocné štruktúry, funkcie ktorá z načítaného grafu vytvorí priebeh, funkcia, ktorá podľa nastavení uzlov v sieti vyhľadáva ciele pre automobily atď..

Pomocné triedy

cAuto : je trieda predstavujúca automobil v pohybujúci sa po sieti. Informácie, ktoré sú v atribútoch triedy uchovávané predstavujú referencie na uzly, ktorým automobil vstúpil do siete a ktorý mu bol priradený ako ciel. Referencie na prípadné automobily stojace v pruhu pred nim a za ním, pozíciu v pruhu, údaj o čakacej dobe, ktorú strávil na danom pruhu, okruhu, príznak informovanosti a ID automobilu.

cStat : je pomocná trieda určená pre potreby štatistik. Užahčuje zápis a výber podľa parametrov. Je využívaná v nasledujúcich 5 triedach.

cPrichOdch : je triedou, ktorá je funkčne prispôsobená na ukladanie štatistik príchodu/odchodu áut z/do uzlov podľa uzlov, ktorými auto do siete vstupovalo/kam smerovalo. Informácie sú ukladané podľa časových intervalov, počas ktorých daný jav nastal. Kedže rovnaké štrukturálne požiadavky majú aj štatistiky príchodu a odchodu áut podľa ciest, má táto trieda dvoje využitie.

cPrejazdy : je trieda, v ktorej sa ukladajú počty ohľadom prechodu áut križovatkami. Tieto informácie sú neskôr využívané pri optimalizácii križovatiek. Dávajú nám poznatok o vyťaženiach jednotlivých smerov.

cCakanie : je štatistická trieda, pre ukladanie informácií o čakacích dobách automobilov počas prechodu určitými cestami.

cDikstra : je trieda na ukladanie smerovacích informácií, ktorými sa riadia automobily na križovatkách. Obsahuje 3 sady týchto informácií, všeobecné, aktualizované a alternatívne.

cAutaSmerujuceDoUzla : je trieda určená na počítanie automobilov smerujúcich do uzlov s predpokladanými príchodmi počas určitých časových intervalov. Tieto informácie sú využívanie hlavne pri určovaní cieľov pre automobily vstupujúce do simulácie.

cZelenaBod, cAutPoradie, cAutPoradie2 : tieto 3 triedy sú používané pri optimalizačných výpočoch. cZelenaBod pri tvorbe zelenej vlny, cAutPoradie a cAutoPoradie na výpočty vyťaženosťi jednotlivých smerov pri optimalizácii križovatiek.

Kapitola 5

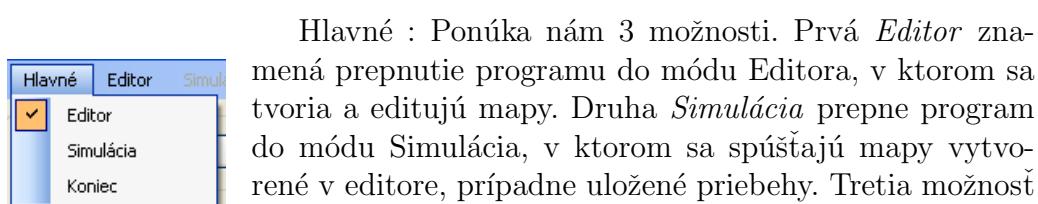
Užívateľská dokumentácia

5.1 Inštalácia

Program je spustiteľný pod rodinou operačných systémov Windows verzie 2000 a vyššie. Inštaláciu programu zahájite spustením inštalačného balíčka s názvom *Simulácia dopravnej siete.msi*. Počas inštalácie budete mať možnosť zmeniť inštalačný adresár programu. V prípade neprítomnosti prerekvizity .Net Framework.u 2.0 alebo vyššie, Vás inštalátor upozorní a ponúkne Vám možnosť stiahnutia a inštalácie tohto programu zo stránok Microsoftu. V prípade nedostupného internetového pripojenia je možné nainštalovať tento program z inštalačného CD. Odporuča sa inštalácia .Net Framework.u stiahnutého zo stránok firmy Microsoft, kde nájdete najaktuálnejšiu verziu tohto programu. Súčasťou inštalačného balíčka je aj niekoľko predpripárených máp.

5.2 Hlavné menu

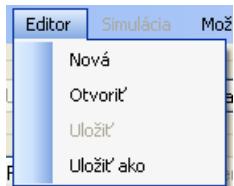
V hlavnom menu na hornej lište programu sa nachádzajú 4 položky. V tejto časti si postupne vysvetlíme možnosti týchto položiek.



Hlavné : Ponúka nám 3 možnosti. Prvá *Editor* znamená prepnutie programu do módu Editora, v ktorom sa tvoria a editujú mapy. Druha *Simulácia* prepne program do módu Simulácia, v ktorom sa spúšťajú mapy vytvorené v editore, prípadne uložené priebehy. Tretia možnosť

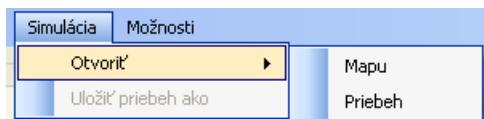
Obrázok 5.1:
Hlavné menu

Koniec skončí beh program.



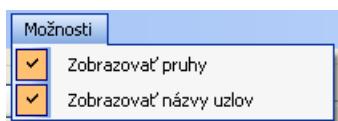
Obrázok 5.2:
Menu editor

Editor : Táto položka hlavného menu je prístupná iba v móde Editor. Ponúka 4 možnosti na výber. Prvá možnosť *Nová* zatvorí prípadne otvorenú editovanú mapu a vytvorí novú, prázdnú. Druhá možnosť *Otvoriť* umožní užívateľovi otvoriť mapu uloženú v súbore. Treťia možnosť *Uložiť* uloží mapu otvorenú v editore do súboru, z ktorého bola mapa načítaná, prípadne naposledy uložená. Posledná štvrtá možnosť *Uložiť ako*, ponúkne uloženie mapy do nového alebo užívateľom vybraného súboru.



Obrázok 5.3: Menu simulácia

Simulácia : Táto položka je prístupná iba v móde Simulácia. Prvé 2 možnosti ponúkajú otvorenia Mapy alebo Priebehu z uložených súborov. Tretia možnosť *Uložiť priebeh ako* umožňuje uložiť od pauzovanú simuláciu.



Obrázok 5.4: Menu možnosti

Možnosti : Prvá z možností tejto položky hlavného menu, *Zobrazovať pruhy* je prístupná iba v móde Editor. V móde Simulácia je automaticky zapnutá, nie je možné jej vypnutie. Názov druhej možnosti *Zobrazovať názvy uzlov* hovorí sám za seba.

5.3 Vykreslovanie

Každý typ uzla je vykreslovaný vzhľadovo rozdielne, aby nedochádzalo k ich zámene. Nad uzlami typu vstup a parkovisko sú pre lepšiu identifikáciu vypisované ich názvy. Editovaný uzol je obklopený *podsvietený* zelenou farbou, aby bolo jasné, ktorý uzol práve editujeme. Na cestách sú v ich strede nakreslené šípky, ktoré ukazujú smer pohybu premávky na nej. Počas simulácie sa farba vykreslovania ciest, resp. pruhov mení s veľkosťou zaplnenia daného pruhu. To iste platí aj pre uzly typu parkovisko, ktorých farba indikuje ich momentálne zaplnenie. Počas editácie sa v mape môžu zobrazovať písmaná A,B a číslice 1,2,3,4, ktoré slúžia pre lepšiu orientáciu na mape

pri danej činnosti. Program ponúka možnosť zväčšenia a zmenšenia mapy pre lepší prehľad. Pri mapách, ktoré presahujú vykreslovaciu plochu poslúžia posúvatká.

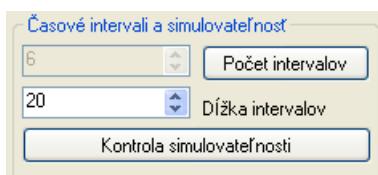
5.4 Mód Editor

Tento mód programu, ktorý sa zapína v hlavom menu voľbou Editor, slúži na vytváranie a upravovanie máp dopravných sietí. Počas editácie je v ľavom ovládacom panely neustále prístupná skupina tlačidiel.



Obrázok 5.5: Editor hlavné

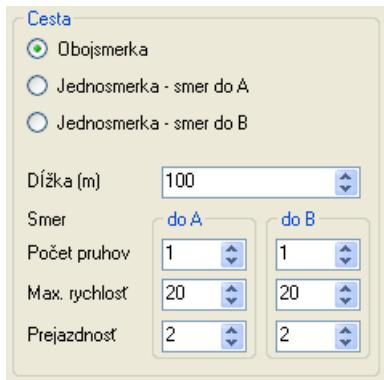
me do mapy do miesta, kde chceme uzol umiestniť. V prípade pridávania cesty je potrebné kliknúť na 2 uzly na mape, medzi ktoré chceme pridávanú cestu umiestniť a medzi ktoré je ešte možné umiestniť cestu. Voľbou tlačidla *Odober* odoberieme uzol/cestu, určený kliknutím do mapy. Tlačidlami *OK* a *Zruš* potvrdzujeme, prerušujeme alebo rušíme práve vykonanú zmenu.



Obrázok 5.6: Editor čas

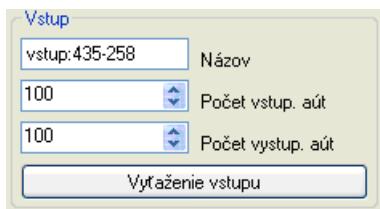
Vrchnými tlačidlami *Uzol* a *Cesta* prepíname obor pôsobenia spodných 5 tlačidiel. V prípade znemožneného tlačidla *Uzol* a používaním tlačidiel *Pridaj*, *Odober* a *Edituj* prevádzame rovnomenne operácie na uzloch. V prípade znemožneného tlačidla *Cesta* sú spomínané operácie prevádzcané na cestách. Stlačením tlačidla *Pridaj* pridávame uzol, cestu. Pred pridaním uzla nám vykročí dialógové okno, v ktorom si zvolíme typ pridávaného uzla(viz ďalej), následne klikneme do mapy do miesta, kde chceme uzol umiestniť. V prípade pridávania cesty je potrebné kliknúť na 2 uzly na mape, medzi ktoré chceme pridávanú cestu umiestniť a medzi ktoré je ešte možné umiestniť cestu. Voľbou tlačidla *Odober* odoberieme uzol/cestu, určený kliknutím do mapy. Tlačidlami *OK* a *Zruš* potvrdzujeme, prerušujeme alebo rušíme práve vykonanú zmenu.

Zvolením tlačidla *Počet intervalov* a odoberením upozornenia bude umožnená zmena počtu intervalov simulácie. Trvanie intervalov nastavíme v políčku naľavo od textu *Dĺžka intervalov*. Stlačením tlačidla *Kontrola simulovateľnosti* bude na mape prevedená kontrola simulovateľnosti, ktorá nás v prípade existencie nepoužiteľného uzla naň upozorní.



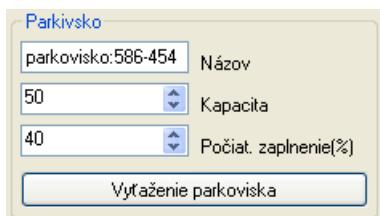
Obrázok 5.7: Editor cesty

Pri pridávaní a úprave ciest budú v mape veľkými písmenami A,B označené počiatočný a konečný uzol cesty. Tieto označenia slúžia pre orientáciu v nastavovaní vlastností cesty. V najvrchnejších 3 riadkoch určíme typ cesty volbou jednej z možností *Obojsmerka*, *Jednosmerka - smer do A/B*. Zvyšné nastavenia cesty sú dostatočne napovedajúce.



Obrázok 5.8: Editor vstup

V nastaveniach uzla typu Vstup je možne okrem nastavenia celkového počtu áut vstúpených/vystúpených do/zo simulácie daným uzlom, rozdeliť tieto počty na každý časový interval. Stlačením tlačidla *Vyťaženie vstupu* sa nám ponúkne nové okno, v ktorom bude možné tieto počty intuitívne poupraviť. K nastaveniam patrí aj názov uzlu.



Obrázok 5.9: Editor parkovisko

Nastavenia parkoviska umožňujú nastaviť názov, kapacitu a počiatočné zaplnenie parkoviska. Stlačením tlačidla *Vyťaženie parkoviska* nám bude umožnené v novootvorenom okne, intuitívne nastaviť priemerné zaplnenie a počet áut navštievujúcich parkovisko s presnosťou na časový interval.

Nastavenia križovatky nám ponúkajú možnosť určenia typu križovatky. Podľa toho je v prípadoch, keď to na danej križovatke má zmysel, umožnené nastavenie hlavných ciest prípadne semaforov. Po zvolení tlačidla *Nastavanie hlavných ciest* sa mapa zredukuje na práve editovanú križovatku a cesty do nej vstupujúce/vystupujúce. Kliknutím na 2 rôzne cesty, ich následným

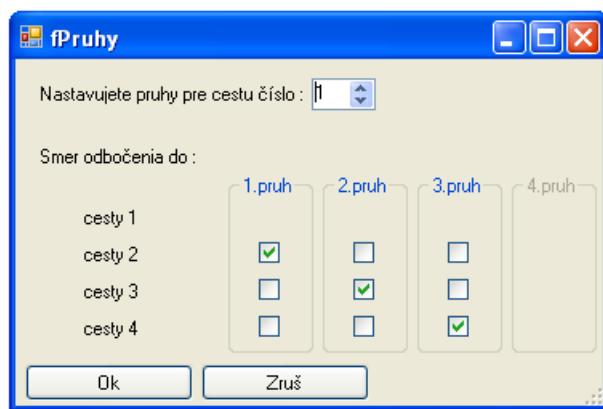
zhrubnutím určíte hlavné cesty. V prípade stlačenia tlačidla *Nastavenie semaforov* sa objavia na hlavnej mape pomocné označenia, číslice 1-4 a v popredí vyskočí dialógové okno.



Obrázok 5.10: Editor križovatka a semafor

Kolonka vedľa textu *Interval číslo* určuje poradové číslo práve upravanej fáze semaforu. Pomocou voľby fáze intervalu a zaškrťávacieho okienka hned vedľa máme možnosť zväčšíť/zmenšiť počet fáz semaforu. V okienku napravo od textu *čas začatia 1. intervalu* sa určuje, po kolkých sekundách od začatia simulácie by mala prvý krát naskočiť prvá fáza semaforu. Okienko pod nadpisom *trvanie intervalu* je dostatočne intuitívne. Zaškrtnutím polička na súradničach x,y v matici nastavíte v upravovanej fáze *zelenú* pre smer z cesty y na cestu x. Ak bude po nejakom zaškrtnutí znemožnené prepínanie medzi fázami, znamená to, že nejaké zo zaškrtnutých smerov sa križujú, čo je v programe zakázané. Odškrtnutím nejakého smeru môže byť zasa znemožnené tlačidlo OK, čo naznačuje, že nejaký z možných smerov križovatky nie je zaškrtnutý v ani jednej fáze, čo taktiež nie je povolené. Ďalšia možnosť nastavenia križovatiek je smerovanie pruhov. Tlačidlo *Nastaviť pruhy*, ktorého stlačením nám vyskočí na popredie okno nastavujúce smerovanie pruhov, je umožnené iba v prípade, že by smerovanie pruhov na danej križovatke malo zmysel.

K lepšej orientácii na mape sa nám počas nastavovania smerovania pruhov na mape objavia číslice určujúce poradové čísla ciest. Kolonka nachádzajúca sa najvyššie v okne určuje poradové číslo cesty, ktorej pruhy momentálne nastavujeme. Zaškrtnutím polička na pozícii x,y v matici , povolíme



Obrázok 5.11: Editor pruhy

Editáciou kruhového objazdu máme možnosť zmeniť maximálny počet príjazdov daného objazdu. Pomocný uzol nemá žiadne dostupné nastavenia.

5.5 Mód Simulácia



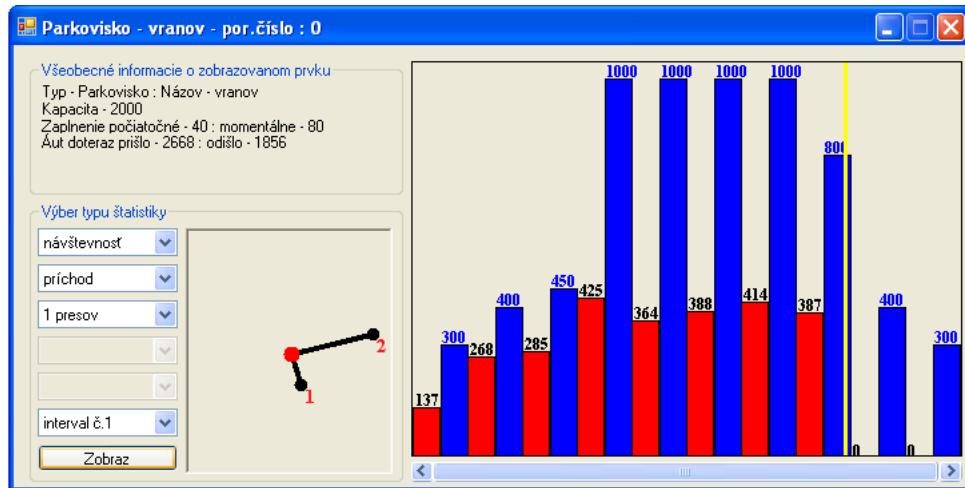
Obrázok 5.12: Simulácia

smerovanie z pruhu číslo y upravovanej cesty do cesty číslo x. Prípad, keď je znemožnené prepínania poradového čísla upravovanej cesty, znamená, že sa nám pruhy upravovanej cesty prekrižujú. Prípade zne- možnenia tlačidla OK, naznačuje, že nie každé možné smerovanie na danej križovatke je umožnené.

Hlavné 3 ovládacie prvky simulácie sú tlačidla *Štart*, *Pauza*, *Stop* a sú umožnené až otvorením nejakej mapy alebo priebehu. Prvé spomínané tlačidlo spúšta chod simulácie. Počas behu simulácie a jej uplynutie je znemožnené. Umožnené je použitím tretieho tlačidla *Stop*. Jeho stlačenie spôsobí vymazanie doterajšieho behu simulácie. Kvôli lepšej viditeľnosti aktívnej pauzy, je tlačidlo *Pauza* počas aktívnosti červeno podsvietené. K ovládacím prvkom simulácie patria aj mierky *Rýchlosť* a *Informovanosť*. *Rýchlosť* zrýchľuje chod simulácie a *Informovanosť* určuje mieru informovanosti vodičov, oboba používaných navigačných zariadení. V časti *Časové údaje* sú prehľadne podávané informácie o čase simulácie.

5.5.1 Štatistiky a optimalizácia

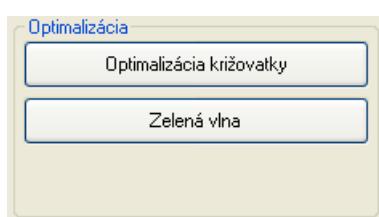
Pozastavením myši nad prvkom simulácie v mape sa zobrazí malé pop-up okno s čiastočnými informáciami o uzle alebo ceste. Kliknutím myšou do mapy na nejaký prvok simulácie bude otvorené nové okno.



Obrázok 5.13: Štatistiky

V ľavom hornom rohu okna sa nachádza oblasť, kde sú uvedené všeobecné informácie o prvku, ktorého štatistiky ideme zobrazovať. Pod touto oblasťou sa kvôli rýchlejšej orientácii nachádza malý nákres prvku a stĺpec s ponúkanými možnosťami zobrazovania štatistik. Po špecifikácii štatistiky a stlačený tlačidla zobraz sa nám v pravej časti okna zobrazí nami nastavená štatistika. V štatistikách znamenajú červené stĺpce skutočné a modré predpokladané hodnoty. Žltá zvislá čiara ukazuje časový priebeh v simulácii.

5.5.2 Optimalizácia



Po ukončení celého priebehu simulácie bude umožnené použitie optimalizačných funkcií programu. Po výbere typu optimalizácie medzi Zelenou vlnou a Optimalizáciou križovatky, je potreba vybrať prvok, skupinu prvkov na danú činnosť.

Obrázok 5.14: Simulácia

Zelená vlna : pre túto možnosť je potrebné vybratie sledu uzlov simulácie spĺňajúcich nasledujúce podmienky.

- Uzly v sledo musia byť postupne spojené použiteľnou cestou.
- Začiatočný a konečný uzol sledu môže byť každého druhu.
- Druhým a predposledným uzlom musia byť pojazdné svetelné križovatky.
- Medzi druhým a predposledným uzlom sa môžu nachádzať uzly typu pomocný a križovatka. V prípade pojazdnej nesvetelnej križovatky musia byť vstupujúca a vystupujúca cesta, medzi danou križovatkou a uzlami sledu, hlavnou.

Po výbere takéhoto sledu Vám bude umožnené previesť úpravu. Pre lepšiu orientáciu pri tvorbe sledu sú uzly a cesty medzi nimi zvýraznené na zeleno.

Optimalizuj križovatku - pre potreby tejto funkcie bude potrebné vybrať pojazdnú križovatku, na ktorej chcete vylepšenie previesť. Následne bude možné previesť danú úpravu.

Po prevedení úpravy Vám bude ponúknutá možnosť uloženia pozmenenej siete ako mapy. Následne máte možnosť hned vyskúšať pozmenenú sieť.

5.6 Ukladanie-načítanie máp-priebehov

Za účelom uloženia máp alebo už bežiacich priebehov umožňuje program ukladanie a opäťovné načítanie. Mapy sú ukladané do súborov s príponou .sds, priebehy do súborov s príponami .sdsp. Mapy sú načítateľné v editore, za účelom editácie a v móde simulácie, pre účely simulácie. Priebehy sú načítateľné iba v móde simulácie.

Kapitola 6

Záver

6.1 Praktické skúsenosti a poučenia

Programovanie programu Simulácia dopravnej siete bola pre mňa prvá skúsenosť s väčším projektom. Pri jeho vývoji a programovaní som si uvedomil dôležitosť činností prevádzaných pred vlastným programovaním. Zistil som, že rozumne a správne navrhnutie funkčnosti, objektového modelu a návrhu programu je dôležitejšie ako som si doteraz myslal. Hovorí sa, že človek sa učí na vlastných chybách. To som si vyskúšal aj ja, pri niekoľkých chybných rozhodnutiach v návrhu, ktoré ma stály značný čas na opravu. Bola to cenná skúsenosť do ďalsieho programátorského života.

Popri programátorských skúseností som sa dozvedel a naučil pracovať s platformou .Net, ktorá predpokladám bude mať čoraz väčšie využitie.

6.1.1 Odchýlky od špecifikácie

Hlavným neúspechom hodnotím optimalizáciu nesvetelných križovatiek, ktorej riešenie sa ukázalo ako komplikovanie a náročné. Rozšírenie programu o solídnú optimalizáciu bolo pri "malosti" projektu neaplikovateľné.

V implementácii chýba funkcionalita, ktorá by ponúkala možnosť sledovať paralelne 2 alebo viac simulácií naraz z dôvodu sledovania vplyvov zmien na premávku. Neprítomnosť tejto funkcionality nepriamo nahradza možnosť spustenia 2 inštancií programu naraz a porovnávanie podrobných zobrazených štatistiky. Tie majú z druhej strany výhodu presných počtov a možnosti porovnávania štatistických údajov o celom behu simulácie aj po prebehnutí celej simulácie.

Riešenie programu sa lísi od predom navrhovaného riešenia aj nezavedením tzv. *Aprogramov* pre automobily. Každému automobilu vstupujúcemu do simulácie mal byť vygenerovaný program, ktorým by sa riadil počas celého behu simulácie. Vytváranie takýchto programov sa stalo po pridaní nastaviteľných parametrov, riadiacich celý priebeh simulácie, netriviálnou a ťažko aplikovateľnou úlohou. *Aprogrami* boli v programe nahradené nastaviteľnými vlastnosťami uzlov, ktoré považujem za viac ako dostatočnú náhradu. Z mojej strany si dokonca myslím, že nastaviteľné vlastnosti uzlov značne zväčsili funkcionality programu. Pretože z programu simulujúceho úplne náhodne sa správajúcu premávkou sa stal program umožňujúci nasimulovanie simulácií s predpokladanými scenármi, intenzitami vyťaženia uzlov a počtami automobilov. Táto vlastnosť rozšírila možnosti využitia programu.

6.2 Ďalší vývoj

Program by pri ďalšom vývoji mohol byť rozšírený hlavne zo strany optimálizácie. Je to rozsiahla a netriviálna téma, ktorá by pridala na funkčnosti a využitiu.

Väčšie priblíženie sa realite by sa v simulácií mohlo dosiahnuť väčšou autonómnotou správania sa vodičov, pridaním určitej inteligencie v rozhodovaní na križovatkách.

Rozšíriť program by bolo možné aj o iný druh dopravy ako napr. električky, pravidelné spoje typu MHD, nezávislých na vlastnostiach vyťaženosťí uzlov.

Literatúra

- [1] Dostupne z www : http://www.kod.tul.cz/info_predmety/Psi/prednasky_2007/prednaska_1_PSI.pdf
- [2] Dostupne z www : http://cs.wikipedia.org/wiki/Počítačová_simulace
- [3] Dostupne z www : http://kam.mff.cuni.cz/kam/kuva/vyuka/programovani/xaver/diskretni_simulace.html
- [4] Troelsen A.: *C# a .Net 2.0 profesionálne*, Zoner Press, Brno 2006, ISBN 80-86815-42-0, str. 47
- [5] Dostupne z www : <http://www.microsoft.com/downloads/details.aspx?displaylang=cs&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5>
- [6] Dostupne z www : http://en.wikipedia.org/wiki/Managed_Extensions_for_C++
- [7] Matoušek, J., Nešetřil, J.: *Kapitoly z diskrétní matematiky*, Nakladatelství Karolinum, Praha 2000, ISBN 80-246-0084-6, str. 102-106
- [8] Dostupne z www : http://en.wikipedia.org/wiki/Object_oriented
- [9] Dostupne z www : http://cs.wikipedia.org/wiki/Zelená_vlna