

OBSAHUJE CD

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Matej Dajčár

Simulácia cestnej siete

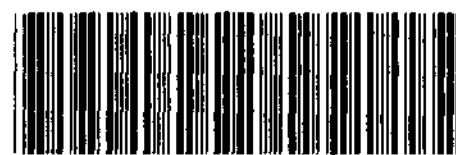
Ústav formálnej a aplikovanej lingvistiky

Vedoucí bakalářské práce: Mgr. Petr Homola

Studijní program: Informatika, Správa počítačových systémů a sítí

2007

KNIHOVNA MFF UK



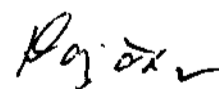
2560034769

Chcel by som poďakovať vedúcemu tejto bakalárskej práce pánovi Mgr. Petru Homolovi za jeho podnety, pripomienky a pomoc pri vypracovaní tejto práce.

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

V Praze dne 5.8.2007

Matej Dajčár



Obsah

1	Uvod	6
1.1	Motivácia	6
1.2	Cieľ	7
2	Návrh	8
2.1	Výber témy práce	8
2.2	Druhy simulácií	9
2.2.1	Diskrétna simulácia	9
2.2.2	Spojité simulácia	9
2.3	Detaily špecifikácie a návrhu programu	10
2.3.1	Názov programu	10
2.3.2	Požiadavky na funkcie programu	10
2.3.3	Vzhľad programu	11
2.3.4	Platforma	11
3	Popis funkčnosti programu	12
3.1	Výber typu simulácie	12
3.2	Určenie trás v cestnej sieti	12
3.3	Výber cieľov jednotlivých áut	14
3.4	Chod programu	15
4	Implementácia	17
4.1	Výber jazyka a vývojového prostredia	17
4.2	Návrh objektového modelu programu	18
4.3	Popis jednotlivých typov objektov a ich funkcií	19
4.4	Organizácia objektov v programe	24

5	Užívateľská dokumentácia	25
5.1	5.1 Inštalácia programu	25
5.2	Popis vzhľadu programu	25
5.3	Ukladanie a načítavanie máp	26
5.4	Pridávanie prvkov do cestnej siete	26
5.5	Práca s prvkami v cestnej sieti	27
5.6	Nastavovanie počtu prichádzajúcich a odchádzajúcich áut . .	27
5.7	Nastavovanie hlavných ciest a semaforov na križovatkách . .	29
5.8	Nastavenia programu	30
5.9	Nápoveda k programu	31
6	Záver	32
	Literatúra	33
	Prílohy	34

Název práce: Simulácia cestnej siete
Autor: Matej Dajčár
Katedra (ústav): Ústav formálnej a aplikovanej lingvistiky
Vedoucí bakalárske práce: Mgr. Petr Homola
e-mail vedoucího: Petr.Homola@mff.cuni.cz

Abstrakt: Program AutoSim má pomôcť používateľovi získať informácie o tom, ako bude vyzeráť dopravná situácia v danej cestnej sieti v určitom momente. Nasimulovaním konkrétnej situácie môže používateľ získať prehľad o vyťažení jednotlivých ciest a o situáciách na jednotlivých križovatkách alebo parkoviskách. Na základe získaných informácií môže používateľ upraviť cestnú sieť tak, aby doprava v nej spĺňala jeho predstavu. Program pomôže používateľovi navrhnuť cestnú sieť, v ktorej sa budú pohybovať autá medzi jednotlivými cieľmi - uzlami cestnej siete, podľa jeho požiadaviek.

Kľúčová slova: cestná sieť, simulácia, doprava, situácia v doprave

Title: Network traffic simulation
Author: Matej Dajčár
Department: Institute of Formal and Applied Linguistics
Supervisor: Mgr. Petr Homola
Supervisor's e-mail address: Petr.Homola@mff.cuni.cz

Abstract: AutoSim application is aimed to help the user to obtain information about the road network traffic in specific moment. By simulation of concrete situation, the user will be able to acquire utilization perspective of particular road and situational perspective of junctions and parking areas. Following acquired information, the user will be able to regulate the road network according to their conception. The program will help the user to project the road network according to their requirements. In this projected network, it will be possible to move the cars between respective destinations - road junctions.

Keywords: network, traffic, simulation

Kapitola 1

Uvod

1.1 Motivácia

Motiváciou pre tvorbu programu boli problémy v cestnej doprave, ktoré vznikajú v súčasnosti skoro vo všetkých väčších, ale aj v podstatnej časti menších miest - tzv. dopravné zápchy. V mestách s veľkým počtom obyvateľov sú dopravné zápchy každodenným problémom, vznikajú hlavne ráno a poobede v čase, kedy väčšina ľudí cestuje do a z práce, školy, na nákupy atď. Aby bolo možné pri návrhu výstavby mestských komunikácií vytvoriť takú cestnú sieť, ktorá čo najviac eliminuje dopravné zápchy a iné dopravné problémy, je potrebné zistiť z plánu navrhovanej cestnej siete problémové miesta (križovatky, cesty, atď.) . Je tiež potrebné pripraviť náhradné riešenia riadenia cestnej premávky pre prípady, že niektorá časť cestnej komunikácie sa stane na určitý čas neprejazdnou, či už kvôli výskytu náhodnej poruchy, plánovanej opravy alebo z ľubovoľného iného dôvodu.

Pre takéto prípady je potrebné dopredu vedieť, ako bude vyzeráť situácia v premávke v konkrétnej cestnej sieti. Odhad situácie, ktorá vznikne, je možné predpokladať na základe údajov, ktoré sme získali dlhodobým pozorovaním v iných cestných sieťach. Na to je potrebné získať nemalé množstvo údajov, ktoré je nutné zisťovať v širokom časovom horizonte a je to dosť prácne.

Druhým, lepším riešením je vytvoriť model cestnej premávky, v ktorom bude pohyb účastníkov cestnej premávky (dopravných prostriedkov) zodpovedať v čo najväčšej miere realite. Na tomto modeli potom môžeme simuláciu dopravy vykonávať ľubovoľne dlho, a v ktoromkoľvek okamžiku môžeme zistiť aktuálnu situáciu v doprave.

1.2 Cieľ

Cieľom práce bolo vytvoriť program, v ktorom si môže používateľ navrhnuť cestnú sieť podľa jeho vlastných požiadaviek. Program používateľovi vytvorí obraz o situácii, ktorá vznikne počas premávky na navrhnutej cestnej sieti. Používateľ môže na základe získaného obrazu zmeniť vlastnosti siete alebo jej štruktúru, niektoré prvky deaktivovať prípadne aktivovať a sledovať zmeny v doprave a situácie v jednotlivých častiach navrhnutej siete.

Cieľom programu nebolo navrhnuť optimalizačné riešenie pre lepšiu organizáciu dopravy v sieti, ale urobiť simuláciu navrhnutej cestnej siete so zohľadnením zadaných údajov o časovom a priestorovom premiestňovaní dopravných prostriedkov a na základe toho vytvoriť obraz o situácii v cestnej sieti.

Program by mal byť pomocným nástrojom pri riešení problémov v doprave a pri testovaní, aký vplyv budú mať navrhnuté zmeny a úpravy cestnej siete. Cestnú sieť tvoria uzly predstavujúce vstupy do cestnej siete, kadiaľ autá do mesta vchádzajú a odchádzajú, križovatky, ktoré spájajú jednotlivé komponenty cestnej siete a ďalej uzly predstavujúce parkoviská, na ktorých autá môžu nejakú dobu stáť. Parkoviská predstavujú určité navštevované zóny v meste, ako napríklad obchodné alebo zábavné centrá, sídliská, komerčné zóny.

Kapitola 2

Návrh

2.1 Výber témy práce

Simuláciu cestnej siete som si zvolil ako tému bakalárskej práce preto, lebo ma zaujímali simulačné metódy a ich využitie a chcel som sa dozvedieť viac o ich možnostiach a použití. Simulácie sa v dnešnej dobe využívajú stále vo väčšej miere a pomáhajú v množstve odvetví. Simulácia je metóda, pri ktorej analyzujeme chovanie určitého objektu na základe chovania jeho matematického modelu, ktorý máme k dispozícii. Vytvorený model by sa mal vlastnosťami čo najviac podobáť reálnemu objektu, aby jeho správanie čo najviac odpovedalo správaniu daného objektu v realite. Matematický model môžeme vytvoriť teoreticky na základe analýzy vlastností reálneho modelu alebo empiricky na základe dát, ktoré sme už predtým získali pri pozorovaní simulovaného objektu. Na danom modeli potom môžeme realizovať rôzne pokusy a experimenty a zisťovať, ako sa bude model vyvíjať a správať v špecifických situáciách alebo pri zadaní špecifických vstupných dát. Takto môžeme odhaliť rôzne problémy a kritické miesta alebo získať nové poznatky o správaní sa objektu, zistenie ktorých až v reálnej situácii by malo nepríjemné následky. Používanie simulácií je rozšírené hlavne v biológii a demografii, kde sa simuluje vývoj a správanie populácií organizmov, meteorológií, kde sa simuluje počasie, v ekonómii, kde sa simuluje napríklad správanie finančného trhu, v chémii a fyzike, telekomunikáciach, doprave atď..

2.2 Druhy simulácií

Poznáme dva hlavné druhy simulácie, a to simuláciu diskretnú a simuláciu spojitú.

2.2.1 Diskrétna simulácia

Diskrétna simulácia je simulácia riadená udalosťami - môžeme ňou simulovať deje, ktorých jednotlivé udalosti prebiehajú v nespojitých časových okamžikoch. Nesleduje chovanie modelu v spojitom čase, ale len v určitých okamžikoch, v ktorých sú vyvolané nejaké zmeny modelu. Čas v diskkrétnej simulácii neprebíha spojitou, ale mení sa pri výskyte spomínaných udalostí. Udalosti, ktoré majú prebehnúť, sú zaznamenávané do simulačného kalendára. V kalendári sú zotriedené podľa času výskytu, od tých, ktoré sa majú vyskytnúť najskôr až po tie posledné. Prevedenie udalosti môže napláňovať výskyt ďalšej udalosti, ktorá sa pridá a zatriedi do simulačného kalendára na miesto príslušné dobe jej výskytu. V samotnej simulácii sa potom postupne spracuje a odstráni prvá udalosť zo simulačného kalendára a prejde sa na spracovanie ďalšej, pokiaľ sú v kalendári naplánované udalosti. Simulačný kalendár býva zväčša implementovaný ako prioritná fronta spojovým zoznamom.

2.2.2 Spojitá simulácia

Spojité simulácie sú na rozdiel od diskkrétnej vhodné na simuláciu dejov, ktoré prebiehajú spojitou a v každom momente potrebujeme vedieť, v akom stave sa systém nachádza. Keďže nie je možné nasimulovať dej úplne spojitou (časové okamžiky môžu byť také malé, že ich je v ľubovoľnom intervale nekonečno mnoho), vykonáva sa spojitá simulácia v dopredu zvolených konštantných časových krokoch čo najmenej dlžky. Po každom kroku sa zistí stav modelu a aké zmeny v ňom nastali. Presnosť získaných údajov je v určitej miere nepriamo úmerná veľkosti časovej periódy medzi jednotlivými krokmi, preto je voľba veľkosti časového kroku dôležitá. Čím je perióda medzi krokmi väčšia, tým menej údajov získame a môžu sa viac líšiť od reality. Ak je veľkosť periódy medzi krokmi menšia, počítač musí vykonať za rovnakú dobu viac krokov a tým viac sa zvyšuje zaťaženie procesora. Je potrebné nájsť vhodný kompromis medzi veľkosťou periódy a požadovanou presnosťou údajov.

(Viac o druhoch simulácie a simulačných modeloch v [2])

2.3 Detaily špecifikácie a návrhu programu

2.3.1 Názov programu

Vytvorený program mal na začiatku názov zhodný s názvom celej práce - Simulácia cestnej siete. Tento názov mal poskytovať dostatočnú informáciu o téme a funkcii programu aj používateľovi, ktorý sa s ním ešte nestretol. Počas práce na programe som sa ale kvôli veľkej dĺžke pôvodného názvu rozhodol premenovať ho na "Autosim". Tento názov neposkytuje takú informáciu o zameraní programu ako pôvodný, ale je kratší a používateľ si ho ľahšie zapamätá.

2.3.2 Požiadavky na funkcie programu

Program má používateľovi vytvoriť prostredie pre pohodlný návrh cestnej siete a na samotnú simuláciu dopravy. Systém a postup vytvárania siete (cestnej mapy) musí byť dostatočne jasný aj menej skúsenému používateľovi, ktorý doteraz s podobným programom nepracoval.

Navrhnutá cestná sieť sa skladá z uzlov a ciest.

Uzly sú rozdelené na:

* vstupy do mesta a výstupy z mesta, kadiaľ autá do siete vchádzajú a odchádzajú. Pri jednotlivých uzloch sa dá nastavovať počet áut, ktorý nimi prejde každým smerom (smerom do cestnej siete alebo von z cestnej siete). V závislosti na dennej dobe sa dajú informácie o počte prichádzajúcich a odchádzajúcich áut meniť, tak ako sa to deje aj v reálnom svete (ranné a poobedňajšie dopravné špičky v doprave).

* križovatky, na ktorých sa spájajú jednotlivé cesty. Na križovatkách je možné určiť typ riadenia dopravy, či to je križovatka so semaformi alebo bez semaforov.

Ak má riadenie semaformi, dá sa na nej nastaviť postupnosť jednotlivých smerov, ako budú mať zelenú, a ako dlho budú jednotlivé stavy trvať. V prípade, že je bez semaforov, bude sa na križovatkách dať nastaviť, či má určenú hlavnú cestu alebo je bez hlavnej cesty.

Ak má určenú hlavnú cestu, je možné na nej zvoliť, ktoré cesty sú hlavné

a ktoré vedľajšie. Ak križovatka nemá hlavné cesty, riadi sa na nej doprava pravidlom pravej ruky.

* parkoviská, ktoré predstavujú určité navštevované oblasti, kde autá určitý čas parkujú. Pri jednotlivých parkoviskách sa dá nastaviť počet prichádzajúcich a odchádzajúcich áut za časový interval, počet môže byť premenlivý podľa dennej doby, tak, ako to je pri vstupoch do mesta. Parkoviská majú okrem toho určenú kapacitu, maximálne koľko áut sa na nich môže nachádzať v jednom okamihu.

Navrhnutú cestnú sieť je možné editovať, meniť vlastnosti jednotlivých uzlov alebo ciest a meniť jej tvar.

Cestnú sieť je možné uložiť perzistentne do súboru, odkiaľ je ju možné opätovne načítať do programu.

2.3.3 Vzhľad programu

Vzhľad programu a užívateľské prostredie má byť prívetivé a pre používateľa príjemné. Väčšina užívateľov si po spustení programu ako prvú vec všimne jeho vzhľad. Preto musí prostredie programu pôsobiť prívetivo, tak, aby jeho vzhľad neodradil užívateľa hneď po prvom spustení programu od skúmania jeho ďalších funkcií a následného používania programu. Pri návrhu prostredia pre používateľa som sa snažil, aby sa dalo v programe dobre orientovať, ovládanie programu aby bolo jednoduché a zrozumiteľné a aby sa program ľahko ovládal. Skoro všetky ovládacie prvky sú použiteľné alebo zobraziteľné už z hlavného formulára a väčšina príkazov z menu je ľahko dostupná pomocou klávesových skratiek.

2.3.4 Platforma

Program má bežať pod operačným systémom MS Windows (verzia 2000 a vyššia). Program využíva technológiu .NET Framework verzie 2.0, ktorá je voľne šíriteľná a jej autorom je spoločnosť Microsoft. Nebola stanovená požiadavka prenositeľnosti programu na iné platformy.

Kapitola 3

Popis funkčnosti programu

3.1 Výber typu simulácie

Z dvoch vyššie spomínaných typov simulácie som si vybral simuláciu spojitú. Zadaná téma by sa dala riešiť aj diskretnou simuláciou, ale bolo by to zbytočne náročnejšie. Po cestnej sieti sa pohybujú autá, ktoré sa pri každom kroku simulácie zobrazujú na mape, takže je potrebné vedieť v každom momente ich súradnice, aby sme ich mohli na mapu vykresliť. Keby sme pohyb každého auta mali pridávať do simulačného kalendára, vyžadovalo by si to neúmernú prácu s pridávaním a odoberaním udalostí z kalendára, keďže áut môže byť v meste veľmi veľa. Autá sa pohybujú po cestách v konštantných intervaloch, takže nám viac vyhovuje simulácia spojitá.

3.2 Určenie trás v cestnej sieti

Aby bolo možné smerovať jednotlivé autá v cestnej sieti, je treba mať uloženú informáciu, ako sa dostaneme z ľubovoľného uzla do iného ľubovoľného uzla. Cestná sieť predstavuje orientovaný ohodnotený graf. Slovo graf môžeme v našom význame definovať ako dvojicu (V, E) , kde V je množina vrcholov a E je množina hrán vedúcich medzi jednotlivými vrcholmi z množiny V . Graf je orientovaný, ak majú jednotlivé hrany určený smer, z ktorého vrcholu smerujú do ktorého, a ohodnotený, ak majú jednotlivé hrany priradené určité hodnoty - váhy. V cestnej sieti môžu byť hrany ohodnotené len nezáporne, cesta nemôže mať zápornú dĺžku, takisto by nemalo žiaden zmysel, ak by bola najvyššia povolená rýchlosť na ceste záporná. Na zis-

tenie trás medzi jednotlivými uzlami existuje viac typov algoritmov, ako sú napríklad *Dijkstrov algoritmus*, *Floyd-Warshallov algoritmus*, *Bellman-Fordov algoritmus* a iné. Na výpočet najkratších ciest medzi vrcholmi som v tomto programe použil Dijkstrov algoritmus vzhľadom k jeho nenáročnej implementácii a výpočtovej zložitosti. V programe je možný výber viacerých kritérií, podľa ktorých sa určujú najkratšie trasy. Je možné vybrať, či sa budú trasy určovať podľa dĺžky ciest, podľa najkratšieho času potrebného na prechod cestou (určí sa podľa celkovej dĺžky cesty a rýchlosti, akou sa autá na danej ceste pohybujú) alebo podľa najmenšieho počtu uzlov, ktorými hľadaná trasa bude viesť. Dijkstrov algoritmus zistí na jedno zavolanie najkratšie trasy zo zvoleného uzla do všetkých ostatných, takže v programe sa prevedie Dijkstrov algoritmus pre každý uzol cestnej siete tak, aby sme poznali trasu medzi ľubovoľnou dvojicou uzlov. Pre istotu uvediem zjednodušený popis Dijkstrovho algoritmu zisťujúceho najkratšiu cestu z jedného uzla do všetkých ostatných. Každý uzol obsahuje premennú značiacu jeho hodnotu v Dijkstrovom algoritme. Uzly sú rozdelené na množinu dočasných a trvalých uzlov. Hodnota trvalých uzlov sa už meniť nebude, hodnota dočasných sa ešte meniť môže. Pred začiatkom sú všetky uzly označené ako dočasné a ich hodnota je nastavená ako nekonečno. V počítači to môžeme označiť napr. maximálnou hodnotou, akú môže obsahovať typ danej premennej. Uzlu, z ktorého zisťujeme trasu, nastavíme hodnotu na 0 a zaradíme ho do množiny trvalých uzlov. Všetkým uzlom, do ktorých smeruje z tohto uzla cesta, zmeníme hodnotu na takú, aké má ohodnotenie daná cesta. Ohodnotenie nám prevedie zvolená ohodnocovacia funkcia. Ďalej už len postupne vyberáme z množiny dočasných uzlov uzol s najmenšou hodnotou, ak taký existuje, prehlásime ho za uzol trvalý a všetkým uzlom do ktorých z neho vedie cesta nastavíme hodnotu na minimum z jeho aktuálnej hodnoty a súčtu hodnoty práve preradeného uzla a ohodnotenia cesty, ktorá spája spomínanú dvojicu uzlov. Algoritmus končí, ak sú všetky uzly prehlásené za trvalé alebo ak už nie je v množine uzlov žiaden uzol, ktorého hodnota je iná ako nekonečno - to znamená že tieto uzly nie sú dostupné. Pri každej zmene hodnoty uzla si zaznamenáme jeho predchodcu, to je uzol, z ktorého sme sa do daného uzla dostali. Spätným prechodom k počiatočnému uzlu, z ktorého sme zisťovali trasu, získame trasu najkratšej cesty.

3.3 Výber cieľov jednotlivých áut

Každé auto má určený cieľ - miesto, kam má ísť. Je to buď vstup/výstup z cestnej siete alebo parkovisko. Ak je cieľom auta vstup/výstup, auto sa po dosiahnutí cieľa z cestnej siete odstráni. Ak je cieľom parkovisko, auto sa po dosiahnutí cieľa odpojí z cesty a pripojí sa do zoznamu áut na parkovisku. Z tohto zoznamu sa autá opäť vracajú do cestnej siete po uplynutí čakacej doby. Výber cieľov pre autá je možné zvoliť náhodne alebo podľa hodnôt, koľko áut má do uzla prísť za časový interval. Túto hodnotu má nastavený každý uzol typu vstup a parkovisko. Ak je zvolený náhodný výber cieľa, pomocou generátora náhodných čísel sa vyberie číslo predstavujúce ID nejakého uzla, ktorý sa potom stáva cieľom daného auta - samozrejme v prípade, ak je vybraný uzol dostupný z uzla, v ktorom sa auto práve nachádza.

V prípade, že cieľ je zvolený podľa uvedených hodnôt, použije sa nasledujúci algoritmus:

- na začiatku sa inicializuje zvolená premenná na 0
- postupne sa prejde zoznam vstupov a parkovísk a pri každom prvku sa daná hodnota inkrementuje o hodnotu uzla, koľko do neho má prísť áut
- pri každej inkrementácii premennej sa jej hodnota uloží na nové miesto do poľa, ktorého veľkosť je rovná súčtu počtu vstupov a parkovísk
- pri samotnom výbere cieľa sa náhodne vygeneruje číslo v intervale 0 - hodnota inkrementovanej premennej, a podľa indexu intervalu, v ktorom hodnota leží v danom poli, sa určí cieľ auta.

Každá cesta má nastavenú rýchlosť, akou sa po nej autá pohybujú. Všetky autá na rovnakej ceste sa pohybujú rovnakou rýchlosťou, to znamená, že sa nepredbiehajú a vzdialenosť medzi autami ostáva rovnaká, až kým sa nedostanú na koniec cesty. Je to kvôli tomu, že každé auto si pamätá vzdialenosť od objektu pred ním, či už od auta alebo ak pred ním nie je žiadne auto, tak od uzla, do ktorého cesta vedie. Pri posúvaní áut sa zmenší len vzdialenosť prvého auta, ostatné vzdialenosť ostávajú rovnaké. Vzdialenosť medzi idúcimi autami je väčšia ako medzi stojacimi autami. Pri pripájaní auta na

novú cestu sa z pruhov cesty vyberie ten, v ktorom je najviac miesta od posledného auta a do tohto pruhu sa auto pripojí. Autá medzi jednotlivými pruhmi na jednej ceste neprechádzajú.

3.4 Chod programu

Chod programu zabezpečujú hodiny - timer. Pri každom tiknutí hodín sa prevedie jeden krok simulácie. Krok simulácie spočíva v prejení a spracovaní všetkých aktívnych objektov, ktoré sa nachádzajú v cestnej sieti. Objekty jednotlivých typov sú uložené v osobitných zoznamoch, každý typ uzlov má vlastný zoznam. Pri spracovávaní sa tieto zoznamy prechádzajú po samostatných prvkoch a pri každom prvku sa vykonajú príslušné akcie.

Na začiatku sa prejdú všetky vstupy. Pri každom vstupe sa skontroluje, či je čas, kedy má vstupom vojsť do cestnej siete nové auto, a zníži sa jeho hodnota. Ak už auto má vojsť a v niektorom pruhu cesty je dostatok miesta na jeho pripojenie, tak sa zo zoznamu, kam sa odkladajú autá, ktoré opúšťajú simuláciu, odpojí auto, nastaví sa mu cieľ, kam má ísť a pripojí sa na cestu do daného pruhu. Nakoniec sa nastaví nový čas odchodu ďalšieho auta. Potom sa skontroluje, či je nejaké auto na ceste smerujúcej do vstupu dostatočne blízko uzla, aby sa odpojilo z cestnej siete. Ak je dosť blízko, zmaže sa, odpojí z cesty a pripojí sa do zoznamu odložených áut.

Po vstupoch sa prejde zoznam parkovísk. Pri každom parkovisku sa skontroluje rovnakým postupom ako u vstupov, či má odísť auto z parkoviska alebo nie, a či je nejaké auto na ceste smerujúcej k parkovisku dostatočne blízko na jeho odpojenie. Rozdiel je len v tom, že každé parkovisko má zoznam áut, ktoré sa na ňom nachádzajú a do tohto zoznamu sa auto, ktoré prichádza na parkovisko, pripojí. Následne pri odchode auta z parkoviska sa auto z tohto zoznamu odpojí. Autám, ktoré odchádzajú z parkoviska, sú nastavené nové ciele, kam smerujú.

Po zozname parkovísk sa prejde zoznam križovatiek. Pri každej križovatke sa prejdú všetky cesty vstupujúce do križovatky a na nich všetky pruhy. Najprv sa zistí, či je to križovatka so semaformi alebo bez. V každom pruhu sa skontroluje, či je prvé auto v ňom dostatočne blízko križovatky, aby ňou prešlo. Ak je dosť blízko, zistí sa, či môže auto prejsť križovatkou alebo nie. Zisťovanie je závislé od toho, či sú zapnuté semaforey a smer auta má zelenú, alebo od toho, či má auto prednosť v jazde pred ostatnými autami na iných cestách (vyhodnocuje sa len v prípade, ak sú nejaké autá na iných cestách blízko križovatky). Vstupujúce smery sú zotriedené podľa Ar-

cus tangensu uhla, pod ktorým vstupujú do križovatky, podľa toho sa určí, či je daná cesta napravo alebo naľavo od inej. Ak auto má prednosť a na ceste, na ktorú sa má pripojiť, je dostatok miesta a v danom kroku sa tam ešte nepripojilo žiadne iné auto, auto sa tam pripojí a zároveň sa odpojí z pruhu, kde sa aktuálne nachádza. V prípade, že sa na zvolenú cestu nedá pripojiť, ostáva stáť pred križovatkou na ceste, kde práve je. Ak sú zapnuté semafore, tak sa ešte zmenší doba do prepnutia na nový stav semaforu a ak už nastavená doba uplynula, tak sa zmení stav semaforu.

Ďalej sa prejdú všetky cesty. Na každej ceste sa v každom aktívnom pruhu posunie prvé idúce auto, ak má dostatok miesta na posunutie. Týmto sa posunú všetky autá od idúceho až po koniec cesty, lebo pozícia áut je určená vzdialenosťou od najbližšieho objektu pred nimi a tá ostáva stále rovnaká. Ak sú v prednej časti cesty nejaké stojace autá, zistí sa, či je možné ich posunúť dopredu a ak to možné je, tak sa posunú. Rad áut, ktoré stoja, môže vzniknúť len pred križovatkou, keď autá čakajú na jej prechod alebo pred parkoviskom, kedy čakajú na zaradenie na parkovisko(v prípade, že je parkovisko plné).

V programe je ešte zoznam pomocných uzlov, ktoré slúžia ako zákruty na spájanie dvoch úsekov ciest. Pomocné uzly môžu mať stupeň maximálne dva. Tento zoznam sa tiež v každom kroku prejde a autá, ktoré sú blízko pomocného uzla, sa prepoja na ďalší úsek cesty.

Kapitola 4

Implementácia

4.1 Výber jazyka a vývojového prostredia

Na vývoj programu som si zvolil jazyk C++ vzhľadom k tomu, že počas štúdia som s ním už pracoval na viacerých projektoch a aj preto, že s ostatnými programovacími jazykmi, ktoré by boli vhodné na vývoj takejto práce, som ešte nemal dostatok skúseností. Za vývojové prostredie som si zvolil Microsoft Visual Studio 2005, ktoré poskytuje dostatok možností a funkcií k tvorbe celej práce. Visual Studio 2005 umožňuje mixovať dokopy riadený a neriadený kód (managed a unmanaged code). V riadenom kóde je použitá automatická spáva pamäte (garbage collector), takže nie je potrebné starať sa o dealokovanie objektov, ktoré sa už nepoužívajú. Pri vytváraní jednotlivých formulárov je použitý riadený kód, k tvorbe samotného jadra simulácie som použil klasický neriadený kód. Program bolo možné vyvinúť celý v riadenom kóde, ale na začiatku tvorby programu som o tom ešte nemal dostatok informácií, preto som použil obidva druhy kódu.

Na správu verzií som použil program Subversion (SVN). Tento program ukladá súbory so zdrojovými kódmi vždy pod číslom nasledujúcej verzie a dokáže zobrazíť všetky zmeny, ktoré v nich boli vykonané oproti predošlej verzii. V prípade, že sa pri programovaní zanesla do programu neznáma chyba, dajú sa zobrazíť všetky zmeny vykonané od poslednej funkčnej verzie a chyba sa potom ľahšie hľadá. Je možné kedykoľvek sa vrátiť k ľubovoľnej skôr uloženej verzii programu a pokračovať v práci na nej. Pri písaní samotnej simulácie som použil knižnicu Standard Template Library (STL), z ktorej som použil hlavne kontajnery list, vector a string. Nemusel som si

tieto štruktúry vytvárať sám, takže mi to uľahčilo prácu a mohol som sa venovať iným veciam v programe (viac o STL v [1]).

4.2 Návrh objektového modelu programu

Program je napísaný objektovým štýlom a využíva sa v ňom viac typov objektov. Objektovo orientované programovanie mi umožnilo pri práci využiť výhody ako dedičnosť, polymorfizmus a zapuzdrenosť. Základnou úlohou pred začatím samotného programovania bolo navrhnuť si taký objektový model programu, ktorý by umožnil splniť všetky požiadavky na program a s ktorým by sa dobre pracovalo. Najprv som sa zaoberal návrhom uzlov cestnej siete. Všetky uzly majú spoločného predka *TUZOL*, ktorý obsahuje základné dátové položky a metódy, ktoré využívajú všetky od neho zdedené typy. Priamo od typu *TUZOL* sú odvodené typy *TPOMOCNY* a *TKRIZOVATKA*.

Typ *TPOMOCNY* neobsahuje navyše žiadne rozširujúce položky oproti typu *TUZOL*, ale objekty tohto typu sa pri vytváraní pridávajú do samostatného zoznamu, ktorý obsahuje len objekty typu *TPOMOCNY*, preto vystupuje v simulácii ako samostatný typ.

Typ *TKRIZOVATKA* obsahuje navyše oproti typu *TUZOL* premenné používajúce sa pre nastavovanie semaforov a hlavných ciest a z metód procedúru, ktorá zotriedi smery vstupujúce do križovatky podľa uhla, pod ktorým vstupujú.

Ďalej je od typu *TUZOL* odvodený typ *TVSTUPPARK*, ktorý je len abstraktným typom a obsahuje položky, ktoré používajú od neho odvodené typy *TVSTUP* a *TPARKOVISKO*. Rozširuje typ *TUZOL* o premenné, v ktorých sa počíta, koľko áut má do uzla prísť za určitý čas a koľko odísť, a o premennú určujúcu, kedy má odísť ďalšie auto. Z metód ma navyše metódu nastavujúcu odchod ďalšieho auta. Od tohto typu je odvodený typ *TPARKOVISKO*, ktorý ma navyše premenné určujúce kapacitu parkoviska a počet áut na parkovisku.

Typ *TVSTUP* je podobne ako *TPARKOVISKO* odvodený od typu *TVSTUPPARK*.

Jednotlivé cesty v cestnej sieti predstavujú objekty typu *TCESTA*. Typ *TCESTA* využíva typy objektov *TSMER*, *TPRUH* a *TFRONTA*. Objekt typu *TSMER* predstavuje jeden smer cesty. *TSMER* obsahuje objekty typu *TPRUH*, ktoré predstavujú pruhy v jednom smere. Každý objekt typu *TPRUH* obsahuje objekt typu *TFRONTA*, v tomto objekte sú uložené autá, ktoré sa v pruhu nachádzajú. V objektoch typu *TFRONTA* sú uložené autá, ktoré sú reprezentované typom *TAUTO*.

Pri zisťovaní najkratších trás sa výsledky Dijkstrovho algoritmu ukladajú do objektu typu *TSMERNIKY*, ktorý predstavuje dvojrozmerné pole celočíselných hodnôt. Toto pole sa indexuje podľa ID uzla, z ktorého chceme zistiť trasu, a podľa ID uzla, kam trasa vedie. Toto pole obsahuje na pozícií (i,j) poradové číslo smeru, ktorý vychádza z uzla, ktorého ID je rovné i, a daným smerom sa dostaneme do uzla, ktorého ID je rovné j.

Autá, ktoré sú počas behu simulácie odstránené z cestnej siete, sú ukladané do objektu typu *TZOZNAM_AUT*.

Náčrt typov objektov prvkov cestnej siete použitých v programe je v časti Prílohy na obrázku 7.1.

4.3 Popis jednotlivých typov objektov a ich funkcií

TYP TUZOL

Typ *TUZOL* je predkom všetkých ostatných uzlov cestnej siete. Z hlavných dátových položiek obsahuje nasledujúce:

- int ID - každý uzol má originálne ID, ktoré ho jednoznačne identifikuje. Podľa ID sa napríklad vyberajú ciele alebo indexuje objekt *TSMERNIKY* obsahujúci informácie o najkratších cestách.
- string meno - každý uzol môže byť na mape označený menom
- int stupeň - značí koľko ciest vedie do uzla
- LSmery smery - zoznam ciest - smerov vstupujúcich do uzla

Ďalej obsahuje premenné na počítanie áut (koľko ich do uzla prišlo a koľko ich z uzla odišlo), premennú určujúcu typ uzla (či je to vstup, parkovisko, križovatka alebo pomocný uzol predstavujúci zákrutu), premennú určujúcu stav uzla (či je uzol aktívny alebo nie) a premenné predstavujúce súradnice uzla na mape.

Z dôležitejších metód obsahuje typ TUZOL tieto:

- * bool mozePridatCestu() - zistí, či je možné k uzlu pripojiť ďalšiu cestu
- * TSMER* vratSmer(int i) - vráti smer na i - tej pozícii v zozname vstupujúcich smerov
- * void prehodSmery(int i, int j) - prehodí i - ty a j - ty smer v zozname smerov
- * bool odstranSmer(TSMER* co) - odstráni smer, ktorý dostane ako parameter, zo zoznamu vstupujúcich smerov
- * void odstran() - odstráni uzol z cestnej siete

Z ďalších metód obsahuje procedúry slúžiace na aktivovanie a deaktivovanie uzla v cestnej sieti.

TYP TPOMOCNY

Typ TPOMOCNY je zhodný s typom TUZOL.

TYP TKRIZOVATKA

Z hlavných dátových položiek, rozdielných od typu TUZOL, obsahuje typ TKRIZOVATKA tieto:

- bool semaforey - značí, či sú na križovatke zapnuté semaforey
- int idSemaf - určuje stav, ktorý je práve na semafore
- int hlavna1, hlavna2 - čísla hlavných ciest
- vector stavySemaf - zoznam jednotlivých stavov semaforu

Z metód obsahuje tieto:

- `double vratUhol(int i)` - vráti arcus tangens uhla, pod ktorým vstupuje do križovatky `i` - ty smer. Používa sa pri zoradovaní vstupujúcich smerov.
- `void zotried()` - zotriedi vstupujúce smery proti smeru hodinových ručičiek.

Križovatky môžu byť len stupňa 3 alebo 4. V križovatkách sú vstupujúce smery zotriedené proti smeru hodinových ručičiek pomocou arcus tangens uhla, pod ktorým vstupujú do križovatky. Podľa toho sa určujú pozície vstupujúcich ciest vzhľadom k ostatným vstupujúcim cestám. Každá križovatka obsahuje zoznam, v ktorom sú uložené stavy semaforu v poradí, v akom ich určil používateľ. Všetky stavy semaforov, ktoré sa môžu v simulácii vyskytovať, sú uložené v štvorrozmernom globálnom poli `bool` stavy. Toto pole sa indexuje stupňom križovatky, číslom aktuálneho stavu na semafore, poradovým číslom smeru (v zozname vstupujúcich smerov), odkiaľ práve chce auto ísť, a poradovým číslom smeru, kam chce auto ísť. Pole obsahuje hodnotu, či auto môže pri danom stave semaforu prejsť zo smeru, na ktorom sa práve nachádza, na smer, kadiaľ má pokračovať v ceste.

TYPY TVSTUP A TPARKOVISKO

Typy `TVSTUP` a `TPARKOVISKO` majú spoločného abstraktného predka typ `TVSTUPPARK`. Z hlavných položiek, ktoré majú tieto typy oproti typu `TUZOL` navyše, sú to položky počítajúce počet prejdenných áut, či už prichádzajúcich alebo odchádzajúcich, a položky, v ktorých sú nastavené hodnoty, koľko áut má cez uzol vojsť/výjsť za určitý čas. Z metód obsahujú navyše len `TPARKOVISKO` metódy pripájajúce a odpájajúce auto na parkoviska, v respektíve z parkoviska.

TYP TCESTA

Hlavné dátové položky, ktoré obsahuje typ `TCESTA` sú:

- `int dlzka`, `int rychlost` - premenné, v ktorých je uložené dĺžka cesty a rýchlosť akou sa po nej autá pohybujú. Rýchlosť je v `m/s`.
- `TSMER smer1`, `smer2` - premenné reprezentujúce dva smery cesty

- TUZOL* od, *kam - ukazatele na uzly odkiaľ a kam cesta vedie

Typ TCESTA nemá žiadne špeciálne metódy, ktoré by bolo potrebné spomenúť. Rýchlosť na ceste môže byť jednou z hodnôt 36, 54, 72 alebo 90 km/hod. Je to kvôli tomu, aby bola veľkosť posunu v kroku simulácie celočíselná, aby nevznikali nepresnosti pri zaokružovaní na celé čísla, keďže súradnice áut môžu byť len celé čísla. Tieto rýchlosti sú pre používateľa zaokruhlené na 30, 50, 70 a 90 km/hod.

TYP TSMER

Každá cesta obsahuje dva objekty typu TSMER. Typ TSMER obsahuje tieto hlavné dátové položky:

- TSMER* opacny - ukazateľ na opačný smer tej istej cesty
- pocPruhov - premenná obsahujúca počet pruhov v danom smere
- TCESTA* rodic - ukazateľ na cestu, ktorej smer patrí
- TUZOL* od, kam - ukazatele na uzly, odkiaľ a kam smer smeruje
- TPRUH pruhy[MAXPRUHOV] - pole jednotlivých pruhov smeru

Z dôležitejších metód, ktoré obsahuje typ TSMER sú to:

- void posunAuta() - posunie autá vo všetkých pruhoch smeru. Posun áut je realizovaný zmenšením vzdialenosti prvého idúceho auta na ceste o hodnotu, o koľko sa má alebo môže posunúť. Takisto, ak sa dá, je zmenšená vzdialenosť stojacích áut čakajúcich pred uzlom na jeho prechod.
- bool mozePripojit() - zistí, či je možné pripojiť auto do nejakého pruhu v danom smere.
- bool pripojAuta(TAUTO* Auto) - pripojí auto do pruhu, v ktorom je najviac miesta .
- TAUTO odpojAuta(TAUTO* Auto) - odpojí auto, ktoré dostane ako parameter, z pruhu.

TYP TPRUH

Typ TPRUH predstavuje jeden pruh v smere na ceste. Obsahuje frontu áut, ktorá obsahuje ukazatele na autá, ktoré sa nachádzajú v pruhu. Každý pruh obsahuje ukazateľ na prvé idúce auto v pruhu a pomocný ukazateľ, kam sa odloží auto, ktoré bolo pripojené do pruhu skôr, ako bol spracovaný. Má ešte premennú určujúcu, koľko miesta je v pruhu od začiatku cesty po najbližšie auto a premenné, v ktorých sú uložené súradnice začiatku a konca pruhu. Definované má procedúry `void odpojAuto(TAUTO* Auto)` a `void pripojAuto(TAUTO* Auto)`, ktoré odpájajú z pruhu alebo pripájajú auto do pruhu, ktoré dostanú ako parameter a ešte procedúru `posunAuto(int posun, int hranica)`, ktorá posunie auta v pruhu o zadaný posun.

TYP TFRONTA

Objekt tohto typu slúži na ukladanie áut v pruhoch. Každý pruh na jednotlivých cestách obsahuje jeden objekt typu TFRONTA. Uložené sú v ňom ukazatele na jednotlivé auta nachádzajúce sa v pruhu. Na samotnú frontu je použitý kontajner list z knižnice STL. Typ má definované metódy určujúce prvé a posledné auto fronty a procedúry na pripájanie a odpájanie áut `void pripoj (TAUTO* Auto)` (pripojí auto na koniec zoznamu áut), `TAUTO* odpojPrve()` (odpojí prvé auto nachádzajúce sa v zozname áut) a `TAUTO* odpojAuto(TAUTO* Auto)` (odpojí auto predané ako parameter procedúry).

TYP TAUTO

Objekty typu TAUTO reprezentujú autá v simulácii. Obsahujú položky pre súradnice auta na mape, položku obsahujúcu ID cieľového uzla, položku, ktorá obsahuje vzdialenosť od najbližšieho objektu v pruhu pred autom a položky obsahujúce ukazatele na objekt pred sebou a na auto nasledujúce za sebou v tom istom pruhu. Obsahujú ešte premennú, ktorá značí, či auto v pruhu ide alebo stojí.

Z metód definovaných pre objekt typu TAUTO je dôležitá len metóda `void vyberCiel()`, ktorá zvolí autu cieľ, kam sa má auto počas simulácie dostaviť.

TYP TZOZNAM_AUT

Simulácia obsahuje jeden objekt typu TZOZNAM_AUT, ktorý sa stará o vytváranie áut, ktoré majú vstúpiť do simulácie a o autá, ktoré z nej odchádzajú. Obsahuje zoznam áut, do ktorého sa autá pri odchode zo simulácie ukladajú a z tohto zoznamu sa potom auto vráti, keď je potrebné vytvoriť auto, ktoré má vstúpiť do simulácie. Šetrí to prácu pri správe pamäte. Pamäť, ktorú zaberá auto odchádzajúce zo simulácie by sa musela po jeho odchode uvoľniť a pri následnej požiadavke na vytvorenie auta by sa musela znova naalokovať. Takto sa auto odloží do zoznamu a keď je potreba, tak sa vráti späť do simulácie. Slúži to ako takzvaný „recyklátor áut“. V súčasnej dobe sú počítače vybavené dostatkom operačnej pamäte, takže miesto, ktoré v pamäti zaberajú nepotrebné auta, nespôsobuje žiaden problém. Zoznam má stanovený maximálny počet áut, ktorý v ňom môže byť odložený a po dosiahnutí tejto hranice sa ďalšie autá, ktoré by sa v ňom chceli uložiť dealokujú.

4.4 Organizácia objektov v programe

Objekty sú v programe uložené v spojových zoznamoch. Zoznamy obsahujú ukazatele odkazujúce na jednotlivé objekty. Ako zoznamy sú použité kontajnery list z knižnice STL. Každý typ uzlov cestnej siete má svoj vlastný zoznam, v ktorom sú uložené len objekty daného typu. V simulácii tak existuje zoznam ukazateľov na parkoviská, zoznam ukazateľov na vstupy, križovatky, pomocné uzly a zoznam ukazateľov na cesty. Objekty sa do zoznamov pridávajú pri svojom vytvorení (prebieha to v ich konštruktoch) a odstraňujú sa zo zoznamov pri ich dealokácii (samotné odstránenie prebieha v deštruktoch objektu). Pri vytvorení cesty medzi dvoma uzlami sa do každého uzlu pridá odkaz na vstupujúci smer a pri odstránení sa z uzla odkaz zase odstráni.

V programe je použitý jeden globálny objekt typu TSMERNIKY, kde sú uložené výsledky Dijkstrovho algoritmu a jeden globálny objekt typu TZOZNAM_AUT, ktorý sa stará o alokáciu a dealokáciu pamäte pre jednotlivé autá. Program obsahuje jedno globálne pole, v ktorom sú uložené možné stavy semaforov.

Kapitola 5

Užívateľská dokumentácia

5.1 5.1 Inštalácia programu

Program je možné získať v podobe inštalačného balíčka. Program funguje pod operačným systémom Windows verzie 2000 a vyššej. Inštalácia prebehne formou inštalačného sprievodcu, v ktorom je možné zvoliť cieľové miesto, kam sa má program nainštalovať. Program pri svojom chode využíva technológiu .NET Framework verzie 2.0. Počas priebehu inštalácie programu sprievodca používateľa upozorní, že je potrebné mať túto technológiu nainštalovanú a podľa verzie inštalačného balíčka ponúkne užívateľovi stiahnutie inštalačných súborov zo stránok spoločnosti Microsoft alebo nainštalovanie z pripojeného súboru. Súčasťou programu je aj niekoľko vytvorených cestných sietí, na ktorých si používateľ môže vyskúšať ako prebieha simulácia.

5.2 Popis vzhľadu programu

Po spustení programu sa užívateľovi zobrazí hlavné okno. V hornej časti obsahuje menu, pod ktorým sú umiestnené tlačidlá slúžiace k rýchlemu ovládaniu simulácie. Tlačidlá sú rozdelené do jednotlivých skupín.

Prvá skupina slúži na ovládanie chodu simulácie. Nachádzajú sa v nej tlačidlá, pomocou ktorých sa simulácia dá spustiť, pozastaviť alebo celkom zastaviť.

Vedľa sa nachádza skupina s tlačidlami slúžiacimi na pridávanie jednotlivých prvkov cestnej siete. Obsahuje tlačidlo na pridanie vstupu, križovatky, parkoviska, pomocného uzla reprezentujúceho zákrutu a tlačidlo pre prida-

nie cesty.

Vedľa tejto skupiny je umiestnení čas simulácie a naľavo od neho sa nachádzajú prvky pre ovládanie rýchlosti simulácie a priblíženia mapy s cestnou sieťou. Všetky funkcie, ktoré zabezpečujú spomínané tlačidlá sú dostupné aj cez menu programu.

V nižšej časti okna sa nachádza oblasť mapy, ktorá bude obsahovať navrhnutú cestnú sieť.

V ľavej časti obrazovky sa vedľa okna programu zobrazujú informačné okná jednotlivých prvkov cestnej siete.

5.3 Ukladanie a načítavanie máp

Každú vytvorenú cestnú sieť je možné uložiť do súboru, z ktorého ju potom je možné opätovne načítať do programu. Na uloženie mapy pod novým názvom slúži položka *Hlavné menu* a v ňom voľba *Uložiť ako...* Na uloženie vykonaných zmien do pôvodného súboru slúži položka v hlavnom menu *Uložiť*. K načítaniu mapy slúži položka *Hlavné menu* a *Načítať mapu*. Súbor, v ktorých sú uložené jednotlivé mapy, by používateľ nemal samostatne meniť.

5.4 Pridávanie prvkov do cestnej siete

Na pridávanie prvkov do cestnej siete slúži druhá skupina tlačidiel, obsahujúca tlačidlá na ktorých sú obrázky jednotlivých prvkov. Pridávanie prvkov je tiež možné previesť kliknutím pravým tlačidlom myši nad oblasťou mapy, zo zobrazeného kontextového menu vybrať možnosť *Pridať uzol* a zo zobrazenej ponuky zvoliť požadovaný typ uzla. Možnosť *Zrušiť výber* slúži na zrušenie pridávania. Ak má používateľ zvolený nejaký typ na pridanie a rozhodne sa, že ho nechce pridať, tak zvolí túto možnosť. Aby sa prvok pridal, musí používateľ kliknúť na tlačidlo so zvoleným typom uzla alebo zvoliť typ z menu a potom daný uzol umiestniť na mapu kliknutím na vybrané miesto. Pre pridanie cesty medzi dvoma uzlami je potrebné zvoliť možnosť *Pridať cestu* alebo kliknúť na tlačidlo s obrázkom cesty a potom kliknúť postupne na obidva uzly na mape, medzi ktorými má pridávaná cesta viesť. Pred spustením simulácie program skontroluje, či má každý uzol požadovaný stupeň. Každá križovatka musí mať stupeň 3 alebo 4, všetky vstupy musia mať stupeň 1.

Pri použití lupy a nastavení velíni malej mierky sa môže simulačná mapa stať taká malá, že sa okolo nej zobrazí miesto, ktoré pri väčšom priblížení už nebude vidieť. Do tohto vzniknutého miesta sa nedajú pridávať žiadne uzly. Okolo mapy sa v takomto prípade vykreslí obdĺžnik, aby mal používateľ predstavu, ktorú časť mapy môže používať.

5.5 Práca s prvkami v cestnej sieti

Jedným kliknutím na mapu na zvolený prvok cestnej siete sa zobrazí jeho informačné okno v ľavej časti obrazovky. Sú v ňom zobrazené údaje prvku ako názov, súradnice a ďalšie hodnoty, ktoré prvku nastavil používateľ. Dvojitým kliknutím na mapu na zvolený prvok sa zobrazí jeho editačné okno, v ktorom sa nastavujú hodnoty prvku. Toto okno slúži aj na odstránenie prvku z cestnej siete, slúži na to tlačidlo *Odobrať* v dolnej časti okna. V tej istej časti je aj tlačidlo *Aktivovať/Deaktivovať*, ktoré slúži na aktivovanie resp. deaktivovanie prvku. Cez deaktivovaný prvok neprechádzajú žiadne autá, simulácia prebieha ako keby tam daný prvok nebol. Ak používateľ deaktivuje uzol, deaktivujú sa zároveň aj všetky cesty, ktoré do neho vchádzajú. Pri aktivovaní uzla sa zároveň aktivujú všetky do neho vchádzajúce cesty. Pri odstránení ľubovoľného uzla z cestnej siete sa takisto odstránia všetky cesty, ktoré do neho vedú.

Po deaktivovaní alebo odobratí prvku z cestnej siete sa pred opätovným spustením simulácie odstránia z cestnej siete všetky autá. Odobraním alebo deaktivovaním prvku sa môže stať nejaká časť siete z inej časti nedostupná, a v takom prípade by bolo potrebné všetkým autám skontrolovať nastavené ciele a prípadne zvoliť nové, čím by sa aj tak zmenil stav cestnej siete.

5.6 Nastavovanie počtu prichádzajúcich a odchádzajúcich áut

Pri vstupoch a parkoviskách je možné nastaviť počet áut, ktoré nimi prejdú v jednotlivých smeroch za minútu. Nastavovanie prebieha v ich editačných oknách.

Je možné nastaviť konštantný počet, ktorý bude rovnaký celý deň, alebo je možné nastaviť premenlivé počty závisiace od dennej doby. Nastavenie konštantného počtu sa deje priamo v editačných oknách jednotlivých prvkov.

Počty áut

Počty prichádzajúcich/odchádzajúcich áut

Smot out: Prichádzajúce auta - smer do mesta

Nastavenie

Od 1 hod Do 2 hod Počet áut 2

Interval 1

Odstániť Uložiť

Pridanie intervalu

Od 2 hod Do 4 hod Počet áut 4

Pridať interval

Zavrieť

Obr. 5.1: Formulár pre nastavovanie počtov áut.

áut. Po nastavení požadovaných hodnôt použijeme tlačidlo *Pridať interval*. Jednotlivé intervaly sa nesmú prekrývať, takže pred samotným pridaním nového intervalu prebehne kontrola, či sa pridávaný interval neprekrýva s niektorým iným a ak nie, tak sa pridá do zoznamu intervalov.

Na prezeranie zoznamu intervalov slúži stredná časť formulára. Je v nej možné meniť nastavené hodnoty intervalov, ako odoberať jednotlivé intervaly zo zoznamu. Počas dennej doby, ktorú nepokrýva žiaden nastavený interval, platí základný počet áut, ktorý je nastavený v samotnom editačnom okne prvku.

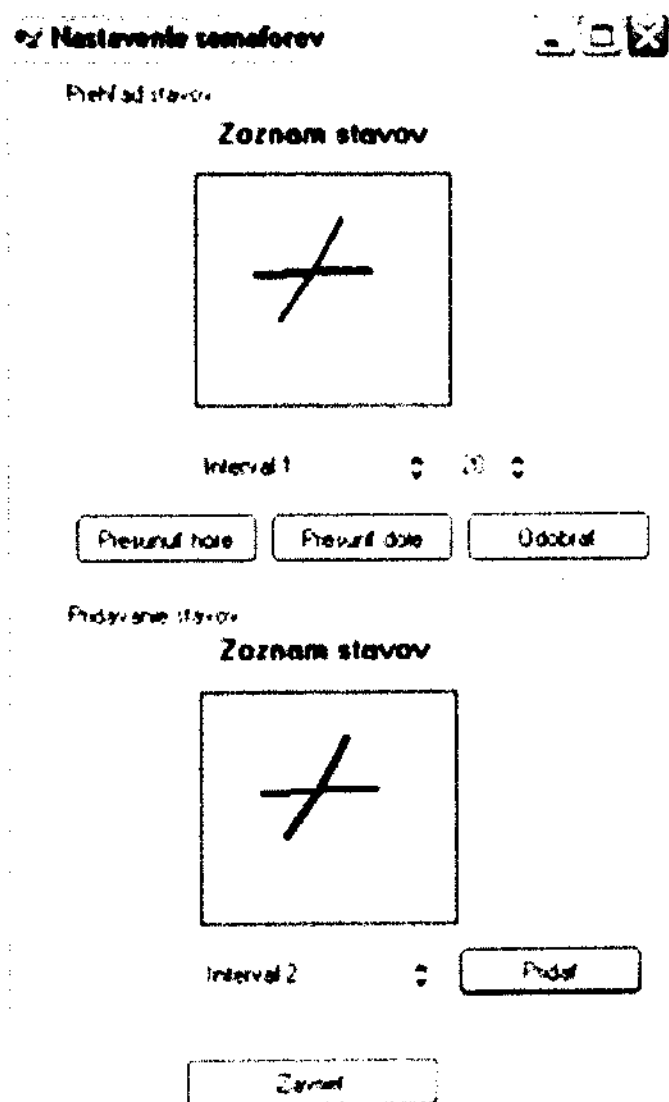
Pre nastavenie jednotlivých intervalov je potrebné v editačnom okne prvku kliknúť na tlačidlo *Detailnejšie nastavenie počtov áut*, a tým sa zobrazí osobitný formulár, v ktorom sa intervaly nastavujú. Nastavovací formulár je zobrazený na priloženom obrázku 5.1. Pre pridávanie nového intervalu a počtu áut, koľko ich má počas doby daného intervalu uzlom prechádzať, slúži spodná časť formulára. V hornej časti je potrebné zo zoznamu zvoliť, či nastavujeme hodnoty pre autá prichádzajúce alebo odchádzajúce a potom v dolnej časti nastaviť začiatok a koniec dennej doby a želaný počet

5.7 Nastavovanie hlavných ciest a semaforov na križovatkách

Nastavenie hlavnej cesty alebo semaforov na križovatke prebieha v editačnom okne križovatky, ktoré sa zobrazí po dvojitom kliknutí na zvolenú križovatkú.

Ak používateľ zvolí, že na danej križovatke majú byť semaforey, je potrebné zvoliť zo skupiny *Typ križovatky* možnosť *So semaformi* a následne nastaviť stavy na semaforocho. Stavy semaforov sa nastavujú v okne, ktoré sa zobrazí po kliknutí na tlačidlo *Nastavenie semaforov*.

Zobrazené okno pre nastavovanie semaforov je na priloženom obrázku 5.2.



Horná časť okna poskytuje prehľad o pridaných stavoch a ich nastaveniach. Je v nej možné jednotlivým stavom aj meniť nastavenia a ich poradie, prípadne zvolený stav odobrať zo zoznamu stavov. Slúžia na to tlačidlá *Posunúť hore*, *Posunúť dole* a *Odobráť*. Prvé dve spomenuté posúvajú zvolený stav o jednu pozíciu vyššie alebo nižšie v zozname stavov a tretie zvolený stav odoberie zo zoznamu. V okienku napravo je možné meniť dobu, po ktorú je interval na semafore aktívny.

Dolná časť okna slúži na pridávanie nových stavov. Je potrebné vybrať stav zo zoznamu a následne kliknúť na tlačidlo *Pridať*. Pri výbere stavov zo zoznamu sa stavy zobrazujú v ohraničenej oblasti nad zoznamom stavov, aby mal používateľ prehľad o tom, aký stav pridáva. Na obrázku sú rovnakou farbou vyznačené cesty, z ktorých na ktoré môžu počas daného stavu prúdiť autá.

Obr. 5.2: Formulár pre nastavovanie semaforov.

Ak má byť križovatka bez semaforov, je potrebné zakliknúť v skupine *Typ križovatky* možnosť *Bez semaforov*. Týmto sa aktivuje druhá skupina volieb, v ktorej je možné zvoliť, či bude na križovatke určená hlavná cesta alebo sa bude doprava riadiť pravidlom pravej ruky.

Ak má byť určená hlavná cesta, je potrebné zvoliť možnosť *S hlavnou cestou*. Zvolením tejto voľby sa v nižšej časti editačného okna aktivuje obrázok s náčrtom križovatky, na ktorom je potrebné zvoliť dve hlavné cesty. Kliknutím na ľubovoľnú cestu z náčrtku sa táto cesta označí ako hlavná a farebne sa zvýrazní. Takto je potrebné označiť dve cesty. Opätovným kliknutím na označenú cestu sa cesta odznačí a je možné označiť inú cestu.

5.8 Nastavenia programu

Položka menu *Nastavenia* ma viacero možností. Prvou z nich je možnosť nastaviť rozmery mapy. Minimálne povolené rozmery mapy sú 1000 x 1000 a maximálne 10 000 x 10 000.

Druhou položkou v poradí je možnosť *Nastavenia počiatočných hodnôt*, ktorá umožňuje nastaviť počiatočné hodnoty prvkov, s ktorými sa budú do siete jednotlivé prvky pridávať. Je možné nastaviť hodnoty vstupov a parkovísk, koľko áut do nich má prichádzať a odchádzať, ďalej kapacitu parkovísk a rýchlosť na cestách, ako sa po nich budú pohybovať autá.

Ďalšou možnosťou je voľba *Zobrazíť podľa hustoty*, ktorá indikuje, či sa cesty počas simulácie budú farebne zvýrazňovať podľa toho, koľko áut na nich bude priemerne na 100 metrov. Táto voľba umožňuje používateľovi ľahšie zistiť, ktoré cesty sú najviac vyťažené a je na nich najviac áut vzhľadom k ich dĺžke. Poslednou možnosťou je voľba *Najkratšie trasy*, ktorá slúži na voľbu kritéria, podľa ktorého sa v cestnej sieti určujú najkratšie trasy medzi uzlami. Je možné zvoliť z troch kritérií, a to podľa najmenšieho počtu uzlov na trase, podľa najmenšieho súčtu dĺžok jednotlivých ciest trasy a podľa výsledného času, ktorý trvá prejdienie všetkých ciest trasy. V kritériu určujúcom trasy podľa počtu uzlov sa do výsledného súčtu nezapočítavajú uzly predstavujúce zákruty. V kritériu určujúcom trasy podľa času trasy sa čas určuje podľa dĺžok jednotlivých ciest a na nich určených rýchlostí. Do času nie je započítaná doba, ktorú bude auto čakať pred križovatkami na ich prechod.

Program ešte umožňuje zobrazit' najkratšiu trasu medzi dvoma zvolenými uzlami. Na zobrazenie trasy je potrebné kliknúť pravým tlačidlom myši na oblasť mapy a zo zobrazeného kontextového menu zvoliť položku *Nájsť najkratšiu trasu*. Následne treba kliknutím na mapu zvoliť dva uzly, medzi ktorými sa má trasa zobrazit'. Cesty, patriace do vybranej trasy, sa zvýraznia červenou farbou a vykreslia hrubšie ako ostatné.

5.9 Nápoveda k programu

Program obsahuje nápovedu, ktorú je možné vyvolať položkou menu *Nápoveda*. Je v nej podrobne zdokumentované ovládanie programu a práca s ním, takže nápoveda pomôže užívateľovi v situácii, keď má nejaký problém alebo mu niečo v programe nie je jasné. Nápoveda je uložená v samostatnom súbore AutoSim.chm.

Kapitola 6

Záver

Výsledkom práce na zadanú tému Simulácia cestnej siete je program Auto-Sim, ktorý umožňuje navrhnuť cestnú sieť a simulovať na nej pohyb áut.

Pri tvorbe programu a celej práce som získal skúsenosti, aké stupne zahŕňa vývoj väčšieho softwarového diela a čo všetko je pri jeho vývoji potrebné vykonať. Pred touto prácou som ešte nemal skúseností s tvorbou programu takéhoto rozsahu.

Počas samotného vývoja programu som prešiel všetkými fázami, ktoré tvorba zahŕňa, od návrhu funkčnej špecifikácie cez návrh objektového modelu, implementačnej časti až po tvorbu programátorskej a užívateľskej dokumentácie.

Po vytvorení programu ešte bolo potrebné otestovať, či program spĺňa všetky požiadavky, ktoré by mal spĺňať. Pri testovaní som zistil, že program vyhovuje celému zadaniu a spĺňa všetky požiadavky, ktoré boli uvedené v zadanej funkčnej špecifikácii.

Pokračovaním vývoja tohto programu by bolo možné rozšíriť ho o radu ďalších funkcií:

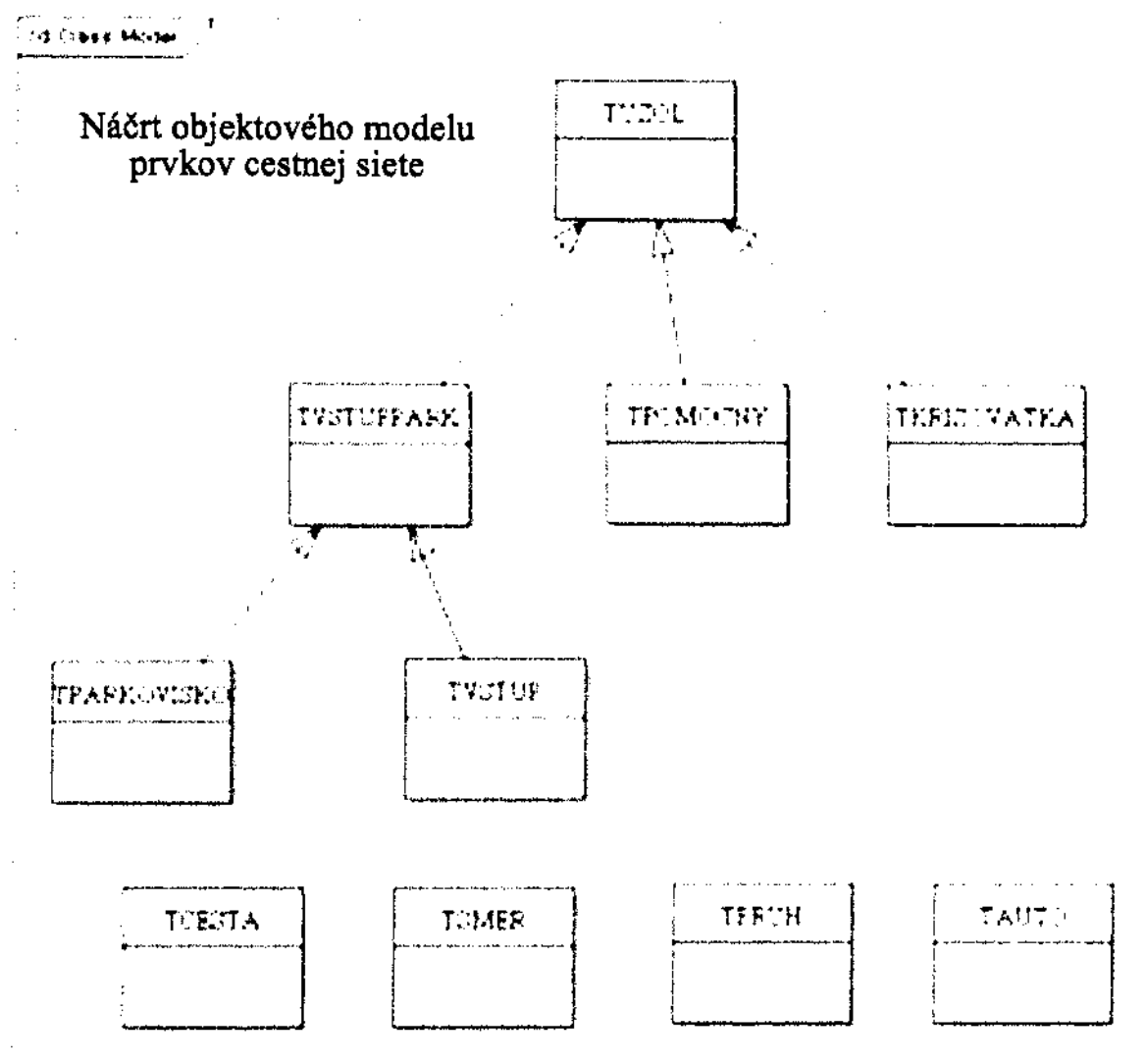
- * implementovanie algoritmov, optimalizujúcich nastavenie cestnej siete
- * pridanie viacerých druhov dopravy
- * pridanie chodcov a mestskej hromadnej dopravy do cestnej siete

Literatúra

- [1] Bruce Eckel: *Thinking in C++ 2nd Edition*, Prentice Hall Inc. A Paramount Communications Company, New Jersey, 1998.
- [2] Nathan Gartner, Carroll J. Messer, Ajay K. Rathi: *Traffic Flow Theory*, Chapter 10, <http://www.tfhrc.gov/its/tft/tft.htm>
- [3] Niklaus Wirth: *Algoritmy a štruktúry údajov*, Alfa, Bratislava, 1989.

Prílohy

Náčrt objektového modelu prvkov cestnej siete použitého v programe.



Obr. 7.1: Náčrt objektového modelu.