|  |
| --- |
| OSTRAVSKÁ UNIVERZITA  PŘÍRODOVĚDECKÁ FAKULTA  KATEDRA INFORMATIKY A POČÍTAČŮ |
| Systém pro simulaci pohybu v křižovatce s využitím Fuzzy přístupů  Diplomová práce |
| Autor práce: Bc. Jan Mikulík  Vedoucí práce: RNDr. Marek Vajgl Ph.D. |
| 2020 |

|  |
| --- |
| UNIVERSITY OF OSTRAVA  FACULTY OF SCIENCE  [DEPARTMENT OF INFORMATICS AND COMPUTERS](http://prf.osu.eu/kip) |
| System for an intersection simulation using fuzzy approaches  DIPLOMA THESIS |
| Author:  Bc. Jan Mikulík  Supervisor:  RNDr. Marek Vajgl Ph.D. |
| 2020 |

(Zadání vysokoškolské kvalifikační práce)

ABSTRAKT

Český text abstraktu

*Klíčová slova:*

*(klíčová slova vypsaná na řádku, oddělená od sebe čárkami)*

**ABSTRACT**

The text of the abstract.

*Keywords:*

čestné prohlášení

Já, níže podepsaný/á student/ka, tímto čestně prohlašuji, že text mnou odevzdané závěrečné práce v písemné podobě je totožný s textem závěrečné práce vloženým v databázi DIPL2.

Ostrava dne

………………………………

podpis studenta/ky

|  |
| --- |
| Poděkování |
| Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval/a samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal/a, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.  V Ostravě dne . . . . . . . . . . . .  . . . . . . . . . . . . . . . . . .  (podpis) |

OBSAH

[ÚVOD 10](#_Toc29971630)

[1 CÍL PRÁCE 11](#_Toc29971631)

[2 SOUČASNÁ ŘEŠENÍ 12](#_Toc29971632)

[2.1 Simulátor dopravní sítě - HakChol Pak 12](#_Toc29971633)

[2.2 Simulace automobilového provozu – Ivan Gregor 12](#_Toc29971634)

[2.3 Traffic Lanes - ShadowTree 13](#_Toc29971635)

[3 požadavky na funkci aplikace 14](#_Toc29971636)

[3.1 Seznam požadavků 14](#_Toc29971637)

[3.1.1 Fuzzy logika 14](#_Toc29971638)

[3.1.2 Detekce křižovatek 14](#_Toc29971639)

[3.1.3 Určení přednosti v jízdě 14](#_Toc29971640)

[3.1.4 Návrh silnic 14](#_Toc29971641)

[3.1.5 Řízení dopravy semafory 14](#_Toc29971642)

[3.1.6 Řízení dopravy policistou 15](#_Toc29971643)

[3.1.7 Uložení/načtení návrhu 15](#_Toc29971644)

[3.1.8 Editace celého návrhu 15](#_Toc29971645)

[3.1.9 Rozlišení trasy pro auta/tramvaje 15](#_Toc29971646)

[3.1.10 Editace pozadí 15](#_Toc29971647)

[3.1.11 Signalizace brzd 15](#_Toc29971648)

[3.1.12 Signalizace odbočení vozidel 15](#_Toc29971649)

[3.1.13 Ovládání vlastního vozidla 15](#_Toc29971650)

[3.1.14 Statistické údaje 16](#_Toc29971651)

[3.1.15 Nastavení simulace 16](#_Toc29971652)

[3.1.16 Režimy aplikace 16](#_Toc29971653)

[3.2 Kritéria pro splnění cílů práce 16](#_Toc29971654)

[3.3 Priorita dokončení požadavků 17](#_Toc29971655)

[4 Fuzzy logika 18](#_Toc29971656)

[4.1 Co je fuzzy logika 18](#_Toc29971657)

[4.1.1 Fuzzy proměnná 19](#_Toc29971658)

[4.1.2 Fuzzy výrazy 19](#_Toc29971659)

[4.1.3 Fuzzy pravidla 20](#_Toc29971660)

[4.1.4 Výpočet … 20](#_Toc29971661)

[4.2 Implementace v aplikaci 20](#_Toc29971662)

[4.2.1 LFLController 21](#_Toc29971663)

[4.2.2 Použití dat z LFLC v aplikaci 22](#_Toc29971664)

[4.2.3 Pohyb vozidel za sebou 24](#_Toc29971665)

[4.2.4 Rozhodnutí udělení přednosti 26](#_Toc29971666)

[4.2.5 Rozdíl obyčejný/ fuzzy 27](#_Toc29971667)

[ZÁVĚR 28](#_Toc29971668)

[RESUMÉ 29](#_Toc29971669)

[SUMMARY 30](#_Toc29971670)

[SEZNAM POUŽITÉ LITERATURY 31](#_Toc29971671)

[SEZNAM POUŽITÝCH SYMBOLŮ 32](#_Toc29971672)

[SEZNAM OBRÁZKŮ 33](#_Toc29971673)

[SEZNAM TABULEK 34](#_Toc29971674)

[SEZNAM PŘÍLOH 35](#_Toc29971675)

ÚVOD

Text

1. CÍL PRÁCE

Cílem práce je aplikace fuzzy přístupu do řízení pohybu vozidel v simulátoru křižovatky. Autor ve svém řešení pokračuje bakalářské práci. Do jím vytvořeného simulačního modelu začlení chování vozidel na základě fuzzy přístupu, čímž by měl navýšit subjektivní real-life pocit z chování simulace.

1. SOUČASNÁ ŘEŠENÍ
   1. Simulátor dopravní sítě - HakChol Pak

Aplikace byla vytvořena studentem na Karlově Univerzitě v Praze roku 2010, na které pracoval v rámci bakalářské práce.

Tato aplikace umožňuje uživateli editaci silničních situací, jež jsou vyznačeny úsečkami. Navrženou situaci lze během simulace libovolně upravovat.

Doprava je řízena pomocí semaforů, které zde jsou pro řízení provozu jediná možnost.

Během simulace lze upravovat její rychlost, aktuální počet vozidel na silnicích a jejich frekvenci generování. Při spuštění simulace se zaznamenávají statistické údaje, jako je celkový počet vozidel, ujetá vzdálenost či dobra jízdy po jednotlivých cestách.

Aplikace je vytvořena v programovacím jazyce C#. Samotnou aplikace ovšem nebylo možno otestovat, jelikož není dostupná ke stažení.

* 1. Simulace automobilového provozu – Ivan Gregor

Tato aplikace byla taktéž vytvořena v rámci bakalářské práce, na Karlově Univerzitě v Praze, roku 2006.

Práce obsahuje především popis programu pro simulaci silničního provozu, který byl vytvořen v jazyce Java.

Silniční situace jsou tvořeny úsečkami znázorňujícími cesty. Tyto situace lze do aplikace pouze vložit, nelze je vytvářet ani editovat.

Doprava je řízena pouze semafory a omezením rychlosti. Jednotlivé silnice zaznamenávají statistické údaje o jízdě vozidel jako je průměrná, maximální a minimální doba jízdy, aktuální a celkový počet vozidel na silnici.

Vozidla jedou po předem definovaných trasách a zároveň mají své fyzické vlastnosti, tedy délku vozidla, váhu, spotřebu, maximální rychlost či zrychlení. O jednotlivých vozidlech se také zaznamenávají údaje, které jsou závislé na jejich vlastnostech.

Aplikaci ovšem nebylo možno otestovat, není veřejně ke stažení. Informace byly získány pouze z dokumentace, a nelze ověřit jejich pravost.

* 1. Traffic Lanes - ShadowTree

Aplikace je určena pro mobilní zařízení s operačním systémem Android, verze 2.3 a vyšší. Jedná se o hru, ve které probíhá simulace zvolené křižovatky, při níž uživatel ovládá vozidla a semafory. Účelem hry je zajistit plynulý provoz, bez způsobení dopravní nehody.

V aplikaci je k výběru několik desítek předem vytvořených křižovatek s různými obtížnostmi. V simulaci lze upravovat frekvenci generování vozidel. Mění se zde také počasí a ubíhá čas, mění se tedy i den a noc, což ovlivňuje způsob jízdy. V aplikaci jsou zobrazeny reálné křižovatky z pohledu shora, což umožňuje její přehledné ovládání

Aplikace je možno stáhnout pomocí služby Google Play.

1. požadavky na funkci aplikace

Před začátkem práce na aplikace byly stanoveny základní funkce, které aplikace bude nabízet. Seznam také zahrnuje několik požadavků, které již dříve byly obsaženy v bakalářské práci autora, na kterou tato práce navazuje.

* 1. Seznam požadavků
     1. Fuzzy logika

Hlavním požadavkem je implementace fuzzy logiky do aplikace. Pomocí této logiky se především určuje rychlost jedoucích vozidel podle dané situace. Účelem použití je plynulejší pohyb vozidel a jejich reakce na danou situaci.

* + 1. Detekce křižovatek

Vzhledem k množnosti vytvoření složitých křižovatek je aplikaci možnost automatické detekce křížení silnic, což značně zjednodušuje její tvorbu.

* + 1. Určení přednosti v jízdě

Při spojení nebo křížení silnic je možno jednoduše zvolit, která ze silnic je hlavní a která vedlejší.

* + 1. Návrh silnic

Dopravní situaci lze vytvořit pomocí křivek znázorňujících silnice. Tyto křivky bude možno vytvořit, propojit či rozpojit nebo také smazat.

* + 1. Řízení dopravy semafory

Do návrhu lze vložit semafory, nastavit časy jejich chodu, nebo je mezi sebou propojit.

* + 1. Řízení dopravy policistou

Vytvořené silniční situace je možno řídit také pokyny policistů, kteří určují povolený směr a dobu jízdy.

* + 1. Uložení/načtení návrhu

Rozpracovaný návrh může být uložen do zvoleného souboru a poté opět načten do aplikace.

* + 1. Editace celého návrhu

S návrhem situace lze libovolně manipulovat, tedy jej posouvat, přiblížit či oddálit, nebo otočit.

* + 1. Rozlišení trasy pro auta/tramvaje

V návrhu jsou typy vozidel, tramvaje a automobily. Trasa, po které vozidlo jede, je zvolena pro každý typ vozidla zvlášť.

* + 1. Editace pozadí

Do návrhu lze vložit vlastní pozadí, kterým je možno pohybovat, změnit jeho velikost, otáčet, nebo jej smazat.

* + 1. Signalizace brzd

Při zpomalení vozidla se rozsvítí zadní brzdová světla.

* + 1. Signalizace odbočení vozidel

Pokud vozidlo odbočuje, směr jeho jízdy je signalizován blinkrem.

* + 1. Ovládání vlastního vozidla

V testovacím režimu lze vložit vlastní vozidlo, které uživatel ovládá změnou jeho rychlosti.

* + 1. Statistické údaje

Během probíhající jízdy uživatele jsou zaznamenávány údaje o provozu, dopravních nehodách či přestupcích, kterých se dopustil. Po dokončení jízdy je uživatel ohodnocen.

* + 1. Nastavení simulace

Simulaci provozu lze nastavit dle potřeb. Může být nastavena frekvence generování vozidel, zapnutí pro každý typ vozidel zvlášť, nebo spuštění semaforů.

* + 1. Režimy aplikace

Aplikace je rozdělena do tří režimů. V editačním režimu se nejprve vytvoří návrh situace. Poté lze spustit testovací režim, ve kterém uživatel ovládá své vozidlo. Vytvořený návrh lze spustit jako spořič obrazovky, při čemž probíhá simulace provozu zobrazena přes celou obrazovku.

* 1. Kritéria pro splnění cílů práce
* Aplikace je funkční, lze spustit a používat bez chyb/pádů aplikace
* Jsou splněny požadavky na funkcionalitu aplikace označené jako základní (must-be v Trellu), hodnocení závislé také na splnění požadavků (should-be)
* Aplikace využívá pro řízení provozu fuzzy logiku, auta akcelerují a brzdí plynule s ohledem na okolní provoz
  1. Priorita dokončení požadavků

Jednotlivé požadavky jsou seřazeny podle priority. Čím větší číslo, tím požadavek důležitější a na jeho dokončení může záviset funkčnost jiných funkcí. U každého požadavku je také předpokládaná doba dokončení.

|  |  |  |
| --- | --- | --- |
| **Priorita** | **Název** | **Předpoklad dokončení** |
| 5 | Fuzzy logika | 9. 2019 |
| 5 | Detekce křižovatek | 9. 2019 |
| 5 | Určení přednosti v jízdě | 10. 2019 |
| 4 | Návrh silnic | 10. 2019 |
| 4 | Řízení dopravy semafory | 11. 2019 |
| 4 | Řízení dopravy policistou | 11. 2019 |
| 3 | Uložení/načtení návrhu | 11. 2019 |
| 3 | Editace celého návrhu | 12. 2019 |
| 2 | Rozlišení trasy pro auta/tramvaje | 12. 2019 |
| 2 | Editace pozadí | 1. 2020 |
| 2 | Signalizace odbočení vozidel | 1. 2020 |
| 2 | Ovládání vlastního vozidla | 1. 2020 |
| 2 | Statistické údaje | 2. 2020 |
| 1 | Nastavení simulace | 2. 2020 |
| 1 | Režimy aplikace | 2. 2020 |

1. Fuzzy logika

Práce se zabývá implementací fuzzy logiky do simulace silniční dopravy. V této kapitole jsou základní informace o tom, co je fuzzy logika, proč a jak je využita. Také je zde popsána samotná implementace v aplikaci ukázán rozdíl oproti původní aplikaci, kde fuzzy logika implementována nebyla.

* 1. Co je fuzzy logika

Fuzzy logika se zabývá, jak již název napovídá, prací s nepřesnými daty. Lze tedy na základě pouze několika hodnot vyvodit závěry, což by při jiném způsobu řešení bylo mnohem složitější, nebo nepřesné.

Pro příklad řekněme, že chceme automatizovaně na základě teploty určovat účinnost klimatizace. Při běžném přístupu by musely být určeny parametry nebo hranice pro jednotlivé teploty, na jejichž základě by se klimatizace spustila nebo vypnula. Díky fuzzy logice tyto hranice nemusí být přesně definovány a určeny budou jen základní podmínky.

K práci s fuzzy logikou jsou využity fuzzy proměnné, což jsou vstupní a výstupní parametry logiky. Tyto proměnné jsou rozděleny na fuzzy výrazy, které znázorňují rozdělení rozsahu vstupních a výstupních proměnné na „skupiny“, pomocí kterých se následně tvoří fuzzy pravidla. Fuzzy pravidla tvoří kombinace jednotlivých fuzzy výrazů pro dané fuzzy proměnné. Pro výpočet výsledné hodnoty je na vstupní hodnoty aplikována aproximace a následně deffuzifikace.

* + 1. Fuzzy proměnná

Pro využití fuzzy logiky musí být definovány proměnné. Tyto proměnné obsahují hodnoty, se kterými se pracuje. Proměnné jsou rozděleny na vstupní, kterých může být více a jednu výstupní.

Proměnná má definovanou svou minimální a maximální hodnotu, a její střední hodnotu, které ovšem určuje rozložení proměnné, nikoliv její střed.

Každá proměnná má své výrazy, které definují jistou množinu, mezi minimální a maximální hodnotou.



1 Příklad fuzzy proměnné

* + 1. Fuzzy výrazy

Výrazy jsou parametry, které rozdělují proměnnou na menší množiny, a jednotlivým hodnotám přiřazuje stupeň příslušnosti.

Stupeň příslušnosti určuje, jak moc hodnota náleží do dané množiny. Tento stupeň má hodnotu mezi 0 a 1, při čemž 0 znamená, že do množiny nepatří a 1 že do množiny zcela náleží.

Pro jednodušší práci má každý výraz také svůj název, s nímž se pracuje při určování pravidel. Například na výše uvedeném obr 1. je označen fuzzy výraz „too close“, který označuje část fuzzy proměnné v jistém rozsahu. Namísto práce s hodnotami tohoto rozsahu je dále pracováno pouze s názvem.

Množiny a stupně příslušnosti jsou pro jednoduchou práci tvořeny grafy, které mají základní tvary. Tyto tvary jsou trojúhelník, čtyřhran a křivka.

* + 1. Fuzzy pravidla

Pravidla určují vztahy mezi vstupními a výstupními proměnnými. Tedy vztahy jednotlivých výrazů.

****

2 Příklad fuzzy pravidel

* + 1. Výpočet …

Aproximace DNF

Defuzzyvikace COG

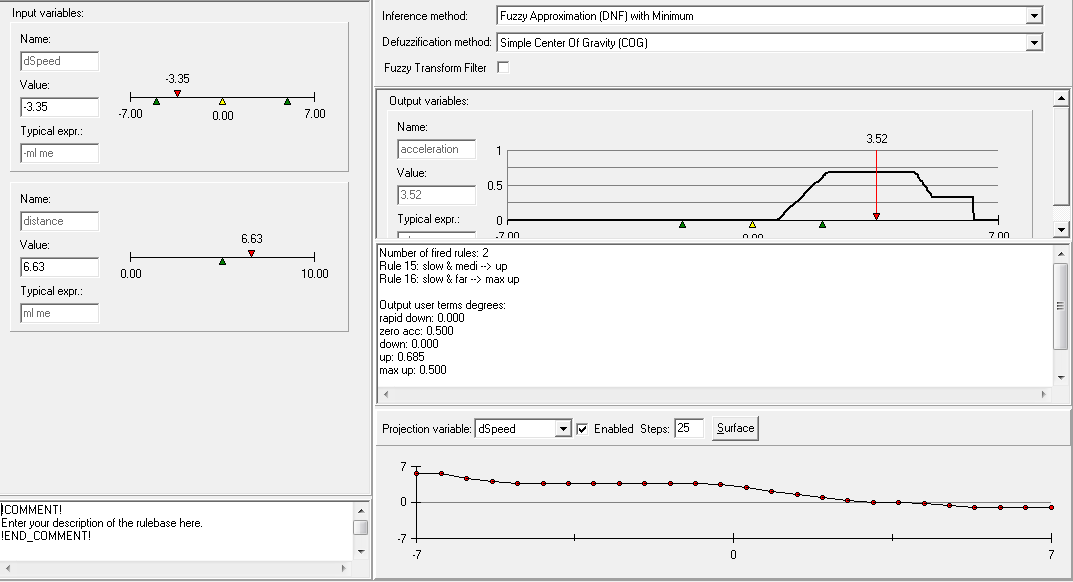
* 1. Implementace v aplikaci

Fuzzy logika v aplikaci je využita pro dvě základní části, a to jsou jízda vozidel za sebou a rozhodování, zda dát přednost v jízdě či nikoliv.

K vytvoření dané logiky byl použit nástroj LFLC, který nabízí mnoho možností při práci s fuzzy logikou.

* + 1. LFLController

Linguistic Fuzzy Logic Controller je specializovaný nástroj založen na teorii fuzzy logiky. Slouží pro práci s fuzzy logikou, modelování a testování pravidel.



3 Testování fuzzy logiky

Nástroj byl vytvořen na Ostravské univerzitě, v institutu pro výzkum a aplikaci fuzzy modelování – IRAFM. Více informací o tomto nástroji a možnost stažení naleznete na webové stránce <http://irafm.osu.cz/en/c101_0/>.

* + 1. Použití dat z LFLC v aplikaci

Pro modelování fuzzy logiky, jak je výše zmíněno, byla použita aplikace LFLC. Vytvořený model lze uložit do souboru s koncovkou \*.rb, který je možno zobrazit v klasických textových editorech, díky čemuž jej lze jednoduše použít i v jiných aplikacích.

Data uložená v tomto souboru určují všechny parametry modelované logiky. Pro vlastní použití jsou ovšem nejdůležitější data, která určují vstupní a výstupní proměnné a také pravidla, která tyto proměnné používají.

Formát uložení proměnných v souboru lze vidět na obrázku č. 4, seznam pravidel dále na obrázku č. 5.



4 Příklad struktury proměnné



5 Příklad fuzzy pravidel

O načtení dat do aplikace se stará třída RuleBaseReader, kde jsou data načítány jednoduchým procházením řádků souboru. Díky neměnné struktuře dat je poté podle klíčových slov rozpoznáno, o které parametry modelu se jedná.

Jakmile jsou data načtena v aplikaci, je vytvořen objekt třídy RuleBaseCalculator , který tvoří celou logiku, tedy výše zmíněnou aproximaci DNF a defuzzyfikacie COG. Parametry této třídy jsou výstupní fuzzy proměnná, a fuzzy pravidla, jež byly již dříve načteny třídou RuleBaseReader.

Požadovaná hodnota se z vytvořeného objektu získá zavoláním metody „calculateByValues“ s parametrem všech vstupních hodnot. Jakmile je tato metoda použita, vypočte se výsledná hodnota výpočtem defuzzyfikace COG

* + 1. Pohyb vozidel za sebou

Jak již název napovídá, jedna z vytvořených logik se stará o určování rychlosti jízdy jedoucího vozidla, které jede za vozidlem druhým. Vozidlo přizpůsobuje svou rychlost podle dané situace, pokud hrozí kolize vozidlo začne brzdit, naopak pokud je cesta volná, může jet rychlostí stanovenou parametry daného vozidla.

Rychlost vozidla je měněna zrychlením, což je výsledek výpočtu. Zrychlení může být kladné, při kterém vozidlo zrychluje, až záporné, kdy naopak zpomaluje. Pokud je zrychlení rovno nule, vozidlo rychlost nemění. Vstupními hodnotami jsou vzdálenost od druhého vozidla a rozdíl jejich rychlostí.



6 Vstupní proměnná vzdálenost



7 Vstupní proměnná rychlost



8 Výstupní fuzzy proměnná zrychlení

Tento způsob určení rychlosti je také použit při zastavení před křižovatkou, pouze s úpravou, kde místo vozidla jedoucího vepředu je „pevný bod“, který nemá žádnou rychlost.



Pravidla pro jízdu za sebou

* + 1. Rozhodnutí udělení přednosti

Druhá vytvořená logika zajišťuje rozhodování, zda při jízdě přes křižovatku vozidlo má zastavit, nebo stihne křižovatkou projet, aniž by došlo ke kolizi.

Ve srovnání s předchozí logikou je tato složitější, neboť pracuje se čtyřmi vstupními proměnnými. Parametry jsou podobné, tedy vzdálenost od křižovatky a rychlost vozidla, avšak dvojnásobné, neboť musí být zvlášť pro vozidlo které je na hlavní silnici i pro vozidlo jedoucí po silnici vedlejší. Výsledkem je naopak pouze hodnota 0 nebo 1, tedy zda jet, nebo ne.



10 Výstup určení přednosti



Pravidla pro určení přednosti

* + 1. Rozdíl obyčejný/ fuzzy

ZÁVĚR

RESUMÉ

SUMMARY

SEZNAM POUŽITÉ LITERATURY

1. **Příjmení, Jméno.** *Název knihy.* Město vydání: Vydavatelství, 2003. 123-4-56-789123-4.

2. **Příjmení1, Jméno1 a Příjmení2, Jméno2.** Název webové stránky. *Název webu.* [Online] Produkční společnost, 23. Září 2006. [Citace: 19. Září 2008.] http://www.urladresa.cz. 12-3456-789-12.

SEZNAM POUŽITÝCH SYMBOLŮ

|  |  |  |
| --- | --- | --- |
| ABC |  | Význam první zkratky. |
| B |  | Význam druhé zkratky. |
| C |  | Význam třetí zkratky. |
|  |  |  |

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH