

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

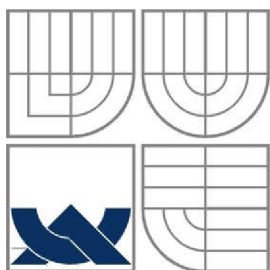
3D AUTOŠKOLA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

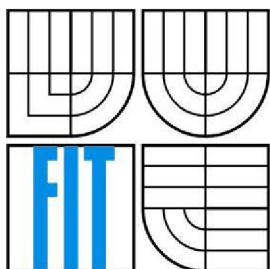
AUTOR PRÁCE
AUTHOR

Bc. Lukáš Pernica

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

3D AUTOŠKOLA

3D DRIVING SCHOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Lukáš Pernica

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Peter Chudý, Ph.D. MBA

BRNO 2009

Zadání diplomové práce

Řešitel: **Pernica Lukáš, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **3D Autoškola**

Kategorie: Počítačová grafika

Pokyny:

1. Nastudujte problematiku simulace dopravního systému a kontroly dodržování pravidel v simulaci.
2. Vytvořte architekturu systému simulátoru a model uživatelského rozhraní.
3. Vytvořte (případně zajistěte z vhodných zdrojů) základní 3D modely a funkční 3D engine, které budou použity v simulátoru.
4. V simulátoru implementujte základní fyzikální model, sledování dodržování prioritních pravidel a vhodné uživatelské rozhraní.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu.

Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1, 2 a část 5.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Chudý Peter, Ing., Ph.D. MBA, UPGM FIT VUT**

Datum zadání: 22. září 2008

Datum odevzdání: 26. května 2009

L.S.



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá popisem a návrhem simulátoru 3D autoškoly. V práci je shrnut základní popis některých dostupných simulátorů a možnosti jejich využití. Na základě podrobné studie dostupných simulátorů je navržen nový model simulátoru. V práci je popsána problematika simulace dopravního systému, kontroly dodržování pravidel v silničním provozu a také navržen nový model uživatelského rozhraní. Nakonec je popsán celý návrh a postup při implementaci simulátoru.

Klíčová slova

Simulátor, autoškola, model dopravního systému, pravidla silničního provozu, uživatelské rozhraní, detekce kolizí, grafický engine, fyzikální engine.

Abstract

This study was created to describe and to design the 3D simulator for a driving schools. In the study there are gathered the basic definitions of available simulators and possibilities of their application in real life. This new simulator is designed on a basis of detailed study of available simulators. In study are described the issues with simulation of traffic system, monitoring compliance with the traffic rules as well as a new user interface model. The whole concept and procedure during implementation of the simulator are described at the end of the work.

Keywords

Simulator, Driving School, traffic model, traffic rules, user interface, collision detection, graphics engine, physics engine

Citace

Pernica Lukáš: 3D Autoškola. Brno, 2009, diplomová práce, FIT VUT v Brně.

3D Autoškola

Prohlášení

Prohlašuji, že jsem tento diplomový projekt vypracoval samostatně pod vedením Ing. Petra Chudého, Ph.D. MBA.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Pernica
22.5.2009

Poděkování

Rád bych tímto poděkoval svému vedoucímu diplomové práce panu Ing. Petru Chudému Ph.D. MBA nejen za jeho vedení a cenné rady, ale také za jeho připomínky a podnětné návrhy, kterými mě vedl a podporoval v průběhu celé práce. Svým kolegům spolužákům děkuji za konzultace a rady, kterými mi byli nápomocní při řešení problémů. Zvláště bych chtěl poděkovat panu Miroslavu Ondrušovi ze společnosti „theasyco“, který mi poskytl konzultace a cenné rady. Dále děkuji své rodině za podporu a poskytnuté zázemí při tvorbě práce a svým blízkým, kteří mě v práci podporovali.

© Bc. Lukáš Pernica, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod	4
2 Modelování a simulace	6
2.1 Princip modelování a simulace	6
2.2 Oblasti použití simulace	7
2.3 Výhody simulačních metod	7
2.4 Problémy simulačních metod	8
2.5 Klasifikace modelů	9
3 Simulátory v dopravě	10
3.1 Simulátory a animace	10
3.2 Simulátory na vyšší úrovni realističnosti	10
3.3 Simulátory na střední úrovni realističnosti	14
3.3.1 3D Fahrschule	14
3.3.2 Další aplikace	17
3.4 Simulátory s nižší úrovní sofistikace	17
4 Analýza problematiky	19
4.1 Analýza aplikace 3D Fahrschule	19
4.1.1 Silné stránky	19
4.1.2 Slabé stránky	21
4.1.3 Hodnocení nezávislých uživatelů	23
4.1.4 SWOT analýza	23
5 Návrh nové aplikace	25
5.1 Rozhlížení	25
5.2 Ovládání motocyklu	25
5.3 Zobrazení zpětných zrcátek	25
5.4 Uživatelské rozhraní – ovládání	26
5.5 Menu s ovládáním	26
5.6 Výhled z automobilu	26
5.7 Návrh nového ovládání	26
5.7.1 Myš	26
5.7.2 Klávesnice	27
6 Problematika simulace dopravního systému	28
6.1 Grafy	28
6.2 Model dopravní infrastruktury	29

6.2.1	Ulice a jízdní pruhy	30
6.2.2	Značky	30
6.2.3	Křižovatky	31
6.3	Kontrola dodržování pravidel v simulaci.....	33
6.3.1	Detekce kolizí	33
6.3.2	Dodržování pravidel.....	34
7	Architektura systému simulátoru.....	35
7.1	Grafický engine.....	35
7.1.1	OGRE.....	35
7.1.2	CrystalSpace	36
7.2	Fyzikální engine	36
7.2.1	Bullet	36
7.2.2	ODE.....	36
7.3	Komponenty simulátoru	37
7.4	Popis funkčnosti	38
7.5	Model uživatelského rozhraní.....	39
7.5.1	Uživatelské rozhraní.....	39
7.5.2	Myš.....	41
7.5.3	Klávesnice	41
7.5.4	Tabulka s ovládacími funkcemi simulátoru.....	42
8	Implementace simulátoru	43
8.1	3D modely v simulátoru	44
8.1.1	Načtení modelů v enginu OGRE.....	46
8.1.2	Model vozidla	46
8.2	Pohyb pohledu řidiče.....	47
8.3	Fyzika	48
8.3.1	Kolize	48
8.3.2	Nastavení parametru auta	49
8.4	Menu aplikace.....	50
8.5	Model města.....	51
8.6	Graf silniční sítě	51
8.6.1	Třída „GraphNode“	52
8.6.2	Třída GraphEdge.....	52
8.6.3	Třída RoadGraph.....	53
8.7	Vytvoření pravidel silničního provozu	56
8.8	Kontrola pravidel silničního provozu při simulaci.....	57
8.8.1	Pravidla kontrolovaná na hraně.....	58

8.8.2	Kontrola pravidel v křižovatce.....	58
8.8.3	Kontrola pravidel po projetí křižovatkou	59
8.9	Umístění dopravních značek.....	60
9	Instalační balík aplikace.....	61
10	Závěr.....	62
	Literatura	64
	Seznam obrázků.....	66
	Seznam příloh	67

1 Úvod

Tato práce se zabývá problematikou simulátoru 3D autoškoly. Simulace, ať již v oblasti informačních technologií, nebo jiných oblastech lidské činnosti, mají v dnešní době (2009) své přední místo. Je to žádaný obor v oblasti počítačových systémů. Na základě počítačových simulací získáváme výsledky, které popisují zkoumanou problematiku, nebo slouží jako vstupy jiným simulacím. Počítačové simulace se využívají v mnoha odvětvích lidské činnosti a přispívají k efektivnímu plánování, vývoji a návrhu nejrůznějších aplikací. Jejich výhodou je bezpečné testování a simulování nebezpečných systémů, které by při reálných testech mohly mít nedozírné následky (testy jaderných výbuchů), nebo je jejich realizace příliš nákladná, či s dnešními prostředky nerealizovatelná (testy ve vesmíru). Simulátor autoškoly přispívá velkou mírou k bezpečnosti, protože nezkušený řidič si může nejen zdokonalit své znalosti, aniž by potřeboval vedení učitele autoškoly a reálné vozidlo, ale nemůže ohrozit sebe ani jiné účastníky silničního provozu. Zkušenosti v silniční dopravě získá v prostředí, které je přímo zaměřené na výuku pravidel silničního provozu. Není zde rušen okolními vlivy, které mohou být pro začínající řidiče nepřijemné.

Technika simulačních zařízení a simulátorů je velice rozsáhlá a tato práce nepokrývá celou její problematiku. Práce se zaměřuje spíše na postup při vývoji simulátoru. První kapitola se zabývá obecnými technikami a popisem průběhu simulačního procesu. Jsou zde shrnuty základní myšlenky a postupy, jak správně přistupovat k modelování a simulacím.

Simulátor autoškoly není žádnou novinkou a v praxi se již řadu let využívají jak softwarové aplikace na počítačích, tak i složitější modely celých automobilů. Při návrhu nové aplikace je dobré si ujasnit cílovou skupinu, pro kterou bude simulátor vyvíjen a zhodnotit dostupné možnosti na trhu. V druhé kapitole této práce je popsáno několik simulátorů autoškol. Jsou rozděleny na tři kategorie, podle typu využití. Simulátory, které využívají profesionální autoškoly, trenažéry v armádě, či výzkumných centrech, mají velice dobře propracovaný software, který je propojen s modelem kabiny reálného vozidla, či simulačními místnostmi. Tato práce má za cíl vytvořit simulátor, který bude možné využívat na běžných osobních počítačích, bez dalších specializovaných prostředků. Z dostupných aplikací na trhu byla vybrána jedna aplikace, která se dá zařadit právě do této kategorie. Na této aplikaci byla provedena podrobná analýza. Na jejím základě jsou vyhodnoceny klady a zápory a shrnuty základní vlastnosti, které jsou důležité v dané kategorii simulátorů. Za využití SWOT analýzy [1] je popsána problematika zkoumaného programu.

Pátá kapitola pojednává o návrhu nové aplikace. Je úzce spjata s předchozí kapitolou, především analýzou a návrhem na základě SWOT analýzy [1]. Silné stránky, vyhodnocené v kapitole 4, jsou dobrým základem pro návrh nového simulátoru. Slabé stránky z kapitoly 4 jsou podnětem pro další možnosti a zlepšení. Nová aplikace by měla být přínosem a poskytovat další možnosti a myšlenky v oblasti simulátorů autoškol.

Problematika dopravních systémů, pravidel silničního provozu a možnosti jejich modelování jsou rozebrány v kapitole 6. Reprezentace pravidel a dopravní síť zasahuje i do oblasti geografických systémů, kde se mohou využívat orientované grafy.

Tato práce si klade za cíl vytvořit základ pro simulátor autoškoly, aby mohl být dále rozšiřován a mohlo se pokračovat v jeho vývoji. Za tímto účelem je nutné vytvořit jádro celého systému. Pro tyto možnosti existuje řada knihoven, na jejichž základě je možné vytvářet rozsáhlé projekty. V sedmé kapitole je rozbor vhodných knihoven, jak pro grafickou část, tak i pro fyziku. Bylo využito knihoven OGRE pro grafickou část a knihovny Bullet pro část fyzikální.

Kapitola 8 popisuje již konkrétní postup a využití techniky při vývoji. Jsou popsány metody využití v jednotlivých modulech a jejich vzájemné propojení. Dále navrhnuté třídy a popis využitých tříd z připojených knihoven.

Poslední devátá kapitola krátce popisuje vytvoření instalačního balíku, který je nutný pro přenositelnost mezi počítači.

2 Modelování a simulace

S ohledem na cíl práce, kterým je simulátor 3D autoškoly, budou v této kapitole uvedeny základní principy, které se využívají v oblasti modelování a simulací, jejich problémy a cíle, se kterými se v této oblasti můžeme setkat.

- V modelování a simulacích je nutné definovat základní pojmy [2,3]:
- Systém – definuje soubor elementárních částí, které mají mezi sebou určité vazby a podle existence se systémy dělí na:
 - o reálné systémy – existující,
 - o nereálné systémy – fiktivní, neexistující, používají se v počítačových hrách.
- Model – je to napodobenina systému jiným systémem. V našem případě to je počítačovým systémem. Model systému musí napodobovat všechny důležité a podstatné vlastnosti. Která vlastnost je důležitá, záleží především na našich požadavcích a představách, které máme na modelovaný systém.
- Modelování – je to proces, při kterém vytváříme model na základě znalostí, které máme, či získáme. Tento proces je náročný a vyžaduje mnohdy znalosti z více oborů. Od kvality modelu se pak odvíjí i výsledky, které získáváme simulací.
- Simulace je metoda získávání nových znalostí o systému experimentováním s jeho modelem. Pro účely simulace musí být model popsán odpovídajícím způsobem, protože ne každý model je pro simulaci vhodný. Tato práce se zaměří především na počítačové simulace, i když samozřejmě existují i jiné druhy simulací.

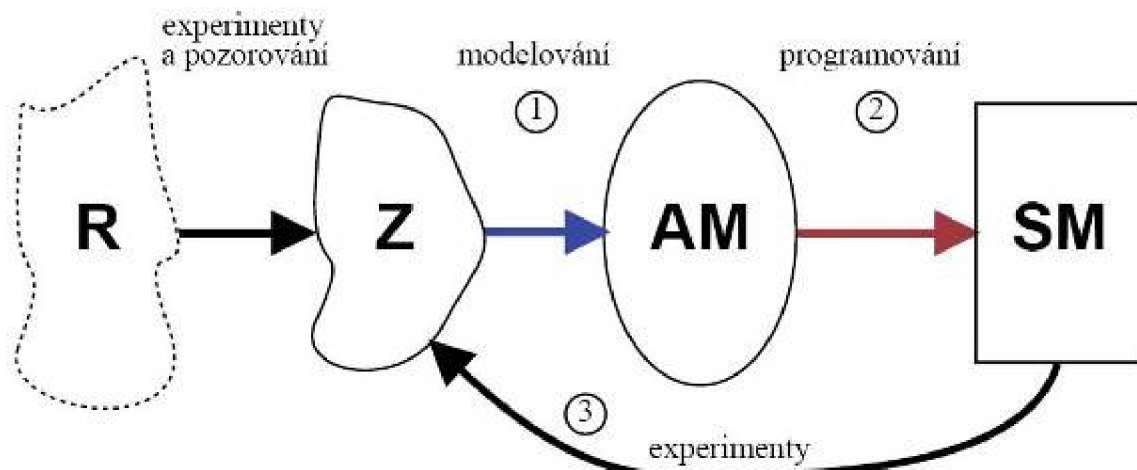
2.1 Princip modelování a simulace

Cílem simulace je získání nových znalostí o zkoumaném systému. Abychom mohli provádět simulaci, je nutné vytvořit vhodný model zkoumaného systému. [3,4]

Postup:

- Nejdříve vytvoříme tzv. abstraktní model, který nezahrnuje všechny naše znalosti o modelovaném systému, ale vybíráme jen ty vlastnosti, které jsou pro naše účely podstatné. Tímto postupem vznikne abstraktní model, který nám dává představu, jak by měl systém vypadat.
- Následně vytvoříme simulační model. Vytváříme ho na základě abstraktního modelu. Simulační model již nic nezjednodušuje a od abstraktního modelu se liší tím, že se simulačním modelem můžeme provádět simulace. Je to spustitelný program, kterému můžeme zadávat na vstupu, parametry a případně ho řídit v jeho průběhu.
- Simulace, které provádíme, nám dávají výsledky. Tyto výsledky jsou informace o chování systému, ze kterých jejich zobecněním získáváme nové znalosti.

V následujícím obrázku je postup znázorněn pomocí symbolů. R – popisuje reálný svět, reálný systém, Z – jsou naše znalosti o realitě, AM – je abstraktní model, který vytvoříme z části našich znalostí, které jsou podstatné pro model, SM – je simulační model, který naprogramujeme z abstraktního modelu.[4]



Obrázek 2.1: Schéma fází modelů při simulacích [4]

2.2 Oblasti použití simulace

V následujícím přehledu jsou shrnuty některé oblasti a příklady, kde se simulační techniky využívají.

- Fyzika: model jaderného reaktoru, model šíření zvuku v místnosti.
- Technika obecně: simulované crash testy automobilů, model mikroprocesoru a elektrických obvodů.
- Doprava: model dopravní situace ve městě.
- Výuka: demonstrační modely, hry (např. simulátor letadla, automobilu).
- Film, počítačové hry: modely pro vizuální efekty. [6]

Z uvedených příkladů vidíme, že použití je rozsáhlé a zasahuje prakticky do všech oblastí. Proto je obvykle při vytváření složitých simulačních modelů nutná spolupráce odborníků z různých oborů. [4]

2.3 Výhody simulačních metod

Obecně můžeme za hlavní výhody simulace v porovnání s reálnými experimenty považovat nižší cenu, většinou i menší časovou náročnost experimentování a naprostou bezpečnost při provádění experimentů. [4]

- Cena je důležité kritérium pro nasazení simulace. Experimenty s reálným systémem mohou být drahé – například crash testy automobilů se dnes provádějí pouze pro validaci výsledků simulačních experimentů a proto je spotřeba automobilů při těchto testech výrazně menší.
- Rychlost experimentování je dalším důležitým faktorem pro simulaci. Simulaci můžeme zrychlovat a zpomalovat podle potřeby (zrychlování je samozřejmě omezeno výkonem našeho počítače). Například růst rostlin je velmi pomalý proces, který lze při simulaci

výrazně urychlit a dosáhnout výsledků podstatně dříve. Naopak pro sledování velmi rychlých dějů v reálných systémech je třeba drahé experimentální vybavení (nebo dokonce nelze tyto jevy sledovat vůbec), kdežto u simulačního modelu můžeme tyto děje libovolně zpomalit. Příkladem může být sledování pohybu atomů v nanotechnologických systémech.

- Bezpečnost je u některých reálných experimentů velký problém, často nelze tyto experimenty z bezpečnostních důvodů vůbec provádět. Příkladem mohou být některé jaderné reakce, šíření epidemií a podobně.
- Simulace dovoluje experimentovat s modely velmi složitých systémů – jsme omezeni pouze výkonem našeho počítače. Reálné experimenty by u tak složitých systémů byly neproveditelné.
- Někdy je simulace jediný způsob jak experimentovat – například v astronomii můžeme simulovat srážky galaxií a podobné situace. Protože výkon počítačů neustále roste, je ekonomicky výhodnější, rychlejší a často jediné možné experimentovat na modelech, než na originálech.

2.4 Problémy simulačních metod

Simulace na počítačích má vedle výhod také závažné nevýhody, se kterými musíme počítat a snažit se je minimalizovat. [2,6]

- Problém validity modelu je nejdůležitější, protože chybný model dává při simulaci chybné výsledky, které mohou při aplikaci způsobit katastrofální následky. Ověřování validity modelu je velmi náročné a mělo by být provedeno ještě před prováděním experimentů, jejichž výsledky chceme použít.
- Náročnost na výpočetní výkon počítačů je obecný problém všech netriviálních simulačních modelů. Většina dnešních superpočítačů je používána právě pro simulace. Vzhledem k neustálému růstu výkonu procesorů je dnes možné provádět relativně náročné simulace i na běžných osobních počítačích.
- Simulací získáváme konkrétní numerické výsledky pro daný experiment. Pokud potřebujeme například zjistit vliv změněných parametrů modelu, musíme celou simulaci opakovat, což může být časově náročné.
- Další nevýhodou je nepřesnost a nestabilita numerického řešení některých modelů. Tyto negativní vlastnosti numerických metod mohou zcela znehodnotit výsledky i v případě ověřeného a jinak bezchybného modelu. Proto je nutné znát vlastnosti použitých numerických metod a používat jen ty, které mají pro daný model vyhovující přesnost.
- Někdy je problémem i vysoká náročnost vytváření modelů — vytvořit kvalitní model mikroprocesoru může být stejně náročné jako navrhovat samotný procesor. Proto se používají modelovací jazyky, které jsou vhodné jak pro návrh, tak i pro simulaci logických obvodů. Obdobně se postupuje i v jiných oblastech – např. ve strojírenství, kde jsou CAD systémy často doplněny i simulačními nástroji.

2.5 Klasifikace modelů

Přesné rozdělení modelů do kategorií je poměrně náročné, protože často neexistují jednoznačná klasifikační kritéria. Modely lze dělit například podle způsobu jejich matematického popisu, podle úrovně abstrakce, podle metod implementace (paralelní, distribuované) a podle celé řady dalších kritérií.

Tradičně dělíme modely na:

- Modely spojité – proměnné modelu mění svůj stav spojitě. Tyto modely jsou popsatelné například diferenciálními rovnicemi.
- Modely diskrétní – stav modelu se mění skokově v diskrétních časových okamžicích. Příkladem takového modelu může být například konečný automat.
- Modely kombinované – obsahují spojitě i diskrétní prvky současně v jednom modelu.

V dalším textu se budu zabývat simulátory, které se řadí do kategorie spojitých simulací. Simulace 3D autoškoly, provozu na komunikacích se v čase mění a uživatel simulace zasahuje po celou dobu do jejího průběhu. Na základě interakce aplikace s uživatelem vzniká vždy nová simulace s jinými výsledky. Výsledky nejsou dostupné až na konci simulace, jak je u některých simulací běžné, ale již v jejím průběhu a uživatel na každý výsledek opět reaguje a dává tak systému nové vstupní hodnoty a parametry. [3]

3 Simulátory v dopravě

Ve světě počítačových technologií se stále více přibližujeme k modelování a zobrazování reálného světa. Stále novější technologie nám denně pomáhají v mnoha odvětvích lidské činnosti a neustále se dostávají do nových oblastí. Některé systémy mohou lidskou práci ulehčovat, mohou zastávat automatizované úkony a nahradit tak práci člověka.

Důležitou součástí je doprava a dopravní infrastruktura. Na trhu je v dnešní době (2009) mnoho systémů, které se zabývají simulací provozu, jízdou autem či na motocyklu, letecké výcvikové simulátory, trenažéry apod. Tyto systémy mohou mít různé využití. Zábavou počínaje, přes řízení a navrhování dopravních situací v reálném prostředí až k procesu výuky. Právě poslední zmíněná kategorie, tedy výuka, je pro tento text podstatná, protože simulátor autoškoly je již svou podstatou výuková aplikace.

3.1 Simulátory a animace

Pro potřeby výuky se dá uvažovat o dvou termínech: simulace a animace. Každý přístup má svoje pro a proti a v každé oblasti musíme nejprve zvážit, kterým směrem se vlastně chceme zabírat a k jakému účelu má být naše dílo využíváno. [5]

Simulace:

- nelze skriptovat,
- neomezené množství cest (můžete jet kamkoliv - výběr možností),
- interaktivní,
- real-Time,
- realistické (fyzika zpětná odezva).

Animace:

- plně skriptovatelná,
- jedna vizuální cesta,
- žádná interakce (video),
- není Real-Time,
- nejednoznačné – nerealistické (animátor „myslí“, kde by měla být fyzikální odezva).

Pro aplikaci autoškoly jako výukového programu je tedy zřejmé, že simulátor je vhodnější. Animace a videa se v autoškolách využívají v hojně míře jako názorná demonstrační pomůcka. Simulátor je však mnohem účelnější, protože dovolí účastníkům si nanečisto vyzkoušet situaci a sám do ní zasahovat.

Následující dělení se zaměří na tři kategorie simulátorů.

3.2 Simulátory na vyšší úrovni realističnosti

Simulátory na vyšší úrovni realističnosti, jak již z názvu vyplývá, modelují a zobrazují realistické situace. V dopravě to mohou být právě simulace řízení silniční, železniční nebo letecké dopravy, které

slouží pro školení pracovníků, či zjišťování možných důsledků činnosti modelovaného objektu. Simulace pomáhají plánovat či zobrazovat systémy, které mají být vytvořeny. Tímto přístupem se mohou předem zjišťovat nedostatky, prohlížet možné důsledky a na základě výsledků získaných simulacemi provádět změny. Simulátory mohou také zobrazovat situaci, která je obrazem reálného světa. Úkony prováděné v takovýchto simulátorech jsou bezpečnější a neohrožují reálné prostředí. Takovými aplikacemi jsou právě automobilové či letecké simulátory. Uživatel simulátoru nemůže nikoho ohrozit, protože se pohybuje pouze ve virtuálním modelu prostředí.

Automobilové simulační systémy jsou využívány především v autoškolách. Mají vysoce propracovaný software aplikace, ale také hardware s reálným uživatelským rozhraním. Jde o výukové systémy, které mají za cíl naučit uživatele, tedy žáka autoškoly, s pomocí instruktora jízdu autem ještě dříve, než do opravdového auta usednou. Důraz je především kladen na realistické ovládání, chování automobilu a samozřejmě na dopravní předpisy v samotné simulaci. Simulátor je zabudován do „interiéru“ auta. Kokpit obsahuje volant, řadící páku, pedály a interiér se všemi stavovými indikátory, stejně jako je možné nalézt v reálném autě. U propracovaných trenažerů nechybí detaily jako např. zpětná zrcátka a funkční světelná a zvuková signalizace.

Příkladem takového trenažeru je např. systém vyvíjený firmou JKZ spol. s.r.o [7]. Ta se zabývá především realizací mechanických částí systému. Software je pak dodáván firmou Presagis [8]. Simulátor této firmy, jak je vidět na obrázku, je sestaven z přístrojové desky automobilu, která je napojena na senzory snímající ovládací prvky a vše komunikuje se softwarovou simulační částí. Zajímavým prvkem jsou i zpětná zrcátka. Obraz dopravní situace za trenažerem je promítán na dvou zobrazovacích jednotkách, které jsou umístěny opačně a jejich obrazy jsou vidět v klasických zrcátkách. Celá simulovaná dopravní situace je promítána na plátno před simulátorem.



Obrázek 3.1: Simulátor autoškoly firmy Presagis [7]

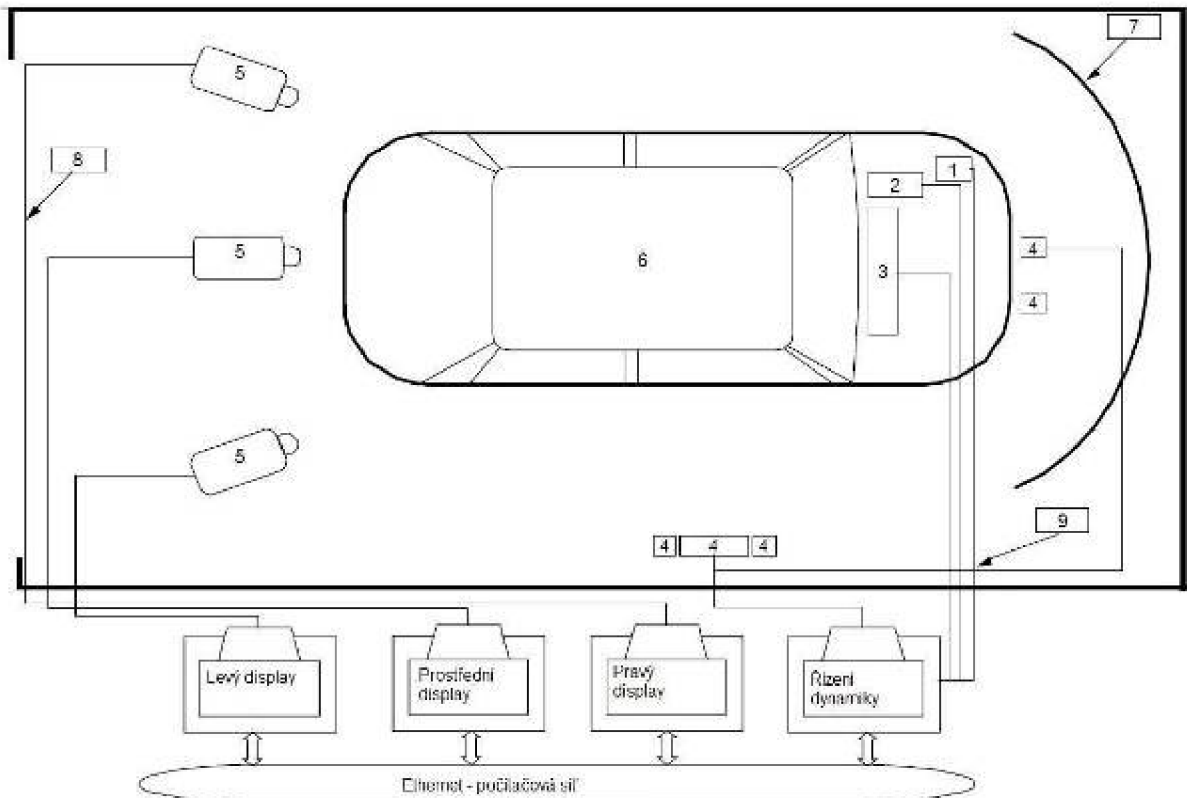
Popsaný systém není jediný od zmíněné firmy [7]. Firma vyrábí i další simulátory např. pro autobusy a nákladní vozidla. Firem zabývajících se vývojem a tvorbou trenažerů je samozřejmě více, ale méně firma představuje výrobce jednoho z nejpoužívanějších simulátorů v českých autoškolách. Protože jsou tyto systémy nákladné, nejsou masově rozšířené. Jinými slovy, takové systémy používají pouze velké autoškoly.

Jak bylo uvedeno, technologie trenažeru se může v mnohém zlepšovat. Dnešní technologie umožňují mnohem propracovanější techniky. Jako budoucí technologie simulátorů se jistě nabízí

technologie 3D brýlí a využití virtuální reality. Na takovýchto systémech se v zahraničí již pracuje ať již na výzkumné či profesionální úrovni.

Simulační systémy v zahraničí jsou mnohem vyspělejší. Rozsáhlý výzkum a vývoj probíhá v USA a také v Německu. [8,9]

Simulátory jsou napojeny na počítačovou síť, která řídí celou simulaci. Jednotlivé části jsou ovládány speciálním systémem a všechny systémy spolu navzájem spolupracují. Jako příklad můžeme uvést simulátor z univerzity Georgie Washingtona [9]. Jako většina propracovaných systému mají promítání situace ve velkém úhlu. Promítání simulace o úhlu 135° je realizováno třemi projektory.



Obrázek 3.2: Schéma komplexního simulátoru [9]

Vysvětlivky

- 1 – senzor brzd
- 2 – senzor akcelerace
- 3 – řízení
- 4 – zvukový systém
- 5 – Tři projektory zobrazující scénu
- 6 – Reálná automobilová kabína
- 7 – Plátno zobrazující 135° úhel obrazu
- 8 – VGA kabely
- 9 – 9-pin sériový konektor pro dekodér a 39-pin konektor

Řidič je posazen do reálného automobilu, který je napojen na čidla, která převádějí signály k řídicím počítačům. Pro představu je na následujícím obrázku (3.3) vidět, že v reálné karoserii

automobilu nejsou tradiční prvky jako motor, hřídele, atd., ale na místa těchto částí jsou napojeny senzory a čidla.



Obrázek 3.3: Detail napojení uživatelského rozhraní simulátoru na senzory [9]

Další pokročilou technologií je hydraulický mechanismus, který pohybuje s celou kabinou. Při simulaci jízdy v automobilu není třeba velkých výchylek v pohybu a nepočítáme-li náraz, nejsou potřeba prudké pohyby. Avšak náklony v zatáčkách, pohyb vpřed a vzad při brzdění a akceleraci dodávají simulaci realistickou složku, která přispívá k celkově autentičtějšímu dojmu. Řidič v simulátoru pozná nerovnosti na vozovce, přejetí obrubníku, či jinou překážku a podvědomě uzpůsobí styl jízdy, což bez systému hydrauliky nemůže být dosaženo.

Na následujících obrázcích jsou ukázky takovýchto hydraulických systémů, které jsou používány například na „Wuerzburg Institute for Traffic Science“ [10] nebo na „University of Central Florida - Center for Advanced Transportation Systems Simulation“ [5].



Obrázek 3.4: Příklady simulátorů vysoké úrovně realističnosti [5,9]

Další systémy pro simulaci jízdy a provozu jsou vyvíjeny a používány velkými firmami jako je např. IBM, či Honeywell a zabývají se například simulací situací v dopravě. Vyhodnocováním situací

získávají data, na jejichž základě pak navrhují a vyrábí nejrůznější dopravní zařízení, jako semaforey, systémy řízení semaforů na křižovatkách, zabezpečení železničních přejezdů apod.

Simulátory na vyšší úrovni realističnosti se zabývají velké profesionální firmy. Odpovídá tomu také vysoká kvalita zpracování.

3.3 Simulátory na střední úrovni realističnosti

Do této kategorie jsou zařazeny aplikace, které mají sloužit pro běžné uživatele. Důležitým aspektem je dostupnost a nenáročnost v takové míře, aby si je každý mohl spustit na PC či notebooku. Zároveň ale mají patřičnou úroveň zpracovanosti. Je zahrnut fyzikální model chování, kvalitní model grafiky a v neposlední řadě i pravidla silničního provozu, pokud se jedná o simulátor autoškol. Především sem patří aplikace, které nemají preferovanou pouze jednu zájmovou oblast a jsou tedy spojením například realističnosti chování auta a silničních předpisů. Ovládání by mělo být umožněno na jakémkoliv počítači, což znamená pomocí klávesnice a myši, případně může být obohaceno o ovládání na volantu, nebo joysticku.

Asi nejvíce zpracovanou a nejvíce známou aplikací je simulátor „3D Fahrschule“.

3.3.1 3D Fahrschule

Tento program je známou 3D autoškolou. Je vyvíjen německou firmou „Besier 3D-Edutainment Wiesbaden“ [11] již řadu let a v současné době (2009) je na trhu nejnovější verze 5.1. Program prochází stále řadou vylepšení a jsou dostupné i aktualizace na starší verze. Program je zaměřen na výuku jízdy autem nebo na motorce, dopravních předpisů a osvojení si řídičských dovedností, ještě před tím, než půjde uživatel tohoto programu do autoškoly.

V úvodním nastavení jsou volitelné tři módy obtížnosti. „Rookie“ je nejjednodušší úroveň. Jsou zde hlídány pouze velké přestupky, jako je jízda na červenou apod. Druhou úrovní je „Beginner“, kde se již zobrazují i chyby typu nedání znamení o změně směru jízdy a podobné. V módu „Normal“ je již každý přestupek penalizován. Jedná se o úroveň, která odpovídá obtížnosti závěrečné zkoušky v klasické autoškolě. Řidič musí užívat zrcátka, musí se rozhlédnout na křižovatce, apod.

Program nabízí výběr mezi třemi auty. BMW 318i, Jeep a Classic car. Tato auta mají podobné chování. Po stránce fyzikálního modelu je stejně jako ve skutečnosti BMW rychlejší. Tedy má rychlejší akceleraci i brzdění. V jiných aspektech se tyto vozy od sebe o mnoho neliší. Z pohledu výukového je v BMW otáčkoměr, přispívající k osvojení správného řazení. Na palubní desce Jeepu chybí a v Classic car je špatně čitelný. Na palubní desce je funkční ukazatel rychlosti, který je dobré sledovat, protože při překročení rychlosti se započítávají trestné body za rychlou jízdu. Dále jsou zde ještě kontrolky směrovek. Více funkčních prvků a detailů není v modelu tohoto vozu zahrnuto. Jde zapnout i pohled třetí osoby, kde jsou potom tyto ukazatele vyznačeny vedle. Tento pohled však již není tak dobře názorný pro účely autoškoly, ale blíží se do oblasti her.



Obrázek 3.5: Interiér automobilu v aplikaci 3D Fahrschule [12]

Je také možné si zvolit 3 druhy motocyklů. Honda Hornet, scooter a chopper. Chování všech tří motocyklů by se dalo popsat jako stejné. Opět největší rozdíl je v akceleraci a rychlosti jakou se motocykly pohybují. Po fyzikální stránce nejsou příliš dobře propracované. Motocykly zatáčejí příliš trhaně a způsob ovládání na PC není přirozený. Z pohledu chování není ovládání motocyklu realistické. I v malých rychlostech motocykl neztrácí stabilitu, nelze spadnout na zem a nedochází k přetočení v důsledku smyku. Přínos pro skutečný provoz je tedy pouze ve zvládnutí dopravních značek a pravidel. Z pohledu přístrojové desky je zde ukazatel rychlosti a kontrolky ukazatelů směru. U Hondy je zastoupen i otáčkoměr. U motocyklů je také kontrolka zařazeného neutrálu. Zařazení neutrálu jde pouze použitím klávesy pro ruční brzdu. To však není z pohledu ovládání motocyklu korektní. Vhodnější by bylo, kdyby byl neutrální stejné jako v reálu mezi prvním a druhým rychlostním stupněm.



Obrázek 3.6: Vzhled motocyklu v aplikaci 3D Fahrschule [12]

Pro řazení rychlostních stupňů je možné si zvolit mezi automatickou převodovkou a manuálním řazením. Manuální řazení je sekvenční, pomocí dvou kláves. Jedna pro zařazení vyššího rychlostního stupně, druhá pro zařazení o stupeň níže. Tento přístup je pro motocykly zcela správný. Pro automobil by se dalo uvažovat i o jiných možnostech, protože v automobilu jde zařadit

z jakéhokoliv místa, kterýkoliv rychlostní stupeň. Ale tato drobnost je spíše subjektivní z pohledu každého uživatele. Jako vhodné doplnění jak u auta, tak i motocyklu by byl ukazatel zařazeného rychlostního stupně. Pro řidiče začátečníky je tento údaj v mnohém nápomocný.

Program nabízí výběr z několika druhů evropských silničních předpisů. Rozdělení je podle země. Na výběr je Německo, Nizozemí, Belgie, Francie a Velká Británie. V každé této zemi je 21 různých cvičení. Úkoly dává instruktor autoškoly, který je zastoupen jak ve zvukové tak v textové podobě. Anglie nemá samostatná cvičení, ale je zařazena v každé zemi jako jedna situace.

Mezi cvičeními různých dopravních situací je jízda ve velkém provozu se semaforem v centru města. Dále na okraji města, kde jsou pouze dopravní značky, jízda na dálničních přivaděčích, mimo město na venkově i v historickém centru města. Také je zařazena lekce jízdy v noci.

V provozu jsou i ostatní účastníci, tedy auta, motocykly a chodci. Při jízdě se započítávají chyby, které uživatel nasbírá a podle toho je mu nácvik konkrétní situace uznán, nebo ho musí opakovat. Zejména je hlídáno projetí na červenou. Při takovémto přestupku se jízda opakuje. Trestné body jsou započítávány za překročení rychlosti, nedání přednosti chodci, nezastavení na stopce či nedání přednosti v jízdě. Na obrazovce se také objevuje v horní části upozornění na daný přestupek.

Řazení do jízdních pruhů není nijak hlídáno a řidič si může přejíždět z pruhu do pruhu, nebo odbočovat, ze kterého pruhu chce.

Zajímavým prvkem jsou zpětná zrcátka. Pohled do nich dává řidič najevo tím, že je příslušnou klávesou zapne. Ty jsou po krátkém okamžiku opět skryty. Je tak zřejmě učiněno proto, že jsou obě zrcátka zobrazena přímo ve výhledu a ne na místech, kde jsou obvyklá. Vhodnější možností by mohlo být stále zobrazení zrcátek, což by bylo více věrohodné normální situaci, avšak musela by být vyřešena kontrola, že se řidič opravdu do zpětného zrcátka podíval.

Jednotlivé lekce dopravních situací je možné opakovat, avšak nevýhodou je stále stejná situace. Pokud uživatel danou situaci již poněkolkrát opakuje, zapamatuje si, že mu např. napravo pojede auto a nalevo půjde chodec. Situace tedy nejsou náhodné, ale předem připravené. Na druhou stranu se ale autoři postarali o to, že nacvičí vždy žádoucí řidičský úkon. Pokud má dát tedy např. řidič přednost zprava, tak v každém opakování bude mít zprava auto, dokud se danou situací nenaučí řešit správně. Po absolvování celého výcviku by se dalo říci, že řidič již zná předem situaci. Neškodilo by proto dodat nějaké náhodnější situace.

Ovládání celé aplikace je nastavitelné. V základním přednastavení aplikace je vše potřebné dobře nastaveno. Vozidla se ovládají šipkami – doleva, doprava, dopředu, brzda. Je možný pohled mírně nahoru a dolů a otočení se dozadu. Při stisknutí klávese „shift“ (v základním nastavení) se pomocí šipek ovládá pohled doleva a doprava. To je nutné při rozhlížení se na křižovatkách. Pohyb je však mírně trhaný a při několikanásobném pohledu doleva a doprava vede spíše ke zmatení a nepřehlednosti situace. Klávesou „mezerník“ se zapínají zpětná zrcátka, která jsou zobrazena jen malý okamžik. Řidič tak má dát najevo, že se do zpětného zrcátka podíval. Dále je možné zvolit automatickou převodovku „A“, nebo manuální „M“ a potom se řadí klávesami „1“ a „2“. Klávesou „L“ se zapínají světla, „backspace“ slouží jako houkačka. „ctrl“ a „alt“ jsou ukazatele směru.

Samotné ovládání je nutné si osvojit a při některých úkonech může být mnoho kláves nepřehledné. Například při brzdění na křižovatce, dávání znamení o směru jízdy a pohledem na stranu. Z naučného pohledu je to však i dobré, protože si uživatel osvojí všechny tyto úkony jeden po druhém.

Grafika celé aplikace je na dobré úrovni. Z zpracování detailů je vidět, že aplikace prošla již delším vývojem a více verzemi. Pro tuto práci může sloužit v mnohém jako dobrý vzor. Méně zobrazených detailů může někdy znamenat i kvalitnější výsledek z pohledu autoškoly.

3.3.2 Další aplikace

Mezi další vznikající aplikaci je pro ukázkou zařazen projekt „Karškolka“. Je to jednoduchý program, ve kterém je zatím pouze pohyb auta se základními vlastnostmi. Má otevřený a dostupný kód, takže každý může vylepšovat a přidávat nové funkce k tomuto projektu, jak bylo i záměrem autora. [14]

Dále vzniklo několik diplomových prací, které však již svým pojetím nejsou určeny přímo jako autoškola. Jednou z nich je „Simulátor jízdy městem“. Zaměřuje se především na grafické detaily a také na fyzikální model auta. Spolupracuje s dalšími projekty, jako je „Generátor krajiny pro simulátor jízdy“. Další pracemi jsou „dopravní simulátor“ a „editor dopravních situací“. [15]

3.4 Simulátory s nižší úrovní sofistikace

Mají preferovanou pouze jednu zájmovou oblast. Jedná se především o počítačové hry, či hry na playstation. U některých her je grafika i fyzikální model propracovaný na vysoké úrovni a tyto aplikace se chovají reálně. Jejich primárním a jediným účelem je však zábava. V určitém směru však mohou také být pomůckou pro zvládnutí řízení automobilu či motocyklu.

Mezi nejlépe propracované aplikace s automobily může být zmíněna hra „Live for speed“¹. Tato hra má velice reálný fyzikální model a řidič musí velice přesně a také správně zacházet jak s volantem, tak i s plynem, řazením, aby „udržel auto na silnici“. Může se při nesprávné manipulaci dostat do smyku, či vylétnout ze zatáčky. Vlastnosti jsou závislé i na površích – asfalt, voda, písek ...

Simulátorem motocyklů na výborné úrovni je „Tourist trophy“². Tvůrci si zde dali práci s dobrou grafikou a také s výborným fyzikálním modelem, kde se motocykly musí ovládat stejně jako v reálném provozu, kde záleží na náklonu do zatáčky, přenášení těžiště, přidávání plynu atd., aby jezdec i motocykl zůstali na trati.

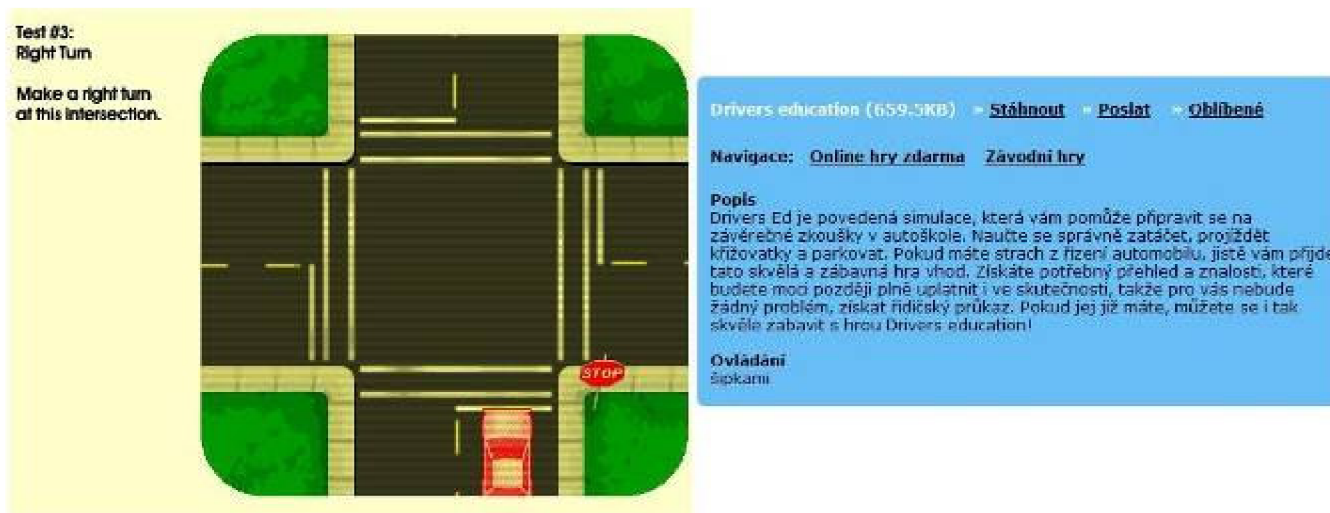
Mezi další aplikace se dají zařadit nejrůznější flash hry na parkování či trénování řídičských dovedností. Mají sice za účel osvojování řídičských dovedností, ale grafika, fyzikální model i ovládání jsou na velice nízké úrovni.

Mezi tyto patří např. „Driver´s ED“³. Je zde trénink nejrůznějších situací od pouhého zatočení či parkování až po odbočování, nebo předjíždění. Ovládání této hry je však velice špatné.

¹ Live for Speed – závodní hra s propracovaným fyzikálním enginem. [16]

² Tourist Trophy – motocyklová simulace od tvůrců série Gran Turismo, týmu Polyphony Digital.

³ Driver´s ED – Počítačová hra simulující primitivní pravidla silničního provozu. [13]



Obrázek 3.7: Náhled jednoduché aplikace „Driver ED“ simulující pravidla silničního provozu [13]

Tyto aplikace se pro výuku autoškoly nehodí a jsou spíše pro odreagování a hraní. Některé mohou mít přínos v naučení či osvěžení nejzákladnějších úkonů a nejjednodušších situací v provozu.

4 Analýza problematiky

Tato kapitola se zaměří na analýzu aplikace 3D Fahrschule. Aplikace 3D Fahrschule je zvolena za referenční model a podle jejích předností a nedostatků jsou navrženy metody a možnosti pro vytvoření nového simulátoru 3D autoškoly. Tímto krokem je získávání znalostí o realitě a reálných systémech. Na již existujícím simulačním modelu provádíme simulace a z nich získáváme data, která následně poslouží jako nové znalosti pro vytvoření nového abstraktního a simulačního modelu. Jedná se tak o získání znalostí, jak bylo popsáno v kapitole 2.

4.1 Analýza aplikace 3D Fahrschule

Aplikace 3D Fahrschule má i přes své dobré zpracování chyby a drobné detaily, které by mohly být zlepšeny. Z následující analýzy bude patrné, jakých chyb by se měl návrh vyvarovat a naopak které detaily by bylo vhodné ponechat. V následujících podkapitolách jsou popsány silné a slabé stránky aplikace 3D Fahrschule a hodnocení uživatelů této aplikace.

4.1.1 Silné stránky

Možnosti nastavení aplikace

Ovládání - Uživatel si může nastavit podle svých požadavků rozložení na klávesnici. Ovládání aplikaci si tedy přizpůsobí tak, jak mu to nejvíce vyhovuje. Volba jednotlivých kláves je přehledná, takže každý bez dlouhého studování lehce nakonfiguruje kteroukoliv klávesu na ovládání. U každého políčka je jednoslovný, případně dvou slovný popis co klávesa znamená a po stisknutí tlačítka si je možné zvolit požadovanou klávesu na příslušnou akci ovládání aplikace. Nastavovat je možné ovládání auta, tedy řízení a dále ovládání pohledů. Tím je myšlen pohled doleva a doprava, dále nahoru a dolů, dozadu, zobrazení pokynu instruktora a zobrazení zpětných zrcátek.

Aplikace také umožňuje ovládání přes joystick. Joystick je možné si navolit podle vlastních požadavků a také je k dispozici jeho kalibrace.

Grafika umožňuje volbu rozlišení, barevnou hloubku, počet snímků za sekundu, úroveň detailů, gama korekci a paměť pro textury. Zvukové volby umožní nastavit celkovou hlasitost a pak i hlasitost motoru vozidla, hlasitost okolních zvuků a v neposlední řadě hlasitost pokynů instruktora.

Kromě nastavení ovládání, grafiky a zvuků je k dispozici i výběr pravidel a kontrol. Jedná se o nastavení, zda se budou zobrazovat výstrahy při určitých přestupcích, či chybách, podpora navádění směru a jednoduché nastavení vlastností vozidla (maximální rychlost, výkon, ...).

Pravidla silničního provozu

Aplikace má na dobré úrovni aplikovaná pravidla silničního provozu. Při jejich porušování je uživatel upozorněn instruktorem na vzniklou chybu. Pokud není chyba příliš závažná, nedostává uživatel pouze body k jeho virtuální zkoušce. Mezi takovéto lehké chyby patří například nedání znamení o změně směru jízdy, nebo nerozhlednutí se na křivatce. Je zahrnuto dokonce i hlídání pohledu do zpětných zrcátek. Pohled do zpětného zrcátka je vyvolán klávesou na zapnutí zobrazení zpětných zrcátek, která se na malý okamžik objeví na obrazovce a následně se sama skryje. Toto je dobré z pohledu kontroly, aby uživatel používal zpětná zrcátka. Zrcátka jsou však k dispozici pouze po

jejich vyvolání a jsou zobrazena uprostřed výhledu a proto zakrývají v době jejich aktivního zobrazení velkou část přímého výhledu.

K závažným přestupkům patří například projetí na červenou na semaforu. Při takovémto přestupku je jízda přerušena, protože instruktor musel zasáhnout do řízení. Lekce se v tomto okamžiku musí dát opakovat. Opakování je od místa než došlo k chybě. Uživatel tedy nemusí opakovat celou jízdu od začátku.

Modulárnost lekcí

V 3D autoškole jsou zahrnuty tréninkové lekce, které pokrývají široký okruh pravidel silničního provozu. Po absolvování každé lekce se zapisuje do výsledkového formuláře výsledek podle toho, kolika chyb a přestupků se uživatel dopustil. Každá lekce je zaměřena na určitý druh situace, která může vzniknout na silnicích. Je zde vše od začátku, tedy odbočování na křižovatkách se semaforů a značkami, dále jízda na křižovatkách s předností zprava, parkování, jízdou po dálnici i jízdou v noci. Uživatel si může spustit kteroukoliv lekci nezávisle na tom, zdali již předchozí lekce absolvoval, či neabsolvoval. To mu umožní mnohem lepší procvičení některých situací.

Při opakování každé lekce je dopravní situace stejná. To se může zdát z prvního pohledu nevhodné, protože po několikátém opakování si uživatel již zapamatuje, co se stane. Pokud je simulace zastavena při přestupku (sražení chodce na přechodu při odbočování vpravo), tak i po vrácení se před chybou situací je onen chodec opět v okamžiku odbočování na přechodu. Důležitým faktem je však to, že tvůrce chtěl naučit uživatele dávat přednost chodcům při odbočování vpravo. Uživatel si tak uvědomí, že chodec tam bude a naučí se jak na takovou situaci reagovat. Pokud by byla dopravní situace generována náhodně, nemusel by chodec být přítomen a zamýšlená lekce na přednost chodci by ztrácela význam. Záleží tedy na individuálním posouzení situace, zda by větší náhodnost byla přínosem či naopak.

Grafika

Grafika je propracovaná na dobré úrovni. Není zde namodelován každý detail okolního prostředí či ostatních účastníků silničního provozu. To ale není na škodu, protože uživatel si všímá prvků, které jsou podstatné. Interiér automobilu je zpracováván velice dobře stejně jako budovy a okolní prostředí. O něco horší grafika je u ostatních účastníků silničního provozu. Tedy u chodců a ostatních vozidel. Především jejich pohyby jsou zpracovány hůře. Důležitou a také dobře propracovanou částí je dopravní značení.

Pro někoho možná horší grafická stránka některých částí může být vnímána negativně, ale celá aplikace není náročná a nevyžaduje žádné výkonné grafické karty a počítačové systémy. Prioritou hry je naučení silničních pravidel a ne vytvoření realistické grafiky. Také je důležité, aby si ji mohl pustit každý uživatel i na starším a počítači či notebooku.

Upozorňování na chyby

V průběhu jízdy je uživatel upozorňován na chyby, kterých se dopustil. Upozorňování je znázorněno přehledným způsobem v horní části, kde se zobrazí učiněná chyba v poloprůhledném okně a zároveň je vyslovena připomínka instruktora. Při lehčích chybách se zobrazí toto upozornění a simulace pokračuje. Například při nedání přednosti v jízdě je navíc zobrazen v pravém horním rohu symbol značky, která nebyla dodržena, tedy v tomto případě symbol „dej přednost v jízdě“ nebo „STŮJ dej přednost v jízdě“. Při spáchání závažnější chyby jako je sražení chodce, nebo projetí na červenou apod. je nejen zobrazeno upozornění, ale simulace je také zastavena a daná situace se opakuje.

Propracovanost prostředí

Prostředí celé hry je dobře zasazeno do reálného světa. Při jízdě v noci je důležité mít zapnutá světla, protože uživatel lépe vidí před sebou osvětlenou vozovku. Ostatní automobily a motocykly dokreslují reálné prostředí a svým chováním vytváří dopravní situace, na které se uživatel učí správně reagovat.

Nedílnou součástí jsou i chodci, kteří jsou zpracováni v celém okolí. Tedy nejen že vstupují do vozovky, ale také chodí po chodníku a uživatel je vnímá jako součást dopravní situace. Autoři pamatovali i na tramvaje a křížení silnice s kolejovou tratí. Celkové zpracování prostředí působí velice dobrým dojmem.

Mluvené instrukce

Jak již jsem zmínil, jsou instrukce zobrazeny nejen textově, ale i hlasově. Právě zvukový vjem od instruktora přidává do jízdy autentičnost opravdové jízdy v autoškole, kde instruktor také mluví se žákem a dává mu příslušné pokyny, rady i výtky při způsobené chybě.

4.1.2 Slabé stránky

Rozhlížení

Jako jeden ze zásadních nedostatků je rozhlížení při jízdě jak v autě, tak i na motocyklu. Prvním nevhodným prvkem je samotné ovládání úkonu, kde musí uživatel stisknout klávesu shift a zároveň šipku směru kam se chce podívat. Již tento samotný úkon je na ovládání složitý, protože ze šipek na ovládání směru jízdy automobilu či motocyklu se stávají šipky na ovládání pohledu. Není tedy možné zároveň řídit vozidlo a dívat se některým jiným směrem než dopředu. I stisknutí několika kláves najednou není vhodné, protože při ovládání ostatních prvků je celá situace matoucí a může dojít ke stisknutí špatné klávesy a podobně. Celý tento manévr vyžaduje soustředění a odvádí tedy pozornost od řízení a situace, protože uživatel se soustředí na ovládání na klávesnici.

Druhým nedostatkem rozhlížení je pohyb. Jde se podívat jenom na maximální úroveň doprava nebo na maximální úroveň doleva, což činí cca 45° na každou stranu. Není umožněno se tedy dívat například jen o 15° na stranu. Menší úhel je sice zobrazen při otočení na danou stranu, ale pouze jako přechod do maximálního úhlu natočení. Pokud je stisk klávesy kratší nedojde až k maximálnímu otočení hlavy, ale jakmile se klávesa pustí, hlava se automaticky otočí zpět do přímého směru. Není možné ji v tomto stavu podržet. Celý pohyb při rozhlížení je také velice prudký a při rozhlédnutí se vícekrát nalevo a napravo po sobě, působí celkový dojem trhaně, což je nepříjemné.

Ovládání motocyklu a řazení rychlostních stupňů

Ovládání motocyklu je velice nereálné. Především náklony při zatáčení a celkové zatáčení celého motocyklu je pomalé a málo dynamické. Manipulace s řídítky je pomalá a při náhlejší změně směru jízdy zprava doleva, či naopak je rychlost pohybu nedostačující.

Na motocyklu není příliš korektní řazení rychlostních stupňů. Při zapnutém manuálním řazení, se pomocí dvou kláves řadí buď o jeden vyšší, anebo o jeden nižší rychlostní stupeň. Toto z pohledu motocyklu odpovídá přesně realitě s tou výhradou, že v reálu je mezi prvním a druhým rychlostním stupněm neutrální, který se musí zařadit při zastavení. Tento fakt v aplikaci chybí. Na motocyklu je zařazení neutrálu umožněno pomocí klávesy pro ruční brzdu, která na motocyklu vůbec není. Pro výuku autoškoly toto považuji za nedostatek, kdy uživatel může získat nesprávné návyky.

Řazení u automobilu by se dalo popsat také jako nereálné, protože se neřadí po dráze řadicí páky a nevyřazuje se neutrál, ale udělat reálné ovládání řadicí páky pomocí klávesnice by bylo velice matoucí a nereálné při splnění podmínky, aby uživatelské rozhraní bylo jednoduché a přívětivé. Zastavení a vyřazení v automobilu je taktéž umožněno pomocí ruční brzdy, ale v tomto případě se již jedná o správnou situaci, jelikož každý automobil ruční brzdu má. V automobilech s automatickou převodovkou je ruční brzda a řadicí páka v poloze „parkování“ stejným prvkem, proto je i v tomto případě zastavení a vyřazení pomocí stisku ruční brzdy korektní.

Zobrazení zpětných zrcátek

Zobrazení zpětných zrcátek je umožněno pomocí klávesy mezerníku (při základním nastavení). Zpětná zrcátka jsou však zobrazena jenom po krátký okamžik a poté jsou automaticky skryta. Tento způsob není zcela vhodný, protože v reálném provozu se řidič do zpětného zrcátka podívá periferně i během jízdy potřebuje-li vědět, jestli za ním někdo je na křižovatce, či má volno při přejíždění z pruhu do pruhu. Zapnutí zpětného zrcátka klávesou je potom spíše na obtíž, protože v 3D Fahrschule se zpětná zrcátka zobrazí ve výhledu řidiče a zakryjí velkou část situace vpředu. Další absencí je levé zpětné zrcátko, které nejde zobrazit žádným způsobem. Tento přístup není realistický a ani z pohledu autoškoly příliš vhodný.

Uživatelské rozhraní – ovládání

Ovládání vozidla je v některých situacích příliš složité. Jak již bylo zmíněno výše, stálé zobrazování zpětných zrcátek je nepohodlné a nevhodně je vyřešeno rozhlížením do stran. Ovládání samotného automobilu či motocyklu není nejvhodnější. Zatáčení je nepříliš dobře vyřešeno, protože při prudším zatočení je těžké dostat se znovu rychle do správného směru. Nejednou se také stane, že si při ovládání uživatel splete klávesu na určitý úkon a poté se snaží napravit situaci místo vnímání situace v dopravě. Celé uživatelské rozhraní aplikace by měl být co nejjednodušší a intuitivní, aby nedocházelo k milným stiskům kláves a už vůbec ne k přemýšlení, která kombinace kláves má být zvolena pro správnou akci.

Menu s ovládáním

Aplikace nemá žádnou nápovědu či menu s ovládáním vozidel, pohledů a ostatních nastavení při hře. Jediné jak uživatel zjistí kterými klávesami hru ovládat, je v základním nastavení aplikace, kde se volí nastavení kláves. Tuto nabídku lze sice i při hře vyvolat klávesou F1, ale je zobrazen pouze pohled do nastavování ovládání. Bylo by vhodné, aby na začátku bylo zobrazeno jednoduché menu s nápovědou s jednoduchým a názorným popisem ovládání, které se automaticky skryje.

Ukazatele paliva a teploty

Na přístrojové desce automobilu jsou zobrazeny ukazatele teploty a paliva. Tyto ukazatele jsou však nefunkční. Palivoměr a ubývání paliva v nádrži by byly zajímavým prvkem, protože si žák/řidič autoškoly uvědomí, že se musí sledovat i jiné přístroje. Některá lekce by tedy mohla být koncipována i s najetím na benzínovou stanicí. Ukazatel teploty má význam především na počátku jízdy, kdy se může simulovat jízda se studeným motorem a tedy i jeho šetrným zacházením. Zajímavým a vhodným prvkem může být přehřátí motoru, aby si uživatel uvědomil tuto skutečnost, kterou v reálném provozu zažije až tehdy, když ji nejméně čeká a potřebuje.

Výhled z automobilu

Při pohledu z kabiny, který je jediným realistickým pohledem jako ve skutečném automobilu, není zobrazení nejlépe namodelováno. Především by bylo dobré, aby bylo vidět část přední kapoty, což umožní lepší orientaci. Je to důležité především pro začínající řidiče, pro získání představ o rozměrech vozidla. Výhled z automobilu zobrazuje malý zorný úhel. Je vidět pouze levá část výhledu a řidič nemá představu o pravém boku automobilu.

4.1.3 Hodnocení nezávislých uživatelů

Pro lepší analýzu této aplikace, která bude pro tuto práci sloužit jako referenční, bylo provedeno na několika osobách testování, kde každý vyjádřil svůj názor na nedostatky a přednosti aplikace „3D Fahrschule“. Každý uživatel nezávisle na předchozích výsledcích testoval různé situace a chování aplikace a zahrnul svůj názor. Po shrnutí všech připomínek a postřehů byly výsledky shrnuty do tabulky 4.1.

položka uživatel	grafika aplikace	ovládání auta	ovládání (jiné)	fyzikální model	pravidla provozu	přínos autoškoly
uživatel 1	3	1	3	3	2	2
uživatel 2	3	4	3	4	1	2
uživatel 3	2	2	3	1	1	1
uživatel 4	4	3	4	2	2	2

*hodnocení: 1 = nejlepší, 5 = nejhorší

Tabulka 4.1: průzkumu hodnocení aplikace „3D Fahrschule“

4.1.4 SWOT analýza

SWOT analýza [1] je metoda, pomocí které je možno identifikovat silné (ang: Strengths) a slabé (ang: Weaknesses) stránky, příležitosti (ang: Opportunities) a hrozby (ang: Threats), spojené s určitým projektem, typem podnikání, opatřením, politikou apod. Jedná se o metodu analýzy užívanou především v marketingu, ale také např. při analýze a tvorbě politik (policy analysis). S její pomocí je možné komplexně vyhodnotit fungování aplikace, nalézt problémy, nebo nové možnosti zlepšení. Je součástí strategického (dlouhodobého) plánování vývoje.

Tato analýza byla vyvinuta Albertem Humphreym, který vedl v 60. a 70. letech výzkumný projekt na Stanfordské univerzitě, při němž byla využita data od 500 nejvýznamnějších amerických společností.

Základ metody spočívá v klasifikaci a ohodnocení jednotlivých faktorů, které jsou rozděleny do 4 výše uvedených základních skupin. Vzájemnou interakcí faktorů silných a slabých stránek na jedné straně vůči příležitostem a nebezpečím na straně druhé, lze získat nové kvalitativní informace, které charakterizují a hodnotí úroveň jejich vzájemného střetu. [1]

Poznátky z kapitol 4.1.1 a 4.1.2 jsou shrnuty v následující tabulce.

SWOT analýza		Interní analýza	
		silné stránky	slabé stránky
Externí analýza	příležitosti	Možnosti rozšiřování aplikace - otevřený kód, možnost rozšiřování aplikace	Rozhlížení ovládání motocyklu, řazení
		Spolupráce s jinými projekty - vytvoření komplexnějšího systému	zobrazení zpětných zrcátek uživatelské rozhraní - ovládání
		Načítání grafických modelů a prostředí ze souborů - možnost nezávislého zlepšování grafiky	menu - nápověda s ovládáním ukazatel paliva a teploty výhled z automobilu
	hrozby	Možnosti nastavení aplikace	Nové uživatelské rozhraní - intuitivnější ovládání, přirozenější rozložení kláves
		Pravidla silničního provozu	
		Modulárnost lekcí	Pohyb hlavy řidiče - realističnost, možnost pohledu do zpětných zrcátek
		Grafika	
		Upozornění na chyby	Nové prvky zobrazení ukazatelů - palivo, teplota, ...
		Propracovanost prostředí	
	Mluvené instrukce		

Tabulka 4.2: SWOT analýza na základě zkušeností s 3D Fahrschule

5 Návrh nové aplikace

5.1 Rozhlížení

Jednou z podstatných vlastností, která by neměla být opomíjena, je rozhlížení a pohledy z automobilu. Charakteristika tohoto problému byla rozebrána výše v předchozí kapitole.

Jednou z možností ovládání a potom i celkového zpracování problematiky rozhlížení a pohledů je plynulé ovládání pomocí myši. Při ovládání na klávesnici jsou možné pouze dvě krajní polohy rozhlédnutí se doleva a doprava. Při ovládání pomocí myši můžeme dosáhnout libovolné polohy hlavy. Ovládání spočívá pouze v pohybech myši, která určuje natočení a směr hlavy řidiče sedícího ve vozidle. Ovládání hlavy a ovládání vozidla je nezávislé. Plynulost natáčení pohledu umožňuje setrvat v natočení hlavy po jakkoli dlouho dobu. Tzn., že stejně jako při reálném řízení může řidič jet dopředu a dívat se například mírně vlevo, což je typické při odbočování doprava. Když řidič zatáčí vpravo, kontroluje si, jestli mu z levé strany nepřijíždí jiné vozidlo apod.

Uživatelé zvyklí na herní simulátory, mohou mít z počátku problém s ovládáním, protože ve hrách se využívá šipek a pravé ruky pro ovládání automobilu. Většina lidí drží myš v pravé ruce, a proto se bude muset vozidlo ovládat levou rukou na šipkách. Tato stránka nemusí být vnímána jako velká závada, protože rychlosti automobilu v autoškole se nedají srovnávat s automobilovými závody a proto i ovládání levou rukou bude dostatečně přesné a pohodové pro jízdu v autoškole.

Další menší nevýhodou je to, že pravá ruka bude ovládat myš, na které se dají použít pro nějaké úkony pouze levé a pravé tlačítko. Z toho důvodu ubude možností na ovládání aplikace a klávesy musí být proto navrženy tak, aby se celé ovládání dalo zvládnout levou rukou.

5.2 Ovládání motocyklu

Protože v naší aplikaci nebudeme jízdu na motocyklu implementovat, nebudu tento problém příliš rozbírat. Jízda na motocyklu z pohledu počítačové simulace v autoškole není příliš reálná a zvládnutí a naučení pravidel je stejné jako v automobilu. Přesto je popsáno několik návrhů na zlepšení této slabé stránky.

V prvním kroku je to samotné ovládání při řízení a zatáčení. Zde by bylo vhodné, aby motocykl mnohem rychleji a pružněji reagoval na pokyny jezdce a také zatáčel rychleji než je tomu v 3D Fahrschule. Řazení na motocyklu, které je sekvenční stejně jako v aplikaci, pozbývá fakt, že mezi prvním a druhým rychlostním stupněm je neutrální. Proto by bylo vhodné přepracování řadicího mechanismu a neutrální zařadit mezi tyto dva rychlostní stupně.

5.3 Zobrazení zpětných zrcátek

Při novém zpracování pohledů, jak bylo popsáno výše, tedy že se hlava bude ovládat pomocí pohybu myši, mohou být zpětná zrcátka zobrazena stále a při natočení na příslušnou stranu se do nich uživatel sám podívá, což je realistické stejně jako ve skutečném automobilu.

5.4 Uživatelské rozhraní – ovládání

Tato položka opět souvisí s návrhem ovládání pomocí myši. Ovládání musí být navoleno takovým způsobem, aby levá ruka mohla ovládat jednoduše všechny podstatné prvky. Na tlačítka myši zvolit směrová světla a zbytek již na klávesnici na levou ruku. Jednoduše je tedy nutné umístit ovládání směru automobilu, řazení rychlostních stupňů a troubení. Jedná se o 7 kláves. Pro ovládání směru automobilu se přirozeně nabízí šipky na klávesnici. Poté musí být jednoduše přístupné řazení rychlostních stupňů, což by mohlo být na klávesách a „shift“ a „ctrl“. Jako poslední často používanou klávesou je troubení. Jako ideální klávesa je „mezerník“ z důvodu, že je velký a k této funkci je ideální, stejně jako je troubení ve vozidle také vždy přímo na očích a dobře přístupném místě k okamžitému použití. Ostatní rozložení kláves jako nastavení zapnutí světel a podobné úkony mohou být na klávesnici „kdekoliv“, protože jejich použití není vyžadování přímo k řízení vozidla, ale ve většině případů se použijí jako přednastavení před jízdou.

5.5 Menu s ovládáním

Velice vhodným prvkem se jeví zobrazení menu s ovládáním a rozložením klávesnice při spuštění aplikace. Na obrazovce se zobrazí na krátký časový okamžik okno s ovládáním, aby uživatel lehce získal potřebné informace o ovládacích prvcích. Toto menu se pak může znovu vyvolávat klávesou, pro kterou bych zvolil tradiční klávesu F1, sloužící pro nápovědu.

5.6 Výhled z automobilu

Tento bod opět souvisí s přepracováním pohledů hlavy pomocí myši. Příspěvkem k lepšímu je malá, ale účinná změna a to namodelování kapoty před vozidlo. Řidič pohledem na přední část kapoty automobilu získá představu o velikosti a o poloze celého vozidla.

5.7 Návrh nového ovládání

Ovládání vozidla v 3D autoškole je po analýze různých již existujících aplikací na trhu navrženo tak, aby bylo co nejvíce uživatelsky příjemné a také aby splňovalo požadavky jednoduchosti a efektivnosti.

Jak již bylo popsáno v předešlých kapitolách, největší změnou je ovládání pomocí myši a klávesnice zároveň, kde myš určuje pohyb hlavy a klávesnice pohyb automobilu.

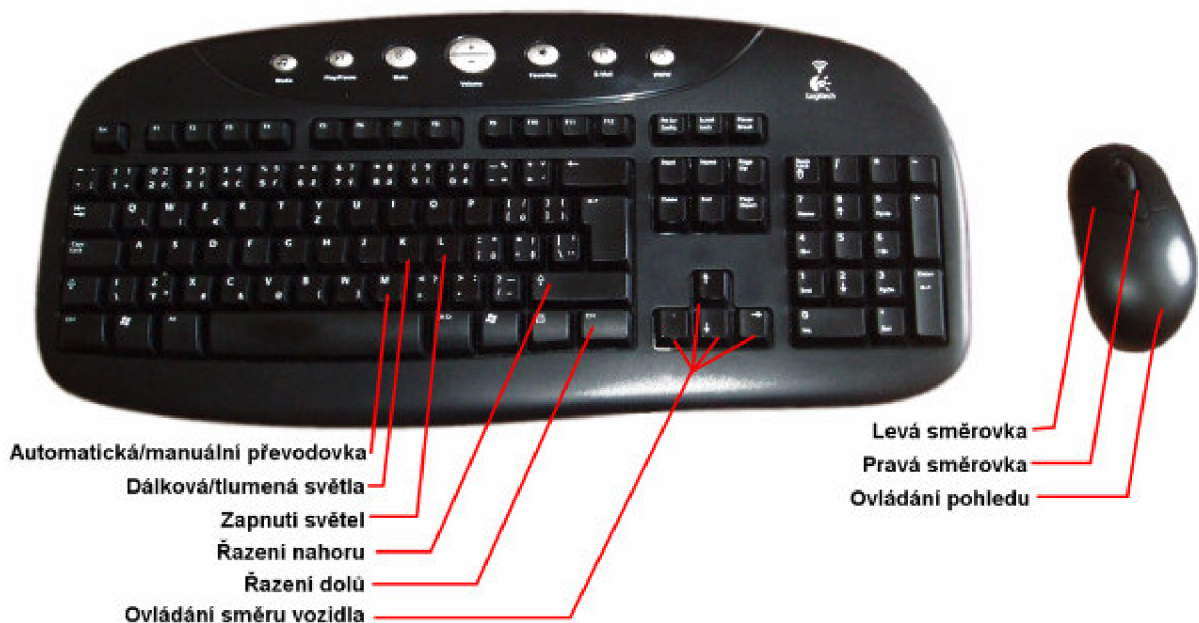
5.7.1 Myš

Pohybem myši ovládá uživatel hlavu řidiče. Tím ovlivní směr pohledu. Na pravém a levém tlačítku myši budou umístěny směrová světla. Je to navrženo z důvodu, že se používají poměrně často a je tedy třeba je mít na lehce dostupném místě.

5.7.2 Klávesnice

Pohyb automobilu je navrhnut na obvyklé šipky. Pro zatočení vlevo je levá šipka, pro zatočení vpravo je pravá šipka. Plyn se přidá šipkou nahoru a brzda je šipkou dolů. Pro změnu směru jízdy budou sloužit klávesy levý „shift“ a levý „ctrl“. Pokud bude zapnuta automatická převodovka - tedy samočinné řazení, bude klávesa „shift“ sloužit k zapnutí pohybu vozidla vpřed a „ctrl“ pro zapnutí zpětného chodu. Při zapnuté manuální převodovce budou sloužit klávesy jako sekvenční řazení. Klávesou „shift“ se zvýší rychlostní stupeň a klávesou „ctrl se sníží rychlostní stupeň. Zpátečka bude umístěna nejnižže. Rychlosti půjdou tedy v tomto pořadí. Zpátečka, neutrál, první rychlost, druhá rychlost, atd. Zvuková siréna by měla být na viditelném místě, a proto jsem zvolil klávesu „mezerník“. Toto jsou nejdůležitější prvky pro ovládání při řízení. Před jízdou bude možné ještě nastavit základní funkce. Klávesou „M“ manuální převodovka, nebo převodovka automatická. Klávesou „L“ zapnutí či vypnutí světel.

Popis ovládání je znázorněn na obrázku 5.1.



Obrázek 5.1: Návrh ovládání aplikace pomocí myši a klávesnice

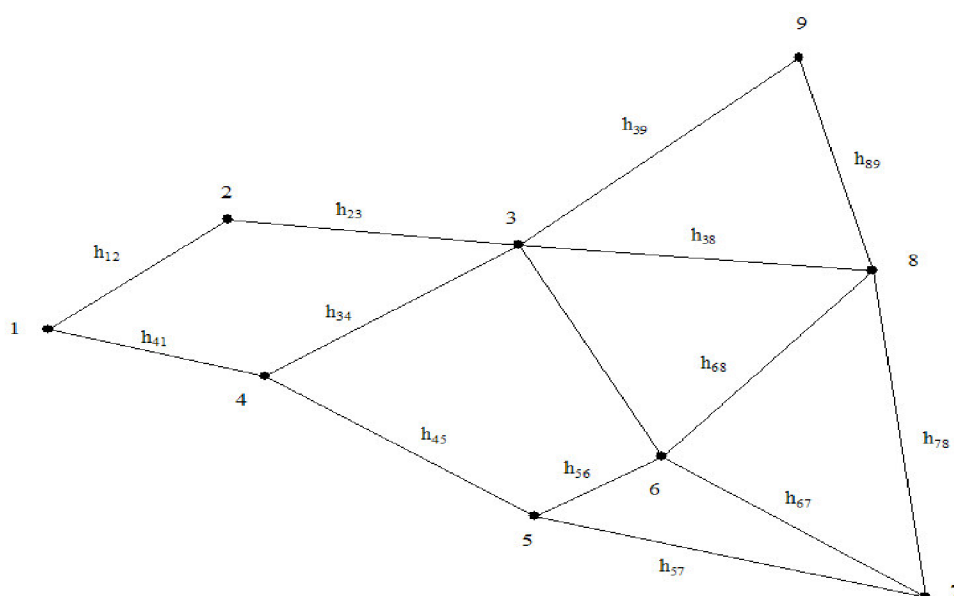
6 Problematika simulace dopravního systému

Pro popis silniční infrastruktury je nutné mít komplexní metodu popisu, která bude robustní a bude svými vlastnostmi umět zahrnout všechny požadavky dopravních situací a pravidel silničního provozu.

V geografických systémech [17] je pro popis a reprezentaci silniční sítě často využíván graf. Podle míry požadavků a nároků na danou aplikaci, se můžeme setkávat s různými grafy.

6.1 Grafy

Graf je soustavou bodů a jejich spojnic, přičemž body se označují jako uzly a spojnice jako hrany. V grafické podobě se uzly vykrešlují pomocí bodových značek, hrany pomocí úseček, lomených čar popř. hladkých čar. Jsou-li dva uzly koncovými uzly jedné hrany, označujeme je jako přilehlé. Každý z přilehlých uzlů je incidentní s přilehlou hranou a naopak přilehlá hrana je incidentní s přilehlými uzly. To si můžeme vysvětlit také tak, že každý počáteční či koncový bod hrany má v topologickém prostoru výskyt ve stejném místě jako některý z přilehlých uzlů. Jedná se tedy o východisko k pojmu spojitost. V praxi se uzly grafu označují pomocí identifikátorů a hrany stejným způsobem. [17]



Obrázek 6.1: Obecný graf s hranami a uzly [17]

Hranám lze přisoudit jednoznačný směr a potom je můžeme označovat jako orientované hrany, stejně tak budeme graf s orientovanými hranami označovat pojmem orientovaný graf. Hranám grafu lze přiřadit popis jejich charakteristické vlastnosti. Takový graf nazýváme graf hranově ohodnocený a uvedenou charakteristikou může být např. skutečná délka linie, kterou hrana reprezentuje. V případě, že takový ohodnocený graf je modelem - obrazem skutečné sítě z reálného světa, může toto ohodnocení vyjadřovat například dobu, kterou bude potřebovat silniční vozidlo (určitých parametrů)

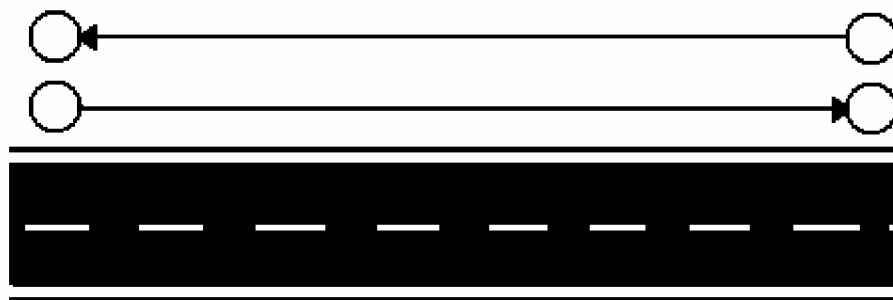
pro překonání konkrétního úseku silnice - ohodnocené hrany. Jinými slovy řečeno, bude to doba jízdy z jedné křižovatky na druhou, tedy z jednoho přilehlého uzlu do druhého. V orientovaném grafu existuje možnost zvolit určité ohodnocení hrany pro její orientaci od počátečního uzlu do uzlu koncového a odlišné ohodnocení pro opačnou orientaci od koncového uzlu do uzlu počátečního. S využitím tohoto mechanismu je možné popsat graf s vyjádřením odlišných vlastností hrany při jedné či opačné orientaci (anizotropie⁴). Anizotropie je možno využít například při potřebě vyjádřit odlišné cestovní rychlosti při cestě z místa A do místa B a naopak.

Uzlům grafu je možné prostřednictvím konkrétních hodnot rovněž přiřadit popis jejich určité charakteristiky. Pak se jedná o graf uzlově ohodnocený. Budeme-li opět uvažovat, že uzlově ohodnocený graf je modelem - obrazem skutečné sítě z reálného světa, může toto ohodnocení uzlu (uzel si představme jako křižovatku silnic) vyjadřovat například dobu, po kterou bude silniční vozidlo určitých parametrů zdrženo na křižovatce, např. při křížení hlavní komunikace nebo při odbočování.

Velice výhodnou kombinaci možností pak může znamenat graf, v němž jsou ohodnoceny hrany i uzly. Takový graf se označuje pojmem graf hranově a uzlově ohodnocený. Z hlediska potřeb a možností zkoumání jevů a procesů vyskytujících se, případně probíhajících, v reálných sítích či jejich okolí, je popisovaný hranově a uzlově ohodnocený graf velice sofistikovaným prostředím nabízejícím široké možnosti použití analytických úloh. [17]

6.2 Model dopravní infrastruktury

Pro popis modelu infrastruktury, která bude znázorňovat dopravní situace v našem simulátoru, bude použito grafu. Pro implementaci se nejvíce hodí poslední ze zmiňovaných typů grafu – orientovaný graf hranově a uzlově ohodnocený.



Obrázek 6.2: Model komunikace orientovaným grafem

Uzly budou křižovatky, či jiné přerušení a hrany budou znázorňovat ulice, silnice, či chodníky, jak to bude nutné pro popis simulace. Orientovaný graf zajistí možnost definovat jízdní pruhy a možnosti chování v jízdních pruzích.

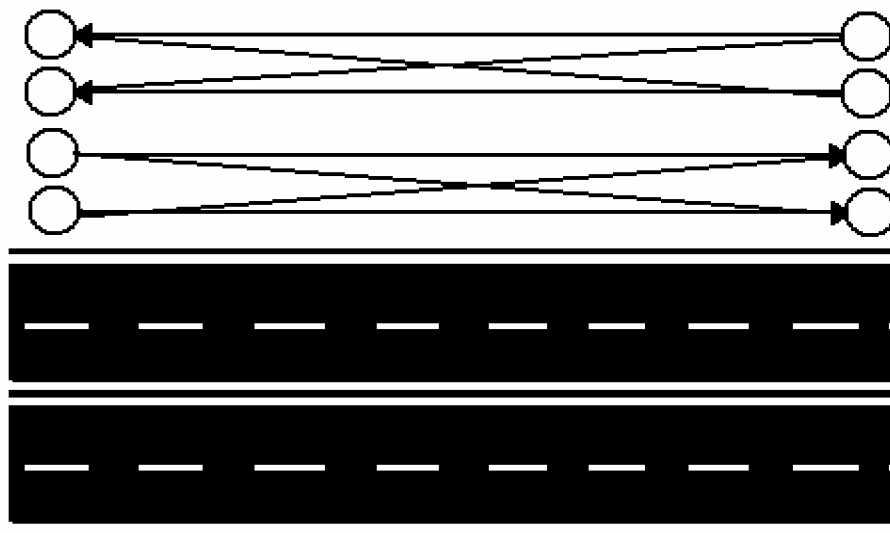
⁴ **Anizotropie** je vlastnost, která označuje závislost určité veličiny na volbě směru.

6.2.1 Ulice a jízdní pruhy

Hrany budou reprezentovat jízdní pruhy. Orientace hrany bude určovat směr jízdního pruhu. Každá hrana bude v sobě obsahovat informaci, k jakému záměru má být využita, tedy jakou situaci na komunikaci právě znázorňuje. V běžné praxi se můžeme setkat s těmito reprezentacemi [16, 17]:

- pruh silnice pro vozidla,
- tramvajový pás,
- chodník pro chodce,
- pruh pro tramvaje a současně pro automobily,
- pěší zóna – spojení komunikace s chodníkem.

Mezi jednotlivými jízdními pruhy, pokud to předpisy dovolují, se může přejíždět. Tato skutečnost může být v grafu zaznamenána tak, že mezi uzly bude více přechodů.

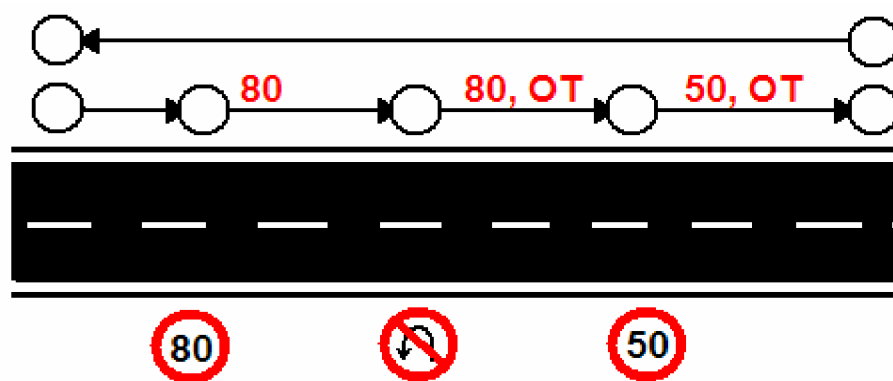


Obrázek 6.3: Model komunikace s více jízdními pruhy orientovaným grafem

Každá hrana bude mít svůj účel a bude tedy možné vyjádřit i předjíždění v protisměrném pruhu a podobné situace.

6.2.2 Značky

Pro značky upravující přednosti v jízdě na křižovatkách platí způsoby modelování, které budou uvedeny později. Ostatní značky, které se mohou nacházet mezi křižovatkami, v našem případě mezi uzly, musí být do naší struktury také zaznamenány. Jedná se například o značky upravující rychlost nebo zákazy předjíždění. Značky jsou platné až do následující křižovatky, nebo je jejich platnost ukončena jinou značkou. Takovou skutečnost zaneseme do grafu dalším uzlem. Hrana z jedné křižovatky do druhé bude rozdělena na více částí a bude mít seznam uzlů, které na ní leží. Tyto uzly ležící na hraně budou mít jiný příznak než uzly křižovatek, aby je bylo možné vhodně procházet a zjišťovat informace, které sebou nesou.

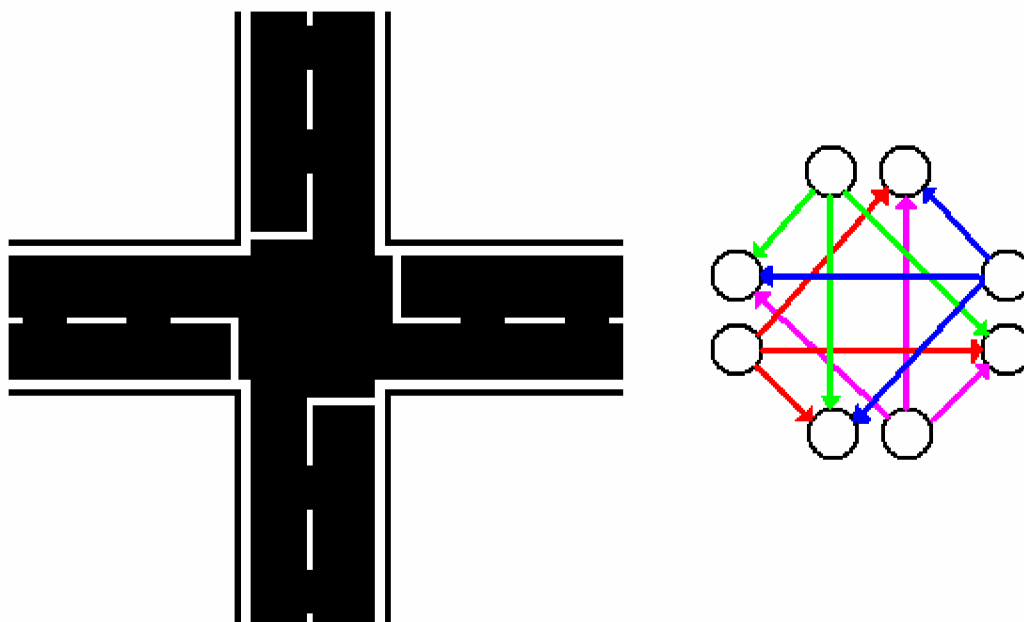


Obrázek 6.4: Model komunikace a dopravních značek orientovaným grafem

Při průjezdu hranou se budou vždy přidávat pravidla, která je nutné dodržovat a pokud budou nějakou značkou zrušena, tak se budou odebírat. Při omezování rychlosti např. z 80km/h na 50km/h se tento údaj přepíše novějším. Jakmile se dorazí do uzlu reprezentujícím křižovatkou, všechna pravidla se odeberou. Pravidla budou uložena v uzlu a budou se ukládat přímo k vozidlu, jakmile vozidlo projede daným uzlem.

6.2.3 Křižovatky

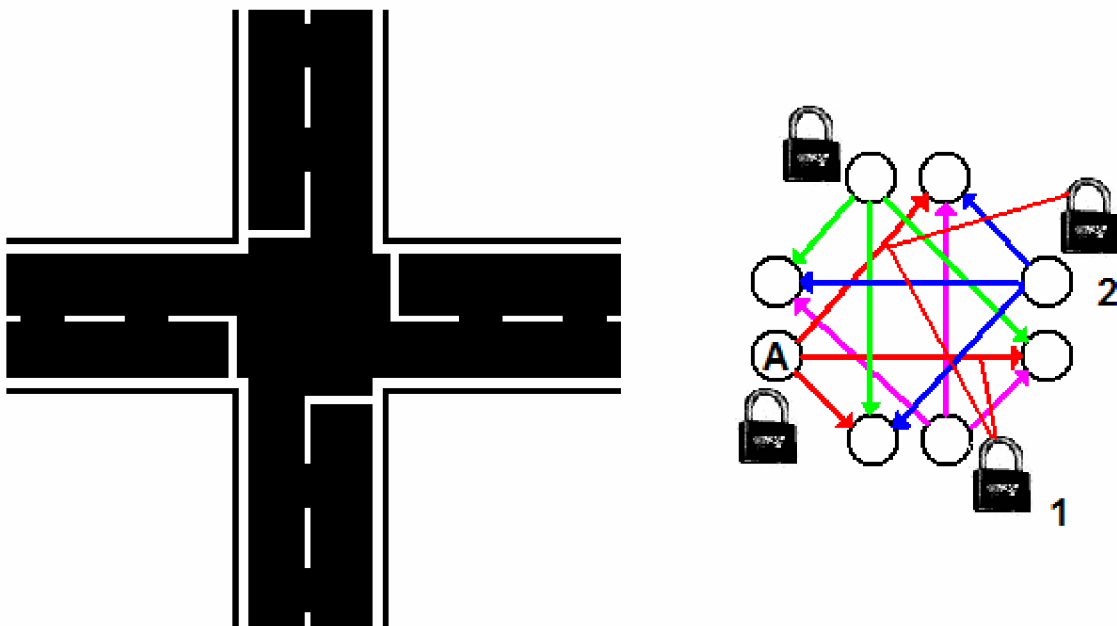
Uzly v grafu budou určovat nějakou změnu v grafu. Takovou změnou může být například křižovatka. Křižovatka je v běžném provozu vcelku jednoduchou situací, protože po zvládnutí pár základních pravidel jedná každý řidič již intuitivně. V křižovatce však musí být znázorněnou hranou, kam je možné pokračovat v jízdě. Na následujícím obrázku je tato situace znázorněna. [16, 17]



Obrázek 6.5: Model křižovatky orientovaným grafem

Již složitějším problémem je vyjádření přednosti v jízdě na křižovatce. Situace bude opět namodelována pomocí hran, které určí, kam může které vozidlo přejet, ale musí být přidána další

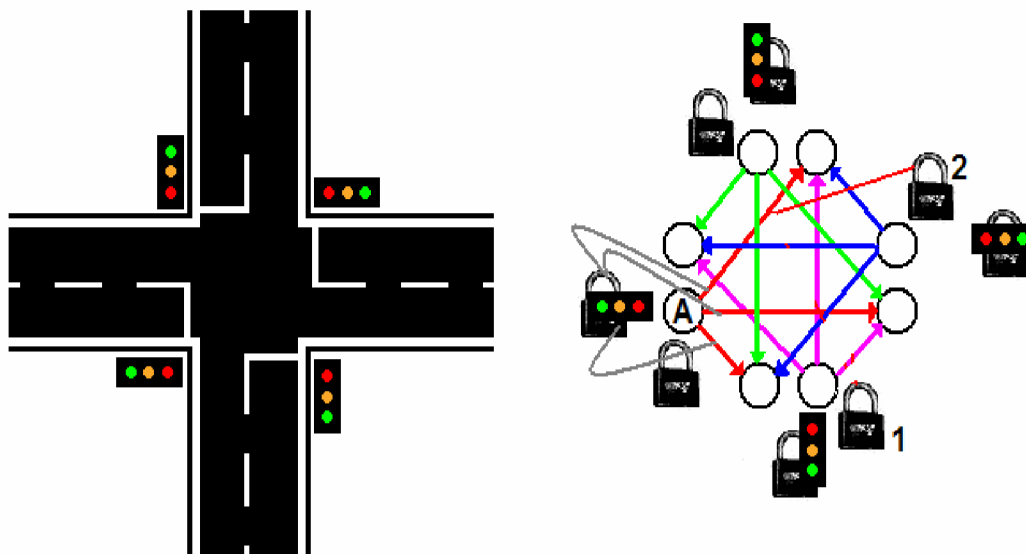
komponenta, která určí, zda je průjezd volný, nebo je v křižovatce vozidlo, kterému se musí dát přednost v jízdě. Pro tuto situaci lze s výhodou využít zámků, které budou u uzlů, avšak nebudou vlastností uzlu, ale samostatnou jednotkou. Jedná se o jednoduchou strukturu, která má stav odemčeno a zamčeno. Ke každé hraně v křižovatce se určí seznam zámků. Aby se přes hranu mohlo projet, budou muset být všechny příslušné zámkové odemčeny, Zámkové budou svázány s uzlem, kterému je nutné dát přednost v jízdě. Pokud do tohoto uzlu vjede vozidlo, zámeček se zamkne. Jiný účastník, který bude chtít využít přechod svázaný s tímto zámkem, bude muset počkat do doby, než bude zámeček odemčen. Zámeček se odemkne v případě, že vozidlo opustí daný uzel. Pro průjezd je nutné, aby byly odemčeny všechny zámkové příslušející požadované hraně. [16, 17]



Obrázek 6.6: Model křižovatky se zámkové orientovaným grafem

Pro přehlednost jsou na obrázku znázorněny pouze příslušnosti zámkové k červeným hranám. Pokud se vozidlo vydá po červené hraně rovně z uzlu „A“, musí dát přednost zprava. Tedy musí být odemčen zámeček číslo 1. Pokud se vydá po červené hraně doprava, nemusí dávat přednost nikomu, a tedy této hraně žádný zámeček nepřísluší. Pokud se rozhodne odbočit vlevo, musí dávat přednost zprava, ale také protijedoucímu vozidlu. K této hraně jsou tedy příslušné zámkové 1 a 2.

Křižovatka řízená světelnými signály bude fungovat na podobném principu, ale navíc bude obsahovat zámkové řízené semaforem. Semaforem v jedné křižovatce mohou být synchronizované, tzn., že semaforem v jednom směru budou mít na obou koncích signál „volno“ a kolmý směr bude mít signál „stop“. Celé řízení křižovatky bude potom střídavě pouštět jeden a druhý směr.



Obrázek 6.7: Model světelné křižovatky se zámky orientovaným grafem

Pro vozidlo jedoucí z uzlu „A“ po červené hraně je vždy příslušející zámek semaforu u uzlu „A“ a poté zámky příslušející hranám podle toho, po které hraně pojedou. Pro rovný směr a odbočení doprava již není přednost v jízdě upravena žádným zámek, pro odbočení doleva je přednost upravena navíc zámek č. 2 – přednost protijedoucímu vozidlu. [16, 17]

6.3 Kontrola dodržování pravidel v simulaci

Kontrola dodržování pravidel v simulaci velkou mírou souvisí s předchozí kapitolou. Účastníci provozu, kteří budou simulováni počítačem, budou se tedy pohybovat v simulaci bez zasahování uživatele autoškoly, budou mít svůj pohyb odvozen od průchodu grafem. Graf byl dříve zvolen i proto, že existují různé algoritmy pro průchody grafem, jako je nalezení např. nejkratší cesty, apod.

Vozidlo pohybující se v simulaci po komunikaci, bude dodržovat předpisy již ze samotné podstaty průchodu grafem, který byl k dodržování pravidel navržen. Vozidlo nebude moci vybočit ze směru nebo vytvářet nepředvídatelné manévry a chyby, protože bude postupovat přesně po hranách grafu.

Samotný průchod grafem však nestačí. Vozidla jedoucí za sebou budou muset respektovat jeden druhého. A v komplikovanějších případech řešit i předjíždění a podobné manévry. Se systémem průchodu grafem bude tedy spolupracovat systém kolizí. Ten bude mít vyšší prioritu. V případě překážky, kdy systém zaznamená, že průjezd je blokován, vozidlo zastaví. Bude se jednat vlastně o „detekci kolizí“, ale s určitým předstihem.

6.3.1 Detekce kolizí

Detekce kolizí se provádí ve scéně, která se skládá z dynamických a statických objektů, aby se zjistilo, zda dynamické objekty nekolidují mezi sebou nebo s objekty statickými. Jestliže kolize nastala, vypočítá se u příslušného kolidujícího dynamického objektu reakce na kolizi. Test pro zjištění kolize by měl být co nejrychlejší, protože je na něm závislá rychlost vykreslování scény. K detekci kolize tedy dochází před každým vykreslením scény.

Dynamické objekty jsou typem objektů, které se ve scéně pohybují a musí se tedy pravidelně aktualizovat jejich umístění v datové struktuře rozdělující scénu a případně i jejich obalová tělesa. Naopak statické objekty jsou ve scéně pevně zakotveny a zařazují se tedy do struktury pouze jednou po jejím vytvoření.

Samotná detekce kolizí se často rozděluje na dvě hlavní části: širší (broad) a užší (narrow). V první části se identifikují dvojice objektů, které mohou kolidovat. V této fázi testování se používají struktury pro rozdělení scény (pravidelná mřížka, segmentový strom, intervalový strom atd.) a různé druhy obalových těles [20] (Axis Aligned Bounding Box, Sphere Bounding atd). Struktury pro rozdělení scény jsou potřebné k systematickému rozdělení prostoru scény, protože dokáží urychlit další testy tak, aby byly vybrány objekty se stejnou vlastností. Většinou se jedná o polohu objektu v prostoru. Smyslem těchto struktur je vyloučit co nejvíce těles, která nemohou s testovaným objektem kolidovat. Použití různého typu struktury je závislé na typu scény, požadavcích aplikace, nebo na paměťové náročnosti struktury.

Protože mají objekty často složitý tvar, umisťují se do obálek, které určitým způsobem aproximují jejich povrch. Obalových těles je více druhů a jejich použití je dáno podle tvaru obalovaného tělesa. Pro těsnější obalení objektu se používají stromy složené z obalových těles. Obalová tělesa jsou opět použita pro vyloučení objektů, které nemohou kolidovat.

V druhé části detekce se zjišťuje, které objekty nalezené v první části kolidují. Testy už se týkají pouze objektů, a proto se provádí test trojúhelník vůči trojúhelníku. Jestliže je zjištěna kolize, dopočítá se na ni reakce na základě kolidujících trojúhelníků. [19]

6.3.2 Dodržování pravidel

Simulace autoškoly má kromě účastníků provozu řízených simulací ještě jednu důležitou součást. Touto součástí je vozidlo, které ovládá uživatel a o které jde v simulátoru autoškoly především.

Kontrola dodržování pravidel u vozidla ovládaného uživatelem bude vycházet z teorií uvedených v předchozí kapitole. Při pohybu vozidla se budou číst informace z uzlů a hran grafů, ale vozidlo se podle nich nebude řídit. Bude řízeno pouze pokyny od uživatele. Dále se budou získávat informace z automobilu uživatele, jako rychlost, poloha a ostatní chování získané vhodnými senzory. Tyto dva druhy dat se budou vyhodnocovat, a pokud se vyhodnotí jako porušení předpisu, či jiné nekorektní chování, bude na to uživatel upozorněn, případně, při vážnější chybě, simulace zastavena.

Nezávisle na tomto systému vyhodnocení bude fungovat i fyzikální engine pro pohyb automobilu, spojený se systémem detekcí kolizí. V případě vyjetí ze silnice, střetu s jiným objektem bude pracovat fyzikální engine podle realistického napodobení skutečnosti kolize.

Podrobný popis systému kolizí použitý v naší aplikaci, je popsán v kapitole zabývající se návrhem a implementací aplikace s použitím fyzikální knihovny Bullet [21].

7 Architektura systému simulátoru

Systém simulátoru autoškoly je navržen s ohledem na jeho co nejširší použití pro běžné uživatele, proto běží na platformě PC pod operačním systémem MS Windows. Ke své činnosti nepotřebuje žádná speciální přídatná zařízení. Ovládání je intuitivní pomocí klávesnice a myši.

Celý systém je navrhován v programovacím jazyce C++, což zajišťuje dobrou podporu jak při vývoji a překladač, tak i při modifikaci. C++ je jedním z nejrozšířenějších programovacích jazyků a proto jsem ho zvolil i pro simulátor autoškoly. Využití samotného C++ se standardními knihovnami by však nebylo efektivní a je proto zapotřebí zvolit vhodný grafický a fyzikální engine.

7.1 Grafický engine

7.1.1 OGRE

OGRE [22] (Object-Oriented Graphics Rendering Engine) je grafický engine orientovaný na zobrazení scény. Jedná se o flexibilní 3D engine napsaný v C++, navrhnutý pro jednoduchý a intuitivní vývoj aplikací s 3D hardwarovou akcelerací. Třídy v knihovnách abstrahují všechny detaily pro používání systémových knihoven jako OpenGL nebo Direct3D. Poskytuje rozhraní založené na objektech scény a jiných vysokoúrovňových scénách.[22]

Ogre není koncipován jako herní engine, ale samozřejmě může a v hojné míře je pro vývoj her používán. Je navržen pro poskytování tříd a nástrojů pro tvorbu grafiky. Ostatní komponenty, jako je zvuk, síťové rozhraní, umělá inteligence, fyzika, apod., se integrují z jiných knihoven. Ogre lze použít k vývoji her, simulací, obchodních aplikací apod. Dokonce i každá hra vyžaduje jiný přístup, protože například u RPG⁵ her je zapotřebí speciálních knihoven pro práci se sítí a u leteckých simulátorů jsou kladeny požadavky na fyzikální systém. Proto se samostatná knihovna pro zobrazování grafiky jeví dobrým a účelným řešením, lepším než komplexní engine obsahující mnoho komponent.

Protože je OGRE vyvíjen přímo s myšlenkou integrace potřebných komponent a knihoven, pro cílený vývoj konkrétních aplikací je navrženo uživatelské rozhraní tak, aby integrace byla jednoduchá. Pro nejznámější a nejpoužívanější komponenty, jako je např. knihovna pro fyziku, existují i speciální balíčky spojující OGRE s vybranou knihovnou.

Ogre je šířen pod licencí LGPL⁶, což znamená, že jsou dostupné i zdrojové kódy a může se použít knihovna i dělat zásahy ve zdrojovém kódu knihovny. LGPL umožňuje i komerční použití. Pro nás je důležité, že knihovna je volně dostupná. [22]

Ogre poskytuje kvalitní dokumentaci, mnoho tutoriálů a ukázkových příkladů různých technik programování v OGRE a také velkou komunitu programátorů na oficiálním fóru.

⁵ **RPG** (zkratka z angl. *Role playing game*, česky nejčastěji překládané jako **hra na hr diny**) je druh hry, kde hráči zaujímají role fiktivních postav (*role-playing* – hraní v roli), které si podle daných pravidel vytvoří a za které v samotné hře jednají

⁶ LGPL - GNU Lesser General Public License (dříve GNU Library General Public License) nebo LGPL je licence svobodného softwaru, publikovaná Free Software Foundation. Byla navržena jako kompromis mezi silně copyleftovou licencí GNU General Public License (GPL) a permisivními licencemi.

7.1.2 CrystalSpace

Druhou knihovnou je knihovna CystalSpace [23]. CystalSpace (dále jen CS) je přenosná 3D knihovna, zahrnující mnoho komponent pro vývoj her a aplikací. Je užitečná i pro projekty nezahrnující 3D grafiku. CS je napsán v jazyce C++ a dovoluje přidávání dalších knihoven pro práci se speciálními komponentami. CS je více kategorizovaný jako 3D aplikační engine. Zahrnuje 3D renderovací jádro, ale také obsahuje přídavné funkce (pluginy) a externí funkce (moduly) jako je zvuk, vstupně výstupní operace a mnohé další. Moduly mohou být používány v aplikacích, ale pokud nejsou požadovány, tak se mohou ponechat bez užitku. CS se snaží integrovat veškeré komponenty, které by mohly být důležité a vhodné pro vývoj aplikací.

CS je šířen pod licencí LGPL, což znamená, že jsou dostupné i zdrojové kódy a může se použít knihovna i dělat zásahy ve zdrojovém kódu knihovny. LGPL umožňuje i komerční použití. Pro nás je důležité, že knihovna je volně dostupná.

CS poskytuje také kvalitní dokumentaci, bohužel však ne takové množství tutoriálů. Na rozdíl od OGRE nevyužívá fórum vývojářů, ale využívá tzv. „mailing list“. Podrobné srovnání je možné nalézt na [24].

Po vyzkoušení základní práce s oběma grafickými enginy, byl zvolen engine OGRE. Pro programátory, kteří nemají zkušenosti s grafickými enginy, je OGRE srozumitelnější a nápomocnější jsou i základní tutoriály. CS je velice dobrý a kvalitní nástroj a pro zkušené vývojáře. Pro programátory, kteří se jím zabývají a znají jeho funkce více do hloubky, je velice užitečný.

7.2 Fyzikální engine

7.2.1 Bullet

Bullet Physics [21] je profesionální otevřená knihovna detekce kolizí a dynamiky tuhého i pružného tělesa napsaná v C++. Knihovna je volně dostupná i pro komerční účely. Je šířena pod licencí ZLIB⁷. Podporuje standardní real-time simulaci fyziky. Pro spojení s enginem OGRE je dostupná komponenta OGRE-Bullet [26], která jednoduchým způsobem propojuje tyto dvě knihovny a usnadní tak práci při vývoji aplikace. [21]

7.2.2 ODE

ODE [25] je otevřená vysoce výkonná knihovna pro simulaci dynamiky tuhých těles. Je napsána v jazyce C++ a je šířena pod licencemi LGPL a BSD. Podporuje detekci kolizí a tření. Je vhodná pro

⁷ ZLIB - Jedná se o licenci svobodného software, kompatibilní s GPL. GPL je licence pro svobodný software, původně napsaná Richardem Stallmanem pro projekt GNU. GPL je nejpopulárnějším a dobře známým příkladem silně copyleftové licence, která vyžaduje, aby byla odvozená díla dostupná pod toutéž licencí. V rámci této filosofie je řečeno, že poskytuje uživatelům počítačového programu práva svobodného softwaru a používá copyleft k zajištění, aby byly tyto svobody ochráněny, i když je dílo změněno nebo k něčemu přidáno. Toto je rozdíl oproti permisivním licencím svobodného softwaru, jejímž typickým případem jsou BSD licence.

simulaci vozidel a prostředí virtuální reality. Pro spojení s engine OGRE je dostupná komponenta OGRE-ODE [27], která usnadní propojení grafické a fyzikální knihovny.

Pro tento projekt je zvolen fyzikální knihovnu Bullet. Pro použití v tomto druhu práce se jeví jako vhodnější, ale knihovna ODE by mohla být také použita.

Všechny zmíněné engine a knihovny mají podporu přímo pro Microsoft Visual studio [28], což je také jejich přednost pro vývojáře MS Windows. Pomocí připravených projektů se po nastudování základní funkčnosti a návodů dají přeložit a spustit základní vývojové prostředky.

7.3 Komponenty simulátoru

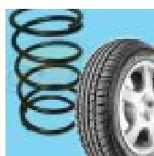
Simulátor 3D autoškoly je komplexní systém, který je ve svém výsledku propracovaný a závisí na mnoho aspektech. Celý systém je možné rozložit na dílčí problémy a zabývat se analýzou a návrhem jednotlivých komponent.

Dekompozice problému na podproblémy je využívána při mnoha návrzích softwarových produktů. Jedná se o rozdělení složitějšího problému na jednodušší problémy. Tyto podproblémy jsou lépe zvládnutelné. Dekompozicí vzniknou lépe zvládnutelné podsystémy, které se řeší samostatně. Při řešení každého podsystému se můžeme na tento podsystém podrobně zaměřit a vzniká tak kvalitnější návrh. Pokud je to nutné, můžeme dále zjemnit podsystém na menší celky a teprve ty řešit samostatně. Nevýhodou je, že podsystémy musí být vhodně spojeny a musí vzájemně spolupracovat tak, aby výsledkem byl plně funkční systém, který byl v prvním návrhu zamýšlen. Testovat by se měli jak jednotlivé podsystémy, tak i celý systém. [29]

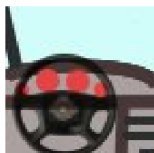
Systém 3D autoškoly je dekomponován na 3 části. První částí je vozidlo, druhou okolí a třetí pravidla silničního provozu. Každá část má 3 podsystémy, které spolu souvisejí. Všechny bloky jsou graficky i s popisem znázorněny níže. Spolupráce celého systému je v kapitole 7.4.



Model vozidel a jejich vlastnosti a chování v simulátoru. Vytvoření základního grafického vzhledu a zapouzdření všech komponent souvisejících s modelem vozidel, jako ovládání, pohyb, fyzika, kolize a dodržování pravidel.



Model fyzikálních vlastností daných knihovnou Bullet. Zahrnuje návrh detekce kolizí a dynamiku tuhého tělesa reprezentující vozidla a překážky v silničním provozu



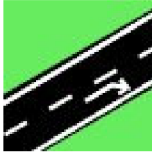
Model uživatelského rozhraní, návrh ovládání aplikace a vozidla, vzhled a ukazatele v interiéru vozidla. Návrh informačních oznámení o dodržování pravidel a pokyny pro jízdu.



Model prostředí budov a města. Definování kolizních objektů (obalových těles) pro realistické chování při detekci kolizí vozidel s budovou. Namodelování různých budov ve vhodném 3D modelovacím prostředí.



Model krajiny a zeleně ve městě. Návrh a vymodelování základního vzhledu terénu a jeho integrace do aplikace. Vytvoření přirozeného prostředí pomocí stromů a zeleně.



Model komunikací. Definování sítě silnic a ulic v simulátoru. Propojení s modelem detekcí kolizí a pravidel silničního provozu, reprezentovaných orientovaným grafem. Dopravní značení.



Definice pravidel silničního provozu. Návrh a implementace orientovaného grafu pro reprezentaci dodržování pravidel a modelování jízdy po silnici a v jízdních pruzích.



Detekce kolizí v simulátoru. Řešení okrajových podmínek a dodržování vzdálenosti vozidel řízených simulátorem.

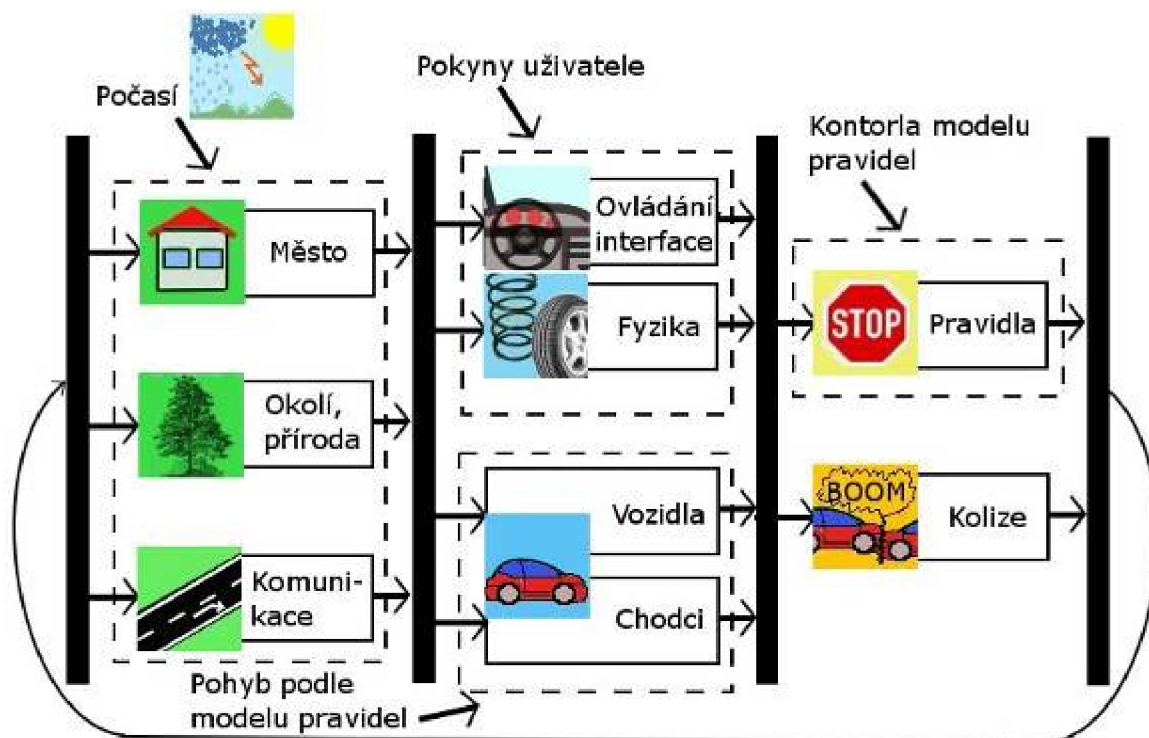


Model počasí v simulátoru. Základní vlastnosti – den/noc, jasno/zataženo/děšť, návrh chování fyzické stránky vozidla v různém počasí.

7.4 Popis funkčnosti

Bloky popsané v předešlé kapitole musí spolu navzájem spolupracovat a bude z nich sestavena celá aplikace. Funkčnost je rozdělena do tří základních kategorií. První kategorií jsou pevně dané objekty, které mohou být definovány předem, druhou částí jsou pohyblivé modely a poslední částí je společná spolupráce komponent dohromady a kontrola správného chování jednotlivých bloků.

1. Model krajiny, města a komunikací, definování dopravního značení
2. Interaktivní části města – vozilo uživatele, ostatní účastníci
3. Kontrola dodržování pravidel, detekce kolizí



Obrázek 7.1: Grafické schéma spolupráce podsystémů v 3D autoškole

7.5 Model uživatelského rozhraní

Důležitým prvkem návrhu celého simulátoru je bezpochyby uživatelské rozhraní. Rozhraní a ovládání vozidla a celé simulace musí být bezpochyby intuitivní a jednoduché a zároveň efektivní. Spojit tyto požadavky je nelehký úkol a v tomto simulátoru je navržen nový krok, který by mohl přispět k realistickému vnímání simulace. Podrobný rozbor existujících i nových možností je ukázán v prvních kapitolách této práce.

7.5.1 Uživatelské rozhraní

Pro ovládání simulátoru bude sloužit bezpochyby klávesnice a myš, protože to jsou standardní periférie používané u PC. Jedním z hlavních požadavků je možnost využití aplikace velkým počtem uživatelů. Není tedy primárním cílem práce připojování speciálních monitorů, volantů a podobných zařízení.

Dále lze ovládání rozdělit na přednastavení voleb pro simulaci a samotné ovládání simulace při jízdě automobilem. První část nemá smysl příliš rozebírat, protože každý uživatel počítače dokáže navolit pomocí myši a intuitivního menu několik základních voleb jako výběr lekce, obtížnosti apod. Mnohem zajímavější a důležitější je část ovládání simulace při řízení automobilu.

K ovládání simulace při jízdě bude uživatel využívat jak klávesnici, tak i myš, což bude přispívat k již zmíněné vyšší realističnosti. Při prvním spuštění může být ovládání nezvyklé, pro hráče závodních her, ale v simulátoru autoškoly nejde o závody, ale o výuku.

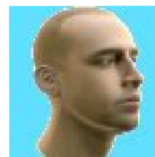
Ovládání se dá rozdělit na dvě oddělné komponenty.

1. Myš bude ovládat pohled uživatele, který tak bude mít větší rozhled a možnost sledovat situaci před sebou i kolem sebe.
2. Klávesnice bude sloužit pro pohyb vozidla požadovaným směrem

Výsledné spojení bude pohyb vozidla s možností pohledů řidiče a zpětná kontrola na přístrojích palubní desky a výhledu z vozidla.



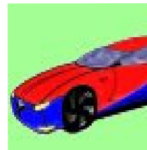
Ovládání myší



Směr pohledu



Ovládání klávesnicí

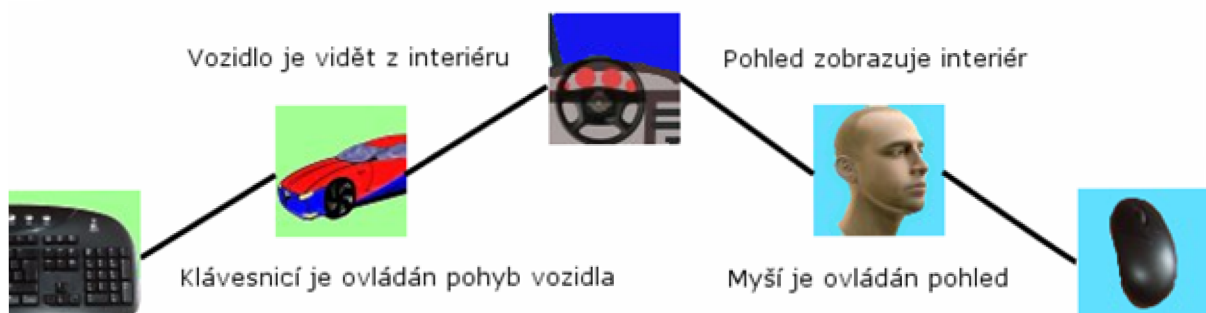


Pohyb vozidla



Výsledné sloučení komponent pohybu a zpětná kontrola výhledem z automobilu a pomocí ukazatelů na přístrojové desce

Spolupráci ovládání aplikace lze vyjádřit následujícím propojením komponent



Obrázek 7.2: Schéma spolupráce komponentů uživatelského návrhu

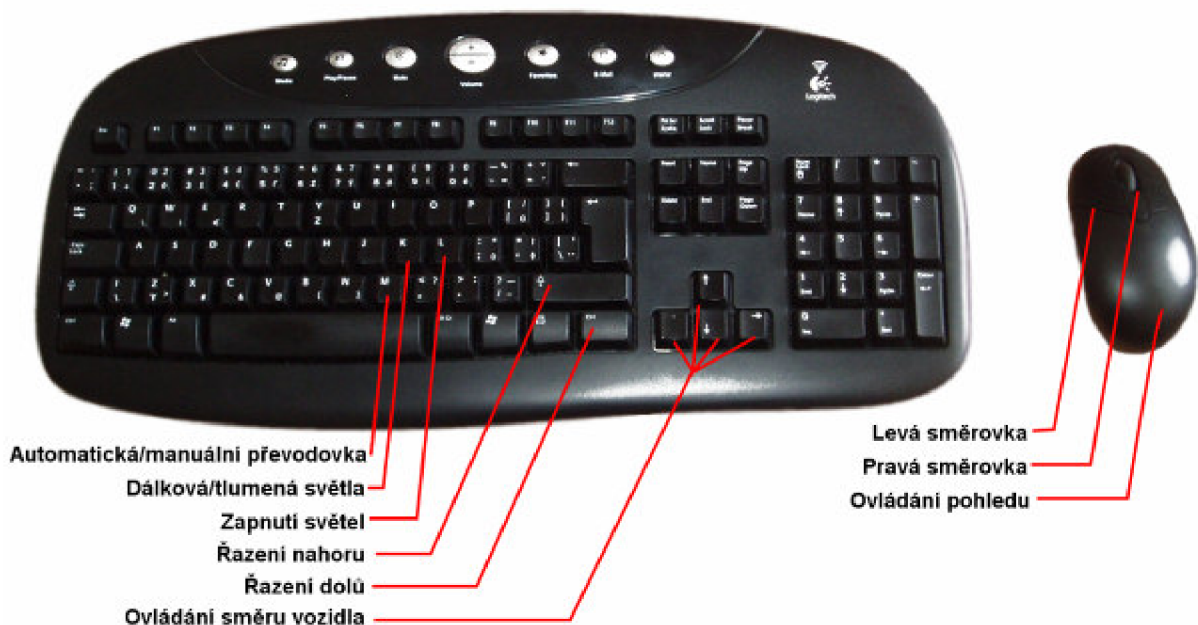
7.5.2 Myš

Pohybem myši ovládá uživatel hlavu řidiče. Tím ovlivní směr jeho pohledu. Na pravém a levém tlačítku myši budou umístěny směrová světla. Je to navrženo z důvodu poměrně častého používání a je tedy třeba je mít na lehce dostupném místě.

7.5.3 Klávesnice

Pohyb automobilu je navrhnout na obvyklé šipky. Pro zatočení vlevo je levá šipka, pro zatočení vpravo je pravá šipka. Plyn se přidá šipkou nahoru a brzda je šipkou dolů. Pro změnu směru jízdy budou sloužit klávesy levý „shift“ a levý „ctrl“. Pokud bude zapnuta automatická převodovka - tedy samočinné řazení, bude klávesa „shift“ sloužit k zapnutí pohybu vozidla vpřed a „ctrl“ pro zapnutí zpětného chodu. Při zapnuté manuální převodovce budou sloužit klávesy jako sekvenční řazení. Klávesou „shift“ se zvýší rychlostní stupeň a klávesou „ctrl se sníží rychlostní stupeň. Zpátečka bude umístěna nejnižše. Rychlosti půjdou tedy v tomto pořadí. Zpátečka, neutrál, první rychlost, druhá rychlost, atd. Zvuková siréna by měla být na viditelném místě, a proto jsem zvolil klávesu „mezerník“. Toto jsou nejdůležitější prvky pro ovládání při řízení. Před jízdou bude možné ještě nastavit základní funkce. Klávesou „M“ manuální převodovka, nebo převodovka automatická. Klávesou „L“ zapnutí či vypnutí světel.

Popis ovládání je znázorněn na obrázku 7.3.



Obrázek 7.3: Návrh ovládání aplikace pomocí myši a klávesnice

7.5.4 Tabulka s ovládacími funkcemi simulátoru

Myš	
horizontální pohyb	pohyb pohledu doleva, doprava
vertikální pohyb	pohyb pohledu nahoru, dolů
levé tlačítko	směrové světlo levé
pravé tlačítko	směrové světlo pravé
Klávesnice	
šipka vpravo	zatočení vpravo
šipka vlevo	zatočení vlevo
šipka nahoru	akcelerace
šipka dolů	zbrzdění
pravý ctrl	podřazení rychlosti
pravý shift	zařazení rychlosti
L	zapnutí/vypnutí světel
M	manuální/automatická převodovka
K	Tlumená / dálková světla

Tabulka 7.1: Popis ovládání 3D autoškoly

8 Implementace simulátoru

Projekt autoškoly je rozsáhlou aplikací, a proto je rozdělen do několika hlavních funkčních modulů, které jsou spolu navzájem propojeny.

- Grafický modul (vozidlo, město),
- modul popisující pohyb,
- modul popisující pravidla silničního provozu.

Grafický modul má dvě hlavní části. První je model auta s přístrojovou deskou a aktivními ukazateli na přístrojové desce. Těmi ukazateli jsou kontrolky, ukazatel rychlosti a ukazatel otáček. Pohyblivé komponenty musely být modelovány odděleně, aby mohly být animovány.



Obrázek 8.1: Model automobilu

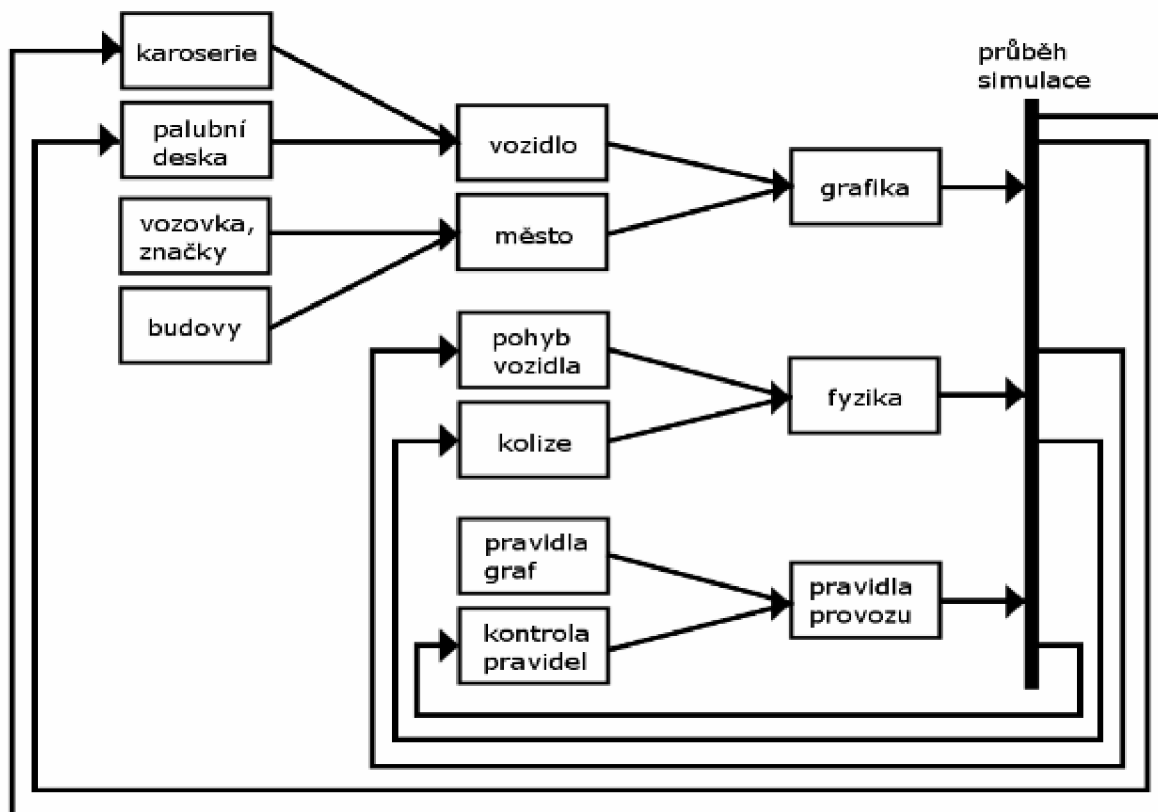
Druhou částí grafického modulu je město skládající se z domů, vozovky a dopravních značek. Tato pro uživatele viditelná část je však pouze malou částí celé aplikace.



Obrázek 8.2: Model města

Druhý modul popisuje pohyb a chování vozidla. Jedná se o fyzikální model simulátoru, včetně detekce kolizí objektů.

Protože v autoškole musí být zaznamenána a kontrolována pravidla silničního provozu, je vytvořen systém zápisu pravidel do orientovaného grafu a jejich následná kontrola. Toto je součástí třetího modulu.



Obrázek 8.3: Propojení funkčních bloků

Cílem tohoto projektu není vytvoření realistické grafiky s využitím pokročilejších metod sledování paprsku apod., ale vytvoření edukační aplikace, která spojuje více oblastí dohromady. Jako důležitý prvek je názornost a jednoduchost. Zobrazování stínů a odlesků je reálné v našem světě., ale v tomto projektu je kladen důraz na jednoduchost a přehlednost. Právě v autoškole je nutné, aby si začátečník osvojil správné návyky při řízení a sledování pravidel dopravního provozu v prostředí, kde ho nerozptylují jiné aspekty, aby potom v reálném prostředí dokázal rychle a správně reagovat.

V následujících kapitolách budou popsány postupy a metody využívané v simulátoru autoškoly.

8.1 3D modely v simulátoru

Pro vytváření modelů v celém simulátoru byl použit modelovací nástroj Blender [30], který je volně dostupný a existuje k němu řada tutoriálů pro začátečníky i pokročilé uživatele. Blender umožňuje vytvářet nové modely i modifikovat existující modely a nanášet na ně textury. Podporuje také načítání a ukládání velkého množství souborových formátů.

Grafický engine OGRE, který využívá tento simulátor, podporuje načítání modelů ve formátu *.mesh a materiálových skriptů ve formátu *.material. Formát „mesh“ není běžným standardem, ale pro exportování těchto formátů existuje plugin „OGRE Mesh Exporter“ [31] do Blenderu, který vytváří přímo požadovaný formát *.mesh. Příklad použití a návod instalace plug-in modulu je dostupný na [31]. Pro přímý převod modelů formátu „3ds“ do formátu „mesh“ lze použít také aplikaci „3ds2mesh“ [32]. Jedná se o konzolovou aplikaci, kde se program spustí s parametrem jména „3ds“ modelu a parametrem jména výstupního modelu. Exportovací program vytvoří „mesh“ soubor a soubor s materiálem.

Model karoserie auta byl převzat z volně dostupných modelů na [33] a upravený podle dalších potřeb. Úpravy bylo nutné provést na podvozku 3D modelu, aby se mohly spojit s pohyblivými částmi kol v aplikaci. Kola jsou načteny jako samostatný model, aby se mohla otáčet. Dále byly přidány objekty pro zpětná zrcátka.

Samostatnou částí bylo namodelování palubní desky, která je upravena pro potřeby aplikace. Jsou vytvořeny texturovací souřadnice a aplikována textura palubní desky [34].



Obrázek 8.4: Model palubní desky

Volant byl vytvořen jako samostatná komponenta, která je přidána pomocí metod OGRE, stejně jako kola vozidla. Detaily na palubní desce, kterými jsou kontrolky a ukazatele, byly vytvořeny zvlášť a jejich vizualizace je vytvořena metodami enginu OGRE.

Dopravní značky byly vymodelovány v Blenderu. Nemusí být však modelována každá dopravní značka, ale stačí pouze základní tvary. Základní tvary značek jsou:

- kruhová - zákazové a příkazové značky,
- trojúhelníková s hranou dole – výstražné značky,
- trojúhelníková s vrcholem dole – značky upravující přednost,
- osmiúhelník - značky upravující přednost,
- čtverec postavený na vrchol - značky upravující přednost,
- obdélníkové malé – informativní značky,
- čtvercové – informativní značky,
- obdélníkové velké – informativní značky.

Pro každý z těchto modelů značek byly vytvořeny texturovací souřadnice. V aplikaci je pomocí metody „Ogre::SetMaterialName()“ zobrazena požadovaná textura dopravní značky. 3D modely jsou tedy univerzální.

Textury pro dopravní značky byly vytvořeny podle vyhlášky o pozemních komunikacích [35] a upraveny v grafickém programu GIMP [36].

8.1.1 Načtení modelů v enginu OGRE

Pro načítání vytvořených modelů v grafickém enginu Ogre [22] je využito metody z manažeru scény „Ogre::createEntity()“. Funkce má parametry „Ogre::createEntity(string EntityName, string MeshName)“ a vytvoří objekt „Ogre::Entity“. Tato metoda načte „mesh“ model i s příloženým materiálem. Materiál lze specifikovat i explicitně pomocí metody „Ogre::setMaterialName(string MaterialName)“. Objekt Entity se přidá ke scéně metodou „Ogre::attachObject(Entity)“. Pro načítání objektů je možné tvořit stromovou strukturu. Pomocí metody „Ogre::addChildNode()“ se vytvoří nový objekt „Node“ z již existujícího objektu „Node“ a tomuto objektu se následně přiřadí objekt „Entity“.

8.1.2 Model vozidla

Model auta má dvě části. První částí je celý model spojený s pohyblivým objektem „Node“ ve scéně, pomocí kterého se pohybuje vozidlo dle pokynů uživatele. Na tento objekt „Node“, ve vhodné stromové struktuře, jsou při inicializaci navázány všechny modelové prvky automobilu – karoserie, palubní deska, volant, ukazatele, kontrolky a zpětná zrcátka. Ve spuštěné simulaci se pohybuje s tímto objektem jako s celkem a není proto nutné rozlišovat jednotlivé části auta. Pohybuje se tedy s rodičovským objektem.

Uživatelské rozhraní, kterým je pohled na přístrojovou desku, tvoří pohyblivé části, jejichž pohyb je nutné ovládat zvlášť. V každém snímku aplikace dojde jak pohybu celého objektu automobilu, tak k pohybu aktivních komponent. Aktivními komponentami jsou ručičky rychlosti a otáčkoměru, kontrolky na přístrojové desce a volant. V atributech třídy zahrnující komponenty auta „Vehicles_Demo“ jsou ukazatele na tyto aktivní prvky. Příslušný prvek je vizualizovaný podle aktuálních hodnot v simulaci - otáčky motoru, rychlost automobilu, natočení kol a zapnutých či vypnutých ovládacích prvků. Pohyby jsou zapouzdřeny v metodách „Vehicles_Demo::moveComponenets()“ a „Vehicles_Demo::flashSignUpdate()“.

Dalšími objekty jsou 3 zpětná zrcátka. Na tyto objekty jsou aplikovány interní textury, do kterých se renderuje pohled z kamery směřované dozadu za automobil. V prvním kroku je vytvořen objekt kamery „Ogre::Camera“ a nastaveny parametry ořezávacích rovin a natočení pohledu. V dalším kroku je vytvořen objekt manuální textury metodou „Ogre::TextureManager::getSingleton().createManual()“. Do zobrazovací oblasti této textury - viewportu - je přidán výstup z vytvořeného objektu kamery. Ve třetím kroku se na model zpětných zrcátek v automobilu nanese materiál. V klasickém případě obsahuje externí materiálový skript informace o textuře. V případě zpětných zrcátek se vytvoří vnitřní reprezentace materiálu, který ponese informace o interní textuře. Materiál se vytvoří metodou „MaterialManager::getSingleton().create()“ a v parametru se zadá vytvořená interní textura. Nyní je možné materiál přidat obvyklým způsobem „setMaterialName()“. Texturovací souřadnice obsahuje model objektu, na který je aplikován materiál. Texturovací souřadnice se vytvoří se při modelování objektu např. v Blendru.

8.2 Pohyb pohledu řidiče

Celá scéna je zobrazena kamerou, s nastavenými parametry ořezávacích rovin a směru natočení kamery. Protože je třeba s objektem kamery pracovat, je ukazatel na objekt kamery atributem třídy „*Vehicles_Demo*“. V návrhu uživatelského rozhraní simulátoru, je jako jeden prvek zvoleno ovládání pohledu pomocí myši, aby bylo dosaženo realistické simulace rozhlížení řidiče. Tento způsob je zvolen s ohledem na typ využití simulátoru na běžných počítačích či noteboocích, kde je k dispozici jeden monitor, zobrazující celou scénu. Pohybem kamery pomocí myši je docíleno plynulého pohybu hlavy řidiče.

V aplikaci je pomocí zachytávání událostí pohybu myši získávána souřadnice relativního pohybu v rovině. Tyto souřadnice jsou přepočítány na úhel natočení. Kamera se otáčí v rovině XZ. Samotné natočení kamery je realizováno metodou „*mCamera->yaw()*“.



Obrázek 8.5: Pohled z kabiny řidiče

Pohled kamery je možné přepnout klávesou „C“. Základní pohled je z kabiny automobilu. Druhý implementovaný pohled je tzv. pohled třetí osoby. Je to pohled zobrazující automobil zezadu. Z tohoto pohledu je vidět celý automobil z větší vzdálenosti, viz obr. 8.6.

Oproti pohledu řidiče (pohled z kabiny), je u tohoto pohledu vypnuta možnost natáčení pohledu myši, protože to již nesimuluje otáčení hlavy řidiče a také je při tomto pohledu dostatečný rozhled i na obě strany vozidla.



Obrázek 8.6: Pohled za automobilem – pohled třetí osoby

8.3 Fyzika

Pro popis fyzikálních vlastností tohoto simulátoru byl použit fyzikální engine Bullet. Jedná se o rozsáhlou knihovnu obsahující mnoho tříd a metod pro práci s objekty. Fyzika použitá v tomto simulátoru se dá zařadit do dvou kategorií:

- modelování pohybu vozidla, dynamika pohybu, odpružení podvozku,
- modelování tuhých těles, domy, dopravní značky a jejich kolizní tělesa.

Pro popsání fyzikálních vlastností vozidla jsou využity tři základní třídy:

- „*OgreBulletCollisions::BoxCollisionShape*“ popisuje kolizní obal karoserie, který je zahrnut do výpočtu kolizí objektů v celé aplikaci.
- „*OgreBulletDynamics::WheeledRigidBody*“ je třída, ve které jsou popsány vlastnosti karoserie, jako je váha, pozice, velikost.
- „*OgreBulletDynamics::VehicleTuning*“ zahrnuje nastavení dynamických vlastností jako například, tuhost pružin a tlumičů.

Objekt třídy „*OgreBulletDynamics::RaycastVehicle*“ popisuje celý automobil a je sestaven z předcházejících tří objektů. Na takto sestavený objekt jsou metodami třídy „*RaycastVehicle*“ přidány kola a model karoserie. Vytvoření modelu karoserie bylo popsáno v předcházející kapitole.

8.3.1 Kolize

Pro popis kolizních objektů tělesa se v knihovně Bullet spojené s grafickým enginem Ogre používá třída „*OgreBulletCollision*“. Třída umožňuje zadávání různých obalových těles, jako kvádry, koule, válce, ... V tomto simulátoru je využito kvádrů pro popis kolizních těles vozidla a domů. Pro reprezentaci dopravní značky je použito kolizního objektu válce. Přidáním kolizního tělesa objektu, se obal s objektem sváže a pohybuje-li se objekt ve scéně, pohybuje se s ním i jeho kolizní těleso. Všechna kolizní tělesa jsou přidávána do seznamu kolizních objektů. Z tohoto seznamu jsou potom počítány kolize.

8.3.2 Nastavení parametru auta

Pro popis fyzikálních vlastností jsou na začátku modulu „*Vehikle_Demo.cpp*“ nadefinovány proměnné, které popisují vlastnosti vozidla. Parametry jsou:

- jeden typ parametrů je využit při vytváření objektu reprezentujícího automobil,
- druhý typ je využit při ovládní vozidla v reálném čase.

Prvním typem parametrů jsou:

- tuhosti pružin a tlumičů,
- délka pružin,
- třecí síla,
- tlak pneumatik,
- rozvor přední/zadní nápravy,
- šířka pneumatik.

Tyto parametry ovlivňují chování vozidla. Jsou použity při vytvoření objektu reprezentujícího automobil.

Druhým typem parametrů jsou hodnoty využívané dynamicky při řízení vozidla. Jsou to parametry:

- rychlosti natočení kol,
- maximální úhel natočení kol,
- výkon motoru,
- síla brzdění.

Jsou využity při výpočtu pohybu. Zatáčení vozidla je realizováno pomocí šipek na klávesnici. Při stisknutí klávesy je přičtena, či odečtena hodnota natočení kol. Podle velikosti této hodnoty se budou kola natáčet pomaleji či rychleji. Hodnota natočení kol se bude inkrementovat / dekrementovat až do hodnoty parametru maximálního natočení kol. Po výpočtu natočení je úhel aplikován na kolo automobilu metodou „*setSteeringValue()*“. Tato metoda aplikuje úhel na každé kolo zvlášť, což například umožní popis složitějšího modelu automobilu, kde je více zatáčecích náprav apod. U tohoto simulátoru je namodelována klasická koncepce, kde je úhel natočení aplikován pouze na levé a pravé přední kolo.

Při stisknutí klávese pro pohyb dopředu se aplikuje hodnota síly motoru. Síla motoru se aplikuje na každé kolo zvlášť, stejně jako u natáčení a je pro to použita metoda „*applyEngineForce()*“. Hnací síla se však musí vypočítat podle otáček motoru a zařazeného rychlostního stupně. Pro výpočet slouží předdefinovaná tabulka, která udává násobící konstantu síly motoru podle otáček a zařazeného rychlostního stupně. Pokud je zařazen vyšší rychlostní stupeň a otáčky motoru jsou v nízkých hodnotách, bude se sice automobil pohybovat, ale motor nemá dostatečný výkon a jeho síla je tedy menší. Při optimálních otáčkách vozidlo zrychluje s plným výkonem. Předdefinovanou tabulku je možné jednoduše upravit a dosáhnout tak jiných vlastností motoru, případně se může v dalších rozšířeních této práce vytvořit více vozidel s různými charakteristikami výkonu motoru.

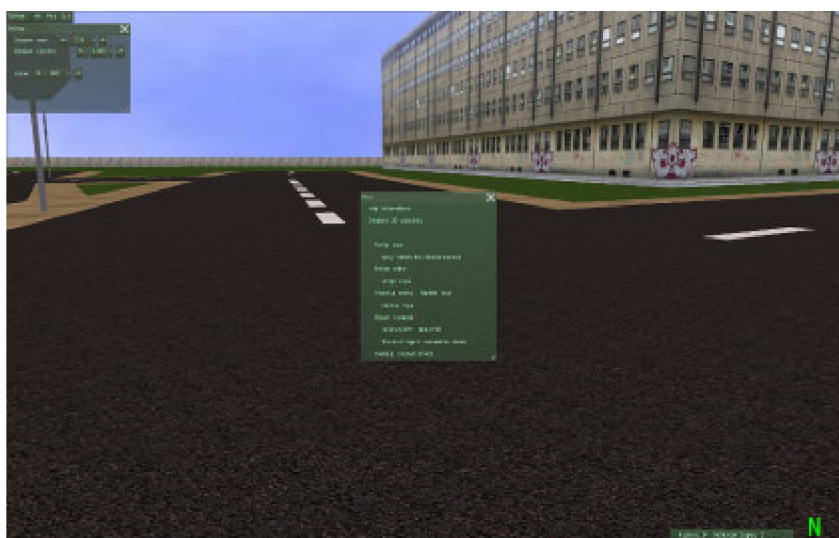
Pro aplikaci automatické převodovky je využito informace o otáčkách motoru. Jsou předdefinovány dva parametry. Jeden pro podřazování na nižší rychlostní stupeň při nízkých otáčkách a druhý pro parametr limitu otáček, při kterém se zařadí automaticky vyšší rychlostní stupeň.

Pro vizualizaci natočení volantu je definován maximální úhel natočení a pohyb vypočítán pomocí úhlu natočení kol.

8.4 Menu aplikace

V horní části okna spuštěné aplikace je umístěno uživatelské menu, které umožňuje nastavení základních vlastností chování vozidla v simulaci a běžné uživatelské volby jako zobrazení nápovědy a vypnutí aplikace.

Pro vytvoření menu je využita třída „*BetaGUI*“. Tato třída poskytuje metody pro vytvoření základního uživatelského rozhraní. Uživatelské rozhraní se vytváří ručně přidáváním nových objektů k hlavnímu objektu uživatelského rozhraní. Grafický návrh menu, jako například ve VS, zde není podporován. Základním objektem je „*BetaGUI::Window*“, ke kterému se přidávají nová okna.



Obrázek 8.7: Menu aplikace

V menu tohoto simulátoru jsou vytvořeny 4 kategorie v levém horním rohu.

Settings – nastavení základních parametrů chování automobilu pomocí posuvníků typu „*BetaGUI::RealButton*“. Tomuto objektu se nastavuje při vytvoření maximální a minimální hodnota, jemné dělení kroku a hrubé dělení kroku, o který se hodnota inkrementuje či dekrementuje. První posuvník v záložce settings umožňuje nastavit poloměr zatáčení automobilu. Druhý posuvník nastavuje rychlost zatáčení, tedy krok s jakým se zvyšuje natočení kol při povelu zatočení volantem. Třetí posuvný bar nastavuje výkon motoru automobilu, od kterého se odráží hodnota zrychlení vozidla v simulaci.

Další položkou v menu je možnost zobrazení nápovědy, která popisuje základní ovládání simulátoru. Menu je vytvořeno jako nové okno s poloprůhledným pozadím, které se zobrazí uprostřed hlavního okna aplikace. Text menu je definován v hlavičkovém souboru „*Vehikle_Listener.h*“. Popis ovládání simulátoru je také popsán v uživatelské příručce.

Uživatel má také možnost zobrazit nebo skrýt informační lištu zobrazenou ve spodní části okna, která zobrazuje aktuální informace při běhu simulátoru. Posledním prvkem menu je tlačítko pro vypnutí aplikace. Tu je také možno vypnout přes tradiční klávesu „*esc*“

8.5 Model města

Nedílnou součástí simulátoru je okolní prostředí, které tvoří město skládající se z budov, silnic, chodníků, travnatých ploch atd. Zobrazování města je vytvořeno jednoduše pomocí třídy „Ogre::manualObject“. Metodami této třídy se mohou zobrazit jednoduché objekty, jako jsou trojúhelníky, obdélníky, kvádry apod. Objekty se vytváří podle zadávaných souřadnic v pořadí definovaném parametry volané metody. Na „manual objekt“ je možné přidat texturní souřadnice a metodou „Ogre::setMaterialName“ potom přidat požadovaný materiál s texturou.

Ve městě je jako podkladová plocha zvolen objekt s texturou travnatého povrchu, přes který jsou zobrazeny ostatní objekty. Pro domy je zvoleno několik textur, které se náhodně načítají, na předdefinované souřadnice zobrazovaných budov. Město tedy nevypadá stále stejně při opakovaném spuštění simulace. Textury pro město byly získány na [37].



Obrázek 8.8: Model města - zobrazení budov

Nejdůležitějším prvkem jsou bezesporu silnice, po kterých se má vozidlo pohybovat a dopravní značky, které definují pravidla silničního provozu. Silnice jsou načítány při průchodu orientovaným grafem, který bude popsán v následujících kapitolách. Důvody pro zavedení orientovaného grafu pro reprezentaci silnic a pravidel silničního provozu byly popsány v teoretické části této práce v kapitole 6.2. Graf v sobě obsahuje souřadnice bodů, které reprezentují silniční síť ve 2D prostoru. Při průchodu jsou tyto body načítány a na načtené souřadnice jsou zobrazeny vozovky s chodníky. Vozovka je realizována pomocí manuál objektu, zobrazeného v rovině XZ a na její povrch je nanášena textura asfaltu. Silnice jsou ohraničeny chodníky, které jsou zobrazovány zvlášť. Načítají se ze stejných souřadnic jako vozovka, ale jejich souřadnice jsou přepočítány podle šířky vozovky.

8.6 Graf silniční sítě

Silnice v simulátoru jsou reprezentovány orientovaným grafem. Orientovaný graf zajišťuje uložení pravidel silničního provozu ke každé hraně a uzly spojující jednotlivé hrany mají ve svém atributu souřadnice bodu na mapě, podle kterých jsou zobrazovány silnice.

Graf je tvořen ze tří modulů. Základní třída „*RoadGraph*“ reprezentuje celou grafovou strukturu a obsahuje metody pro manipulaci s grafem. Obsahuje seznamy uzlů a seznamy hran celé grafové struktury. Další dva moduly tvoří třída „*GraphNode*“, která obsahuje objekty reprezentující uzly grafu a metody pro manipulaci s nimi a třída „*GraphEdge*“, která obsahuje objekty hran grafu a metody pro manipulaci s nimi. Protože ve třídě „*GraphEdge*“ se využívá metod a proměnných třídy „*GraphNode*“ a opačně, aby mohl být graf provázán, musí být v modulu „*GraphNode.h*“ includován modul „*GraphEdge.h*“ přes definici „`#pragma once`“ a stejně tak i v „*GraphEdge.h*“, aby nedošlo k zacyklení při linkování knihoven.

8.6.1 Třída „*GraphNode*“

Třída definuje uzly grafu. Jako atributy obsahuje souřadnice uzlu, vektor hran, které do uzlu vstupují a vektor hran, které z uzlu vystupují. Při vytvoření objektu se jako parametry konstrukturu předají souřadnice uzlu. Tyto souřadnice, lze pak získat metodami „`getX()`“ a „`getY()`“. Pro přidání vstupních a výstupních hran jsou definovány metody „`addInEdge()`“ a „`addOutEdge()`“, které jako parametr obsahují ukazatel na objekt hrany. Metoda „`isNode()`“ vrací true v případě, že souřadnice zadané v parametru funkce se shodují se souřadnicemi tohoto uzlu. Jinými slovy, metoda objektu zjistí zda souřadnice v parametru patří tomuto objektu.

8.6.2 Třída *GraphEdge*

Atributy třídy jsou pravidla silničního provozu, která jsou charakteristická pro každou hranu. Pro definování pravidel jsou vytvořeny tyto proměnné:

- speed - nejvyšší povolená rychlost,
- enable - povolení vjezdu na hranu – (zákaz vjezdu, jednosměrka),
- standing - zákaz zastavení nebo stání,
- stop - stůj, dej přednost v jízdě,
- stophalf - dej přednost v jízdě,
- directOrder - přikázaný směr jízdy,
- edgeEnable – hrana, ze které je přikázán směr.

Dále atributy obsahují ukazatel na objekt uzlu, ze kterého hrana vystupuje a ukazatel na objekt uzlu, do kterého hrana vstupuje a také ukazatel na objekt hrany vedoucí opačným směrem. Jedná se vlastně o silnici, kterou reprezentují oba jízdni směry. Objekt hrany má také při svém vytvoření vypočítán atribut směru, což je úhel v rovině XZ souřadného systému XYZ ze simulovaného světa. Při vytvoření objektu hrany se v konstrukturu naplní výše uvedené atributy. Pravidla silničního provozu jsou nastavena na výchozí hodnoty, které se po načtení dopravních značek u příslušných hran změní.

Hodnoty nastavené při vytvoření hrany:

- Speed - 50km/h (normální rychlost ve městě),
- enable - true (všechny hrany mají povolen vjezd),
- standing - 0 (není zde žádný zákaz),
- stop – false (není zde přednost),
- stophalf – false (není zde přednost),
- directionOrder – 0 (žádný přikázaný směr),
- edgeEnable – NULL (neukazuje na žádnou hranu).

8.6.3 Třída RoadGraph

Obsahuje metody pro práci s grafem, které se využívají při sestavování a procházení grafu a při kontrole pravidel silničního provozu. Třída obsahuje seznam všech uzlů grafu, seznam všech hran v grafu, proměnnou pro uchování hrany, která byla detekována jako nejbližší k určité pozici (nejčastěji k pozici automobilu) a předposlední nejbližší hrana. Stejně tak obsahuje proměnné pro uchování posledního a předposledního nejbližšího uzlu k určité pozici. Pro možnosti optimalizace je definován seznam nejbližších hran k nějaké hraně a seznam nejbližších uzlů k nějakému uzlu. Tyto seznamy slouží pro zrychlení prohledávání grafu za předpokladu, že vozidlo se pohybuje po grafu. Potom není nutné procházet celý graf, ale pouze tyto menší seznamy možných následníků.

Metody ve třídě RoadGraph [21]

Metoda „*addNode()*“ vytvoří nový objekt uzlu a přidá ho do seznamu. Ukazatel na nově vytvořený objekt vrací jako návratovou hodnotu.

Pro zjištění existence nějakého uzlu je vytvořena metoda „*isExistNode()*“. V této metodě se nachází cyklus, který projde přes prvky seznamu a pokud nalezne uzel s hledanými souřadnicemi, vrací hodnotu „*true*“. Tato metoda je využita při sestavování grafu, kdy může nastat případ, že se na stejnou souřadnici v reprezentaci silniční sítě narazí vícekrát, například při křížení silnic. V grafu je tento stav samozřejmě zaznamenán pouze jedním objektem uzlu.

Pro vytvoření spojení je definována metoda „*createConnect()*“. Tato metoda má jako parametry souřadnice dvou uzlů, mezi kterými se vytváří spojení. Nejprve se zjistí, zda uzly v grafu již existují. Uzly mohou existovat, pokud se vytváří propojení na křižovatce. Pokud neexistují, vytvoří se nové objekty uzlů metodou „*addNode()*“. Mezi uzly se vytvoří objekt hrany a vytvoří se příslušná propojení. Nové hraně se předá vstupní a výstupní uzel a uzlům se do jejich seznamu přidá hrana z uzlu vystupující a druhému hrana do uzlu vstupující. Do seznamu všech hran v grafu se přidá nově vytvořený objekt hrany.

Metoda „*searchNearNodeAll()*“ dostane jako parametry souřadnice v rovině a nalezne nejbližší uzel grafu k tomuto bodu. Tato metoda je využita, pro nalezení nejbližšího uzlu grafu k aktuální pozici vozidla v simulaci. Metoda projde seznam všech uzlů v grafu a pro každý vypočítá vzdálenost dvou bodů v rovině podle vztahu 8.1. Hledá se minimální vzdálenost bodu a uzlu.

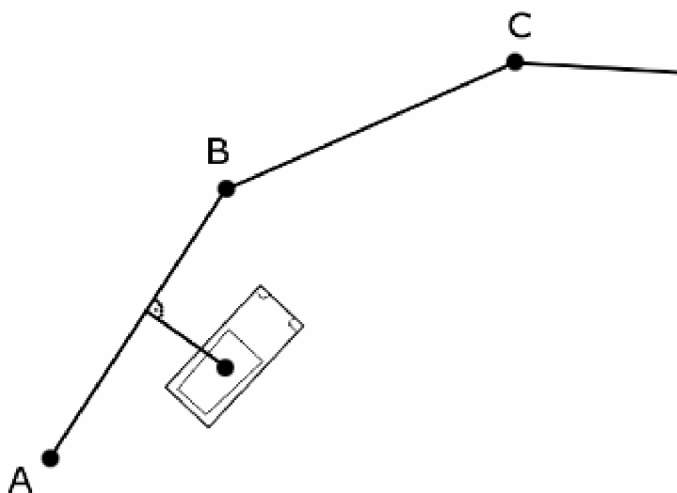
$$v = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (8.1)$$

Protože prohledávání všech uzlů grafu může být časově náročné, je k dispozici i optimalizovaná metoda, která vyhledává nejbližší uzel z užšího seznamu vhodných kandidátů. Pro tento účel je definována metoda „*setNearestNodes()*“, pro sestavení seznamu vhodných kandidátů.

Tato metoda předpokládá, že je nastaven atribut „*nearNode*“ – poslední nejbližší uzel. Vhodnými kandidáty jsou ty uzly, které jsou přes nějakou jednu hranu spojeny s tímto uzlem. Projde se tedy seznam hran vystupujících a vstupujících do tohoto uzlu a uzly z opačných konců hran jsou zařazeny do užšího výběru.

Metoda prohledávající užší seznam uzlů se jmenuje „*searchNearNode()*“. Tato metoda musí mít ošetřeny stavy, kdy není definován atribut „*nearNode*“ a v takovém případě se zavolá metoda pro prohledání všech uzlů grafu. Po nalezení nejbližšího uzlu se zjistí, zda je nově nalezený uzel shodný s uzlem v atributu „*nearNode*“. Pokud je nalezený uzel jiný, nastaví se atribut „*nearNode*“ na tento uzel a sestaví se nový seznam vhodných kandidátů.

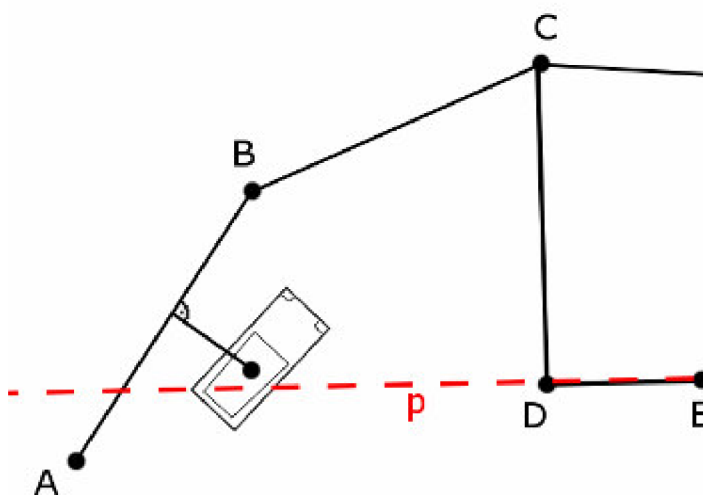
Pro určení hrany, po které se pohybuje vozidlo, je definována metoda „*searchNearEdgeAll()*“. Tato metoda využívá algoritmu pro stanovení vzdálenosti bodu od přímky. Přímka, která má od bodu reprezentujícího vozidlo nejmenší vzdálenost viz obr. 8.9, je hranou, po které se vozidlo pohybuje.



Obrázek 8.9: Nalezení nejbližší hrany

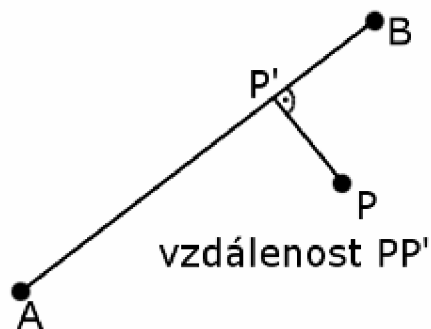
Metoda výpočtu vzdálenosti bodu od přímky nejprve vypočítá kolmý průmět bodu na přímku, určí bod na přímce a následně již vypočítá vzdálenost mezi dvěma body podle vztahu 8.1.

Tento postup je na první pohled správný, ale hrana je pouze úsečka a tento fakt musí být zohledněn. Je nutné určit vzdálenost bodu od úsečky. Mohlo by totiž dojít ke stavu, kdy přímka nejbližší vozidlu má úsečku, ležící na ní, úplně jinde. Situaci vystihuje obr. 8.10. Přímka „*p*“ je nejbližší vozidlu, ale hrana „*DE*“ leží v mnohem větší vzdálenosti.

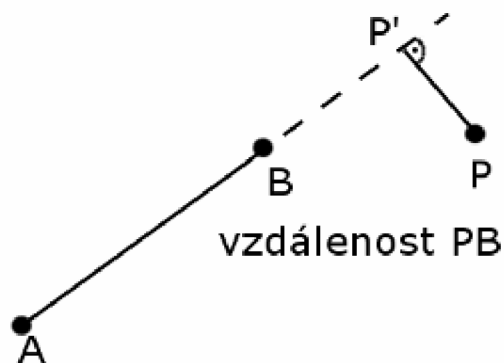


Obrázek 8.10: Vzdálenost vozidla od přímky - nesprávná interpretace

Zjištění vzdálenosti bodu od úsečky lze zajistit upravením algoritmu pro určování vzdálenosti bodu od přímky. Nejprve se určí kolmý průmět P' bodu P na přímku a následně se určí, zda bod P' leží na úsečce mezi body AB . Pokud leží mezi těmito body, lze vypočítat vzdálenost PP' - obr. 8.11. Pokud leží mimo přímku za bodem B , určuje vzdálenost PB - obr. 8.12, obdobně pro bod A .

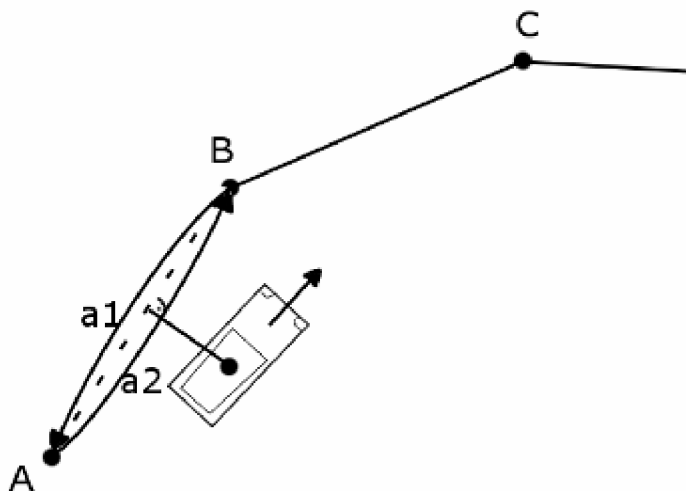


Obrázek 8.11: Průmět bodu P na hraně



Obrázek 8.12: Průmět bodu P mimo hranu

Nalezená hrana musí být ještě upřesněna, protože vzdálenost je stejná jak pro hranu ve směru vozidla tak i pro hranu opačnou viz obr. 8.13. Proto je nutné atribut směru hrany porovnat se směrem vozidla a vybrat správnou hranu.



Obrázek 8.13: Určení správné hrany podle směru

Směr hrany není přesně totožný se směrem vozidla jak je vidět na obrázku 8.13. Pro určení směru se tedy porovnává rozdíl úhlů hrany „a1“ a úhlu, ve kterém se pohybuje automobil a úhlu hrany „a2“ a směru automobilu. Protože například hrana vedoucí pod úhlem 2° a vozidlo jedoucí pod úhlem 359° jsou téměř totožné, ale rozdíl pouhým odečtením úhlů je velký, musí se provádět korekce výpočtu tak, že se určí odchylka úhlu k referenčnímu úhlu 180° a od tohoto se potom vypočítá rozdíl úhlu hran a vozidla a vybere se minimální úhel.

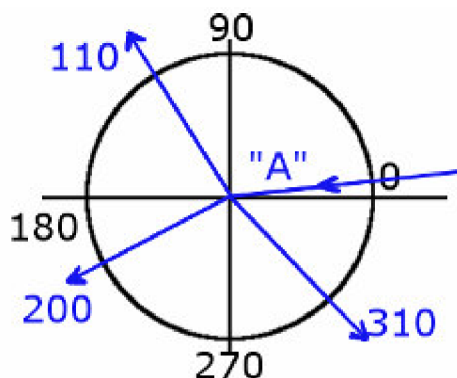
Stejně jako u vyhledávání nejbližšího uzlu, existuje k metodě „*searchNearEdgeAll()*“ optimalizovaná metoda „*searchNearEdge()*“. Ta opět využívá seznamu hran, které jsou následníky, či předchůdci poslední aktuální hrany. Tento seznam se vytváří v metodě „*setNearestEdges()*“.

8.7 Vytvoření pravidel silničního provozu

Pravidla silničního provozu se vytvoří při vytváření grafu silnic. S každou vytvořenou hranou se přečte i pravidlo, které reprezentuje dopravní značka. Pravidla jsou připisována ke hraně do proměnných, které byly popsány v kapitole 8.6.2. Pokud hrana nemá žádnou dopravní značku, ponechají se pravidla z přednastavených hodnot.

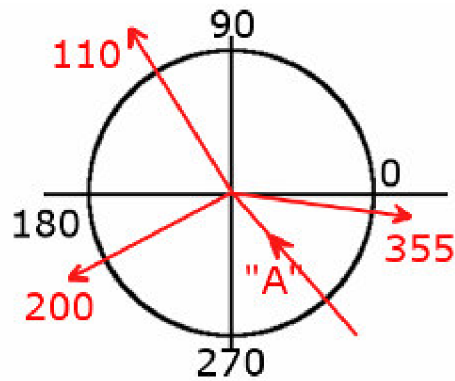
Příkázaný směr jízdy nestačí pouze připsat ke hraně, u které se nachází. Aby se jednoduše mohlo kontrolovat dodržení tohoto pravidla, je připsán i atribut ke hraně, na kterou je směr příkázán. Pravidlo příkázaného směru se musí upřesnit až po sestavení celého grafu. Pro tento účel se vytváří seznam hran, které obsahují pravidlo příkázaného směru jízdy a po vytvoření celé grafové struktury se tento seznam projde a dokončí se vytvoření pravidla.

Při procházení seznamu se pro každou uloženou hranu, u které je značka příkázaný směr jízdy (označme ji „A“), určí uzel „U“, do kterého hrana směřuje. Protože každý uzel v grafu má seznam hran z něj vystupujících, známe všechny hrany, na které by potenciálně mohl směřovat příkázaný směr jízdy. Podle směru hrany „A“ a směru každé hrany vystupující z uzlu „U“ se nalezne hrana příslušející pravidlu. Nebude se počítat hrana opačná k hraně „A“. Směr hrany je reprezentován úhlem v rovině souřadného systému. Pro určení směru nejvíce nalevo, nejvíce napravo a průběžné hrany od hrany „A“ se dá nalézt minimální a maximální úhel, nebo rozsah úhlu pro příkázaný směr jízdy rovněž viz obr. 8.14.



Obrázek 8.14: Určení nejpravější/nejlevější hrany podle hodnoty úhlu

Tento stav však není zaručen vždy a hrana nejvíce napravo nemusí mít minimální hodnotu úhlu, jak je znázorněno na obrázku 8.15., a proto je nutné provést korekce. Jednou z možností je úhly přepočítat k referenční hodnotě 180°. Pokud hrana „A“ bude mít směr 180°, může se vycházet z předpokladu určení minima a maxima úhlu ostatních hran. Hrany, jejichž úhel je větší jak 360°, nebo menší jak 0°, jsou přepočteny do rozsahu 0°-360°.



Obrázek 8.15: Určení směru hrany podle hodnoty úhlu - nesprávná interpretace

Do atributu „edgeEnable“ nalezené hrany se nastaví ukazatel na hranu „A“.

8.8 Kontrola pravidel silničního provozu při simulaci

Pro kontrolu pravidel je nutné určit hranu, po které se vozidlo pohybuje. Tyto metody byly popsány v předcházející kapitole. Kontrola pravidel je rozdělena na tři části.

- pravidla kontrolovaná na hraně,
- pravidla kontrolovaná v křižovatce,
- pravidla kontrolovaná po průjezdu křižovatkou.

Nezávisle na tom, kde se vozidlo nachází, zda na hraně nebo v křižovatce, se detekuje vzdálenost vozidla od hrany. Pokud vzdálenost vozidla od hrany přesáhne šířku vozovky, je detekován stav „vozidlo mimo vozovku“ a řidič je na to upozorněn, jak je znázorněno na obrázku 8.16.



Obrázek 8.16: Detekce vyjetí mimo vozovku

8.8.1 Pravidla kontrolovaná na hraně

Prvním pravidlem je zákaz vjezdu a jednosměrná ulice – kontroluje stav povolená / zakázaná hrana a pokud vozidlo vjede na zakázanou hranu, jedná se o přestupek.

U pravidla zákaz stání a zákaz zastavení se kontroluje čas, po který vozidlo stojí na hraně. Protože výklad podle silničního zákona nelze přesně převést na čas, je čas přizpůsoben vhodnosti pro tento simulátor. Pro zákaz zastavení je doba definována jako čas stání do 5sec. Pro zákaz stání je definována doba stání na vozovce jako čas do 60sec.

Kontrola překročení rychlosti je jednoduché porovnání aktuální rychlosti vozidla a hodnoty získané z atributu hrany.



Obrázek 8.17: Pravidlo „nevyšší dovolená rychlost 70km/h“ - kontrolované na hraně

8.8.2 Kontrola pravidel v křižovatce

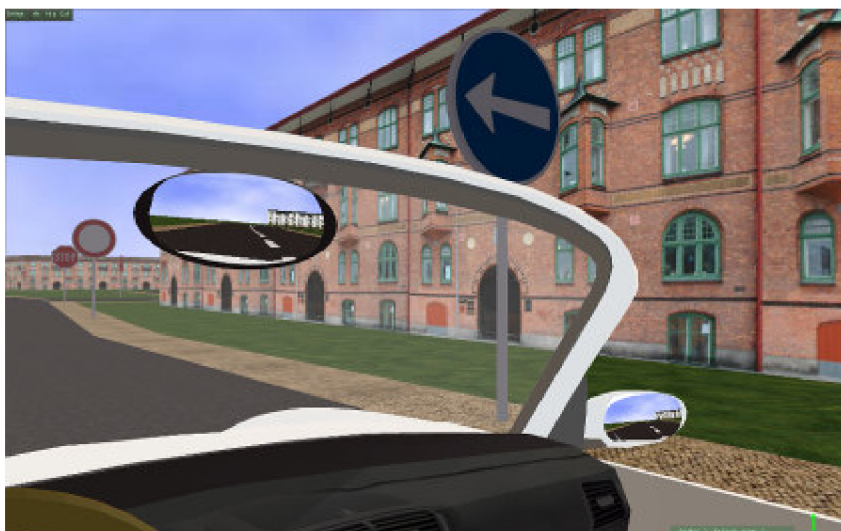
Zde jsou kontrolována pravidla upravující přednost v jízdě. Pro pravidlo „stůj, dej přednost v jízdě“ se kontroluje, zda vozidlo opravdu zastavilo, tedy snížilo rychlost jízdy na 0 km/h. Přesné místo zastavení není definováno, protože vyhláška stanoví zastavení na místě, odkud má řidič náležitý rozhled do křižovatky. Přednost v jízdě je bez ostatního provozu kontrolována způsobem, kdy se sleduje zda vozidlo před křižovatkou alespoň sníží svoji rychlost.



Obrázek 8.18: Pravidlo "stůj, dej přednost v jízdě" - kontrované v křižovatce

8.8.3 Kontrola pravidel po projetí křižovatkou

Po projetí křižovatkou, což je detekováno stavem, že se změní aktuální hrana na jinou, se kontroluje přikázaný směr jízdy. Pokud předcházející hrana měla pravidlo přikázaného směru jízdy, je nové hraně detekován stav povolené hrany. Toto provázání hran bylo vytvořeno při vytváření pravidel a jejich zápisu k hranám.



Obrázek 8.19: Pravidlo „přikázaný směr jízdy“ - kontrované po projetí křižovatkou

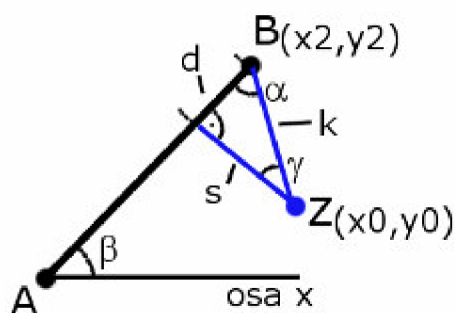
8.9 Umístění dopravních značek

Poloha dopravních značek není udána přesnou souřadnicí. Před zobrazením dopravní značky se provedou výpočty 8.2 - 8.4, které určí souřadnici dopravní značky. Vypočítá se vzdálenost od středu vozovky. Střed vozovky je totiž reprezentován hranou, kolem které je vozovka zobrazena. Pro značku je určen pouze směr, tedy ke které hraně patří a vzdálenost „d“ od uzlu, do kterého hrana směřuje. Směr hrany je úhel „ β “. Umístění dopravní značky „Z(x0,y0)“ se po znalosti šířky silnice „s“ vypočítá podle vlastností pravoúhlého trojúhelníku. Situaci znázorňuje obrázek 8.20.

$$\gamma = \cos^{-1} \frac{\sqrt{d^2 + s^2}}{d} \quad (8.2)$$

$$\alpha = \beta - 180^\circ + \gamma \quad (8.3)$$

$$\begin{aligned} x_0 &= x_2 + \sin \alpha * d \\ y_0 &= y_2 - \cos \alpha * d \end{aligned} \quad (8.4)$$



Obrázek 8.20: Výpočet souřadnic pro umístění dopravní značky

9 Instalační balík aplikace

Celý projekt je tvořen ve vývojovém prostředí Microsoft Visual Studio 2005 [38] (dále jen VS) s vloženými knihovnami grafického enginu OGRE a knihovnami fyzikálního enginu Bullet. Aby bylo aplikaci možno spouštět i mimo vývojové prostředí, což je samozřejmým předpokladem, je nutné vytvořit instalační balík pro celou aplikaci.

Pro vytvoření instalačního balíku lze využít například programu WIX [39], nebo přímo integrovaných komponent Visual Studia.

Ve VS 2005 se vytvoří nový projekt typu „Setup Project“, ve kterém se nastaví atributy pro aplikaci. Výsledkem je soubor setup.exe a instalační balík *.msi.

Instalačnímu balíku se nastaví potřebné parametry, pole použitých technik a knihoven. Pro funkci aplikace v OgreBullet je nutné mít v souboru resources.cfg nastaveny cesty k mediím, které jsou v aplikaci využívány. Jedná především o 3D modely, materiály, textury, fonty apod. Všechny soubory se VS pomocí položky add->File přidají do projektu. Je nutné vytvořit přesnou adresářovou strukturu, která se vytvoří po nainstalování aplikace. Adresářová struktura je daná podle rozdělení souborů s knihovnami a médii a v tomto případě se řídí pravidly vytváření aplikací OGRE enginu [22]. Protože se aplikace renderuje pomocí D3D knihovny, musí se přidat do balíku potřebná „d3dx9_37.dll“ knihovna. Další DDL knihovny, které jsou využívány při vývoji ve VS 2005 a novějších, je nutné dodat volbou „add->Merge Modules“. To zajistí přidání a následné nainstalování vývojových knihoven potřebných k běhu aplikace.

Po nastavení požadovaných parametrů se projekt přeloží v režimu release a výsledkem je setup.exe soubor a balík *.msi.

10 Závěr

Cílem této práce bylo vytvořit simulátor 3D autoškoly, který by byl použitelný jako výuková aplikace začínajícím řidičům. Protože problematika simulátoru autoškoly je velice rozsáhlým tématem, bylo jako hlavní předpoklad zvoleno vytvoření jádra simulátoru autoškoly, které může být dále rozšiřováno a bude podporovat vytváření a připojování nových komponent.

V prvních krocích je důležité pochopení základních vlastností simulací a technik při nich využívaných. Při vývoji simulátoru bylo nutné si nejprve stanovit za cíl, k čemu bude aplikace sloužit a podle toho navrhnout způsob postupu při vývoji. Jako vhodné se ukázalo studium již existujících simulátorů, ve kterých se lze seznámit s používanými technikami. Bylo provedeno několik testování běžně dostupných aplikací.

Poznatky, získané při testování existujících simulátorů, se staly základem pro návrh nového simulátoru. Klíčovým prvkem, který byl vyhodnocen jako nejdůležitější, je uživatelské rozhraní aplikace. Uživatelské rozhraní velkou mírou ovlivňuje celkový dojem uživatele simulátoru. Proto bylo navrženo trochu netradiční řešení, kdy uživatel simulátoru využívá pro ovládání automobilu obou rukou. Jedna ovládá automobil a druhá ovládá pohled řidiče. Pro dosažení toho efektu bylo po testování různých možností zvoleno ovládání automobilu na klávesnici a ovládání pohledu pomocí myši. Celkový návrh nové aplikace, z pohledu uživatelského rozhraní, je popsán v páté kapitole.

Velice podstatnou částí u simulátoru autoškoly je problematika dopravního systému. Za tímto účelem byly nastudovány možnosti reprezentace dopravních systémů. Byly použity metody využívané v jiných oborech. Pro reprezentaci silniční sítě, se jako vhodný prvek jeví orientovaný graf. Orientovaný graf je využíván pro podobné účely v geografických systémech a jeho použití bylo zvoleno i pro tento simulátor. Hrany v simulátoru reprezentují vozovku a uzly grafu reprezentují křižovatky a změny na vozovce. Pro tento účel byly vyvinuty třídy „GraphNode“ a „GraphEdge“, které zapouzdřují atributy a metody pro sestavení a práci s grafem.

Pro návrh jádra simulátoru jsou využity volně dostupné knihovny. Hlavním požadavkem bylo, aby knihovny umožnily kvalitní základ simulátoru a jejich použití nebylo komplikované. Pro grafickou stránku je využito knihovny OGRE. Jedná se o knihovnu využívanou při vývoji počítačových her, která má dobrou dokumentaci, dostupné tutoriály pro začínající programátory i početnou komunitu na diskusním fóru. Fyzikální model je založen na knihovně Bullet, která poskytuje metody pro detekci kolizí a dynamiku tuhého i pružného tělesa.

Detailně jsou popsány metody využívané při sestavování a kontrole pravidel silničního provozu. Pro reprezentaci sítě je vytvořena grafová struktura obsahující uzly a hrany, které nesou informace o vozovce a pravidlech. Kontrola pravidel je vyhodnocována podle pozice vozidla na vozovce.

Protože aplikace využívá dynamické knihovny, které jsou důležité ke spuštění, je vytvořen instalační balík zajišťující rozbalení a nainstalování potřebných komponent. Tato problematika je nastíněna v poslední deváté kapitole této práce. Simulátor je implementován ve vývojovém prostředí Microsoft Visual Studio 2005 [38]. Toto prostředí poskytuje i možnost vytváření instalačních balíčků.

Práce zahrnuje širokou oblast zaměření od grafiky počínaje, až po simulování a uživatelská rozhraní. Práce si nebere za cíl vytvořit realistický model světa s propracovanou grafikou a využití metod jako je např. metoda sledování paprsku apod. Jejím cílem je, aby aplikace byla spustitelná na běžných počítačích každého uživatele a mohla běžet v reálném čase. Edukační myšlenka projektu zahrnuje i míru detailů a zobrazovaných prvků. Je kladen důraz na prostředí, ve kterém začínající

řidič vidí základní informace a není rozptylován ostatními aspekty běžného provozu, aby se naučil techniky řízení vozidla a dodržování pravidel silničního provozu.

V dalším pokračování a zlepšování práce je možnost vytvořit kompletní pravidla silničního provozu, která budou obsahovat i křižovatky řízené světelnými signály. Velice zajímavou oblastí může být projekt vytvoření ostatních účastníků silničního provozu, jejichž chování bude vytvořeno na základě umělé inteligence, či podobných technik. Také je možnost vytvořit lekce, které by uživatel absolvoval, a vyhodnocovala by se jeho úspěšnost v řízení vozidla a zvládnutí jednotlivých lekcí.

V rámci práce se podařilo navrhnout a implementovat simulátor autoškoly. Vytvořený simulátor lze považovat za velmi perspektivní z hlediska budoucího použití především díky jeho širokému využití mezi uživateli a snadné modifikovatelnosti s ohledem na přidávání nových prvků (vytvoření cvičných lekcí, rozšíření pravidel a přidání ostatních účastníků silničního provozu).

Literatura

- [1] CHAPMAN, Alan. Swot analysis : SWOT analysis method and examples, with free SWOT template [online]. c1995-2008 [cit. 2008-11-10]. Dostupný z WWW: <<http://www.businessballs.com/swotanalysisfreetemplate.htm>>.
- [2] KINDLER, Evžen. *Simulační programovací jazyky*. 1. vyd. Praha : SNTL, 1980. 280 s.
- [3] RÁBOVÁ , Zdena, a kol. *Modelování a simulace : skriptum*. Brno : VUT, 1992.
- [4] PERINGER , Petr. *Modelování a simulace : Studijní opora*. Brno : VUT FIT, 2006. 83 s.
- [5] *Center for Advanced Transportation Systems Simulations : University Transportation Center* [online]. University of Central Florida (Florida) : c1999-2008 [cit. 2008-11-17]. Dostupný z WWW: <<http://catss.ucf.edu/>>.
- [6] *Simulation* [online]. 2009 [cit. 2008-12-12]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Simulation>>.
- [7] *JKZ s.r.o* [online]. 2008 [cit. 2008-10-22]. Dostupný z WWW: <<http://www.jkzsim.cz/cz/ridicke-trenazery/>>.
- [8] *Presagis Canada Inc* [online]. c2008 [cit. 2008-10-22]. Dostupný z WWW: <<http://www.presagis.com/>>.
- [9] *Center for Intelligent Systems Research : Driver Simulator Laboratory* [online]. The George Washington University (Virginia) : [2008] [cit. 2008-10-22]. Dostupný z WWW: <<http://www.cisr.gwu.edu/index.html>>.
- [10] *Würzburger Institut für Verkehrswissenschaften GmbH* [online]. 2008 [cit. 2008-11-22]. Dostupný z WWW: <<http://www.wivw.de>>.
- [11] 3D-Fahrschule [online]. Besier 3D-Edutainment Wiesbaden (Deutschland) : c2001-2009 [cit. 2009-01-05]. Dostupný z WWW: <<http://www.3dfahrschule.de/>>.
- [12] 3D-Fahrschule : Screenshots Gobesdorf [online]. Besier 3D-Edutainment Wiesbaden (Deutschland) : 2001-2009 [cit. 2009-01-05]. Dostupný z WWW: <http://www.3d-simulator.de/3DFS/v2-01/uk_shots_L01.htm>.
- [13] Drivers education [online]. Superhry.cz, c2004-2008 [cit. 2008-11-02]. Dostupný z WWW: <<http://www.superhry.cz/games/600>>.
- [14] Vionsoft : KarŠkolka [online]. c2007-2009 [cit. 2008-11-02]. Dostupný z WWW: <<http://karskolka.vionsoft.cz/clanek.aspx?clanek=kars>>.
- [15] HÁK, Roman. 4Drivers - The traffic simulator [online]. c2006 [cit. 2008-12-17]. Dostupný z WWW: <<http://maverick.webz.cz/4Drivers/>>.
- [16] Live for Speed : online racing simulator [online]. c2002-2009 [cit. 2009-01-04]. Dostupný z WWW: <<http://www.lfs.net/>>.
- [17] PEŇÁZ , Tomáš. *Síťové analýzy v prostředí GIS*. Ostrava : VŠB - Technická univerzita Ostrava, Hornicko-geologická fakulta, Institut geoinformatiky, 2006. 32 s. Dostupný z WWW: <http://gisak.vsb.cz/~pen63/Systemy_GIS_v_PO/Sitove_analyzy_GIS.pdf>.
- [18] HÁK, Roman. *Dopravní simulátor*. [s.l.], 2007. 36 s. České vysoké učení technické v Praze Fakulta elektrotechnická. Vedoucí bakalářské práce Doc. Ing. Jiří Žára, CSc. Dostupný z WWW: <<https://dip.felk.cvut.cz/browse/details.php?f=F3&d=K13136&y=2007&a=hakr1&t=bach>>.

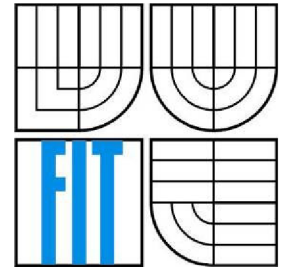
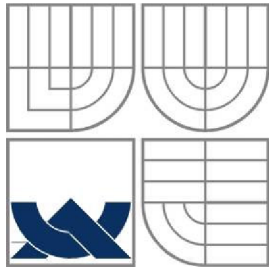
- [19] VAŠÍČEK , Jan. Detekce kolizí. Plzeň, 2008. 42 s. Západočeská univerzita v Plzni. Fakulta aplikovaných věd. Katedra informatiky a výpočetní techniky. Vedoucí diplomové práce Doc. Dr. Ing. Ivana Kolingerová. Dostupný z WWW: <http://herakles.zcu.cz/~skala/MSc/diploma_data/DP_2008_Vasicek_Jan.pdf>
- [20] DITCHBURN, Keith. Collisions [online]. University of Teesside : c2004-2008 [cit. 2009-01-06]. Dostupný z WWW: <<http://www.toymaker.info/Games/html/collisions.html>>.
- [21] Bullet : Collision Detection and Physics Wiki [online]. c2004-2005 [cit. 2008-09-10]. Dostupný z WWW: <http://www.bulletphysics.com/mediawiki-1.5.8/index.php?title=Main_Page>.
- [22] Ogre : Object-Oriented Graphics Rendering Engine [online]. Torus Knot Software, c2000-2007 [cit. 2008-10-10]. Dostupný z WWW: <<http://www.ogre3d.org/>>.
- [23] Crystal Space [online]. Crystal Space Team , [2008] [cit. 2008-10-06]. Dostupný z WWW: <<http://www.crystalspace3d.org>>.
- [24] Arcamoria : Crystal Space or Ogre 3D [online]. Walt Collins, c2007-2008 , March 2007 [cit. 2008-12-17]. Dostupný z WWW: <<http://www.arcanoria.com/CS-Ogre.php>>.
- [25] Open Dynamics Engine [online]. Russel Smith, c2000-2006 , Monday May 28 2007 [cit. 2008-12-19]. Dostupný z WWW: <<http://www.ode.org/>>.
- [26] Ogre : OgreBullet [online]. c2000-2007 [cit. 2008-12-19]. Dostupný z WWW: <<http://www.ogre3d.org/addonforums/viewforum.php?f=12>>.
- [27] OgreODE [online]. 2008 , 09:38, 24 October 2008 [cit. 2008-12-19]. Dostupný z WWW: <<http://www.ogre3d.org/wiki/index.php/OgreODE>>.
- [28] Microsoft Visual Studio [online]. 2009 Microsoft Corporation, c2009 [cit. 2009-01-05]. Dostupný z WWW: <<http://www.microsoft.com/cze/msdn/produkty/vstudio/default.aspx>>.
- [29] KŘENA, Bohuslav, KOČÍ, Radek. Úvod do softwarového inženýrství : Studijní opora. Brno : VUT FIT, 2006. 100 s.
- [30] Blender : free open source 3D content creation suite [online]. 2009 [cit. 2009-05-05]. Dostupný z WWW: <<http://www.blender.org/>>.
- [31] Blender Exporter : OGRE Meshes Exporter [online]. 2009 [cit. 2009-04-28]. Dostupný z WWW: <http://www.ogre3d.org/wiki/index.php/Blender_Exporter>.
- [32] 3DS2Mesh [online]. 2009 [cit. 2009-04-28]. Dostupný z WWW: <<http://www.ogre3d.org/wiki/index.php/3DS2Mesh>>.
- [33] Dewantoro Network [online]. 2008 [cit. 2008-09-17]. Dostupný z WWW: <<http://www.dewantoro.net/>>.
- [34] Butzi [online]. c2004-2009 [cit. 2009-04-12]. Dostupný z WWW: <http://www.butzi.cz/modely/08_cayenne_gts/08_cayenne_gts_04.jpg>.
- [35] Dopravní-značení [online]. 2009 [cit. 2009-04-15]. Dostupný z WWW: <<http://www.dopravni-znacen.eu/>>.
- [36] GIMP : GNU Image Manipulation Program [online]. c2001-2009 [cit. 2009-05-02]. Dostupný z WWW: <<http://www.gimp.org/>>.
- [37] CGTextures [online]. 2009 [cit. 2009-04-17]. Dostupný z WWW: <<http://www.cgtextures.com/>>.
- [38] Microsoft Visual Studio 2005 [online]. c2009 [cit. 2009-05-04]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/library/ms950416.aspx>>.
- [39] WIX : Windows Installer XML [online]. c2005-2009 [cit. 2009-05-04]. Dostupný z WWW: <<http://wix.sourceforge.net/>>.

Seznam obrázků

Obrázek 2.1: Schéma fází modelů při simulacích [4]	7	
Obrázek 3.1: Simulátor autoškoly firmy Presagis [7]	11	
Obrázek 3.2: Schéma komplexního simulátoru [9].....	12	
Obrázek 3.3: Detail napojení uživatelského rozhraní simulátoru na sensory [9].....	13	
Obrázek 3.4: Příklady simulátorů vysoké úrovně realističnosti [5,9]	13	
Obrázek 3.5: Interiér automobilu v aplikaci 3D Fahrschule [12]	15	
Obrázek 3.6: Vzhled motocyklu v aplikaci 3D Fahrschule [12].....	15	
Obrázek 3.7: Náhled jednoduché aplikace „Driver ED“ simulující pravidla silničního porvozu [13]	18	
Obrázek 5.1: Návrh ovládání aplikace pomocí myši a klávesnice.....	27	
Obrázek 6.1: Obecný graf s hranami a uzly [17]	28	
Obrázek 6.2: Model komunikace orientovaným grafem.....	29	
Obrázek 6.3: Model komunikace s více jízdními pruhy orientovaným grafem.....	30	
Obrázek 6.4: Model komunikace a dopravních značek orientovaným grafem.....	31	
Obrázek 6.5: Model křižovatky orientovaným grafem	31	
Obrázek 6.6: Model křižovatky se zámky orientovaným grafem	32	
Obrázek 6.7: Model světelné křižovatky se zámky orientovaným grafem.....	33	
Obrázek 7.1: Grafické schéma spolupráce podsystémů v 3D autoškole	39	
Obrázek 7.2: Schéma spolupráce komponentů uživatelského návrhu	40	
Obrázek 7.3: Návrh ovládání aplikace pomocí myši a klávesnice.....	41	
Obrázek 8.1: Model automobilu	43	
Obrázek 8.2: Model města.....	43	
Obrázek 8.3: Propojení funkčních bloků	44	
Obrázek 8.4: Model palubní desky	45	
Obrázek 8.5: Pohled z kabiny řidiče	47	
Obrázek 8.6: Pohled za automobilem – pohled třetí osoby	48	
Obrázek 8.7: Menu aplikace	50	
Obrázek 8.8: Model města - zobrazení budov	51	
Obrázek 8.9: Nalezení nejbližší hrany	54	
Obrázek 8.10: Vzdálenost vozidla od přímkky - nesprávná interpretace	54	
Obrázek 8.11: Průmět bodu P na hraně	Obrázek 8.12: Průmět bodu P mimo hranu	55
Obrázek 8.13: Určení správné hrany podle směru	55	
Obrázek 8.14: Určení nejpravější/nejlevější hrany podle hodnoty úhlu.....	56	
Obrázek 8.15: Určení směru hrany podle hodnoty úhlu - nesprávná interpretace	57	
Obrázek 8.16: Detekce vyjetí mimo vozovku	57	
Obrázek 8.17: Pravidlo „nevyšší dovolená rychlost 70km/h“ - kontrolované na hraně.....	58	
Obrázek 8.18: Pravidlo "stůj, dej přednost v jízdě" - kontrolované v křižovatce	59	
Obrázek 8.19: Pravidlo „příkázaný směr jízdy“ - kontrolované po projetí křižovatkou	59	
Obrázek 8.20: Výpočet souřadnic pro umístění dopravní značky	60	

Seznam příloh

1. Uživatelský manuál
2. CD-ROM



3D Autoškola

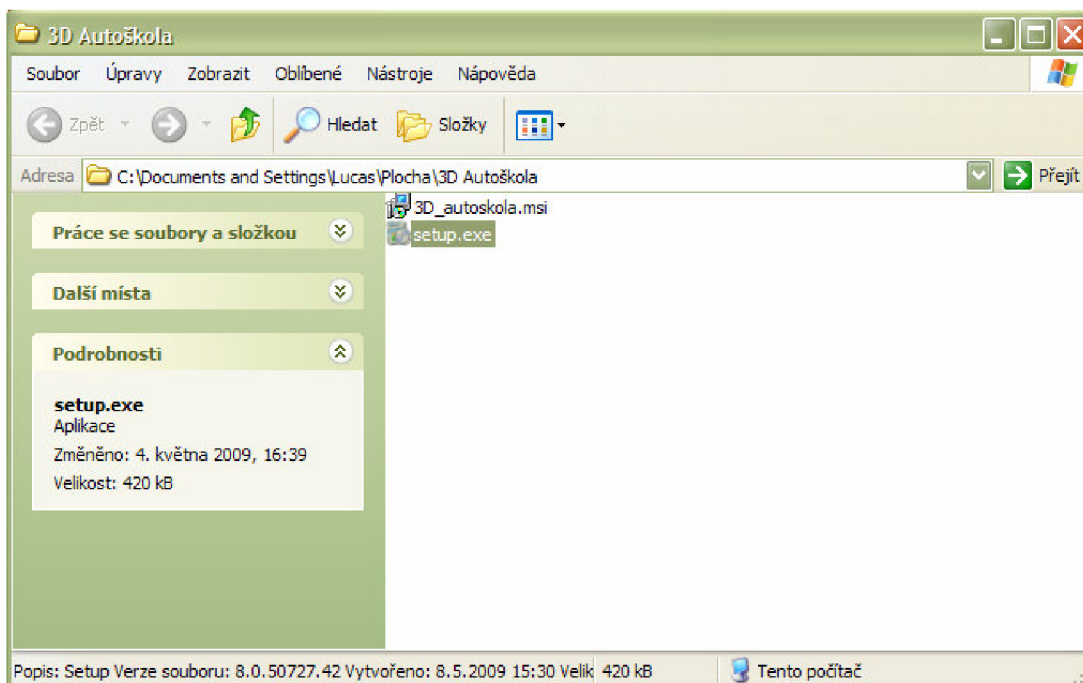
Uživatelský manuál

Autor: Bc. Lukáš Pernica

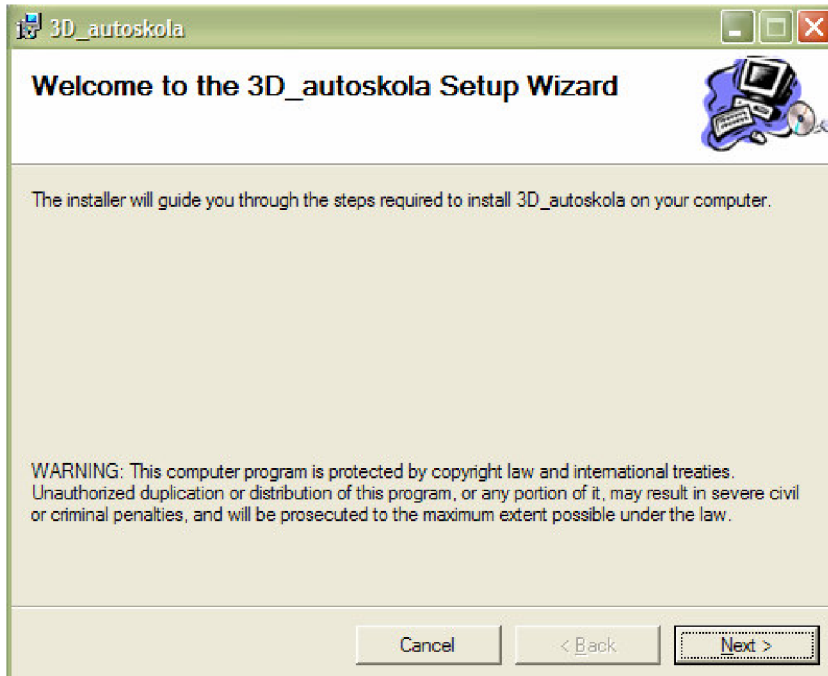
VUT FIT
Brno 2009

Instalace aplikace

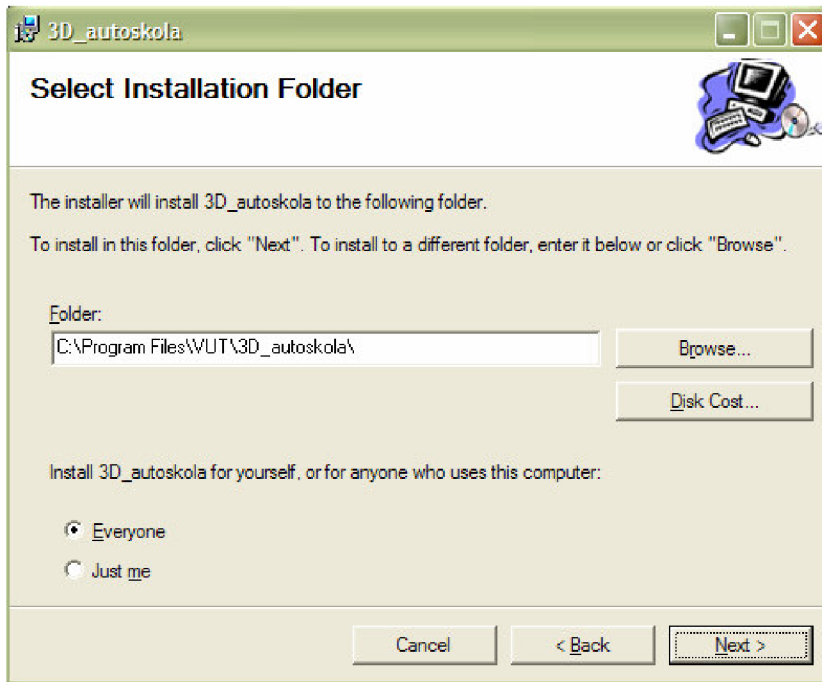
Aplikace se instaluje pomocí „setup.exe“ souboru...



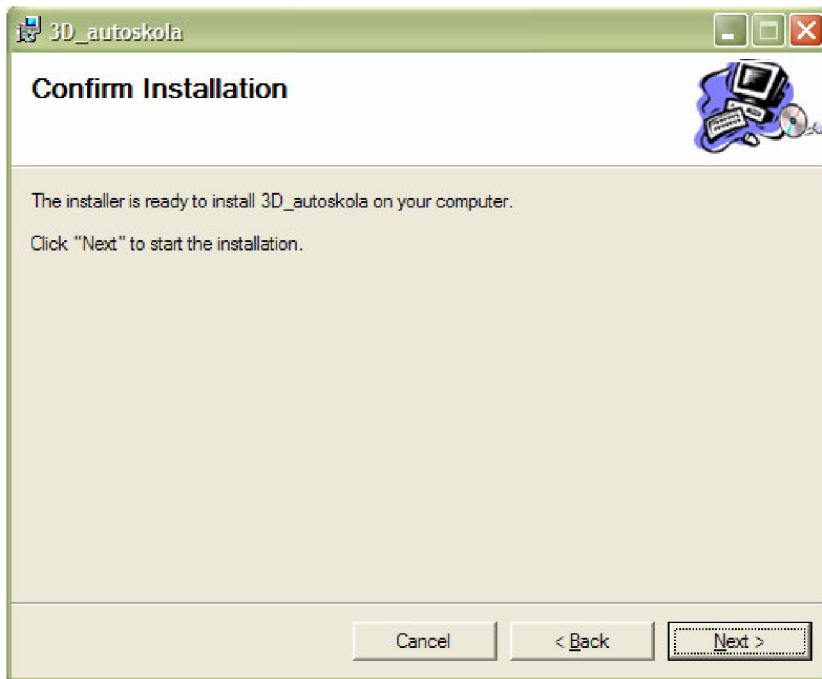
Po spuštění pokračujte volbou NEXT...



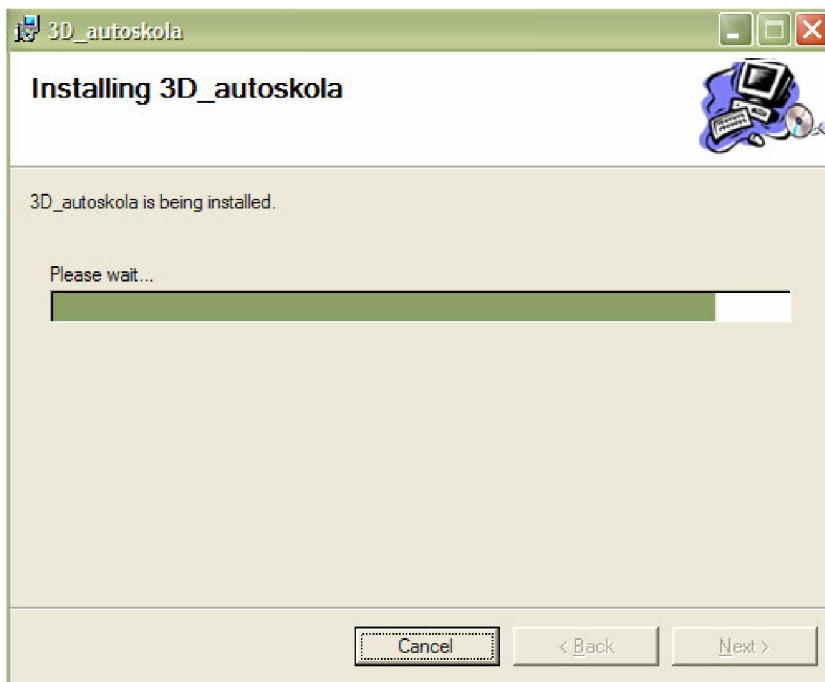
Výběr adresáře, do kterého se má aplikace nainstalovat. Můžete ponechat přednastavenou cestu, pokud si nepřejete instalovat do jiné složky, a pokračovat NEXT...



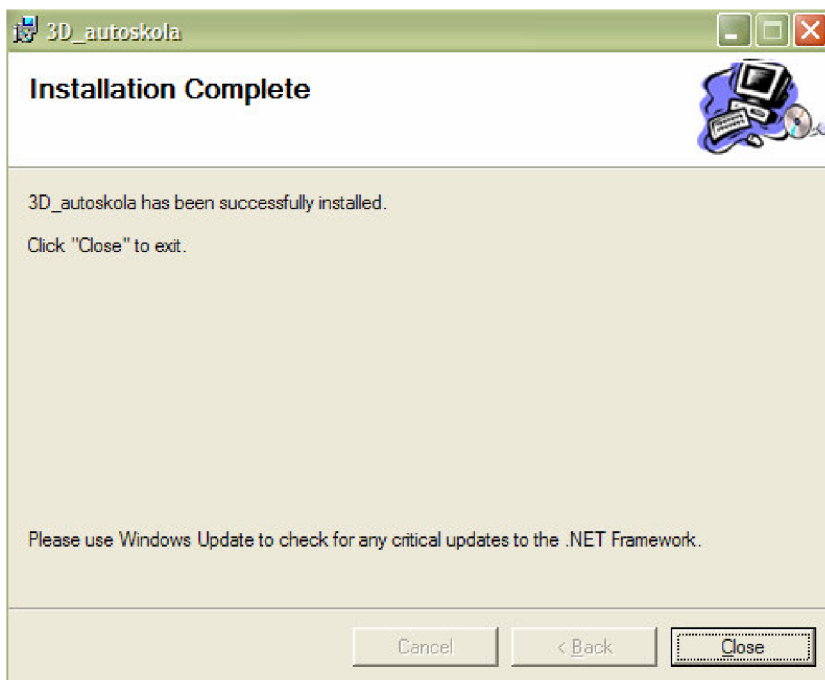
Setup je nastaven a instalace se spustí pomocí NEXT...



Instalace může trvat několik minut, prosím počkejte...



Jakmile je instalace dokončena objeví se následující obrazovka, kterou ukončíte volbou CLOSE.



Gratulujeme,

aplikace byla nainstalována a je připravena ke spuštění. Na Vaší pracovní ploše je nyní umístěna ikona pro spuštění.

Spuštění aplikace

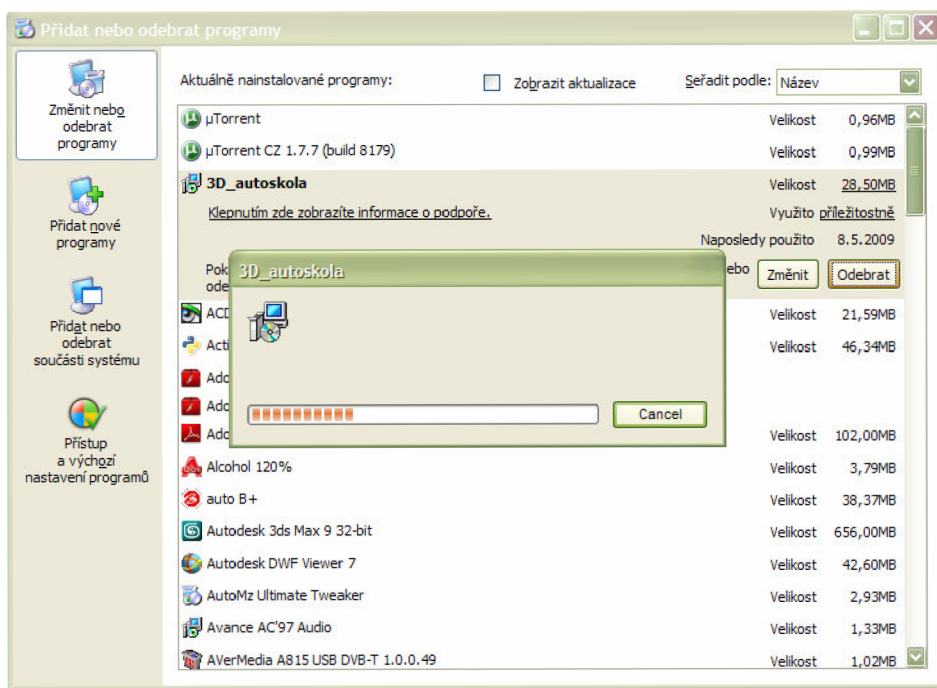
Aplikace spustí ikonou „3D_autoskola“. Po spuštění se objeví následující okno, ve kterém můžete nastavit parametry vaší grafické karty.



Oinstalování aplikace

Pro odebrání aplikace z Vašeho počítače použijte průvodce „Přidat odebrat programy“, který se nachází v ovládacích panelech.

Aplikaci jednoduše odinstalujete tlačítkem ODEBRAT...



Akce může trvat několik minut, vyčkejte prosím, až se okno samo zavře.

Gratulujeme,
Aplikace byla úspěšně odebrána z Vašeho počítače.

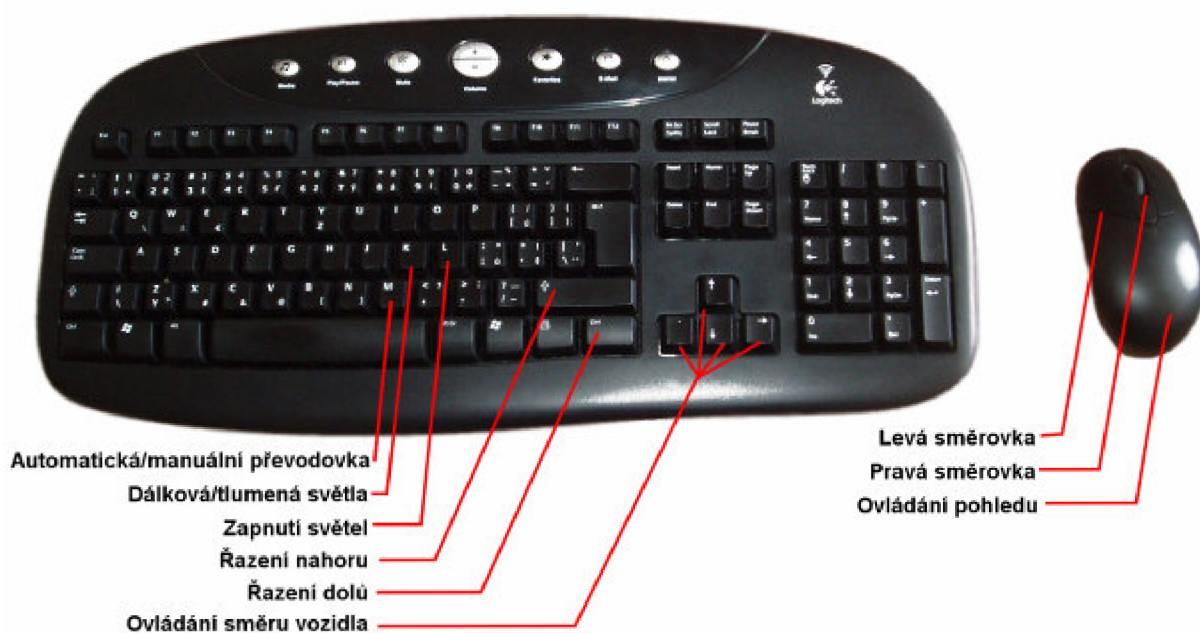
Ovládání aplikace

Základní nastavení aplikace je lze ovládat pomocí myši, kdy je nutné najet myší do horního levého rohu, ve kterém se zobrazí kurzor. V menu je možno nastavit Základní parametry vozidla:

- Natáčení kol - úhel natočení
- Rychlost zatáčení kol
- Výkon motoru

Dále je možno zobrazovat nápovědu s ovládáním a zobrazovat stavový řádek s informacemi zobrazený ve spodní části okna.

Ovládání simulátoru je znázorněno na následujícím obrázku a popsáno v tabulce.



Přehled kláves pro ovládání aplikace

Myš	
horizontální pohyb	pohyb pohledu doleva, doprava
vertikální pohyb	pohyb pohledu nahoru, dolů
levé tlačítko	směrové světlo levé
pravé tlačítko	směrové světlo pravé
Klávesnice	
šipka vpravo	zatočení vpravo
šipka vlevo	zatočení vlevo
šipka nahoru	akcelerace
šipka dolů	zbrzdění
pravý ctrl	podřazení rychlosti
pravý shift	zařazení rychlosti
L	zapnutí/vypnutí světel
M	manuální/automatická převodovka
K	Dálková / tlumená světla

O aplikaci

Simulátor autoškoly je vytvořen s knihovnamy OGRE a Bullet. Může sloužit pro začínající řidiče, kteří si chtějí na počítači osvojit znalosti pravidel silničního provozu.

Text diplomové práce se zabývá popisem a návrhem simulátoru 3D autoškoly. V práci je shrnut základní popis některých dostupných simulátorů a možnosti jejich využití. Na základě podrobné studie simulátoru 3D Fahrschule je navržen nový abstraktní model simulátoru. V práci je popsána problematika simulace dopravního systému, kontroly dodržování pravidel v silničním provozu a také navržen nový model uživatelského rozhraní. Nakonec je popsán celý návrh a postup při implementaci simulátoru.

Tato aplikace vznikla v roce 2009 jako diplomový projekt na VUT FIT v Brně.