



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

3D AUTOŠKOLA

THESIS TITLE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN HLIPALA

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. PETER CHUDÝ, Ph.D., MBA

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Hlipala Martin**

Obor: Informační technologie

Téma: **3D Autoškola
Driving School - Rules of the Road**

Kategorie: Modelování a simulace

Pokyny:

1. Nastudujte problematiku simulace silničního provozu a metodiku kontroly dodržování pravidel silničního provozu. Analyzujte platná pravidla systému a vytvořte jejich seznam dle priority implementace.
2. Vytvořte architekturu systému simulátoru a 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 reprezentaci pohybu účastníků provozu a jejich vzájemné interakce.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a část 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% 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: 1. listopadu 2016

Datum odevzdání: 17. května 2017

L.S.



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Práca rieši problematiku simulácie pravidiel cestnej premávky a interaktívneho začlenenia užívateľa aplikácie do scény, pričom mu umožňuje aplikovať jeho schopnosti a vedomosti z oblasti pravidiel cestnej premávky priamo v scéne virtuálneho sveta. Aplikácia odhaľuje priestupky spáchané užívateľom a informuje ho o tom. Práca bola realizovaná pomocou technológie Unity 3D. Pre tvorbu a editáciu 3D objektov a častí scény, mapovanie textúr na 3D objekt, tvorbu a mapovanie materiálov na 3D objekty je využívaný nástroj Blender.

Abstract

This project is focused on the simulation of traffic rules and user interaction in a virtual scene, where it allows to examine user's knowledge and skills in context of the traffic law and rules. The designed application is monitoring user's activity and identifying traffic violations caused by the user's bad behaviour. Based on the gathered information, the user is provided with an information about the rules he/she disobeyed. This application was designed using the Unity 3D technology. 3D modeling software Blender was used for creating, texturing and assigning material properties to 3D objects used in the project.

Klíčové slová

Pravidla cestnej premávky, autoškola, Unity3D, Blender, 3D model

Keywords

Traffic rules, driving school, Unity3D, Blender, 3D model

Citácia

HLIPALA, Martin. *3D autoškola*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Peter Chudý, Ph.D., MBA

3D autoškola

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána doc. Ing. Petra Chudého, Ph.D., MBA. Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....
Martin Hlipala
17. mája 2017

Podakovanie

Touto cestou sa chcem poďakovať vedúcemu práce doc. Ing. Petrovi Chudému, Ph.D., MBA, za jeho odborný prístup, pomoc a čas ktorý mi venoval.

Obsah

1	Úvod	6
2	Simulácia cestnej premávky	7
2.1	Simulátor autoškoly	7
2.1.1	Profesionálne simulátory pre výcvik v autoškolách	7
2.1.2	Moderné simulátory na báze pohybovej platformy	8
2.1.3	Desktopové simulátory	8
2.2	Platné pravidlá cestnej premávky	11
2.2.1	Vodorovné dopravné značenie	11
2.2.2	Zvislé dopravné značenie	11
2.2.3	Riadenie cestnej premávky svetelnými signálmi	12
2.3	Analýza zaznamenaných dopravných priestupkov	13
2.4	Kontrola dodržiavania pravidiel cestnej premávky	14
3	Návrh architektúry simulátoru	15
3.1	Návrh architektúry systému	15
3.2	Návrh užívateľského rozhrania	16
3.2.1	Ovládacie prvky	16
3.2.2	Zobrazenie simulačných dát	17
3.3	Dostupné nástroje pre realizáciu návrhu	18
3.3.1	XNA herné jadro	18
3.3.2	LeadWerks	18
3.3.3	Unity 3D	18
3.3.4	Irrlicht 3D	18
3.3.5	Bullet SDK	19
3.3.6	Nvidia PhysX	19
3.4	Priorita pravidiel pre implementáciu	19
4	Fyzikálne jadro, vývojové prostredie a modely	21
4.1	Fyzikálne jadro	21
4.1.1	Vybraný fyzikálny systém	22
4.1.2	Ray-casting vo fyzikálnom systéme	22
4.1.3	Kolízne teleso	23
4.1.4	Priestor a orientácia v ňom	24
4.1.5	Všeobecné pohybové rovnice pre simuláciu	25
4.1.6	Rovnice popisujúce dynamiku vozidla	28
4.2	Vývojové prostredie Unity	30
4.3	Základná scéna	30

4.3.1	Modely priestorových objektov	30
4.3.2	Reprezentácia terénu	31
4.3.3	Cesty a križovatky	32
4.3.4	Budovy	33
4.3.5	Priestorovy model automobilu	34
4.3.6	Dopravné značenie	37
5	Implementácia	39
5.1	Fyzikálny model	39
5.2	Scéna simulácie	40
5.3	Automobil	42
5.4	Implementácia pravidiel a dopravných značení	45
5.5	Reprezentácia pohybu účastníkov premávky	47
5.6	Interakcia medzi prvkami simulácie	48
6	Testovanie	50
6.1	Metodika testovania	50
6.2	Vyhodnotenie testovania	51
7	Záver	52
	Literatúra	53
	A DVD	57
	B Dotazník	58

Zoznam obrázkov

2.1	Profesionálny simulátor vozidla.	8
2.2	Moderná pohybová plošina.	8
2.3	Náhľad na aplikáciu "Driving School Simulator".	9
2.4	Náhľad na aplikáciu "3D Fahrschule ".	10
2.5	Náhľad na aplikáciu "City Car Driving".	10
2.6	Vzniknutá škoda v dôsledku dopravnej nehody.	14
2.7	Dôsledok nesprávnych návykov vodiča.	14
3.1	Architektúra aplikácie.	16
3.2	Rozloženie tlačidiel ovládania na klávesnici.	17
3.3	Logá uvedených nástrojov	19
4.1	Raycast v praktickom príklade.	22
4.2	Lúče reagujú iba na nastavenú fyzikálnu vrstvu.	22
4.3	Komplexné kolízne teleso.	23
4.4	Primitívne kolízne teleso.	23
4.5	Súradný systém.	24
4.6	Automobilový súradný systém podľa Pacejku.	26
4.7	Výšková mapa terénu vo formate RAW.	32
4.8	Výsledok interpretácie výškovej mapy.	32
4.9	Tvorba mestského okruhu.	33
4.10	Segment cesty, ktorý sa replikuje.	33
4.11	Počítačové modely komerčných budov a rodinných domov.	33
4.12	Modely panelových domov.	33
4.13	Umiestňovanie vrcholov objektu na základe predlohy.	34
4.14	Pohľad na objekt v 3D priestore.	34
4.15	Hrubá aproximácia karosérie automobilu.	35
4.16	Aplikovaný subdivision modifikátor.	35
4.17	Finálna geometria karosérie vozidla.	35
4.18	Finálna geometria zadnej časti karosérie vozidla.	35
4.19	Detailný 3D objekt.	36
4.20	Normálová mapa.	36
4.21	Priradené materiály v exteriéri.	36
4.22	Priradené materiály v interiéri.	36
4.23	Svetelný modul.	37
4.24	Svetelné značenia z modulov.	37
4.25	Mapovanie povrchu zložitého objektu na 2D plochu.	38
4.26	Aplikovanie textúry na namapovaný povrch objektu.	38

5.1	Tvorba ciest pomocou EasyRoads3D.	40
5.2	Výsledný model cesty vytvorený pomocou pluginu EasyRoads3D.	40
5.3	Príklad seamless textúry.	41
5.4	Príklad textúry billboardu flóry.	41
5.5	Správna orientácia volantu v automobile.	43
5.6	Nastavenie východzích polôh ukazateľov.	43
5.7	Vozidlo v jeho finálnej podobe.	44
5.8	Logické objekty vozidla.	44
5.9	Križovatka riadená svetelnými semaforami.	46
5.10	Kontrola prednosti v klasickej križovatke.	46
5.11	Implementácia pravidla "Zákaz vjazdu všetkým vozidlám".	47
5.12	Implementácia pravidla "Stoj! Daj prednosť v jazde".	47
5.13	Vizualizácia bodov trasy.	48
5.14	Body trasy lemujú celú cestnú sieť.	48

Zoznam tabuliek

2.1	Tabuľka príčin vzniku dopravných nehôd za rok 2015	13
-----	--	----

Zoznam symbolov

ϕ	Úhol klonenia okolo osi X automobilového súradného systému
ψ	Úhol rotácie okolo osi Z automobilového súradného systému
F_x	Pozdĺžne sily pôsobiace na vozidlo
F_y	Priečné sily pôsobiace na vozidlo
M_z	Zarovnávacia moment
δ	Úhol vytočenia predných kolies
η_{CT}	Miera zapojenia spojky
γ_{DR}	Stály prevodový pomer
Ω	Koeficient mechanického odporu prevodovky
τ_i	Krútiaci moment na vstupe
τ_o	Krútiaci moment na výstupe
ζ_{TH}	Miera stlačenia plynového pedálu
a	Zrýchlenie
AFR	Vyjadrenie palivovej zmesy - pomer paliva a vzduchu
d	Vzdialenosť
f_{acce}	Faktor akcelerácie
F_{TB}	Spätná sila smerujúca do motora generovaná prevodovkou
k	Preklz kolies
M_ϕ	Klonivý silový moment
n	Odmocnina súčtu druhých mocnín koeficientov vplyvujúcich z popisu roviny kolízneho telesa
n_G	Aktuálne zaradený prevodový pomer
Q	Zobecnená sila
q	Zobecnená súradnica
r	Faktor zmeny rotácie okolo osi Z automobilového súradného systému
R_{wh}	Polomer kolesa
RPM	Otáčky motora
RPM_{max}	Maximálne možné otáčky motora
RPM_{min}	Otáčky motora v situácii keď nedochádza k akcelerácii

s	Dráha
T	Kinetická energia
t	Čas
TR	Funkcia odozvy plynového padálu
U	Potenciálna energia
u	Rýchlosť pozdĺž osi X automobilového súradného systému
v	Rýchlosť pozdĺž osi Y automobilového súradného systému
v_0	Počiatočná rýchlosť
V_g	Rýchlosť
v_{auto}	Rýchlosť automobilu
X	Súradnica osi X automobilového súradného systému
Y	Súradnica osi Y automobilového súradného systému
Z	Kinetická energia
δW	Virtuálna práca
K_a, K_b, K_c, K_d	Koeficienty vyplývajúce z popisu roviny kolízneho telesa
r_1, r_2, r_3	Súradnice X, Y a Z vzdialeného bodu

Zoznam skratiek

2D 2-dimenzionálny priestor.

3D 3-dimenzionálny priestor.

BFF lokálny súradný systém automobilu.

CPU Centrálny procesor.

GPU Grafický akcelerátor.

HW Hardvér.

LIDAR Laserové detekovanie a meranie vzdialeností.

LOD Technológia pre optimalizáciu geometrie objektov podľa ich vzdialenosti od kamery.

PC Osobný počítač.

RAW Obrazový formát v ktorom sa ukladajú aj výškové mapy.

SW Softvér.

Kapitola 1

Úvod

Keď v roku 1908 automobilka Ford pod vedením Henryho Forda uviedla na trh vozidlo s modelovým označením Model T, čím spôsobila vo veľkej miere rozvoj automobilizmu. Model T bolo prvé, cenovo dostupné vozidlo určené širokému spektru zákazníkov [20].

Podobné kroky vedú k masovému rozšíreniu automobilov čím vzniká potreba organizovanej výuky nových vodičov automobilov, a zároveň je základných pravidiel cestnej premávky.

Vzhľadom k uvedenej situácii vznikajú prvé autoškoly, ktoré si kladú za úlohu pripraviť budúcich vodičov motorových vozidiel, a naučiť ich platné dopravné predpisy.

S rozvojom spoločnosti, a pokrokom v technológiách, sú moderné vozidlá stále viac dostupnejšie širokým masám, pričom získanie vodičského oprávnenia podľa aktuálnej legislatívy je otvorené väčšine osôb.

K dňu 15.3.2016 pripadalo na jeden osobný automobil v Českej Republike 2,14 osôb, a tento pomer nepretržite klesá [43].

Z toho vyplýva, že účastníkov cestnej premávky je stále viac, a spolu s týmto faktorom stúpa riziko kolízie s iným vozidlom, s možnými vážnymi následkami. Z tohoto dôvodu vznikajú interaktívne počítačové simulátory, ktorých úlohou je prispieť ku skvalitneniu prípravy vodičov v autoškolách a minimalizovať tak počet dopravných kolízií. Existuje však viacero dôvodov vzniku dopravných nehôd, medzi časté prípady patrí situácia, kedy je vinník dopravnej nehody pod vplyvom alkoholu alebo iných omamných a psychotropných látok. Ďalšou častou príčinou je nedostatočné venovanie pozornosti premávke kvôli obsluhu mobilných komunikačných zariadení prípadne tiež neznalosť základných fyzikálnych zákonov a precenenie šoférskej schopnosti alebo možnosti svojho vozidla. Väčšinu z uvedených faktorov simulátory nevedia priamo ovplyvniť, nakoľko nedisponujú prostriedkami k tomu, aby tieto skutočnosti dostatočne verne nasimulovali.

Existujú však príčiny dopravných nehôd, ktorých výskyt je možné obmedziť použitím vhodných simulačných technológií v procese výuky nových vodičov. Typicky sa jedná o nedôsledné dodržiavanie dopravného značenia upravujúceho prednosť a rýchlosť premávky. Uvedený scenár dopravnej situácie bude predmetom tejto bakalárskej práce.

Kapitola 2

Simulácia cestnej premávky

Táto kapitola sa bude zaoberať spektrom existujúcich simulátorov pre autoškoly a súčasne uvedie analýzu plazných pravidiel cestnej premávky.

2.1 Simulátor autoškoly

Typický digitálny simulátor autoškoly je interaktívny softvér, ktorý je možné spustiť na výpočtovom zariadení akým je počítač, mobil či tablet. Cieľom počítačového simulátora je vytvoriť abstraktný model určitej podmnožiny reálneho sveta a umožniť tak jeho užívateľovi skúšať operácie či úkony, ktoré by v skutočnom svete mohli mať za následok újmu na zdraví, prípadne by mohli viesť k vzniku veľkej škody na majetku.

Existuje niekoľko riešení interaktívnych simulátorov zameraných na pravidlá cestnej premávky. Niektoré z nich sa prioritne zameriavajú na grafické spracovanie, iné na samotnú simuláciu pravidiel cestnej premávky, pričom grafické, prípadne audio prvky presúvajú do ústrania.

Simulátory dopravných predpisov a cestnej premávky je možné deliť do dvoch skupín, simulátory pre účely výcviku žiakov autoškôl a simulátory pre domáce použitie.

2.1.1 Profesionálne simulátory pre výcvik v autoškolách

Jedná sa o skupinu dopravných simulátorov vytvorených s cieľom skvalitnenia prípravy žiakov autoškoly. Dôvodom ich existencie je nácvik dopravných pravidiel a techniky vedenia motorového vozidla pred prvou jazdou vo vozidle autoškoly.

V uvedenom prípade sa nejedná iba o softvérové riešenie. Súčasťou simulátora je aj hardvér, verne napodobňujúci kabínu vozidla. Profesionálne simulátory väčšinou obsahujú palubnú dosku vozidla, sedadlá, volant, radiacu páku, ukazatele, pedále, spätné zrkadlá a množstvo ďalších ovládacích prvkov pochádzajúcich z reálneho vozidla s cieľom čo najvernejšie napodobniť modelovaný automobil. Scéna je väčšinou premietaná na viacero monitorov, poprípade na plátno.

Pri simulátoroch určených pre autoškoly sa prioritne dbá na simuláciu pravidiel cestnej premávky, a prostredia kabíny automobilu, ostatné aspekty akými sú počítačová grafika, zvuky, či prvky efektov prípadne distrakčnej interakcie sa zanedbávajú, pretože niesú pre definovaný účel dôležité.

Tento typ profesionálnych simulátorov je väčšinou prevádzkovaný s využitím architektúry klient-server, vďaka ktorej je simulátor modulárny, čo umožňuje získavať data o si-

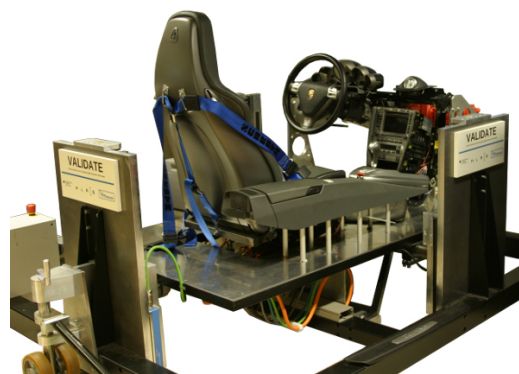
mulácii pre softvér tretích strán, prípadne pre hardvérové prvky, alebo vytvárať analýzy z priebehu simulácie [40].

2.1.2 Moderné simulátory na báze pohybovej platformy

Jedná sa o veľmi pokročilé simulátory, ktoré okrem spomenutých pravidiel, či grafických prostredí, simulujú taktiež pohyb kabíny automobilu v priestore, čo vytvára ďalší rozmer simulácie a v užívateľovi vyvoláva reálnejší pocit z jazdy. Pohybové platformy sú založené na aktuálnych princípoch využívajúcich hydrauliku, prípadne elektro-mechanický princíp. Tieto simulátory fungujú tak, že softvér simulátoru prepočítava sily pôsobiace na kabínu vozidla, vrátane náklonov, ktoré potom simuluje pomocou samotnej pohybovej platformy. Výpočítane údaje sa pomocou ovládača platformy interpretujú na samotné hydraulické alebo elektro-mechanické aktuátory, ktoré svojou súčinnosťou naklonia kabínu vozidla do požadovaných pozícií. Tieto simulátory sa delia do kategórií podľa počtu simulovaných osí rotácii [7].



Obr. 2.1: Profesionálny simulátor vozidla [41].



Obr. 2.2: Moderná pohybová plošina [10].

2.1.3 Desktopové simulátory

Sú to aplikácie vo forme počítačových hier pre desktopové zariadenia. Ich účelom je vzdelávanie hrou. Sú to väčšinou softvérové riešenia, ktoré podporujú viacero druhov ovládacích prvkov. Priaznivci uvedeného žánru využívajú na ovládanie prvky v tvare volantu. Pre užívateľov, ktorí špeciálne ovládače nevlastnia, je k dispozícii možnosť využiť štandardné ovládacie počítačové prvky, ako sú klávesnica a myš.

Na rozdiel od profesionálnych simulátorov, tieto nezvyknú byť založené na architektúre klient-server, ale sú iba samostatnými klientmi, z ktorých je zložité získať aktuálne dáta za účelom analýzy. Tieto simulátory využívajú multimediálne efekty a spestrujú tak užívateľovi jazdný zážitok pokročilou grafikou, prípadne vierohodnými akustickými prvkami. Pretože sa tento projekt bude zaoberať práve predstaveným typom simulátorov, bude nasledovať analýza troch v súčasnej dobe existujúcich riešení.

Driving School Simulator

Simulátor je produktom firmy Ovilex Software. Je dostupný pre zariadenia iOS, Android, ale i desktop PC. Jedná sa o simulátor z roku 2016, ktorý však mal od roku 2014 už niekoľko vydaní [28].

Medzi kladné stránky tohto simulátora patrí obsiahnutie počítačom riadenej dopravy či simulačná scéna zasadená do viacerých prostredí. Aby simulátor zaujal svojich užívateľov, poskytuje im možnosť výberu z viacerých typov vozidiel a viacerých modelov pre vybraný typ.

Ako bonus počíta tzv. experience points a udeľuje užívateľovi hráčsku úroveň. Medzi hlavné nedostatky patrí benevolentná kontrola pravidiel cestnej premávky. Systém nepenalizuje zmeny jazdných pruhov bez použitia smerovky. Naopak pri rozjazde vozidla v pruhu neopodstatnene požaduje od užívateľa, aby použil ľavú smerovku. Navyše umožňuje iba jazdu s automatickou prevodovkou. Uvedený titul taktiež nerieši pravidlo pravej ruky, tzn.: prednosť z prava.



(a) Zobrazenie dát [36].



(b) Exteriér vozidla [29].

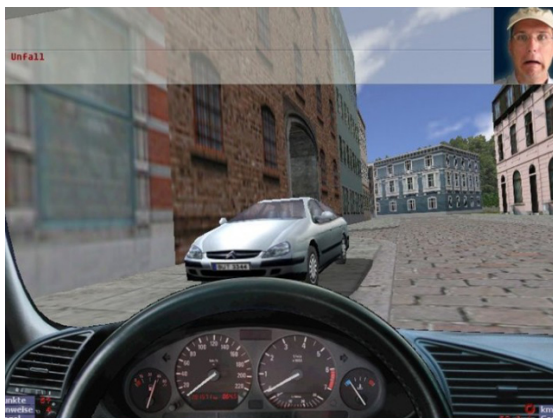
Obr. 2.3: Náhľad na aplikáciu "Driving School Simulator" [28].

3D Fahrschule

Vydanie prvej verzie tohto simulátora sa datuje do konca 90-tých rokov 20. storočia. Jedná sa o SW, ktorý môže byť použitý ako profesionálny simulátor, má však aj verzie pre domáce použitie. Výrobcom je spoločnosť 3D-edutainment a je určený pre zariadenia pracujúce s operačným systémom Windows [2].

Medzi silné stránky tohto simulátora patrí opäť prítomnosť dopravy riadenej počítačom. Simulátor taktiež prináša novinku vo forme kontroly pohľadu vodiča, pred vjazdom do križovatiek vyžaduje kontrolu slepých uhlov a spätných zrkadiel, čo podvedome vedie užívateľa k osvojeniu správnych návykov. Ďalšou silnou stránkou je dôkladná kontrola pravidiel cestnej premávky. S výnimkou nulovej tolerance vjazdu do križovatky, čo neumožňuje vojsť do križovatky do takej miery, aby mal užívateľ náležitý rozhľad na celú križovatku. Bonusom simulátora je cvičisko s kuželmi.

Negatívnou stránkou je však podľa užívateľských hodnotení všetko ostatné, od grafiky, cez zvuky, fyzikálny model až po samotné ovládanie.



(a) Odhalenie priestupku [5].



(b) Exteriér vozidla [12].

Obr. 2.4: Náhľad na aplikáciu "3D Fahrschule" [2].

City Car Driving

Jedná sa o simulátor, ktorého vývoj prebehol v Rusku. Distribútorom simulátoru je firma Forward development a je určený pre zariadenia s operačným systémom Windows [8].

Medzi kladné vlastnosti, je možné zaradiť implementovanú umelú inteligenciu, ktorá riadi dopravu a možnosť výberu z viacerých prostredí. Premávku je možné nastavovať aj súčasne meniť vlastnosti premávky, akými sú agresivita a hustota dopravy. Spracovanie simulátoru je veľmi detailné, s užívateľsky príjemnou grafikou. Simulátor opäť poskytuje možnosť výcviku na cvičisku. Záporným vlastnostiam dominuje príliš zložité ovládanie, a obmedzená stabilita programu vedúca k stavom, kde program prestane pracovať. Samotné riadenie vozidla môže iritovať užívateľov, ktorí nevládajú špeciálne predurčené ovládače ako napríklad volant alebo joystick.



(a) Interiér vozidla [42].



(b) Exteriér vozidla [3].

Obr. 2.5: Náhľad na aplikáciu "City Car Driving". [8]

Záver analýzy existujúcich riešení

Realizovaný rozbor ukazuje, že najväčším problémom uvedených simulátorov je zložitosť a spôsob ovládania. Ďalším bodom potenciálneho zlepšenia je neúplnosť kontroly pravidiel cestnej premávky, a v niektorých prípadoch nedostatočné grafické spracovanie.

2.2 Platné pravidlá cestnej premávky

V skutočnom svete je pozemná doprava riadená predpismi. Jedná sa o právny formát v podobe zákona a vyhlášok doplnujúcich či upravujúcich znenie zákona, ktoré sa zaväzuje dodržiavať každý účastník premávky. Vyhlášky sa týkajú aj osôb, ktoré niesú držiteľmi vodičského oprávnenia. Takými osobami sú napríklad cyklisti, alebo vodiči iných nemotorových vozidiel.

Spomínané predpisy sú pre Slovenskú republiku definované v zákone č. 8/2009 Z. z. [22], ktorý je vykonávaný vyhláškou č. 9/2009 Z. z. [21], Ministerstva vnútra Slovenskej republiky.

Základné dopravné značenia upravujúce rýchlosť jazdy a prednosti v jazde sa podľa vyhlášky členia do troch skupín a to:

- vodorovné dopravné značky,
- zvislé dopravné značky,
- svetelné semafóry.

2.2.1 Vodorovné dopravné značenie

Vodorovné dopravné značky su popisované v §6 vyhlášky č. 9/2009 Z. z. [21].

(1) Vodorovné dopravné značky (č. V 1a až V 17) sú značky zobrazené ako čiary, šípky, nápisy, symboly alebo významové symboly na ceste a jej súčiastiach, samostatne alebo v spojení so zvislými dopravnými značkami, prípadne s dopravnými zariadeniami, ktorých význam zdôrazňujú alebo spresňujú [21].

(2) Vodorovná trvalá dopravná značka je značka zhotovená na povrchu cesty, parkoviska alebo na inej dopravnej ploche aplikovaním náterových látok, termoplastických materiálov, materiálov tvrdnúcich za studena, vopred pripravených materiálov, prípadne inými prostriedkami alebo iným dobre viditeľným a zrozumiteľným spôsobom [21].

2.2.2 Zvislé dopravné značenie

Zvislé dopravné značenie je bližšie špecifikované v §5 vyhlášky č. 9/2009 Z. z. Zvislé dopravné značenie môže byť nositeľom viacerých priorít a druhov dopravných pravidiel či významov. Zvislé dopravné značky sa podľa vyhlášky delia do skupín [21]:

- Výstražné značky (skupina A).
- Značky upravujúce prednosť a dodatkové tabuľky s tvarom križovatky (skupina P).
- Zákazové značky (skupina B).
- Príkazové značky (skupina C).
- Informatívne prevádzkové značky (skupina IP).

- Informatívne smerové značky (skupina IS).
- Informatívne iné značky (skupina II).
- Dodatočné tabuľky (skupina E).

Kde jednotlivé skupiny ustanovujú nasledovné [21]:

- A - Výstražné značky, ktoré upozorňujú na miesta, kde účastníkovi cestnej premávky môže hroziť nebezpečenstvo.
- P - Značky upravujúce prednosť, ustanovujú a spresňujú účastníkom cestnej premávky prednosti v jazde.
- B - Zákazové značky, ustanovujú účastníkovi cestnej premávky zákazy a obmedzenia.
- C - Príkazové značky, ustanovujú účastníkovi cestnej premávky príkazy a povinnosti.
- IP - Informatívne prevádzkové značky.
- IS - Informatívne smerové značky.
- II - Informatívne poskytujúce potrebné informácie dopravného významu alebo slúžia pre orientáciu účastníka cestnej premávky.
- E - Dodatočné tabuľky doplňujúce či spresňujúce iné zvislé dopravné značenie.

Zvislá dopravná značka je vo vyhláške č. 9/2009 Z. z. definovaná ako značka zobrazená na podkladovej tabuli, ktorá je umiestnená zvislo na stĺpiku alebo na inej nosnej konštrukcii pevne spojenej so zemou [21].

2.2.3 Riadenie cestnej premávky svetelnými signálmi

Riadenie cestnej premávky svetelnými zariadeniami je ustanovené v §9 vyhlášky č. 9/2009 Z. z., kde je uvedené, že sa využívajú svetelné signály najmä trojfarebnej sústavy s plnými svetlami alebo so smerovými signálmi [21].

Vyňatok z vyhlášky: (3) Pri riadení cestnej premávky na križovatke a) signál s červeným svetlom „Stoj!“ znamená pre vodiča povinnosť zastaviť vozidlo pred priečnou súvislou čiarou (č. V 5a až V 5c), a kde taká čiara nie je, pred svetelným signalizačným zariadením, b) signál so súčasne svietiacim červeným svetlom „Stoj!“ a žltým svetlom „Pozor!“ znamená pre vodiča povinnosť pripraviť sa na jazdu, c) signál s plným zeleným svetlom „Voľno“ znamená pre vodiča možnosť pokračovať v jazde, a ak dodrží ustanovenia o odbočovaní a také odbočovanie nie je zakázané dopravnou značkou, možnosť odbočiť vpravo alebo vľavo a ak dodrží ustanovenia o otáčaní a otočenie dovoľuje dopravná značka, možnosť otočiť sa; vodič je pritom povinný dať prednosť chodcom prechádzajúcim vo voľnom smere po priechode pre chodcov a cyklistom prechádzajúcim vo voľnom smere po priechode pre cyklistov, pritom ich nesmie ohroziť ani obmedziť; ak svieti signál pre opustenie križovatky (č. S 4) umiestnený v protilahlom rohu križovatky, neplatí pre odbočovanie vľavo alebo vpravo § 19 ods. 4 zákona, d) signál s plným žltým svetlom „Pozor!“ znamená pre vodiča povinnosť zastaviť vozidlo pred priečnou súvislou čiarou (č. V 5a až V 5c), a kde taká čiara nie je, pred svetelným signalizačným zariadením; ak je však vozidlo pri rozsvietení tohto signálu už tak blízko, že by vodič nemohol vozidlo bezpečne zastaviť pred priečnou súvislou čiarou (č. V 5a až V 5c), a kde taká čiara nie je, pred svetelným signalizačným zariadením,

smie pokračovať v jazde; ak svetlo tohto signálu svieti prerušovane, nejde o križovatku s premávkou riadenou svetelnými signálmi

(4) Na skončenie platnosti signálu so zeleným svetlom „Voľno“ podľa odseku 3 písm. c), e) a g) možno vodiča vozidla upozorniť pravidelným striedavým zhasínaním a rozsvetovaním tohto signálu [21].

2.3 Analýza zaznamenaných dopravných priestupkov

V skutočnom svete sa každým dňom stávajú dopravné nehody zapríčinené zlým technickým stavom vozidiel, porušením pravidiel cestnej premávky alebo kombináciou týchto dvoch faktorov. Pre účely realizácie tejto práce je zaujímavé zameranie sa na nehody spôsobené porušením pravidiel. Skúmanie hlavných príčin nehodovosti z hľadiska porušovania pravidiel je veľmi dôležité pre pochopenie zmýšľania vodičov a umožní lepšie určiť okruh pravidiel, na ktoré by mal simulátor dbať a ktoré naopak môže opomenúť, alebo nesimulovať až do detailu. Táto analýza bude vychádzať z vyhodnotenia dopravno-bezpečnostnej situácie za rok 2015, ktorú vydalo dopravné oddelenie policajného zboru Slovenskej Republiky [25].

Tabuľka 2.1: Tabuľka príčin vzniku dopravných nehôd za rok 2015 [25].

Poradie	Smrteľné dopravné nehody	Dopravné nehody
1	Porušenie povinnosti vodiča (nevenovanie sa vedeniu vozidla, telefonovanie a pod.)	porušenie povinnosti vodiča (nevenovanie sa vedeniu vozidla, telefonovanie a pod.)
2	neprimeraná rýchlosť jazdy	neprimeraná rýchlosť jazdy
3	nesprávny spôsob jazdy	nesprávne otáčanie a cúvanie
4	porušenie ustanovení o chodcoch	nesprávna jazda cez križovatku
5	nesprávna jazda cez križovatku	porušenie povinností účastníka cestnej premávky
6	nesprávne predchádzanie	nedodržanie vzdialenosti medzi vozidlami
7	porušenie povinností účastníka cestnej premávky	nesprávny spôsob jazdy
8	nesprávne otáčanie a cúvanie	porušenie ustanovení o chodcoch
9	nesprávne správanie sa na železničnom priecestí	nesprávne odbočovanie
10	nesprávna jazda v jazdných pruhoch	nesprávne predchádzanie

Zo štatistických údajov plynie, že v percentuálnom počte z usmrtených je 51% vinníkov a 49% nevinných [25]. Z toho je možné vyvodiť záver, že nedostatočné vzdelanie a nesprávne

návyky vodičov vedú k verejnému ohrozovaniu a nešťastiam (Obr. 2.6 a obr. 2.7). Aj preto sú simulátory potrebné vo fáze prípravy vodičov v autoškolách ale aj mimo nej.

Výsledok analýzy a vytýčenie prioritných pravidiel

Príčiny v tabuľke 2.1 sú zoradené od najvyššieho výskytu po najnižší. Príčinu číslo 1 nie je možné simulovať za bežných podmienok, pretože sa očakáva, že užívateľ simulátora nebude telefonovať, ide o odlišnú situáciu ako nastáva v skutočnom vozidle, kde pri tomto priestupku zohráva svoju rolu viacero faktorov. Simuláciu si užívateľ zapne väčšinou vo voľnom čase, naopak vo vozidle sedia aj ľudia ktorých riadenie vozidla nebaví, alebo je na nich vyvíjaný časový nátlak.



Obr. 2.6: Vzniknutá škoda v dôsledku dopravnej nehody [23].



Obr. 2.7: Dôsledok nesprávnych návykov vodiča [24].

2.4 Kontrola dodržiavania pravidiel cestnej premávky

Neodmysliteľnou súčasťou dopravných simulátorov je kontrola dodržiavania dopravných predpisov. Väčšinou sa jedná o proprietárne heuristiky a spôsoby kontroly pravidiel, preto niesú svojími autormi zverejňované. Vo všeobecnosti je však možné určiť akými typmi pravidiel sa simulátory zaoberajú. Z analýzy už existujúcich simulátorov vyplýva, že sa zaoberajú predovšetkým kontrolou rýchlosti, prejazdu svetelnými križovatkami, a dodržiavania zvislých dopravných značiek. Trochu v ústraní je kontrola vodorovného značenia, čo je spôsobené komplikovaným spôsobom kontroly tohto typu značenia, pretože sa jedná o súvislé, na seba naväzujúce sekvencie pravidiel, pri vyhodnocovaní ktorých je potrebné brať do úvahy mnoho okolností, ktoré sa dynamicky menia. Navyše takéto kontroly vodorovného značenia vyžadujú istú dávku výpočtového výkonu, a dlhý testovací cyklus. Niektoré simulátory taktiež simulujú nečakané situácie ako sú napríklad technické poruchy vozidla, kde sa merá čas reakcie vodiča napríklad na rozsvietenie indikátora nízkej hladiny motorového oleja, prípadne sa simulujú situácie, kde je narušený jazdý pruh prekážkou. Zaujímavým prvkom sú krízové situácie kde dochádza k náhlemu vstupu chodca alebo vysokej zveri na vozovku.

Kapitola 3

Návrh architektúry simulátoru

Táto kapitola popisuje návrh architektúry simulátora z pohľadu systémového návrhu systému a príslušného užívateľského rozhrania. V záverečnej časti kapitoly budú analyzované možné nástroje pomocou ktorých by bolo možné navrhnutú architektúru realizovať.

3.1 Návrh architektúry systému

Návrh architektúry aplikácie by mal brať v úvahu rozsiahlosť prostredia a preto bude zvolený prístup kde je celá aplikácia jedná veľká scéna vrátane hlavného menu a sumáru výsledkov z jazdy. Tento prístup je zvolený s myšlienkou načítať všetky potrebné súčasti pred spustením celej aplikácie a tým minimalizovať množstvo času potrebného pre načítanie skriptov a prvkov scény, ako pasívnych tak aj aktívnych, pri prechode z hlavného menu do samotnej simulácie, a taktiež aj pri prípadnom opätovnom spustení simulácie po jej ukončení. Predchádzajúci slovný popis je vyobrazený na obrázku 3.1.

Stav "Hlavné menu" bude nastavený ako východiskový stav, to znamená, že po spustení aplikácie bude tento stav pripravený a viditeľný. Uvedený stav bude realizovaný tak, že v hlavnej scéne sa vyberie miesto, ktoré bude tvoriť dynamické pozadie hlavného menu. Následne sa na toto miesto uloží nefunkčný model vozidla, ktorý bude slúžiť na dekoratívne účely a v pravej strane obrazovky sa vyobrazia ikony "Play" a "Exit", ktoré umožnia užívateľovi spustiť simuláciu alebo ukončiť aplikáciu.

Stav "Bežiacia simulácia" reprezentuje samotný beh simulácie, počas ktorého dochádza ku kontrole porušení simulovaných dopravných predpisov a v prípade, že dojde k odhaleniu porušení predpisu, tak konečný automat prejde do stavu "Registrácie porušenia pravidla".

Stav "Registrácia porušenia pravidla" ma za úlohu zaevidovať porušenie predpisu a posunúť automat do stavu "Informácia o porušení dopr. predpisu"

Stav "Informácia o porušení dopr. predpisu" vyobrazí v hornej časti obrazovky správu informujúcu užívateľa o porušení dopravného predpisu, a následne prejde späť do stavu bežiackej simulácie, kde bude simulácia pokračovať.

Stav "Výpis všetkých porušených predpisov" vypíše zoznam všetkých porušených predpisov počas celej jazdy, a po stlačení tlačidla "OK" sa aplikácia prepne do stavu hlavného menu, kde opäť môže užívateľ ukončiť aplikáciu, prípadne si simuláciu zopakovať.

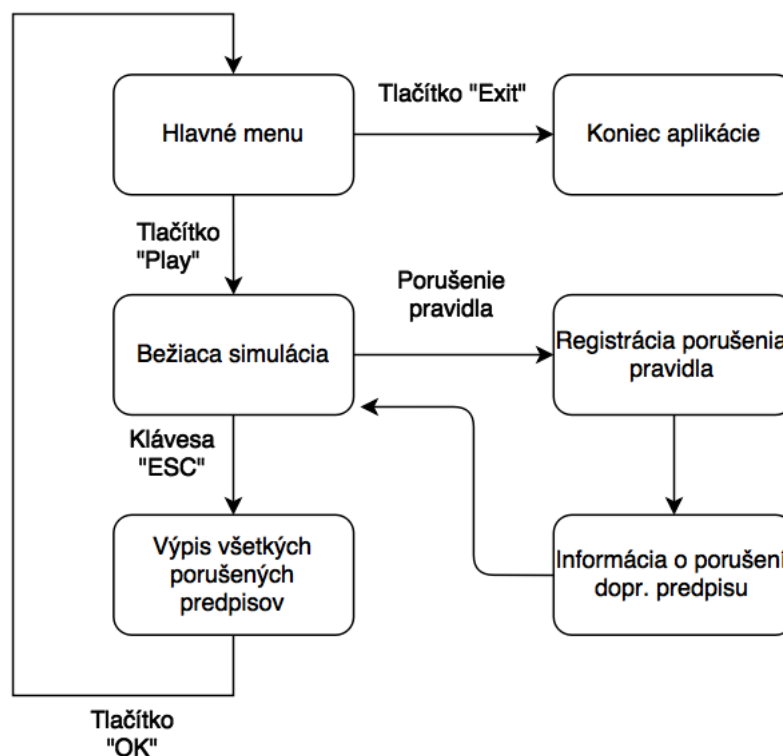
3.2 Návrh užívateľského rozhrania

Táto kapitola popisuje návrh užívateľského rozhrania, vhodný spôsob ovládania simulátoru a spôsob interpretácie údajov simulácie užívateľovi.

Užívateľské rozhranie definuje spôsob komunikácie užívateľa s programom. Je nevyhnutné navrhnuť užívateľské rozhranie tak, aby umožňovalo efektívne ovládanie simulátoru, a nepredstavovalo pritom pre užívateľa prekážku. Ďalej by malo rozhranie užívateľovi umožniť intuitívne ovládanie celej simulácie.

3.2.1 Ovládacie prvky

Vzhľadom k charakteru aplikácie, kde sa simuluje jazda vozidlom po ceste, je vhodné využívať analógové ovládacie prvky. Takýto druh herných alebo simulačných ovládacích prvkov však nieje súčasťou bežnej výbavy počítačov, a zaobstaranie takéhoto ovládacieho prvku môže byť finančne náročné. Preto sa ponúka riešenie využiť počítačovú myš ako hlavný ovládací prvok. Počítačová myš môže fungovať ako analógovo-digitálny vstup pre simuláciu. Os X pohybu počítačovej myši môže byť využitá ako vstup pre zatáčanie vozidla. V takomto prípade ale bude nutné vyhotoviť grafickú reprezentáciu volantu automobilu v 3D priestore s určitou mierou precíznosti, aby bolo užívateľovi na prvý pohľad jasné, do akej miery pootočil volantom, a vedel, čo môže od automobilu očakávať. Plynový pedál a brzdový pedál by za predpokladu vhodného nastavenia fyzikálneho modelu vozidla mohli byť napojené na digitálny vstup ľavého tlačidla myši (plyn) a pravého tlačidla myši (brzda) pre zjednodušenie ovládania. Ovládanie prevodovky, spojky a smerových svetiel môže byť opäť za predpokladu správneho nastavenia aplikácie napojené na digitálny vstup, ale tento



Obr. 3.1: Architektúra aplikácie.

krát z klávesnice. Ovládacie prvky by mali byť vzhľadom k faktu, že užívateľ bude mať po väčšinu času simulácie jednu ruku položenú na počítačovej myši, v dosahu druhej ruky. Ponúka sa vhodné logické a fyzické rozloženie kláves, vyobrazené na obrázku 3.2.

Pre vyobrazené rozloženie kláves platí nasledujúca sémantika:

- A,Z - Zmena prevodového stupňa (A - smerom hore, Z - smerom dole).
- C - Spojkový pedál.
- Q,E - Smerové svetla.
- W - Výstražné svetla.

Takáto konfigurácia umožní plynulý prejazd zákrutami, a realistickejší pocit z jazdy na rozdiel od konvenčného nastavenia ovládania pomocou šípok na klavesnici, kde by zatáčanie nebolo plynulé, a mohlo pôsobiť na užívateľa negatívnym dojmom.



Obr. 3.2: Rozloženie tlačidiel ovládania na klávesnici.

3.2.2 Zobrazenie simulačných dát

Ďalšou súčasťou užívateľského rozhrania simulátoru je sprostredkovanie simulačných dát ako je napríklad v prípade automobilového simulátoru rýchlosť vozidla, otáčky motora a podobne. Do tejto oblasti užívateľského rozhrania spadá taktiež množina informácií vyplývajúcich z porušenia dopravných predpisov.

Existuje viacero spôsobov, akými je možné sprostredkovať užívateľovi prevádzkové data o vozidle.

Jedná z možných variant, je realizácia 3D ukazateľov dát priamo v scéne. V prípade ukazateľov prevádzkových dát vozidla, by sa jednalo o realistickú 3D palubnú dosku umiestnenú v interieri vozidla, s funkčnými ručičkami, poprípade LCD displejmi. Pozícia takto navrhnutých ukazateľov je voči kamere absolútna, a ich svetelné vlastnosti sú ovplyvňované ich okolím a svetelným zdrojom v priestore scény, čo prispieva k pocitu realizmu zo simulácie. Zobrazenie informácií o porušení predpisov druhým spôsobom by ale bolo nevhodné, z dôvodu, že by si ho užívateľ nemusel vôbec všimnúť.

Druhá možná, a vcelku bežná varianta, je zobrazenie prístrojov na palubnej doske automobilu, ako sadu virtuálnych ukazateľov realizovaných vo vrstve prekrývajúcej 3D scénu. Táto vrstva je stále rovnako viditeľná, bez ohľadu na náklon kamery, a jej vykreslenie neovplyvňuje v žiadnom prípade zdroj svetla v scéne a pôsobí akoby bola na kamere "nalepená".

Pozicionovanie prvkov v takejto vrstve je vždy relatívne voči kamere a umiestnenie samotných prvkov užívateľského rozhrania je z pohľadu užívateľa nemenné. Takto realizované zobrazenie väčšiny simulačných dát môže pôsobiť nerealisticky.

Preto sa bude užívateľské rozhranie realizovať kombináciou oboch vyššie spomenutých spôsobov.

- Prevádzkové informácie o vozidle budú vyobrazené obdobným spôsobom aký bol popísaný v prvej variante, tj. realizáciou 3D ukazateľov formou modelovaním funkcionality palubnej dosky.
- Informačné správy o porušení dopravného predpisu budú realizované spôsobom popísaným v druhej novej variante, a to vo vrstve prekrývajúcej 3D scénu.

3.3 Dostupné nástroje pre realizáciu návrhu

V tejto podkapitole budú popísané niektoré dostupné nástroje, pomocou ktorých by bolo možné navrhnutú architektúru zrealizovať.

3.3.1 XNA herné jadro

XNA je súbor nástrojov vytvorených spoločnosťou Microsoft, určených pre vedenie a vývoj projektov, primárne zameraných na tvorbu videohier. XNA je založené na technológii .NET, s možnosťou exportu výslednej aplikácie pre platformy Windows NT, Windows Phone a Xbox 360. Obsah je vytváraný pomocou nástroja XNA Game Studio a následne spúšťaný pomocou špeciálneho prehrávača XNA. Ďalšou možnosťou je export do binárnych súborov a spustiteľných .exe súborov [14].

3.3.2 LeadWerks

LeadWerks je vývojové prostredie vytvorené rovnomenou spoločnosťou LeadWerks. Jedná sa o súbor nástrojov a editorov určených pre tvorbu 2D a 3D interaktívneho obsahu. Pomocou tejto sady nástrojov je možné spravovať, či upravovať textúry, normal mapy, osvetľovacie mapy či vytvárať časticové systémy. Pre tvorbu samotnej scény je určený editor scén. Tento engine využíva pre výpočet fyzikálnych javov knižnicu newton physics. Jedná sa o platený software [16].

3.3.3 Unity 3D

Je to multiplatformové vývojové prostredie, používaný na vývoj hier pre desktopové, mobilné a webové prostredie. Pomocou Unity je taktiež možné vytvárať hry pre PlayStation 3, PlayStation Vita, Wii-U, Xbox 360. Z desktopových platforiem sú to Linux, Mac OS a Windows. Z mobilných zariadení sa jedná o Windows mobile, iOS, Android. Pre webové prostredie je potreba mať nainštalovaný web player a flashplayer [11].

3.3.4 Irrlicht 3D

Multiplatformová Open-source render knižnica pre jazyky C++/C#. Knižnica obsahuje sadu funkcií a algoritmov využívaných pre vyobrazovanie scén a objektov v reálnom čase [15].

3.3.5 Bullet SDK

Multiplatformový fyzikálne jadro využívané celosvetovo v komerčných aplikáciach. Bullet možno vidieť v akcii napríklad v počítačovej hre GTA IV kde je súčasťou proprietárneho fyzikálneho systému Endorphine. Disponuje detekciou kolízií v 3D priestore, simuláciou elastických a pevných telies. Používa sa v počítačových hrách a vo filmových efektoch. Bullet je dostupný pod zľib licenciou [6].

3.3.6 Nvidia PhysX

NVIDIA® PhysX® je fyzikálne jadro poskytujúce fyzikálne prepočty v reálnom čase, ktoré je využívané v mnohých moderných PC hrách. PhysX je optimalizovaný pre vykonávanie výpočtov mnohými jadrami GPU alebo CPU procesorov. GPU rodiny GeForce s podporou pre PhysX zrýchľuje celý proces behu fyzikálnej simulácie pomocou využitia nevyužívanej dostupnej kapacity grafických procesorov pre fyzikálne výpočty, čím odbreňuje CPU, a preto je toto fyzikálne jadro za určitých podmienok mnohokrát výkonnejšie oproti iným, dostupným produktom. Toto fyzikálne jadro je priamo vbudované do Unity 3D, kde sa využíva ako hlavný prostriedok simulácie fyzikálneho priestoru [27].



Obr. 3.3: Logá uvedených nástrojov

3.4 Priorita pravidiel pre implementáciu

Po návrhu architektúry systému a užívateľského rozhrania, je potrebné vymedziť, akými pravidlami cestnej premávky sa bude navrhovaný simulátor zaoberať. Podstatou tohto návrhu je vybrať určitú podmnožinu platných dopravných pravidiel a v rámci vybranej podmnožiny určiť prioritu implementácie.

Vzhľadom k predchádzajúcej analýze, z ktorej vyplýva, že významným faktorom pri vážnych dopravných nehodách je nedodržanie maximálnej povolenej rýchlosti na nebezpečných úsekoch, bude simulátor prioritne dbať na kontrolu maximálnej povolenej rýchlosti, a bude sa zameriavať na simuláciu nebezpečných úsekov, hlavne horských priechodov.

Ďalej, budú pridelené priority implementácie pre jednotlivé pravidlá podľa nasledujúceho zoznamu, kde sú položky zoradené zostupne od najvyššej priority, po najnižšiu

- Maximálna povolená rýchlosť v obci.

- Maximálna povolená rýchlosť mimo obec.
- Prejazd križovatkou upravenou svetelným zariadením.
- Prejazd križovatkou upravenou značkou "STOJ, DAJ PREDNOSŤ V JAZDE!".
- Koniec hlavnej cesty.
- Kontrola vjazdu do ulice s označením značkami zákazu vjazdu, zákaz odbočenia, prikázaný smer jazdy, prípadne ich kombináciou.
- Pravidlá pasívneho charakteru, kde značky slúžia iba pre informatívne účely, a je na zväžení užívateľa, akú dôležitosť im priradí.

Kapitola 4

Fyzikálne jadro, vývojové prostredie a modely

Táto kapitola uvádza popis fyzikálneho jadra simulátoru a priestorových modelov použitých v simulátore.

4.1 Fyzikálne jadro

Navrhovaný simulátor má za úlohu simulovať nielen dopravné predpisy, ale aj vozidlo autoškoly so základným fyzikálnym modelom. Fyzikálny model sa simuluje odlišným spôsobom ako kontrola pravidiel. Aby bol aj tento požiadavok naplnený, je potrebné vybrať vhodný spôsob, ktorým by sa dal zrealizovať. Existujú dva rôzne prístupy používané v praxi.

Prvou možnosťou je naprogramovať celý fyzikálny systém od začiatku, podľa potreby danej aplikácie. Tento prístup je vhodný najmä pre veľké, profesionálne simulátory. Výhodou tohto typu fyzikálnych systémov je fakt, že sú navrhované a koncipované podľa potrieb konkrétneho projektu, a nehladá sa na ich iné, alternatívne využitie. Svoj účel spĺňajú lepšie ako univerzálne fyzikálne systémy, ale za cenu času potrebného pre vývoj takéhoto fyzikálneho systému "na kľúč". S prihliadnutím k tomu, že táto práca sa zaoberá simulátorom, ktorý by mal byť vyhotovený v relatívne krátkom čase, možnosť tvorby vlastného fyzikálneho systému neprípada v úvahu najmä kvôli časovej náročnosti a preto je potrebné zvážiť ďalšie dostupné alternatívy.

Možným východiskom je využitie, už existujúcich, viacúčelových univerzálnych fyzikálnych systémov. Medzi najznámejšie patria systémy Bullet, Newton, ODE, alebo PhysX. V kapitole tretej kapitole boli popísané dva z nich a to Bullet [6] a PhysX [27]. Jedná sa o fyzikálne systémy, ktoré sú dlhodobo vyvíjané nezávislými skupinami (napr. v prípade Bullet), alebo renomovanými spoločnosťami (napr. v prípade PhysX). Tieto fyzikálne systémy sa využívajú v širokom spektre aplikácií, či profesionálnych simulácií. Najviac ich využíva herný a filmový priemysel, za účelom tvorby realistických scén. Univerzálne fyzikálne systémy implementujú širokú škálu fyzikálnych subsystémov, preto niesú pre konkrétny typ aplikácie výpočtovo optimálne v porovnaní so systémami, ktoré vznikli pre účely konkrétnej simulácie.

Pre potreby navrhovaného simulátoru bude využitý univerzálny, viacúčelový fyzikálny systém.

4.1.1 Vybraný fyzikálny systém

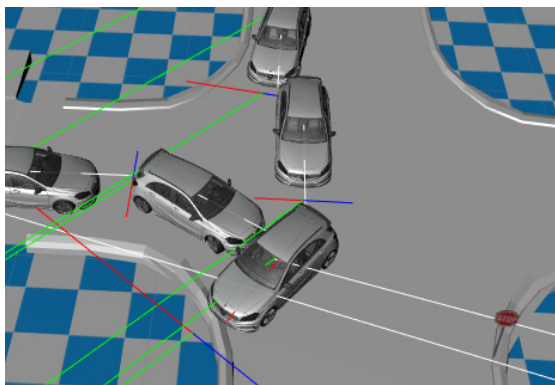
Ako bolo uvedené v predchádzajúcej kapitole, PhysX je univerzálny, viacúčelový fyzikálny systém vyvíjaný spoločnosťou Nvidia. Navrhovaný simulátor autoškoly bude využívať tento systém z dvoch dôvodov. Prvým a hlavným dôvodom je, že vývojové prostredie, ktoré bolo vybrané pre implementáciu tohto simulátora, je od počiatku jeho vývoja navrhnuté tak, aby dokázalo efektívne spolupracovať s týmto fyzikálnym systémom. Druhým dôvodom je možná HW akcelerácia fyzikálnych výpočtov u zariadení, ktoré majú vo svojom HW vybavení zakomponované moderné grafické karty od spoločnosti Nvidia, čo v nemalom merítku prispeje k zníženiu náročnosti aplikácie na CPU zariadení [27].

PhysX ďalej umožňuje využívanie takzvaných fyzikálnych vrstiev. Jedná sa o logické rozdeľovanie objektov fyzikálnej simulácie do skupín, ktorým je možné nastavovať rôzne atribúty. Popísané zoskupovanie objektov navyše zjednodušuje kontrolu nad vzájomnými vzťahmi jednotlivých objektov z hľadiska detekcie kolízie. Príklad: nech A, B, C sú tri rôzne skupiny telies existujúcich vo fyzikálnom systéme. Nastavením boolovej hodnoty kolízií medzi skupinou A a B na hodnotu true, medzi skupinou A a C na hodnotu true a medzi skupinou B a C na hodnotu false, je dosiahnutý efekt, kde telesá zo skupiny A budú kolidovať s telesami zo skupiny A, B a C, telesá zo skupiny B a C budú kolidovať s telesami z vlastnej skupiny a telesami zo skupiny A.

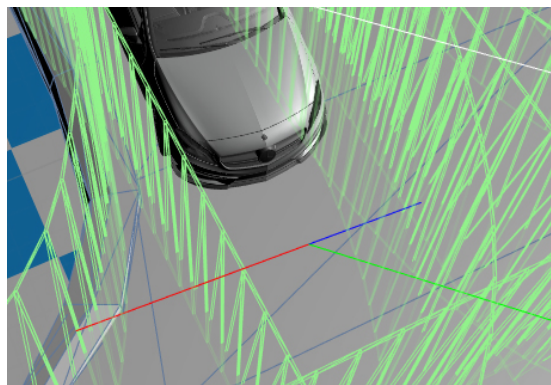
Systém PhysX podporuje definíciu fyzikálnych vlastností povrchu telies, teda umožňuje vytvárať fyzikálne relevantné materiály, ktorých vlastnosti následne berie zohľadňuje pri simulácii síl v priestore fyzikálnej simulácii. Väčšinou sa jedná o nastavenia vlastností akými sú napríklad hustota, pružnosť, miera pohĺtenia energie dopadajúceho telesa, koeficient povrchového trenia a im podobné [27].

4.1.2 Ray-casting vo fyzikálnom systéme

Raycasting je vo všeobecnosti metóda pomocou ktorej je možné získať informácie o konkrétnom bode scény. Vychádza zo simulácie projekčného lúča smerujúceho od jeho zdroja po najbližší bod dopadu na povrch objektu (Obr. 4.1 a obr. 4.2). Základným parametrom je maximálna vzdialenosť, po ktorú je lúč platný a po ktorú môže byť hľadaný bod nájdený.



Obr. 4.1: Raycast v praktickom príklade.



Obr. 4.2: Lúče reagujú iba na nastavení fyzikálnu vrstvu.

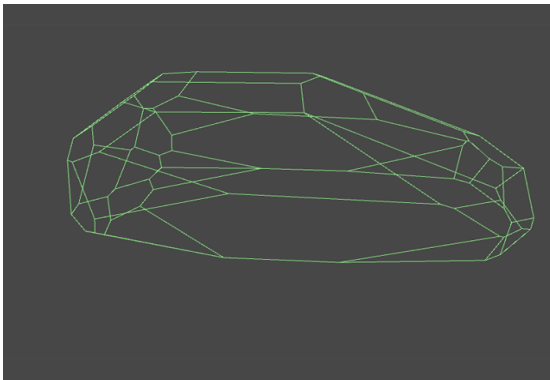
Raycasting je metóda využívaná ako vo fyzike tak aj pri niektorých typoch renderingu realistických scén. V prípade simulátoru autoškoly sa využíva vo fyzikálnom systéme a to pre získavanie informácií o dopravnom značení v niektorých prípadoch a pre kontrolu vzdialeností vozidiel pre účely vozidiel riadených počítačom. V systéme PhysX sa na raycasting vzťahuje rozdelenie objektov do fyzikálnych vrstiev.

4.1.3 Kolízne teleso

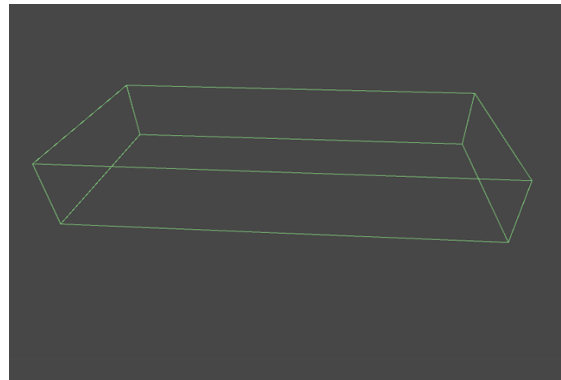
Kolízne teleso alebo inak collider je neviditeľný objekt umiestnený v simulovanom priestore fyzikálneho systému, vytvorené za účelom detekcie kontaktu tohoto telesa s iným kolíznym telesom. Spracovávanie a obsluhu kolíznych telies má na starosti výlučne fyzikálny systém a jednotlivé kolízne telesá vždy spadajú do niektorej z fyzikálnych vrstiev [37].

Kolízne telesá rozdeľujeme do dvoch skupín

- primitívne (Obr. 4.3),
- komplexné (Obr. 4.4).



Obr. 4.3: Komplexné kolízne teleso.



Obr. 4.4: Primitívne kolízne teleso.

Primitívne kolízne telesá sú popisované matematickými funkciami, prípadne šiestimi plochami tj. ôsmimi vrcholmi. Tieto telesá sú vzhľadom k nárokom na výpočtový výkon efektívnejšie a menej náročné v porovnaní s komplexnými telesami. Zápornou stránkou jednoduchých kolíznych telies je fakt, že nedokážu dostatočne precízne popísať zložitejšie tvary objektov.

Prístup pre výpočet kolízií pri zložitých kolíznych telesách je odlišný od primitívnych kolíznych telies. Pri tomto type kolíznych telies hovoríme o N uholníkoch, ktoré môžu byť uzavreté alebo neuzavreté. Počítanie kolízií sa vykonáva pre každú plochu definovanú v tomto N uholníku [11].

Všeobecne platí, že detekcia kolízií je založená na meraní vzdialenosti bodu jedného kolízneho telesa od plochy iného kolízneho telesa. Každé fyzikálne jadro však využíva odlišné techniky, ako kolíziu dvoch telies zaznamenať, preto nieje možné tieto prístupy generalizovať a vytvoriť jeden univerzálny algoritmus. Základný vzťah pre získanie vzdialenosti bodu od roviny, z ktorého sú odvodené niektoré z techník, je následovný:

$$d = \frac{\|K_a \times r_1 + K_b \times r_2 + K_c \times r_3 + K_d\|}{\|n\|} \quad (1)$$

Kde r_1, r_2, r_3 reprezentujú súradnice bodu, ktorého vzdialenosť zisťujeme (r_1, r_2, r_3 sú v poradí os x, y, z).

Koeficienty K_a, K_b, K_c, K_d vyplývajú z popisu roviny voči ktorej vzdialenosť bodu meriame n je odmocnina súčtu druhých mocnín koeficientov K_a, K_b, K_c, K_d [39].

4.1.4 Priestor a orientácia v ňom

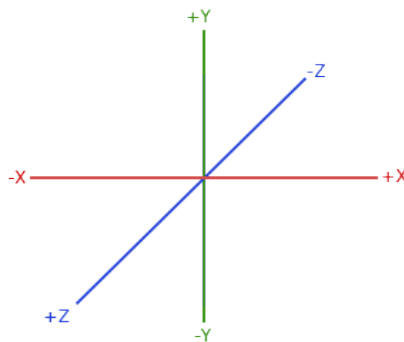
Priestor je z fyzikálneho hľadiska charakteristika hmotných telies určujúca ich vzájomné vzdialenosti a orientácie. Priestor v ktorom žijeme a ako ho dnes poznáme pozostáva z štyroch dimenzií:

- čas,
- dĺžka,
- šírka,
- výška.

Z hľadiska simulácie fyzikálnych javov je takéto chápanie priestoru nevyhnutné. Priestor sa logicky rozdeľuje na časti pomocou troch os x, y, z , (Obr. 4.5) aby s ním bolo možné pracovať, a využívať matematické vzťahy popisujúce tento priestor a javy, ktoré sa v ňom odohrávajú, napríklad pohyb alebo sily a podobne.

Všeobecne pozíciu objektu v takto definovanom priestore určujeme usporiadanou trojicou hodnôt $[x, y, z]$ kde:

- x - vyjadruje hodnotu pozície objektu voči šírke priestoru.
- y - vyjadruje hodnotu pozície objektu voči výške priestoru.
- z - vyjadruje hodnotu pozície objektu voči dĺžke priestoru.



Obr. 4.5: Súradný systém [17].

Ako je patrné z predchádzajúceho textu, štvrtý rozmer, ktorým je čas, nemá podiel na samotnej pozícii objektu. Má však podiel na zmene pozície objektu, tj. pohybu ktorý bude popísaný v jednej z nasledujúcich kapitolách.

Pre rôzne účely jednoznačného mapovania polohy telesa v priestore boli vyvinuté rôzne súradné systémy. Základné členenie súradných systémov podľa priestoru ktorý popisuje:

- Lokálne súradné systémy.
- Globálne súradné systémy.

Rôzne súradné systémy môžu meniť významy jednotlivých osí priestoru.

V prípade lokálnych súradných systémov hovoríme o systémoch, ktoré uvažujú pozíciu telesa vrámci svojho vlastného, lokálneho priestoru. Súradnice v lokálnom súradnicovom systéme majú nulový bod $[0, 0, 0]$ reprezentujúci východiskovú pozíciu umiestnený v mieste ich globálnej pozície v nadradenom priestore [19].

V prípade globálnych súradných systémov hovoríme o systémoch, kde jednotlivé hodnoty samotných rozmerov určujú pozície telies voči nadradenému priestoru [19].

Pridaním telesa do lokálneho priestoru iného telesa, sa tento priestor stáva globálnym priestorom pre všetky v ňom existujúce telesá.

V prípade umelých fyzikálnych systémov sú vzájomné vzťahy medzi jednotlivými telesami a súradnými systémami implementované a popisované stromovými grafmi.

4.1.5 Všeobecné pohybové rovnice pre simuláciu

V rámci simulácii autoškoly je pohyb nevyhnutným aspektom, ktorý je nutné zohľadniť a bez ktorého by celá práca strácala význam. V predchádzajúcej podkapitole bolo popísané, čo je to súradný systém a pozícia telesa v rámci tohto súradného systému. Pozícia telesa v priestore je úzko spätá s pohybom telesa v priestore. V prípade pohybu sa berie v úvahu aj štvrtá dimenzia systému, a tou je čas. Pohyb definujeme ako zmenu polohy telesa v čase. Pohyb je možné definovať pomocou matematického aparátu.

$$V_g = u + at \quad (2)$$

$$s = v_0t + \frac{1}{2}at^2 \quad (3)$$

$$s = \frac{1}{2}(u + V_g)t \quad (4)$$

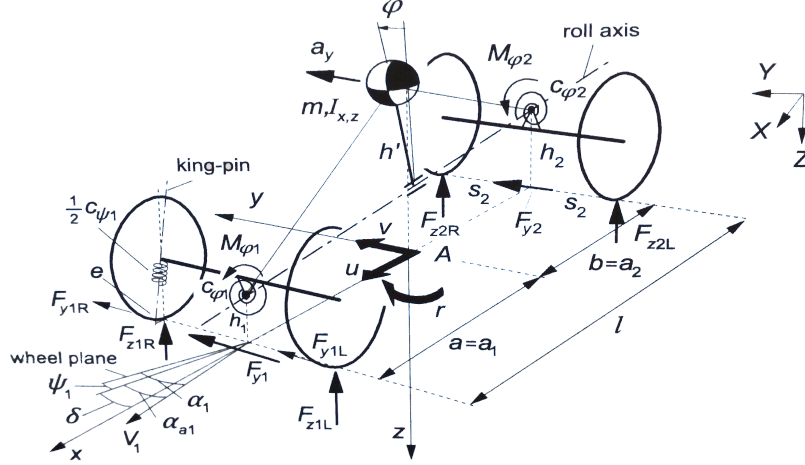
$$V_g^2 = u^2 + 2as \quad (5)$$

$$s = -1 \times V_g t + \frac{1}{2}at^2 \quad (6)$$

Pre účely výpočtu pohybu pre n-rozmerné priestory sa každá z rovníc využije n-krát a to práve jeden krát pre každú z dimenzii okrem času.

Táto teória je postačujúca pre simuláciu jednoduchých objektov. Problémom však je, že sa automobil neradí medzi jednoduché objekty. Predchádzajúce vzťahy nebrali do úvahy kinetickú energiu, rotácie ani náklony či rotácie telies s ktorými pracovali. Všetky tieto faktory sú ale zásadné pri simulácii pohybu automobilu. Fyzik a matematik Hans B. Pacejka vytvoril model, ktorý tieto charakteristiky pohybu berie na zreteľ, a ktorým je možné popísať pohyb automobilu [30]. Vo svojom matematickom modeli vozidla (Obr. 4.6), využíva Euler-Lagrangeové rovnice, ktoré počítajú s automobilovým BFF súradným systémom. V Pacejkovom modeli pohybu jednoduchého vozidla figurujú externé zobecnené sily

(Q_i) asociované so zjednotenými súradnicami (q_i), ktoré môžu pôsobiť na automobil. Interné sily tvorené pružinami a tlmičmi vozidla sa taktiež započítávajú medzi zobecnené externé sily pôsobiace na automobil a spoločne tvoria súčasť celkovej práce W , ktorú musí vozidlo vykonať na to, aby zmenilo svoju pozíciu za jednotku času.



Obr. 4.6: Automobilový súradný systém podľa Pacejku [30].

Lagrangová rovnica pre súradnicu q_i má nasledujúci tvar:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i \quad (7)$$

Pre pohyb po rovine sú z hľadiska rovníc zaujímavé osi x a y polohy vozidla v priestore, ďalej úhol vytočenia prednej nápravy vzhľadom k lokálnej osi X vozidla a úhol náklonu vozidla pri zatáčaní. Pre situácie keď rotácia vozidla okolo osi Z je v relatívne nízkych hodnotách, je možné pohybové rovnice derivovať priamo z rovnice 7. Pre prípady, kde úhol rotácie okolo osi Z , môže naberať vyššie hodnoty, Pacejkov model dopočítava hodnoty rýchlostí u, v a r pohybu lokálneho súradného systému automobilu v rámci globálneho súradného systému pomocou vzťahu(8). Následne hodnoty jednotlivých zložiek u, v a r sú spoločne s hodnotou bočného náklonu vozidla použité ako pohybové premenné [30].

$$\begin{aligned} u &= \dot{X} \cos \psi + \dot{Y} \sin \psi \\ v &= -\dot{X} \sin \psi + \dot{Y} \cos \psi \\ r &= \dot{\psi} \end{aligned} \quad (8)$$

Na základe získaných hodnôt u, v, r je ďalej možné vyjadriť kinetickú energiu vozidla. Prvý člen rovnice (7) pre každú zo súradníc (x, y, ψ) sa vypočíta ako:

$$\begin{aligned}
\frac{\partial T}{\partial \dot{X}} &= \frac{\partial T}{\partial u} \frac{\partial u}{\partial \dot{X}} + \frac{\partial T}{\partial v} \frac{\partial v}{\partial \dot{X}} = \frac{\partial T}{\partial u} \cos\psi - \frac{\partial T}{\partial v} \sin\psi \\
\frac{\partial T}{\partial \dot{Y}} &= \frac{\partial T}{\partial u} \frac{\partial u}{\partial \dot{Y}} + \frac{\partial T}{\partial v} \frac{\partial v}{\partial \dot{Y}} = \frac{\partial T}{\partial u} \sin\psi - \frac{\partial T}{\partial v} \cos\psi \\
\frac{\partial T}{\partial \dot{\psi}} &= \frac{\partial T}{\partial r} \\
\frac{\partial T}{\partial \psi} &= \frac{\partial T}{\partial u} v - \frac{\partial T}{\partial v} u
\end{aligned} \tag{9}$$

Následnou úpravou rovnice (7) po dosadení vyjadrených hodnôt z rovníc (9) je možné eliminovať faktor rotácie entity vozidla okolo osi z a to postupným násobením takto získanej rovnice hodnotami $\cos\psi$ a $\sin\psi$ a následným ich pričítaním a odčítaním. Takto získané rovnice predstavujú vyvážený stav (ekvilíbrio) v smere osí x a y (respektíve u a v) [30].

Pre jednotlivé prvky tohto matematického modelu vznikajú nasledujúce vzťahy:

$$\begin{aligned}
\frac{d}{dt} \frac{\partial T}{\partial u} - r \frac{\partial T}{\partial v} &= Q_u \\
\frac{d}{dt} \frac{\partial T}{\partial v} + r \frac{\partial T}{\partial u} &= Q_v \\
\frac{d}{dt} \frac{\partial T}{\partial r} - v \frac{\partial T}{\partial u} + u \frac{\partial T}{\partial v} &= Q_r \\
\frac{d}{dt} \frac{\partial T}{\partial \dot{\phi}} - \frac{\partial T}{\partial \phi} + \frac{\partial U}{\partial \phi} &= Q_\phi
\end{aligned} \tag{10}$$

Takto vznikajú pohybové rovnice pre pohyb vozidla na rovnom povrchu, do ktorých je ešte potrebné doplniť hodnoty zjednotených externých síl, hodnoty kinetickej energie T a potenciálnej energie U na to, aby boli kompletne [30].

Zjednotené sily Paceykov model vyjadruje z virtuálnej práce δW , ktorá je definovaná ako:

$$\delta W = \sum_{j=1}^4 Q_j \delta q_j \tag{11}$$

Pre modelovaný systém vozidla (hodnoty súradníc x, y, ψ, ϕ) bude táto virtuálna práca vyjadrená nasledovne:

$$\delta W = \sum F_x \delta x + \sum F_y \delta y + \sum M_z \delta \psi + \sum M_\phi \delta \phi \tag{12}$$

Odkiaľ je možné vyjadriť jednotlivé externé generalizované sily pre každú modelom obsiahnutú súradnicu:

$$\begin{aligned}
Q_u &= \sum F_x = F_{x1} \cos\delta - F_{y1} \sin\delta + F_{x2} \\
Q_v &= \sum F_y = F_{x1} \sin\delta + F_{y1} \cos\delta + F_{y2} \\
Q_r &= \sum M_z = a F_{x1} \sin\delta + a F_{y1} \cos\delta + M_{z1} - b F_{y2} + M_{z2} \\
Q_\phi &= \sum M_\phi = -(k_{\phi 1} + k_{\phi 2}) \dot{\phi}
\end{aligned} \tag{13}$$

Ak sa ľavé a pravé koleso nápravy automobilu vyjadria ako pozdĺžne sily, tlmiče podvozku automobilu sú reprezentované výslednými lineárnymi momentmi okolo hlavných osí automobilového súradného systému a ak hodnoty náklonu vozidla spolu s hodnotou sklonu náklonových osí sú považované za nízke ($\theta_r \approx (h_2 - h_1)/l$), potom je možné kinetickú energiu vozidla definovať pomocou nasledujúceho vzťahu [30].

$$T = \frac{1}{2}m((u - h'\phi r)^2 + (v + h'\dot{\phi})^2) + \frac{1}{2}I_x\dot{\phi}^2 + \frac{1}{2}I_y(\phi r)^2 + \frac{1}{2}I_z(r^2 - \phi^2 r^2 + 2\theta_r r \dot{\phi}) - I_{xz}r\dot{\phi} \quad (14)$$

Poslednou zložkou, ktorú je dôležité vyjadriť, je potenciálna energia U . Potenciálna energia U vzniká v pružinách nápravy automobilu, pričom za výšku automobilu od plochy po ktorej jazdí, sa berie vzdialenosť tejto plochy od ťažiska automobilu. Za týchto podmienok je potenciálna energia vypočítaná nasledujúcim vzťahom [30].

$$U = \frac{1}{2}(c_{\phi 1} + c_{\phi 2})\phi^2 - \frac{1}{2}mgh'\phi^2 \quad (15)$$

Dosadením predchádzajúcich vzťahov (13), (14) a (15) do vzťahov (10) vznikajú kompletne pohybové rovnice popisujúce pohyb vozidla po rovinnom podklade. Ak by sa uvažoval podklad s meniacou sa výškou, je potrebné tento fakt zohľadniť v pohybových rovniciach, a pridať ďalšie osi pohybu podľa vzoru rovníc ktoré sa vzťahujú na scény s plochým povrchom [30].

4.1.6 Rovnice popisujúce dynamiku vozidla

V predchádzajúcej kapitole boli objasnené pohybové rovnice na základe Pacejkovho modelu. Simulácia ale musí zohľadniť aj ďalšie subsystemy existujúce v rámci vozidla, aby sprostredkovala realistický pocit z ovládania automobilu. Medzi základné subsystemy či komponenty automobilu patria pohonná jednotka, prevodové ústrojenstvo a diferenciál.

Pohonná jednotka (v prípade riešeného simulátoru budeme pod pojmom motor automaticky uvažovať o atmosférickom spalovacom motore), je systém generujúci krútiaci moment na základe fyzikálnych princípov premeny chemickej energie (paliva) na mechanickú energiu a tepelnú energiu [26]. Pre účely jednoduchého simulátoru stačí pohonnú jednotku modelovať na základe základných znalostí o spalovacích motoroch a pozorovaní správania skutočného spalovacieho motora v skutočnom automobile. Podrobný model s nízkou mierou abstrakcie voči reálnemu systému by sa v uvedenom prípade minul účelu. Model pohonnej jednotky berie v úvahu pomer paliva a vzduchu, mieru a priebeh odozvy na pokyn od plynového pedálu, spätnú väzbu z prevodovky. Výsledkom je množstvo výstupného krútiaceho momentu. Krútiaci moment a priebeh odozvy plynového pedálu sú funkcie mapované pomocou interpolačných polynómov na intervale $\langle 0; RPM_{max} \rangle$, kde hodnota RPM_{max} predstavuje maximálne možné otáčky motora a krokom funkcie nie je čas, ale aktuálne otáčky motora. Prírastok otáčok motora v prípade akcelerácie (stlačeného plynového pedálu) počíta popisovaný model podľa rovnice (16) a pre prípad spomalenia počas jazdy, podľa rovnice (17). Výsledné generované množstvo krútiaceho momentu sa získá z funkcie priebehu krútiaceho momentu na základe novo získaných aktuálnych otáčok motora. Opäť sa bude prepočet výstupu pohonnej jednotky líšiť podľa miery otvorenia škrtiacej klapky. V prípade akcelerácie sa výstupná hodnota zmodifikuje konštantou upravujúcou pomer medzi skutočným a simulovaným prostredím. V prípade spomalenia sa navyše výstupná

hodnota prenásobí zápornou hodnotou spätnej väzby od prevodovky, čo vytvorí simuláciu brzdenia motorom.

$$\Delta RPM = TR \left(\frac{RPM}{1000} \right) \times f_{acce} \times AFR \times \zeta_{TH} \times F_{TB} \quad (16)$$

$$\Delta RPM = -1 \times TR \left(\frac{RPM}{1000} \right) \times F_{TB} \quad (17)$$

Prevodové ústrojenstvo a diferenciál - Prevodové ústrojenstvo má za úlohu prevádzať krútiaci moment generovaný pohonnou jednotkou do viacerých úrovní, a umožniť tak vozidlu dosahovať vyššie rýchlosti a to vhodným posuvom otáčok motora voči rýchlosti otáčania kolies (teda rýchlosti vozidla) pomocou prevodov [26]. Jednotlivé úrovne predstavujú samotné prevodové stupne. Zaradením konkrétneho prevodového stupňa sa upravia prevodové pomery celej pohonnej sústavy a sprístupní sa spektrum otáčok motora (s otáčkami sa spája množstvo generovaného krútiaceho momentu), ktoré by pri rovnakej rýchlosti ale pri zaradení iného prevodového stupňa nebolo možné dosiahnuť. Zaradený supeň je modifikovaný stálym prevodom ktorý nemožno z pohľadu užívateľa voliť. V matematickom modeli pre účely riešeného simulátoru, predstavujú jednotlivé prevodové stupne hodnoty prevodových pomerov vozidla, ktoré na základe aktuálne zvoleného prevodového stupňa figurujú v rovnicach (18) a (19) popisujúcich správanie prevodovky. Výstupom je zmodifikovaná hodnota krútiaceho momentu. Model prevodovky je popísaný nasledujúcim vzťahom.

$$\tau_o = n_G \times \gamma_{DR} \times \tau_i \times \eta_{CT} \quad (18)$$

Následne je dôležité stanovenie dolnej hranice otáčok motora, na základe aktuálnej konfigurácie prevodov.

$$RPM_{min} = \frac{v_{auto}}{2\pi \times R_{wh} \times 60} \times n_G \times \gamma_{DR} \quad (19)$$

Samotná spätná väzba pre pohonnú jednotku je získaná interpoláciou medzi hodnotou 1 a empiricky získanou konštantou, pričom krok tejto interpolácie opäť nieje čas, ale aktuálne otáčky motora.

$$fb = \left(\left(\frac{1}{RPM_{max}} \times RPM \right) \times (\Omega - 1) \right) + 1 \quad (20)$$

Prerozdelenie výstupného krútiaceho momentu medzi kolesá hnacej nápravy, je úlohou diferenciálu. Pre účely simulátora autoškoly a za predpokladu že sa bude jazdiť po pevnom povrchu, bude modelom diferenciálu jednoduchý vzťah, kde sa krútiaci moment predelí rovnomerne medzi poháňané kolesa. Je nutné dodať, že sa jedná o zjednodušenú abstrakciu a v skutočnosti by takto navrhnutý diferenciál neplnil svoju pôvodnú úlohu.

4.2 Vývojové prostredie Unity

Unity 3D poskytuje viacero knižníc a vizuálnych nástrojov pre správu interaktívneho obsahu obsiahnutého v scénach. V Unity je možné programovať, prípadne skriptovať tromi spôsobmi.

- jazyk C#,
- JavaScript,
- Boo script.

Tieto jazyky sú prispôsobené pre použitie v Unity. Menovaný engine poskytuje pre každý z jazykov sadu knižníc, či funkcií vytoverných špeciálne pre toto prostredie [37].

Po zvážení všetkých kladov a záporov každého z vymenovaných jazykov, padlo rozhodnutie, že projekt bude vytvorený v jazyku C#. Hlavným dôvodom bola skutočnosť, že uvedené riešenie umožňuje využívať podporu väčšiny základných knižníc z pôvodnej nemoifikovanej verzie jazyka C#. Ide teda o rozšírenie pôvodného jazyka, nie iba o samotnú adaptáciu jazyka pre prostredie Unity, ako je tomu vo zvyšných dvoch prípadoch. To poskytuje väčší repertoár funkcií a dátových typov, čo súčasne poskytne väčšiu voľnosť pri samotnej implementácii. Unity používa pre účely svojho primárny fyzikálneho systému, fyzikálne jadro PhysX ktoré bolo uvedené na začiatku tejto kapitoly. Súčasťou Unity je sofistikovaný správca zdrojových súborov a editor scény.

4.3 Základná scéna

Všeobecne platí, že základnými prvkami každej scény sú kamera, svetelný zdroj a objekty scény. Scéna je držiteľom informácie o vzájomných vzťahoch entít, ktoré v nej existujú. Tieto vzťahy sú ukladané formou datových stromov. Scéna ako taká, sa v prípade transformácii javí ako globálny súradný systém ktorý sa vzťahuje na všetky v nej entity umiestnené. Transformácie v scéne sa aplikujú vždy na určitý uzol stromu. Ak tento uzol nieje listom, potom je ovplyvnená globálna pozícia všetkých jeho synovských uzlov. Transformácie sú realizované pomocou transformačných matíc, ktoré fungujú ako modifikátory objektov. Transformácia vychádza z aktuálnej pozície entity v priestore scény. Zložité pohyby entity v priestore scény sú zložené z dielčích základných transformácií. Základné transformácie predstavujú posun, rotáciu, skosenie a zmenu mierky entity.

4.3.1 Modely priestorových objektov

Dôležitou súčasťou modelovania prostredia je tvorba priestorových objektov, bez ktorých by scéna reprezentujúca abstraktnú realitu strácala zmysel a simulácia by sa tak odohrávala vo voľnom priestore, čo je pre simulátor autoškoly neprípustné. Pre tvorbu priestorových objektov bol použitý modelovací program Blender.

Blender je open-source software pre modelovanie 3D objektov. Svojím užívateľom poskytuje celú škálu možností od samotnej tvorby 3D objektov, renderingu, tvorby animácií na základe sledovania pohybu pomocou kamery, animovania, pridelovania materiálových vlastností na časti 3D objektov, uv mapovanie textúr na 3D objekty, a mnoho ďalších. Blender sa tak vďaka svojim schopnostiam stáva vhodným nástrojom pre modelovanie technikou subdivision surface modeling, ktorá je vo veľkej miere využitá pri tvorbe modelov pre tento projekt [9].

Subdivision surface modeling je technika pre tvorbu high-poly modelov. Princíp tejto techniky je založený na postupnom zvyšovaní počtu vrcholov tzv.: vertices, za účelom zvýšenia rozlíšenia a zjemnenia povrchu výsledného 3D objektu. Každá iterácia predelí už existujúcu plochu tzv.: face, rovnakým počtom vertikálnych a horizontálnych hrán tzv.: edges [38].

Na základe takto vytvoreného high-poly modelu je možné vytvoriť mapy normálových vektorov a nerovnostných máp, pomocou ktorých je možné vykonať optimalizácie 3D objektov. Optimalizácia 3D objektu spočíva v znížení počtu vrcholov, čím sa zrýchlia prepočty transformácií objektu v priestore, ale na úkor vizuálnej jemnosti a precíznosti. Táto strata sa dá kompenzovať spomínanými mapami normálových vektorov a mapami povrchových nerovností.

Vzájomné polohy vrcholov, jednotlivé plochy a hrany definujú výsledný tvar 3D objektu. Orientácia plochy je popisovaná opäť pomocou normálových vektorov.

4.3.2 Reprezentácia terénu

Pod pojmom terén sa všeobecne rozumie časť zemského povrchu vrátane jeho nerovností, na ktorom sú umiestnené objekty akými sú napríklad flóra, kamene, vodstvo, technické zariadenia, cestné komunikácie a podobne. Terén predstavuje neoddeliteľnú súčasť prostredia v ktorom sa odohráva simulácia, a má vplyv na samotnú simuláciu ako z estetického hľadiska, tak aj z hľadiska funkčného. Správnym navrhnutím terénu, je možné simulovať nebezpečné úseky a rizikové faktory, ktoré môžu v určitých situáciách nastať a môžu pri cestovaní krajinou ovplyvniť bezpečnosť cestnej premávky. Navrhnutý simulátor, na rozdiel od už existujúcich riešení, simuluje rizikové úseky ciest, čo z veľkej miery závisí na vhodnom a prepracovanom modeli terénu. Opäť existuje mnoho spôsobov pomocou ktorých je možné vytvárať terény pre účely simulácie. Každý zo spôsobov má svoje výhody [18].

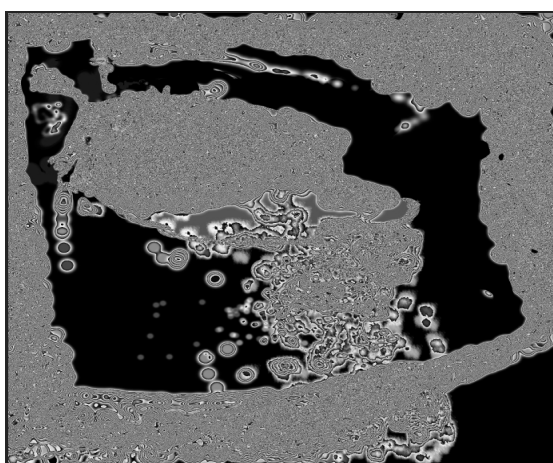
Jedným z možných spôsobov je modelovanie zemského povrchu metódou subdivision surface modeling. Táto metóda bola popísaná v predchádzajúcich kapitolách. Nevýhodou takto tvoreného terénu je pamäťová náročnosť a náročnosť na výpočtový výkon počítača. Takto tvorený terén vyžaduje vyššiu réžiu v prípade modifikácii už existujúceho modelu ale aj pri jeho využívaní. Využívanie takéhoto typu terénu vyžaduje ďalšie operácie.

- Terrain tiling - rozdelenie terénu na menšie, na seba naväzujúce dlaždice. Vyžaduje vytvorenie inteligentného načítania a uvoľňovania samotných dlaždíc podľa ich vzdialenosti od simulovanej oblasti tak, aby nedochádzalo k obmedzovaniu pozitívnej užívateľskej skúsenosti, ale k vytvoreniu ilúzie, že sa jedná o celistvý blok zemského povrchu. Táto metóda sa vyžaduje pri terénoch s väčšou rozlohou. V prípade, že by sa simulácia odohrával v malom priestore, je možné terrain tiling preskočiť, a načítať celý terén ako jeden objekt.
- Tvorba úrovni LOD (Level of detail) - Jedná sa o optimalizáciu vykresľovania objektu vzhľadom k vzdialenosti objektu od kamery. V moderných technológiach ale tento spôsob optimalizácie ide do ústrania a nahrádzajú ho automatické procesy optimalizácie modelu, ktoré sa odohrávajú v reálnom čase. Tieto procesy využívajú rôzne druhy interpolácii a algoritmov Marching cubes.

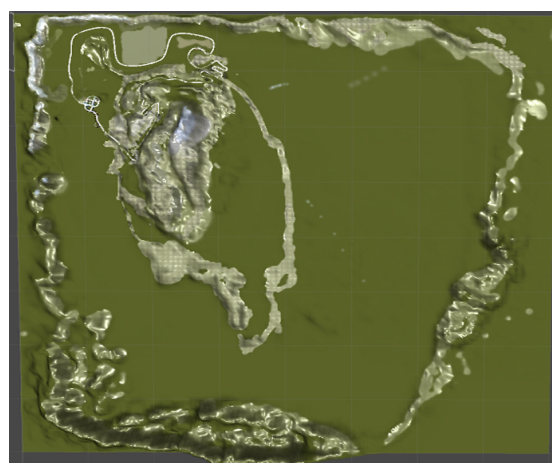
V moderných aplikáciach je terén reprezentovaný volumetrickými dátami. Jednotkou takto vytvoreného terénu je voxel - volumetrický pixel. Voxely sú väčšinou objekty v tvare kocky a ich hladké povrchy a zaoblenia sa získavajú pomocou algoritmov Marching cubes.

Takýto terén je zložený z voxelov a má svoju hĺbku. Pomocou takto tvorených terénov je možné simulovať deformáciu terénu nárazom, alebo prejazdom vozidla. Vzhľadom k HW nárokom takto tvoreného terénu, sa však v simuláciach ešte nevyužíva. Tento typ reprezentácie terénu by bol nevhodným riešením pre realizovaný simulátor.

Použitým spôsobom je generovanie modelu terénu na základe výškovej mapy [18]. Výšková mapa je súbor, ktorý obsahuje informácie o výškovom profile terénu. Hodnota každého pixelu takto tvorenej mapy sa reprezentuje určitou oblasťou terénu, ktorá je položená do určitej výšky od nulovej pozície entity terénu. Výška do akej bude oblasť terénu umiestnená je závislá od nastavenia koeficientov. Tieto koeficienty sa správajú ako modifikátor a modifikujú hodnoty pixelov vo výškových mapách. Výšková mapa môže byť uložená v rôznych formátoch. V prípade unity sa jedná o formát RAW a jeho 16 bitovú verziu, v ktorej je hodnota každého pixelu vyjadrená šesťnástimi bitmi (Obr. 4.7). Výsledný terén je na obrázku č.4.8.



Obr. 4.7: Výšková mapa terénu vo formate RAW.



Obr. 4.8: Výsledok interpretácie výškovej mapy.

Výhodou takto definovaného terénu je jednoduchá optimalizácia vykreslením len tej časti objektu, ktorá je vo viditeľnej oblasti kamery. Takto definovaný terén sa generuje priebežne, čo znižuje nároky na video pamäť a výkon grafickej karty. Toto generovanie je v Unity automatizované [37].

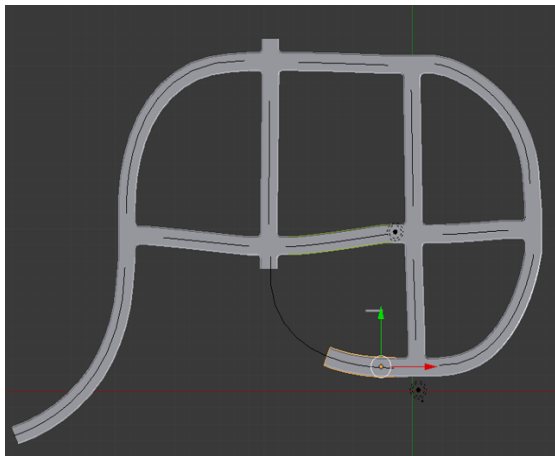
4.3.3 Cesty a križovatky

Cesty je možné tvoriť klasickým modelovaním podľa popisu z predchádzajúcich kaptiol, prípadne generovaním modelu podľa predpisu. V prípade ciest sa jedná o predpis vo forme kriviek.

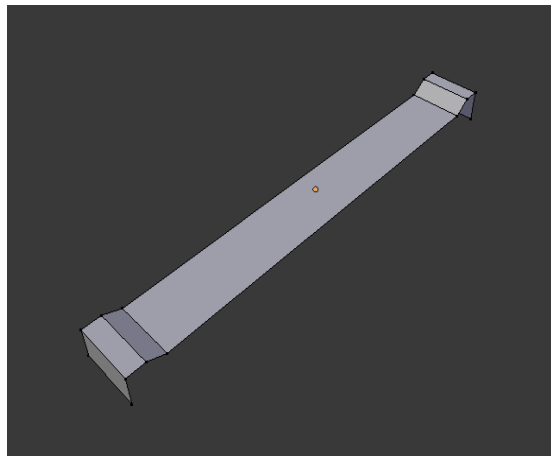
Pri tvorbe návrhovaného simulátora sú cesty a križovatky tvorené kombináciou viacerých technológií. Križovatky sú z dôvodu jednoduchšej tvorby a dosadenia jednotlivých modulov križovatiek a ciest, ktoré tieto križovatky spájajú, modelované manuálne, pomocou nástroja Blender, metódou subdivision surface modeling.

Cesty umiestnené v meste sú tvorené taktiež pomocou programu Blender, avšak modelovaný je jeden segment cesty, ktorý sa replikuje pomocou modifikátorov pozdĺž preddefinovanej krivky spájajúcej križovatky vid. (Obr. 4.9) a (Obr. 4.10).

Pre tvorbu modelov ciest umiestnených mimo mesta, teda na miestach kde je počet križovatiek minimálny, je tento prístup odlišný. Tvorba dlhých a jednoduchých úsekov ciest bez križovatiek je založená na generovaní a modifikácii objektu na základe kriviek, ktoré definujú výsledný tvar modelu.



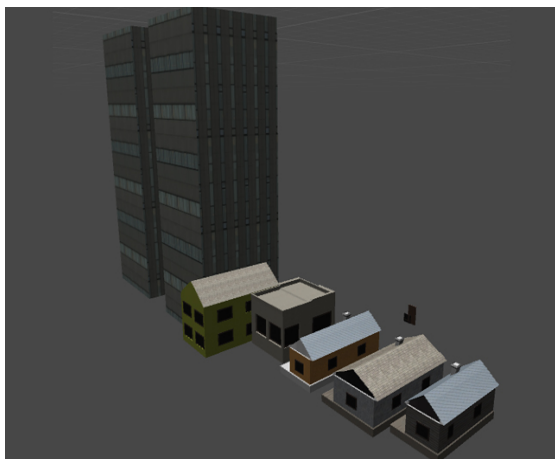
Obr. 4.9: Tvorba mestského okruhu.



Obr. 4.10: Segment cesty, ktorý sa replikuje.

4.3.4 Budovy

Samotný simulátor zahŕňa 7 modelov budov, z ktorých päť budov reprezentuje klasické rodinné domy (Obr. 4.11), využitých pri tvorbe scény mimo mestskej zástavby. Jeden model predstavuje komerčnú budovu (Obr. 4.11) a jeden model predstavuje sídliskový panelový dom (Obr. 4.12). Najzložitejším a najprepracovanejším z nich je panelový dom.



Obr. 4.11: Počítačové modely komerčných budov a rodinných domov.



Obr. 4.12: Modely panelových domov.

Tvorba panelového domu je zo všetkých budov časovo najnáročnejšia. Panelový dom, je vzhľadom k jeho rozlohe a zložitosti vytvorený z niekoľkých rôznych objektov reprezentujúcich technické segmenty akými sú brána, okná, plný panel, panel s oknom, balkón,

vchodový segment a prízemné segmenty, ktoré po zložení tvoria systém s vizuálnym charakterom podobným sídliskovým stavbám. Vzhľadom k vlastnostiam popísaného návrhu, je možné niektoré budovy rôzne kombinovať.

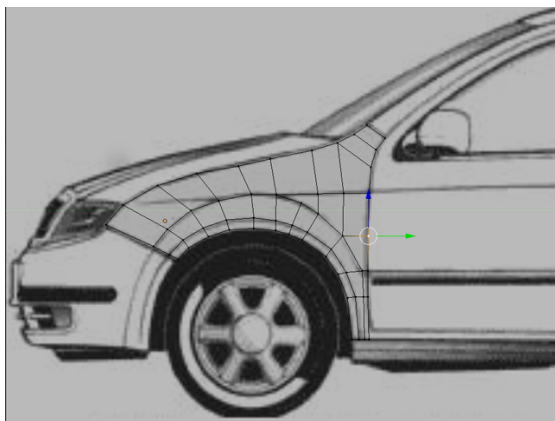
Model komerčnej budovy bol vytvorený ako klasické obdĺžnikové teleso, na ktoré je premietaná textúra imitujúca okna a povrchovú úpravu budovy. Aby mesto pôsobilo realisticky rozmanitejším dojmom, boli pre objekt komerčnej budovy pripravené viaceré textúry, s rôznou grafikou.

Zvyšné modely budov boli tvorené klasickým spôsobom modelovania metódou subdivision surface modeling, pričom vznikli ako fiktívne objekty inšpirované skutočnými budovami.

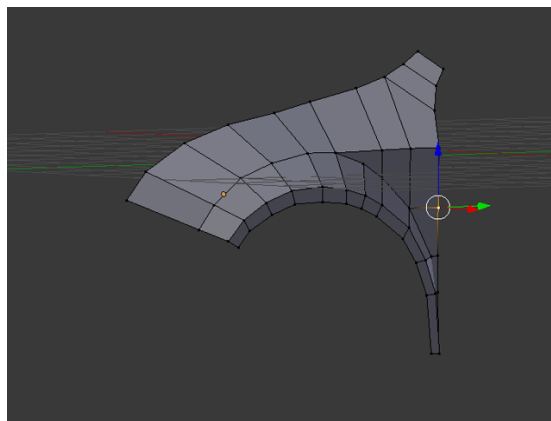
4.3.5 Priestorový model automobilu

Model exteriéru automobilu vznikol na základe skutočnej predlohy modelu Fábria od automobilky Škoda. Za účelom dosiahnutia vysokej miery podobnosti modelu so skutočnou predlohou, boli pri modelovaní jednotlivých súčastí karosérie využité technické nákresy, získané z voľne dostupných zdrojov [35]. Celkovo proces tvorby modelu pozostával z manuálneho definovania polohy vrcholov modelu s ohľadom na to, aby takto vzniknutá sieť vrcholov kopírovala povrch a hrany z nákresu predlohy. Podmienkou použiteľnosti technických nákresov bola dostupnosť bokorysu, nákresu predeje aj zadnej strany modelovaného objektu. Nákres pôdorysu vozidla nieje nevyhnutný ale pre presnejšie vyhotovenie modelu je žiadaný. Tvorba takto komplexného modelu sa rozkladá do nasledujúcich etáp:

- Príprava nákresov objektu - etapa kedy sa technické nákresy orezávajú v grafickom editore tak, aby boli po premietnutí na primitívne obdĺžnikové teleso dodržané všetky proporčné vlastnosti a hrany k sebe v priestore správne doliehali.
- Modelovanie low-poly modelu (Obr. 4.13) - etapa v ktorej dochádza k hrubej definícii tvaru objektu podľa predlohy.



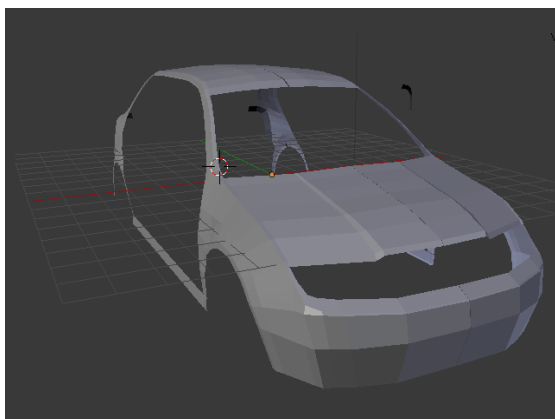
Obr. 4.13: Umiestňovanie vrcholov objektu na základe predlohy.



Obr. 4.14: Pohľad na objekt tvorený na (Obr. 4.13) v 3D priestore.

- Spresňovanie modelu (Obr. 4.15) - etapa, ktorá ma za úlohu zvýšiť počet vrcholov za účelom spresnenia a zjemnenia tvoreného objektu. V tejto etape sa vytvárajú nové hrany na vhodných miestach a spresňujú tak aproximáciu povrchu modelu.

- Aplikovanie subdivision modifikátoru (Obr. 4.16) - v tejto etape sa objekt zjemňuje do finálnej podoby. Ide o automatické dopočítavanie hrán objektu.



Obr. 4.15: Hrubá aproximácia karosérie automobilu.

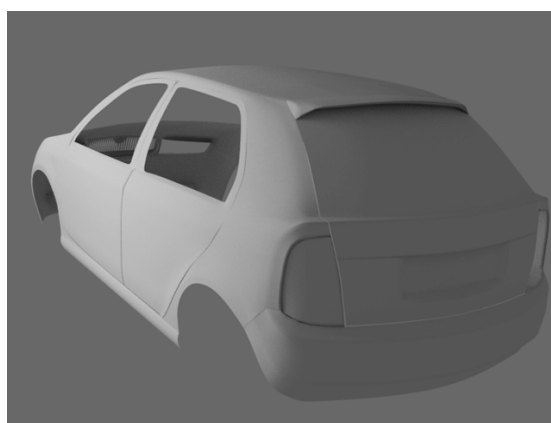


Obr. 4.16: Aplikovaný subdivision modifikátor.

- Definovanie ostrých hrán objektu - Po aplikovaní subdivision modifikátoru je výsledný model zaoblený a neobsahuje žiadné ostré hrany. Skutočné objekty, ako je napríklad karoséria automobilu, obsahujú ostré prelisy, otvory či hrany. Preto je potrebné prideliť váhy jednotlivým hranám, čím sa ovplyvní aplikovaný modifikátor a vytovria sa tak ostré hrany na zaoblenom objekte.
- Finalizácia geometrie objektu (Obrázky 4.17 a 4.18) - etapa počas ktorej sa odstránia duplicitné vrcholy. V prípade, ak by nedošlo k odstráneniu duplicitných vrcholov, vznikali by chyby pri aplikovaní osvetlenia. Počas tejto etapy sa kontroluje orientácia jednotlivých plôch objektu. V prípade nezrovnalostí musí dôjsť k predefinovaniu normály tejto plochy, aby objekt nepôsobil vo finálnej scéne ako neuzavretý.

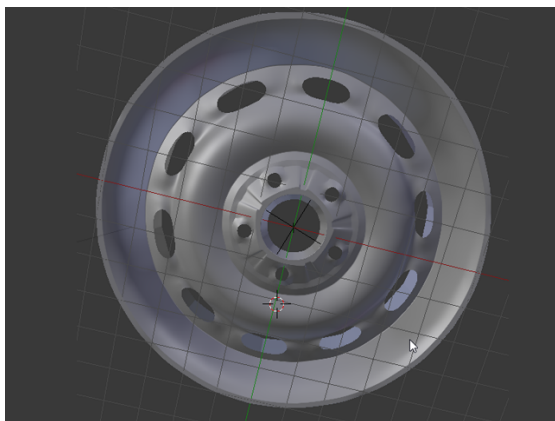


Obr. 4.17: Finálna geometria karosérie vozidla.

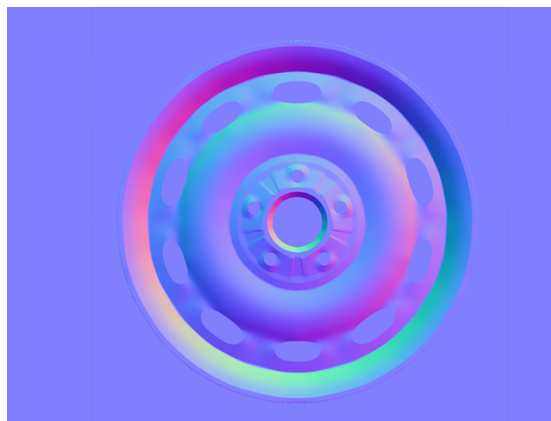


Obr. 4.18: Finálna geometria zadnej časti karosérie vozidla.

- Optimalizácia objektu - v tejto etape sa vytvárajú normálové mapy (Obr. 4.20), ktoré modifikujú osvetľovací model objektu spôsobom, pri ktorom sa rovinný objekt s aplikovanou normálovou mapou javí ako priestorový. Touto cestou je následne možné znížiť množstvo vrcholov objektu, ktoré budú nahradené normálovou mapou.



Obr. 4.19: Detailný 3D objekt.



Obr. 4.20: Normálová mapa získaná z objektu (Obr. 4.19).

- Pridelovanie materiálov (Obrázky 4.21 a 4.22) - posledná etapa, v ktorej sa na jednotlivé časti vytvoreného objektu priradí materiál, ktorý ovplyvňuje svetelné vlastnosti danej oblasti.



Obr. 4.21: Priradené materiály v exteriéri.



Obr. 4.22: Priradené materiály v interiéri.

K exteriéru vozidla patria modely kolies, svetiel či interiériu, ktoré boli modelované podobným spôsobom ako exteriér a ich vývoj prechádzal hore uvedenými etapami.

Interiér vozidla je rozvrhnutý na 2 časti:

- Detailná časť potrebná k správnej funkčnosti simulácie - jedná sa o detailne spracovanú oblasť interiériu, ktorá pozostáva z viacerých detailných objektov. V tomto

prípade sa jedná o palubné prístroje, ukázatele a volant. Tieto objekty museli byť navrhnuté tak, aby ich pozície a rotácie bolo možné programovo meniť, a tieto pohyby pôsobili realisticky.

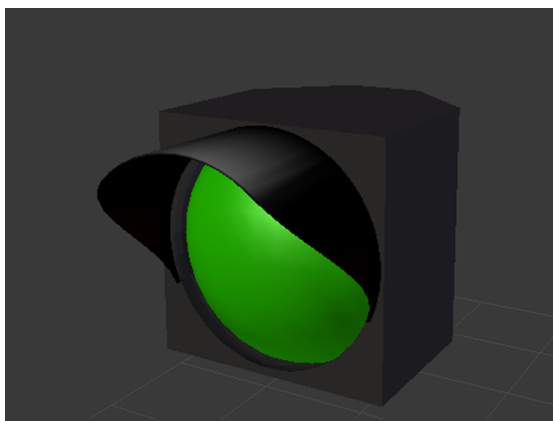
- Časť, ktorá nemá žiadnu funkcionality - časť interiéru ktorá je vytvorená iba za účelom vizuálneho kontextu. Túto časť nebolo nutné detailne spracovávať, keďže užívatelia simulácie sa na túto časť objektu nebudú zameriavať vôbec, alebo iba minimálne. To vytvára priestor pre možnú optimalizáciu a využitie textúr miesto modelovania priestorových detailov.

Tvorba zložitých objektov je veľmi časovo náročná a zaberá značnú časť tvorby simulátora, z tohto dôvodu bol vytvorený iba jeden model vozidla. Prípadné ďalšie modely vozidiel alebo ich častí, boli prevzaté z voľne dostupných zdrojov [4].

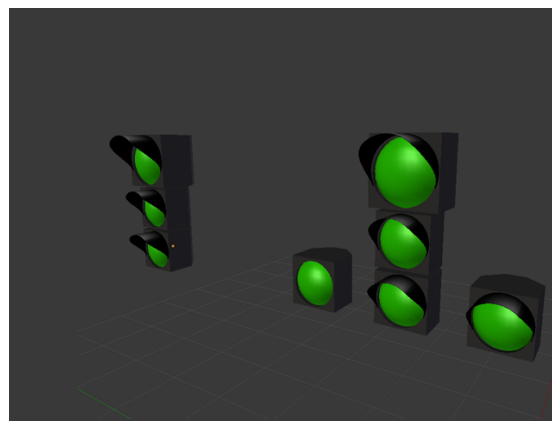
4.3.6 Dopravné značenie

V simulátore existujú 2 druhy dopravného značenia - aktívne svetelné a pasívne zvislé.

Svetelné značenie je navrhnuté ako modulárny systém jednotlivých svetelných elementov (Obr. 4.24). Samotný semafor svetelnej križovatky sa do finálnej podoby dostane spojením troch svetelných modulov. K tomu dochádza až pri konštruovaní scény v editore Unity3D. Svetelný modul, vyobrazený na obrázku 4.23, bol navrhnutý tak, aby sa dal použiť pre ktorékoľvek svetelné zariadenie a aby bolo možné zmeniť jeho vizuálne farebné, svetelné a signalizačné vlastnosti jednoduchou zámennou aktuálneho materiálu za iný. Modul pozostáva z dvoch objektov a to signalizačnej časti, a vizuálneho obalu. Zámennou materiálu signalizačnej časti je možné jednoducho zmeniť význam modulu voči pravidlám cestnej premávky.

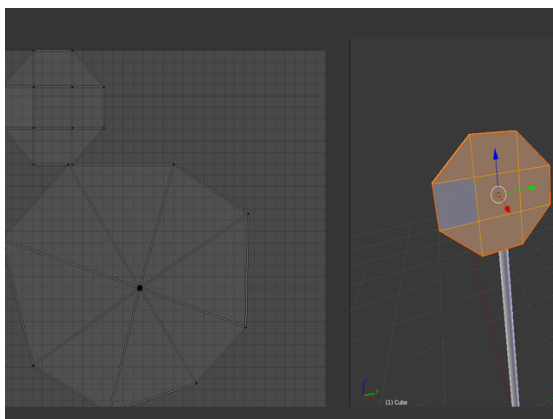


Obr. 4.23: Svetelný modul.



Obr. 4.24: Svetelné značenia z modulov.

Klasické zvislé značenie pozostáva zo šiestich rôznych modelov. Každý model reprezentuje inú významovú kategóriu dopravných značiek, pričom konečný význam danej značky dodá až textúra obsahujúca konkrétne symboly. Tento prístup vyžaduje textúry rovnakej veľkosti, a modely so správnym nastavením UV máp, pomocou ktorých je možné správne premietnúť textúru na objekt. Proces UV mapovania je zachytený na obrázku 4.25. Výsledne premietnutie 2D textúry na 3D objekt je možné vidieť na obrázku 4.26.



Obr. 4.25: Mapovanie povrchu zložitého objektu na 2D plochu.



Obr. 4.26: Aplikovanie textúry na nama-povaný povrch objektu.

Kapitola 5

Implementácia

V tejto kapitole bude popísaná samotná implementácia funkčných častí simulátoru a zostavenie výslednej scény.

5.1 Fyzikálny model

Na to aby vozidlo mohlo v systéme fungovať, musí byť na samotný model vozidla aplikovaný komponent tuhého telesa fyzikálneho systému, ktorému sa nastaví jeho fyzikálne vlastnosti podľa skutočných údajov [1], vrátane hmotnosti objektu (v prípade bežných vozidiel je to priemerne 1,4 tony tj. 1400kg, závisí od modelu) a tiež kolízne telesá, ktoré sú nevyhnutné k tomu, aby vozidlo reagovalo na ďalšie objekty v scéne. Niektoré kolízne telesá sa používajú ako prvky simulačnej logiky a slúžia pre rôzne typy detekcií prechodu vozidla sekciami scén či spínačov alebo zakázaných zón. Kolízne telesá sú nevyhnutnou súčasťou kostry modelu automobilu vďaka ktorým tento systém funguje. V prípade projektu simulátoru autoškoly sú použité 3 rôzne druhy kolíznych telies ktoré sú podporované v Unity (klasický box collider, wheel collider, mesh collider).

Kolízne teleso obdĺžnikovitého tvaru (Box collider), je definované ôsmimi vrcholmi. Tento typ kolízneho telesa sa využíva predovšetkým ako objekt logiky vďaka jeho nízkemu nároku na výpočtový výkon stroja. Jeho existencia v scéne znamená nižšiu cenu za výpočet v porovnaní s Mesh Colliderom. V kontexte s vozidlom je tento typ kolízneho telesa spájaný s využitím ako detektor vjazdu / výjazdu z definovanej zóny, prípadne funkcia spínača alebo budiča akcií v scéne, prípadne reakcií iných objektov na prítomnosť vozidla v ich okolí.

Kolízne teleso kolesa (Wheel collider), je špeciálny typ kolízneho telesa, ktorý simuluje koleso a základnú implementáciu odpruženia. Je to nositeľ informácií o kontaktnom bode kolesa s povrchom a získava informácie o priečných a pozdĺžnych silách pôsobiacich na koleso vozidla. Ďalšou jeho funkciou je aplikácia krútiaceho momentu na konkrétne koleso, ktorú vykonáva vzhľadom k aktuálnemu treniu kolesa s vozovkou. Unity zatiaľ neumožňuje v prípade kolíznych telies aplikovať sily na okolie vzhľadom k fyzikálnym vlastnostiam materiálu okolitých telies, preto je potrebné povrch simulovať prepočtami a následnou úpravou koeficientov trenia na prepočítané hodnoty, a to všetko za behu programu. Tento prístup umožní do istej miery simulovať priebeh prílnavosti kolies k podkladu a tým ovplyvniť správanie vozidla.

Kolízne teleso definované sieťou vrcholov a trojúholníkov (Mesh collider), je kolízne teleso používané hlavne pre opis zložitých tvarov alebo karosérie vozidla. Mesh collider na základe predpisu vo formáte modelu vytvorí konvexné alebo nekonvexné kolízne teleso

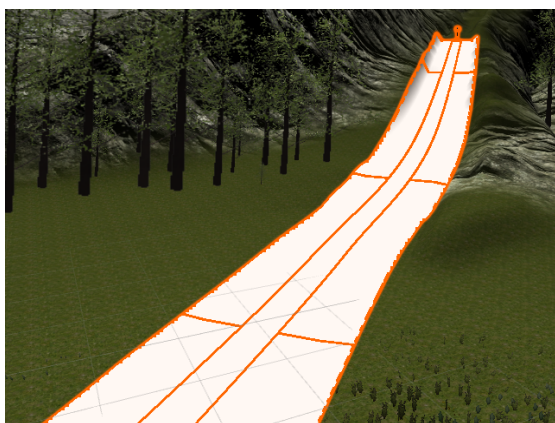
spĺňujúce tento predpis. Problémom takto definovaných kolíznych telies však je náročnosť na výpočet a kontrolu kolízií. Z uvedeného plynie vhodnosť obmedzeného využitia tohto typu kolízneho telesa na nutné minimum. Výhodou tohto typu kolízneho telesa je fakt, že ho je možné vďaka charakteru popisu programovo vygenerovať.

Samotný fyzikálny model dynamiky vozidla pozostáva zo štyroch komponent a to Engine.cs, Transmission.cs, Stabilizer.cs, Differential.cs. Tieto skripty sú vytvorené na základe kapitoly č.4. Samotná aplikácia skriptov fyzikálneho modelu sa vykoná priradením týchto skriptov objektu, ktorému bol predtým priradený komponent tuhého telesa.

5.2 Scéna simulácie

Tvorba prostredia ako finálneho celku sa rozdeľuje do štyroch etáp. V prvej etape sa implementuje a tvorí samotný podklad pomocou metód popísaných vo štvrtnej kapitole. Editor prostredia Unity poskytuje nástroje pre úpravu výškových máp v reálnom čase. Pomocou tohto nástroja je vytváraná výšková mapa terénu s aktuálnym náhľadom na priestorový model zvolenej oblasti. Vďaka uvedenému nástroju sa celý proces prvej etapy urýchlil.

Druhou etapou tvorby prostredia bola vzhľadom k charakteru projektu, tvorba cestných komunikácií. Samotné princípy a prístupy tvorby modelu ciest boli popísané vo štvrtnej kapitole. Mestský okruh vrátane križovatiek je z pohľadu implementácie prostredia jeden model, na ktorý budú naväzovať generované úseky ciest. Generované úseky ciest spájajú vjazd a výjazd z mesta a vedú skrz vytvorenú krajinu. Na týchto generovaných úsekoch je testovaná najmä zručnosť a správna úvaha vodiča. Pre generovanie ciest je využitý plugin EasyRoads3D pre prostredie Unity, ktorý zabezpečuje správne generovanie cesty podľa preddefinovanej krivky a jej jednotlivých bodov (Obr. 5.1 a obr. 5.2). Táto etapa so sebou prináša jeden z najväčších problémov tvorby prostredia a to nutnosť úpravy terénu tak, aby model cesty pôsobil, že je súčasťou terénu aj keď sa jedná o 2 rôzne objekty. Riešením je pracná manuálna úprava okolia cesty, pričom problémy vznikajú s nízkym rozlíšením výškovkej mapy. V dôsledku tejto vlastnosti nieje možné úplne dokonale terén upraviť. Celkovo je v projekte približne 11 kilometrov cestných komunikácií z čoho väčšina leží mimo obcí.



Obr. 5.1: Tvorba ciest pomocou EasyRoads3D.



Obr. 5.2: Výsledný model cesty vytvorený pomocou pluginu EasyRoads3D.

Tretia etapa pozostáva z umiestnenia stavieb a dopravných zariadení. Stavby sú v meste umiestňované tak, aby vytvárali miernú dezorientáciu pre užívateľa a súčasne umožňovali vizuálny klam väčšieho priestoru. Jednotlivé objekty sa do scény pridávajú jednoduchým pretiahnutím modelu z prieskumníka a následným nastavením mierky a pozície objektu v rámci scény. Obdobne sa rozmiestňujú dopravné značky a infraštruktúra popri komunikácii.

V záverečnej etape sa dopĺňa terén o detaily vo forme textúr a modely rastlín. Textúrovanie terénu bolo možné realizovať pomocou Unity editoru. Pri textúrovaní dochádza k prekryvaniu viacerých textúr a ich normálových máp v jednej oblasti tak, aby nebolo možné voľným okom spozorovať miesta s výskytom opakujúcej sa textúry. Textúry pre tento účel musia spĺňať štandard tzv. seamless textúry. Ide o textúry ktorých protiahlé okraje na seba nadväzujú tak, že po priložení dvoch ištancií takejto textúry nevzniká viditeľný prechod alebo predel a vizuálne tieto pôsobia ako jeden celok. Ukážka tohto typu textúry je na obrázku 5.3. Stromy vznikli v editore Unity a ich tvorba spočívala v zhrubovaní kriviek definujúcich kmeň, či vetvy stromu. Stromy sú generované ako objekty na základe predpisu formou kriviek. Rastliny a trávy sú implementované ako tzv. billboardy. Billboardy sú primitívne plochy s premietnutou textúrou rastliny vrátane jej priesvitnosti tj. alpha kanálu farby. Príklad textúry rastliny možno vidieť na obrázku 5.4. Kľúčová vlastnosť billboardov je ich rotácia okolo lokálnej osi Y, tak, aby normalový vektor plochy billboardu smeroval k pozícii kamery. To spôsobuje vizuálny klam priestorového vyhotovenia stebiel trávy a rastlín.



Obr. 5.3: Príklad seamless textúry [33].



Obr. 5.4: Príklad textúry billboardu flóry [34].

5.3 Automobil

Implementácia automobilu začína nainportovaním potrebných modelov samotnej karosérie a interiéru vozidla, kolesa a volantu. V stromovej štruktúre scény je potrebné umiestniť automobil ako celok vrátane všetkých prídavných modelov pod spoločný uzol (ďalej iba rodičovský uzol) stromu, aby tieto v scéne figurovali ako jeden objekt. Následne sa v rodičovskom uzle vytvoria 4 uzly reprezentujúce:

- vizuálne časti vozidla,
- logické objekty vozidla,
- vizuálne modely kolies,
- wheel colliderov kolies.

Po zostavení podstromu časti scény podľa predchádzajúceho popisu, sa rodičovskému uzlu vozidla priradia komponenty mesh-collider a rigid-body. Pre rigid-body sa nastavujú vhodné hodnoty hmotnosti spolu s koeficientom odporu vzduchu. Následne bolo potrebné nastaviť vhodnú orientáciu a pozíciu každého kolesa zvlášť. Orientácia by mala byť nastavená tak, aby lokálna os Y objektu kolesa smerovala smerom hore od povrchu terénu, lokálna os Z bola nasmerovaná v smere jazdy vozidla a lokálna os X smerovala od strán vozidla. V prípade, že takéto nastavenie orientácie nieje v súlade s vizuálnou stránkou modelu, je potrebné pre daný model vytvoriť rodičovský uzol (bude sa jednať o tzv. vnorovanie úzlov) s vyššie deklárouanou orientáciou a následne orientáciu samotného modelu v rámci lokálneho súradného systému jeho rodičovského uzlu upraviť tak, aby bol dosiahnutý požadovaný vizuálny stav. V prípade úprav vnorovaním úzlov sa bude, z hľadiska skriptovania, brať za objekt najvyšší rodičovský uzol vzniknutý postupným vnorovaním pôvodného objektu.

Požadované orientácie vychádzajú z implementácie skriptu, ktorý interpretuje transformácie a dáta z kolíznych telies kolies existujúcich vo fyzikálnom systéme, na vykresľované objekty kolies. Za účelom vizuálnej interpretácie transformácii kolíznych telies kolies a odpruženia podvozku vznikol skript `wheelTransformAnimation.cs`, ktorý tieto prepočty implementuje a aplikuje na požadované objekty. Skript `wheelTransformAnimation.cs` sa priradzuje každému z wheel colliderov a jeho parametrom je odkaz na vizuálny model kolesa, na ktorý sa majú výstupné hodnoty skriptu aplikovať.

Implementácia vozidla pokračuje importovaním modelu volantu a palubnej dosky, vrátane 3D ukázačkov (Obr. 5.5 a 5.6). Volant a ručičky ukázačkov prevádzkových hodnôt sa opäť podľa vzoru kolies upravujú tak, aby tento krát os ich rotácie bola reprezentovaná lokálnou osou Z. To znamená, že os rotácie ručičiek ukázačkov bude smerovať smerom do palubnej dosky, tak ako je to na obrázku 5.5. Interpretáciu rotácie volantu má na starosti skript, ktorý spracováva vstupné zariadenia.

Mechanizmus ukazateľov a interpretácie digitálnych dát na ručičkové ukazatele je naimplementovaný pomocou skriptu Dashboard.cs. Skript Dashboard.cs využíva ku konverzii dát interpoláciu medzi dvoma úhlami na základe rozsahu ukazateľa a aktuálnej hodnoty dát, ktoré sa majú zobraziť.



Obr. 5.5: Správna orientácia volantu v automobile.



Obr. 5.6: Nastavenie východziech polôh ukazateľov.

Simuláciu smerových svetiel a uchovávanie ich stavu má na starosti skript turnSignals.cs, ktorého vstupom je udalosť klávesnice a výstupom je nastavenie a zmena materiálov indikátora na palubnej doske, prípadne objektov bodového svetla umiestnených v exteriéri vozidla na základe určitého časového intervalu.

Následne sa na vozidlo aplikujú materiály s rôznymi svetelnými vlastnosťami, čím je vizuálna časť implementácie vozidla hotová (Obr. 5.7).

Simulácia pohľadu vodiča je z hľadiska pocitu zo simulácie kľúčová, preto je potrebné nastaviť pohľad kamery tak, aby čo najviac korešpondoval so skutočným pohľadom vodiča vozidla. Implementácia pohľadu preto pozostáva z umiestnenia prázdneho objektu do lokálneho priestoru interiéru vozidla, pozícia ktorého bude referenčnou pre aktualizáciu pozície kamery. Problémom zostáva skutočnosť, že takáto jednoduchá implementácia pohľadu žiadnym spôsobom nezohľadní prudkú akceleráciu, či brzdenie a pôsobí tak veľmi umelo a nereálne. Preto je aj do tejto oblasti implementácie zakomponovaný fyzikálny systém. Implementácia v tomto prípade zahŕňala umiestnenie a nastavenie referenčného objektu kamery, ktorý je upevnený na objekt vozidla pomocou fyzikálneho kĺbu typu Hinge joint. Referenčný objekt pre kameru sa pridaním rigid-body komponentu stáva súčasťou fyzikálnej simulácie a správnym nastavením limitov fyzikálneho kĺbu je možné dosiahnuť simuláciu pohybu hlavy podľa miery zrýchlenia či brzdienia.

Predposledný krok implementácie je zameraný na pridávanie logických objektov do vozidla, nutných z hľadiska funkčnosti simulácie (Obr. 5.8). Jedná sa o kolízne telesá fungujúce v móde spínača, ktoré sú aktivované po kontakte s inými kolíznymi telesami na rovnakej fyzikálnej vrstve. Hlavným skriptom je skript carRuleCheckBolton.cs ktorý zbiera dáta o dopravných značkách umiestnených popri ceste.

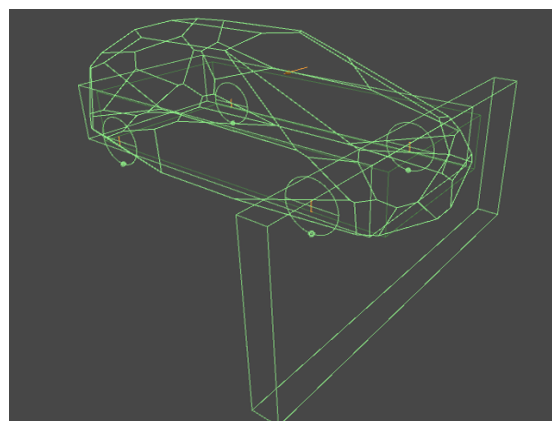
Na záver ostáva už iba priradenie jednotlivých skriptov rodičovskému uzlu celého vozidla a nastavenie parametrov či referencií jednotlivých skriptov. Na vozidlo sa pridávajú skripty simulujúce jeho vlastnosti a javy v ňom prebiehajúce. Sústava týchto skriptov tvorí komplexný systém vozidla, ktorý je hlavným predmetom simulácie.

Hlavný uzol novo vzniknutého systému vozidla obsahuje tieto hlavné skripty:

- userControl.cs - spracovanie vstupných ovládacích prvkov a následná reakcia na ich události,
- carCamera.cs - simulácia pohľadu vozidla,
- dashboard.cs - interpretácia prevádzkových dát na transformácie objektov ukazateľov,
- 2x stabilizer.cs (pre každú nápravu jeden) - stabilizátor nápravy vyrovnáva silové napätia na oboch stranách nápravy tak, aby vozidlo bolo stabilnejšie a neprevrácalo sa v zákrutách,
- engine.cs - matematický model motoru vozidla,
- transmission.cs - matematický model prevodovky vozidla, využívajúci prevodové pomery skutočného vozidla [13].,
- differential.cs - prerozdelenie krútiaceho momentu medzi kolesa nápravy,
- turnSignals.cs - správa smerových svetiel vozidla.



Obr. 5.7: Vozidlo v jeho finálnej podobe.



Obr. 5.8: Logické objekty vozidla.

5.4 Implementácia pravidiel a dopravných značení

Implementácia sledovania dodržiavania pravidiel je založená na detekcii kolízií logických objektov. Pravidlá formou skriptov definujú rôzne sekvencie krokov, ktoré musia byť dodržané na to, aby bolo pravidlo splnené. V prípade porušenia tejto sekvencie je pravidlo vyhodnotené ako nesplnené, a užívateľ simulácie je upovedomený o priestupku.

V prípade tohto simulátoru autoškoly sú logické objekty poskytujúce informácie o dopravnom pravidle (ďalej iba info feed), a objekty ktoré tieto info feedy čítajú (ďalej iba feed receiver) priradzované do vlastných fyzikálnych vrstiev. Objekt typu feed receiver je umiestňovaný na vozidlo a počas prejazdu rôznymi info feedmi, aktualizuje niektoré údaje akými sú maximálna povolená rýchlosť alebo informácie o následujúcej križovatke, ktoré využíva počítačom riadená premávka.

V prípade implementácie pravidiel cestnej premávky nič ako univerzálne pravidlo nieje možné implementovať. Každá dopravná značka skúma rôzne okolnosti za ktorých je platná či neplatná. Preto je každá dopravná značka implementovaná vlastným skriptom a rôznym počtom logických objektov.

Kontrola rýchlostí vozidla je vykonávaná na strane vozidla, kde sa porovnáva aktuálna rýchlosť s maximálnou povolenou rýchlosťou v oblasti do ktorej ako poslednej vozidlo vstúpilo. Táto oblasť je ohraničená info feedmi, ktoré držia informáciu o rýchlostnom limite.

Kontrola značky "Stoj, daj prednosť v jazde!" pozostáva z kontroly troch logických objektov, kde dva z nich detekujú vjazd a výjazd z kontrolovanej zóny a tretí definuje oblasť v ktorej musí vozidlo zastaviť (Obr 5.12). Pri tomto pravidle je potrebné zohľadniť smer, ktorým vozidlo ide, nakoľko táto značka platí iba v jednom smere. Smer príjazdu vozidla je možné vyvodiť z poradia aktivácie jednotlivých logických objektov.

Implementácia kontroly predností v jazde v križovatke je o niečo zložitejšia. Jednou z možností bolo využitie metódy ray-casting a takto kontrolovať okolie vozidla, čo by ale bolo neefektívne a nepresné. Preto je táto kontrola implementovaná pomocou troch logických objektov pre každý z jazdných pruhov. Dva logické objekty reprezentujúce hranu jazdného pruhu ohraničujú logický objekt definujúci oblasť križovatky v rámci jazdného pruhu. Logika kontroly počíta počet vozidiel ktoré vošli do križovatky. Ak simulované vozidlo vojde v križovatke do protismeru jazdy v dobe keď je počet oproti idúcich vozidiel v križovatke väčší ako 0, potom sa simulované vozidlo dopustilo ohrozenia iných účastníkov premávky (tj. nedalo prednosť v jazde). Logická sústava kontrolných objektov je zachytená na obrázku 5.10.

Kontrola prejazdu svetelnou križovatkou musí navyše brať do úvahy aktuálny stav svetelných semaforov, a kontrolovať vjazd do križovatky na signál "Stoj". Implementácii samotného pravidla musí logicky predchádzať implementácia samotných svetelných semaforov.

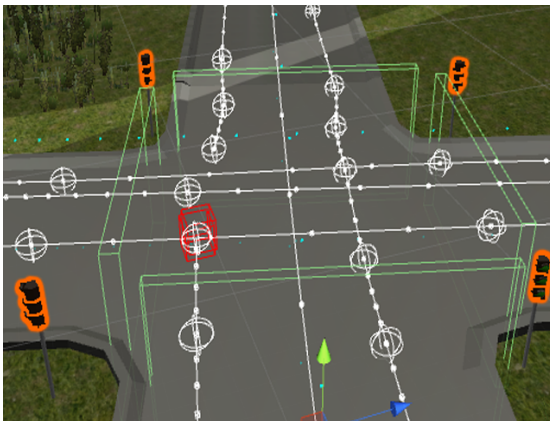
Svetelný semafor je v editore zložený z troch svetelných modulov, ktorých tvorba bola popísaná v štvrtej kapitole. Každému zo svetelných modulov semaforu je priradený skript semaphoreLight.cs ktorý definuje farbu a význam jeho svetelnému modulu. Správanie semaforu je definované pomocou konečného automatu naimplementovaného v skripte Semaphore.cs, v ktorom sa dá nastaviť napríklad preblikávanie signálu "Voľno" pred zmenou na signál "Pozor" a podobne.

Svetelný semafor je následne vložený do scény. Semaforu je ďalej pridelený práve jeden logický objekt, ktorý funguje ako neviditeľná stena umiestnená na hrane križovatky. Logický objekt semaforu má priradený skript redLightRule.cs, ktorý aktivuje kontrolu prejazdu vozidla logickým objektom. Simulované vozidlo má naimplementované počítadlo prejazdov cez hranu križovatky. Pri vjazde vozidla do križovatky sa hodnota počítadla inkrementuje a pri

opustení križovatky následne dekrementuje, prípadne nastaví na nulu. Na základe hodnoty počítadla je možné určiť, či pri prejazde logickým objektom svetelného semafora vozidlo vchádza do križovatky, alebo naopak, z nej vychádza. V prípade že je hodnota počítadla pri vjazde do logického objektu semaforu nulová a aktuálny stav semaforu znamená signál "Stoj!", je tento vjazd do križovatky vyhodnotený ako prejazd na červenú. V prípade signálu voľno, platia vo vnútri križovatky klasické pravidlá cestnej premávky. Rozmiestnenie logických objektov je možné vidieť na obrázku 5.9.

Skupina svetelných semaforov vytvára organizovaný systém semaforov, ktorý je implementovaný opäť formou konečného automatu s pomocou časovača, kde sú jednotlivé stavy dynamicky meniteľné. Celý semaforový systém je parametrizovateľný, vrátane jednotlivých konečných stavov a preto je možné simulovať rôzne druhy križovatiek. Systém semaforov predstavuje uzol v ktorom sú umiestnené všetky semafory spadajúce pod spoločný systém. Implementácia takéhoto systému je napísaná v skripte `semaphoreSystem.cs`

Množina systémov semaforov sa nazýva centrálny semaforový systém, a ten je implementovaný v skripte `centralSemaphoreSystem.cs`. Centrálny systém semaforov má za úlohu udržiavať nadefinované poradie v ktorom sa budú meniť stavy svetelných križovatiek. Pomocou tohto systému je možné simulovať zelené vlny.



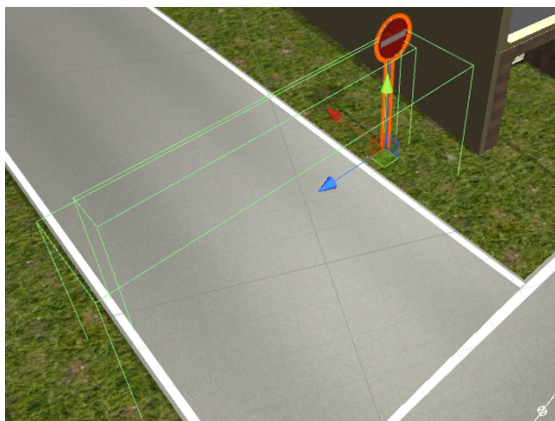
Obr. 5.9: Križovatka riadená svetelnými semaforami.



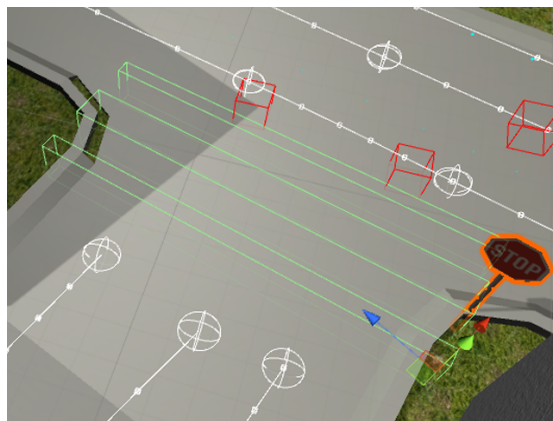
Obr. 5.10: Kontrola prednosti v klasickej križovatke.

Pravidlo zákazu vjazdu sa odlišuje od vyššie uvedených prístupov v tom, že je implementované dvoma logickými objektami, ktoré sa vzájomne presahujú, viď obrázok 5.11. V prípade, že pri vjazde vozidla do kontrolovanej oblasti bol aktivovaný logický objekt definujúci rozsah platnosti pravidla, ide o porušenie zákazu vjazdu. Táto implementácia sa líši od ostatných kvôli charakteru miest kam sa značka zákazu vjazdu umiestňuje. Jedná sa väčšinou o jednosmerné ulice, a pri príjazde k značke od jej zadnej strany (tj. v správnom smere jazdy) sa nejedná o porušenie predpisu zákazu vjazdu, detekcia porušenia pravidla by tak bola chybná.

V tejto podkapitole bola popísaná implementácia troch problémových a špecifických spôsobov detekcie dopravných priestupkov. Ostatné pravidlá sú implementované obdobným spôsobom. Pri implementácii pravidiel ide predovšetkým o definovanie oblasti platnosti, a detekciu vjazdu vozidla, avšak nie je možné určiť univerzálne vzájomné polohy kontrolných objektov ani jednotný spôsob ako tieto kontroly implementovať.



Obr. 5.11: Implementácia pravidla "Zákaz vjazdu všetkým vozidlám".



Obr. 5.12: Implementácia pravidla "Stoj! Daj prednosť v jazde".

5.5 Reprezentácia pohybu účastníkov premávky

Účastníkom premávky sa z pohľadu simulácie rozumie automobil riadený počítačom. Zostavenie takéhoto automobilu v scéne je obdobné ako pri implementácii simulovaného automobilu. Na rozdiel od simulovaného automobilu, však automobil riadený počítačom nepotrebuje k svojej činnosti skripty popisujúce dynamiku vozidla. Tá je v tomto prípade iba aproximovaná. Implementácia účastníka premávky spočíva vo vytvorení autonómneho správania vozidla, pričom musí reagovať na svoje okolie. Na autonómneho účastníka premávky sa vzťahujú rovnaké pravidlá cestnej premávky ako na simulované vozidlo, čo bolo nutné v popise jeho správania zohľadniť.

Implementácia využíva prístup sledovania na seba naväzujúcich bodov trasy (takzvaných waypointov). Manuálne rozmiestňvanie bodov by bolo z hľadiska časovej náročnosti a presnosti takto vytvorenej trasy nevhodné. Preto sú tieto body generované na základe predpisu formou krivky. V tomto prípade sa jedná o Catmull-Rom interpolačnú krivku [31]. Pomocou tejto krivky je určená približná trajektória počítačom riadených vozidiel. Pozdĺž nadefinovanej krivky sa v nastavenom intervale generujú waypointy (Obr. 5.13 a obr. 5.14).

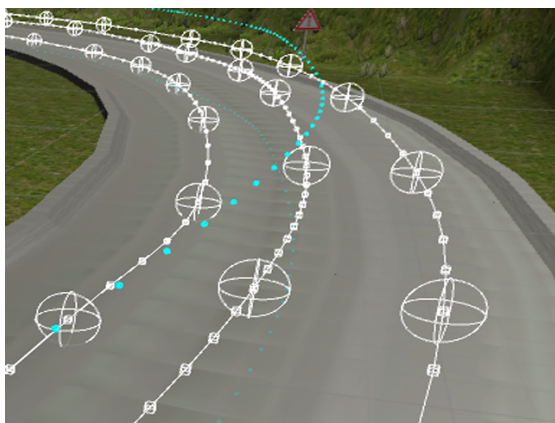
Pre každý jazdný pruh je potrebné mať nadefinovanú samostatnú krivku. Každý waypoint v sebe ukrýva datovú štruktúru, ktorá udržiava informáciu o nasledujúcom bode trasy. Celá trasa je teda implementovaná ako jednosmerný zoznam.

V prípade križovatky, má waypoint umiestnený pred jej hranou priradenú, okrem spomenutej datovej štruktúry, aj ďalšiu datovú štruktúru, ktorá plní dva účely. Prvým účelom je indikácia blížiacej sa križovatky. Túto informáciu hlavný skript vozidla zúžitkuje tak, že upraví rýchlosť a správanie vozidla, ktoré riadi. Druhým účelom je poskytnutie informácie o topológii nasledujúcej križovatky. Sústava boolových datových typov určuje ako vyzerá križovatka, a ktoré výjazdy sú dostupné. Pre každý možný budúci smer vozidla je uchovávaná referencia na zoznam waypointov daného smeru a referencia na nasledujúci waypoint, ktorým by sa vozidlo v prípade výberu daného smeru riadilo.

Riadiaci skript počítačom riadeného vozidla uchováva informáciu o aktuálnom a nasledujúcom bode trasy. Pri dosiahnutí určitého okolia nasledujúceho bodu, sa nasledujúci bod zmení na aktuálny, a ďalší bod sa opäť načíta z datových štruktúr už aktuálneho bodu trasy.

Samotné riadenie vozidla je tak implementáciou pravidelného prepočtu uhlu medzi smerovým vektorom pohybu vozidla, a pozíciou nasledujúceho bodu trasy. Na základe získaných výpočtov sa nastaví uhol vytočenia predných kolies a tým sa zmení smer jazdy vozidla.

Prispôsobenie jazdy dopravným predpisom je dosiahnuté obdobným spôsobom info-feedov ako v prípade užívateľom riadeného vozidla, avšak s tým, že pre vozidla riadené počítačom sú vytvárané osobitné info-feedy, ktoré presne definujú hodnoty na základe ktorých sa zmení správanie vozidla. Počítačom riadené vozidlá parvidelne monitorujú svoje okolie pomocou "ray-castov" a získavajú tak informácie o okolitých objektoch, svetelných semaforoch, či užívateľom riadenom vozidle. Hlavnou komponentou je skript AICar2.cs v ktorom je uložená implementácia popísaná v tejto podkapitole.



Obr. 5.13: Vizualizácia bodov trasy.



Obr. 5.14: Body trasy lemujú celú cestnú sieť.

5.6 Interakcia medzi prvkami simulácie

Interakcia v simulátore autoškoly pozostáva zo vzájomných reakcií vozidiel. Interakcia zo strany vozidla ovládaného užívateľom je logický úplne závislá na samotnom užívateľovi. Pre počítačom riadené vozidlá je interakcia implementovaná dvomi spôsobmi. Istú časť interakcie riadi skript počítačom riadeného vozidla. Zvyšok je riadený logickými objektami umiestnenými v scéne.

Počítačom riadené vozidlo obsahuje logický objekt, ktorý slúži ako zdroj pre ray-castové lúče. Táto implementácia bola inšpirovaná známym systémom LIDAR, ktorý sa využíva v skutočných autonómnych vozidlách. V tomto prípade ale metódou "ray-castingu" skript nezískava úplnú podobu okolia vozidla, ale vyhľadá najbližšie vozidlo v smere jazdy a kontroluje vzdialenosť medzi riadeným vozidlom a detekovaným vozidlom. Na základe získanej vzdialenosti následne riadiaci skript reguluje rýchlosť vozidla ktoré ovláda.

Interakcia vozidiel v križovatkách je implementovaná pomocou skriptov, ktoré nastavujú vlastnosti logických objektov umiestnených na hrane križovatky. Vlastnosti popisujúce stav a prechodnosť križovatky následne počítačom riadené vozidlá načítavajú z logických objektov, ktoré sú nositeľmi týchto informácií. Potom na základe programu rozhodnú, ako budú danú situáciu riešiť či ostanú stáť pred hranou križovatky, alebo križovatkou prejdú.

Pre skvalitnenie interakcií je na každom počítačom riadenom vozidle niekoľko vhodne umiestnených kolíznych telies v roli logických objektov, ktoré napomáhajú sledovať blízke okolie vozidla, prípadne odhaľovať objekty, ktoré metóda ray-castingu nezaznamenala. V prí-

pade aktivácie niektorého z týchto colliderov riadiaci skript aplikuje núdzové správanie a vozidlo zastaví.

Časť implementácie interakcie počítačom riadených vozidiel s dopravným značením bola popísaná v predchádzajúcich kapitolách.

Kapitola 6

Testovanie

Testovanie má za úlohu vyhodnotiť kladné a záporné vlastnosti vytvoreného simulátoru.

6.1 Metodika testovania

Výsledná aplikácia bola testovaná na vzorke dvadsiatich respondentov, ktorým bola aplikácia predvedená a následne im bolo umožnené si aplikáciu vyskúšať. Po absolvovaní testov vyplnil každý z testerov odpovede na otázky v dotazníku a absolvoval voľnú konverzáciu, kde bol dotazovaný na jeho pocity a postrehy z užívania vytvorenej aplikácie.

Dotazník bol rozdelený na 2 časti, časť zameranú na grafické spracovanie a časť zameranú na funkčné aspekty simulácie. V dotazníku boli dva druhy otázok, otázky v ktorých užívateľ hodnotil niektorú časť simulátoru na stupnici 1 - 5 ,kde číslo 5 predstavuje najvyššie možné hodnotenie:

- Odpoveď "slabé" je nositeľom hodnoty 1 na stupnici.
- Odpoveď "dostatočné" je nositeľom hodnoty 2 na stupnici.
- Odpoveď "dobré" je nositeľom hodnoty 3 na stupnici.
- Odpoveď "kvalitné" je nositeľom hodnoty 4 na stupnici.
- Odpoveď "výborné" je nositeľom hodnoty 5 na stupnici.

Miera funkčnosti simulátoru bola ďalej overovaná formou jednoduchej prípadovej štúdie. Základné vlastnosti ako je *grafické spracovanie* a *miera funkčnosti simulátora*, na ktoré bol návrh simulátoru zameraný, boli overené stanovaním dvoch hypotéz:

- *Grafické spracovanie simulátoru je prepracované.*
- *Simulátor je pre svoj účel dostatočne funkčný.*

Následne bola zisťovaná pravdivosť jednotlivých hypotéz. K tomu slúžili dáta získané z vyplnených testovacích dotazníkov.

6.2 Vyhodnotenie testovania

Hypotéza, *Grafické spracovanie simulátoru je prepracované*, bola potvrdená na základe výsledku časti dotazníku zameranej na grafické spracovanie aplikácie. Celkové hodnotenie dosiahlo hodnoty 4.1, pričom najhorší výsledok dosiahlo hodnotenie užívateľského rozhrania.

Hypotéza, *Simulátor je pre svoj účel dostatočne funkčný*, bola potvrdená na základe výsledku časti dotazníku zameranej na funkčné aplikácie. Celkové hodnotenie dosiahlo o niečo nižšej hodnoty 3.6, pričom najhorší výsledok dosiahlo hodnotenie počítačom riadenej dopravy.

Zo spracovaných výsledkov vyplynulo nasledovné:

- Respondentom sa páčilo celkové grafické spracovanie aplikácie s výhradami k farebnému spracovaniu hlavného menu a menším chybám v objektoch.
- Jazdu testerí popísali ako dostatočne realistickú pre účely výuky autoškoly.
- Spracovanie kontroly pravidiel cestnej premávky bolo hodnotené kladne.
- Respondenti by uvítali dynamickejšie prostredie.
- Hodnotenie počítačom riadenej premávky popisovalo túto súčasť ako zaujímavé oživenie scény.
- Správanie počítačom riadenej premávky by vyžadovalo čiastočne vylepšit.
- Počet umiestnených dopravných značiek bol hodnotený ako dostatočný s pripomienkou k počtu križovatiek, ktorý by nebolo na škodu zvýšiť a pridať kruhové objazdy.
- Návrh ovladačích prvkov bol netradičný a podľa väčšiny respondentov lepší ako klasické ovládanie klávesnicou.

Je nutné si ale uvedomiť, že užívatelia sú zvyknutí na komerčné produkty, na ktorých vývoji pracujú tímy ľudí, a pri ktorých je časová dotácia na vyhotovenie mnohonásobne väčšia ako časové okno vyhradené pre túto prácu. Vzhľadom k pomerne kladnému hodnoteniu je možné vyvodit', že simulátor má svoje silné stránky a jeho vyhotovenie je vyhovujúce pre účely, pre ktoré bol navrhovaný. Dôležité je ale prihliadnúť najmä ku kritizovaným častiam implementácie a na základe spätnej väzby tieto časti vylepšit'. Táto spätná väzba tvorí pridanú hodnotu, na základe ktorej je možné nasmerovať budúci vývoj, a na základe ktorej je možné navrhnúť možné vylepšenia či rozšírenia aplikácie.

Kapitola 7

Záver

Cieľom tejto práce bolo vytvoriť simulátor autoškoly, ktorý má napomáhať pri príprave budúcich vodičov. Práca bola zameraná na prvky, ktoré v obdobných aplikáciách niesú bežné. Ťažiskom práce bolo návrhnúť prostredie simulácie, vytvoriť grafické modely a potrebné súčasti, vymyslieť a implementovať spôsob kontroly dodržiavania dopravných predpisov a do určitej miery simulovať dynamiku vozidla.

Simulácia je zasadená do rôznorodého prostredia počínajúc malým mestom, cez horské priechody až po rovinatej nížiny. Vďaka charakteru prostredia je možné simulovať nielen dopravné predpisy, ale aj nebezpečné či neprehľadné úseky ciest, na ktorých čakajú užívateľa nečakané udalosti, akými sú zúžená cesta, alebo cesta bez vodorovného značenia.

Prostredie pozostáva z 11 km ciest a mestského okruhu, pozdĺž ktorých je umiestnených 7 druhov budov v rôznych farebných kombináciach. Cesty sú doplnené o dopravné značenie vzťahujúce sa ku konkrétnym situáciám.

Následne bol vytvorený systém dopravných semaforov, ktorý tvorí značnú časť interaktívneho dopravného značenia. Systém podporuje následné možné rozšírenia a je jednoducho škálovateľný čo do počtu riadených križovatiek tak aj do počtu priepustných stavov pre každú križovátku zvlášť.

Bolo navrhnuté a implementované simulované vozidlo s ohľadom na jeho skutočnú predlohu. Z dôvodu reálneho pocitu zo simulácie, boli navrhnuté palubné ukazatele tak, aby pôsobili čo možno najreálnejšie a preto boli vytvorené ako 3D modely, ktoré sú súčasťou scény. Pohľad z vozidla napodobuje pozíciu a správanie vodičovej hlavy, teda aj jeho pohľadu.

Taktiež bola vytvorená simulácia účastníkov cestnej premávky, ktorú reprezentujú vozidla riadené počítačom. Účastníci premávky jazdia po celej cestnej sieti, a sú naprogramovaní tak, aby dodržiavali dopravné predpisy. Keďže implementácia týchto vozidiel je v celku zložitá záležitosť, má svoje rezervy a nieje dokonalá. Niekedy nastávajú situácie, keď počítačom riadené vozidlo spôsobí kolíziu, či ignoruje dopravné značenie. Do istej miery však toto správanie pridáva celému priebehu simulácie na realizme, keďže aj v skutočnom svete vznikajú dopravné situácie, kedy sa môže vozidlo riadené užívateľom dostať do kolízií spôsobenej iným účastníkom premávky.

Tvorba niektorých textúr bola súčasťou procesu tvorby 3D objektov. Zvyšné textúry boli prevzaté z voľne dostupných zdrojov [32].

Aktuálny stav práce predstavuje základ, ktorý implementuje všetky podstatné súčasti, a na ktorom je možné postaviť ďalšie možné rozšírenia, pridať nové pravidlá, či zložitejšie križovatky alebo nové modely vozidiel.

Literatúra

- [1] Auto-data.net: Skoda - Fabia I(6Y) - 2.0(115hp). [online], Naposledy navštíveno 13. 11. 2016.
URL https://www.auto-data.net/en/?f=showCar&car_id=14158
- [2] Besier 3D-Edutainment Wiesbaden: 3D-Driving-School. [online], Naposledy navštíveno 13. 11. 2016.
URL http://www.3dfahrschule.de/uk_index.htm
- [3] CCD mods: VAZ 2105 – 1.5.3. [online], Naposledy navštíveno 20. 10. 2016.
URL <http://www.ccdmods.com/wp-content/uploads/2016/12/BA3-2105.jpg>
- [4] CGtrader.com: 3D models for VR / AR, 3D printing and computer graphics. [online], Naposledy navštíveno 16. 11. 2016.
URL <https://www.cgtrader.com/free-3d-models>
- [5] Computer-bild: Screenshot-4-3D-Fahrschule-Europa-Edition. [online], Naposledy navštíveno 20. 10. 2016.
URL <http://i.computer-bild.de/imgs/7/4/4/2/7/8/Screenshot-4-3D-Fahrschule-Europa-Edition-745x585-bc465898240ff1f9.jpg>
- [6] COUMANS, E.: Bullet Physics SDK. [online], Naposledy navštíveno 1. 11. 2016.
URL <https://github.com/bulletphysics/bullet3>
- [7] Evžen Thöndel: Electric Motion Platform for Use in Simulation Technology. [online], Naposledy navštíveno 4. 5. 2017.
URL <http://www.pragolet.cz/elektromechanicky-aktuator>
- [8] Forward Development: City Car Driving 1.5 Description. [online], Naposledy navštíveno 13. 11. 2016.
URL <http://citycardriving.com/products/citycardriving>
- [9] GUMSTER, J. V.: *Blender For Dummies*. For Dummies, třetí vydání, 2015, ISBN 9781119039532.
- [10] HLRS: HLRS High Performance Computing Center Stuttgart - Motion Platform. [online], Naposledy navštíveno 26. 11. 2016.
URL https://www.hlrs.de/fileadmin/sys/public/solution_services/driving_simulator/RTEmagicC_motion_platform_small.png
- [11] HOCKING, J.: *Unity in Action: Multiplatform Game Development in C# with Unity 5*. Manning Publications, první vydání, 2015, ISBN 9781617292323.

- [12] <https://3d-driving-school.jaleco.com/>: Screenshots of 3D Driving-School Europa-Edition 5.1. [online], Naposledy navštíveno 20. 10. 2016.
URL https://cdn.jaleco.com/gen_screenshots/en-US/windows/3d-driving-school/large/3_3d-driving-school3.jpg
- [13] Ing.PAULÍK Marián: Zaujímavé technické údaje prevodoviek, prevodové pomery: 1.časť. [online], Naposledy navštíveno 3. 5. 2017.
URL <http://www.mpaulik.szm.com/prevodovky1.htm>
- [14] JAEGERS, K.: *XNA 4.0 Game Development by Example: Beginners Guide*. Packt Publishing, 2015, ISBN 9781849690669.
- [15] KOSCIANSKI, A.: *Game Programming with Irrlicht*. CreateSpace Independent Publishing Platform, 2011, ISBN 9781460905142.
- [16] Leadwerks software: Leadwerks Game Engine. [online], Naposledy navštíveno 1. 11. 2016.
URL <https://www.leadwerks.com/>
- [17] Learn OpenGL: Right handed coordinate system. [online], Naposledy navštíveno 3. 5. 2017.
URL https://learnopengl.com/img/getting-started/coordinate_systems_right_handed.png
- [18] LI, Z.; ZHU, C.; GOLD, C.: *Digital Terrain Modeling: Principles and Methodology*. CRC Press, 2004, ISBN 9780415324625.
- [19] MAILING, D.: *Coordinate Systems and Map Projections*. Pergamon, druhé vydání, 1992, ISBN 9780080372334.
- [20] MCCALLEY, B. W.: *Model T Ford: The Car That Changed the World*. Krause Publications, 1994, ISBN 0873412931.
- [21] Ministerstvo vnútra Slovenskej Republiky: VYHLÁŠKA Ministerstva vnútra Slovenskej republiky č. 9/2009 Z. z. [online], Naposledy navštíveno 1. 12. 2016.
URL http://www.minv.sk/swift_data/source/policia/dopravna_policia/zakony/UZ%209_2009%20k%201.1.2014.pdf
- [22] Ministerstvo vnútra Slovenskej Republiky: Zákon č. 8/2009 Z. z. [online], Naposledy navštíveno 1. 12. 2016.
URL http://www.minv.sk/swift_data/source/policia/dopravna_policia/zakony/UZ%208_2009%20k%201.1.2014.pdf
- [23] Ministerstvo vnútra Slovenskej Republiky: Dopravná nehoda. [online], Naposledy navštíveno 13. 5. 2017.
URL http://www.minv.sk/swift_data/source/policia/dopravna_policia/foto/foto_dn/2/0brazok017.jpg?v=2
- [24] Ministerstvo vnútra Slovenskej Republiky: Dopravná nehoda. [online], Naposledy navštíveno 13. 5. 2017.
URL http://www.minv.sk/swift_data/source/policia/dopravna_policia/foto/foto_dn/2/0brazok%20020.JPG?v=2

- [25] Ministerstvo vútra Slovenskej Republiky: Vyhodnotenie dopravno-bezpečnostnej situácie za rok 2015. [online], Naposledy navštíveno 17. 11. 2016.
URL http://www.minv.sk/swift_data/source/policia/dopravna_policia/dn/Vyhodnotenie%20DBS%20za%20rok%202015%20def..pdf
- [26] NEWTON, T.: *How Cars Work*. Black Apple Press, první vydání, 1999, ISBN 9780966862300.
- [27] Nvidia: 3D-Driving-School. [online], Naposledy navštíveno 10. 1. 2017.
URL [NVIDIAPHYSXSDK3.3.4Documentation](http://www.nvidia.com/content/PhysXSDK3.3.4Documentation)
- [28] OviLex software: Driving School Simulator | OviLex Software - Mobile, Desktop and Web Development. [online], Naposledy navštíveno 13. 11. 2016.
URL <http://www.ovilex.com/app/driving-school-simulator/>
- [29] OviLex Software: Screenshot of Driving School Simulator. [online], Naposledy navštíveno 20. 10. 2016.
URL <http://www.ovilex.com/wp-content/uploads/2014/09/screen9.jpg>
- [30] PACEJKA, H.; BESSELING, I.: *Tire and vehicle dynamics*. Butterworth-Heinemann, třetí vydání, 2012, ISBN 9780080970165.
- [31] RIESENFELD, R.; BARNHILL, R.: *A Class of Local Interpolation Splines*. Academic Press, 2014, ISBN 9781483235868.
- [32] Textures.com: Free texture resource. [online], Naposledy navštíveno 16. 4. 2017.
URL <http://www.textures.com>
- [33] Textures.com: BuildingsHighRise0291. [online], Naposledy navštíveno 8. 2. 2017.
URL <https://www.textures.com/download/buildingshighrise0291/17236>
- [34] Textures.com: NaturePlants0014. [online], Naposledy navštíveno 8. 2. 2017.
URL <https://www.textures.com/download/natureplants0014/107841>
- [35] The-blueprints.com: Skoda Fabia. [online], Naposledy navštíveno 20. 12. 2016.
URL https://www.the-blueprints.com/blueprints/cars/skoda/991/view/skoda_fabia/
- [36] Truegamers.it: Assi e due di picche del volante. [online], Naposledy navštíveno 26. 10. 2016.
URL <http://www.truegamers.it/wp-content/uploads/2014/10/driving-school-simulator-4.jpg>
- [37] Unity Software Ltd: Unity Script Reference. [online], Naposledy navštíveno 20. 4. 2017.
URL <https://docs.unity3d.com/ScriptReference/>
- [38] VAUGHAN, W.: *Digital Modeling*. New Riders, první vydání, 2012, ISBN 9780321700896.
- [39] Vidová, I.: PV189 Mathematics for Computer Graphics. [online], Naposledy navštíveno 2. 2. 2017.
URL <http://trymath.sweb.cz/collisionDetection.html>

- [40] Virage simulation: CAR DRIVING SIMULATOR. [online], Naposledy navštíveno 26. 10. 2016.
URL <http://viragesimulation.com/vs500m-car-simulator-training-and-research/>
- [41] Virage simulation: CAR DRIVING SIMULATOR – VS500M. [online], Naposledy navštíveno 26. 10. 2016.
URL <http://viragesimulation.com/wp-content/uploads/2013/10/vs550m-1.jpg>
- [42] Youtube.com: Let's Try: City Car Driving (3D Instructor) [Commentary] [HD]. [online], Naposledy navštíveno 20. 10. 2016.
URL http://img.youtube.com/vi/euroEj9_a18/mqdefault.jpg
- [43] ÚAMK: Kde jezdí nejvíce aut? [online], Naposledy navštíveno 6. 5. 2017.
URL <http://www.uamk.cz/aktuality/1310-kde-jezdi-nejvice-aut>

Príloha A

DVD

Priložené DVD obsahuje zdrojové kódy, modely, texty a všetky súbory potrebné k spusteniu aplikácie.

Príloha B

Dotazník

Testovací dotazník, ktorý bol využitý pri zhodnotení stavu aplikácie.

Testovací dotazník

Pred Vami práve leží testovací dotazník určený k získaniu užívateľskej spätnej väzby k simulátoru, ktorého vývoj je predmetom mojej Bakalárskej práce. Simulátor sa má zameriavať na simuláciu jazdy motorového vozidla, kontrolu dopravných predpisov a simuláciu účastníkov premávky.

Grafické spracovanie:

Ako by ste ohodnotili grafické vyhotovenie simulátoru ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako hodnotíte užívateľské rozhranie ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako Vás zaujalo prostredie do ktorého je zasadená simulácia ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Pôsobí pohľad z kabíny vozidla realisticky ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Parametre simulácie:

Je rozsah mestského okruhu dostatočne veľký?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako by ste ohodnotili jazdný model vozidla?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako hodnotíte počítačom riadenú premávku ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako hodnotíte kvalitu odhaľovania dopravných priestupkov ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Ako hodnotíte návrh ovládacích prvkov ?

slabé	dostatočné	dobré	kvalitné	výborné
-------	------------	-------	----------	---------

Tu popíšte vaše postrehy, a pocity z testovania simulátora:

Ako by ste testovaný simulátor vylepšili ?
