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

DRIVING SCHOOL - RULES OF THE ROAD

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. PAVEL MELCER**
AUTHOR

VEDOUCÍ PRÁCE **doc. Ing. PETER CHUDÝ, Ph.D. MBA**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Melcer Pavel, Bc.**
Program: Informační technologie Obor: Počítačová grafika a multimédia
Název: **3D Autoškola**
Driving School - Rules of the Road
Kategorie: Modelování a simulace

Zadání:

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 budu 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

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Chudý Peter, doc. Ing., Ph.D. MBA**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 1. listopadu 2018

Abstrakt

V této práci je řešena problematika trénování řidičských dovedností s využitím 3D simulátoru autoškoly, který kontroluje dodržování pravidel silničního provozu. Existující simulátory jsou rozděleny podle charakteristik a popsány. Text obsahuje soupis implementovaných pravidel a souhrn vhodných nástrojů. Výsledná aplikace je postavena na Unreal enginu a text popisuje jednotlivé fáze vývoje.

Abstract

This work deals with driving skills training problematics using a driving school 3D simulator, which monitors the observance of road traffic rules. Existing simulators are categorized by characteristics and described. The text contains a list of implemented rules and a summary of suitable tools. The resultant application is based on the Unreal engine and the text describes the various stages of development.

Klíčová slova

trenažér vozidla, Unreal Engine, dopravní pravidla, simulátor autoškoly, Blender, herní jádro

Keywords

vehicle simulator, Unreal Engine, Traffic rules, Driving simulator, Blender, Game engine

Citace

MELCER, Pavel. *3D Autoškola*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Peter Chudý, Ph.D. MBA

3D Autoškola

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doc. Ing. Petera Chudého, Ph.D. MBA. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Melcer
30. července 2019

Poděkování

Děkuji panu doc. Ing. Peteru Chudému, Ph.D. MBA za jeho vedení a podnětné návrhy.

Obsah

Seznam symbolů

F_H	hnací síla v podélném směru [N]
O_f	odpor valení [N]
O_s	odpor sklonu [N]
O_{vz}	odpor vzduchu [N]
O_a	odpor zrychlení [N]
F_{zP}	síla působící na přední nápravu [N]
F_{zZ}	síla působící na zadní nápravu [N]
G_v	tíha vozidla [N]
M_{MV}	kroutící moment na výstupu hřídele [Nm]
n_{MV}	počet otáček hřídele [1]
i_{PV}	poměr převodové soustavy [1]
m_v	hmotnost vozidla [kg]
g	tíhové zrychlení $[m \cdot s^{-2}]$
η_{PV}	účinnost převodového ústrojí [1]
δ	součinitel vlivu rotujících částí [1]
v_x	náporová rychlost vzduchu $[m \cdot s^{-1}]$
p_d	dynamický tlak $[N \cdot m^{-2}]$
S_x	čelní plocha vozidla $[m^2]$
c_x	součinitel odporu vzduchu [1]
ρ	hustota vzduchu $[kg \cdot m^{-3}]$
F_{zK}	radiální reakce vozovky [N]
e	rameno valení [m]
r_d	dynamický poloměr kola [m]
M_k	hnací moment na kole [Nm]
F_k	hnací síla na kole [N]
f	součinitel odporu valení [1]
T	testovací kritérium
α	hladina významnosti
v	stupeň volnosti
m	průměrné bodové hodnocení
n	počet testovaných uživatelů
s	směrodatná odchylka
μ	očekávaná hodnota
RPM	počet otáček za minutu
PV	převodová soustava
MRT	Modular road tool
H	hypotéza

Kapitola 1

Úvod

Začátek řízení dopravních vozidel se převážně datuje k 18. století, kdy roku 1769 Francouz Nicolas Joseph Cugnot poprvé převezl čtyři pasažéry s maximální rychlostí 9 km za hodinu. První vozidla jezdila na parní pohon, který byl neustále zrychlován, ale jeho provozní náročnost bránila v rozšíření. Karl Benz v roce 1885 uvedl první automobil pohybující se pomocí spalovacího motoru, díky kterému byla jízda jednodušší, přesto nedošlo k rozšíření, kvůli vysoké pořizovací ceně. Zlom nastal až v roce 1908, kdy Henry Ford představil cenově dostupný automobil s názvem Ford model T. S rostoucím počtem vozidel na komunikacích, bylo potřeba začít vzdělávat a kontrolovat řidičské schopnosti, a tedy v roce 1910 Hugh Stanley Roberts založil první školu pro motorová vozidla British School of Motoring (BSM) v Londýně [36].

Kolem roku 1920 bylo v Československu registrováno necelých 5000 vozidel, ale kvůli vyšší dostupnosti a lepšímu komfortu se za 100 let počet zvýšil tisíckrát na 5,7 miliónu [20]. Stinnou stránkou vysokého počtu automobilů je počet dopravních nehod, kdy jich během roku 2017 nastalo 103 821, z toho 502 smrtelných [35] a přestože se meziročně snížil počet smrtelných nehod o 7,9 %, celkový počet nehod vzrostl o 5 %. Proto je nutné striktně vzdělávat a kontrolovat nové řidiče, kdy kvůli rozvoji výpočetní techniky a počítačové grafiky nastala změna ve způsobu výuky v autoškolách. Noví studenti jsou nejprve vzděláváni v dopravních předpisech a jízdu trénují na trenažérech ještě před usednutím do skutečného vozidla. Výhodou je praktický trénink nově získaných informací bez ohrožení ostatních účastníků dopravního provozu. Právě takovým trenažérem se předložená práce zabývá.

Ve druhé kapitole jsou uvedeny některé existující trenažéry a budou popsána některá dopravní pravidla, jež budou v práci kontrolována. Na začátku třetí kapitoly jsou zmíněna některá fyzikální jádra, a poté popsána nejpopulárnější volně dostupná herní jádra v současné době. Ve čtvrté kapitole jsou uvedeny fyzikální vlastnosti, využívající se při simulování chování automobilu. Pátá kapitola se věnuje architektuře simulátoru a uživatelskému rozhraní se způsoby ovládání aplikace. Šestá kapitola uvádí postup práce za účelem vymodelování simulovaného města, zejména vytvoření a získání 3D modelů komponent města. Sedmá kapitola se zaměřuje na implementaci simulátoru v herním jádře Unreal Engine a popisuje strukturu vytvořeného projektu, zajímavé aspekty při tvoření modelového města. Současně je také uveden způsob řízení vozidel v Unreal. Na závěr se kapitola věnuje kontrole dodržování silničních pravidel. Poslední kapitola uvádí způsob a výsledek testování aplikace.

Kapitola 2

Simulace silničního provozu a kontrola dodržování dopravních pravidel

Simulování dopravních vozidel je uplatňováno v mnoha oborech. V urbanismu pomáhá při úpravách územních plánů. Inteligentní řízení dopravy simulací získává informace o chování plynulosti dopravy v čase, zejména délky kolon v různém období, dobu čekání jednotlivých účastníků nebo množství vyprodukovaných emisí v dané lokalitě. Na základě získaných informací je doprava efektivně řízena. Rovněž v automobilovém průmyslu je simulace využita při návrhu nových vozidel, neboť před zhotovením finančně nákladného prototypu lze ladit parametry vedoucí k lepší aerodynamice, výkonu motoru nebo jízdnímu komfortu [39].

Následující práce se věnuje modelováním automobilu v grafickém virtuálním prostředí. Tato problematika je využívána mnoha různými způsoby, ale nejčastěji v herním průmyslu. Zde ovšem není kladen důraz na kompletní realističnost vozidla a model neposkytuje kompletní možnosti ovládání. Například hodně her zanedbává převod stupňů rychlosti nebo signalizaci změn směru jízdy. Nejzásadnější je ovšem absence kontroly dodržování silničních pravidel. V této práci je kladen důraz právě na realistické převedení a ztvárnění skutečného světa. Díky čemuž lze aplikaci použit v ve výcviku nových řidičů.

2.1 Trénování řidičských dovedností

V důsledku rozšíření vozidel za první republiky vstoupila v platnost řada státních předpisů v dopravě a byly otevírány školy ke vzdělávání v řidičských dovednostech. Název Autoškola se ustálil až roku 1964, dříve školy nesly pojmenování Autoučiliště. Doprava začala být regulována dopravními značkami a pravidly silničního provozu. V autoškolách se osnova výuky mnohokrát vyvíjela až do podoby, kterou známe dnes [2].

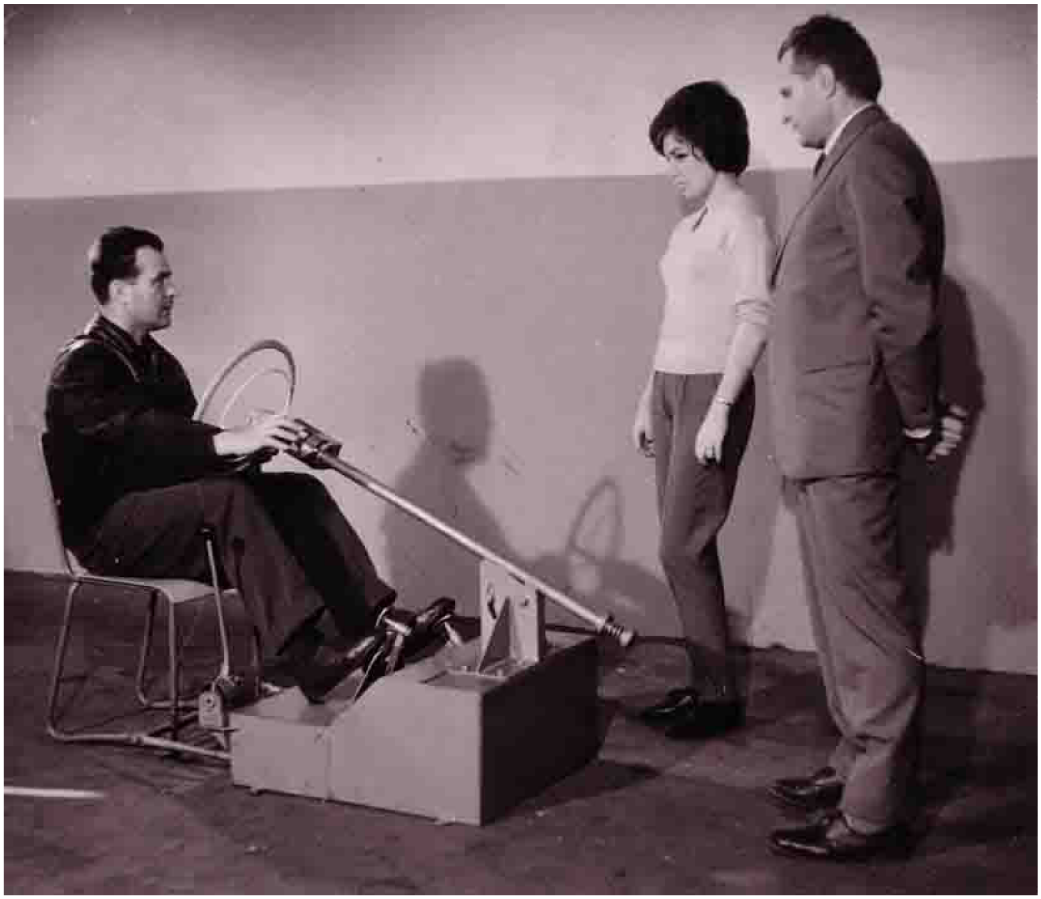
Uchazeč o řidičský průkaz, který žádá o povolení k řízení vozidla na veřejných pozemních komunikacích, musí absolvovat několika týdenní kurz zakončený závěrečným testem. Kurzy začínají studiem dopravních pravidel a předpisů se souběžným tréninkem na profesionálních trenažérech. Student si na nich osvojuje řízení vozidla s praktickým dodržování předpisů. Tyto trenažéry lze rozdělit do dvou kategorií, kdy v první kategorii je vozidlo simulováno s fyzikálním působením pohybu na osobu a do druhé kategorie patří pouze softwarová řešení.

Hardwarové trenažéry

Kvůli nedostatku vozidel v autoškolách postupně vznikaly nejrůznější cvičné trenažéry. Na začátku 60. let byl v Československu představen první Trenažér K1 2.1, který pomáhal studentům v osvojení manipulace s volantem, rozjezdem nebo řazením rychlostí. Vývojem dalších trenažérů se věnoval podnik Automobilní opravárenský závod (zkr. AOZ) Olomouc. Vývoj byl zaměřen na sofistikovanější zařízení než K1. První model ART 65 2.2 přišel s projekčním plátnem, kde byly promítány diapozitivy pro navození reálnějšího pocitu z jízdy [2].

Slabou stránkou dřívějších systémů, které nevyužívaly výpočetní techniku, byla absence zpětné reakce na událost vykonanou řidičem a neexistovala vazba mezi více účastníky dopravního provozu. Díky rozvoji 3D počítačové grafiky lze nyní trénovat a hlavně zpětně kontrolovat mnoho jiných řidičských zdatností, které byly dříve zanedbávány. Například dříve nešlo kontrolovat průjezd světelnou křižovatkou nebo správnou reakci na dopravní značení.

V současnosti probíhá vývoj v několika firmách, ale nelze nezmínit JKZ Olomouc. Firma vznikla z bývalé vývojové skupiny podniku AOZ. AT-208 VRT 2.3 je nejnovější model, který využívá interiéru vozu Škoda Fabia. V portfoliu JKZ je také trenažér kamiónu, autobusu nebo cyklo-trenažér [2].



Obrázek 2.1: K1 - první v ČSR, zdroj [2]

Obrázek 2.2: ART 65, zdroj [2]

Obrázek 2.3: AT-208 VRT - interiér vozu Škoda Fabia, zdroj [26]

Hardwarové trenažéry jsou často zkonstruovány z prvků pocházejících ze skutečných vozidel. Těmito součástmi jsou: volant, palubní deska, sedadla, řadící páka a další. Jejich silná stránka spočívá především v simulaci pohybu kabiny v prostoru. Simulační software vypočítává fyzikální síly, které by na vozidlo během jízdy působily. Následně pomocí hyd-

(a) volant G920, zdroj [27]

(b) řadící páka, zdroj [28]

Obrázek 2.4: Periferní zařízení Logitech

raulik nebo elektro-mechanických systémů vypočtené síly přenáší do konstrukce trenažéru. Pomocí nich lze trénovat rovněž jízdní komfort, zejména adekvátní sílu stisku pedálů brzdy pro trénink plynulého brzdění.

Softwarové aplikace

Následující trenažéry nejsou sestaveny z fyzických komponent. Jedná se o počítačové, konzolové nebo mobilní aplikace, jejichž cílem je pouze vizuální simulace jízdy. Kvůli absenci fyzikálních sil působících pří jízdě, nelze získat shodný zážitek během jízdy jako u předchozího typu. Rovněž neobsahují části ze skutečných vozidel, ale lze připojit podobná periferní zařízení. Zejména počítačový volant s řadící pákou, viz obr. 3.2.

Aplikace simulující jízdu lze rozdělit do tří kategorií. Mezi první patří čistě zábavné hry, například série Grand Theft Auto, závodní hry a další. Věrně simulují pohyb vozidla, ale jsou opomíjeny aspekty nutné k trénování jízdy, jako kontrola pravidel, zapínání pásu nebo souhra manipulace spojky a plynu u manuálního řazení.

Do druhé skupiny patří aplikace, které lze rovněž považovat za hry. Je u nich ovšem kladen silný důraz na věrohodnost řízení. Lze ovládat blinkry, zpětná zrcátka mají svůj význam, apod. Poslední skupina je nejstriktnější a obsahuje aplikace vyvinuté právě pro potřeby autoškol. Vedou ke správné jízdě a za nedodržení pravidel lze obdržet penalizaci. Hranice mezi druhou a třetí skupinou je velmi tenká, jelikož lze nalézt mnoho aplikací spadajících do druhé kategorie, které formou hry cíleně vedou k tréninku jízdy, čímž jsou podobné třetí kategorii. Ve zbytku této části budou uvedeny některé existující aplikace druhé a třetí skupiny.

- **SimExam**

Vyvinuto firmou SimSpace zabývající se simulačními nástroji s různorodou tématikou. Nabízí letecké simulátory, simulace proudění vzduchu nebo zbraní. Prvním produktem byl právě simulátor vozidel SimExam, který slouží k učení základního řízení. Podporuje externí volant s pedály a obraz lze zobrazovat až na 3 displeje současně.

Silnou stránkou je velký důraz na důsledné napodobení autoškol, neboť úkony požadované při zahájení aplikace silně vedou k tréninku řidičských návyků před započetím samotné jízdy. Například uživatel je nucen k připnutí bezpečnostními pásy, nastartování motoru a

odjištění parkovací brzdy. Bohužel tyto úkony jsou splněny pouhým kliknutím do zaškrtávacího pole v dialogu, který je zobrazen při zahájení jízdy. Teprve až jsou zaškrtnuty všechny požadované pole, lze ovládat vozidlo.

Při řízení je nutné manuálně přeřazovat rychlostní stupně, ale není zde zabudovaná interakce se spojkou. V průběhu jízdy hlasový trenažér informuje o aktuálních přestupcích a sděluje další požadované akce. Slabou stránkou je zastaralá grafická vizualizace a již zmíněná neexistence spojky. Pohybem myši dopředu je přidáván plyn a posunem dozadu dochází k brzdění, toto nestandardní ovládání bylo ze začátku matoucí, ale výhodou je umožnění plynulejší regulace, neboť klávesy mají pouze dva stavy. Mírně stinnou stránkou tohoto ovládání je nutnost rozhlížet se nikoli myší, ale pomocí klávesnice, jak bylo původně očekáváno. Po ukončení jízdy je zobrazen soupis všech přestupků, které nastaly během jízdy [37].

Obrázek 2.5: SimExam, zdroj [38]

- **City Car Driving**

Realistický herní simulátor pro řízení, který byl vydán v roce 2016 a vytvořen Forward Development, Ltd. Silnou stránkou je dobré vizuální zpracování a velice propracovaná simulace vozidla. Na začátku jízdy je nutné se připoutat, ale oproti SimExam je úkon proveden stiskem klávesy. Rovněž nastartování je provedeno klávesou a nikoli zakliknutím v dialogu. Díky tomu simulátor poskytuje mnohem lepší dojem, neboť evokuje pocit skutečné jízdy v reálném vozidle.

Spojka zde má velký význam, neboť bez správné manipulace s ní a plynem dochází k vypnutí motoru. Motor je rovněž vypnut při řazení převodů bez spojky. Slabou stránkou je neexistence hlasového instruktora a závěrečného přehledu všech chyb, přestože jsou během tréninku důsledně kontrolovány silniční pravidla a vypisována v informačním hlášení. Problematické je také ovládání, jelikož při výchozím rozvržení klávesnice je velmi náročné koordinovat jednotlivé úkony při řízení, což lze vyřešit připojením externího volantu [21].

Obrázek 2.6: City car driving, zdroj [21]

2.2 Platná dopravní pravidla a předpisy v simulaci

Automobilová doprava je regulována dopravními pravidly, které musí každý řidič dodržovat. Platným zákonem během této práce je zákon č.361/2000 Sb. [19], který upravuje provoz na pozemních komunikacích a vyhláška č. 294/2015 Sb.[18] definující dopravní značení na pozemních komunikacích. Implementace všech pravidel je nad rámec této práce, a proto budou v následující části uvedeny výňatky pravidel, které budou kontrolovány. Pravidla jsou řazena podle priority implementace.

- **Rychlost jízdy §18**

Řidič vozidla, o maximální přípustné hmotnosti nepřesahující 3 500 kg, a autobusu smí jet mimo obec rychlostí nejvýše 90 $km.h^{-1}$a na silnici pro motorová vozidla rychlostí 110 $km.h^{-1}$. Na dálnici je povolena maximální rychlost 130 $km.h^{-1}$, u ostatních motorových vozidel nejvýše 80 $km.h^{-1}$. V obci je maximální rychlost 50 $km.h^{-1}$platná pro všechny druhy motorových vozidel [19].

- **Jízda křižovatkou §22**

Mezi křižovatky patří místa na pozemní komunikaci, kde se kříží dvě a více komunikací. Provoz je zde řízen dopravními značkami, předností zprava nebo světelnými signály, které využívají tří barev. Zelený signál povoluje jízdu v daném směru a červený přikazuje zastavení v daném směru. Bliká-li na nich oranžové světlo jsou světelné signály mimo provoz a řidiči jsou povinni se řídit dopravním značením. Používány jsou pojmy hlavní a vedlejší komunikace, kde vozidla na hlavní komunikaci mají vždy přednost neodbočují-li vlevo, odbočují-li, pak platí pravidlo o odbočování. Vozidla na vedlejší komunikaci jsou vždy povinna dát přednost vozidlům na hlavní komunikaci. Tvar a typ komunikace je znázorněn na dodatkové tabulce [19].

- **Odbočování §21**

Během odbočování na křižovatce nebo na místo mimo pozemní komunikaci má vozidlo povinnost signalizovat změnu jízdy a při odbočování nesmí ohrozit řidiče jedoucí za ním [19].

- **Zastavení a stání §25**

Stání je definováno jako uvedení vozidla do klidu nad dobu povolenou pro zastavení. Zastavení znamená uvést vozidlo do klidu na dobu nezbytně nutnou k neprodlenému nastoupení nebo vystoupení přepravovaných osob anebo k neprodlenému naložení nebo složení nákladu.

Řidič smí zastavit a stát vpravo ve směru jízdy co nejblíže k okraji pozemní komunikace a na jednosměrné komunikaci může zastavit vpravo i vlevo. Dále smí zastavit v jedné řadě a rovnoběžně s okrajem komunikace; nedojde-li k ohrožení bezpečnosti a plynulosti silničního provozu, smí v obci řidič vozidla o celkové hmotnosti nepřevyšující 3 500 kg zastavit a stát kolmo, popřípadě šikmo k okraji pozemní komunikace nebo zastavit v druhé řadě. Při zastavení i stání musí zůstat volný alespoň jeden jízdní pruh široký nejméně 3 m pro každý směr jízdy. Zajíždí-li řidič za účelem zastavení nebo stání k okraji pozemní komunikace nebo k chodníku, musí dávat znamení o změně směru jízdy. Během vyjíždění od okraje musí také řidič dávat znamení o změně směru jízdy a nesmí ohrozit ostatní účastníky provozu [19].

V práci je kontrolován zákaz zastavení na místě definovaném zákazovou značkou.

- **Zákaz vjezdu**

V práci je kladen důraz na kontrolu dopravní značky Zákaz vjezdu ve všech směrech a Zákaz vjezdu v jednom směru, které jsou definovány v pravidlu [18].

- **Objíždění §16**

Pokud řidič při objíždění vozidla, jež zastavilo, resp. stojí nebo při objíždění překážky provozu na pozemních komunikacích či chodce, vybočuje ze směru jízdy, nesmí omezit ani ohrozit protijedoucí řidiče a ohrozit ostatní účastníky provozu na pozemních komunikacích. Rovněž musí dát znamení o změně směru jízdy [19].

- **Předjíždění §17**

Předjíždí se vozidlo zleva. Zprava pouze vozidla měnící směr vlevo a je jasno o dalším směru jeho jízdy. Pokud vozidlo při předjíždění vybočuje ze směru své jízdy, musí řidič dát znamení o změně směru. Po předjetí a zařazení před předjížděné vozidlo je opět povinen dát znamení. Ve všech případech nesmí ohrozit předjížděné vozidlo [19].

- **Železniční přejezd §28 a §29**

Před železničním přejezdem je stanovena maximální rychlost na 30 $km.h^{-1}$. Svítí-li bílé přerušované světlo signálu, je rychlost stanovena na 50 $km.h^{-1}$. Na přejezd nesmí vozidlo vjíždět v těchto případech: je-li dávána výstraha dvěma červenými střídavě přerušovanými světly signálu přejezdového zabezpečovacího zařízení, sklápějí-li se závory nebo je-li dávána výstraha přerušovaným zvukem houkačky [19].

- **Osvětlení §32**

Během jízdy musí být rozsvícena obrysová světla a potkávací světla nebo denní světla [19].

2.3 Dopravní značení

Pozemní dopravní komunikace je označena dopravními značkami, které je nutné při jízdě dodržovat. Dělí se na dvě základní podkategorie.

Svislá značení

V této práci existují pouze svislé značky v provedení definovaném ve vyhlášce [18] jako §1 a část 1., tzv. na pevném sloupku, budově nebo jiné konstrukce pevně připevněné se zemí. Značka platí pro celou pozemní komunikaci v daném směru jízdy, pokud není přerušena jinou. Dále se značky dělí podle jejich dopravního významu. V aplikaci se nachází následující typy dopravních značek.

- Zákazové značky - definují zákazy nebo omezení, které musí respektovat účastníci na pozemních komunikacích.
- Značky upravující přednost - definují a upravují přednost v jízdě na pozemních komunikacích.
- Příkazové značky - účastníci v provozu jsou povinni se řídit danými příkazy.
- Informativní značky - sdělují dodatečné informace na komunikacích.
- Výstražné značky - upozornění na místa s nutnou zvýšenou opatrností.

Vodorovná značení

V této práci jsou použita pouze značení z kategorií podélné a příčné čáry, šipky, označení stání a parkovišť, označení zastávek. Kontrolovány budou zejména značky, které způsobují porušení platných předpisů.

Kapitola 3

Architektura simulátoru a uživatelského rozhraní

Na začátku kapitoly jsou předloženy cíle, které by měly být v rámci této práce splněny. Poté se kapitola věnuje struktuře herních jader, kdy jsou zde popsána tři aktuálně nejpopulárnější herní jádra, a v souvislosti s nimi jsou rovněž zmíněny prostředky, které slouží k výpočtům fyzikálního chování objektů ve scéně. Následně jsou uvedeny fyzikální vlastnosti vozidla, které je nutné simulovat při pohybu. Zbytek kapitoly se věnuje architektuře a rozhraní vytvořené aplikace.

Cíle simulátoru

Na rozdíl od některých existujících aplikací je v této aplikaci cílem kontrola často zanedbávaného železničního přejezdu, neboť v řadě simulátorů chybí železniční síť. Rovněž je kladen důraz na manuální převod rychlostních stupňů, neboť stále se na pozemních komunikacích objevuje většina vozidel s tímto řazením. S tím souvisí implementace způsobu plynulé regulace míry působení na jednotlivé pedály při ovládání pomocí klávesnice a také je očekávána podpora pro připojení externího volantu s pedály. Stejně jako v ostatních aplikacích je cílem také propracované dopravní značení.

3.1 Prostředky ke tvorbě vizuálních aplikací

Jde o sofistikované nástroje, jejichž cílem je abstrahovat vývoj od detailů jako zajištění multiplatformnosti, vykreslování a sestavení jednotlivých snímků na grafické kartě, odchytávání vstupů od uživatele, podpora zvuků, vizuální editor scény a mnoho dalších. Tato abstrakce zrychluje vývoj, neboť se lze rovnou zaměřit na tvorbu samotného produktu. Rovněž je urychlena integrace nových funkcionalit, které poskytuje moderní hardware, jelikož úpravu provádějí právě vývojáři herních jader, nikoli her samotných [23].

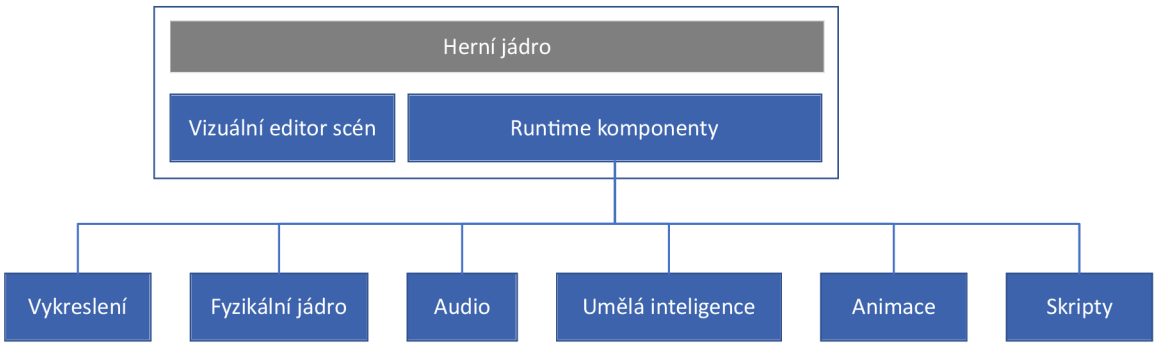
Ogre

Object-Oriented Graphics Rendering Engine je multiplatformní open source šířený pod licencí MIT. Nejnovější verze využívá programovatelnou pipelinu, ale tato verze není plně podporována a neexistuje doposud mnoho studijního materiálů.

Na rozdíl od ostatních prostředků v této části nelze Ogre řadit mezi herní jádra. Jedná se čistě jen o 3D engine, jehož cílem je flexibilita. Zaměřuje se pouze na vykreslování a neobsahuje komponenty pro práci se zvukem, umělou inteligencí a sítěmi. Výhodou je možnost seznámit se hlouběji s vývojem, neboť je zde větší přístup k OpenGL či DirectX. Nedostatkem je nutnost implementace ostatních komponent a absence vizuálních nástrojů k tvorbě scény. Z těchto důvodů nebyl Ogre v práci použit. [34].

Architektura herního jádra

Herní jádra jsou složena z komponent uvedených na obrázku 3.1 [30].



Obrázek 3.1: Komponenty v herním jádře

- **Audio**

Komponenta zajišťuje práci se zvukovými nahrávkami. Umožňuje editaci zvukových stop, kontroluje, zda má být v dané lokaci slyšet nebo reguluje hlasitost nahrávky. Bez zvuků hra ztrácí požadovanou atmosféru, proto každé herní jádro má tuto komponentu zabudovanou. Rozdíl je pouze v počtu implementovaných zvukových efektů.

- **Umělá inteligence**

V rámci této práce je nutný pohyb ostatních vozidel na pozemní komunikaci. Musí jezdit ve správném směru, regulovat rychlost a zastavovat před světelným semaforem. Herní jádra nabízejí vždy několik možných řešení k implementaci inteligence. K pohybu je nejčastěji používán navigační povrch NavMesh, který je automaticky vygenerován charakteristikou povrchu nebo podle okolních statických objektů. Akce jsou vybírány dle řízení konečného automatu, jehož implementace je čistě na vývojářích, komponenta jen poskytuje prostředky k jeho sestavení.

- **Animace**

Scény obsahují především dynamické objekty, u kterých je nutné simulovat jejich pohyb. Animace definuje polohu všech části daného objektu v čase. Herní jádra umožňují tvorbu jednoduchých animací, ale náročnější skeletální animace jsou vytvářeny v externích nástrojích, kde je pohyb zaznamenán z fyzického světa pomocí čidel. Výběr aktuální animace ve hře je řízen stavovým automatem.

- **Skripty**

Obsahuje řadu podpůrných funkcí usnadňující definici herních mechanik či pravidel dle specifikace chování objektů, které je potřeba v běhu aplikace programově řídit podle vstupních událostí. Komponentu má každé jádro, liší se pouze ve způsobu použití.

- **Fyzikální jádro**

Jedná se často o knihovny sloužící k výpočtu fyzikálních rovnic, díky nimž je ve hrách simulována fyzika. Pozitivní stránkou pro jejich použití při vývoji je jejich znovu-použitelnost a jsou vyvíjeny již mnoho let, kdy již byly použity v mnoha finálních produktech, proto bývají dobře optimalizované a vyladěné. Vývoj vlastní fyziky mnohonásobně prodražuje celý projekt, a proto i velké vydavatelské firmy využívají již existujících knihoven, které jsou přizpůsobovány pro specifické potřeby firem.

- **Vykreslení**

Nejdůležitější komponenta obaluje a zjednodušuje nízko-úrovňové API OpenGL nebo DirectX při komunikaci s grafickou kartou. Obsahuje vyladěné implementace řady algoritmů nutných při kreslení scény. Zejména Frustrum a Occlusion ořezaní scény. LOD objektů snižující počet kreslených polygonů kvůli efektivnímu využití výkonu grafické karty. Kvůli odstínění API grafické karty jsou shadery pro různé efekty vytvářeny pouze ve vizuálním editoru. V něm je materiál sestaven propojením různých bloků a kompilace je provedena automaticky.

3.2 Přehled fyzikálních jader

Obrázek 3.2: Loga uvedených fyzikálních knihoven

Bullet SDK

Nabízí především řešení pro detekci kolizí objektů ve 3D prostoru. Používá se k simulaci chování pevných nebo měkkých těles jako jsou tkaniny, lana nebo deformovatelné objekty. Bullet je akcelerován na GPGPU pomoci OpenCL nebo CUDA, která je proprietární technologií firmy Nvidia a funkční pouze na grafických kartách této firmy. Bullet je poskytován jako open source a šířen pod volnou licencí zlib. Uplatňuje se ve filmovém a herním průmyslu. Nalézt jej lze ve filmech Sherlock Holmes, Mega-mysl nebo Hancock. Z her je například použit v závodní sérii Dirt. Rovněž je zabudován v herním jádře RAGE od firmy Rockstar games, který je od roku 2006 základem všech jejich her. Bullet je zabudován také v 3D nástrojích Blender nebo Cinema 4D [5].

Nvidia PhysX SDK

Technologie vyvinutá firmou Nvidia, která byla do 3 verze proprietární a fungovala pouze na jejich grafických kartách. Dne 3.12.2018 byla vydána nová verze, která situaci mění. Nvidia

uvolnila 4. verzi jako open source a je šířena pod licencí BSD 3. PhysX je multiplatformní fyzikální jádro, podporující řadu zařízení s cílem provádět fyzikální výpočty v reálném čase na GPGPU, které obsahují vysoký počet paralelních jader. 3D modelovací nástroje firmy Autodesk Maya a Max využívají právě této technologie. Rovněž velká spousta her provádí výpočty pomocí PhysX, neboť v dnešní době jej využívají nejpopulárnější herní jádra Unity, Unreal Engine nebo Cryengine. Nově uvedená 4. verze PhysX SDK byla přepracovaná pro použití v robotice, inteligentních vozidlech nebo obecně v počítačovém vidění, a právě kvůli těmto cílům je nyní volně k dispozici [33].

Havok Physics

Havok Physics byl vyvinut v roce 2000 firmou Havok, patří tedy mezi nejstarší fyzikální jádra. Nástroj je multiplatformní a na rozdíl od výše zmíněných je stále proprietární. Vlastněn je od roku 2015 firmou Microsoft. Vydavatelská firma Ubisoft má Havok zabudovaný ve svém vlastním herním jádře a je použit v řadě nově vydaných her, například je na něm postavena herní série Assassin's creed od své první verze vydané v roce 2007. [24].

3.3 Přehled herních jader

Unity

Firma Unity Technologies na konferenci Apple v roce 2005 uvedla herní jádro 3.3. První verze byla pouze pro operační systém OS X, ale v novějších verzích byla přidávaná podpora pro další platformy. Dnes se jedná o multiplatformní nástroj podporující více než 25 platforem, včetně zařízení pro virtuální realitu. Unity slouží ke tvorbě 3D i 2D her. Díky intuitivnímu a jednoduchému ovládání je velmi populární. Uplatnění nachází především u menších projektů, například tzv. Indie hry, které jsou vyvíjeny malým týmem bez finanční podpory vydavatelů, jsou postavené právě na Unity. Webové a mobilní hry běží na Unity, protože u ostatních jader doposud není vysoká podpora.

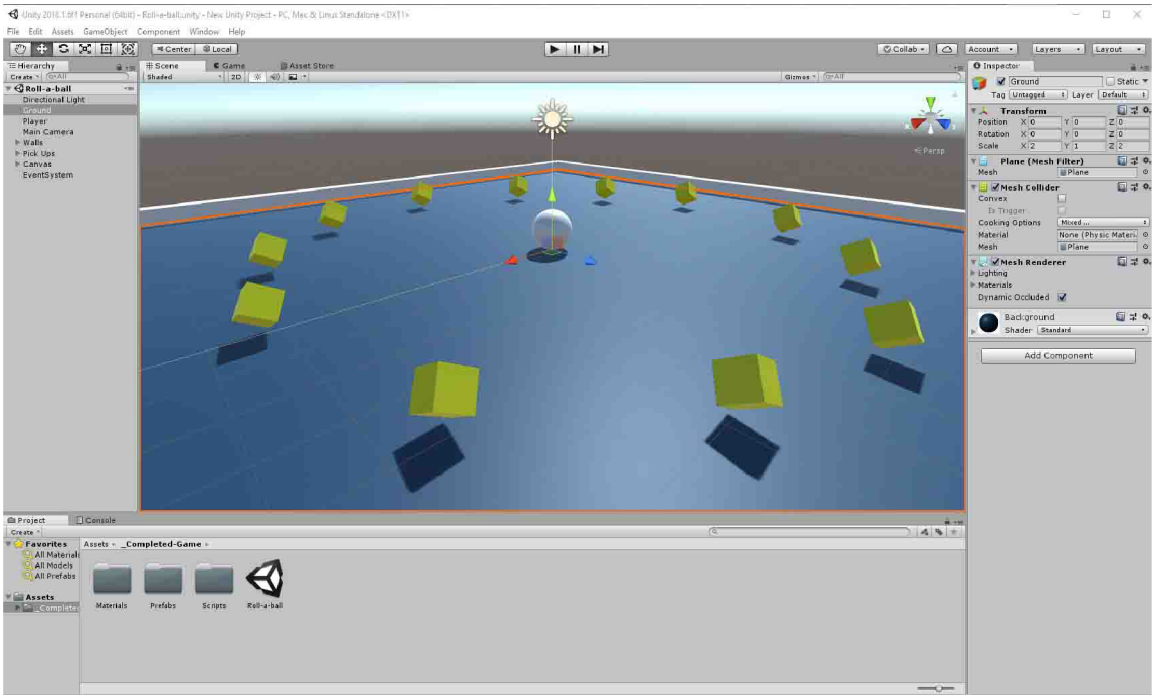
Unity umožňuje import rozsáhlejšího počtu 3D formátů než ostatní nástroje, které podporují jen FBX. V Unity lze importovat 3D formáty dae, 3ds, max nebo blend, který je použit pro uložení v modelovacím nástroji Blender. Skripty lze psát v programovacích jazycích Bo, UnityScript (odvozen od jazyku Javascript) a zhruba 80 % aplikací je napsáno v jazyce C# od firmy Microsoft.

Unity je dostupné s vizuálním editorem pro tvorbu scén, ve kterém lze aplikaci okamžitě spustit a ladit parametry hry. Slabou stránkou je zpracování materiálů a světel, a proto je vizuální výsledek horší než u Unreal nebo Cry enginu. Výhodou je ovšem velmi rozsáhlý a relativně levný katalog tzv. Assetů, které je možno zakoupit a využívat. Unity je bezplatný pro nekomerční použití, při komerčním použití je cena odvozena od ročního výdělku [4].

CryEngine

Vyvinut firmou Crytek v roce 2002. Zpočátku byl využit pouze v technologickém demu firmy Nvidia, ale nakonec byl použit v samostatné hře Far Cry. Bezplatné použití pro nekomerční účel je umožněno až od páté verze CryEngine a pro komerční u této verze ze začátku platila licence "Pay what you want", ale v roce 2018 byla licenční politika změněna na 5% příjem z celkového výnosu.

CryEngine je využíván zejména ve vývoji AAA herních titulů na PC a herních konzolích. Grafická vizualizace je mnohem realističtější než u Unity, ale srovnatelná s Unreal



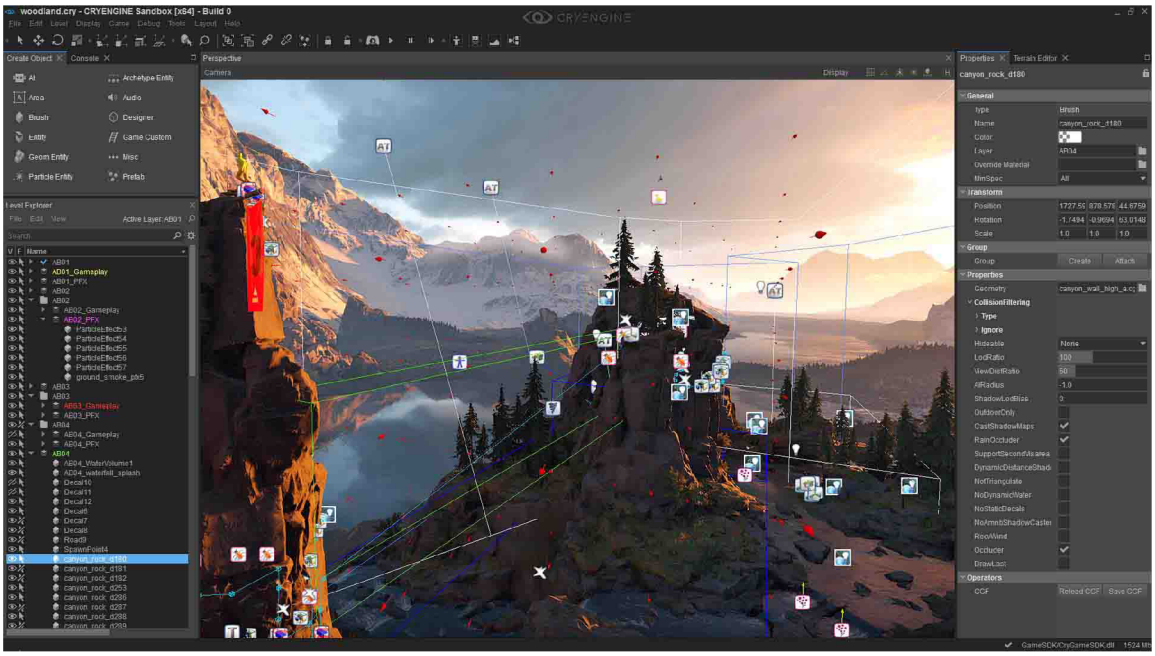
Obrázek 3.3: Unity 3D

Engine. Nejnovější verze obsahuje podporu pro DirectX 12, virtuální zařízení, a rovněž lze nově aplikace programovat programovacím jazykem C#, předchozí verze podporovaly pouze jazyky C++ a Lua [11].

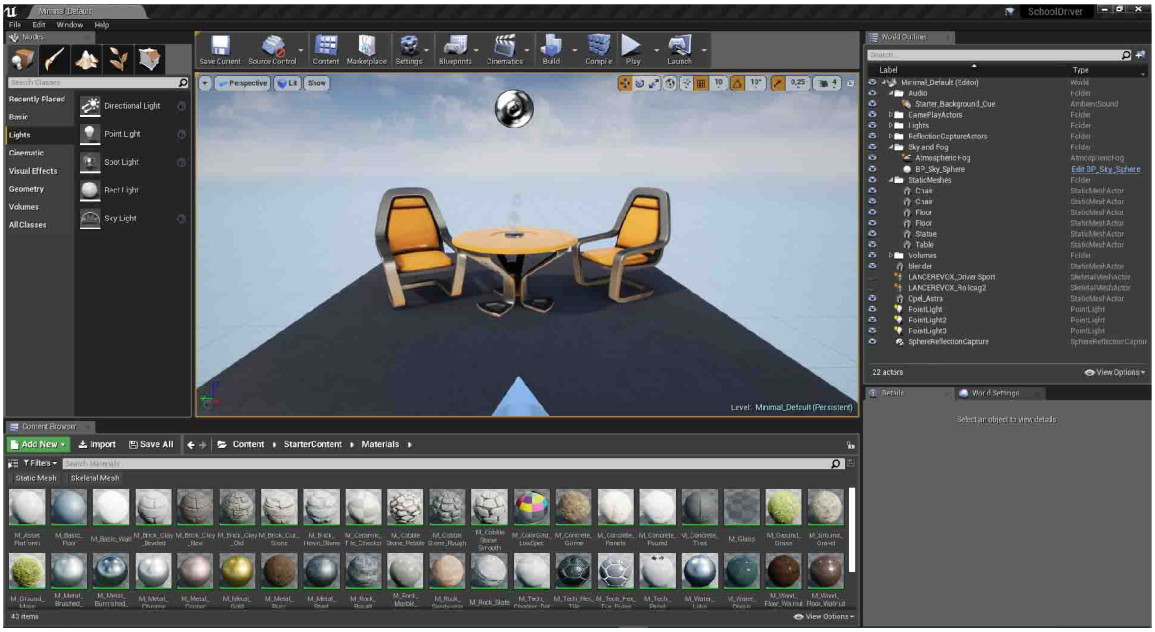
Unreal Engine

Populární herní jádro je vyvíjeno společností Epic Games. Poprvé bylo použito ve hře Unreal roku 1998. Stejně jako CryEngine je vhodný zejména pro velké projekty, a přestože jsou mobilní platformy podporovány, v počtu vyvinutých aplikací na chytrých zařízeních stále vede Unity.

Grafický výsledek je srovnatelný s CryEngine, ale silnou stránkou je lepší uživatelská přívětivost, díky čemuž je vhodnější spíše pro začátečníky. Verze 4 rovněž přichází s bezplatným použitím pro nekomerční aplikace, kdy v komerčních produktech Epic Games požaduje 5% příjem z výnosu.



Obrázek 3.4: CryEngine, zdroj [12]



Obrázek 3.5: Unreal Engine

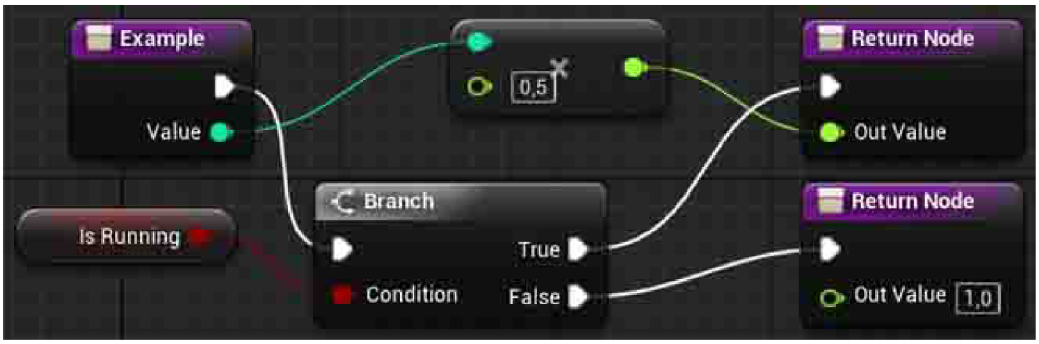
Unreal podporuje programování dvěma ekvivalentními způsoby. Aplikace lze programovat v jazyce C++ a systémem vizuálního skriptování zvaný Blueprint. Tento způsob byl ve 3. verzi nazývaný jako Kismet. Ve 4. verzi je Blueprint více integrován do herního jádra, kde je nejvyšší logika celé scény umístěna v tzv. Level Blueprint. Vizuální skriptování je plnohodnotná alternativa k C++. Funkčnost je tvořena v grafovém editoru propojováním bloků reprezentující funkce, proměnné, vstupy nebo události. Na obrázcích 3.1 a 3.6 je ukázka rozdílu implementace funkce v C++ a Blueprint.

- **C++ varianta**

```
float Example(int value)
{
    if (IsRunning) { return value * 0.5f; }
    else { return 1.0f; }
}
```

Pseudokód 3.1: C++ implementace funkce

- **Blueprint varianta**



Obrázek 3.6: Blueprint implementace funkce

Unreal vyniká propracovaností editoru materiálů, které se podílejí na velmi realistické grafice, kterou je UnrealEngine populární. Navíc má nejnovější verze 4.22 plnou podporu nových grafických karet RTX firmy NVidia, které obsahují dodatečný hardware pro metodu sledování paprsků v reálném čase.

Každá scéna je nazývána Level a je modelována v Unreal editoru. Unreal využívá pro výpočet fyziky knihovnu PhysX 3.2, ale ve vývoji je vlastní fyzikální systém Chaos, který by měla obsahovat verze 4.23 [41].

Požadavky od herního jádra

Při výběru vhodného nástroje k vytvoření simulátoru bylo podmínkou, aby byl bezplatný, což všechny výše uvedené nástroje splňují. Kritériem tedy bylo, aby programovacím jazykem bylo C++ a editor byl uživatelsky přívětivý. Vybrán byl UnrealEngine 4, jelikož Unity nemá podporu C++ a CryEgine byl doporučován jen za předpokladu základních znalostí s vývojem her.

3.4 3D modelovací a zvukové prostředky

V rámci Unreal jsou objekty pouze umisťovány do herního světa a nelze je zde vytvářet. K tomu slouží externí 3D modelovací nástroj. Ze začátku modelování probíhalo v Cinema4D, neboť měla nejintuitivnější uživatelské prostředí. Beta verze Blenderu 2.8 přišla se zásadně upraveným rozhraním oproti verzi 2.79, a přestože je nová verze zatím stále pouze pro testovací účely, již od prvního vydání byla velmi stabilní. Modely v simulátoru byly tedy vytvořeny v nástroji Blender. Ke tvorbě a úpravě textur byl použit bezplatný nástroj Gimp.

Pro práci se zvukem byla použita vestavěná komponenta v Unreal Engine. Pro vytvoření zvuku motoru při různých hodnotách RPM byl využit nástroj FMOD, který poskytuje možnost navázat aktuální zvukovou stopu na vstupní parametry.

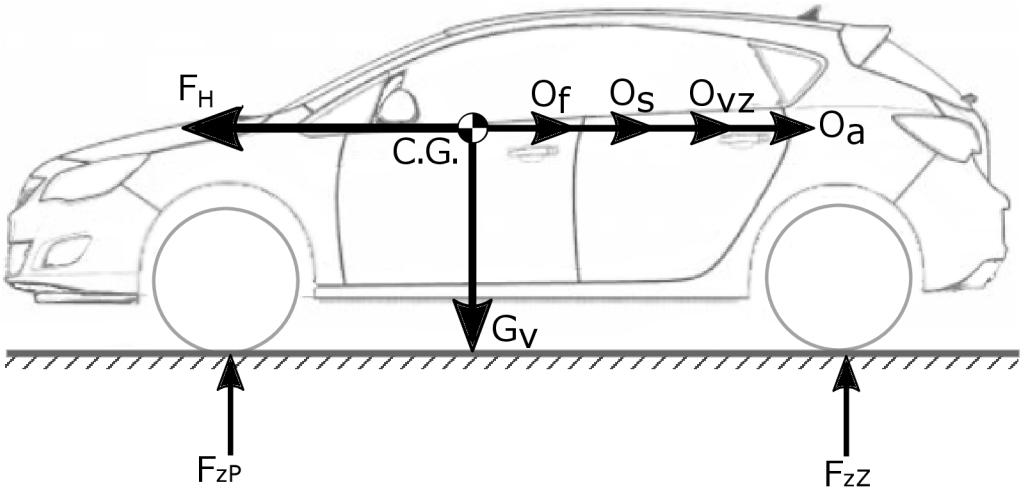
3.5 Simulační rovnice vozidla

Při pohybu vozidla je nezbytné simulovat fyzikální chování automobilu. Jedná se o sofistikovaný problém, neboť vozidlo je složeno z mnoha komponent, které mezi sebou reagují, čímž dochází k přenášení fyzikálních sil.

Na obrázku 3.7 jsou znázorněny základní vnější síly působící na automobil. Rovnovážný stav sil v ose X popisuje rovnice 3.1, kde F_H je hnací síla v podélném směru, O_f je odpor valení, o_s je odpor sklonu, O_{vz} je odpor vzduchu a O_a je odpor zrychlení. Rovnováhu sil ve směru osy Z definuje vztah 3.2, kde F_{zP} resp. F_{zZ} je síla působící na přední resp. zadní nápravu a G_v je tíha vozidla [44].

$$O_f + O_s + O_{vz} + O_a - F_H = 0 \tag{3.1}$$

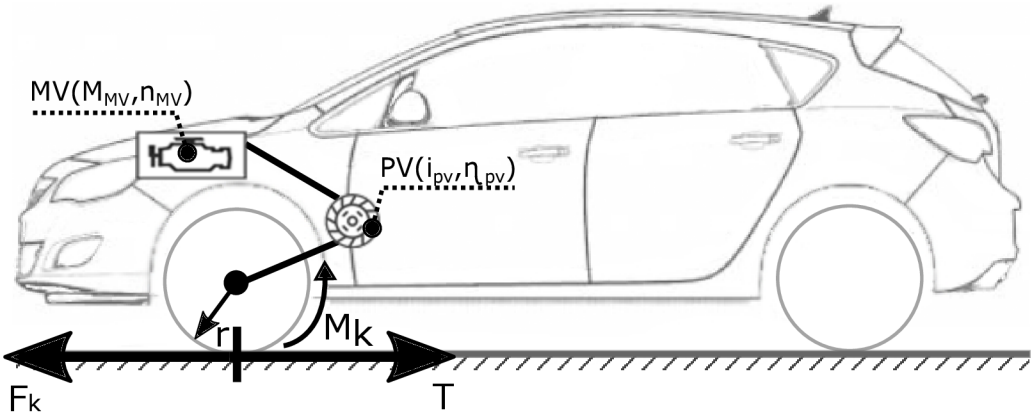
$$F_{zP} + F_{zZ} - G_V = 0 \tag{3.2}$$



Obrázek 3.7: Základní fyzikální síly na vozidle, zdroj [25]

Obrázek 3.8 popisuje strukturální model pohonu. Zdrojem výkonu pro pohyb je motor MV, který je charakterizován kroutícím momentem na výstupu hřídele M_{MV} a otáčkami hřídele n_{MV}. Převodovou soustavou PV, jenž je charakterizována celkovým převodovým

poměrem soustavy i_{PV} a účinností převodového ústrojí η_{PV}. Hnací moment na kole je dán vztahem 3.3, působící síla na kole dle 3.4 a pro celkovou sílu všech kol platí vztah 3.5.



Obrázek 3.8: Strukturální model pohonu, zdroj [25]

$$M_K = M_{MV} \cdot i_{PV} \cdot \eta_{PV} \tag{3.3}$$

$$F_{Ki} = \frac{M_{Ki}}{r} \tag{3.4}$$

$$F_H = F_K = \sum_i F_{Ki} \tag{3.5}$$

Jízdní odpory

Jedná se o síly, které působí proti pohybu vozidla, kdy odpor valení a odpor vzduchu působí vždy proti pohybu vozidla. Odpor sklonu a odpor zrychlení působí jen za určitých podmínek.

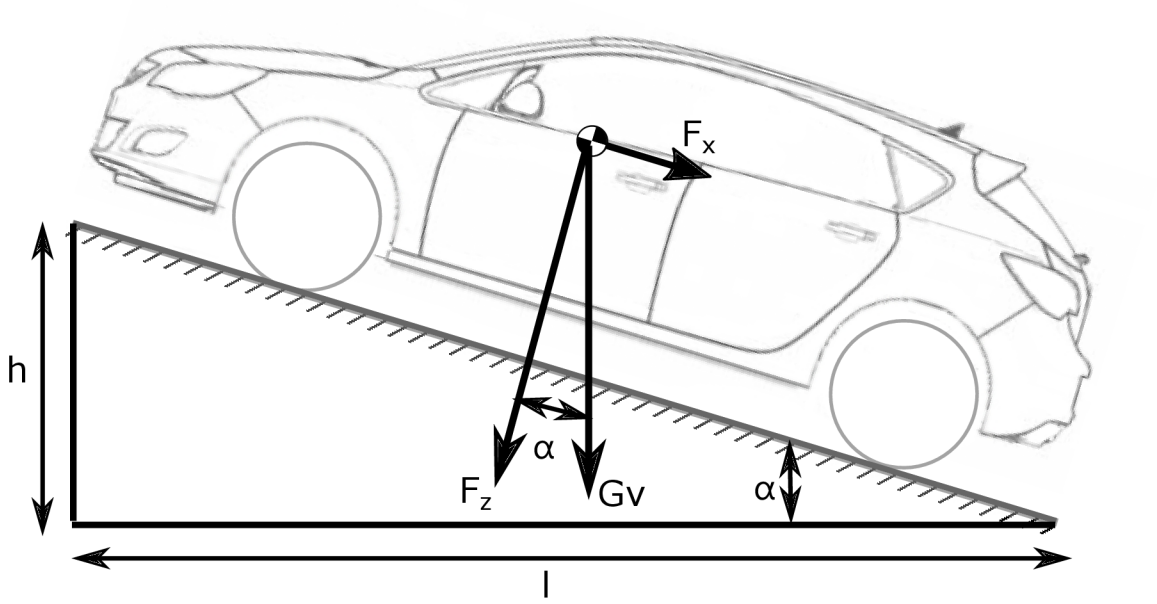
- **Odpor sklonu**

Výpočet odporu O_s vychází ze silového působení na těleso na nakloněné rovině znázorněném na obrázku 3.9. Výškové uspořádání vozovky je charakterizováno převýšením vozovky h vztaženým na délku l jejího průmětu do vodorovné roviny. Tato charakteristika je označována jako sklon s a udává převýšení komunikace v centimetrech na metr její délky [25].

Síla F_x je rovnoběžná se směrem jízdy, kdy při jízdě do kopce působí proti směru pohybu a při jízdě po spádu působí ve směru pohybu. Výpočet sil lze vyjádřit rovnicí 3.6, kde m_v je hmotnost vozidla a g tíhové zrychlení. Celkový odpor je získán podle vztahu 3.7.

$$F_x = G_v \cdot \sin\alpha = m_v \cdot g \cdot \sin\alpha \tag{3.6}$$

$$O_s = G_v \cdot \sin(\arctan\frac{s}{100}) \tag{3.7}$$



Obrázek 3.9: Fyzikální síly působící na nakloněné rovině, zdroj [25]

- **Odpor zrychlení**

Změní-li se rychlost vozidla, začnou na něj působit setrvačné síly, které představují odpor zrychlení, jenž je vypočítán podle vztahu 3.8, kde δ je součinitel vlivu rotujících částí, přičemž u osobního automobilu nabývá hodnot v rozsahu 1.04 až 1.5.

$$O_{zr} = \delta \cdot m_v \cdot a \tag{3.8}$$

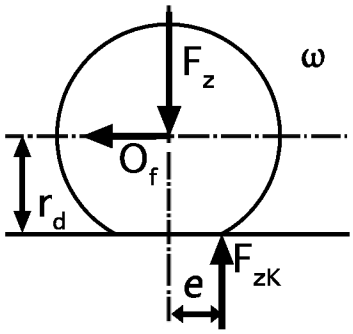
- **Odpor vzduchu**

Na pohybující se vozidlo působí aerodynamické síly, které jsou přímo ovlivněny tvarem, povrchem a rozměrem vozidla. Síly se rovněž mění podle náporové rychlosti vzduchu v_x a fyzikálních vlastností vzduchu. Odpor vzduchu je závislý na dynamickém tlaku p_d, jenž je stanoven na základě Bernouliho rovnice, čelní plochy vozidla S_x a součinitele odporu vzduchu c_x. Výsledná hodnota odporu je získána dle 3.9, kde ρ je hustota vzduchu.

$$O_v = p_d \cdot c_x \cdot S_x = \frac{1}{2}\rho \cdot v_x^2 \cdot c_x \cdot S_x \tag{3.9}$$

- **Odpor valení**

Při pohybu vozidla je v místě styku kol s tuhou podložkou vytvářena stopa, kdy v její přední části dochází ke stlačování pneumatiky a v zadní částí nastává obnovení kruhového tvaru. Při deformaci pláště pneumatiky vzniká odpor valení. Jednotlivé síly působící na kolo jsou znázorněny na obrázku 3.10.



Obrázek 3.10: Fyzikální síly působící na kolo, zdroj [25]

Ve vzniklé stopě je radiální reakce vozovky F_{zK} přesunuta vpřed ve směru jízdy o rameno valení e. Reakce F_{zK} je spolu se silou F_z, která působí na kolo ve svislém směru, v rovnovážném stavu k odpovídajícímu momentu M_k, díky čemuž lze sestavit a po úpravě dostat rovnici 3.10.

$$O_f = F_z \cdot \frac{e}{r_d} \tag{3.10}$$

Sílu F_z lze vyjádřit dle 3.11, čímž po dosazení do rovnice 3.10 je získána rovnice pro výsledný odpor valení 3.12, kde f je součinitel odporu valení, jenž je ovlivněn mnoha faktory. Zásadní je struktura povrchu vozovky, například asfalt má jiný součinitel valení než čerstvý sníh a odpor je rovněž závislý na úrovni nahuštění pneumatik.

$$F_z = G_v \cdot \cos\alpha \tag{3.11}$$

$$O_f = F_z \cdot \frac{e}{r_d} = G_v \cdot \cos\alpha \cdot \frac{e}{r_d} = G_v \cdot \cos\alpha \cdot f \tag{3.12}$$

Pozice vozidla

V každém aktualizačním kroku je nejprve vypočítána hodnota síly F_x podle vzorce 3.1, která je použitá k výpočtu zrychlení vozidla podle druhého Newtonova zákonu, kde m je hmotnost vozidla.

$$a = \frac{F_x}{m} \tag{3.13}$$

Numerickou metodou pro integrování je získána nová hodnota rychlosti vozidla, kde je integrován vztah 3.14 v čase, kde dt je časový rozdíl mezi aktualizačními voláními. Aktuální pozice vozidla je vypočtena následnou integrací rovnice 3.15 [32].

$$v_{new} = v_{old} + a \cdot dt \tag{3.14}$$

$$P_{new} = P_{old} + v \cdot dt \tag{3.15}$$

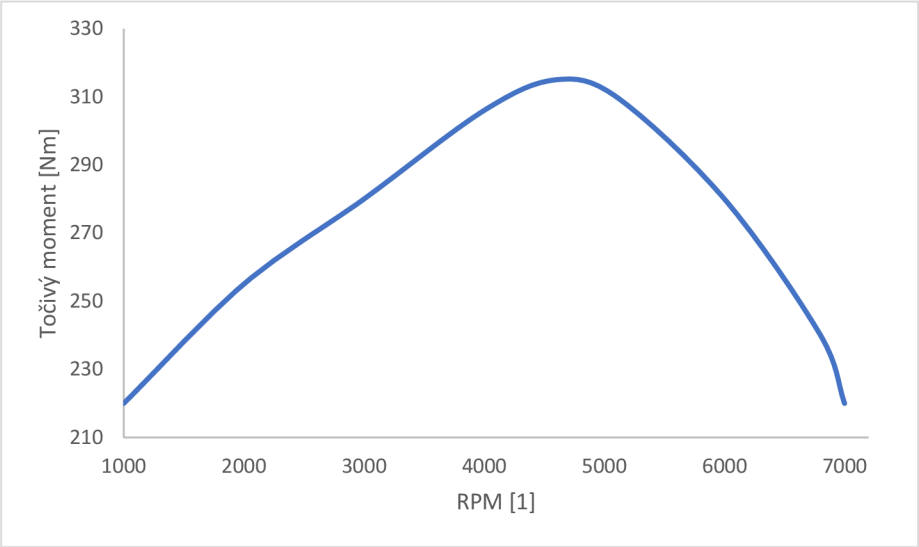
Model pohonné soustavy vozidla

Motor generuje kroutící moment, který je odvozen podle rychlosti otáčení motoru. Rychlost je vyjádřena pomocí RPM udávající počet otáček za minutu. Vztah mezi hodnotou

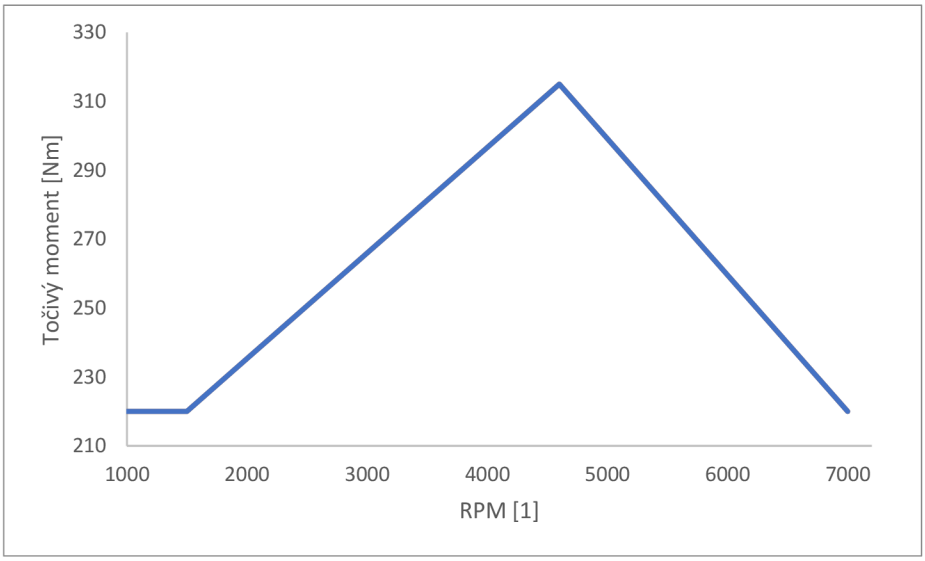
kroutícího momentu a RPM je definován křivkou kroutícího momentu nacházející na obrázku 3.11. Simulace dynamiky motoru ve hrách je pro urychlení výpočtu aproximována přímkami, kde výsledná křivka je uvedena na obrázku 3.12 [3].

Motor má definovaný minimální a maximální provozní rozsah, je-li hodnota RPM pod minimem dochází k vypnutí motoru. Pokud je hodnota RPM nad maximální hodnotou, nachází se motor v tzv. červené zóně a dochází k jeho destrukci.

Aktuální hodnota kroutícího momentu je získána vynásobením maximální hodnoty definované křivkou při daném RPM a mírou stisku pedálu regulující plyn. Míra stlačení nabývá hodnot mezi 0 až 1, kde 0 je definována pro nestisknutý pedál.



Obrázek 3.11: Křivka kroutícího momentu
zdroj [32].



Obrázek 3.12: Aproximovaná křivka kroutícího momentu
zdroj [32].

Kroutící moment vycházející z motoru je před přivedením na kola transformován přes převodovou soustavu a diferenciál vozidla. Konverzní rovnice 3.16 je rozšířením vztahu 3.4, kde u je jednotkový vektor udávající směr vozidla, x_d je poměr diferenciálu a x_g je převodový poměr, který nabývá u nižšího převodu vyšších hodnot, díky čemuž má vozidlo vyšší kroutící moment, ale nižší rychlost.

Každá převodová soustava obsahuje výkonové ztráty, neboť dochází k přeměně určité energie v teplo, z tohoto důvodu se v rovnici nachází konstanta η_{PV}.

$$F_{Ki} = \frac{u \cdot M_{MV} \cdot x_g \cdot x_d \cdot \eta_{PV}}{r} \tag{3.16}$$

Nová hodnota RPM je aktualizována vztahem 3.17, který vychází ze zpětného přepočtu z rychlosti otáčení kol v_k, neboť pokud není sešlápnuta spojka, jedná se o propojený systém. Při aktivní spojce je hodnota odvozena podle množství plynu vedoucího do motoru.

$$RPM = v_k \cdot x_g \cdot x_d \cdot \frac{60}{2\pi} \tag{3.17}$$

Spojka vozidla

Jedná se o zařízení nacházející se mezi motorem a převodovou soustavou. Účelem spojky je rozpojení soustavy, která přenáší kroutící moment od motoru ke kolům, za účelem rozjetí, zastavení nebo změny převodu vozidla.

Kroutící moment M_{MV} v 3.16 je násoben silou stisku spojky, která je definována stejně jako u plynového a brzdového pedálu, kdy hodnoty jsou v rozsahu 0 až 1.

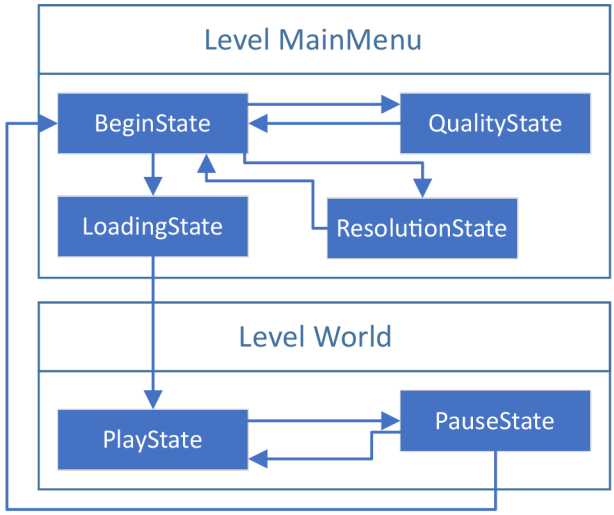
3.6 Architektura simulátoru

V Unreal Enginu jsou všechny objekty nacházející se ve scéně (světla, kamery, statické nebo dynamické modely) umístěny v prvku zvaném Level. Po jeho načtení je aktivován Level Blueprint, který má přímý přístup ke všem objektům a běží až do ukončení aplikace nebo načtení jiné scény. Je tedy vhodné, aby simulátor obsahoval alespoň dva Levely pro herní svět a hlavní menu s možnostmi. V menu není nutné načítat do paměti počítače data, která jsou potřebná až v tréninku.

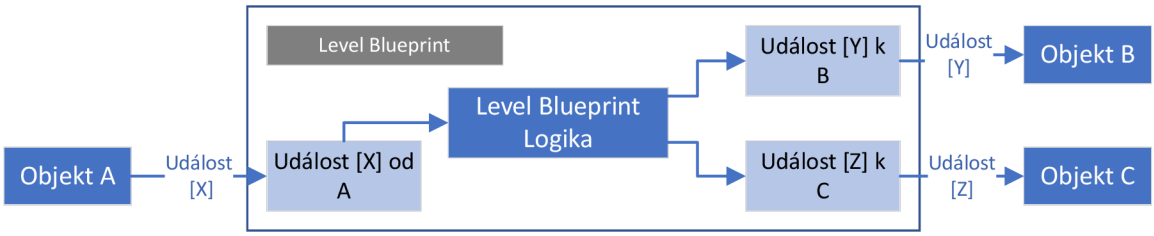
Schéma propojení Levelů je na obrázku 3.13, kde při spuštění aplikace je aktivní Level MainMenu menu s výchozím stavem BeginState, ze kterého se přechází na jednotlivé nabídky v menu nebo spouští trénink přechodem do stavu LoadingState a zahájením načítání levelu World, který obsahuje vymodelované město s okolím. Ve stavu PauseState je pozastaven trénink a zobrazen soupis přestupků. Z PauseState se lze vrátit do PlayState nebo ukončit trénink a přejít zpět do BeginState.

Aplikace propojuje objekty ve scéně dvěma způsoby. Prvky, které interagují pouze mezi sebou navzájem, obsahují přímou referenci. Naopak pokud akce přichází od více objektů, je vhodnější událostní řízení, jelikož nelze reference ukládat, neboť mohou dynamicky vznikat či zanikat. Každému programovatelnému prvku lze definovat tzv. Event Dispatcher, který je zaslán všem elementům, které mu naslouchají. Výsledná architektura propojení je znázorněna na obrázku 3.14, kde objekt A vyvolá Event Dispatcher X. Level blueprint této události naslouchá a podle své logiky vyvolá události k dalším objektům.

Základní architektura pro řízení simulátoru je uvedena na obrázku 3.15. Od připojených periferií přicházejí vstupy a skript řídící vozidlo na ně reaguje. Upravuje parametry využívané při výpočtu pohybu fyzikálním jádrem (spojka, brzda, plyn a natočení volantu)



Obrázek 3.13: Diagram stavů a přechodů mezi Levely



Obrázek 3.14: Události level blueprint [15]

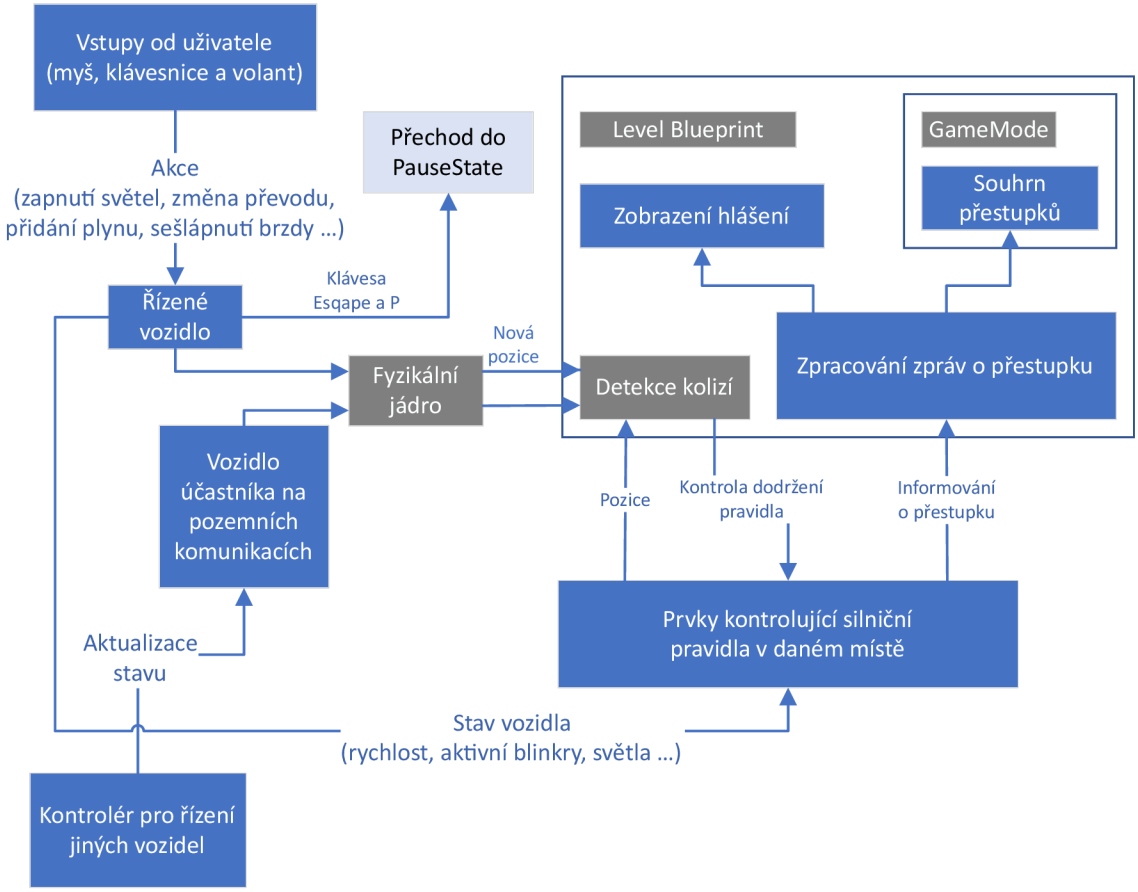
nebo aktivuje osvětlení apod. Fyzikální jádro na základě upravených hodnot vypočte novou pozici vozidla v daném snímku. Podobně jsou řízeny vozidla ostatních účastníků, které jsou řízeny speciálním kontrolérem, který pomocí stavového automatu nastavuje parametry řízení.

V detektoru kolizí je vyhodnocena interakce mezi vozidly a kolizními prvky, které slouží ke kontrole silničních pravidel. Je-li detekována kolize, probíhá vyhodnocení, zda došlo k přestupku. Dojde-li k porušení pravidla je o něm informován level blueprint, který jej vypíše na obrazovku a uloží do objektu GameMode, kde je uchován. Přejde-li uživatel do stavu PauseState jsou přestupky z GameMode použity pro zobrazení přehledu všech přestupků.

3.7 Uživatelské rozhraní

Hlavní menu

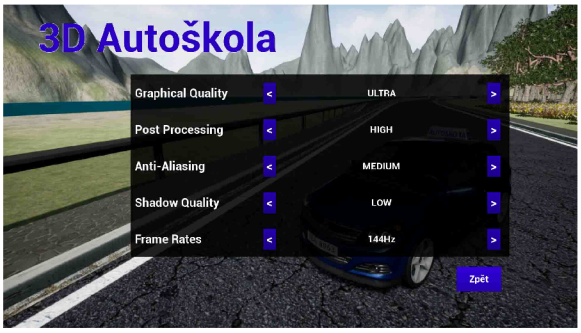
Při zahájení aplikace je zobrazeno nejprve hlavní menu, jedná se tedy o důležitý grafický element, který musí zaujmout a být intuitivní a zajímavý, neboť rozhoduje, zda uživatel bude setrvávat v simulátoru nebo jej opustí. Jako pozadí je zvolen záběr vytvořený ze simulace pro bližší představu, co uživatel může od aplikace očekávat ještě před spuštěním tréninku. Na obrázku 3.16 je výsledná podoba menu, odkud lze spustit jízdu a nastavit kvalitu vykreslování nebo rozlišení obrazovky 3.17. Stiskem tlačítka Spustit, dochází k na-



Obrázek 3.15: Architektura simulátoru

čítání levelu s herním světem. Načítání trvá delší dobu, proto je zobrazena obrazovka se stavem načtení 3.18. Podkladem pro funkční prvky byl po vyhodnocení názorů několika testujících vybrán styl černé barvy s průhledností a barva funkčních prvků je zvolena podle barvy řízeného vozidla.

Obrázek 3.16: Hlavní menu



(a) Nastavení kvality zobrazování

(b) Rozlišení

Obrázek 3.17: Podoba stavů k nastavení aplikace

Obrázek 3.18: Načítání herního světa

Soupis přestupků

Jestliže je trénink pozastaven stiskem klávesy Escape nebo P, přechází aplikace do stavu PauseState, ve kterém lze upravit kvalitu zobrazení, ukončit aplikaci nebo přejít do hlavního menu, například za účelem resetování tréninku. Stav rovněž poskytuje souhrnnou tabulku zaznamenaných přestupků během jízdy s výpočtem jejich ceny, která byla stanovena dle sazebníku pokut České republiky. Ve skutečném světě je poplatek za autohavárii individuální podle rozsahu poškození vozidla. Jelikož aplikace poškození nevyhodnocuje, je cena stanovena podle ceny poškození na vozidle při mém zasažení přecházející srny přes pozemní komunikaci.

Obrázek 3.19: Soupis přestupků

Uživatelské prvky během jízdy

Dojde-li pří jízdě k porušení pravidla, je nutno řidiče informovat o přestupku. V pravém horním rohu se zobrazuje současně až 5 polí s hláškou o přestupku 3.20, které jsou nahrazeny novou. Informace jsou odstraňovány po stanoveném intervalu nebo je-li vyčerpán volný prostor při novém hlášení. Pozadí je rovněž mírně průhledné, ale oproti ostatním prvkům v uživatelském rozhraní má odstín červené.

Obrázek 3.20: Hlášení o přestupku během jízdy

Ve vozidle je hlavním ukazatelem stavu vozidla palubní deska, která se nachází za volantem v místě, na které řidič dobře vidí. Nejdůležitějšími prvky na palubní desce jsou otáčkoměr, který udává aktuální počet otáček motoru, a rychloměr, na němž je zobrazena aktuální rychlost jízdy. Rovněž jsou důležité ukazatele hladiny paliva a teploty motoru, které jsou také ručičkové, ale mívají menší velikost. Na desce se nachází mnoho dalších ukazatelů, které jsou znázorněny určitým grafickým symbolem a jsou pod světleny při akti-

vaci. U moderních vozidel lze na palubní desce nalézt také chybové stavy od externích čidel vyhodnocujících funkčnost různých částí, například pneumatik [40].

V rámci této práce nelze z časových důvodů implementovat simulaci hladiny paliva nebo teploty vozidla, a proto palubní deska v trénovacím vozidle obsahuje pouze otáčkoměr a rychloměr. Signalizace aktivních blinkrů se děje blikáním ikon šipek v daném směru na palubní desce. Symboly jsou v Unreal vytvořeny prvkem Decal, který v případě aktivace využívá emisního materiálu.

Obrázek 3.21: Palubní deska vozidla

Jelikož implementované vozidlo podporuje práci se spojkou, je potřeba obejít problém související s klávesnicí, kdy je rozpoznáno pouze zda je klávesa stisknuta, a nikoliv síla stisku. Nedostatek je řešen ukazateli uvedenými na obrázku 3.22, kde stiskem klávesy dochází k plnění ukazatele ve tvaru obdélníku a aktuální stav naplnění je brán jako vstup pro řízení. Pro lepší přehlednost aktuální rychlosti vozidla a počtu otáček motoru jsou rovněž uvedeny jejich hodnoty.

Obrázek 3.22: Ukazatele pedálů a hodnot

3.8 Ovládání aplikace a jízdy

Na základě testování existujících řešení byl vyhodnocen problém v řízení pomocí klávesnice, neboť je-li kladen požadavek na podporu spojky, je ovládání pomocí kláves značně komplikované. Proto aplikace, kromě ovládání pomocí klávesnice, podporuje použití externího volantu Logitech G920 s pedály, ale kvůli nedostatku vyššího počtu volantů za účelem testování aplikace, nepodporuje jiné volanty než právě Logitech G920. V příloze B se nachází přehled navrženého ovládání.

Kapitola 4

3D modely v simulaci

Následující kapitola popisuje kroky vedoucí ke tvorbě virtuálního města, ve kterém budou trénovány řidičské dovednosti. Kapitola je rozdělena na jednotlivé části nutné k namodelování celého prostředí. Ve všech případech byl použit k úpravě a modelaci těles 3D modelovací nástroj Blender. Použita byla verze nástroje 2.8, která je ve vývoji a aktuálně je stále v beta fázi.

4.1 Vozidlo autoškoly

Z důvodu požadavku na umístění kamery v interiéru vozidla při jízdě je kladen nárok na kvalitu modelu, proto byl zvolen volně dostupný model vozu Opel Astra [9], jelikož vůz je detailně namodelován, včetně interiérových prvků. Stažený model bylo následně nutné kvůli vysokému počtu polygonů manuálně upravit na vhodných místech funkcí unsubdivide. Rovněž byl po stažení pouze ve formátu pro nástroj Cinema4D a musel být nejprve transformován do formátu, který podporuje Blender.

Následně bylo u modelu 4.1 nutné upravit materiálové vlastnosti. Zejména přidat nové, neboť původně model obsahoval jedinou barvu pro celý interiér, ale v aplikaci musí být materiál sedadel jiný než u volantu. Následně byly polygonální sítě komponent rozbaleny do 2D z důvodu UV mapování textur.

Obrázek 4.1: Model Opel Astra

Poslední krok v modelovacím nástroji spočíval v dodání skeletu vozidla, který obsahuje armatury umístěné u prvků pohybujících se v simulaci. Kořenovým uzlem skeletu je arma-

tura Vehicle_Base, která obsahuje 8 dalších armatur, kdy 4 jsou umístěny u kol vozidla a ostatní slouží k pohybu ručiček na přístrojové desce, otáčení volantu a pohybu páčky u přepínání signalizace.

Pro import do Unreal Enginu bylo nutné zajistit správnou pozici vozidla v prostoru, neboť fyzikální jádro v Unreal enginu vyžaduje orientaci vozidla ve směru osy x, osa y musí směřovat doleva a osa z nahoru. Upravený model byl vyexportován do formátu FBX, který jako jediný Unreal podporuje. Po načtení bylo vozidlo obarveno pomocí materiálů, které firma Epic nabízí k volnému použití [13].

Obrázek 4.2: Opel Astra exteriér

Obrázek 4.3: Opel Astra interiér

4.2 Dopravní komunikace

Síť pozemních dopravních komunikací v modelovaném městě je vytvořena s použitím volně dostupné Blueprint implementace Modular Road Tool [10], která umožňuje modelování trasy pomocí spline křivek. MRT je vysoce modulární, jelikož definuje pouze typy různých

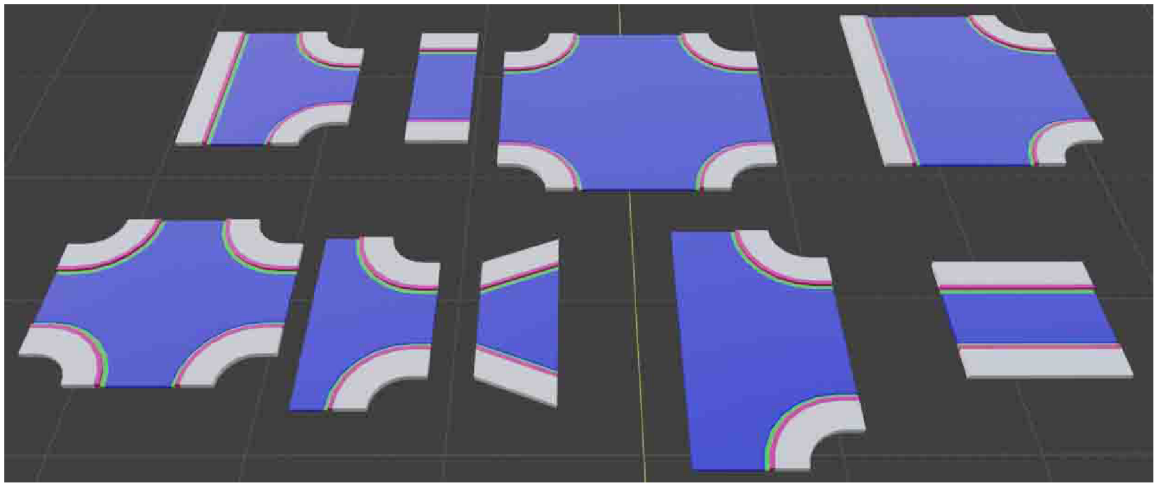
komponent (vozovka, chodník, zábradlí nebo osvětlení), kterými lze sestavit komunikaci. Výhodou nástroje je dynamický výpočet transformací jednotlivých komponent na základě natočení křivky a její aktuální délky, kdy je křivka rozdělena na segmenty, které mají velikost polygonální sítě reprezentující povrch komunikace, a u nového segmentu je měněno měřítko, dokud není vytvořen nový segment. Po umístění MRT do scény se nastaví požadované polygonální sítě a materiály ke všem vyžadovaným komponentám. Nástroj byl modifikován pro vlastní potřeby, zejména přidání materiálového slotu pro obrubník a okraje vozovky.

MRT nabízí sadu modelů cest, ty ale nebyly vhodné ke tvorbě dopravních cest vyskytující se v ČR. Proto jsou vymodelovány nové bloky dopravní komunikace, které jsou na obrázku 4.4. Pro modelaci silnice byl použit zrcadlový modifikátor, díky čemuž byla modelována pouze jedna strana vozovky a opačná je automaticky symetrická.

Křižovatka byla zhotovena rovněž pomocí zrcadlového modifikátoru, kdy bylo zaoblení vytvořeno pouze na jedné straně a přes modifikátor aplikováno na opačné straně. Finální dokončení křižovatek se liší podle počtu křížených cest. Průsečná [6] křižovatka je dokončena použitím druhého zrcadlového modifikátoru, kdy symetricky je umístěna opačná strana. Křižovatka styková vyžadovala manuální dokončení opačné strany s využitím poloviny vymodelované komunikace a propojením dohromady. Pro potřeby MRT je komunikace triangulována a dostatečně podrozdělena na menší polygony a rovněž jsou chodníky odděleny do samostatných objektů, neboť poté lze při tvorbě definovat, který chodník bude použit.

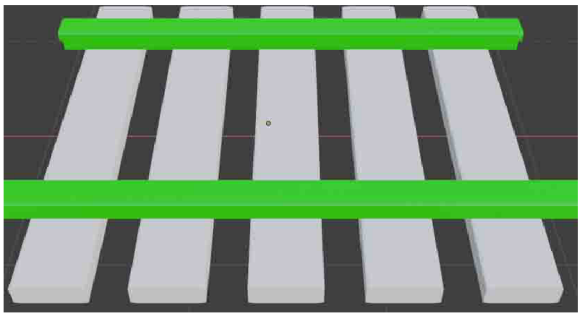
Textury chodníků a asfaltu pocházejí z webových stránek [7]. V Unreal je vytvořen základní materiál MaterialBase s řadou parametrů, které lze nastavit v instancích tohoto materiálu, zejména výběr požadované textury, dobarvení asfaltu nebo výběr textury s vodorovným značením na vozovce. Výsledná podoba po nanesení materiálů v Unreal Engine je na obrázku 4.5.

V modelovaném městě se také nachází železniční sít, která je sestavena z bloku zobrazeného na obrázku 4.6.



Obrázek 4.4: Modulární segment pozemní komunikace

Obrázek 4.5: Výsledek s materiály



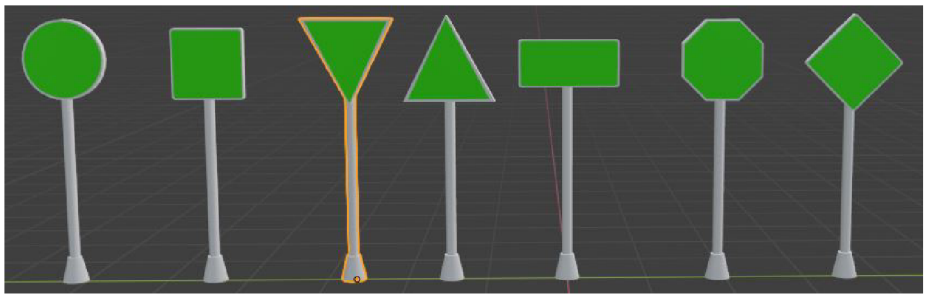
Obrázek 4.6: Vymodelované železniční koleje

4.3 Městské budovy

Budovy ve městě pocházejí z volně dostupných zdrojů [42], [22] nebo [8]. Jednotlivé modely bylo nutné v Blenderu upravit definováním materiálů u jednotlivých částí a rozbalit polygonální síť do 2D k namapování textur, které jsou převzaty z [7], [1] nebo od Epic Games.

4.4 Dopravní značení

Pro svislé dopravní značení byl vymodelován sloup reprezentující držák, na kterém jsou značky umístěny. Několik dopravních značení má vždy shodný tvar, proto je jeden model 4.8 rozdělen pouze aplikací jiného materiálu se správně namapovanou texturou, které byly vytvořeny v nástroji Gimp. Podklad byl převzat ze zdroje [45] a namapován vzhledem k UV souřadnicím modelu. Různé rychlostní limity byly upravovány rovněž v Gimpu. Má-li daný tvar značení více textur, je k ní přidružen blueprint skript, který na základě vybraného typu zvolí požadovanou texturu.



Obrázek 4.7: Modely svislého dopravního značení

Obrázek 4.8: Dopravní značení s texturou

4.5 Světelné semafory

Modely semaforů a skript pro jejich řízení pocházejí z volně dostupného assetu [31]. Skript bylo nutné modifikovat, jelikož nefungoval pro potřeby křižovatky v aplikaci. Bylo nutné zajistit souběžnost střídání zeleného signálu s červeným signálem na různých stranách křižovatky. Rovněž skript neobsahoval podporu pro semafor u železničního přejezdu, který byl vymodelován 4.9.

Obrázek 4.9: Semafor železničního přejezdu

Kapitola 5

Implementace simulátoru

Následující kapitola popisuje praktickou část práce, spočívající ve vytvoření aplikace v Unreal Engine. Nejprve je zmíněn postup nutný ke zprovoznění herního jádra a vytvoření nového projektu v něm. Poté jsou uvedeny kroky vedoucí ke zhotovení města, kde probíhá jízda. Další dvě podkapitoly se věnují implementaci vozidel a poslední kapitola je věnována kontrole dodržování silničních pravidel.

5.1 Vytvoření a nastavení projektu

Instalace nástroje Unreal Engine se provádí stažením požadované verze herního jádra v aplikaci Epic Games Launcher. V období zahájení této práce byla použita nejnovější verze 4.21. V průběhu tvorby byl pořízen externí volant s pedály, ale jelikož plugin ke komunikaci se zařízením není kompatibilní s verzí 4.21, bylo nutné přenést vývoj na verzi 4.20. Naneštěstí Unreal dovoluje přechod vývoje pouze na vyšší verzi, nikoli naopak, proto musely být vytvořené a získané prvky ručně přenášeny a upravovány pro verzi 4.20.

Projekt byl vytvořen podle šablony Basic Code, která podporuje vývoj v C++. Ve vytvořeném projektu bylo nutné aktivovat plugin LogitechSDK pro komunikaci s externím volantem a plugin FMOD, který poskytuje pokročilejší ovládání zvukových stop. Rovněž bylo třeba přidat do souboru DrivingSchool.Build.cs (dle kterého je generován Visual Studio projekt) reference na zabudované moduly PhysXVehicles (simulování jízdy vozidel) a AIModule (podpora inteligentního řízení ostatních postav).

Struktura projektu

Projekt obsahuje v kořenovém adresáři složky Content a Source. Vedle nich se nachází soubor DrivingSchool.uproject, sloužící k otevření projektu v Unreal editoru a provedení akce v kontextovém menu vyvolaném stiskem pravého tlačítka na soubor. Lze tak spustit hru bez editoru a vygenerovat/aktualizovat projekt ve Visual studiu pro kompilaci c++ souborů. Všechny hlavičkové a zdrojové soubory se nachází ve složce Source.

Složka Content obsahuje všechny importované textury, polygonální sítě, animace nebo zvuky, kde v podsložce FMOD je uložen zvuk motoru, jenž je vytvořen v nástroji FMOD. Rovněž se ve složce nacházejí všechny ostatní prvky vytvořené v editoru. V podsložce Buildings jsou umístěny budovy použité ke tvorbě města, složka Cars obsahuje modely vozidel a komponenty pro inteligentní řízení ostatních vozidel nebo automobilu autoškoly, v Maps jsou umístěny levely s pomocnými soubory pro herní svět a hlavní menu. Materiály použité

v herním světě jsou v adresáři Material, komponenty kontrolující pravidla silničního provozu jsou obsaženy v Rules a UI obsahuje definici vzhledu a chování uživatelského rozhraní.

5.2 Princip tvorby v Unreal Engine

Vývoj v Unreal oproti Unity probíhá mírně odlišně, jelikož projekt v jazyce C++ je potřeba kompilátorem přeložit do knihovny dll, která je načítána editorem nebo samotnou hrou. Z tohoto důvodu je nezbytný kompilační program. Zvolen byl nástroj Visual studio s rozsáhlou podporou pro Unreal.

Velkou nevýhodou oproti Unity je problematické hledání chyb v kódu C++, neboť nastane-li neoprávněný přístup do paměti v dll, dojde k chybovému stavu s ukončením editoru. Jeho opětovné spuštění trvá určitou dobu, přičemž pád nastane okamžitě po dokončení kompilace, neboť Unreal Editor funkcí Hot Reload načítá dll ve chvíli, kdy je změněno, přičemž často chyba při vývoji nesouvisela s implementovanými funkcemi, ale nesprávným použitím meta specifikátorů sloužících k definování charakteristiky funkce nebo třídy v rámci herního jádra.

Naproti tomu implementace pomocí Blueprint využívá stejného principu jako C#, kdy funkce v Blueprintu jsou zkompilovány do mezikódu a teprve za běhu aplikace interpretovány [14]. Nedochází tedy k pádu Editoru, jelikož je chyba odchycena, dříve než dojde k narušení paměti, a vypsána do logu. Díky uvedenému principu je vývoj v Blueprint mnohem rychlejší, ale efektivita využití výpočetního výkonu je nižší.

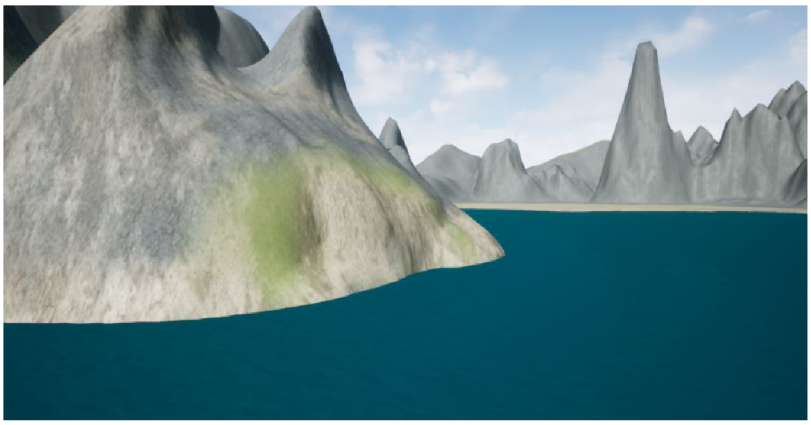
Řada prvků je implementována právě v Blueprint, ovšem C++ k němu nemá přímý přístup. Unreal tento nedostatek řeší způsobem, kdy se v C++ vytvoří abstraktní třída obsahující pouze deklaraci metod. Ze strany C++ jsou již funkce známy, jen se definice nachází až v Blueprint.

Jak bylo zmíněno v části 3.6, tvorba v Unreal Engine z větší části spočívá v naslouchání a reakci na určitou událost. Každý programovatelný prvek již implicitně obsahuje dvě základní události. BeginPlay je vyvolán při aktivaci levelu. Událost Tick je volána s nastaveným časovým intervalem, kde nejmenší časový interval je vázán k rychlosti výpočtu jednoho snímku.

5.3 Vytvoření herního města

Pro potřeby tréninku jízdy je vytvořeno fiktivní město Rasnarov, kdy na začátku byl v levelu World vytvořen prvek Landscape reprezentující povrch herního světa. Hory a jezero v okolí města jsou vytvořeny v editoru nástrojem Sculpt, kterým je modifikována výška vytvořeného landscape v požadovaném místě. Vodní hladina je vytvořena přidáním roviny pokryté vodním materiálem. Hory byly přidány kvůli zabránění výhledu na konec povrchu a vyjetí z něj. K blokování přístupu za vytvořený terén rovněž slouží v Unreal prvek Blocking Volume.

Při budování města byla nejprve vytvořena silniční síť, kdy bylo potřeba zajistit dostatečný počet situací, kde lze kontrolovat navržená pravidla popsané v kapitole 2. Železniční síť je vytvořena podobně jako silniční, jen využívá jiného 3D modelu. U vytvořené infrastruktury byly rozvrženy hlavní a vedlejší komunikace a umístěny značky potřebné pro daný úsek komunikace. Následně k silnicím byly přidány budovy s několika různými texturami. Celková scéna byla nakonec osazena přírodními prvky v podobě stromů a kamení. Na obrázku 5.2 se nachází vymodelované město Rasnarov.



Obrázek 5.1: Jezero a hory v okolí města

Obrázek 5.2: Město pro trénování jízdy

5.4 Implementace vozidla v simulátoru

Následující sekce se zabývá implementací trénovacího vozidla, které uživatel řídí.

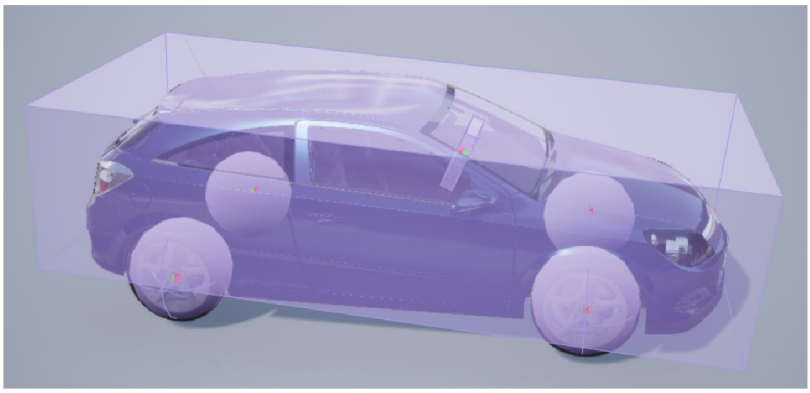
Základní fyzikální model

Po dokončení importu modelu, který obsahuje armatury, Unreal vytvoří tři soubory definující jeho charakter. Soubor typu Skeletal Mesh obsahuje polygonální síť a přehled všech

materiálů použitých v modelu, kde obarvení je provedeno přiřazením požadovaného materiálu do materiálového slotu. V souboru typu Skeleton se nacházejí armatury uvedené v části 4.1. Na základě tohoto typu je vygenerován soubor Physics Assets, jenž obsahuje fyzikální charakteristiky.

Pro funkční chování dle fyzikálních zákonů, je nutné k armatuře Vehicle_Base a jednotlivým kolům dodat komponenty typu Physics Bodies. Zobrazeny jsou na obrázku 5.3, kde kvádr je použit k pokrytí kabiny vozidla, a kola jsou obalena koulí. Unreal umožňuje vygenerovat pokrytí na základě samotné polygonální sítě, ale fyzikální výpočty jsou poté náročnější.

Physics Bodies definuje vlastnosti jako hmotnost vozidla, odpor vzduchu nebo působení gravitace. Neobsahuje-li model Physics Bodies, neprovádí fyzikální jádro žádné výpočty fyziky na modelu.



Obrázek 5.3: Komponenty Physics Bodies umístěné na vozidle.

Skript řídící vozidlo autoškoly

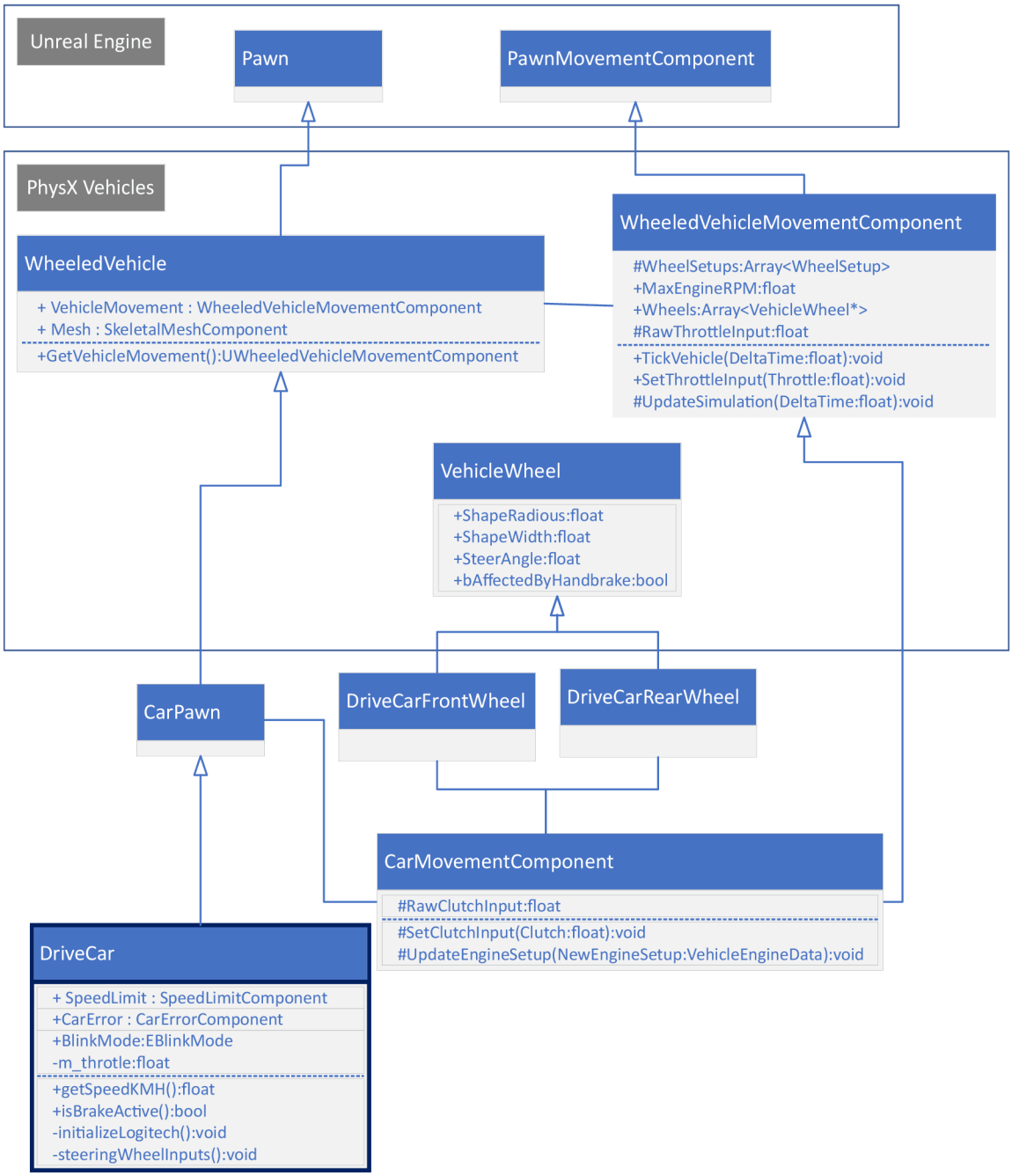
Hlavní funkcionalita pro ovládání řízeného vozidla je napsána v jazyce C++, neboť se jedná o nejdůležitější prvek v aplikaci, kdy je potřeba zajistit nejrychlejší reakce na vstupy od uživatele. Zásadním kritériem pro výběr C++ je ale neexistence Blueprint funkcí v pluginu LogitechG, který je požit ke komunikaci s externím volantem.

Fyzikální pohyb vozidla

Unreal Engine do verze 4.14 obsahoval podporu vozidel přímo v samotném herním jádře. Epic Games tuto podporu ve verzi 4.15 [17] z jádra odstranil a přemístil do externího vestavěného modulu [16]. V modulu PhysxVehicles se nachází tři pod-moduly, kde první slouží k propojení 3rdParty knihovny Vehicle z PhysX, druhý modul přidává funkcionalitu pro Unreal Engine editor a třetí, nejdůležitější, obsahuje třídy, které jsou uvedené na diagramu 5.4.

Třída WheeledVehicle je odvozena od třídy Pawn, která reprezentuje vozidlo nacházející se ve scéně. Obsahuje komponentu odvozenou od bázové třídy WheeledVehicleMovementComponent, ve které jsou deklarovány základní proměnné a metody pro řízení motoru. K řízení čtyřkolového vozidla slouží třída WheeledVehicleMovementComponent4W, kde jsou volány PhysX funkce, ale jelikož stávající implementace nepodporuje práci se spojkou, bylo nutné aplikovat princip dovolující vlastní implementaci, který je uvedený na diagramu 5.4.

Třída CarPawn je poděděná od třídy WheeledVehicle. V konstruktoru třídy je předefinováno vytváření komponenty VehicleMovement využitím vlastní třídy CarMovementComponent vycházející z bázové třídy WheeledVehicleMovementComponent. Uvedeným postupem je umožněna vlastní implementace části kódu využívající Physx pro doplnění podpory spojky.



Obrázek 5.4: Diagram propojení s PhysXVehicles

Hlavní část řízení vozidla je implementována ve skriptu DriveCar, kde jsou v konstruktoru inicializovány kontrolní komponenty SpeedLimit a CarError, je načten modul Logi-

techG zprostředkovávající komunikaci s externím volantem a nastaveno pole WheelSetups. V tomto poli jedna položka reprezentuje jedno kolo na vozidle, kdy při nastavení je kolo propojeno s armaturou dané části na modelu. Kolo je reprezentováno jako komponenta třídy DriveCarRearWheel, resp. DriveCarRearWheel, jedná-li o zadní kolo, resp. přední kolo. Třídy jsou odvozeny od třídy VehicleWheel z modulu PhysXVehicles, která definuje rozměr, ovlivnění ruční brzdou nebo maximální úhel při natočení volantu. Na základě skutečného automobilu DriveCarRearWheel nedovoluje otáčení kol, ale je povolena ruční brzda, DriveCarFrontWheel naopak dovoluje otáčení a je zakázána ruční brzda.

V metodě SetupPlayerInputComponent jsou funkcemi BindAxis a BindAction z komponenty PlayerInputComponent definováni delegáti reagující na příchozí vstupy od uživatele, kde daný vstup je reprezentován názvem akce definované v nastavení projektu. Unreal vstupy rozděluje na Axis a Action, kde Axis definuje, jakou hodnotu daný vstup reprezentuje a Action, zda-li byla klávesa stisknuta. Axis je vhodné použít pro natáčení volantu, kde vstup pro natočení doleva vrací -1 a doprava 1. Action zase pro aktivaci světel nebo klaksonu.

Vstupy z externího volantu jsou získány metodami z modulu LogitechG. Jde především o funkci WheelButtonTriggered, kde vstupem je identifikace tlačítka na volantu a je vrácena hodnota true, nastala-li na něm změna. Pro identifikaci nastaveného stupně převodu na řadicí páce slouží funkce WheelButtonIsPressed. Hodnoty z pedálů a natočení volantu jsou získány v metodě GetControlValue, kde je funkcí WheelGetState z LogitechG obdržena struktura DIJOYSTATE obsahující aktuální stav externího zařízení. Získané hodnoty je potřeba převést do rozsahů vyžadovaných komponentou typu WheeledVehicleMovementComponent, neboť hodnoty od periferního zařízení jsou celočíselné a komponenta vyžaduje rozsah 0 až 1, resp. -1 až 1 u pedálů, resp. volantu.

Dojde-li k sešlápnutí pedálů pomocí klávesnice, jsou při trvajícím stisku inkrementovány proměnné m_throttle, m_brake nebo m_clutch o určitou hodnotu, čímž se postupně plní indikátor síly stlačení. Naopak není-li stisknuta, dochází k dekrementaci, ale na základě poznatků z testování je dekrementace pomalejší.

Při každém volání události Tick jsou funkcemi SetThrottleInput, SetSteeringInput, SetBrakeInput, SetClutchInput a SetTargetGear z komponenty CarMovementComponent aktualizovány pohybové parametry. Komponenta v metodě UpdateSimulation provede výpočet pohybu vozidla funkcemi z PhysX.

Finalizace vozidla

Každé vozidlo obsahuje zrcátka poskytující pohled do širšího okolí vedle vozidla. V aplikaci jsou dodány v Blueprint skriptu DriveCarBlueprint, který je poděděný od třídy DriveCar popsané v předchozí části. V rámci Blueprintu jsou dodatečné prvky umisťovány v prostoru za pomoci grafického editoru.

Odraz zrcátek je vytvořen kamerou SceneCaptureComponent2D, která renderuje vyžadovaný pohled do textury RenderTarget, a materiál používající získanou texturu je přiřazen materiálovému slotu daného zrcátka. Vykreslování zrcátek silně ovlivňuje rychlost výpočtu jednoho snímku, proto nejsou textury aktualizovány v každé události Tick, ale s časovým intervalem 0,05 sekundy.

Přepínání mezi pohledem 5.6 uvnitř kabiny a zvenku je provedeno umístěním dvou komponent typu Kamera. Venkovní kamera je umístěna na rameni směřující ve směru k modelu vozidla. Vždy je aktivní právě jedna z nich, kdy stiskem patřičné klávesy dochází k výměně aktivity.

Obrázek 5.5: Zpětná zrcátka na vozidle

(a) Z kabiny vozidla (b) Vnější pohled

Obrázek 5.6: Režimy pohledů

Je-li stisknuta klávesa pro ovládající světla, jsou aktivovány nebo deaktivovány čtyři přední světlomety svítící bíle a dva zadní reflektory v podobě obdélníkového světla svítící červeně. Kontrola pravidla definující nutné zapnutí osvětlení probíhá, pokud vůz jede více než 5 $km.h^{-1}$. Signalizační osvětlení je vytvořeno podobně, pouze je odstín světel oranžový a svítí s vyšší intenzitou.

V aplikaci je zabudováno několik zvuků, které jsou vyluzovány z vozidla. Při nastartování tréninku je aktivován zvuk symbolizující nastupování a zavření dveří automobilu. Při stisku klávesy pro nastartování, resp. vypnutí motoru, je aktivován zvuk jeho náběhu, resp. vypínání. Během jízdy je produkován zvuk závislý na aktuální hodnotě RPM, kde pomocí pluginu FMOD je zvuk parametrizován právě touto hodnotou. Je-li aktivována signalizace vlevo nebo vpravo, je vyluzován její zvuk a rovněž troubení při stisku klaksonu.

Na obrázku 5.7 je vidět kolizní box, pomocí kterého je detekována autohavárie s ostatními vozidly. Jakmile dojde k průniku s vozidlem je informován Level Blueprint o přestupku týkající se této události.

Umístění vozidla do scény je provedeno definováním startovací pozice pomocí speciálního prvku a rovněž nastavením DriveCarBlueprint jako výchozí Pawn třídy v Blueprintu CarDrivingSchoolModeBP, který je poděděný od objektu GameMode uchovávajícím globální hodnoty daného levelu.

Obrázek 5.7: Přidání ostatních prvků na vozidle

5.5 Interakce ostatních účastníků provozu

Součástí zadání práce je reprezentace pohybu účastníků provozu a jejich vzájemné interakce. V aplikaci jsou těmito účastníky vozidla pohybující se po definovaných trasách a reagující na dopravní pravidla. V následujících podkapitolách je popsána jejich implementace.

- **Základní princip inteligentního řízení**

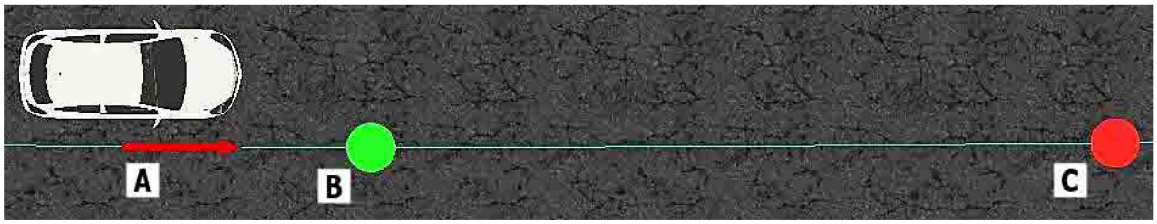
V Unreal Engine existují dva základní typy objektů Actor a Pawn, které se nachází ve scéně. Actor má předem definované chování a není externě řízen, naopak Pawn poskytuje přiřazení komponenty Controller, která jej řídí. Vozidlo reagující na vstupy uživatele řídí PlayerController a vozidla ostatních účastníků AIController.

Jakmile je AIConteroler aktivován, inicializuje komponentu Blackboard a spouští Behavior Tree. Blackboard uchovává aktuální stav řízeného vozidla, například maximální povolenou rychlost v daném místě, referenci na překážku před vozidlem nebo křižovatku. Behavior Tree slouží k rozhodování o akcích, které se mají vykonat.

- **Trasa vozidel**

Cestu vozidel po městě je možno implementovat využitím prvku Checkpoint, který se umístí do scény a následně vozidlo směruje k nejbližšímu bodu na dané trase. Ve vytvořeném městě, ale nejsou silnice rovné, a jelikož nelze body dostatečně definovat zaoblení trasy, docházelo k vjíždění vozidla do vedlejšího jízdního pruhu nebo na chodník.

Z tohoto důvodu je vytvořen prvek AutomaticCarPath, kdy je po jeho umístění do scény trasa vytyčena spline křivkou. Propojení vozidla s trasou je provedeno přidáním reference danému vozidlu. Každé inteligentně řízené vozidlo je odvozeno od Blueprintu CarAIBaseBP, kde jsou v události BeginPlay a voláním funkce AddCar z AutomaticCarPath přidány pomocné prvky 5.8 pohybující se společně s vozidlem.

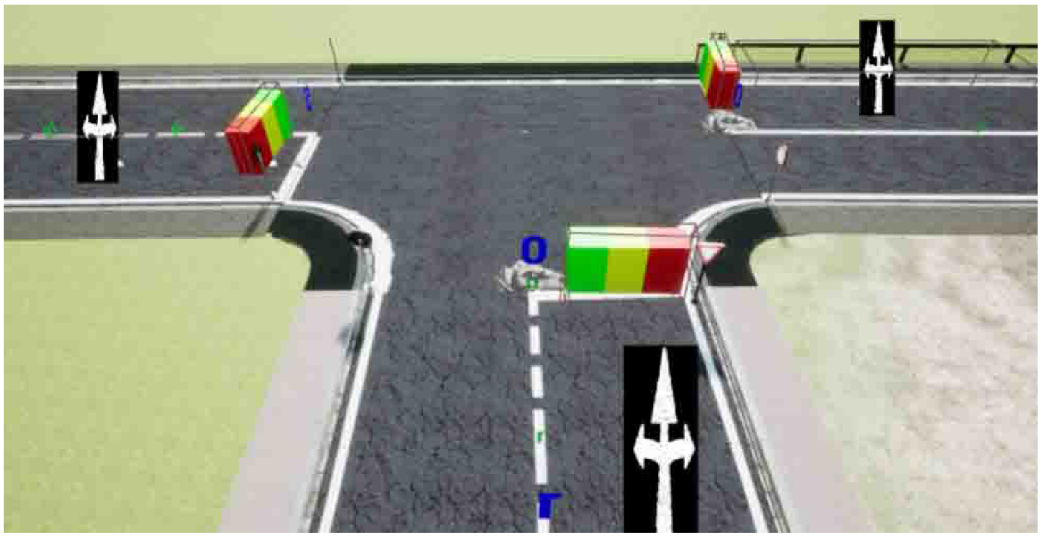


Obrázek 5.8: Pohyb vozidla po křivce

Prvky znázorněné červenou barvou slouží k vypočtení míry působení na plynový pedál vozidla a zelené k natočení volantu. Prvek A je kvůli lepší vizualizaci zvětšen, ale ve skutečnosti se nachází ve středu vozidla. Pomocí něj je vypočítána míra působení na brzdový pedál při zatáčení. Pokud se vozidlo vzdálí od prvku A, jsou aktivovány brzdy ke snížení rychlosti za účelem navrácení středu vozidla na trasu. Prvek C slouží k rozpoznání zatáček před vozidlem. Zvětší-li se úhel mezi vektorem směřujícím ve směru pohybu vozidla a vektorem k prvku C nad zadanou mez, je nastavena síla plynu na 0. Prvek B se nachází 2 metry před vozidlem a vypočítá se rotace k vozidlu, která je převedena do rozsahu -1 až 1, sloužící jako hodnoty k natočení volantu.

- **Interakce v prostředí**

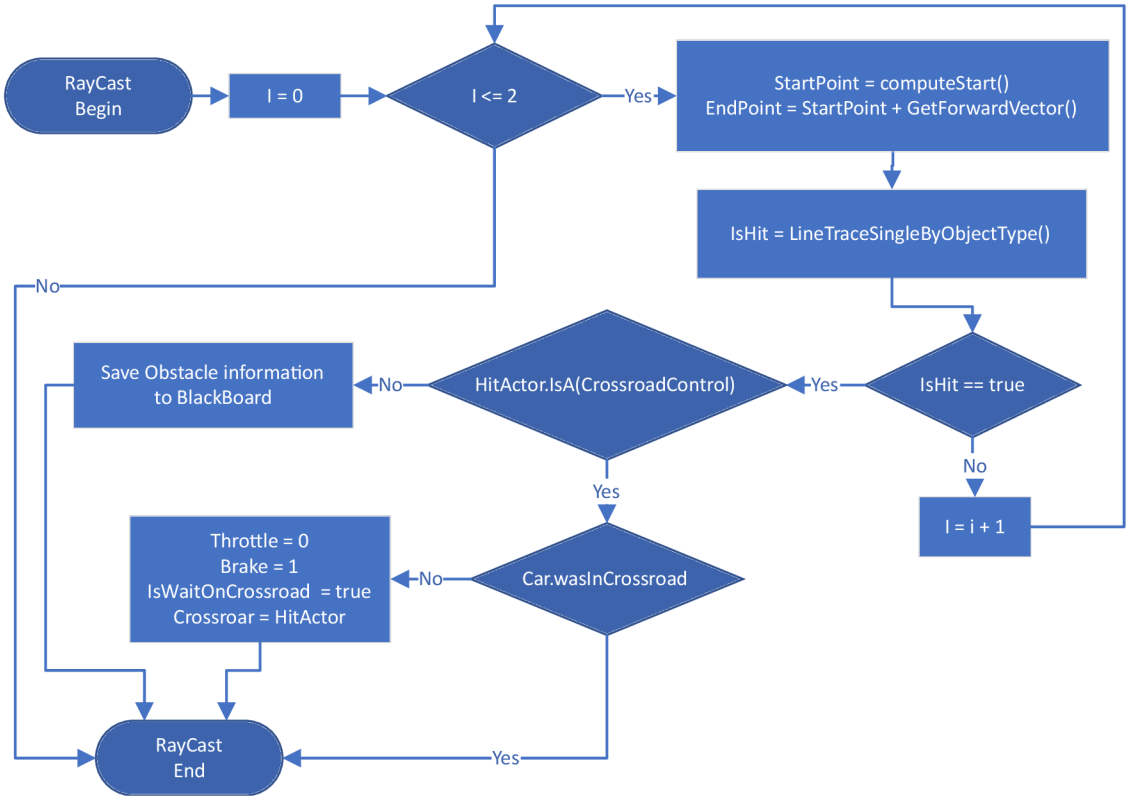
Na obrázku 5.9 jsou zobrazeny kontrolní prvky pro průjezd křižovatkou. Na hranicích je umístěn objekt CrossroadControl, který definuje řízení provozu (dopravní značení, přednost zprava nebo světelnými signály) a v proměnné LightSignal je uložena reference na související semafor. CrossroadControl obsahuje asociativní pole Priorities, kde klíčem je směr jízdy a hodnotou je reference na prvek třídy BeforeCrossroadCheck definující přednost. Objekt se vždy nachází před hranicí křižovatky a vytyčuje prostor k vyhodnocení počtu stojících vozidel metodou GetOverlappingActors.



Obrázek 5.9: Kontrolní prvky křižovatky

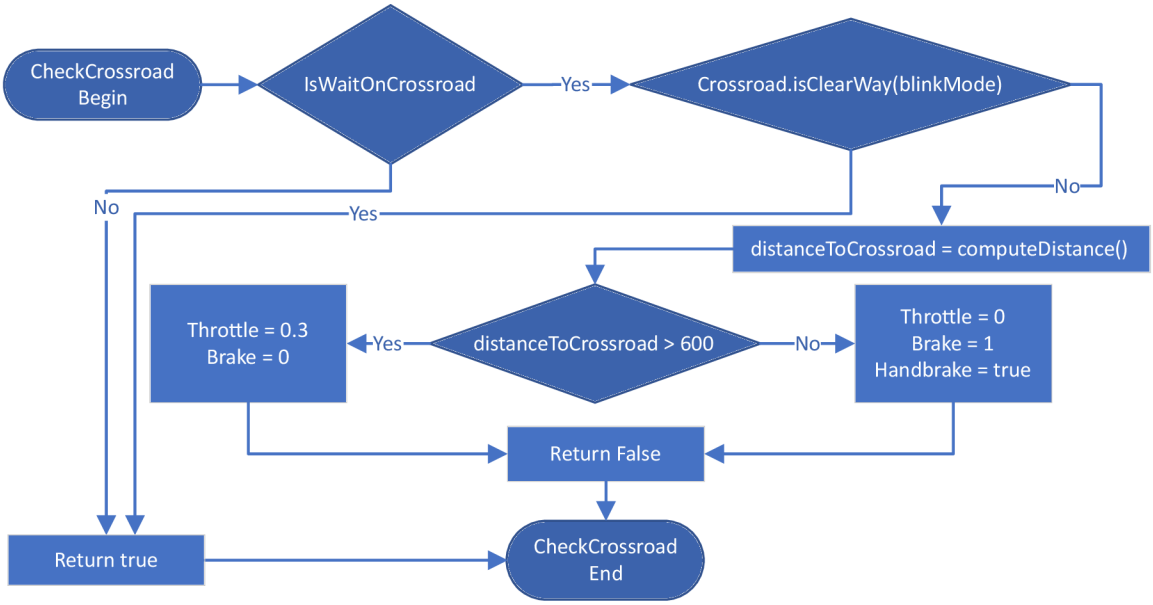
Vozidla ostatních účastníků jsou řízena objektem CarAIController. Metoda Possess v CarAIController inicializuje komponentu typu Blackboard a spouští Behavior Tree. Do Blackboardu jsou ukládány boolean proměnné (požadavek na zastavení, čekání na křižovatce nebo nastavení, zda je cesta volná), dále hodnoty (aktuální maximální rychlost v daném úseku, vypočtená míra působení na plynový nebo brzdový pedál nebo míra otočení volantu) a referenci na překážku nebo křižovatku před vozidlem.

Behavior Tree každou sekundu aktivuje službu ThrottleService, která v Blackboardu aktualizuje uložené hodnoty. Základní funkcionalita je znázorněna na diagramu 5.10. V každém volání události Tick je funkcí CheckCrossroad vyhodnoceno, zda vozidlo smí vjet do křižovatky. Pokud vjezd není povolen, čeká dále. V opačném případě metoda RayCast detekuje nové překážky nebo křižovatky před vozidlem.



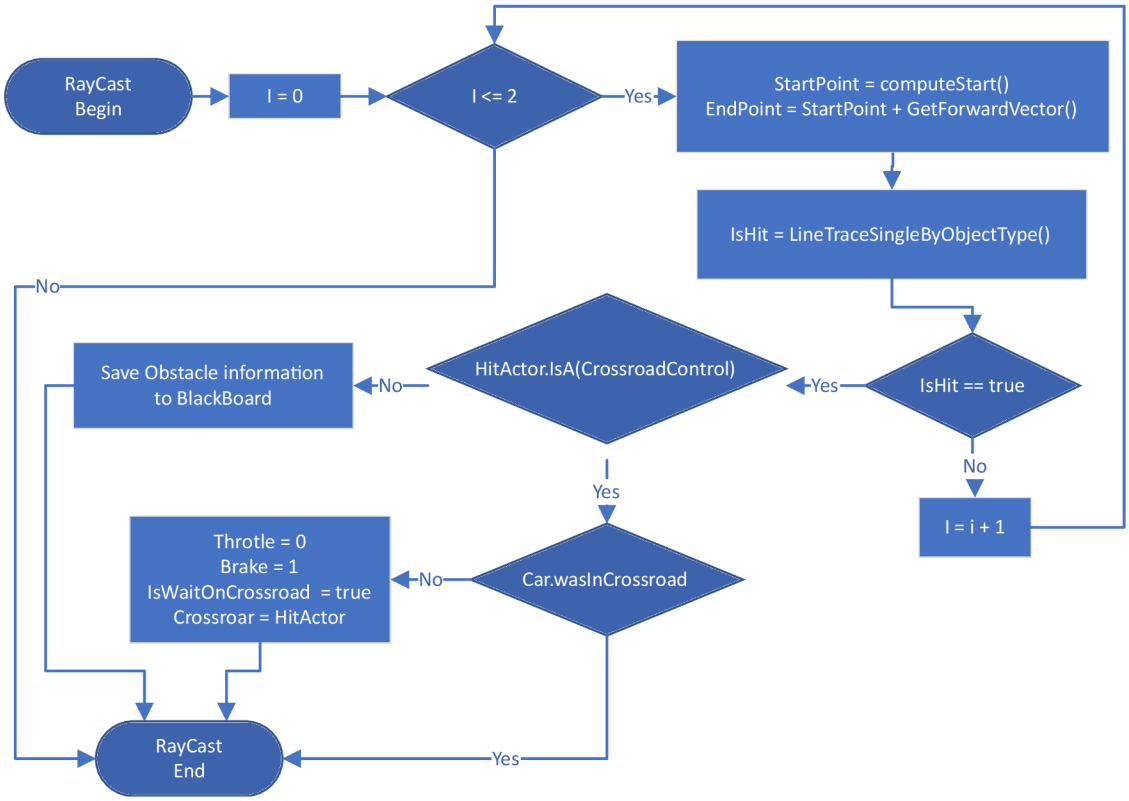
Obrázek 5.10: Vývojový diagram Throttle služby

Metoda CheckCrossoad uvedená na diagramu 5.11 v BlackBoardu ověří, zda vozidlo musí čekat. Když je proměnná IsWaitOnCrossroad rovna false, je vrácena hodnota true, udávající povolení k vjezdu do křižovatky. Jinak funkce IsClearWay ze třídy CrossroadControl zkontroluje počet stojících vozidel v objektech BeforeCrossroadCheck, které mají přednost před směrem jízdy vozidla, předaného parametrem blinkMode. Parametr nabývá hodnot 0 pro přímý směr, 1 pro levý směr a 2 pro pravý směr. IsClearWay vrátí false, pokud směr není volný nebo svítí-li červený signál u světelně řízené křižovatky. Nelze-li aktuálně pokračovat, funkce vrací False a dochází k výpočtu vzdálenosti od hranice. Když je vzdálenost větší, než určitá hodnota dochází ke zpomalování jízdy, ale ne k úplnému zabrzdění. Pod zadanou vzdálenostní mez již musí vozidlo plně zastavit.



Obrázek 5.11: Vývojový diagram metody CheckCrossroad

Implementace ThrottleService se nejdříve nacházela v Blueprint, ale kvůli funkci RayCast uvedené na diagramu 5.12 bylo nutné celou část přepsat do C++. Důvodem byla vysoká náročnost na výkon procesoru, neboť ve funkci probíhá v každém volání hledání kolize s jinými objekty.



Obrázek 5.12: Vývojový diagram metody Raycast

Princip spočívá ve vyslání třech paprsků před vozidlo zobrazených na obrázku 5.13. Počátek paprsků je vypočten podle aktuální pozice pomocných zelených koulí FrontLeft a FrontRight umístěných na vozidle. Délka paprsku se dynamicky mění podle aktuální rychlosti vozidla, neboť se vzrůstající rychlostí je brzdná dráha delší. Koncový bod je získán vynásobením požadované vzdálenosti s vektorem ve směru jízdy a připočtením k počátečnímu bodu.

Obrázek 5.13: Vyslané paprsky pro detekci překážky

K trasování paprsků Unreal poskytuje funkci LineTraceSingleByObjectType, které je potřeba předat parametr třídy CollisionObjectQueryParams, definující jaké typy objektů má paprsek testovat. Rovněž je nutné třídou CollisionQueryParams specifikovat prvky, které mají být ignorovány. Pro správnou funkčnost aplikace musí být ignorováno samotné vozidlo a všechny prvky umístěné na komunikaci kontrolující dodržování platných pravidel při jízdě uživatele. Jednotlivý výčet by nebyl možný, proto je každý takový prvek poděděn od třídy RulesActorBase, která je ignorována obecně. Je-li detekována kolize s překážkou, je uložena do Blackboard. Pokud je kolize s objektem CrossroadControl, musí být ověřeno, zda se vozidlo nenachází uprostřed křižovatky, neboť paprsek mohl zasáhnout kontrolní prvek umístěný v protisměru.

Metoda CalculateInputs aktualizuje sílu plynu a brzd. Pokud před vozidlem není žádná překážka, jsou síly regulovány podle aktuální rychlosti a maximální povolené rychlosti. Je-li detekována překážka, požadovaná rychlost je odvozena podle rychlosti protijedoucího vozidla, není-li vyšší než maximální dovolená rychlost.

5.6 Kontrola dodržování pravidel

Následující část popisuje implementaci kontroly navržených silničních pravidel během jízdy.

K informování Level Blueprint o přestupku z C++ kódu je vytvořena základní komponenta TraficRules, kterou obsahují všechny kontrolní objekty. V komponentě je deklarován delegát NewTrafficViolation typu DYNAMIC_MULTICAST_DELEGATE s parametry specifikující text chybové hlášky a typ přestupku definovaném v ETrafficError.

Informační událost o přestupku je vytvořena v C++ funkcí Broadcast() nad NewTrafficViolation a v Blueprint metodou CallNewTrafficViolation. LevelBlueprint naslouchá událostem a v CarDrivingSchoolModeBP aktualizuje slovník Violations, kde klíčem je ETraf-

ficError a hodnotou je počet vzniklých přestupků. Podobně jsou implementovány hlášky CarError, které slouží k výpisu informací o motoru a vozidle.

Rychlost jízdy

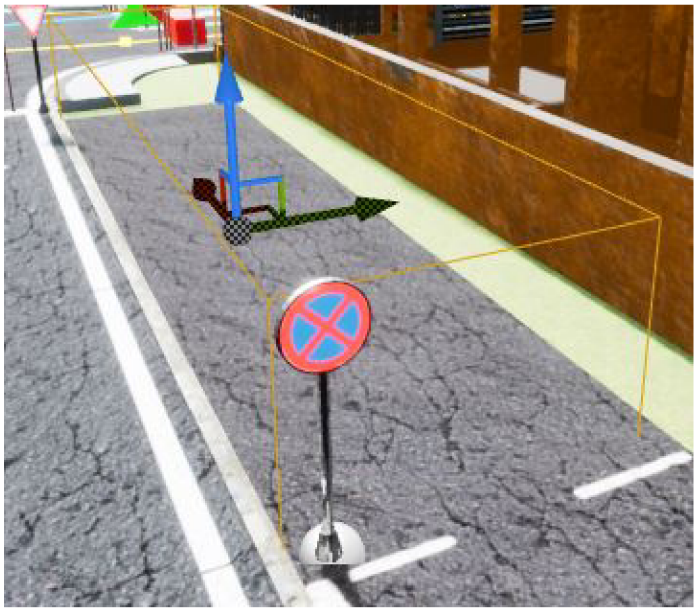
Samotná kontrola překročení maximální rychlosti v daném úseku probíhá v komponentě SpeedLimit, která je uložena ve třídě DriveCar. SpeedLimit v každém volání metody TickComponent porovnává aktuální rychlost jízdy s maximální povolenou rychlostí, jež je uložena v kontrolním prvku. Je-li aktuální rychlost vyšší než maximální dovolená rychlost je delegátem NewTrafficViolation vyvolána událost o přestupku.

Pokud vozidlo projede objektem SpeedBorder, který obsahuje kolizní těleso Trigger, je vyvolána událost BeginOverlap(Trigger). V této události je aktualizována maximální povolená rychlost v komponentě SpeedLimit. Projede-li vozidlo řízené kontrolérem CarAIController, je rovněž aktualizována maximální rychlost uložená v Blackboard komponentě.

Zákaz zastavení

Implementace zákazu zastavení na vymezeném místě se nachází v Blueprint skriptu NoStop. Obsahuje kolizní těleso definující místo, kde řidič nesmí zastavit. V události Tick je v každém volání zjištěna rychlost vozidla. Pokud je rychlost menší než 2 $km.h^{-1}$, skript vyhodnotí porušení silničního pravidla řidičem. Následně je nastaveno další vyvolání události až za určitý čas, kvůli odfiltrování neustálé kontroly, zda řidič stále stojí na daném místě, neboť není schopen vyjet z místa během jednoho snímku.

Událost Tick je po spuštění aplikace vypnuta. Aktivace je provedena v události BeginOverlap, která je vyvolána, pokud vozidlo začne projíždět kolizním prvkem. Opětovně je deaktivována po vyjetí z kolizního objektu v události EndOverlap. V aplikaci se prvek nachází za každou značkou zákazu zastavení, viz obrázek 5.14.



Obrázek 5.14: Vytyčení zákazu zastavení.

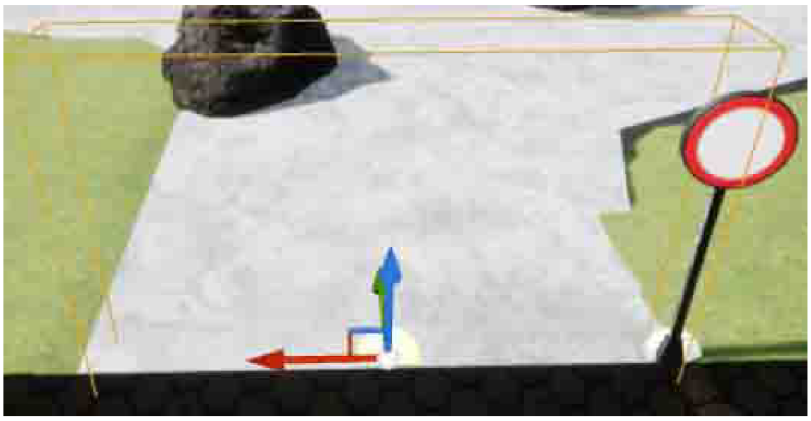
Stůj, dej přednost v jízdě

Před každou značkou "Stůj, dej přednost v jízdě", která se nachází těsně před křižovatkou, je každý řidič povinen zastavit. Kontrola probíhá podobně jako u pravidla zákaz zastavení, jenže zde je vyžadováno zastavení každého vozidla na hranici křižovatky. V kontrolovaném prostoru se tedy musí nacházet právě vždy jen jedno vozidlo, což je řešeno vytyčením místa akorát dostatečně velkým pro jedno vozidlo.

Kontrola je aktivována vjetím do kolizního prvku v události BeginOverlap, kde je nastavena proměnná WasStoped na false. V události Tick je tato proměnná změněna na true, pokud je rychlost vozidla menší než 1 $km.h^{-1}$. Vyhodnocení zda řidič zastavil na požadovaném místě je provedeno po vyjetí vozidla z kolizního objektu v události EndOverlap ověřením proměnné WasStoped. Pokud nedošlo k zastavení, je vyvolána informace o přestupku.

Zákaz vjezdu ve všech směrech

Porušení pravidla je detekováno v Blueprintu NoEntry, kde je umístěn kolizní prvek PlaceCheck, který při průniku s vozidlem vyvolá událost BeginOverlap(PlaceCheck). Každé vyvolání této události indikuje nepovolený vjezd a okamžité informování Level Blueprintu o přestupku.



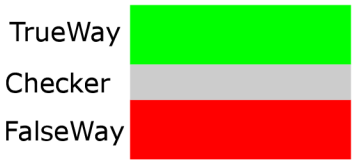
Obrázek 5.15: Vytyčení zákazu vjezdu ve všech směrech.

Zákaz vjezdu v jednom směru

Blueprint skript NoEntryOneWay rozšiřuje předcházející kontrolu o povolení jízdy v jednom směru. Kontrola je provedena využitím kolizních těles TrueWay, Checker a FalseWay, které jsou umístěny dle schématu 5.16.

Při vyvolání události BeginOverlap(TrueWay) je nastavena proměnná WrongDirection na false, v BeginOverlap(FalseWay) na true. V události BeginOverlap(Checker) je vyhodnocen přestupek, pokud je WrongDirection nastavena na true.

Uvedený postup za jistých okolností detekoval několikanásobné projetí vozidla jednotlivými kolizními prvky. Proměnná PrintError, jenž je při spuštění aplikace nastavena na true, odfiltrovává špatné detekce. Pokud je proměnná WrongDirection v BeginOverlap(Checker) rovna true a zároveň se PrintError rovná true, dochází k vyvolání hlášení o přestupku, negaci PrintError a nastavení časovače, který za určitý čas obnoví výchozí stav.



Obrázek 5.16: Schéma umístění kolizních těles ve skriptu NoEntryOneWay

Zastavení na světelném semaforu

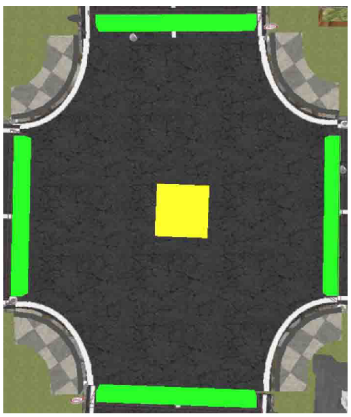
Vjezd do křižovatky, která je řízená světelnými semafory, pravidla silničního provozu zakazují, je-li aktivní červený signál. Implementace této kontroly se nachází v Blueprintu CrossroadControlBP. Prvek je kvůli řízení vozidel ostatních účastníků, jak je popsáno v 5.5, umístěn vždy před vjezdem do křižovatky v daném směru. Nachází-li se u světelně řízené křižovatky, je v proměnné LightSignal uložena reference na semafor. Rovněž je CrossroadControlBP umístěn u železničních přejezdů.

Kontrola je v události BeginOverlap(LightChecker), jež je vyvolána při průjezdu kolizním tělesem LightChecker v CrossroadControlBP. Pokud se jedná o světelnou křižovatku, porušení pravidla nastane, je-li proměnná isRed, jež se nachází v Blueprintu TrafficLightsBP implementující semafor, rovna true. V případě železničního přejezdu přestupek nastává je-li proměnná isActive rovna true.

Kontrola signalizace při průjezdu křižovatkou

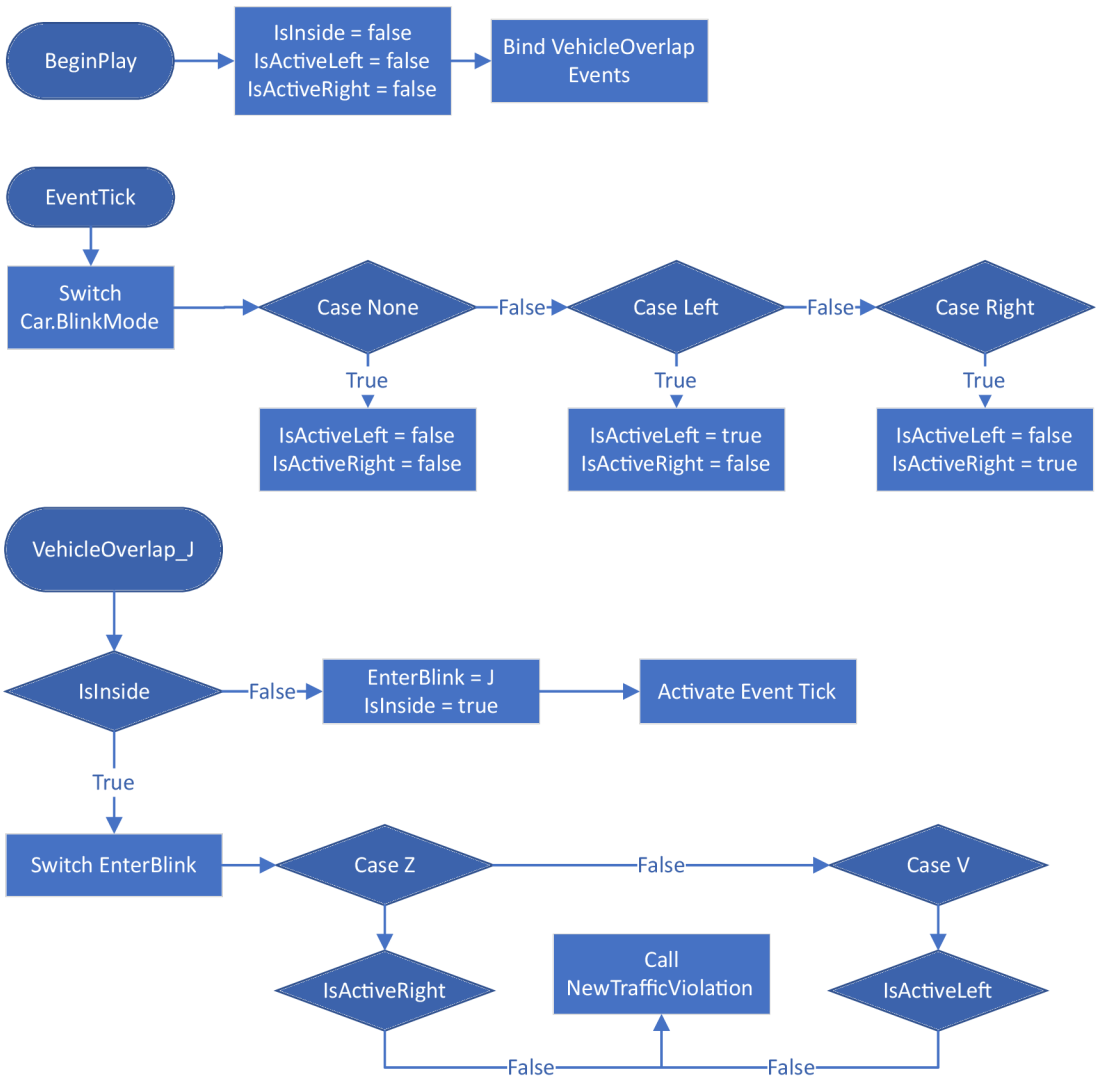
Detekce přestupku je implementována v Blueprintu BlinkManager, který s prostředím neinteraguje přímo, ale prostřednictvím kontrolních prvků BlinkCheck. Na obrázku 5.17 je znázorněna jejich pozice, kde zeleně je označen BlinkCheck, nacházející se přes celou šířku dopravní silnice na hranici křížení. BlinkManager je znázorněn žlutě a je pro přehlednost umístěn uprostřed křižovatky, ale jelikož je s křižovatkou propojen nepřímo, na jeho umístění nezáleží.

Pro propojení objektu BlinkManager s kontrolními prvky slouží asociativní pole, kde klíčem je výčet světových stran (J, Z, S, V) a hodnotou je reference na kontrolní prvek. Po zahájení aplikace je v události BeginPlay procházeno pole a inicializováno naslouchání události VehicleOverlap, která je generována, projede-li vozidlo kolizním tělesem Trigger nacházející se v BlinkCheck.



Obrázek 5.17: Umístění prvků pro kontrolu signalizace

Princip algoritmu kontroly je znázorněn na diagramu 5.18. Kvůli lepší přehlednosti je zde uvedena pouze jedna ze čtyř reakcí na událost VehicleOverlap.



Obrázek 5.18: Diagram algoritmu kontroly signalizace

K uložení stavu vozidla při průjezdu křižovatkou slouží proměnné IsInside, které uchovávají, zda se vozidlo nachází v křižovatce a IsActiveLeft nebo IsActiveRight, které charakterizují aktivitu levé nebo pravé světelné signalizace při odbočování.

Pokud je proměnná IsInside rovna false a dojde k odchycení události VehicleOverlap, reprezentuje tento stav vjetí vozidla do křižovatky. Je aktivována událost Tick, nastavena hodnota IsInside na true a uchována informace, z jakého směru vozidlo přijelo.

Událost Tick je následně volána opakovaně a ukládá aktivitu signalizace do výše zmíněných proměnných podle proměnné BlinkMode nacházející se ve skriptu DriveCar.

Vozidlo vyjelo z kontrolované křižovatky, je-li odchycena nová událost VehicleOverlap a proměnná IsInside je rovna true. V takovém případě je směr jízdy vozidla vyhodnocen podle prvku BlinkCheck, jež nastavil proměnnou IsInside na true, a aktuálním BlinkCheck, co vyvolal událost VehicleOverlap. Na základě získaného směru je ověřena správnost signalizace při průjezdu.

Kapitola 6

Testování aplikace

Jelikož tato práce klade důraz na trénink s využitím externího zařízení v podobě volantu, pedálů a mechanického převodu, bylo nutné dovézt k testovacím uživatelům externí volant s příslušenstvím. Samotné testování probíhalo na čtyřech místech, kdy na každém místě bylo domluveno více lidí najednou. Aplikaci testovali uživatelé v uvedeném pořadí.

- Rodina a přátelé.
- Kolegové v IT firmě.
- Sousedé.
- Studenti ze střední školy.

Všem testujícím bylo předem popsáno ovládání na fyzickém zařízení s využitím přílohy B. Testování začínalo spuštěním aplikace, kdy byly pozorovány reakce na úvodní obrazovku, nabídek k nastavení a spuštění tréninku. Po načtení scény byl uživatelům vysvětlen princip řízení implementovaného vozidla, kdy u lidí nevlastnící řidičské oprávnění bylo nutné vysvětlit význam a funkci jednotlivých pedálů,a naopak řidiči byli upozorněni na odlišnou citlivost pedálů oproti reálnému vozidlu. Řízení nejprve probíhalo pomocí klávesnice a po určité době byl aktivován volant. Po ukončení výcviku byli lidé požádání o vyplnění dotazníku uvedeném v příloze E, který na úvodu vyžaduje základní informace o uživateli a poté ohodnocení určitých aspektů vytvořené aplikace. Ohodnocení je provedeno pomocí bodové stupnice od 1 do 10, kde bod 1, resp. 10, udává nejhorší, resp. nejlepší hodnocení.

Zkoumané hypotézy

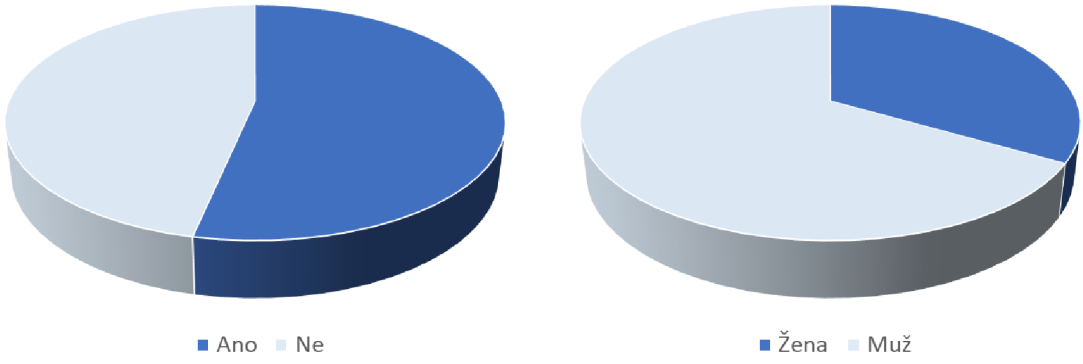
V rámci testování byly dokazovány hypotézy, které zkoumají věrohodnost navrženého města s řízením vozidla a potvrzení předpokladu o přínosu externího volantu s podporou spojky. Definované znění hypotéz je následující:

- H_1: Model města obsahuje věrohodný dopravní systém.
- H_2: Řízení pomocí externího volantu přináší značný přínos v ovládání.
- H_3: Řízení s podporou spojky zvyšuje pocit reálné jízdy.
- H_4: Fyzikální systém vozidla je na dostatečné úrovni pro trénink.

Charakteristika testovaných uživatelů

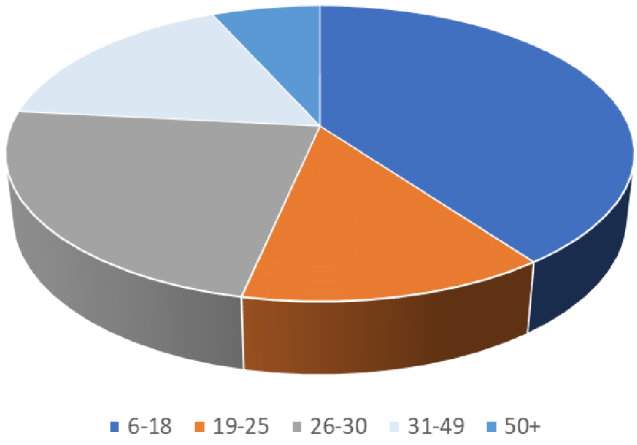
Aplikaci otestovalo celkem 30 uživatelů, kteří pocházejí ze čtyř různých lokalit uvedených výše. Vzdělání v oblasti informatiky měli uživatelé pouze ve druhé skupině testovaných uživatelů. Všichni ostatní pocházeli z jiných oblastí a mají různou úroveň vzdělání, proto je spektrum uživatelů široké.

Jelikož se testování zúčastnili studenti střední školy, kteří nemohou vlastnit řidičské oprávnění, je v uvedeném grafu 6.1 až 47 % lidí bez absolvované autoškoly. Celkové věkové rozložení je na grafu 6.2.



(a) Absolvování autoškoly (b) Pohlaví

Obrázek 6.1: Grafy absolvované autoškoly a pohlaví



Obrázek 6.2: Graf věkových skupin uživatelů

Výsledky hypotéz

Průměrné bodové hodnocení m se směrodatnou odchylkou testových otázek se nachází v tabulce 6.1. U každé z ověřovaných hypotéz je stanovena nulová hypotéza, která definuje, v jakém případě bude hypotéza přijata, a alternativní, která je přijata v opačném případě.

Stanovení alternativní hypotézy je podle minimálního bodového hodnocení dané otázky, které je považován za úspěch.

Ve všech hypotézách je stanovena kritická hodnota na 1,699, která je získaná z tabulky [29], kdy hladina významnosti α je 0,05 a stupeň volnosti ν je 29, jenž je vypočten podle rovnice 6.1, kde n je počet testovaných uživatelů. Testovací kritérium T je vypočteno podle rovnice 6.2. Alternativní hypotéza je přijata, pokud testovací kritérium je větší než kritická hodnota [43].

$$\nu = n - 1 \tag{6.1}$$

$$T = \frac{m - \mu}{s} \cdot \sqrt{n} \tag{6.2}$$

Tabulka 6.1: Výsledné průměrné hodnocení otázek a směrodatná odchylka.

No.	Text otázky	Průměr	Směrodatná odchylka
1.	Vzhled uživatelského rozhraní	8,13	1,36
2.	Ovládání aplikace	5,97	2,75
3.	Externí volant	9,47	0,88
4.	Řízení s podporou spojky	8,93	1,46
5.	Reálnost simulace vozidla	7,03	1,52
6.	Model města a okolí	7,57	1,63
7.	Dopravní značení	8,73	1,24
8.	Informování o přestupcích během jízdy	8,23	1,86
9.	Seznam přestupků	8,67	1,81
10.	Hodnocení aplikace	8,77	0,88
11.	Přínos aplikace	8,10	2,01

Hypotéza 1

Hypotéze H_1 *"Model města obsahuje věrohodný dopravní systém"* se věnuje otázka 7. *"Dopravní značení"*, která má průměrné hodnocení 8,73 a směrodatnou odchylku 1,24. Nulová a alternativní hypotéza jsou:

$$\begin{aligned} H_{10} &: \mu \leq 7 \\ H_{1A} &: \mu > 7 \end{aligned} \tag{6.3}$$

Testové kritérium T je 7,678, a lze tedy zamítnout H_{10} a přijmout H_{1A}, neboť T je větší než kritická hodnota 1,699. Hypotéza H_1 je potvrzena.

Hypotéza 2

Hypotéze H_2 *"Řízení pomocí externího volantu přináší značný přínos v ovládání"* se věnuje otázka 3. *"Externí volant"*, která má průměrné hodnocení 9,47 a směrodatnou odchylku 0,88. Nulová a alternativní hypotéza jsou:

$$\begin{aligned} H_{20} &: \mu \leq 8 \\ H_{2A} &: \mu > 8 \end{aligned} \tag{6.4}$$

Testové kritérium T je 9,083, a lze tedy zamítnout H_{20} a přijmout H_{2A}, neboť T je větší než kritická hodnota 1,699. Hypotéza H_2 je potvrzena.

Hypotéza 3

Hypotéze H_3 *"Řízení s podporou spojky zvyšuje pocit reálné jízdy"* se věnuje otázka 4. *"Řízení s podporou spojky"*, která má průměrné hodnocení 8,93 a směrodatnou odchylku 1,45. Nulová a alternativní hypotéza jsou:

$$\begin{aligned} H_{30} &: \mu \leq 7 \\ H_{3A} &: \mu > 7 \end{aligned} \tag{6.5}$$

Testové kritérium T je 7,258, a lze tedy zamítnout H_{30} a přijmout H_{3A}, neboť T je větší než kritická hodnota 1,699. Hypotéza H_3 je potvrzena.

Hypotéza 4

Hypotéze H_4 *"Fyzikální systém vozidla je na dostatečné úrovni pro trénink"* se věnuje otázka 5. *"Reálnost simulace vozidla"*, která má průměrné hodnocení 7,03 a směrodatnou odchylku 1,51. Nulová a alternativní hypotéza jsou:

$$\begin{aligned} H_{40} &: \mu \leq 7 \\ H_{4A} &: \mu > 7 \end{aligned} \tag{6.6}$$

Testové kritérium T je 0,120, a nelze tedy zamítnout H_{40}, neboť T je menší než kritická hodnota 1,699. Hypotéza H_4 tedy není potvrzena.

Vyhodnocení testování

Potvrzeny byly tři hypotézy. Hypotéza zabývající se chováním vozidla nebyla potvrzena, což bylo zmíněno, také jako postřeh pro vylepšení. Uvedena byla odlišná citlivost pedálů od skutečného vozidla a pozdější reakce vozidla.

Rovněž byla nalezena chyba u semaforů, která je nyní opravena, kdy při přechodu z červeného signálu, nebyl aktivován oranžový signál, jako u reálných semaforů.

Kapitola 7

Závěr

Cílem diplomové práce bylo vytvořit aplikaci, která zábavnou formou vede ke zlepšování řidičských dovedností v ovládání vozidla a dodržování silničních pravidel, která jsou analyzována v souvislosti s jinými trenažéry. Rovněž bylo nutné vybrat vhodné nástroje k vytvoření výsledné aplikace. Jelikož důležitým cílem bylo rozšíření řízení vozidla o manuální převod s podporou spojky, byly popsány fyzikální vlastnosti a charakteristiky důležitých komponent vozidel.

Požadované cíle se podařilo splnit. Byla vytvořena aplikace, která obsahuje jedno město s okolní krajinou, ve které probíhá trénink. V simulovaném prostředí se nachází dopravní infrastruktura a řada vyhotovených dopravních značení, která definují platná pravidla pro daný úsek komunikace. Ve městě jsou umístěny budovy několika modelů rozlišené jinou texturou a v krajině se nachází vegetace, jezero a po obvodu mapy hornatý terén. Simulátor kromě silničních cest obsahuje také železniční koleje k umožnění tréninku průjezdu na železničních přejezdech. Silnice jsou rozděleny na různé úseky, na kterých se trénuje. Zejména různé povolené rychlosti, jednosměrná silnice, zákazy vjezdu nebo zastavení. Důležitá je rovněž kontrola správného průjezdu křižovatkou, kdy je vyhodnoceno, jestli uživatel zastavil vozidlo na červený signál u světelně řízené křižovatky nebo zda signalizoval změnu směru jízdy. Vozidlo s manuálním převodem je plynule ovladatelné pomocí klávesnice a hlavně externím volantem s pedály. V aplikaci se nachází také další účastníci provozu v podobě jiných automobilů, které jsou řízeny stavovým automatem. Informativním přínosem pro uživatele je přehled počtu přestupků s celkovou částkou peněžité pokuty, která je vypočtena podle skutečných pokut.

Aplikace je postavená na herním jádře Unreal Engine 4. Finální výsledek byl testován 30 uživateli, kteří vyplnili dotazník. Podle výsledků na dotazované otázky bylo potvrzeno 3 ze 4 hypotéz. Rovněž z testování vyplynula řada možných dalších vylepšení, zejména přidání kruhového objezdu nebo zdokonalení jízdních vlastností vozidla pro větší ovladatelnost. Rozšířit aplikaci je také možné přidáním chodců a kontrol dalších pravidel, například ověřováním signalizace či nepovolené změny jízdního pruhu, nedáním přednosti nebo pomoci virtuální reality kontrolovat, zda se řidič na hranici křižovatky podíval do zpětných zrcátek.

Zhotovený trenažér splnil očekávané předpoklady a představuje základ pro další vývoj v jiné práci. Neboť vytvoření plně použitelné simulace pro potřeby autoškol přesahuje rámec jedné diplomové práce. Přesto během provedeného testování trenažér pobavil mnoho testovaných uživatelů.

Literatura

[1] 3dxo: *Free 3D Models, Textures and Photos.* [Online; navštíveno 10.1.2019].
URL https://www.3dxo.com/

[2] Autotrenažéry: *Autotrenažéry.sk - Z histórie.* [Online; navštíveno 13.12.2018].
URL http://www.autotrenazery.sk/o-nas/z-historie

[3] Blundell, M.; Harty, D.: *Multibody systems approach to vehicle dynamics.* Elsevier Butterworth-Heinemann, 2004, ISBN 9780750651127.

[4] Brodkin, J.: *How Unity3D Became a Game-Development Beast.* [Online; navštíveno 10.1.2019].
URL https://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/

[5] Bullet: Bullet Real-Time Physics Simulation. [Online; navštíveno 25.12.2018].
URL https://pybullet.org/wordpress/

[6] Centrum dopravního výzkumu: *VR 67:Nákup řidičského simulátoru pro Dopravní VaV centrum).* [Online; navštíveno 20.3.2019].
URL https://www.cdv.cz/file/vr-67b-krizovatky-urovnove/

[7] CGTextures: *Textures.* [Online; navštíveno 10.1.2019].
URL https://www.textures.com/

[8] CGtrader: *CGtrader 3D models.* [Online; navštíveno 10.1.2019].
URL https://www.cgtrader.com/

[9] cj ems: *Free3d: Opel Astra 3d model.* [Online; navštíveno 5.1.2019].
URL https://free3d.com/3d-model/opel-astra-667.html

[10] CoquiGames: *Unreal Engine Forum: Modular Road Tool.* [Online; navštíveno 9.1.2019].
URL https://forums.unrealengine.com/community/community-content-tools-and-tutorials/1492096-free-modular-road-tool

[11] Crytek: *CryEngine.* [Online; navštíveno 12.1.2019].
URL https://www.cryengine.com/

[12] Crytek: *Cryengine 5.* [Online; navštíveno 13.12.2018].
URL https://www.cryengine.com/files/features/categories/New_UI_4.jpg

[13] Epic Games: *Automotive Materials Pack.* [Online; navštíveno 8.1.2019].
URL https://www.unrealengine.com/marketplace/automotive-material-pack

[14] Epic Games: *Blueprint Compiler Overview*. [Online; navštíveno 10.3.2019].
URL https://docs.unrealengine.com/en-US/Engine/Blueprints/TechnicalGuide/Compiler

[15] Epic Games: Level Blueprint. [Online; navštíveno 12.1.2019].
URL https://docs.unrealengine.com/en-us/Engine/Blueprints/UserGuide/Types/LevelBlueprint

[16] Epic Games: *PhysXVehicles*. [Online; navštíveno 13.3.2019].
URL https://api.unrealengine.com/INT/API/Plugins/PhysXVehicles/index.html

[17] Epic games: *Unreal Engine 4.15 Release Notes*. 2017, [Online; navštíveno 5.5.2019].
URL https://docs.unrealengine.com/en-US/Support/Builds/ReleaseNotes/4_15/index.html

[18] Česká republika: *Dopravní značky - vyhláška č. 294/2015 Sb.* [Online; navštíveno 12.12.2018].
URL https://www.ibesip.cz/Besip/media/Besip/data/web/soubory/legislativa/novela-294-2015.pdf

[19] Česká republika: *Zákon č. 361/2000 Sb.* [Online; navštíveno 12.12.2018].
URL https://www.ibesip.cz/getattachment/Tematicke-stranky/Pravidla-silnicniho-provozu/361_01_10_2018.pdf

[20] Česká tisková kancelář: *Počet aut v Česku se za století zvýšil tisíckrát*. 2018, [Online; navštíveno 12.12.2018].
URL https://www.ceskenoviny.cz/zpravy/pocet-aut-v-cesku-se-za-stoleti-zvysil-tisickrat/1656046

[21] Forward Global Group: *City Car Driving*. [Online; navštíveno 16.12.2018].
URL https://store.steampowered.com/app/493490/City_Car_Driving/

[22] Free3D: *Free 3D models*. [Online; navštíveno 10.1.2019].
URL https://free3d.com/

[23] Halpern, J.: The What and Why of Game Engines. [Online; navštíveno 5.4.2019].
URL https://medium.com/@jaredehalpern/the-what-and-why-of-game-engines-f2b89a46d01f

[24] Hawok: Havok Physics. [Online; navštíveno 25.12.2018].
URL https://www.havok.com/physics/

[25] Široký, J.: *Pohybsilničníchvozidel*. [Online; navštíveno 5.5.2019].
URL http://homen.vsb.cz/~s1i95/mvd/Moodle/2_4.pdf

[26] JKZ: *Řidičský trenažer AT-208 VRT*. [Online; navštíveno 13.12.2018].
URL http://jkzsim.cz/cz/ridicske-trenazery/at-208-vrt/

[27] Logitech: *Driving Force Racing Wheel*. [Online; navštíveno 13.12.2018].
URL https://www.logitechg.com/en-ch/products/driving/driving-force-racing-wheel.html

[28] Logitech: *Driving Force Shifter* . [Online; navštíveno 13.12.2018].
URL https://www.logitechg.com/cs-cz/products/driving/driving-force-shifter.html

[29] Masarykova Univerzita: *Tabulky kvantilů vybraných rozdělení pravděpodobnosti* . 2002, [Online; navštíveno 25.6.2019].
URL https://is.muni.cz/el/1431/jaro2016/Bi5045/um/Tabulky_rozdeleni.pdf

[30] Matýšek, M.: Herní engine Unity. [Online; navštíveno 25.12.2018].
URL https://developer.nvidia.com/physx-sdk

[31] Medel Design: *FREE DOWNLOAD Small City Model Pack! for UE4 - Medel Design.* [Online; navštíveno 10.2.2019].
URL https://www.youtube.com/watch?v=SN7xm-1EBXs

[32] Monster, M.: *Car Physics for Games.* 2017, [Online; navštíveno 3.3.2019].
URL http://www.asawicki.info/Mirror/Car%20Physics%20for%20Games/Car%20Physics%20for%20Games.html

[33] Nvidia: PhysX SDK. [Online; navštíveno 25.12.2018].
URL https://developer.nvidia.com/physx-sdk

[34] OGRE: OGRE3D. [Online; navštíveno 25.12.2018].
URL https://www.ogre3d.org/

[35] Policie ČR: *Ročenka nehodovosti na pozemních komunikacích v České Republice za rok 2017.* 2018, [Online; navštíveno 12.12.2018].
URL https://www.policie.cz/soubor/rocenka-nehodovosti-2017-rar.aspx

[36] Russell, P.: *A Brief History of Driver Education in the UK.* 2002, [Online; navštíveno 12.12.2018].
URL http://derf.org.uk/history.htm

[37] SimSpace: *SimExam.* [Online; navštíveno 12.12.2018].
URL http://www.simspace.es/solutions/driving-simulators/simexam/

[38] Simspace: *Simexam.* [Online; navštíveno 12.12.2018].
URL http://www.simspace.es/solutions/driving-simulators

[39] Sovani, S.: *Steering toward the future. Advantage*, ročník VI, č. 3, 2012: str. 6–9.
URL https://www.ansys.com/-/media/ansys/corporate/resourcelibrary/article/aa-v6-i3-full-version.pdf

[40] Subaru: *Monitoring tlaku pneumatik (Tire pressure monitoring system).* [Online; navštíveno 2.3.2019].
URL http://www.subaru.cz/intra/files/files/TPMS_CZ.pdf

[41] Thacker, J.: *Check out Chaos, UE4's new 'Hollywood physics' system.* [Online; navštíveno 5.4.2019].
URL http://www.cgchannel.com/2019/03/check-out-chaos-ue4s-new-hollywood-physics-engine/

[42] Turbosquid: *3D Models for Professionals*. [Online; navštíveno 10.1.2019].
URL https://www.turbosquid.com/

[43] Univerzita Palackého: *Testování statistických hypotéz*. 2011, [Online; navštíveno 25.6.2019].
URL http://ach.upol.cz/user-files/intranet/07-testovanihypotez-2012-1347562606.pdf

[44] Vlk, F.: *Dynamika motorových vozidel: jízdní odpory, hnací charakteristika, brzdění, odpružení, řiditelnost, ovladatelnost, stabilita*. Nakladatelství a vydavatelství Vlk, 2000, ISBN 80-238-5273-6.

[45] Wikipedia: *Seznam dopravních značek v Česku*. [Online; navštíveno 10.2.2019].
URL https://cs.wikipedia.org/wiki/Seznam_dopravn%C3%ADch_zna%C4%8Dek_v_%C4%8Cesku

Příloha A

Použité dopravní značky v simulaci

Výstražné značky

Informují o výstrahách na pozemních komunikacích.

A6a
Zúžena vozovka z obou stran

Upozornění na místo, kde se vozovka oproti předcházejícímu úseku výrazně snižuje.

A10
Světelné signály

Upozornění na neočekávané místo, které je řízeno světelnými signály.

A11
Přechod pro chodce

Upozornění na neočekávaný přechod pro chodce nebo není viditelný z dostatečné vzdálenosti.

A7b
Zpomalovací práh

Upozornění na umělou nerovnost za účelem snížení rychlosti

A31a
Návěstní deska (240m)

Upozornění na železniční přejezd za 240 metrů.

A31b
Návěstní deska (160m)

Upozornění na železniční přejezd za 160 metrů.

A31c
Návěstní deska (80m)

Upozornění na železniční přejezd za 80 metrů.

A32a
Železniční výstražný kříž

Upozornění na jednokolejný železniční přejezd. Pokud řidič musí zastavit před přejezdem, zastaví u této značky.

Značky upravující přednost

P2
Hlavní pozemní komunikace

Řidiči na hlavní cestě mají přednost.

P3
Konec hlavní pozemní komunikace

Končí přednost na hlavní silnici a dochází ke změně přednosti.

P4
Dej přednost v jízdě!

Označení vedlejší silnice, nutno dát přednost v jízdě.

P6
Stůj, dej přednost v jízdě!

Nutno zastavit na místě s nejlepším rozhledem do křižovatky a nutno dát přednost v jízdě.

Zákazové značky

B1
Zákaz vjezdu všech vozidel v obou směrech

Zákaz vjezdu ve všech směrech.

B2
Zákaz vjezdu všech vozidel

Zákaz vjezdu v daném směru.

B20a
Nejvyšší dovolená rychlost

Zákaz překročení rychlosti vyjádřené na značce.

B20b
Konec nejvyšší dovolené rychlosti

Ukončení platnosti předcházející značky nejvyšší dovolené rychlosti.

B24a
Zákaz odbočování vpravo

Na nejbližší křižovatce nelze odbočit vpravo.

B24b
Zákaz odbočování vlevo

Na nejbližší křižovatce nelze odbočit vlevo.

B28
Zákaz zastavení

Zákaz zastavení vozidla, kromě okolností vyvolaných na pozemních komunikacích

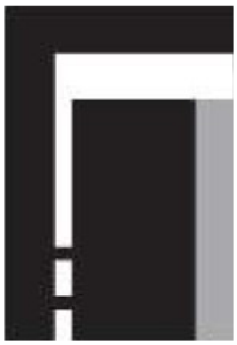
Vodorovná značení

V této práci budou použita pouze značení z kategorií podélné a příčné čáry, šipky, označení stání a parkovišť, označení zastávek. Kontrolovány budou zejména značky, které způsobují porušení platných předpisů.

V1a
Podélná čára souvislá

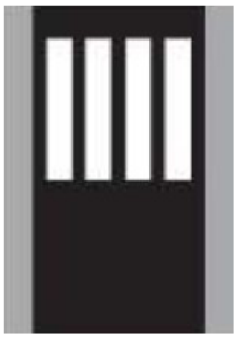
Odděluje protisměrné jízdní pruhy, které je zakázáno přejíždět

V5
Příčná čára souvislá



Vyznačuje hranici křižovatky a vyznačuje místo zastavení před ní

V7
Přechod pro chodce



Vyznačuje místo pro přecházení chodců a vyznačuje místo, před kterým musí vozidlo zastavit, je-li nutno

V9a
Směrové šipky

Stanovuje následující směr jízdy v jízdním pruhu

Příloha B

Popis ovládání

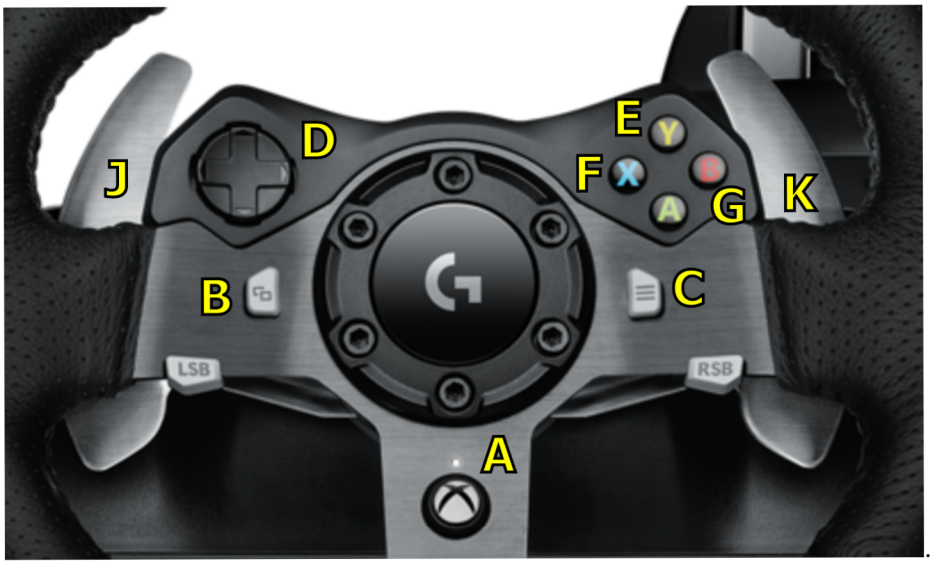
B.1 Klávesnice

Obrázek B.1: Ovládací klávesy

- A - Přepínání pohledu kamery
- B - Spojka
- C - Brzda
- D - Plyn
- E - Klakson
- F - Ruční brzda
- G - Zapnutí/vypnutí světel
- H - Aktivace motoru
- J - Deaktivace motoru
- K - Levý blinkr
- L - Pravý blinkr
- M - Natočení volantu doleva
- N - Natočení volantu doprava

- O - Řazení stupňů převodovky
- P - automatická/manuální převodovka
- Q - Neutrál

B.2 Volant Logitech G920



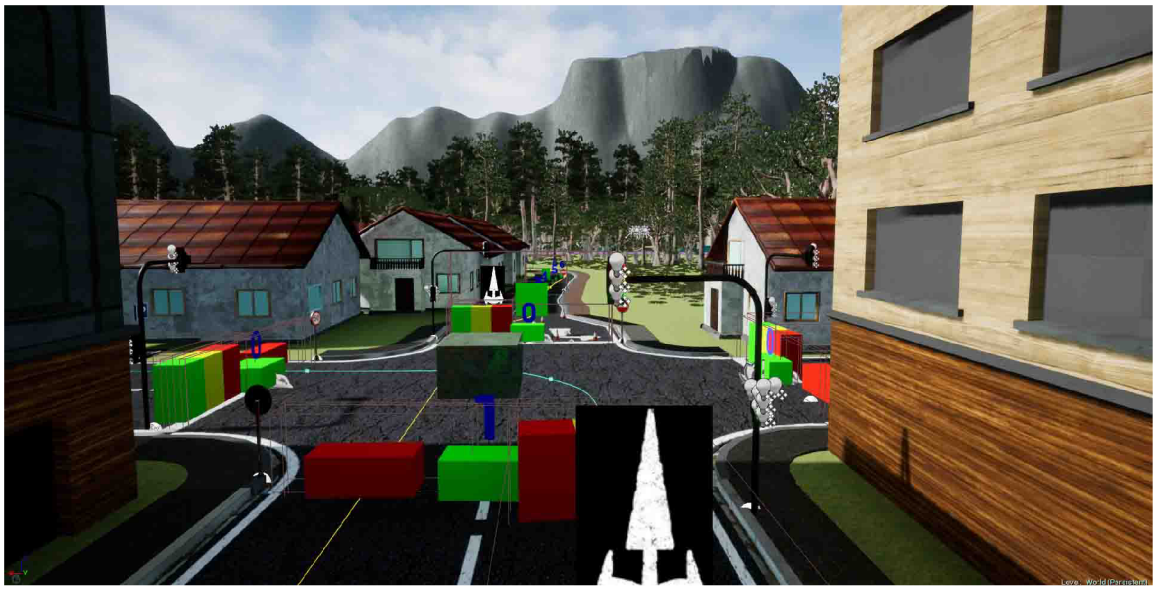
Obrázek B.2: Ovládací tlačítka na volantu

- A - Přepínání ovládání volant/klávesnice
- B - Aktivace motoru
- C - Deaktivace motoru
- D - Natáčení pohledu
- E - Zapnutí/vypnutí světel
- F - Klakson
- G - Ruční brzda
- J - Levý blinkr
- K - Pravý blinkr

Příloha C

Obrázky aplikace

Obrázek C.1



Obrázek C.2

Obrázek C.3

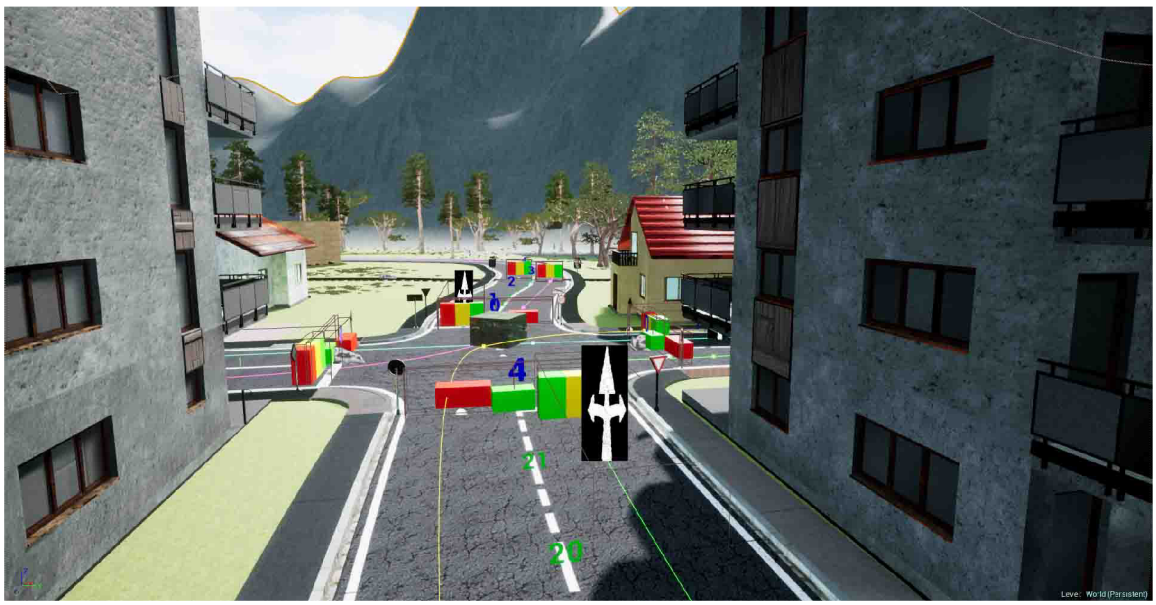
Obrázek C.4

Obrázek C.5

Obrázek C.6

Obrázek C.7

Obrázek C.8



Obrázek C.9

Obrázek C.10

Příloha D

Obsah DVD

- bin
- unreal_project
- video
- images
- documentation

Příloha E

Dotazník

Testovací dotazník k Diplomové práci na téma 3D autoškola.

Tento dotazník slouží k vyhodnocení simulátoru, který byl vytvořený v rámci diplomové práce. Aplikace klade za cíl trénink řidičských dovedností skupiny B zábavnou interaktivní formou a kontrolu dodržování silničních pravidel. Vaše odpovědi jsou anonymní a budou použity pouze k vyhodnocení práce.

Pohlaví

- ☐ Muž
- ☐ Žena

Absolvování autoškoly

- ☐ ANO
- ☐ NE

Věková kategorie

- ☐ 6 – 18
- ☐ 19 – 25
- ☐ 26 – 30
- ☐ 31 – 49
- ☐ 50 a více

Hodnocení aplikace

1. Vzhled uživatelského rozhraní (Hlavní menu, přehled přestupků, vzhled informačních prvků)

Jak hodnotíte navržené uživatelské rozhraní v aplikaci?

Nepřívětivé 1 ○○○○○○○○○○ 10 **Přívětivé**

2. Ovládání aplikace

Jak hodnotíte navržené rozložení ovládání pomocí klávesnice?
(Klávesy pro řízení blinkrů, zatáčení, brzda, plyn …)

Neovladatelné 1 ○○○○○○○○○○ 10 **Jednoduše ovladatelné**

3. Externí volant

Jaký je podle vás přínos v ovládání pomocí externího volantu oproti klávesnici?

Žádný 1 ○○○○○○○○○○ 10 **Velký**

4. Řízení s podporou spojky

Jak hodnotíte trénink řízení s využitím spojky u automobilu?
(Mnoho existujících aplikací opomíjí v simulaci spojku)

Nepotřebná 1 ○○○○○○○○○○ 10 **Potřebná**

5. Reálnost simulace vozidla

Jak hodnotíte řízení simulovaného vozidla?

6. Model města a okolí

Jak hodnotíte město a okolí?

7. Dopravní značení

Jak hodnotíte umístění a počet dopravních značení?

8. Informování o přestupcích během jízdy

Jak hodnotíte hlášky během řízení?

9. Seznam přestupků

Jak hodnotíte souhrnnou tabulku s počtem přestupků a odpovídající pokutou?

10. Hodnocení aplikace

Jak celkově hodnotíte aplikaci?

11. Přínos aplikace

Má potenciál k využití pro trénování řidičských dovedností?

Postřehy ke zdokonalení aplikace
