

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

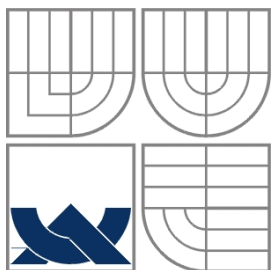
3D AUTOŠKOLA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

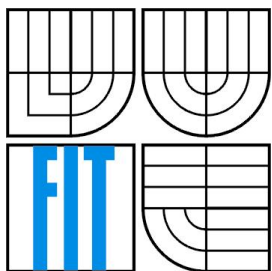
AUTOR PRÁCE
AUTHOR

Bc. PROKOP ŘEHÁČEK

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

3D AUTOŠKOLA

3D DRIVING SCHOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PROKOP ŘEHÁČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETER CHUDÝ, Ph.D. MBA

Abstrakt

Hlavním tématem práce je vytvoření simulátoru pro výukové účely, který je schopen simulovat dopravní situace a kontrolovat správné dodržování dopravních předpisů. V práci je představen moderní herní engine Unity 3D a vývoj simulátoru v jazyce C# pro tento engine. Tvorba herního simulátoru a hry obecně je složena z mnoha disciplín, a proto se tato práce zabývá i tématy, jakými je tvorba 3D modelů, práce se zvuky, návrh modelu virtuálního města a jiné.

Abstract

The main topic of this thesis is to create a simulator for educational purposes, which is able to simulate the traffic situations and to control the compliance of traffic regulations. Modern game engine Unity 3D and development of simulator in C # for this engine are introduced within this thesis. Creating a simulator or computer game in general, imposes multiple requirements in different disciplines, and therefore, this work deals with themes such as the creation of 3D models, work with audio content, design of virtual city model, etc.

Klíčová slova

3D, Autoškola, Simulace, Doprava, Unity 3D, C#, Pravidla silničního provozu

Keywords

3D, Driving School, Simulation, Traffic, Unity 3D, C#, Traffic Rules

Citace

Prokop Řeháček: 3D Autoškola, diplomová práce, Brno, FIT VUT v Brně, 2014

3D Autoškola

Prohlášení

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

.....
Bc. Prokop Řeháček
28.5. 2014

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Petru Chudému, Ph.D., MBA za odbornou pomoc při tvorbě práce. Dále bych chtěl poděkovat své rodině a přítelkyni za podporu během mého studia.

© Prokop Řeháček, 2014

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

Obsah

Obsah.....	1
Seznam zkratk.....	4
1 Úvod.....	5
2 Simulace a pravidla silničního provozu pro autoškolu.....	6
2.1 Simulátory silničního provozu.....	7
2.2 Analýza pravidel silničního provozu.....	7
2.2.1 Seznam implementačně prioritních pravidelných.....	7
2.2.2 Dopravní značky.....	8
2.2.3 Rychlost jízdy.....	8
2.2.4 Zastavení a stání.....	9
2.2.5 Odbočování a jízda křižovatkou.....	9
2.2.6 Řízení provozu světelnými signály.....	10
2.2.7 Směr jízdy a způsob jízdy v jízdnicích.....	11
2.2.8 Objíždění a předjíždění.....	12
3 Technologické nástroje a prostředky.....	13
3.1 Výběr herního enginu.....	13
3.1.1 DX Studio.....	13
3.1.2 Torque 3D.....	14
3.1.3 Unity 3D.....	15
3.1.4 Zvolený herní engine pro 3D Autoškolu.....	15
3.2 Další použité nástroje.....	16
4 Architektura simulátoru a uživatelské rozhraní.....	17
4.1 Architektura simulátoru.....	17
4.1.1 Podpora více jazyků.....	17
4.2 Systém kontroly dopravních předpisů.....	18
4.2.1 Kontrola jízdy v jízdnicích a směru jízdy.....	18
4.2.2 Kontrola dodržování rychlosti.....	19
4.2.3 Kontrola odbočování a jízdy křižovatkou.....	19
4.2.4 Kontrola řízení provozu světelnými signály.....	20
4.2.5 Kontrola zastavení a stání.....	20
4.3 Uživatelské rozhraní.....	21
4.3.1 Hlavní menu a nastavení.....	21
4.3.2 Ovládání simulátoru.....	22
4.3.3 Palubní deska.....	22
4.3.4 Dialogy a informace během jízdy.....	23
4.3.5 Vyhodnocení jízdy.....	23
5 Modely pro simulátor.....	24
5.1 Model automobilu.....	24
5.2 Model města a vodorovné dopravní značení.....	24
5.2.1 Tvorba vlastních modelů vozovky.....	25
5.3 Svislé dopravní značky.....	26
6 Implementace.....	27

6.1	Základní struktura projektu v Unity 3D.....	27
6.1.1	Způsob programování v Unity 3D.....	27
6.2	Tvorba města.....	28
6.2.1	Realizace návrhu města v Unity 3D.....	28
6.3	Pomocné objekty pro určení porušení pravidel silničního provozu.....	29
6.3.1	Pomocná tělesa pro určení jízdy mimo vozovku.....	29
6.3.2	Pomocná tělesa pro změny maximální povolené rychlosti.....	29
6.3.3	Pomocná tělesa pro zákaz zastavení.....	29
6.3.4	Pomocná tělesa pro jednosměrnou pozemní komunikaci.....	30
6.3.5	Pomocná tělesa pro projetí křižovatkou.....	30
6.3.6	Pomocná tělesa pro světelnou křižovátku.....	31
6.4	Implementace stavového automatu pro řízení aplikace.....	31
6.4.1	Správce stavů a rozhraní stavů.....	31
6.4.2	Stavy aplikace.....	32
6.5	Základní fyzikální model automobilu.....	33
6.5.1	Vložení do scény a úprava fyzikálních vlastností.....	33
6.5.2	Skript pro pohyb a ovládání automobilu.....	34
6.6	Grafické uživatelské rozhraní přístrojové desky.....	36
6.7	Ostatní účastníci provozu.....	37
6.7.1	Reprezentace pohybu účastníků provozu.....	37
6.8	Implementace světelné křižovátky.....	38
6.9	Dodržování prioritních pravidel.....	39
6.9.1	Výpis zpráv o přestupcích.....	39
6.9.2	Kontrola zastavení na značce „Stůj, dej přednost v jízdě“.....	40
6.9.3	Kontrola porušení zákazu zastavení.....	41
6.9.4	Kontrola jízdy mimo vozovku.....	41
6.9.5	Kontrola překročení maximální povolené rychlosti.....	42
6.9.6	Kontrola směru jízdy po jednosměrné komunikaci.....	43
6.9.7	Kontrola jízdy křižovatkou.....	44
6.9.8	Kontrola jízdy světelnou křižovatkou.....	45
6.9.9	Interakce s ostatními účastníky provozu.....	46
6.9.10	Vyhodnocení jízdy.....	46
7	Testování.....	48
7.1	Kazuistika – případová studie.....	48
7.1.1	Stanovení hypotéz.....	48
7.1.2	Sestavení dotazníku.....	48
7.1.3	Výsledky testování.....	49
7.1.4	Výsledky textových otázek.....	50
7.1.5	Testování hypotéz.....	50
8	Vyhodnocení a závěr.....	54
	Literatura.....	55
	Seznam příloh.....	58
	Příloha 1.....	59
	Příloha 2.....	62

Seznam zkratek

3D	Three-dimensional Space	Trojrozměrný prostor
GUI	Graphical User Interface	Grafické uživatelské rozhraní
XML	Extensible Markup Language	Rozšiřitelný značkovací jazyk
WAV	Waveform Audio File Format	Formát audio souboru ve vlnové formě
FBX	Filmbox	Filmbox formát
OS	Operating System	Operační systém
GNU	GNU's Not Unix	GNU není Unix

1 Úvod

Řízení motorového vozidla je v dnešní době již běžnou věcí. V České republice připadne průměrně 2,27 obyvatele na jeden automobil (statistika k 30.6. 2012) [1]. Vysoký počet automobilů, a s tím spojený vysoký počet řidičů, s sebou nese kromě výhod i svá úskalí, zejména v podobě dopravní autonehody s následky zranění či smrti. Neméně nebezpečné jsou dopravní přestupky, které mohou vést k nehodám a občan přichází o peníze, body nebo i samotný řidičský průkaz. Podle statistik ministerstva dopravy [2] byl počet nehod v roce 2013 5. nejvyšší od roku 1990. Dále je uvedeno, že policie ČR za prvních jedenáct měsíců loňského roku šetřila 77 318 nehod, při kterých bylo 529 osob usmrceno, 2 566 osob těžce zraněno a 20 872 osob zraněno lehce. Odhadnutá hmotná škoda policií na místě nehody je 4 511,75 mil. Kč.

Uvedená čísla jsou stále velmi vysoká, proto se tato práce snaží přispět k bezpečnosti na českých vozovkách tvorbou komplexního vzdělávacího simulátoru pro nácvik řešení různých dopravních situací, se kterými se může řidič motorového vozidla setkat.

Ve druhé kapitole jsou uvedena vybraná pravidla silničního provozu a vymezeny priority pravidel, která jsou implementována do výsledné aplikace. Jelikož je kontrola dopravních předpisů velmi obsáhlým tématem, simulátor bude zaměřen především na dopravní značení a křižovatky. Třetí kapitola je věnována technologickým nástrojům (Unity 3D, C#, Blender, 3Ds Max) a také důvodům proč byly zvoleny tyto konkrétní technologie. Dále jsou zde uvedeny zdroje vytvořené/převzaté/upravené, které se dále využívají v 3D autoškole. Tedy dopravní značení, silnice, model automobilu a další. Čtvrtá kapitola je zaměřena na návrh aplikace, jenž je rozdělen na části věnující se uživatelskému rozhraní, systému kontroly jednotlivých skupin silničních pravidel a reprezentací součástí simulátoru - např. pozemní komunikace a křižovatek. Pátá kapitola se přesune od teoretického úvodu k více praktickým součástem ve vývojovém procesu 3D autoškoly, a konkrétně k získání a přípravě 3D modelů pro simulátor. Šestá kapitola je věnována implementaci v Unity 3D. Je v ní popsán postup tvorby města a dopravního systému v Unity 3D. Dále obsahuje šestá kapitola konkrétní algoritmy a postupy pro kontrolu dodržování pravidel silničního provozu.

Unity 3D umožňuje publikovat aplikace na webové stránky, a proto se nabízí tohoto využít pro testování reálnými uživateli, čemuž je věnována sedmá kapitola. Testování je provedeno především kvůli zpětné vazbě od uživatelů. K vyhodnocení testování byla použita metoda zvaná kazuistika neboli případová studie, v níž byly na začátku stanoveny hypotézy, které se následně ověřily metodou *Testování hypotéz o průměru pro jeden výběr*. V poslední kapitole je pak zhodnocení práce a návrh na její další pokračování.

2 Simulace a pravidla silničního provozu pro autoškolu

Jednou z podmínek pro udělení řidičského průkazu je odborná způsobilost k řízení motorového vozidla [4]. Výuku a výcvik k získání řidičského oprávnění a již zmíněné způsobilosti k řízení motorového vozidla může provádět výlučně provozovatel autoškoly. Předmětem výuky studentů autoškoly je získání potřebných teoretických i praktických znalostí, dovedností a návyků k řízení motorových vozidel v provozu na pozemních komunikacích [19].

Některé autoškoly začínají první praktické lekce v simulátorech založených na virtuální realitě, kterým se říká trenažery. Ukázku takového trenažeru můžete vidět na obrázku 2.1. Této metody se využívá především proto, aby si student autoškoly mohl vyzkoušet první kroky pro ovládání vozidla v bezpečí a nebyl omezován a stresován ostatními účastníky provozu. Při lekcích v trenažeru je sice student zdánlivě ochuzen o autentické smyslové vjemy z reálné jízdy, ale může si osvojit základní úkony potřebné k ovládání vozidla tak, aby je již při první reálné jízdě zvládl bez problémů [20].



Obrázek 2.1: Řidičský trenažer [20]

Výše uvedený typ simulátoru je pro svou cenu ve většině případů pouze nadstandardem některých autoškol a běžný uživatel k němu nemá přístup. Tato práce se věnuje edukativnímu simulátoru, který si dává za úkol spíše připravit uživatele na různé dopravní situace, než na osvojení úkonů pro řízení, jak tomu je u řidičského trenažeru. Takovýchto simulátorů existuje několik, ale jejich kvalita se výrazně liší. Následující podkapitola se věnuje několika příkladům profesionálních simulátorů, pro učení silničních pravidel a další podkapitola se věnuje již rozboru samotných pravidel.

2.1 Simulátory silničního provozu

Na internetu lze nalézt mnoho jednoduchých 2D webových her, které se věnují vzdělávání v oblasti silničních předpisů, ale jsou většinou zaměřeny na jednu konkrétní oblast pro trénování (parkování, křižovatky atd.). Za zmínku stojí například Drivers Ed, Valet Parking nebo Driving School GT. Tato práce se sice zabývá 3D autoškolou, ale i z výše uvedených her je možné čerpat inspiraci.

Z profesionálních 3D simulátorů, které umožňují jízdu ve městech a kontrolují pravidla silničního provozu, jsem vybral následující: Driving Simulator 2012, 3D Fahrschule.

Driving Simulator 2012 se snaží být co nejvíce realistickým simulátorem pro řízení automobilu, uživatel zde musí dávat znamení při změně směru jízdy, zapínat světla a v případě manuální převodovky i řadit. Hra nevyučuje pravidla silničního provozu, protože předpokládá, že je uživatel již zná, a proto slouží především k procvičování již nabytých znalostí [21].

3D Fahrschule umožňuje jízdu v evropských zemích například v Belgii, v Německu, ve Francii, v Nizozemí a ve Velké Británii. V programu lze zvolit jízdu s instruktorem, tedy na rozdíl od předchozího programu Driving Simulator 2012, 3D Fahrschule uživatele učí pravidla silničního provozu. V nabídce je i možnost realizace volných jízd a lze dokonce volit i jiný typ vozidla (motocykl, terénní vozidlo nebo automobil) [22].

2.2 Analýza pravidel silničního provozu

Pravidla silničního provozu se nacházejí zejména v zákonu č. 361/2000 Sb. o provozu na pozemních komunikacích a o změnách některých zákonů (zákon o silničním provozu), ve znění pozdějších předpisů, v němž je upravena též oblast přestupků proti bezpečnosti a plynulosti provozu, bodové hodnocení řidičů nebo oblast řidičských oprávnění a řidičských průkazů [3].

Výsledný program autoškoly je soustředěn především na osobní automobily, protože jejich procentuální zastoupení převyšuje jiné kategorie motorových vozidel [17], a proto se soustředí na pravidla v zákoně, která se týkají řidičského oprávnění třídy B. Pro tuto práci je nejdůležitější díl 3. výše uvedeného zákona s názvem „Pravidla provozu na pozemních komunikacích“. V následujících podkapitolách jsou probrána pravidla, která byla vybrána pro implementaci do výsledné aplikace. Nejprve je však uveden prioritní seznam vybraných pravidel silničního provozu. Zdroj informací a citací pro vytvoření podkapitol je především [4].

2.2.1 Seznam implementačně prioritních pravidelných

Při návrhu aplikace bylo vybráno několik pravidel silničního provozu, jenž si tato práce dává za úkol implementovat do výsledného produktu. Z důvodu velkého objemu práce, byl vytvořen list seřazený podle priority, která určuje pořadí při implementaci tak, aby při případném nestihnutí implementace některé položky, simulátor obsahoval ty podstatnější pravidla.

Při vytváření listu bylo zohledněno několik faktorů, jako například: objem práce potřebný k implementaci, důležitost pro celkovou koncepci aplikace a jiné. Finální seznam je následující:

1. Dopravní značením.
2. Kontrola dodržování maximální povolené rychlosti.
3. Kontrola zastavení a stání.

4. Odbočování a jízda křižovatkou.
5. Křižovatka řízená světelnými signály.
6. Směr jízdy v jízdních pruzích.
7. Objíždění a předjíždění.

2.2.2 Dopravní značky

Rozlišují se dopravní značky svislé a vodorovné. Svislé dopravní značky se dělí na stálé, proměnné a přenosné. Plocha proměnné dopravní značky se může měnit. Přenosná svislá značka je umístěná na červenobíle pruhovaném sloupku (stojánku) nebo na vozidle. Vodorovné dopravní značky jsou stálé a přechodné.

Svislé dopravní značky

Druhy svislých značek jsou uvedeny v následujícím seznamu. Uživatel 3D autoškoly se setká v simulátoru se všemi druhy dopravních značek..

- Výstražné značky, které upozorňují na místa, kde účastníku provozu na pozemních komunikacích hrozí nebezpečí a kde musí dbát zvýšené opatrnosti.
- Značky upravující přednost, které stanoví přednost v jízdě v provozu na pozemních komunikacích.
- Zákazové značky, které ukládají účastníku provozu na pozemních komunikacích zákazy nebo omezení.
- Příkazové značky, které ukládají účastníku provozu na pozemních komunikacích příkazy.
- Dodatkové tabulky, které zpřesňují, doplňují nebo omezují význam dopravní značky, pod kterou jsou umístěny.

Vodorovné dopravní značky

Vodorovné dopravní značky se užívají samostatně nebo v kombinaci se svislými značkami, popřípadě s dopravními zařízeními, jejichž význam zdůrazňují nebo zpřesňují. Vodorovné dopravní značky jsou vyznačeny barvou nebo jiným srozumitelným způsobem; přechodná změna místní úpravy provozu na pozemních komunikacích je vyznačena žlutou nebo oranžovou barvou.

2.2.3 Rychlost jízdy

Rychlost jízdy musí řidič přizpůsobit především svým schopnostem, vlastnostem vozidla a nákladu, předpokládanému stavebnímu a dopravně technickému stavu pozemní komunikace, její kategorii a třídě, povětrnostním podmínkám a jiným okolnostem, které je možno předvídat; smí jet jen takovou rychlostí, aby byl schopen zastavit vozidlo na vzdálenost, na kterou má rozhled.

Maximální povolená rychlost pro motorové vozidlo o hmotnosti nepřevyšující 3500 kg je mimo obec 90 km·h⁻¹ a na dálnici 130 km·h⁻¹. V obci smí jet řidič nejvýše rychlostí 50 km·h⁻¹, a jde-li o dálnici nebo silnice pro motorová vozidla, nejvýše 80 km·h⁻¹. Místní úpravou lze rychlost v obci zvýšit, ale jen maximálně o 30 km·h⁻¹.

2.2.4 Zastavení a stání

Stání je definováno jako uvedení vozidla do klidu nad dobu dovolenou 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 nejbližší 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.

Řidič nesmí zastavit a stát v těchto situacích (výčet je zredukován pro potřeby simulátoru):

- Na přechodu pro chodce nebo na přejezdu pro cyklisty a ve vzdálenosti kratší než 5m před nimi.
- Na křižovatce a ve vzdálenosti kratší než 5m před hranicí křižovatky a 5m za ní; tento zákaz neplatí v obci na křižovatce tvaru „T“ na protější straně vyústující pozemní komunikace.
- V připojovacím nebo odbočovacím pruhu.
- V jízdních pruzích vyznačených na vozovce mimo pravého jízdního pruhu.

2.2.5 Odbočování a jízda křižovatkou

Zákon definuje křižovatku jako místo, v němž se pozemní komunikace protínají nebo spojují; za křižovatku se nepovažuje vyústění polní nebo lesní cesty nebo jiné účelové pozemní komunikace na jinou pozemní komunikaci. Hranice křižovatky je místo vyznačené vodorovnou dopravní značkou „Příčná čára souvislá“, „Příčná čára souvislá se symbolem Dej přednost v jíždě!“ nebo „Příčná čára souvislá s nápisem STOP“ viz. obrázek 2.3; kde taková dopravní značka není, tvoří hranici křižovatky kolmice k ose vozovky v místě, kde pro křižovatku začíná zakřivení okraje vozovky. Křižovatka s řízeným provozem je křižovatka, na které je provoz řízen světelnými, případně i doprovodnými akustickými signály, policistou, vojenským policistou nebo usměřňován strážníkem obecní policie.



Obrázek 2.3: Vodorovné dopravní značení, které značí hranici křižovatky [12]

Při odbočování musí řidič dávat znamení o směru jízdy, nesmí ohrozit řidiče jedoucí za ním a musí dbát zvýšené opatrnosti. Pokud řidič při odbočování opouští průběžný pruh, musí co nejdříve vjet na odbočovací pruh, je-li tento vyznačen. Před odbočením se musí řidič zařadit co nejbližší k příslušnému okraji podle směru odbočení.

Řidič odbočující vlevo musí dát přednost v jízdě protijedoucím motorovým i nemotorovým vozidlům, tramvajím jedoucím v obou směrech a vozidlům jedoucím ve vyhrazeném jízdním pruhu, pro něž je tento jízdní pruh vyhrazen. Řidič odbočující vpravo musí dát přednost v jízdě vozidlům jedoucím ve vyhrazeném jízdním pruhu, pro něž je tento jízdní pruh vyhrazen. Tam, kde je povolena jízda podél tramvaje vlevo, musí dát přednost v jízdě i tramvaji.

Vozidlo přijíždějící na křižovatku po vedlejší komunikaci označené značkou „Dej přednost v jízdě!“ nebo „Stůj, dej přednost v jízdě!“ musí dát přednost v jízdě vozidlům přijíždějícím po hlavní pozemní komunikaci. Na příkaz značení „Stůj, dej přednost v jízdě!“ musí řidič zastavit vozidlo na takovém místě, odkud má do křižovatky náležitý rozhled. Pokud není přednost ustanovena předchozím značením, musí dát řidič přednost v jízdě vozidlům přijíždějícím zprava. Řidič nesmí vjet do křižovatky, pokud mu situace nedovoluje pokračovat v jízdě v křižovatce a za křižovatkou, takže by byl nucen zastavit vozidlo v křižovatce.

Řidič vjíždějící na kruhový objezd označený dopravními značkami „Kruhový objezd“ společně se značkou „Dej přednost v jízdě!“ nebo „Stůj, dej přednost v jízdě!“ musí dát přednost v jízdě vozidlům jedoucím po kruhovém objezdu.

2.2.6 Řízení provozu světelnými signály

Při řízení provozu na křižovatce se užívá zejména světelných signálů tříbarevné soustavy s plnými signály, které můžete vidět na obrázku 2.4, nebo se směrovými signály.



Obrázek 2.4: Semafor s plnými terči a popisem jednotlivých signálů [13]

Signál s červeným světlem „Stůj!“ znamená pro řidiče povinnost zastavit vozidlo před vodorovnou dopravní značkou značící hranici křižovatky nebo před světelným signalizačním zařízením. Pokud svítí současně červený a žlutý signál, řidič má povinnost připravit se k jízdě. Plný zelený signál „Volno“ znamená, že řidič může pokračovat v jízdě a dodrží-li ustanovení o odbočování, může odbočit vpravo nebo vlevo, přičemž musí dát přednost chodcům přecházejícím ve volném směru po přechodu pro chodce. Svítí-li světlo tohoto signálu „Signál pro opuštění křižovatky“ umístěný v protilehlém rohu křižovatky, neplatí pro odbočování vlevo povinnost dávat

přednost v jízdě protijedoucím vozidlům. Signál se žlutým světlem „Pozor!“ znamená povinnost zastavit vozidlo před vodorovnou dopravní značkou značící hranici křižovatky nebo před semaforem; je-li však vozidlo při rozsvícení signálu již tak blízko, že by řidič nemohl vozidlo bezpečně zastavit, smí pokračovat v jízdě. Svítí-li světlo tohoto signálu přerušovaně, nejde o křižovatku s provozem řízeným světelnými signály, a proto platí pravidla pro projíždění křižovatkou z předchozí kapitoly.

2.2.7 Směr jízdy a způsob jízdy v jízdnicích pruzích

Jízdnicí pruh je definován zákonem jako část vozovky dovolující jízdu vozidel jiných než dvoukolových (motocyklů) v jednom jízdnicím proudu za sebou. Krajnice je část pozemní komunikace ležící mezi okrajem jízdnicího pruhu a hranou pozemní komunikace, skládá se zpravidla ze zpevněné a z nezpevněné části.



Obrázek 2.2: Vodorovné dopravní značky, které oddělují jízdnicí pruhy [12]

Na českých silnicích se jezdí vpravo, a pokud možno při pravém okraji vozovky, pokud není ustaveno jinak. Na krajnici smí řidič vjet jen při zastavení a stání nebo jestliže je to nutné, při předjíždění, vyhýbání, odbočování nebo otáčení, ale musí přitom dbát zvýšené opatrnosti.

Mimo obec se na komunikaci o dvou nebo více jízdnicích pruzích vyznačených na vozovce v jednom směru jízdy jezdí v pravém jízdnicím pruhu. V ostatních pruzích se smí jet, jestliže je to nutné k objíždění, předjíždění nebo odbočování. V obci ve stejné situaci smí řidič užívat k jízdě kteréhokoliv jízdnicího pruhu a nepovažuje se za předjíždění, jedou-li vozidla v jednom z jízdnicích pruhů rychleji než vozidla v jiném jízdnicím pruhu. Mimo obec se také nepovažuje za předjíždění souběžná jízda v hustém provozu.

Přejíždět z jednoho jízdnicího pruhu do druhého smí řidič jen tehdy, neohrozí-li a neomezí-li řidiče jedoucího v jízdnicím pruhu, do kterého přejíždí; přitom musí dávat znamení o změně směru jízdy.

Na obrázku 2.2 jsou vyobrazena značení, která oddělují jízdnicí pruhy, a při jejich přejíždění platí následující pravidla:

Vodorovná dopravní značka „Podélná čára souvislá“ nesmí být přejížděna, pokud to není nutné k objíždění, odbočování na místo ležící mimo komunikaci. Odděluje-li značka jízdnicí pruhy s protisměrným provozem, musí řidič jet vpravo od tohoto značení. „Dvojitá podélná čára souvislá“

nesmí být přejížděna nebo nákladem přesahována, pokud to není nutné k objíždění. Pro protisměrné pruhy, platí stejné pravidlo jako pro „Podélnou čáru souvislou“. Značka „Podélná čára přerušovaná“ se smí přejíždět při dodržení příslušných ustanovení. Poslední vodorovnou značkou je „Podélná čára souvislá doplněná čarou přerušovanou“ a jsou-li touto značkou odděleny jízdní pruhy s protisměrným provozem, platí pro řidiče význam jen pravé čáry z této dvojice čar. Pravidla z tohoto odstavce byla čerpána z [12].

2.2.8 Objíždění a předjíždění

Při objíždění stojícího vozidla nebo překážky provozu anebo chodce, kdy dochází k vybočení ze směru jízdy, musí řidič dávat znamení o změně směru jízdy a nesmí ohrozit ani omezit protijedoucí řidiče a ohrozit ostatní účastníky provozu.

Předjíždí se vlevo. Vpravo se předjíždí vozidlo, které odbočuje vlevo a není-li již pochybnost o dalším směru jeho jízdy. Řidič, který při předjíždění vybočuje ze směru své jízdy, musí dávat znamení o změně směru jízdy a nesmí ohrozit řidiče jedoucí za ním. Při zařazování před vozidlo, které řidič předjíždí, musí dávat znamení o změně směru jízdy a nesmí ohrozit předjížděné vozidlo. Řidič v předjížděném dopravním prostředku nesmí zvyšovat rychlost jízdy ani jinak bránit předjíždění.

3 Technologické nástroje a prostředky

Vývoj hry, nebo v tomto případě vzdělávacího simulátoru, je velmi široký a komplexní úkol. Zahrnuje velké množství různých disciplín, jako například: umělá inteligence, animace, digitální malování, editace zvuku a další. Všechny tyto oblasti znalostí je možné zhmotnit ve stovky multimediálních a datových aktiv. Aby bylo možné z těchto aktiv vytvořit výsledný produkt je třeba příslušného softwaru – herní engine [5].

Důvody, proč použít již hotový engine a nevytvářet vlastní, lze vyčíst i z definice herního engine od Jeffa Warda „Herní engine existuje proto, aby abstrahoval detaily běžných prací spojené s herním vývojem, jako renderování, fyziku a vstupy, aby se vývojáři mohli soustředit na detaily, které dělají jejich hru unikátní [6].“

3.1 Výběr herního engine

Při výběru herního engine byly stanoveny tyto hlavní požadavky:

- zdarma,
- 3D,
- jazyk C++, C#, Java nebo Javascript,
- dostupnost literatury, dokumentace a tutoriálů,
- import 3D modelů z aplikací Blender a 3Ds Max.

Do užšího výběru byly vybrány následující enginey DX Studio, Torque 3D a Unity 3D.

3.1.1 DX Studio

DX Studio je komplexní vývojové prostředí pro tvorbu interaktivní 3D grafiky vytvořené firmou Worldweaver Ltd. DX Studio obsahuje vlastní real-time editor (viz Obrázek 3.1), ve kterém může vývojář vytvářet 2D a 3D uživatelské rozhraní a scény. Vkládání 3D modelů do scény je možno pomocí drag and drop.



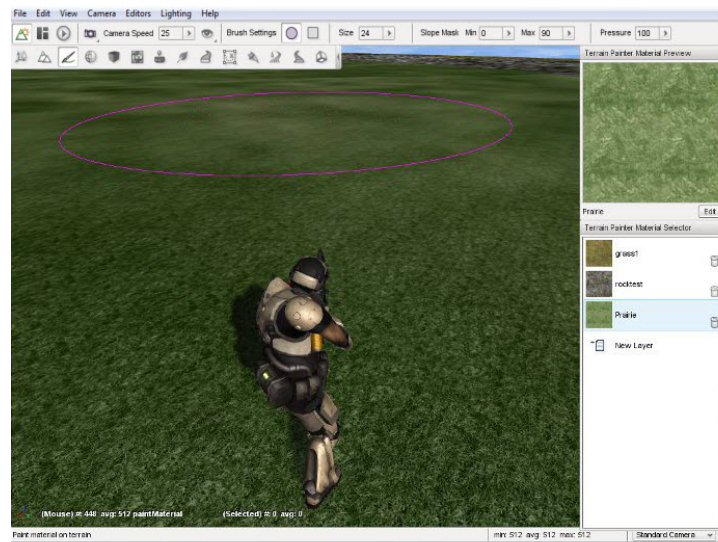
Obrázek 3.1: Editor DX Studio [8]

DX studio podporuje terén neomezené délky a možnost jeho vytváření přímo v editoru, včetně kreslení přímo na terén. Dále umožňuje práci s shadery, světly a stíny.

Fyzika je v engineu zajištěna pomocí PhysX řešení od společnosti nVidia. Nadstandardní možnosti DX Studia jsou například různé vizuální efekty tvořené pomocí vestavěného částicového systému, za jehož pomoci můžeme vytvořit např. oheň, dým a exploze. K vdechnutí života do scén neboli interaktivity se v DX Studiu používá skriptování v jazyce Javascript. Poslední stabilní verze vyšla v roce 2010, proto bohužel není komunita kolem DX Studia moc aktivní. Na svých stránkách mají rozsáhlou dokumentaci ve formě wiki stránek. Informace byly čerpány z webu produktu [8].

3.1.2 Torque 3D

V roce 2011 zakladatelé GarageGames prodávali herní game engine za 100\$, což byla v té době velmi inovativní věc a v roce 2012 uvolnili Torque 3D engine jako opensource pod MIT licencí [9]. Torque 3D je plně vybavený herní engine, který poskytuje časem otestovanou základnu pro multiplayer a grafický engine nové generace. Engine poskytuje skriptovací jazyk TorqueScript, s jehož pomocí mohou uživatelé začít vytvářet vlastní hry. Dále je k dispozici celý zdrojový kód Torque 3D, v případě potřeby upravovat nebo vytvářet vlastní herní engine. Na obrázku 3.2 je zobrazen The World editor, jenž je centrálou pro tvorbu her v Torque 3D a zastřešuje spolupráci s ostatními programy. Vytvářejí se v něm prostředí a úrovně her.

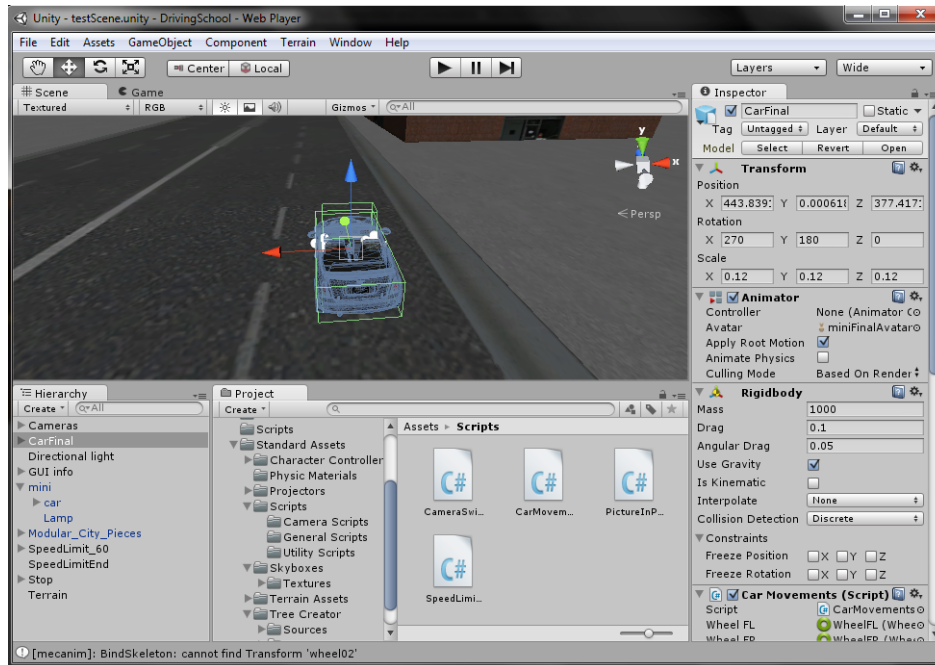


Obrázek 3.2: The World editor pro Torque 3D [10]

Torque 3D podporuje standardní grafické a zvukové formáty pro budování scén. Fyzikální engine je, stejně jako u DX studia, poskytován z třetí strany, konkrétně lze použít např. PhysX[23] nebo Bullet[24]. Na webu je k dispozici rozsáhlá dokumentace a diskusní fórum komunity, také vyšlo několik knih zabývajících se vývojem her v Torque 3D. Výsledné hry lze publikovat pro OS Windows nebo na web, kde je uživatel spouští uvnitř webového prohlížeče [10].

3.1.3 Unity 3D

Podle autorů knihy Unity 4.x Cookbook reprezentuje Unity pravou demokratizaci herního vývoje [5]. Engine i editace multimédií jsou uživatelsky přívětivé a univerzální. Pro skriptování si může vývojář



Obrázek 3.3: Editor scén Unity 3D

vybrat mezi třemi jazyky Javascript, C# a Boo. Psaní kódu je zajištěno pomocí editoru třetí strany, jako výchozí je nastaven editor MonoDevelop, jež se řadí mezi otevřený software. Na obrázku 3.3 je zobrazena asi nejsilnější stránka Unity, kterou je editor scén. V editoru vytváří uživatel jednotlivé scény případně úrovně hry. Jsou zde nástroje na vytváření terénu (nanášení stromů, textur atd. přímo na povrch), práce s jednotlivými herními objekty (nastavení fyzikálních vlastností, umístění ve scéně atd.). Vkládání externího obsahu do scény se provádí metodou *drag and drop*. Pokud uživatel stiskne tlačítko se symbolem trojúhelníku (*play*), přepne se do herního módu a může vyzkoušet výslednou scénu přímo v editoru. Unity má obrovskou uživatelskou základnu (přes 1,2 milionu v létě 2013) a velmi aktivní uživatelskou komunitu [7].

Vyšlo také několik publikací věnujících se vývoji her v Unity 3D. Výslednou hru, scénu nebo prezentaci je možné spouštět na Windows, OS X a GNU Linux. Další možností je vytvořit program pro web, který se spouští přímo v prohlížeči s Unity 3D pluginem nebo nativním klientem v Google Chrome. Informace o produktu byly čerpány z webu Unity 3D [11].

3.1.4 Zvolený herní engine pro 3D Autoškolu

Po přihlédnutí k počátečním požadavkům, které splnily všechny tři herní enginey, bylo rozhodnuto ve prospěch Unity 3D, a to především kvůli jeho aktivní komunitě, dostupnosti zdrojů a výbornému editoru, který je neustále vyvíjen. Dalším důležitým faktorem pro Unity je jeho multiplatformnost a

především možnost publikovat simulátor přímo na webu, kde si jej mohou uživatelé vyzkoušet přímo bez instalací (pokud nepočítáme instalaci pluginu Unity 3D).

3.2 Další použité nástroje

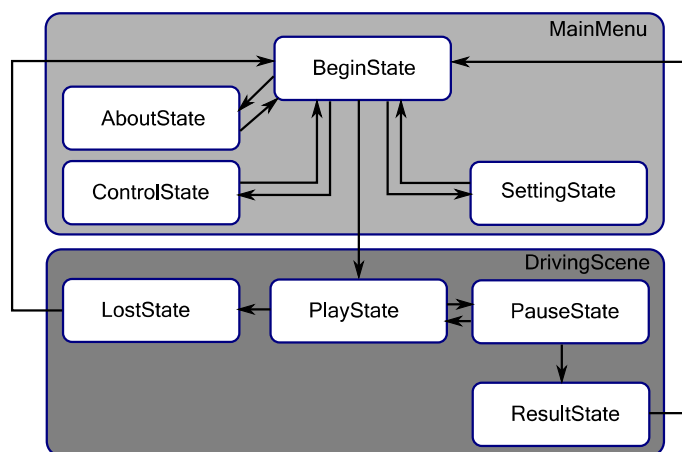
Pro práci s modely byl nejprve vybrán 3D modelovací nástroj Blender. Jedná se o otevřený software, který je zcela zdarma. Během hledání volně dostupných modelů pro simulátor 3D autoškoly se objevil problém s formátem max, který se nachází na velkém množství serverů s modely k volnému užití. Soubory tohoto typu mají binární a uzavřený formát profesionálního programu Autodesk 3ds Max a v současné době je nelze otevřít nebo konvertovat v jiné aplikaci. Firma Autodesk, která tento produkt vyvíjí, nabízí většinu svých aplikací a nástrojů volně k užití studentům pod speciální licenci a Autodesk 3ds Max není výjimkou. Z výše uvedených důvodů byly pro práci a úpravu modelů zvoleny oba modelovací nástroje. Velmi obsáhlým zdrojem 3D modelů je 3dwarehouse [25], kde jsou veškeré modely uloženy ve formátu programu SketchUp, který byl také použit, ale pouze pro konverzi do formátů kompatibilních s programem Blender.

Dalším vjemem, který napomáhá k vyšší realističnosti a k názornosti simulátoru, je zvuk. Výsledná aplikace by mohla obsahovat například tyto zvuky: zvuk motoru, tikání při zapnutém odbočovacím světle, klakson atd. Pro tyto účely byl vybrán volný otevřený software Audacity, jenž slouží k editaci a nahrávání zvukových stop.

4 Architektura simulátoru a uživatelské rozhraní

4.1 Architektura simulátoru

Při návrhu architektury jádra aplikace byl využit přístup, který uvádí Terry Norton v knize Learning C# by Developing Games with Unity 3D Beginner's Guide[26], tedy vytvořit stavový automat pro jednotlivé fáze aplikace (hlavní menu, nastavení, herní stav, pozastavená hra apod.) a přepínání jednotlivých scén nebo úrovní. Tento přístup byl aplikován na problematiku simulátoru 3D Autoškoly a na obrázku 4.1 můžete vidět výsledný stavový automat reprezentující návrh architektury.



Obrázek 4.1: Stavový automat aplikace

Jak vidíte na obrázku, 3D autoškola se bude skládat ze dvou scén. První scéna je určena pro hlavní menu a nastavení a vlastní spuštění simulace. Při stisknutí tlačítka „Start“ přepne stavový automat do stavu PlayState a nahraje se druhá scéna, která obsahuje kompletní model města s dopravním značením. Uživatel může kdykoli během jízdy pozastavit hru a vstoupit do mezi-herního menu, ve kterém může ukončit jízdu, aplikaci nebo opět pokračovat v jízdě.

4.1.1 Podpora více jazyků

Pro větší přístupnost nečesky mluvícím uživatelům byl navržen systém pro podporu více jazyků. V základní sestavě jsou obsaženy pouze český a anglický jazyk, ale s možností rozšíření o další jazyky. Textové řetězce použité v 3D Autoškolě budou uloženy v souboru XML s jednoduchou strukturou, kterou můžete vidět níže na obrázku 4.2.


```

<languages>
  <language>
    <LanguageName>English</LanguageName>
    <Settings>Settings</Settings>
    ...
  </language>
  <language>
    <LanguageName>Čeština</LanguageName>
    <Settings>Nastavení</Settings>
    ...
  </language>
</languages>

```

Obrázek 4.2: XML struktura pro podporu více jazyků

Jednotlivé jazyky a jejich přeložené textové řetězce se nacházejí v uzlech *language*. První potomek tohoto uzlu *LanguageName* obsahuje název daného jazyka a následující uzly obsahují již samotné textové řetězce. V případě zájmu o další rozšíření jazykové podpory aplikace, lze do XML souborů pouze přidat další uzel s jazykem a přeloženými řetězci a aplikace jej při zpracování souboru zahrne do nastavení, kde bude moci uživatel přepínat mezi jazyky.

4.2 Systém kontroly dopravních předpisů

Pravidla silničního provozu jsou velmi komplexní a dají se rozdělit na mnoho částí. Priority a výběr pravidel, které bude simulátor kontrolovat, jsou uvedeny v kapitole 2. V následujících podkapitolách jsou jednotlivá pravidla převedena na konkrétní návrhy pro implementaci kontroly. Dopravní značení nemá svou vlastní podkapitolu, protože je přidružené k jednotlivým tématům.

Pro detekci stavů nebo spouštění událostí využívá Unity 3D kolizních těles. Tato kolizní tělesa mohou být neviditelná a mohou se například použít, pokud potřebujete změnit aktuální stav maximální povolené rychlosti, po projetí kolem svislého dopravního značení „Maximální povolená rychlost 60“. V tomto případě vytvoříme kvádrové kolizní těleso přes pruh, u kterého chceme změnit rychlost, a pokud uživatel projede tímto tělesem, spustí se událost (spouštěč trigger), která může být libovolně nadefinována.

4.2.1 Kontrola jízdy v jízdnicích pruzích a směru jízdy

Před samotným detekováním směru a kontroly jízdy je nejprve nutné provést návrh reprezentace jízdnicích pruhů. Během jízdy musí aplikace kontrolovat následující:

- dodržení směru jízdy daného jízdnicího pruhu,
- jízdu mimo vozovku,
- jízdu přes plnou nebo dvojitou čáru.

Pro kontrolu dodržování směru jízdy budou použity správně natočená kvádrová kolizní tělesa, u kterých lze určit kolize s konkrétní stěnou (pokud automobil narazí do správné stěny kolizního tělesa, jede ve správném směru jízdy). Tato směrová tělesa budou vytvořena pro každý pruh a rozmístěna v rozestupech po celé délce pruhu.

Dále bude pozemní komunikace lemována kolizními tělesy, která při kolizi upozorní uživatele, že opouští vozovku. Uvnitř simulátoru se nastaví stavová proměnná určující jízdu mimo vozovku na hodnotu `true`. Při opětovné kolizi, tedy navrácení zpět na vozovku se proměnná nastaví na původní hodnotu `false`.

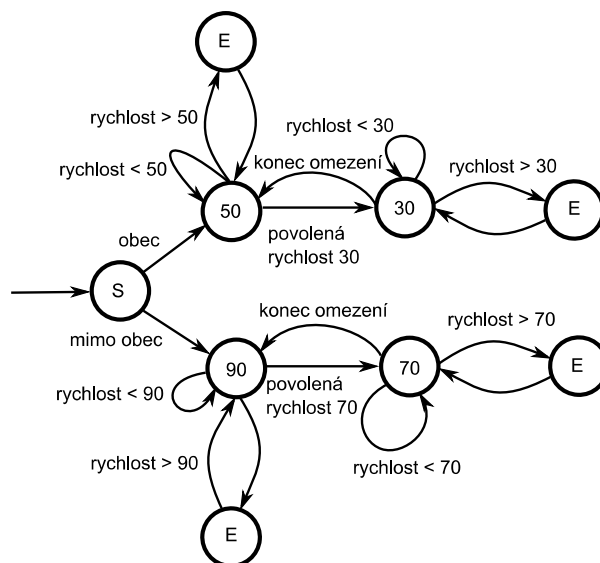
Vodorovné dopravní značení jízdních pruhů budou interně definovány, podobně jako kraj vozovky, vždy skupinou kolizních těles rozložených po celé délce pruhu.

4.2.2 Kontrola dodržování rychlosti

V simulátoru může uživatel jezdit ve městě i vyjet ven z města, proto se při spuštění konkrétní úrovně nastaví výchozí hodnota rychlosti buď na $50 \text{ km} \cdot \text{h}^{-1}$, pokud uživatel začíná jízdu ve městě nebo $90 \text{ km} \cdot \text{h}^{-1}$ v případě, že startuje mimo město.

Kontrola rychlosti funguje jako stavový automat. Na začátku tedy přejde automat do stavu, který je dán výchozí rychlostí. Pokud uživatel jede pomaleji, než je současná maximální povolená rychlost, setrvává ve stejném stavu. V případě, že se řidič setká se značkami, které mohou měnit maximální povolenou rychlost (konec obce, maximální povolená rychlost) přejde automat do stavu s novou rychlostí. Když je ve stavu nové rychlosti a projede přes křižovatku, pak se obnoví původní rychlost.

Při nedodržení maximální povolené rychlosti přejde automat do stavu zobrazení chyby, která vyvolá funkci pro vykreslení dialogu pro překročení rychlosti. Automat se hned vrací zpět do stavu kontroly rychlosti. Zjednodušenou verzi automatu můžete vidět na obrázku 4.3.



Obrázek 4.3: Zjednodušený konečný automat reprezentující kontrolu rychlosti

4.2.3 Kontrola odbočování a jízdy křižovatkou

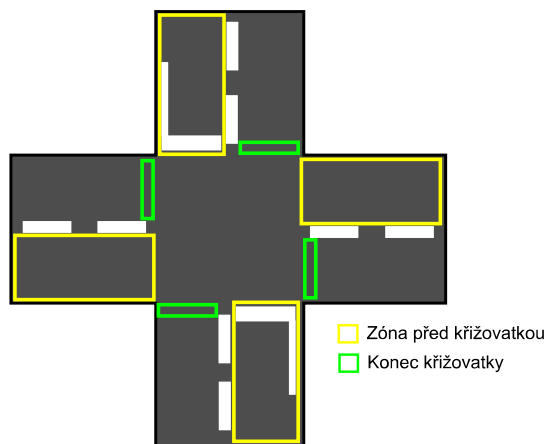
Na schématu, které se nachází v obrázku 4.4, je znázorněn návrh reprezentace křižovatky vzhledem k možnostem programu Unity 3D.

V ústí pruhu na pozemní komunikaci, jehož směr jízdy směřuje do křižovatky, je zóna (tvořena jako kvádrové kolizní těleso), která je dále definována značkou nebo předností, vycházející z dané křižovatky. V případě, že je před křižovatkou značka „Stůj, dej přednost v jízdě“, musí řidič

v této zóně před křižovatkou zastavit (aktuální rychlost automobilu se musí rovnat nule), rozhlédnout se a případně pokračovat v jízdě.

V pruzích, které vedou ven z křižovatkou, se nachází kolizní těleso, jenž definuje konec křižovatkou. Toto těleso je určeno například pro obnovení rychlosti, jenž byla omezena na některé komunikaci před křižovatkou.

Pokud dojde v křižovatce k jakékoli kolizi s ostatními účastníky provozu, je to bráno jako nedání přednosti v jízdě nebo špatnému vyhýbání se při odbočování. Pro každý druh pruhu bude definován jiný název kolizních těles, proto bude systém schopen na základě kolize s určitým pruhem moci rozhodnout, zda je přejetí přes pruh validní (např. přerušovaná čára) nebo se má vyvolat chyba (čára plná).



Obrázek 4.4: Reprezentace křižovatky pro Unity 3D s kolizními tělesy

4.2.4 Kontrola řízení provozu světelnými signály

Před křižovatkou, která je řízena světelnými signály, se bude nacházet těleso, které nastaví simulátor do stavu určeného pro tento typ křižovatek. V případě, kdy na semaforu bliká prostřední žluté světlo „Pozor!“, jsou světelné signály ignorovány a provádí se kontrola pro běžnou křižovatkou popsaná v předchozí kapitole. Stav semaforu s rozsvíceným červeným signálem „Stůj“ nebo stav s rozsvíceným signálem „Pozor!“ uzamknou pro řidiče křižovatkou a simulátor kontroluje, jestli řidič nevjede do křižovatkou. Ve druhém zmiňovaném případě, je řidič vozidla dle zákona povinen zastavit před hranicí křižovatkou nebo semaforem; je-li však vozidlo při rozsvícení signálu již tak blízko, že by řidič nemohl vozidlo bezpečně zastavit, smí pokračovat v jízdě [4]. Tato skutečnost je ošetřena přidáním druhého kolizního tělesa těsněji před křižovatkou, pokud byl u předchozího tělesa signál „Volno“ a u druhého se změnil na „Pozor!“, může uživatel pokračovat v jízdě a bude ošetřen jako následující poslední stav. Posledním stavem je rozsvícený zelený signál „Volno“, v tomto případě může uživatel vjet vozidlem do křižovatkou, a pokud nedojde ke kolizím s ostatními účastníky provozu, je to bráno jako správné projetí křižovatkou.

4.2.5 Kontrola zastavení a stání

Jestliže uživatel zastaví a není zrovna před křižovatkou, zkontroluje systém, jestli může na daném místě zastavit a případně zobrazí dialog o porušení předpisu.

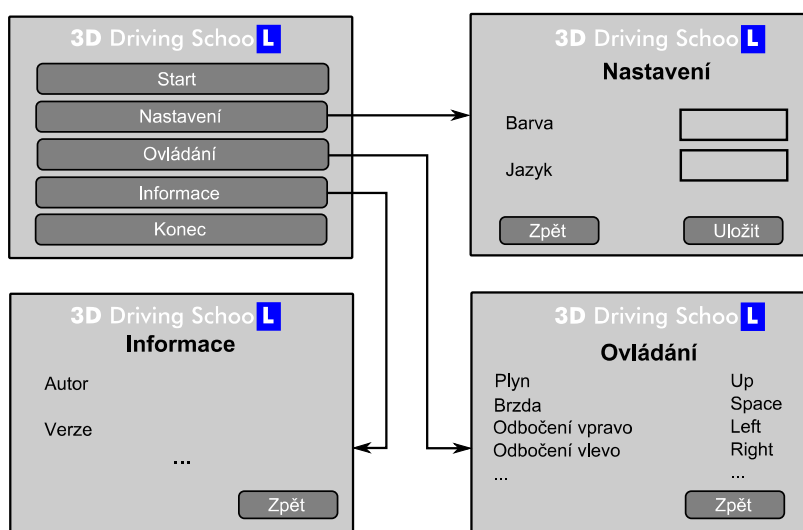
4.3 Uživatelské rozhraní

Důležitou složkou většiny her, která zajišťuje potěšení z jejich užívání, je kvalita vizuálních prvků a grafické uživatelské rozhraní (dále GUI). Prvky GUI ovlivňují způsob interakce uživatele s počítačovou hrou a také styl, jak hra předává uživateli aktuální informace (například porušení předpisu, aktuální stav bodů) [5].

4.3.1 Hlavní menu a nastavení

Součástí většiny her a simulátorů je hlavní menu, ve kterém si uživatel nastaví parametry hry, ovládání, hlasitost a především mu menu umožní samotnou hru spustit. Při navrhování systému bylo rozhodnuto, že program 3D autoškoly se bude skládat z jedné úrovně, kde je umožněno uživateli jezdit volně po městě. V budoucím rozšíření simulátoru by mohlo být obsaženo například několik úrovní, kdy každá úroveň bude obsahovat více obtížných dopravních situací.

V menu se bude nacházet položka „Start“, která přesune uživatele do scény s městem, kde může zahájit volnou jízdu. Další důležitým prvkem je „Nastavení“, jenž bude mít vlastní obrazovku a zde si uživatel zvolí barvu automobilu (v případném rozšíření i druh) a jazyk aplikace. Informativní položka „Informace o 3D Autoškolě“ bude obsahovat informace o aktuální verzi aplikace a o autorovi. Pod tlačítkem ovládání se bude skrývat obrazovka s informacemi o ovládání vozidla. Poslední možnost hlavního menu je položka „Konec“, která uzavře aplikaci.



Obrázek 4.5: Návrh obrazovek nabídky menu

Unity 3D umožňuje tvorbu trojrozměrného hlavního menu a tato možnost byla i vybrána pro 3D Autoškolu. Menu tedy bude umístěno do scény před automobil stojící na křižovatce a při nastavování barvy automobilu, se bude změna promítat přímo na model v menu. Navrhované schéma obrazovek je znázorněno na obrázku 4.5.

4.3.2 Ovládání simulátoru

Před samotnými návrhy rozložení ovládání na klávesnici, bylo nejprve zanalyzováno, které úkony potřebuje uživatel autoškoly provádět při simulaci jízdy automobilem. Výčet úkonů, které bude moci uživatel 3D autoškoly používat je následující:

- přidávání plynu,
- brzdění,
- použití ruční brzdy,
- řazení (ve výsledné verzi zatím pouze automatické),
- zatáčení,
- dávání znamení o změně směru jízdy,
- zapnutí světel (automatické),
- rozhlížení se vpravo a vlevo,
- změna pohledu (kamera za automobilem/v karoserii).

Některé úkony, jako připoutání se a sešlápnutí pedálu spojky, nebudou pro jednoduchost vyžadovány a implementovány. Také bylo navrženo, že program bude zaměřen především na procvičení dopravních předpisů a nikoli na ovládání automobilu (na tento typ cvičení jsou vhodnější řídičské trenažery), proto se implementuje pouze automatická převodovka.



Obrázek 4.6: Rozvržení ovládání na klávesnici [16]

Přidávání plynu, brzdění a zatáčení je ve výchozím nastavení přiřazeno ke klávesám šipky, nebo čtveřice kláves WSDA. Ruční brzda je obdobně jako u běžných závodních her na klávese mezerník. Rozhlížení vlevo zajišťuje klávesa „Q“ a vpravo „E“. Zapnutí a vypnutí blinkru při odbočování ze směru jízdy vlevo je nastaveno na klávesu „R“ a vpravo na klávesu „F“. Výchozí nastavení kláves je koncipováno tak, aby ovládání bylo co nejvíce intuitivní a rozložené pro obě ruce, pravá ovládání vozidla, levá ovládání kamer, rozhlížení atd. Na obrázku 4.6 je schéma klávesnice a vyznačené ovládání.

4.3.3 Palubní deska

Palubní deska je obvykle umístěna za volantem před řidičem a obsahuje informace o stavu automobilu ve formě kontrolních přístrojů. Jak vypadá běžná palubní deska, je zobrazeno na obrázku 4.7. Dvěma nejvíce sledovanými a také nejvíce viditelnými ukazateli jsou rychloměr (udává aktuální rychlost vozidla) a otáčkoměr. Dalšími dvěma ukazateli, které bývají velikostně menší, ale neméně důležité, jsou měřič teploty motoru a ukazatel stavu paliva. Na palubní desce jsou ve většině případů šipky vpravo a vlevo, které blikají, pokud řidič zapnul znamení o změně směru jízdy (v tom případě bliká jen konkrétní šipka, dle zadaného směru odbočení) nebo při zapnutí

výstražných světel (blikají obě kontrolky). Při zapnutí světel se u některých automobilů podsvítí palubní deska a u jiných má svou kontrolku. Pro simulátor byl zvolen pouze rychloměr (jelikož je v simulátoru pouze automatická převodovka, nedává zobrazení otáčkoměru smysl) a kontrolky pro odbočovací světla.



Obrázek 4.7: Schéma běžné palubní desky [18]

4.3.4 Dialogy a informace během jízdy

Uživatel 3D Autoškoly musí být během jízdy informován, jak zvládá dodržovat předpisy a pokyny virtuálního instruktora autoškoly. Pokud poruší uživatel nějaký předpis, například nezastaví na křižovatce, před kterou se nachází značka „Stůj, dej přednost v jízdě!“, objeví se na obrazovce dialog ve formátu, jaký můžete vidět na obrázku 4.8. Vlevo je umístěna značka nebo symbol pro pravidlo, které porušil a vpravo se nachází citace ze zákona o daném přestupku a celé to zastřešuje nadpis, který upozorňuje na porušení pravidel silničního provozu. Dialog je lemován červenou výstražnou barvou, aby bylo jasné, že uživatel nepostupuje správně.



Obrázek 4.8: Schéma dialogu při porušení předpisu

4.3.5 Vyhodnocení jízdy

Aby měl simulátor pro uživatele i nějakou vypovídající hodnotu, bylo navrženo zakomponování vyhodnocení jízdy. Výstupem používání simulátoru bude tedy vyhodnocení, kolika přestupků se uživatel dopustil během jízdy. Tato data budou vyobrazena v tabulce, pokud uživatel zvolí ukončení jízdy. Mimo přestupků zde bude i informace o najetých kilometrech a času jízdy. Jako bonus se uživateli spočítá částka, kterou by zaplatil, kdyby byl pokutován Policií ČR (tyto částky bude možné měnit s lokalizací simulátoru. V základním nastavení budou pouze částky pro Českou republiku a to v českých korunách). Sumy pro jednotlivé přestupky jsou převzaty z webu [27], ale nejedná se o aktuální ani oficiální sazebník, proto jsou výsledné částky pouze orientační. Další hodnocení zatím nebude implementováno, ale pokud bude program v budoucnu rozšířen o jízdy s instruktorem, bylo by vhodné zadat hodnocení, jak uživatel správně uposlechl pokynů.

5 Modely pro simulátor

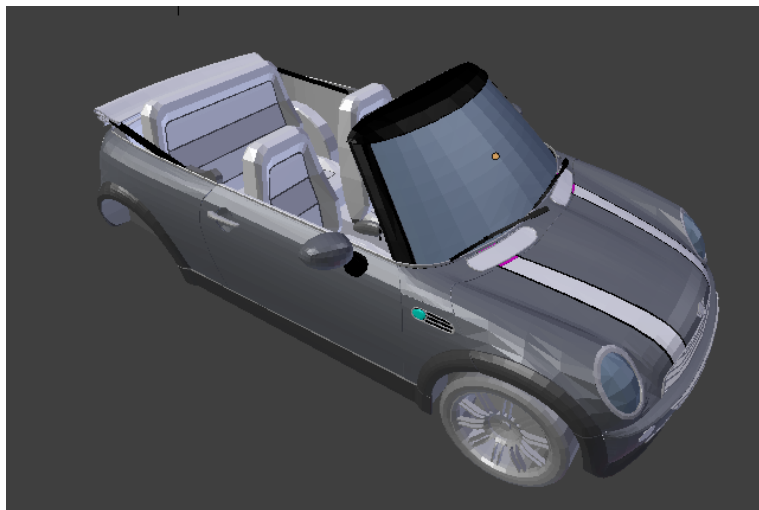
Podle změny zákona o autoškolách z roku 2011 musí řidiči vykonávat zkoušku ve větších městech, protože je zde větší provoz a koncentrace dopravního značení.

Z těchto důvodů je logická orientace simulátoru především na jízdu velkým městem s různými druhy křižovatek, s množstvím přikázaných směrů apod. Další důležitou částí autoškoly je automobil, ve kterém jezdí student. Pro větší možnosti výhledu byl vybrán automobil typu kabriolet. V následujících podkapitolách je popsáno získávání, úpravu a vytváření modelů pro simulátor.

5.1 Model automobilu

Na internetu existuje mnoho stránek s 3D modely k volnému užití. Mezi nejznámější patří stránky tf3dm.com a archive3d.net.

Z několika stovek modelů byl vybrán model Mini Cooper Cabrio [14], který můžete vidět na obrázku 5.1. V případě budoucího rozšíření simulátoru 3D Autoškoly by bylo vhodné přidat více modelů automobilů případně jiných dopravních prostředků. Příprava modelu pro Unity 3D vyžadovala několik úprav: ponechání pouze jednoho kola (kola se vytvoří jako speciální herní objekty v Unity 3D, a pro tyto účely stačí jedno kolo, ze kterého se duplikují ostatní), oddělení volantu od palubní desky a opětovnému přiřazení textur v programu Unity, otočení auta o 180°, odstranění kamery a pozadí ze scény. Tyto úkony byly provedeny v 3D modelačním programu Blender a model byl následně uložen do formátu fbx, který je nativně podporován Unity.



Obrázek 5.1: Obrázek upraveného modelu Mini Cooper Cabrio z editoru Blender

5.2 Model města a vodorovné dopravní značení

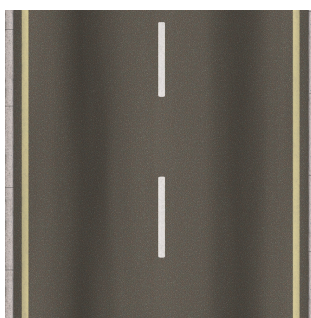
V Asset store programu Unity existuje několik pluginů nebo řešení jak vytvořit své vlastní město, ale žádný z nich není volně k užití. Jedním ze zajímavých způsobů tvorby města a jeho komunikací je

Modular City Street Pack. Tento balík obsahuje několik desítek druhů silnic, křižovatek a dálnic, které se dají na sebe napojovat a tím vytvářet město podobně jako je tomu u dětských stavebnic, nebo modelů železnic, kdy jednotlivé koleje napojujete do sebe, aby vám vznikla výsledná trať pro vlakovou soupravu.

Při hledání bezplatných alternativ byl nalezen Unity Game Engine - Modular City Kit Free od Jamese Andrta. Jedná se také o modulární balík, ale v nižším rozlišení, a proto je určen spíše pro mobilní zařízení nebo webovou aplikaci. Nabízené silnice nedostačují požadavkům na simulátor autoškoly, jsou zde pouze čtyřproudové silnice.

5.2.1 Tvorba vlastních modelů vozovky

Z výše uvedeného balíku Modular City Kit Free byly nakonec použity pouze budovy. Při tvorbě jednotlivých dílů modulární vozovky se vycházelo z jedné textury, kterou můžete vidět na obrázku 5.2.



Obrázek 5.2: Výchozí textura využitá pro tvorbu vlastního modulárního balíku

Tato textura byla následně upravena v editoru GIMP a bylo vytvořeno přes 20 textur pro vozovku (více proudu silnice, různé druhy pruhů atd.) a 10 druhů textur pro křižovatky.

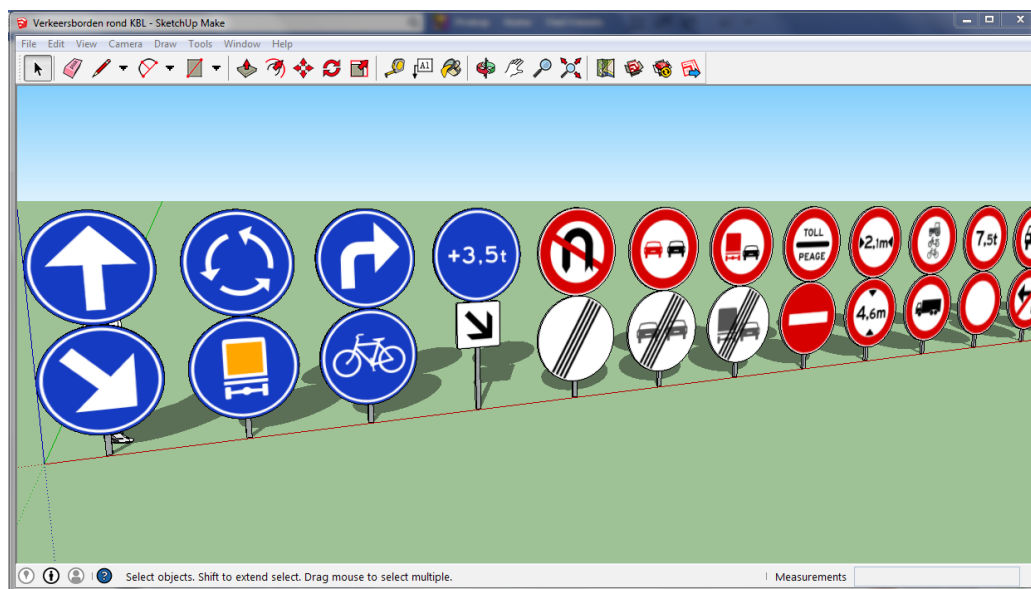
V modelačním programu Blender byly vytvořeny jednoduché modely plochy, na které se namapovaly textury. Jelikož Blender neumí při exportování modelu do formátu fbx vložit textury přímo do exportovaného souboru, bylo nutné výsledné modely konvertovat do formátu fbx včetně textur pomocí Autodesk FBX convertor 2013. Takto vytvořené modely jsou již plně kompatibilní s Unity 3D a lze pomocí nich budovat silniční síť přímo v IDE unity. Ukázku rozpracovaného města skládaného z modulů, lze vidět na obrázku 5.3.



Obrázek 5.3: Ukázka vytvořených modulů pro tvorbu města v Unity 3D

5.3 Svislé dopravní značky

Internetový archiv volně dostupných 3D modelů 3D Warehouse [25] obsahuje několik balíčků obsahujících modely svislého dopravního značení. Všechny modely na tomto portálu jsou ve formátu pro aplikaci SketchUp a jsou volně k užití. Formát skp bohužel není podporován v Unity 3D, proto bylo nutné nejprve vyexportovat modely v programu SketchUp do formátu Collada. Následně byly upraveny značky v programu Blender (aby byly značky z různých balíčků stejně vysoké a celkově vzhledově jednotné). Obdobně jako u vodorovného dopravního značení bylo nutné převést finální modely do formátu fbx a následně je importovat do Unity 3D. Na obrázku 5.4 můžete vidět ukázkový balík z 3D Warehouse, který obsahuje pouze svislé dopravní značení kruhového tvaru.



Obrázek 5.4: Ukázka značek z balíku modelů [15]

6 Implementace

V následujících podkapitolách jsou rozebrány principy a postupy převedení vytvořeného návrhu do funkční aplikace pomocí Unity 3D. Pro lepší pochopení některých stěžejních algoritmů je využito znázornění pomocí blokových schémat.

6.1 Základní struktura projektu v Unity 3D

Projekt je rozdělen do osmi složek: Code, GUI, Models, Prefabs, Scene, Sounds, Standard Assets a Terrains. Složka Code obsahuje veškeré C# skripty a xml soubory s daty, které se používají v simulátoru 3D Autoškoly, a je rozdělena do dalších čtyř podsložek: Interfaces, Scripts, States a XmlData.

Obsah jednotlivých skriptů je rozebrán v dalších kapitolách. GUI obsahuje obrázky ve formátu PNG, které se používají v grafickém uživatelském rozhraní 3D Autoškoly, například rychloměr, logo aplikace a jiné. Jak již název napovídá, ve složce Models se nacházejí 3D modely používané ve scéně simulátoru. Modely jsou rozděleny do složek Buildings, Roads, Mini (obsahuje model automobilu Mini) a Signs. Veškeré modely jsou ve formátu fbx. Adresář Prefabs obsahuje objekty typu Prefab. Prefab je znovupoužitelný Game Object uložený v Project View. Prefaby mohou být vkládány do libovolného počtu scén a to v jakémkoli počtu. Když se vloží Prefab do scény, vytvoří se instance. Všechny instance jsou propojeny s původním Prefabem [29]. Ve složce Prefabs jsou například uloženy pomocné objekty popsané v kapitole 6.3. Adresář Scene obsahuje soubory se scénami pro uživatelské menu a město pro jízdu v 3D Autoškolě. Sounds zahrnuje zvuky, které jsou použity během užívání aplikace. Zvuky jsou uloženy ve standardním formátu WAV. Složka Standard Assets je zde implicitně a obsahuje sadu různých objektů, skriptů a jiné, které je možné využívat i upravovat pro vlastní účely. V Adresáři Terrains lze najít soubor, který reprezentuje terén, jenž je použit v obou scénách aplikace 3D Autoškola.

6.1.1 Způsob programování v Unity 3D

Jak již bylo zmíněno v kapitole 3.1.3, v Unity 3D se programuje pomocí skriptů v jazycích C#, Javascript nebo Boo. Pro implementaci 3D Autoškoly byl zvolen jazyk C#, a proto jsou všechny vytvořené skripty pouze v tomto jazyce a popisovaný přístup k programování taktéž.

Tělo skriptu je C# třída, která implicitně obsahuje dvě metody Start a Update. Start je určen pro inicializaci a Update je volán s každým snímkem spuštěné aplikace. Třidu lze rozšiřovat o vlastní metody a také je možné přidávat nebo ubírat standardní metody Unity 3D (například FixedUpdate, LateUpdate, OnGui a jiné). Uvnitř třídy lze také vytvářet proměnné, do kterých vkládáme Game Objecty přímo ze scény, a ty pak můžeme pomocí skriptu upravovat nebo s nimi manipulovat.

Po vytvoření zamýšlené funkcionality lze skript přiřadit některému Game Objectu ve scéně, a tímto je zajištěno, že skript bude spuštěn při nahrání scény, ve které se nachází příslušný Game Object. Ze skriptu, přiřazeného ke Game Objectu, lze snadno přistupovat k vlastnostem daného Game Objectu, jeho komponentám a případně k jeho podobě.

6.2 Tvorba města

Při návrhu města bylo hlavním záměrem, aby model obsahoval co největší koncentraci v simulátoru implementovaných silničních pravidel. Jednotlivé části města a jejich značení jsou vytvořeny podle skutečných křižovatek a komunikací. Systém dopravního značení byl získán virtuálním procházením ulic pomocí Google Street View[30] a vytvořením zjednodušeného schématu, které je možné zreprodukovat do simulátoru pomocí balíku značek a modulárního systému silnic viz. kapitola 5.

Stanovené požadavky na strukturu města byly především takové, aby město obsahovalo jeden či více typů komunikací a značení z tohoto výčtu:

- křižovatka se světelnou signalizací,
- jednosměrná komunikace,
- kruhový objezd,
- více proudová komunikace,
- zákaz zastavení,
- přechody pro chodce,
- zákaz vjezdu.

Na obrázku 6.1 můžete vidět již zhotovený model města z ptáčích perspektivy. Jsou zde vyznačeny obsažené komponenty z výše uvedeného výčtu.



Obrázek 6.1: Plán vytvořeného města z ptáčích perspektivy

6.2.1 Realizace návrhu města v Unity 3D

Modely silnic a značek, jejichž tvorba byla popsána v kapitole 5.2, jsou umístěny v Unity 3D ve složce Models a v jejich podsložkách pro silnice, svislé dopravní značení a budovy.

Nyní můžeme modely vkládat přímo do scény. Pokud například chceme vytvořit delší silnici se dvěma pruhy rozdělenou přerušovanou čarou, vložíme do scény příslušný model a ten následně pomocí zkratky ctrl+D duplikujeme a duplikáty přesuneme tak, aby na sebe navazovaly. Například

kruhový objezd je složen z 8 částí (4 odbočky a 4 rovné části s vyústěním). Po celém městě jsou budovy rozmístěny zcela náhodně, ale vždy kopírují okraje vozovek. Pro větší rozmanitost a lepší uživatelský zážitek byly po městě rozmístěny i stromy.

Některé značky jako „STOP dej přednost v jízdě“ byly seskupeny do nového objektu, který obsahuje i kolizní tělesa, která slouží pro určení dodržení předpisů pro danou značku. Konkrétní popis tvorby těchto pomocných těles je popsán v kapitole 6.3, kde jsou vysvětleny i další pomocná tělesa, například pro určení správného směru při jízdě jednosměrnou komunikací.

6.3 Pomocné objekty pro určení porušení pravidel silničního provozu

V Unity 3D je každý objekt ve scéně Game Object. Ke každému Game Objectu lze přidávat komponenty jako kolizní tělesa, skripty, fyzikální vlastnosti apod.

Lze vytvářet i prázdné herní objekty, které neobsahují žádný 3D model, a lze jim také přidávat komponenty. Pro účely aplikace 3D Autoškoly se nejvíce hodí prázdný objekt s Box Collider nastavený na isTrigger, což znamená, že tímto objektem lze projet nebo projít a při tomto projetí lze „odchytit“ události vstup a výstup z kolizního tělesa. Využití tohoto přístupu hned uvedu na názorném příkladu.

Potřebujeme určit, zda uživatel při jízdě zastavil na stopce. Pro tento případ stačí před křižovatkou vložit kvádrové kolizní těleso, kdy se při vstupu do tohoto tělesa začne sledovat, zda uživatel snížil rychlost na $0 \text{ km} \cdot \text{h}^{-1}$ a při výstupu se jen ověří zdali zastavil a vyvolá se případná chyba. V následujících podkapitolách uvedu další situace a jejich řešení pomocí kolizních těles.

6.3.1 Pomocná tělesa pro určení jízdy mimo vozovku

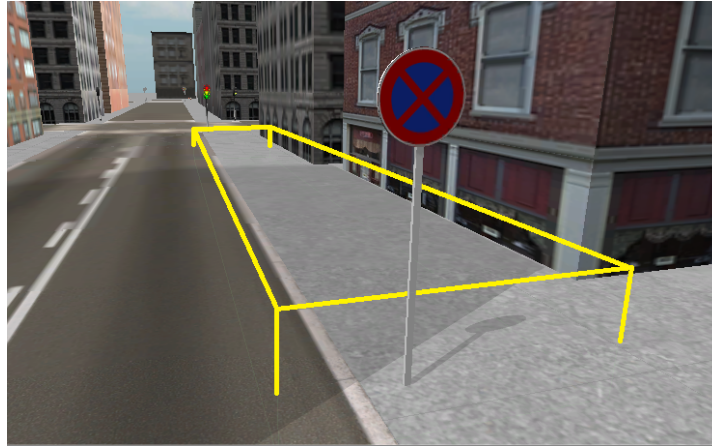
Aby uživatel nemohl beztrestně jezdit mimo vozovku, je vozovka ohraničena pomocí Box Collider, ale pro větší toleranci nejsou umístěny přímo u hrany. Pokud tedy uživatel sjede s vozidlem mimo dopravní komunikaci je o tom upozorněn. Kdyby i přes upozornění pokračoval dále v jízdě vedle komunikaci a dostal se na okraj města, tak jsou zde umístěny hraniční Box Collider a uživatel jimi nemůže projet.

6.3.2 Pomocná tělesa pro změny maximální povolené rychlosti

U hranice značek upravujících maximální povolenou rychlost je umístěno kolizní těleso zasahující pouze do pruhu, pro který je značka platná. Jméno tohoto tělesa je ve formátu SpeedLimit_XX, kde XX je maximální povolená rychlost v kilometrech za hodinu. Dále je možné umístit těleso s názvem SpeedLimitEnd, které po projetí uživatelem nastaví aktuální maximální povolenou rychlost na původní hodnotu (počáteční hodnota je přednastavená na $50 \text{ km} \cdot \text{h}^{-1}$, jelikož se jedná o jízdu městem)

6.3.3 Pomocná tělesa pro zákaz zastavení

Zde se jedná o opačný případ, než byl u dopravní značky „Stůj, dej přednost v jízdě“, kdy řidič nesmí zastavit u kraje vozovky, ale řešení je velmi podobné.



Obrázek 6.2: Box Collider vymezující zónu, kde se nesmí zastavit

Ke kraji vozovky se umístí Box Collider, který začíná u dopravní značky „Zákaz zastavení“ a bude končit před křižovatkou. Pokud úsek obsahuje odbočky, je možné vložit více takovýchto těles, aby kopírovaly okraj vozovky. Ukázka konkrétní situace je znázorněna na obrázku 6.2.

6.3.4 Pomocná tělesa pro jednosměrnou pozemní komunikaci

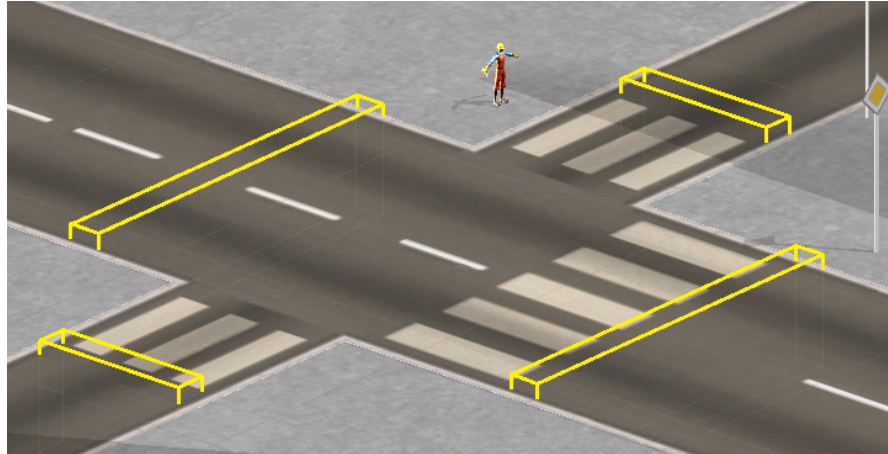
Určení směru jízdy není v 3D prostředí úplně triviálním problémem. Po delší úvaze bylo navrženo vlastní řešení pomocí dvou kolizních těles, která jsou do sebe z části zanořena. První těleso má název DirectionRight a druhé DirectionWrong. Tyto dvojice těles jsou rozmístěny po celé délce jednosměrné ulice.

Pokud vjede automobil do prvního tělesa (DirectionRight) a uvnitř „narazí“ do druhého tělesa (DirectionWrong), je vše v pořádku a přejde se znovu do počátečního stavu. Jestliže jako první narazí do DirectionWrong a posléze do DirectionRight, systém tímto pozná, že auto jede v protisměru a zobrazí se chybová zpráva.

6.3.5 Pomocná tělesa pro projetí křižovatkou

Jak se řeší nezastavení na stopce, bylo již uvedeno, a nyní bude popsáno řešení problému, jak ověřit, že uživatel dával správné znamení o změně směru při odbočování nebo například dával znamení o změně směru a přitom projel křižovatkou rovně.

Ke každé hranici křižovatky se umístí Box Collider, a jejich jména budou CrossRoadBoarderX, kde X je číslo 0-3. Příklad takové křižovatky a jejich těles můžete vidět na obrázku 6.3. Pokud automobil vjede do křižovatky přes CrossRoadBoarder0 a dává znamení, že odbočuje vpravo, pak musí vyjet z křižovatky přes těleso CrossRoadBoarder1.



Obrázek 6.3: Objekty pro určení správného projetí křižovatkou

6.3.6 Pomocná tělesa pro světelnou křižovatkou

Světelná křižovatka obsahuje stejná tělesa jako normální křižovatka viz výše, ale je rozšířena o další čtyři kolizní tělesa, která se nachází u hranic křižovatek, ale pouze v pruhu, kterým se vjíždí do křižovatky. Při výstupu z tohoto tělesa se pomocí skriptu zjistí, jaký právě svítí světelný signál na semaforu, a pokud se jedná o červenou, tak je z toho vyvozena příslušná chyba.

6.4 Implementace stavového automatu pro řízení aplikace

Jak již bylo zmíněno v kapitole 4.2, aplikace 3D Autoškoly využívá pro řízení aplikace (přepínání mezi obrazovkami menu, nahrávání scén a jiné) stavového automatu. V následujících podkapitolách je popsána implementace tohoto automatu a jednotlivých stavů, kterých může dosahovat.

6.4.1 Správce stavů a rozhraní stavů

Správce stavů se nachází ve skriptu StateManger. Třída StateManger zahrnuje proměnnou activeState typu IStateBase, která obsahuje aktuální stav aplikace. Pokud je potřeba změnit stav na jiný, tak se nový stav přiřadí do této proměnné pomocí metody SwitchState. V metodě Update se pak periodicky volá metoda StateUpdate aktuálního stavu, v níž se nachází logika tohoto stavu (například odchyťování událostí kláves aplikace a následná reakce na tuto událost). Obdobně se v metodě OnGui volá periodicky metoda ShowIt aktuálního stavu, která má na starost vykreslování a interakci s grafickým uživatelským rozhraním pro daný stav.

Důležitou funkcí StateManageru je zachování objektu, ke kterému je přiřazen během přepínání mezi scénami. Standardně se po nahrání nové scény smažou všechny objekty původní scény a nahrají se nové. Tímto způsobem by nebylo možné například nastavit v menu barvu automobilu a poté tuto informaci přenést do jízdy ve městě odehrávající se ve druhé scéně. Z těchto důvodů se v metodě Awake zabrání zničení objektu při nahrávání, a to metodou DontDestroyOnLoad(GameObject).

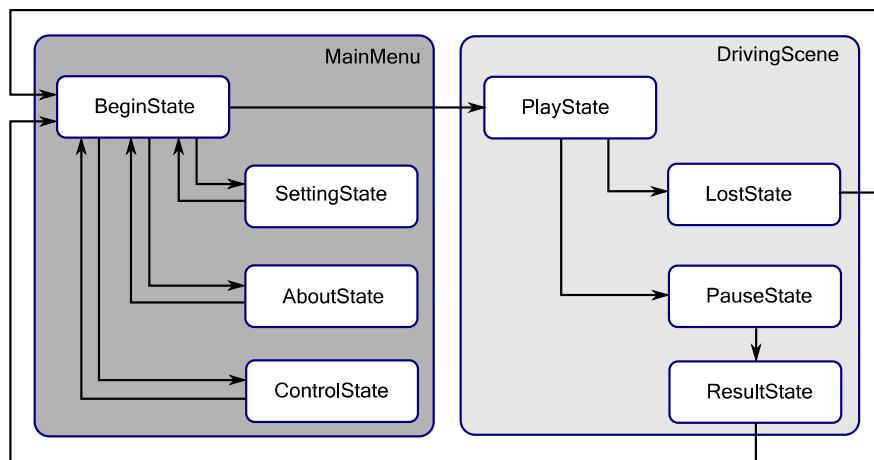
Pro vložení správce stavů do aplikace 3D Autoškola, je ve scéně MainMenu vytvořen Game Object pojmenovaný GameManger a skript StateManager je k němu přidán jako komponenta. Dále se do objektu GameManager vloží model automobilu i s jeho komponentami a kamery užívané v aplikaci. Vše, co se nachází v GameManageru, je přenášeno mezi scénami včetně komponent, proto se nemusí do scény určené pro jízdu znovu vkládat automobil a kamery.

Na obrázku 6.4 je zobrazen stavový automat aplikace, na kterém je znázorněno i v jaké scéně se nachází jednotlivé stavy.

6.4.2 Stavy aplikace

V předchozí kapitole zmíněný IStateBase je rozhraní, které je implementováno v jednotlivých stavech aplikace. Rozhraní obsahuje dvě metody StateUpdate a ShowIt, jejichž funkce byla popsána výše.

Po spuštění aplikace 3D autoškola je nahrána scéna MainMenu a nastaví se stav BeginState. Tento stav obsahuje rozcestník s tlačítky, jak bylo navrženo. V konstruktoru stavu BeginState se nahraje scéna MainMenu v případě, že se tak již nestalo, a nastaví se poloha automobilu, před kameru pro menu a vypnou se veškeré komponenty, které nejsou současně potřebné (například TrafficRulesControl, PlayerControl a jiné). Dále se aktivuje pouze jedna kamera a to CameraMenu. Metoda ShowIt zajišťuje vykreslení a interakci s jednotlivými tlačítky rozcestníku. Pokud je aktivováno některé z tlačítek měnících stav, je pomocí metody SwitchState přepnuto na nový stav. Tlačítko „Konec“ ukončí aplikace a tlačítko „Start“ nejprve nahraje scénu DrivingScene a poté přepne stav na PlayState.



Obrázek 6.4: Stavový automat aplikace

Stav SettingsState se přepne po stisknutí tlačítka „Nastavení“. V tomto stavu si může uživatel změnit barvu automobilu nebo jazyk aplikace. Konstruktor získá referenci na komponentu PlayerControl (tento skript umožňuje změnu barev automobilu) a uloží si aktuální barvu a jazyk aplikace, pro případ, že uživatel nebude chtít uložit změněné nastavení. Metoda ShowIt zajišťuje vykreslení a interakci všech tlačítek a ostatních prvků GUI. Tlačítka „Uložit“ a „Zrušit“ změni stav aplikace zpět na BeginState, ale v jednom případě se uloží změny do proměnných s aktuální barvou a jazykem a v druhém případě se naopak nastavení zruší a nastaví se původní hodnoty.

Stavy AboutState a ControlState jsou pouze informativního charakteru, tedy obsahují pouze jedno tlačítko pro návrat zpět do hlavního menu (stavu BeginState), jinak se vykreslují pouze textová

pole (GUI.Box) s informacemi o ovládní a o aplikaci. Pro návrat do menu z těchto stavů lze také použít tlačítko *Escape*, což je zajištěno v metodě *StateUpdate*.

Konstruktor stavu *PlayState* aktivuje potřebné komponenty pro jízdu a kontrolu pravidel a také aktivuje kameru pro sledování automobilu *CameraCar*. Konstruktor obsahuje na rozdíl od ostatních konstruktorů stavů argument *clearData* typu *bool*. Pokud je tento argument nastaven na *true*, zavolá se v konstruktoru metoda *ClearData* u komponenty *TrafficRulesControl* a tím se zajistí, že výsledek jízdy nebude obsahovat data z předchozích jízd. Metoda *StateUpdate* zjišťuje, zda uživatel nestiskl klávesu *Escape*. V kladném případě se přepne stav na *PauseState*. Dále se kontroluje, jestli nebyla nastavena proměnná *HitPedestrian* na *true* (uživatel srazil chodce). V případě, že ano, automat se přepne do stavu *LostState*.

PauseState nastaví v konstruktoru *Time.timeScale* na 0, čímž se pozastaví jízda. *Time.timeScale* se používá také na efekty zpomalení, například hodnota 0.5 by způsobila, že by hra běžela dvakrát pomaleji než je reálný čas [28]. Dále se v konstruktoru vypne zvuk motoru, protože kvůli pozastavení času by běžel v nepříjemné smyčce. V metodě *ShowIt* je pak vykreslen rozcestník, kdy má možnost uživatel pokračovat v jízdě, ukončit jízdu nebo ukončit hru. Stisknutí tlačítka „Pokračovat“ se znovu nastaví hodnota *Time.timeScale* na 1 a přepne se do stavu *PlayState*. Ukončení jízdy přepne uživatele do stavu *ResultState*.

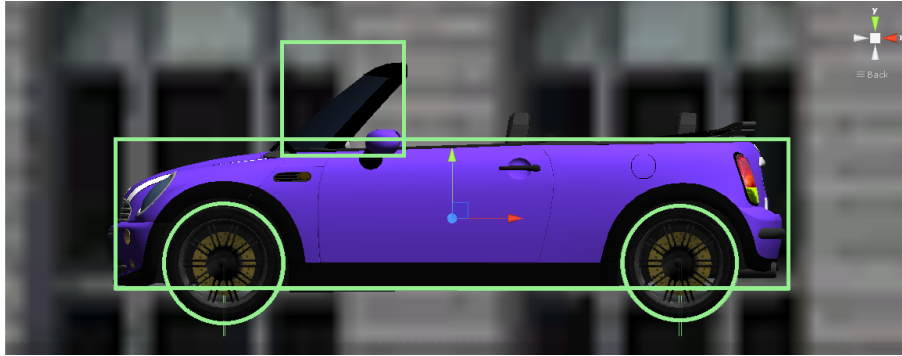
ResultState získá v konstruktoru referenci na komponentu *GameData*, kam se ukládá počet přestupků porušených během jízdy. V metodě *ShowIt* se pak vykreslí přehledná tabulka s názvem přestupku, počtem porušení a hypotetickou pokutou za takovýto počet prohřešků. Uživateli je nabídnuto tlačítko pro návrat do menu, nebo může stisknout klávesu *Escape*.

Posledním stavem, do kterého se může uživatel dostat, je již zmíněný *LostState*, který nastane při přejetí jezdce. Stav pozastaví hru a zobrazí informaci o trestném činu s tlačítkem „Ukončit jízdu“, které přepne stav do hlavního menu, tedy *BeginState*.

6.5 Základní fyzikální model automobilu

6.5.1 Vložení do scény a úprava fyzikálních vlastností

Automobil se do scény vloží pouhým přetažením. Aby se auto chovalo podle fyzikálních zákonů, musí se vložit do modelu komponenta *Rigid Body*. V této komponentě lze následně nastavit vlastnosti, jako je hmotnost, odpor vzduchu při pohybu, zapnutí působení gravitační síly na těleso a možnost deaktivace fyzikálního enginu pro těleso. Hmotnost byla následně nastavena na 1305kg (pohotovostní hmotnost *Mini Cooper Cabrio* dle [31]), odpor vzduchu na 0.05 (Tato hodnota nemá v dokumentaci *Unity 3D* uvedeny jednotky, ale je zde uvedeno, že 0 znamená žádný odpor vzduchu a při nekonečnu objekt ihned zastaví) a působení gravitační síly bylo zapnuto. Pokud se v tomto stavu spustí hra, automobil propadne terénem, protože neobsahuje žádné kolizní těleso, které by mohlo interagovat s povrchem terénu.



Obrázek 6.5: Rozvržení kolizních těles pro model automobilu

Je nutné tedy dodat automobilu kolizní tělesa, která pokryjí povrch automobilu. Pro tyto účely lze zvolit i takzvaný Mesh Collider, kdy je kolizní těleso reprezentováno sítí daného modelu, ale především z důvodu výpočetní náročnosti se doporučuje tento typ kolizních těles užívat co nejméně. Vhodnější je složit základní kolizní tělesa (Box Collider, Sphere Collider a Capsule Collider), tak aby přibližně kopirovala povrch modelu [32]. Jak je ukázáno na obrázku 6.5, pro kolize automobilu je použito dvou Box Colliderů. Kola mají speciální Wheel Collider, které jsou speciálně určené pro kola vozidel.

6.5.2 Skript pro pohyb a ovládání automobilu

Nyní již auto nepropadne terénem po spuštění hry, ale stále není schopné pohybu. Pro tyto účely byl vytvořen skript, který se poté přiřadil k automobilu jako jeho komponenta (tímto způsobem je skript automaticky spuštěn při startu aplikace a ze skriptu se může snáze přistupovat k jednotlivým komponentám či částem modelu).

Cílem práce není vytvořit dokonalý fyzikální model pohybu a ovládání automobilu, ale především kontrola dopravních předpisů zaměřených na výuku, a proto je model značně zjednodušený. Při vytváření skriptu pro ovládání automobilu byly částečně využity informace ze série tutoriálů o tvorbě závodní hry [33].

Skript manipuluje především s koly automobilu, a to jak s Wheel Collider, tak s modely kol (3D modely jsou v Unity 3D a jejich skriptech reprezentovány typem Transform). Wheel Collider jsou určeny přesně pro tyto účely, tedy modelování kol vozidel. Použitý princip rozpoohybování automobilu je následující: Pokud uživatel drží tlačítko šipka dopředu nebo dozadu, nastaví se zadním Wheel Colliderům kroutivá síla (N·m) Motor Torque a tímto se automobil rozjede dopředu nebo dozadu v závislosti na znaménku hodnoty kroutivé síly. Při stisknutí šipek vlevo nebo vpravo se změní úhel předních kol a automobil je tak schopen odbočovat. V objektu Wheel Collider je myšleno i na brzdění vozidla a kolu lze nastavit Brake Torque. Stisknuté tlačítko mezerník tedy způsobí, že se nastaví Motor Torque zadních kol na 0 a zároveň se nastaví Brake Torque, a auto začne postupně brzdit a poté zastaví.

Popsaná funkcionality je implementována především v metodě FixedUpdate, kde se kontrolují stisknutá tlačítka, a na základě jejich stisknutí se provedou potřebná nastavení Wheel Colliderů. Dále se v této metodě vypočítává aktuální rychlost automobilu, která je potřebná i pro další

skripty a kontrolu dodržování některých dopravních pravidel (maximální povolená rychlost, zákaz zastavení apod.). Rychlost se vypočítá pomocí rovnice 6.1.

$$v = 2\pi \cdot r \cdot \omega \cdot 0.06 \quad [\text{km} \cdot \text{h}^{-1}] \quad (6.1)$$

Kde r je poloměr kola a ω je úhlová rychlost kola v otáčkách za minutu. Vynásobením 0.06 převedeme rychlost na kilometry za hodinu.

Aktuální rychlost automobilu je také použita pro určení zařazeného stupně. Rozmezí jednotlivých stupňů řazení a jejich rychlostní rozsahy můžete vidět v tabulce 6.1.

Tabulka 6.1: Tabulka rychlostních stupňů a jejich rozmezí v $\text{km} \cdot \text{h}^{-1}$

Rychlostní stupeň	Minimální rychl. [km·h⁻¹]	Maximální rychlost [km·h⁻¹]
Zpátečka	< 0	-50
Neutrál	0	0
1	> 0	20
2	> 20	40
3	> 40	60
4	> 60	80
5	>80	155

Správné řazení rychlostí u automobilu se určuje pomocí otáček, ale jelikož v aplikaci 3D autoškola je pouze automatická převodovka a zobrazené rychlosti jsou pouze informativní, je pro zjednodušení rychlostní stupeň určen pomocí výše uvedené tabulky. Data uvedená v tabulce byla vytvořena pomocí autorových zkušeností s řízením motorového vozidla.

S řazením souvisí také zvuk motoru, který se mění se zvyšující se rychlostí a změnami rychlostních stupňů. Do objektu auta byla tedy přidána další komponenta a to Audio Source. Na serveru freesound.org[35] byl zvolen jeden z mnoha zvuků motoru (z důvodů absence zvuku pro automobil Mini, byl vybrán zvuk motoru Ford Mustang). Se zvyšující se rychlostí se zvyšuje výška (Pitch) zvuku motoru. Tato funkcionality je zajištěna voláním metody EngineSound v metodě FixedUpdate. V metodě EngineSound se nejprve zjistí minimální a maximální hodnota aktuálního rychlostního stupně (viz. Tabulka 6.1) a následně se vypočítá výška zvuku pomocí vzorce:

$$\frac{(\text{aktuální rychlost} - \text{minimální rychlost})}{(\text{maximální rychlost} - \text{minimální rychlost})} + 1 \quad (6.2)$$

Pokud je aktuální rychlost neutrál, je výška automaticky zvolena jako 1. V případě rychlostního stupně zpátečka je nastaveno rozpětí minimální a maximální rychlosti na 10-50 $\text{km} \cdot \text{h}^{-1}$, protože při ponechání záporných hodnot dochází k nepříjemnému zkreslení zvuku. Výsledek ze rovnice 6.2 je vložen do proměnné Pitch objektu Audio Source.

6.6 Grafické uživatelské rozhraní přístrojové desky

Na obrázku 6.6 jsou čtyři hlavní prvky uživatelského rozhraní spojené s ovládáním automobilu. Jedná se o rychloměr, řadící páku a zpětné zrcátko. O dynamické zobrazení rychloměru a symbolu řadící páky se stará skript DrivingGUI a zpětné zrcátko je zobrazeno pomocí skriptu RearView.



Obrázek 6.6: Grafické uživatelské rozhraní automobilu

Analogový rychloměr se skládá z ciferníku s hodnotami rychlostí a ručičky. Grafické podklady jsou převzaty z webu [34]. Obrázek byl upraven v editoru Gimp, kde byla oddělena ručička od ciferníku a dopsány hodnoty rychlostí.

Ve Skript DrivingGUI jsou nejvíce podstatné metody FixedUpdate a OnGUI. Metoda OnGUI je volána pro vykreslování a zpracování GUI událostí, může být volána i několikrát za snímek [37]. Nejprve vykreslíme ciferník a posléze ručičku. Ručička se musí otáčet podle aktuální rychlosti, proto se nejprve zjistí aktuální rychlost a poté se z ní vypočítá úhel, o který se má ručička otočit. Rychlost vydělíme 180 (to je maximální hodnota na rychloměru) a poté pomocí funkce Mathf.Lerp tuto hodnotu interpolujeme mezi hodnotami 0 a 270 (270° je úhel sevřený ručičkou ukazující na 0 a ručičkou ukazující na 180). Tímto se získá úhel, o který se pak otočí ručička pomocí funkce GUIUtility.RotateAroundPivot.

V metodě FixedUpdate se kontroluje, zda uživatel stiskl tlačítko pro signalizaci změny směru jízdy. Pokud ano, nastaví se proměnná blinker na true a časovač na 5 vteřin, což je doba po kterou bude kontrolka blikat.

Zobrazení symbolu řadící páky a jeho dynamického chování je dosaženo tak, že skript DrivingGUI obsahuje pole s 2D texturami, které obsahuje 7 textur pro jednotlivé stupně. V metodě OnGUI se získá hodnota aktuální zařazené rychlosti, což odpovídá indexu v poli s texturami, kde se vykreslí příslušná textura.

Jedním z rozdílů mezi volnou a placenou verzí Unity 3D je, že ve volné verzi není Render Texture, což je textura, na které lze zobrazit to, co vidí některá z kamer ve scéně. Z tohoto důvodu

nebylo možné vytvořit zpětná zrcátka přímo na modelu automobilu a bylo nutné zvolit jiné řešení. Tímto řešením je vykreslení pohledu kamery namířené na scénu za automobilem přímo na obrazovku jako menší obdélník vpravo nahoře.

6.7 Ostatní účastníci provozu

Jedním bodem zadání diplomové práce bylo vytvořit reprezentaci účastníků silničního provozu, v simulátoru 3D autoškoly. Takovým účastníkem provozu jsou například chodci pohybující se po chodníku. V následující podkapitole bude popsána implementace skriptu, který slouží pro pohyb chodců po chodnících v modelu města.

6.7.1 Reprezentace pohybu účastníků provozu

Standardní balík Assets Unity 3D obsahuje postavu dělníka typu CharacterControl. Tento Game Object obsahuje vše potřebné pro implementaci chodce: animace chůze, metody pro směřování postavy a její posun tímto směrem. Na obrázku 6.7 je zobrazen zmíněný dělník, jenž byl použit jako chodec ve výsledné aplikaci.



Obrázek 6.7: Chodec procházející se po chodnících křižovatky

Navržený skript pro chodce funguje tak, že se nastaví pole directions a pole degrees. Pole directions obsahuje časy chůze pro jednotlivé úseky a pole degrees obsahuje úhly zahýbání před úseky. Pokud nastavíme například pole directions prvky 10, 5, a pole degrees s prvky 90, -90, na výchozí pozici se otočí o 90 stupňů vpravo a půjde nejprve deset vteřin rovně, poté zahne o 90 stupňů doleva a půjde znovu rovně 5 vteřin, pak se otočí o 180 stupňů a vrátí se stejným způsobem. Takto chodí chodec neustále jednu trasu tam a zpět. Prvky dvou polí se nastaví podle potřeby pro danou křižovatku s přechody pro chodce nebo pro chodník.

Vlastní implementace se nachází v souboru Pedestrian.cs a skládá se ze dvou metod: Awake a Update. V metodě Awake se inicializují proměnné, do proměnné timer se uloží první hodnota z pole directions a do proměnné head se uloží první prvek z pole degrees. Poté se otočí model chodce o úhel heading.

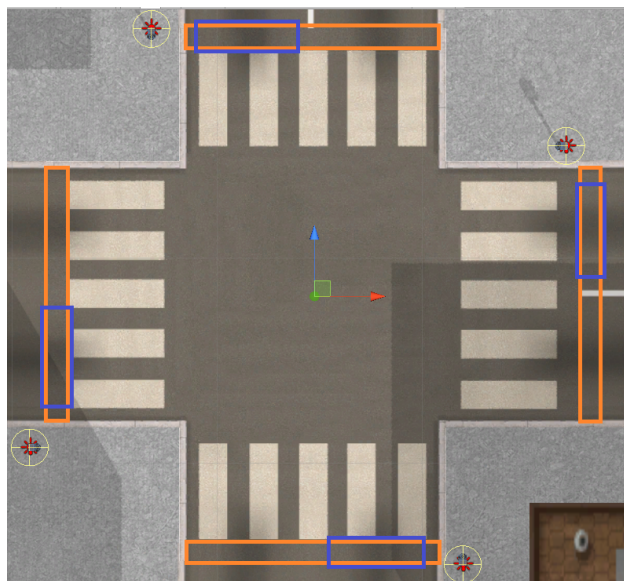
Při každém zavolání metody Update se sníží proměnná timer o Time.deltaTime, a pokud je aktuální hodnota timeru menší než nula, změní směr chůze. V případě, že jde chodec cestu tam (chodec chodí z bodu A do bodu B a zpět, cesta tam znamená z A do B) a nejedná se o poslední úsek

cesty, přičte se k heading následující prvek pole degrees a do timeru se uloží následující hodnota z pole directions (chodec odbočí a pokračuje v chůzi po určenou dobu). Pokud se jedná o poslední úsek, tak se od heading odečte 180, nastaví se proměnná back na true a do timeru se opět uloží hodnota posledního úseku z pole directions (chodec se otočí a jde poslední úsek znovu, ale opačným směrem). Při cestě zpět se chová algoritmus obdobně, jen prochází pole directions a degrees opačným směrem od proměnné heading se odečítají hodnoty z directions nikoli přičítají.

Po nastavení proměnné heading se vypočítá směrový vektor, který určuje směr chodcovi chůze. Tento vektor předáme objektu chodce pomocí metody Controller.SimpleMove(forward * speed), kde forward je zmíněný směrový vektor a speed je rychlost chůze.

6.8 Implementace světelné křižovatky

System semaforů na světelné křižovatce vyžaduje svůj vlastní skript pro správné přepínání světel. Strukturu křižovatky můžete vidět na obrázku 6.8. Skládá se ze čtyř semaforů, čtyř Box Colliderů CrossRoadBoarder (na obrázku jsou označeny oranžovou barvou a jsou určeny pro správné projetí křižovatkou, jsou tedy shodné s objekty pro křižovátku bez světelných signálů) a čtyř Box Colliderů LightBoarder0-3 (na obrázku jsou označeny modrou barvou). Semafony jsou sdruženy do dvou skupin Line1 a Line2, podle shodné změny barev. Všechny výše popsané Game Objecty jsou seskupeny v objektu TrafficLightCrossRoad, který obsahuje jako komponentu skript TrafficLightsControl. Tento skript zajišťuje přepínání světel semaforů každých 20 vteřin.



Obrázek 6.8: Světelná křižovatka s kolizními tělesy

Třída TrafficLightControl obsahuje dvě pole Game Objectů Lane1Lights a Lane2Lights. Do těchto polí přiřadí objekt Lights, který je obsažen v každém modelu semaforu a skládá se ze tří světel typu Point (GreenLight, OrangeLight a RedLight). V metodě Start, která slouží pro inicializaci, se u první linie semaforů zhasnou všechna světla kromě červené a ve druhé linii zůstanou rozsvícena pouze zelená světla. Metoda FixedUpdate při každé iteraci, kdy je volána, snižuje proměnnou counter o Time.deltaTime. Pokud je counter roven nule, nastaví se znovu na dvacet vteřin a zavolá se

coroutine `SetTrafficLightsToGo`, která jako první argument bere linii světel, jež se má nastavit na zelenou, a jako druhým argumentem je linie, která se nastaví na červenou. Pro určení, která linie se má kdy nastavit na zelenou či červenou, se používá proměnná `firstLane`, jež je nastavena na `true` pokud u první linie svítí zelená a na `false` v případě červené.

Coroutine `SetTrafficLights` zapíná a vypíná světla jednotlivých linií semaforů. Ke konkrétním světlům semaforu se přistupuje pomocí metody `FindChild`, která nalezne příslušné světlo pomocí jeho názvu, a poté lze změnit stav světla pomocí `gameObject.SetActive(bool state)`, kde se za `state` vloží `true`, pokud má světlo svítit a `false` v opačném případě.

6.9 Dodržování prioritních pravidel

Dodržování veškerých předpisů podporovaných aplikací 3D Autoškola je implementováno ve skriptu `TrafficRulesControl.cs`. Tento skript je obsažen jako komponenta v objektu `PlayerCar`. V následujících podkapitolách je popsáno, jakým způsobem byly implementovány jednotlivé kontroly pravidel silničního provozu.

6.9.1 Výpis zpráv o přestupcích

Jak bylo navrženo v kapitole 4.3.4, při porušení některého z pravidel je na obrazovku vypsána zpráva s dopravní značkou a popisem přestupku. Pro tyto účely bylo vytvořeno asociativní pole `messagesStates` typu `Dictionary`. Tento `messagesStates` obsahuje klíč typu `string` (jednoduchý název přestupku) a `bool` (je aktuálně porušován příslušný dopravní předpis?).

V následujících podkapitolách jsou popsány algoritmy pro kontrolu podporovaných předpisů. Tyto algoritmy nastavují prvky z `messagesStates` na `true`, pokud si přejí vypsát příslušnou hlášku o porušení předpisů. V metodě `OnGui` se pravidelně procházejí všechny prvky `messagesStates` a podle jejich hodnoty se vykreslují na obrazovku.

Hláška je vykreslena jako `GUI.Box` s texturou dopravní značky vlevo a popisem přestupku vpravo. Ukázkovou hlášku ze hry můžete vidět na obrázku 6.9.

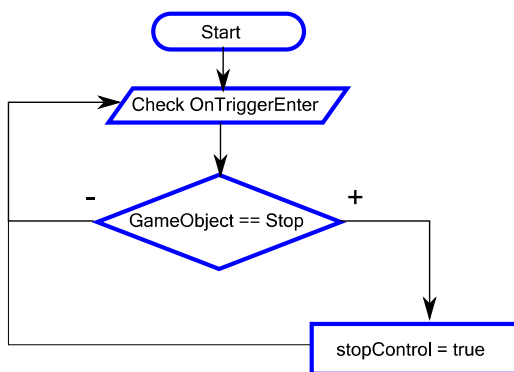


Obrázek 6.9: Ukázka zobrazení hlášek o porušení dopravních předpisů v 3D Autošcole

6.9.2 Kontrola zastavení na značce „Stůj, dej přednost v jízdě“

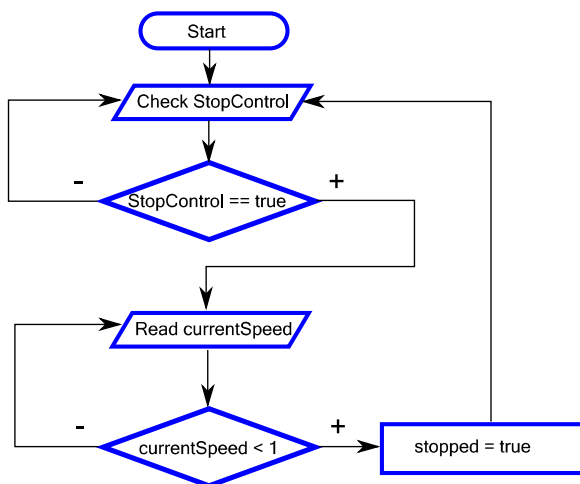
Kontrola zastavení před značkou „Stůj, dej přednost v jízdě“ je implementována ve třech paralelně spouštěných metodách: FixedUpdate, OnTriggerEnter a OnTriggerExit.

Na blokovém schématu na obrázku 6.10 je znázorněna implementace v metodě OnTriggerEnter. Metoda OnTriggerEnter(Collider someCollider) je volána pokud objekt (v našem případě automobil) narazí do nějakého Collideru (tento objekt je pak uložen v argumentu metody, tedy zde například someCollider), který je nastaven jako trigger. Po spuštění skriptu TrafficRulesControl se v metodě OnTriggerEnter zkontroluje, jestli se GameObject someCollider nejmenuje „Stop“, a pokud ano, tak se nastaví proměnná stopControl na true.



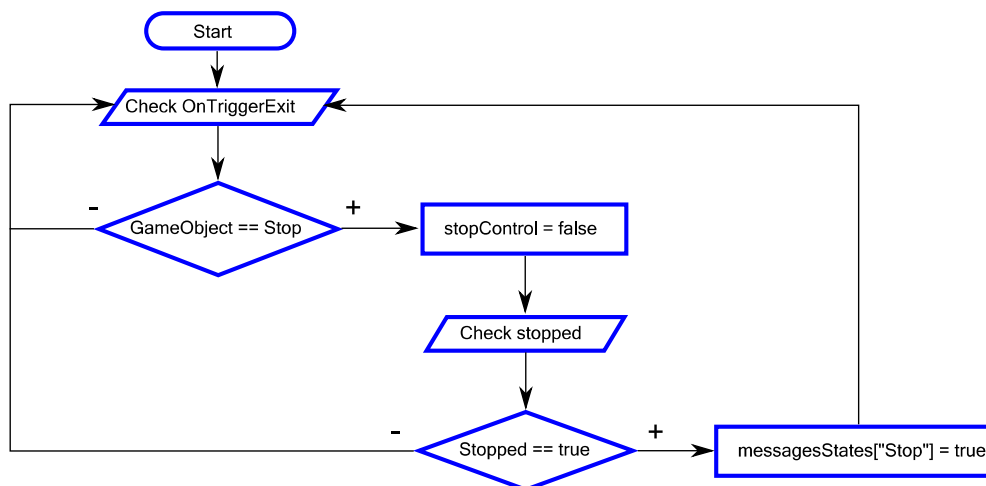
Obrázek 6.10: Blokové schéma, kontroly kolize s objektem Stop

Proměnná stopControl se kontroluje v metodě FixedUpdate. V případě, že je nastavena na true, získá se aktuální rychlost automobilu a pokud je hodnota rychlosti menší než $1 \text{ km} \cdot \text{h}^{-1}$, tak se nastaví proměnná stopped na true. Tímto se uchová informace o zastavení automobilu uvnitř BoxCollideru Stop, tedy zóně před křižovatkou, kde je řidič automobilu povinen zastavit. Na obrázku 6.11 je vyobrazeno schéma této kontroly uvnitř metody FixedUpdate.



Obrázek 6.11: Blokové schéma kontroly zastavení

Nyní je nutné zkontrolovat, zdali před vjetím do křižovatky auto zastavilo. Na obrázku 6.12 je opět znázorněno pomocí blokového schématu, jak byla implementována závěrečná kontrola v metodě OnTriggerExit.



Obrázek 6.12: Blokové schéma kontroly zastavení při opuštění kolizního tělesa Stop

Pokud uživatel opustí Game Object Stop, nastaví se proměnná stopControl na false, protože už není potřeba kontrolovat, zda automobil zastavil. Dále se zkontroluje proměnná stopped, tedy jestli mělo alespoň jednou vozidlo během jízdy v Game Objectu Stop rychlost $0 \text{ km} \cdot \text{h}^{-1}$. V případě, že má proměnná stopped hodnotu true, nastaví se v messagesStates hodnota prvku „Stop“ na true. O messagesStates se stará metoda OnGui, která vykreslí všechny zprávy, u kterých je nastaveno true.

Výše uvedené metody jsou volány buď v pravidelných intervalech, nebo pokud je vyvolá událost (např. OnTriggerEnter), a proto jsou blokové diagramy ve smyčce a nejsou ukončeny.

6.9.3 Kontrola porušení zákazu zastavení

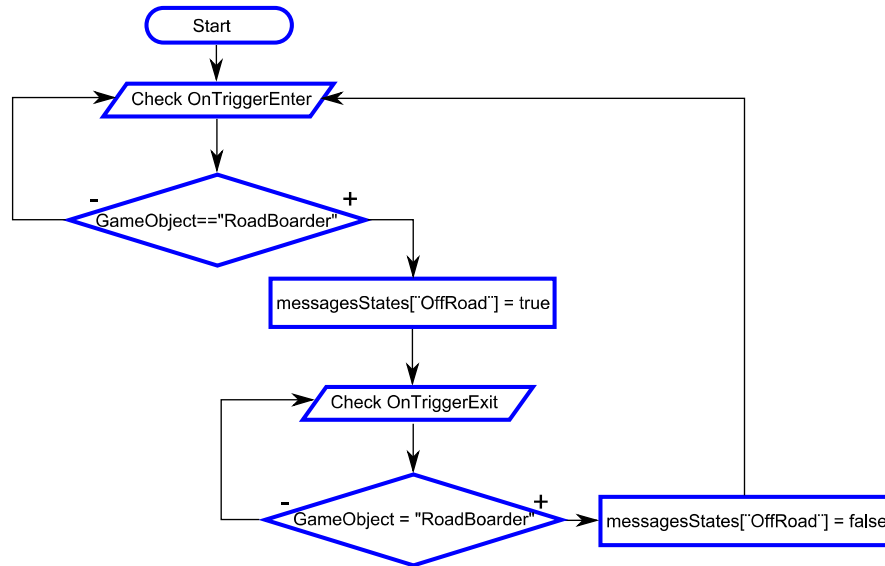
Obdobně jako byla v předchozí kapitole zóna, kde musí automobil zastavit, tak v případě zákazu zastavení existuje zóna, kde naopak uživatel nesmí zastavit své vozidlo. Kontrola je opět řešena v metodách FixedUpdate, OnTriggerEnter a OnTriggerExit.

Když uživatel koliduje s Box Colliderem, který má název „noStopping“ nastaví se proměnné stopControl a noStoppingControl na true a tím se zajistí kontrola v metodě FixedUpdate. Pokud je nastavená proměnná stopControl, provádí se stejná kontrola zastavení jako v případě značky „Stop, dej přednost v jízdě“ a když je k tomu povolena i proměnná noStoppingControl a dojde k zastavení, nastaví se v messagesStates hodnota prvku „NoStopping“ na true a vykreslí se zpráva o nepovoleném zastavení.

6.9.4 Kontrola jízdy mimo vozovku

Vytvořený model města obsahuje rovné komunikace, kolmé křižovatky a kolmé odbočky, proto lze kolem komunikací vytvořit BoxCollidery, které kopírují kraj vozovek a slouží pro detekci opuštění

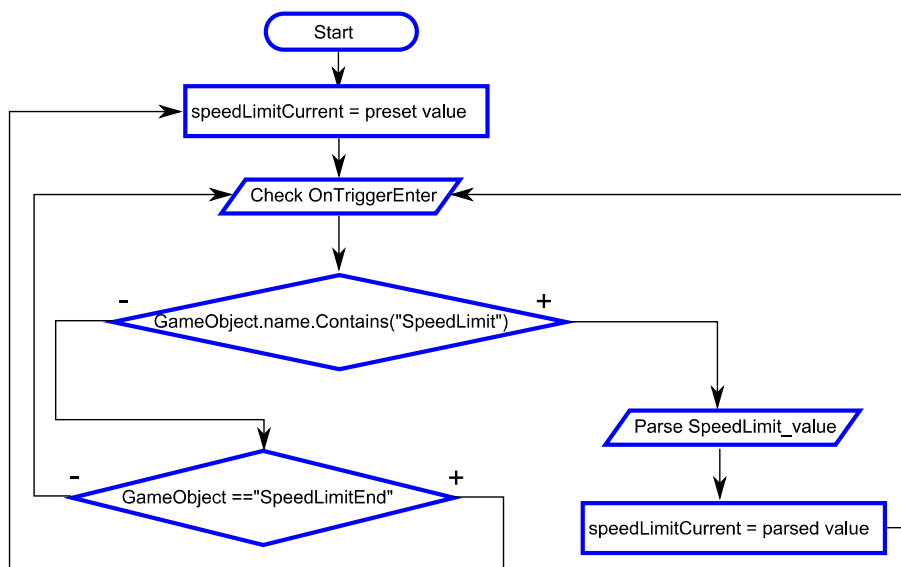
vozovky. Kontrola probíhá pouze v metodách OnTriggerEnter a OnTriggerExit. V případě, že automobil vyjede mimo vozovku a narazí do BoxCollideru, který se jmenuje „RoadBoarder“, nastaví se v messagesStates prvek „OffRoad“ na true a tím se zobrazí zpráva o jízdě mimo vozovku. Pokud uživatel opustí BoxCollider „RoadBoarder“, nastaví se zpět příslušný prvek v messagesStates na false a zpráva již není vykreslována. Tento algoritmus je znázorněn na obrázku 6.13.



Obrázek 6.13: Blokové schéma kontroly jízdy mimo vozovku

6.9.5 Kontrola překročení maximální povolené rychlosti

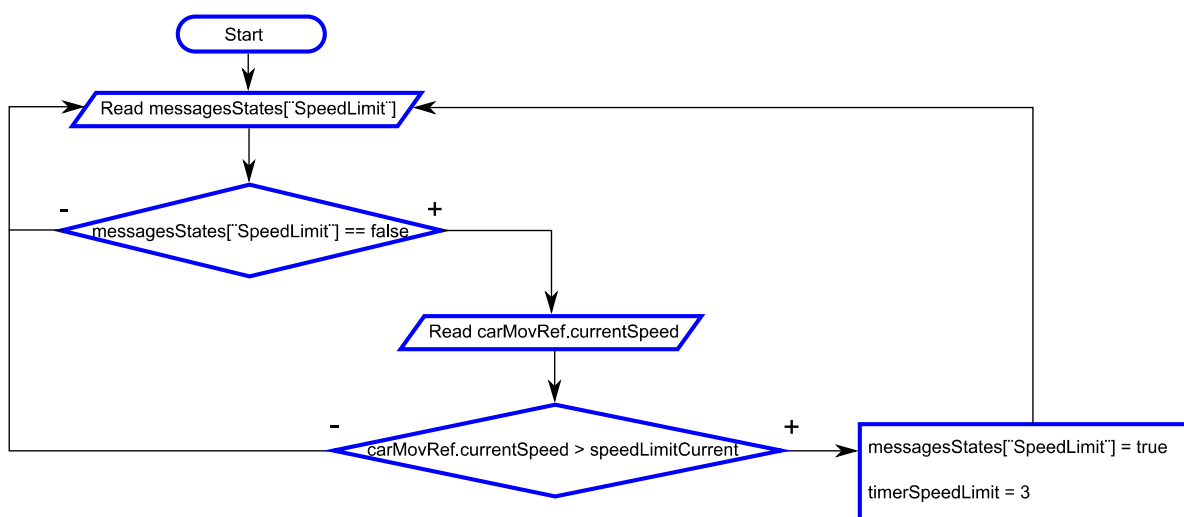
Primárně je maximální povolená rychlost nastavena na 50 km·h⁻¹, protože se simulace odehrává ve městě. Tato hodnota je uložena v proměnných speedLimit a speedLimitCurrent. Hodnota speedLimit zůstává neměnná a speedLimitCurrent se mění podle maximální povolené rychlosti na komunikaci, po které právě jede uživatel.



Obrázek 6.14: Blokové schéma změny aktuální povolené rychlosti

Ve skriptu TrafficRulesControl se v souvislosti s rychlostí řeší dvě věci: kontrola změny aktuální povolené rychlosti a kontrola porušení maximální povolené rychlosti. První je zajištěna v metodě OnTriggerEnter a je znázorněna na obrázku 6.14. Pokud uživatel projede objektem, v jehož názvu se nachází „SpeedLimit_“, z názvu se zjistí jakou hodnotu má dané omezení rychlosti (např. SpeedLimit_50 – znamená, že maximální povolená rychlost je 50 km·h⁻¹) a tato hodnota se nahraje do proměnné speedLimitCurrent. V případě kolize s Box Colliderem s názvem „SpeedLimitEnd“ se aktuální povolená rychlost vrátí na původní přednastavenou hodnotu, tedy do proměnné speedLimitCurrent se přiřadí hodnota z proměnné speedLimit.

Samotná kontrola překročení maximální povolené rychlosti je řešena v metodě FixedUpdate, kde se periodicky kontroluje současná rychlost vozidla s aktuální povolenou rychlostí. Pokud uživatel poruší předpis a překročí rychlost, nastaví se prvek „SpeedLimit“ v messagesStates na true a je vypsána zpráva o překročení povolené rychlosti. Tato zpráva je vypsána po dobu tří vteřin, ale pokud i po této době dochází k porušení maximální povolené rychlosti, prodlouží se doba trvání o další tři vteřiny.



Obrázek 6.15: Blokové schéma kontroly dodržování maximální povolené rychlosti

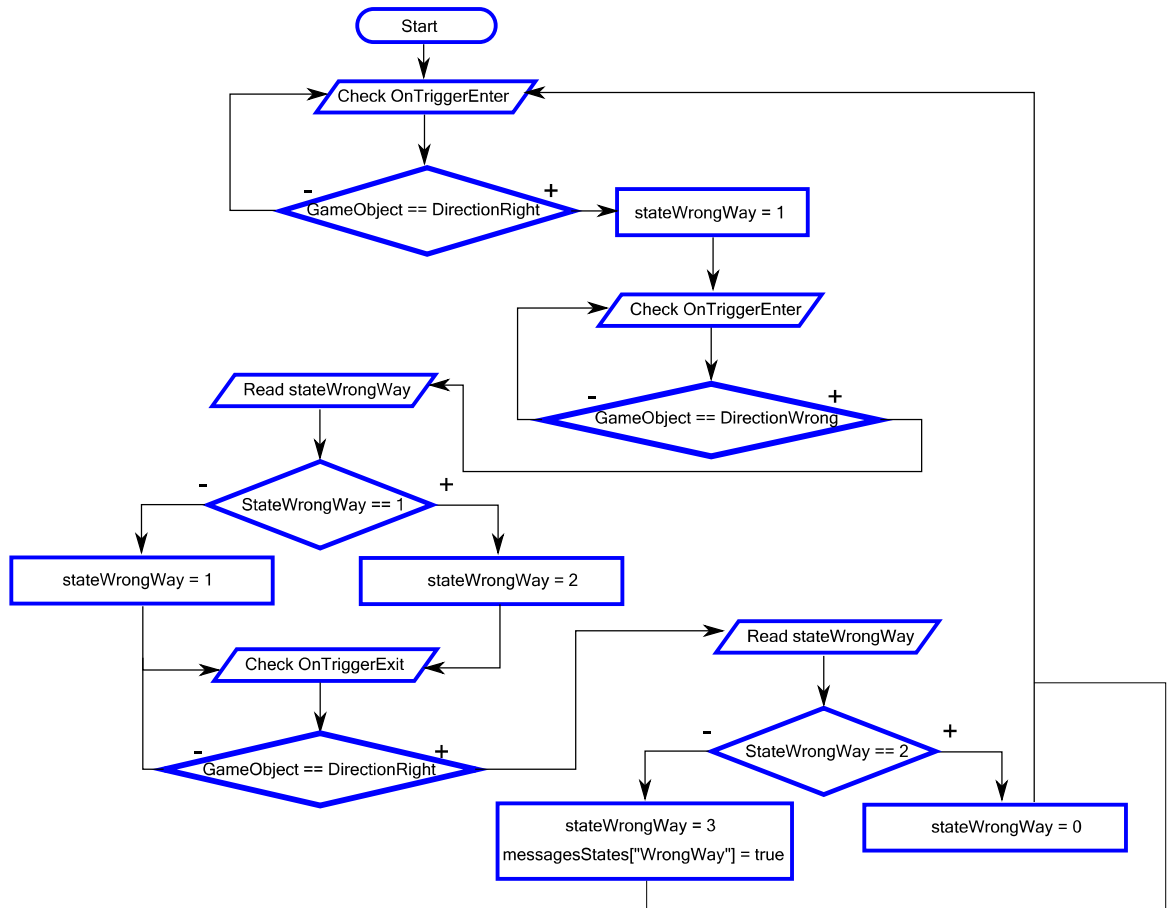
Na obrázku 6.15 je znázorněna tato kontrola pomocí blokového schématu. Ke kontrole tedy dochází, pouze v případě, že není aktuálně vypsána zpráva o porušení rychlosti, tedy messagesStates[“SpeedLimit”] je nastaven na false. Uvnitř schématu je uvedena i proměnná timerSpeedLimit, která je dekrementována ve FixedUpdate o Time.deltaTime a pokud je nulová, nastaví se i messagesStates[“SpeedLimit”] na false, čímž je zajištěno vykreslení po určenou dobu.

6.9.6 Kontrola směru jízdy po jednosměrné komunikaci

Jak bylo popsáno v kapitole 4.2.1, kontroly dodržování správného směru jízdy v jednosměrné komunikaci je dosaženo pomocí dvou do sebe zanořených Box Colliderů (DirectionRight a DirectionWrong). Kontrola je implementována v metodách OnTriggerEnter a OnTriggerExit.

Na obrázku 6.16 je znázorněn postup algoritmu kontroly pomocí blokového schématu. Kolize uživatele s Box Colliderem vyvolá metodu OnTriggerEnter a v případě, že jméno tohoto objektu je

„DirectionRight“, nastaví se stavová proměnná stateWrongWay na 1. Pokud poté uživatel narazí do tělesa „DirectionWrong“ a jeho předchozí stav byl 1, přejde se do stavu 2 (nastaví se proměnná stateWrongWay na 2). Poté se při opuštění prvního Box Collideru RightWay zkontroluje, jaký je současný stav v proměnné stateWrongWay. V případě, že je stav 2, přejde se do stavu 0 (počáteční stav, protože projetí proběhlo vpořádku), v opačném případě se přejde do stavu 3 (špatné projetí) a v messagesStates se nastaví prvek „WrongWay“ na true, čímž se vypíše příslušná zpráva.



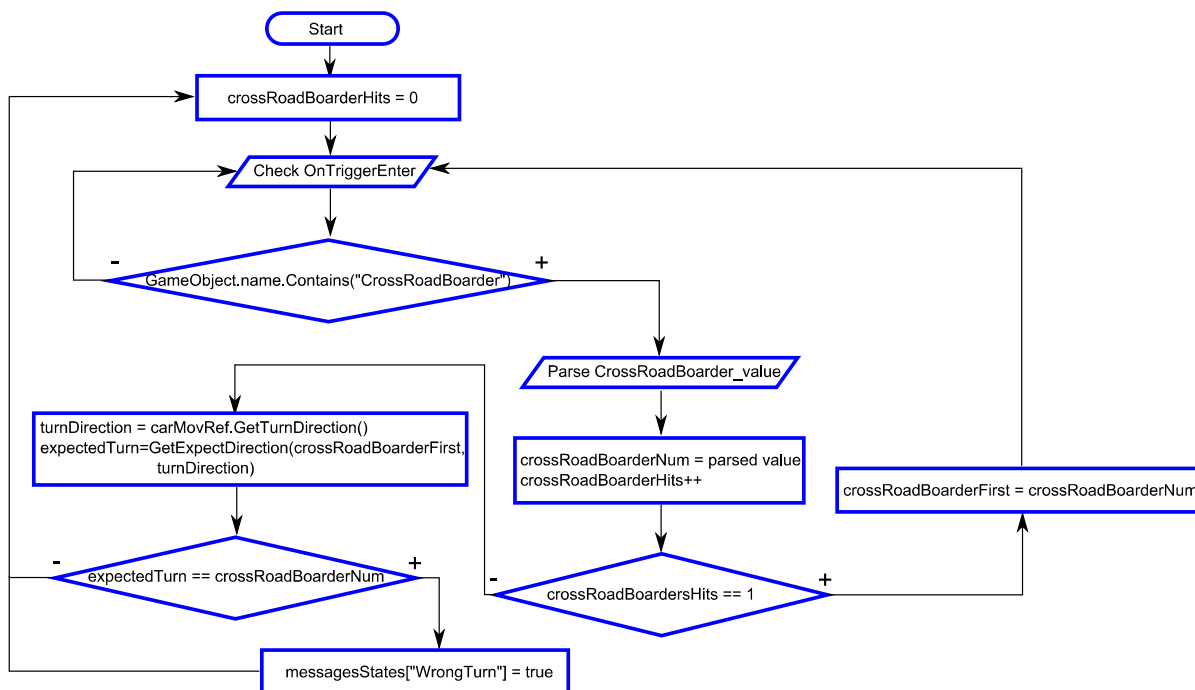
Obrázek 6.16: Blokové schéma algoritmu kontroly směru jízdy po jednosměrné komunikaci

6.9.7 Kontrola jízdy křižovatkou

U každého kraje křižovatky je jeden Box Collider s názvem crossRoadBoarder0-3, jak bylo popsáno v kapitole 6.3.5. Kontrola správného projetí křižovatkou je implementována pouze v metodě OnTriggerEnter. Alogritmus kontroly je znázorněn pomocí blokového schématu na obrázku 6.17.

Pro udržení informace o stavu projíždění křižovatkou jsou určeny proměnné crossRoadBoarderFirst (číslo hranice křižovatky z které automobil vyjíždí) a crossRoadBoarderHits (kolik Box Colliderů typu crossRoadBoarder automobil zasáhl). Při kolizi s objektem crossRoadBoarder se nejprve uloží číslo tohoto objektu a inkrementuje se počet zásahů v crossRoadBoarderHits. Pokud se jedná o první zasažené kolizní těleso, uloží se jeho hodnota do crossRoadBoarderFirst. V případě kolize s druhým tělesem se již kontroluje, zda uživatel správně

použil signalizaci o změně směru jízdy. Ze třídy carMovements se pomocí metody GetTurnDirection získá směr aktivované či neaktivované signalizace.



Obrázek 6.17: Blokové schéma algoritmu pro jízdu křižovatkou

Tato metoda vrací tři možné hodnoty: 1 – odbočení vpravo, -1 – odbočení vlevo a 0 - jízda rovně (neaktivovaná signalizace). Pomocí této návratové hodnoty se vypočítá předpokládaná hodnota, do jakého kolizního tělesa crossRoadBoarder měl uživatel narazit při vyjíždění z křižovatkou. Poté se srovná předpokládaná hodnota se skutečnou, a pokud se neshodují, vykreslí se chybová hláška o špatném projetí křižovatkou (v messagesStates se nastaví prvek „WrongTurn“ na true).

6.9.8 Kontrola jízdy světelnou křižovatkou

Světelná křižovátka obsahuje stejné objekty crossRoadBoarder jako normální křižovátka, a proto se o správné projetí křižovatkou (kontrola správného užití signálů o změně směru) stará stejná logika, která byla popsána v předchozí kapitole. Odlišností od předchozího druhu křižovatkou je kontrola, zdali uživatel nevjel do křižovatkou, když byl rozsvícen signál s červeným světlem „Stůj“. Tato kontrola se provádí pouze v metodě OnTriggerExit.

Jak bylo popsáno v kapitole 6.3.6, světelná křižovátka je rozšířena o kolizní tělesa lightBoarder. Pokud je zavolána metoda OnTriggerExit pro těleso obsahující název lightBoarder (tedy, že uživatel vjel do křižovatkou přes toto těleso), zjistí se jaké číslo je v názvu tohoto tělesa. Poté se získá reference na komponentu TrafficLightsControl, která se nachází u rodiče objektu lightBoarder. Z této reference, lze získat aktuální hodnotu proměnné firstLane, jež reprezentuje stav (true - signál se zeleným světlem „Volno“, false – signál s červeným světlem „Stůj“) první linie semaforů dané křižovatkou. Pokud bylo číslo objektu lightBoarder 1 nebo 3, nalézá se objekt v první linii, a pokud se jednalo jedno z čísel 0 a 2, pak je objekt v druhé linii. Chybová hláška se tedy vypíše

(nastaví se prvek „lightB“), pokud je firstLane true a lightBoarder je v druhé linii, nebo je firstLane false a lightBoarder je v první linii.

6.9.9 Interakce s ostatními účastníky provozu

Náraz neboli kolize ovládaného automobilu s jiným kolizním tělesem vyvolá funkci OnCollisionEnter s argumentem typu Collision, ve kterém je obsažen Game Object, do kterého se narazilo. Pro detekci chodce srážky s chodcem se tedy v metodě OnCollisionEnter zjistí, jaký je název sraženého objektu, a pokud je toto jméno „Pedestrian“ nastaví se ve třídě GameData proměnná pedestrianHit na true. Změna stavu proměnné pedestrianHit je zaznamenána ve stavu PlayState a ten na tuto změnu reaguje přepnutím do stavu LostState. Stav LostState pozastaví hru a informuje hráče, že se dopustil trestného činu tak vážného, že již nemůže pokračovat v aktuální jízdě a pomocí tlačítka se může vrátit do hlavního menu a případně spustit hru znovu. Na obrázku 6.18 je zobrazena zpráva, která se uživateli objeví při srážce s chodcem.

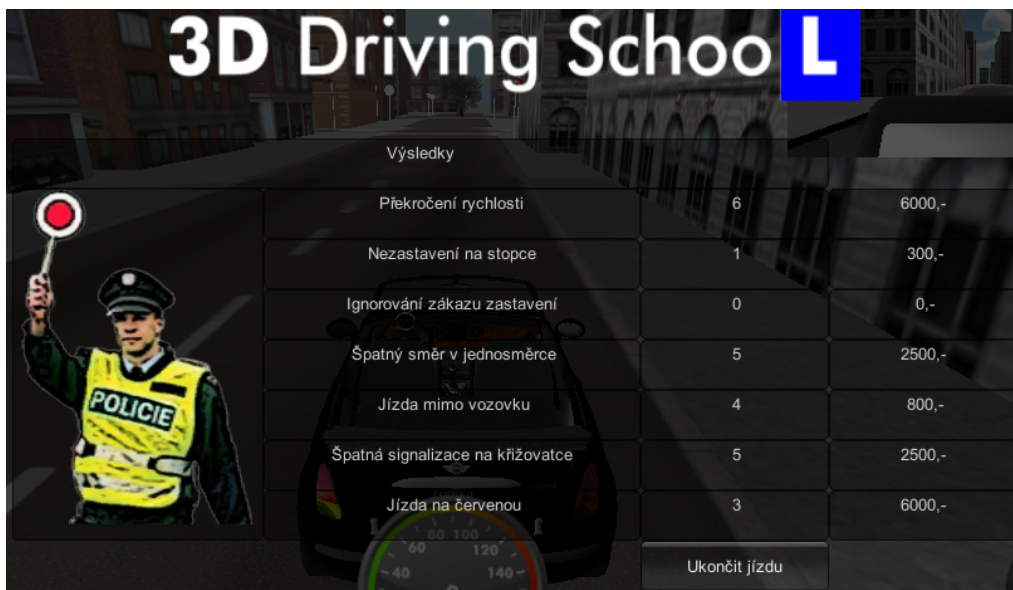


Obrázek 6.18 : Obrazovka stavu LostState vyvolaná kolizí s chodcem

6.9.10 Vyhodnocení jízdy

Vyhodnocení jízdy bylo navrženo způsobem, že po ukončení jízdy se uživateli zobrazí tabulka s přestupky a výší jejich hypotetické pokuty. Pro implementaci je nutné zajistit počítání porušení pravidel silničního provozu a následné vypočítání sumy za pokutování přestupků. V předchozích kapitolách, zabývajících se kontrolou jednotlivých přestupků, se při porušení silničního pravidla nastavil příslušný prvek v messagesStates na true, což má za následek vykreslení zprávy. Pro udržování aktuálního počtu konkrétních přestupků bylo vytvořeno ve třídě GameData asociativní pole driveResults, kde klíč je typ String (jednoduchý název přestupku, které jsou shodné s klíči v messagesStates) a hodnota je typ int (počet přestupků). Při inicializaci jsou všechny prvky nastaveny na nulu.

Inkrementace prvků pole je prováděna vždy ve stejných místech ve zdrojovém kódu, jako je nastavování výpisů zpráv do pole messagesStates. Po ukončení jízdy je k dispozici kompletní seznam počtu porušených pravidel silničního provozu.



Obrázek 6.19: Obrazovka s výsledky jízdy

Hodnoty výše pokut pro přestupky, jsou uloženy spolu s textovými řetězci podporovaných jazyků aplikace ve třídě TextManager. K datům se přistupuje pomocí metody GetValueByKey. Ve třídě LostState, jenž vypisuje výslednou tabulku, se při získání celkové pokuty za jeden přestupek, získá hodnota pokuty a vynásobí se počtem porušení silničního pravidla. Na obrázku 6.19 je ukázkový snímek obrazovky s výsledky jízdy.

7 Testování

Díky možnosti publikování finální aplikace na web, kde může simulátor 3D Autoškoly vyzkoušet každý s nainstalovaným pluginem Unity 3D, bylo testování značně zjednodušeno, především kvůli snadné přístupnosti pro testovací uživatele. Nicméně verze pro web postrádá možnost nastavení kvality a rozlišení aplikace, proto není vizuální vjem na tak dobré úrovni. Také v této verzi nejsou podporovány stíny, jenž přidávají mnohem více na realističnosti a celkovému dojmu z pohybu v trojrozměrné scéně. Větší přístupnost webové aplikace tyto nedostatky předčila, a proto byla použita tato varianta sestavení aplikace.

Zajištění testovací skupiny uživatelů bylo provedeno především pomocí sociálních sítí Facebook a Twitter. Na těchto sítích byl sdílen odkaz na web s aplikací s prosbou o otestování a následné vyplnění hodnotícího dotazníku, jež se nachází v příloze 1. Mezi pokyny pro otestování byl pouze stručný návod, kde bylo uvedeno, jak mohou uživatelé přepnout češtinu a jak zjistit, kterými klávesami se ovládá automobil. Do testování se nakonec zapojilo 27 testovacích uživatelů, kteří byli ochotni testovat a následně vyplnit dotazník.

Pro testování byla zvolena metoda zvaná kazuistika neboli případová studie. Jedná se o jednu z metod kvalitativního výzkumu. Bývá charakterizována jako detailní studium jednoho, či malého počtu případů za účelem aplikace získaných poznatků při porozumění případů obdobných [37]. Tato metoda se používá v psychologii, sociologii, politologii a jiných vědách. Jelikož záměrem testování je zjistit především uživatelskou přívětivost a použitelnost aplikace, lze kazuistiku použít i v tomto případě.

7.1 Kazuistika – případová studie

7.1.1 Stanovení hypotéz

Před zahájením procesu testování bylo stanoveno několik hypotéz, jejichž potvrzením či vyvrácením se testování zabývalo. Zde je uveden seznam těchto hypotéz:

- Jedná se o intuitivní a snadno použitelnou aplikaci.
- Uživatelé si dokáží představit, že by aplikaci využívali jako studijní pomůcku k získání řidičského oprávnění třídy B.
- Model města má propracovaný systém dopravního značení.
- Úroveň realistického chování automobilu je na dostatečné úrovni pro studijní účely.

7.1.2 Sestavení dotazníku

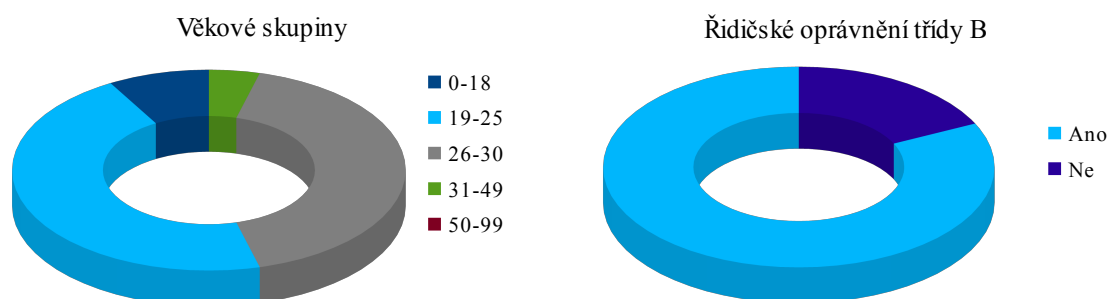
Dotazník byl sestaven způsobem, který umožnil úvodní zjištění informací o uživateli následované možností vyhodnocení jednotlivých aspektů aplikace. V poslední části byl vyhrazen volnější prostor pro vyjádření názorů jednotlivých respondentů.

Účelem prvních dvou otázek dotazníku bylo zjištění, zda má testovací uživatel řidičské oprávnění třídy B a do jaké věkové skupiny patří. Další otázky mají již hodnotící charakter a

využívají stupnice 1-10, kdy hodnocení 1 je nejhorší a hodnocení stupněm 10 nejlepší. Jedná se o otázky vztažené k uživatelskému rozhraní, ovládání automobilu, modelu města, informační zprávy, celkového hodnocení a potenciálu aplikace. Otázky jsou také kladeny tak, aby se ověřily stanovené hypotézy. Dotazník byl vytvořen pomocí nástroje Google Docs pro tvorbu formulářů, který umožňuje i následné publikování a průběžné ukládání výsledků do tabulkového formátu.

7.1.3 Výsledky testování

Testování se zúčastnili uživatelé ze 4 z navržených 5 definovaných věkových skupin, z toho 5 nevlastnilo řidičské oprávnění třídy B. Vzorek testovacích uživatelů byl tedy velmi pestrý, a proto nebyly výsledky ovlivněny jako v případě vybrání pouze studentů informatiky, kteří mají odbornou znalost o fungování aplikací. Na obrázku 7.1 jsou zobrazeny grafy, jež zobrazují složení uživatelů dle věku a vlastnictví řidičského oprávnění třídy B.



Obrázek 7.1: Grafy zobrazující věkové skupiny a vlastnictví řidičského oprávnění mezi uživateli

Z výsledků hodnotících otázek, bylo vypočítáno průměrné hodnocení a tyto hodnoty byly zaneseny do tabulky 7.1.

Tabulka 7.1: Průměrné hodnocení jednotlivých otázek testovacího formuláře

Otázka	Průměrné hodnocení
Spuštění aplikace	9,8
Uživatelské rozhraní	7,9
Ovládání automobilu	7,0
Model města	6,9
Hlášení o porušení silničních pravidel	8,8
Reálnost chování automobilu	5,1
Vyhodnocení jízdy	8,1
Celkové hodnocení	7,3
Potenciál aplikace	7,8

7.1.4 Výsledky textových otázek

Zajímavé informace v dotazníku byly uvedeny v posledních textových otázkách, kde byli uživatelé tázáni na náměty na zlepšení, defekty a další připomínky.

Mezi návrhy na vylepšení bylo například návrh na změnu kláves pro signalizaci změny směru jízdy, protože se to uživatelům nezdálo příliš intuitivní. Jeden z uživatelů také zmínil, že při ovládání automobilu pomocí šipek má s tímto rozložením větší problémy. V dalších verzích aplikace by bylo určitě vhodné vytvořit nastavitelné rozložení kláves pro ovládání 3D Autoškoly. Několikrát byla zmíněna lepší fyzikální chování automobilu. Téma diplomové práce bylo velice rozsáhlé, proto nebyl prostor vyvinout více realistický fyzikální model, nicméně je to validní poznámka a v případných dalších verzích programu by bylo přínosné více propracovat tuto stránku aplikace. Další připomínkou bylo obtížnější udržení konstantní rychlosti a neporušení maximální povolené rychlosti, jelikož má auto relativně velké zrychlení. Uživatel musel neustále stlačovat a uvolňovat tlačítko pro akceleraci, a proto bylo obtížnější dodržovat ostatní předpisy. Tento nedostatek byl ihned opraven zmírněním zrychlení automobilu. Mezi poslední zásadnější připomínky patřilo přidání ostatních automobilů. Pokud by se jednalo o komplexní simulaci dopravního provozu, bylo by nutné se tímto zabývat v další samostatné práci. Dále byly zmíněny pouze drobnosti jako jsou chybějících vysvětlivek, denní a noční jízda, přidání bočních zrcátek a rádio. Tento výčet je užitečný pro další pokračování projektu, ačkoli například rádio určitě není vhodný prvek pro výuku autoškoly.

Další otázka se týkala chyb či defektů, na které uživatelé během testování narazili. Někteří je již zmínili v otázce o vylepšení aplikace. Jeden uživatel objevil chybu v signalizačních světlech semaforů, kde se místo žlutého světla „Pozor“, nacházelo světlo červené. Tato byla odstraněna a ve finální verzi se již nenachází. Také bylo nahlášeno, že na kruhovém objezdu se nehlasí jízda v protisměru, ale tato kontrola není v současné verzi implementována, proto nemohlo dojít k této detekci. Jeden z uživatelů si všiml možnosti projíždět skrz dopravní značky. Jelikož byl tento jev zamýšlený v návrhu, nejedná se o chybu. Nicméně při dalším vývoji aplikace by tato funkcionality mohla být znovu přehodnocena.

V poslední otázce s dalšími komentáři se objevily spíše pochvalné reakce. V jedné z nich hodnotitel ocenil realizaci chodců a jejich interakci s hráčem. Také byla pozitivně ohodnocena propracovanost města a provedení kruhových objezdů.

7.1.5 Testování hypotéz

Pro testování hypotéz byla vybrána metoda *Testování hypotéz o průměru pro jeden výběr*, jenž se používá v případech, kdy se u náhodného výběru n lidí měří nějaká spojitá veličina a chce se určit, že průměr populace, z níž výběr pochází, je roven určitému číslu (takzvané očekávané hodnotě).

Většina statických metod obsahuje porovnání různých metod, či skupin případů. Číselná hodnota tohoto porovnání se nazývá efekt. Tvzení, že efekt je nulový, se nazývá nulová hypotéza. Nulová hypotéza je obvykle opak toho, co má být dokázáno. Další hypotézou je takzvaná alternativní hypotéza, která obecně tvrdí, že efekt není nulový. Na začátku práce se tedy formulují dvě hypotézy nulová a alternativní, kde alternativní přesně vymezuje, jaká situace nastává, pokud nulová hypotéza neplatí.

Pro ověření hypotéz se zavádí testová statistika (testové kritérium), jenž má obecný tvar, který je uveden v rovnici 7.1.

$$Z = \frac{\text{pozorovaná hodnota} - \text{očekávaná hodnota}}{\text{směrodatná chyba pozorované hodnoty}} \quad (7.1)$$

Pozorovaná hodnota je průměr naměřených hodnot (\bar{x}), směrodatná chyba je neznámá, a proto se místo ní používá pouze její odhad s_z vypočítaný z výběru. Pak se již nejedná o testovou statistiku Z, ale o testovou statistiku T, jejíž vzorec je uveden v rovnici 7.2

$$T = \frac{\bar{x} - \mu}{s_z} \quad (7.2)$$

Tím, že se směrodatná chyba nahradila jejím odhadem, se vnesla do výpočtu další nejistota, kterou je třeba brát v úvahu. Z těchto důvodů nemá statistika T již normální rozdělení, ale takzvané *Studentovo rozdělení* o $n-1$ stupních volnosti. Stupňů volnosti je několik druhů, ale tento se používá pro vybranou metodu *Testování hypotéz o průměru pro jeden výběr*. Směrodatná chyba se vypočítá pomocí rovnice 7.3, kde s je směrodatná odchylka a n je počet respondentů.

$$s_z = \frac{s}{\sqrt{n}} \quad (7.3)$$

V případě testování hypotéz stanovených pro 3D Autoškolu, bylo využito pouze jednostranné alternativy, která je v rovnici 7.4. Pro tuto alternativu se zamítá nulová hypotéza v případě, že platí nerovnice 7.5, tedy pokud je hodnota testové statistiky větší nebo rovna kritické hodnotě

$$H_1: \mu > \mu_0 \quad (7.4)$$

$$T \geq t_{1-\alpha}(df) \quad (7.5)$$

Informace k této kapitole byly čerpány z [38].

Testování hypotézy o intuitivnosti aplikace

První stanovenou hypotézou byla intuitivnost a snadné používání aplikace. Této hypotéze byla věnována otázka „Jak byste zhodnotil/a uživatelské rozhraní aplikace (menu, nastavení...)?“ z testovacího dotazníku. Průměrné hodnocení bylo 7,9 se směrodatnou odchylkou 2,18. Jako úspěšné hodnocení byla stanovena hodnota větší než 3/5 z maximálního hodnocení 10, což je rovno 6. Tedy nulová hypotéza:

$$H_0: \mu \leq 6 \quad (7.6)$$

$$H_1: \mu > 6 \quad (7.7)$$

Nyní se vypočítá statistika T:

$$T = \frac{\bar{x} - 6}{\frac{s}{\sqrt{n}}} = \frac{7,9 - 6}{\frac{2,18}{\sqrt{27}}} = 4,53 \quad (7.8)$$

Z tabulky[39] se určí hodnota kritické hodnoty, která je rovna kvantilu $t_{1 - \alpha}(df)$ Studentova t rozdělení. Pro $\alpha = 0,05$ a $df = 27 - 1$ je hodnota rovna 1,706.

Jelikož hodnota statistiky $T = 4,53$ překračuje kritickou hodnotu 1,706, řekneme, že můžeme zamítnout nulovou hypotézu. Kladné hodnocení intuitivnosti aplikace je významné na hladině $p < 0,05$.

Testování hypotézy o potenciálu aplikace

Druhou stanovenou hypotézou bylo tvrzení, že si uživatelé v budoucnu dokáží představit aplikaci 3D Autoškoly jako učební pomůcku pro studenty autoškol. Této hypotéze se věnovala otázka „Myslíte si, že má aplikace potenciál stát se učební pomůckou pro žáky autoškoly?“, jež se nacházela v testovacím dotazníku. Průměrné hodnocení odpovědi na tuto otázku bylo 7,8 se směrodatnou odchylkou 2,04. Hodnota úspěšného hodnocení byla stanovena na 6. Nulová a alternativní hypotéza je pak následující

$$H_0: \mu \leq 6 \quad (7.9)$$

$$H_1: \mu > 6 \quad (7.10)$$

Nyní se vypočítá statistika T:

$$T = \frac{\bar{x} - 6}{\frac{s}{\sqrt{n}}} = \frac{7,8 - 6}{\frac{2,04}{\sqrt{27}}} = 4,58 \quad (7.11)$$

Tabulková hodnota kvantilu $t_{1 - \alpha}(df)$ Studentova t rozdělení je pro $\alpha = 0,05$ a $df = 26$ rovna 1,706. Jelikož hodnota statistiky $T = 4,58$ překračuje kritickou hodnotu 1,706, řekneme, že můžeme zamítnout nulovou hypotézu. Kladné hodnocení potenciálu aplikace je významné na hladině $p < 0,05$.

Testování hypotézy o propracovanosti modelu města

Třetí hypotézou, která byla stanovena, je tvrzení, že model města má propracovaný systém dopravního značení. V testovacím dotazníku byla této hypotéze věnována otázka „Jak byste hodnotil/a model města a jeho dopravní systém?“. Průměrné hodnocení odpovědi na tuto otázku bylo 6,9 se směrodatnou odchylkou 2,12. Hodnota úspěšného hodnocení byla stanovena na 6. Nulová a alternativní hypotéza je pak následující

$$H_0: \mu \leq 6 \quad (7.12)$$

$$H_1: \mu > 6 \quad (7.13)$$

Nyní se vypočítá statistika T:

$$T = \frac{\bar{x} - 6}{\frac{s}{\sqrt{n}}} = \frac{6,9 - 6}{\frac{2,12}{\sqrt{27}}} = 2,21 \quad (7.14)$$

Tabulková hodnota kvantilu $t_{1-\alpha}(df)$ Studentova t rozdělení je pro $\alpha = 0,05$ a $df = 26$ rovna 1,706. Jelikož hodnota statistiky $T = 2,21$ překračuje kritickou hodnotu 1,706, řekneme, že můžeme zamítnout nulovou hypotézu. Kladné hodnocení potenciálu aplikace je významné na hladině $p < 0,05$.

Testování hypotézy o úrovni realističnosti jízdy automobilu

Poslední stanovená hypotéza je, že úroveň realistického chování automobilu je na dostatečné úrovni, pro studijní účely. Na toto téma byla testovacím uživatelům položena otázka „Jak byste zhodnotil/a reálnost chování automobilu?“. Průměrné hodnocení odpovědi na tuto otázku bylo 5,1 se směrodatnou odchylkou 2,16. Hodnota úspěšného hodnocení byla stanovena na 6. Nulová a alternativní hypotéza je pak následující:

$$H_0: \mu \leq 6 \quad (7.15)$$

$$H_1: \mu > 6 \quad (7.16)$$

Nyní se vypočítá statistika T:

$$T = \frac{\bar{x} - 6}{\frac{s}{\sqrt{n}}} = \frac{5,1 - 6}{\frac{2,16}{\sqrt{27}}} = -2,17 \quad (7.17)$$

Tabulková hodnota kvantilu $t_{1-\alpha}(df)$ Studentova t rozdělení je pro $\alpha = 0,05$ a $df = 26$ rovna 1,706. Protože hodnota statistiky $T = -2,17$ nepřekročila kritickou hodnotu 1,706, nemůžeme zamítnout nulovou hypotézu. Na základě výběru nelze dokázat, že by realističnost jízdy byla dostačující.

Úspěšnost testování hypotéz

Byly otestovány čtyři hypotézy pomocí metody *Testování hypotéz o průměru pro jeden výběr*. U tří hypotéz byla zamítnuta nulová hypotéza, čímž byla dokázána alternativní hypotéza významná na hladině $p < 0,05$. U jedné hypotézy se nepodařilo dokázat její pravdivost. Výsledky testování jsou zaznamenány v tabulce 7.2

Tabulka 7.2: Výsledky testování hypotéz

Hypotéza	Potvrzení
Jedná se o intuitivní a snadno použitelnou aplikaci	Ano
Uživatelé si dokáží představit, že by aplikaci využívali jako studijní pomůcku k získání řidičského oprávnění třídy B.	Ano
Model města má propracovaný systém dopravního značení.	Ano
Úroveň realistického chování automobilu je na dostatečné úrovni pro studijní účely.	Ne

8 Vyhodnocení a závěr

Cílem práce bylo vytvořit funkční aplikaci 3D Autoškoly, která má za cíl napomáhat při studiu dopravních předpisů, pomáhat prevenci před dopravními nehodami, ale i zopakování předpisů zábavnou formou. Jedním z cílů byl také dostat aplikaci k uživatelům v co nejpřístupnější formě.

Tyto cíle se splnit podařilo. Byla vytvořena aplikace, která je snadno přístupná z webového prohlížeče a to pro operační systémy Microsoft Windows a OS X. Aplikaci je také možno přeložit pro další operační systémy (Android, iOS, GNU/Linux a jiné), ale již se nejedná o program přístupný z webových stránek. 3D Autoškola je velmi dobře přístupná, uživatel potřebuje pouze plugin Unity 3D, které je stále více používáno pro různé 3D interaktivní aplikace a nejedná se pouze o hry.

Nad rámec zadání byly vytvořeny modulární části silnic, ze kterých lze snadno skládat síť dopravních komunikací města, což otevírá další možnosti pro případné rozšiřování aplikace o jiná města. Dále bylo zajištěno velké množství svislých dopravních značek. Z těchto prvků bylo vytvořeno jedno menší město s dopravním systémem, jenž obsahuje jednosměrné silnice, světelnou křižovatku, zákazy, kruhové objezdy a jiné. Ve městě se také nachází další účastníci provozu. Těmi jsou chodci, kteří se pohybují po přechodech pro chodce. K tomuto městu byl poté vytvořen systém, jenž kontroluje několik přestupků, které byly nadefinovány v návrhu aplikace. Jedná se konkrétně o 8 možných porušení předpisů: překročení rychlosti, nezastavení na značce „Stůj, dej přednost v jízdě“, zastavení na zákazu zastavení, jízda v protisměru v jednosměrné komunikaci, jízda mimo vozovku, špatná signalizace změny směru jízdy při průjezdu křižovatkou, jízda na červenou a poslední nejzávažnější přestupek sražení chodce. Aby měla jízda městem i nějakou výstupní hodnotu, bylo implementováno vyhodnocení jízdy, které obsahuje konkrétní přestupky, počet jejich porušení a hypotetickou výši pokuty za tyto přestupky.

Po implementaci byla aplikace publikována na webu, kde byla otestována více než dvěma desítkami testovacích uživatelů. Během testování se objevilo několik menších defektů, z nichž byla většina opravena a ve finální verzi se již nenachází. Pro vyhodnocení dotazníku bylo využito případové studie neboli kazuistiky. Byly stanoveny čtyři hypotézy, které byly následně testovány metodou *Testování hypotéz o průměru pro jeden výběr*. Tři hypotézy byly dokázány jako pravdivé. Aplikace byla hodnocena velmi kladně a majoritní část uživatelů pozitivně hodnotila, který by mohl být naplněn dalším vývojem aplikace. Z testovacího dotazníku vyplynulo také mnoho zajímavých návrhů na další rozšíření 3D Autoškoly.

Aplikace byla navržena a implementována, tak aby ji bylo možné snadno rozšiřovat o další vlastnosti. Pokud by některý student chtěl pokračovat ve vývoji, například v podobě diplomové práce, nabízí se téma simulace silničního provozu, jež svým rozsahem naplňuje požadavky na další diplomovou práci. Dále je možné snadno rozšířit počet pravidel silničního provozu, která lze kontrolovat během jízdy. Pro větší uživatelský požitek z používání aplikace by také bylo vhodné rozšířit Autoškolu o módy, jako například: jízda s instruktorem, trénink parkování, jízda v noci či za špatného počasí a mnoho dalších.

Bylo dosaženo cíle vytvoření komplexní aplikace, která se snaží přispět ke zlepšení nebo připomenutí znalostí dodržování vybrané části pravidel o silničním provozu. 3D Autoškola je snadno přístupná, a ze zdrojového projektu lze vytvořit aplikace pro mnoho platforem. To ještě více umocňuje její potenciál stát se učební pomůckou pro studenty autoškoly.

Literatura

- [1] Počet obyvatel na 1 auto v krajích ČR. *Sdružení automobilového průmyslu* [online]. 2012, 2. srpna 2012 [cit. 2014-01-08]. Dostupné z: <http://www.autosap.cz/zakladni-prehledy-a-udaje/pocet-obyvatel-na-1-auto-v-krajich-cr/>
- [2] SOBOTKA, Petr. Informace o nehodovosti na pozemních komunikacích v České republice za období leden až listopad 2013. In: *Informace o nehodovosti na pozemních komunikacích v České republice za období leden až listopad 2013* [online]. 2013 [cit. 2014-01-08]. Dostupné z: <http://www.policie.cz/soubor/2013-11-informace-pdf.aspx>
- [3] Pravidla silničního provozu. *BESIP* [online]. © 2012 [cit. 2014-01-08]. Dostupné z: <http://www.ibesip.cz/cz/legislativa/pravidla-silnicniho-provozu>
- [4] ZÁKON O SILNIČNÍM PROVOZU. In: *č. 361/2000 Sb.* 2013. Dostupné z: <http://www.ibesip.cz/data/web/soubory/legislativa/zakon-361-zneni-od-01012013.doc>
- [5] SMITH, Matt. *Unity 4.x cookbook: over 100 recipes to spice up your Unity skills*. Birmingham: Packt Publishing, 2013, iv, 369 s. ISBN 978-1-84969-042-3.
- [6] WARD, Jeff. What is a Game Engine?. In: *Game Career Guide* [online]. 2008 [cit. 2014-01-08]. Dostupné z: http://www.gamecareerguide.com/features/529/what_is_a_game_.php
- [7] BLACKMAN, Sue. *Beginning 3d game development with unity: all-in -one, mult-platform game development*. 2nd edition. S.l.: Apress, 2013. ISBN 978-143-0248-996.
- [8] *DX Studio: 3D Engine & Editor for Windows* [online]. © 2003-2010 [cit. 2014-01-08]. Dostupné z: <http://www.dxstudio.com/index.aspx>
- [9] PREISZ, Eric. Torque 3D to be Released on Github under the MIT license!. In: *GarageGames* [online]. 2012 [cit. 2014-01-08]. Dostupné z: <http://garagegames.com/community/blogs/view/21876>
- [10] *Torque 3D* [online]. © 2000-2014 [cit. 2014-01-08]. Dostupné z: <http://www.garagegames.com/products/torque-3d>
- [11] *Unity: Game Engine* [online]. © 2014 [cit. 2014-01-08]. Dostupné z: <http://unity3d.com/>
- [12] Vodorovné dopravní značky. In: *Vodorovné dopravní značky* [online]. © 2013 [cit. 2014-01-08]. Dostupné z: <http://www.ibesip.cz/data/web/soubory/legislativa/vodorovn-dopravn-zna-ky-komplet.pdf>

- [13] VYHLÁŠKA č. 30/2001 Sb. In: *č. 30/2001 Sb.* 2001. Dostupné z: http://www.ibesip.cz/data/web/soubory/legislativa/besip-20120808_vyhlaska_30_2001-FINAL.pdf
- [14] Car 3D Model. *Archive 3D* [online]. © 2007-2013 [cit. 2014-01-09]. Dostupné z: <http://archive3d.net/?a=download&id=d7a6baf6>
- [15] Verkeersborden Koninkrijk Benjiland (rond). In: *SketchUp* [online]. 1/15/11 [cit. 2014-05-27]. Dostupné z: <https://3dwarehouse.sketchup.com/model.html?id=7b6ceca04f9e7627ea33d1e82afdde21>
- [16] EZsee USB large print keyboard - Black and White. *Www.foreignlanguagekeyboard.com* [online]. www.foreignlanguagekeyboard.com [cit. 2014-01-09]. Dostupné z: http://www.foreignlanguagekeyboard.com/index.php?main_page=product_info&cPath=5&products_id=5&zenid=0e2d06a243543b7be51ad54656497716
- [17] Složení vozového parku v ČR: - souhrnné registrace k 30.6.2013. *Sdružení automobilového průmyslu* [online]. 2013 [cit. 2014-01-09]. Dostupné z: <http://www.autosap.cz/sfiles/a1-9.htm>
- [18] How to Read the: Dashboard Warning Lights - Instrument Panel. *2pass.co.uk: Everything for the learner driver* [online]. © 1996-2014 [cit. 2014-01-09]. Dostupné z: <http://www.2pass.co.uk/dashboard-lights.htm>
- [19] Zákon: o získávání a zdokonalování odborné způsobilosti k řízení motorových vozidel a o změnách některých zákonů. In: *247/2000 Sb.* 2000. Dostupné z: <http://www.mdcz.cz/NR/rdonlyres/2E42747A-5504-430E-A34B-5E5C88C41E29/0/2472000.doc>
- [20] Řidičský trenážér - osobní automobil. *Horázný Autoškola* [online]. 2013 [cit. 2014-01-13]. Dostupné z: <http://www.ridicak.cz/trenazer.php>
- [21] Driving Simulator 2012. [online]. © 2008 - 2012 [cit. 2014-01-14]. Dostupné z: <http://www.jakouhru.cz/Detail/351/driving-simulator-2012>
- [22] Staňte se řidičem v 3D Driving School. In: *AutoškolaOnline.eu* [online]. [cit. 2014-01-14]. Dostupné z: <http://www.autoskolaonline.eu/aktuality/clanek/zabava/stante-se-ridicem-v-3d-driving-school/>
- [23] PhysX SDK. *NVIDIA Developer Zone* [online]. © 2014 [cit. 2014-05-26]. Dostupné z: <https://developer.nvidia.com/physx-sdk>
- [24] *BULLET Physics Library: Real-Time Physics Simulation* [online]. [cit. 2014-05-26]. Dostupné z: <http://bulletphysics.org/wordpress/>
- [25] 3D Warehouse. *SketchUp* [online]. © 2014 [cit. 2014-05-26]. Dostupné z: <https://3dwarehouse.sketchup.com/>
- [26] *Learning C# by Developing Games with Unity 3D*. Birmingham: Packt Publishing, 2013. ISBN 978-184-9696-586.

- [27] VRZAL, Vojtěch. Sazebník pokut: Kolik za co zaplatíme?. *TipCars* [online]. 6.12.2005 [cit. 2014-05-26]. Dostupné z: <http://www.tipcars.com/magazin-sazebnik-pokut-kolik-za-co-zaplatime-1218.html>
- [28] Time.timeScale. In: *Unity Script Reference* [online]. © 2014 [cit. 2014-05-27]. Dostupné z: <http://docs.unity3d.com/Documentation/ScriptReference/Time-timeScale.html>
- [29] Prefabs. In: *Unity - Unity Manual* [online]. © 2014 [cit. 2014-05-27]. Dostupné z: <https://docs.unity3d.com/Documentation/Manual/Prefabs.html>
- [30] *Street View: Google Maps* [online]. © 2014 [cit. 2014-05-27]. Dostupné z: <https://www.google.com/maps/views/streetview>
- [31] *MINI.cz* [online]. [2014] [cit. 2014-05-27]. Dostupné z: <http://www.mini.cz/>
- [32] Colliders. In: *Unity - Tutorials* [online]. 13 Mar 2014 [cit. 2014-05-27]. Dostupné z: <http://unity3d.com/learn/tutorials/modules/beginner/physics/colliders>
- [33] Tutorial Series : Car Controller. In: *FlatTutorials* [online]. 8 August 2013 [cit. 2014-05-27]. Dostupné z: <http://www.flattutorials.com/2013/08/tutorial-series-car-controller-script.html>
- [34] Satisfaction Meter clip art. In: *Clker.com: Free Clipart* [online]. 11-14-2011 [cit. 2014-05-27]. Dostupné z: <http://www.clker.com/clipart-satisfaction-meter.html>
- [35] *Freesound.org* [online]. [2014] [cit. 2014-05-27]. Dostupné z: <http://freesound.org/>
- [36] MonoBehaviour.OnGUI(). In: *Unity Script Reference* [online]. © 2014 [cit. 2014-05-27]. Dostupné z: <https://docs.unity3d.com/Documentation/ScriptReference/MonoBehaviour.OnGUI.html>
- [37] HENDL, Jan. *Úvod do kvalitativního výzkumu*. 1. vyd. Praha: Karolinum, 1997, 243 s. ISBN 80-718-4549-3.
- [38] ZVÁROVÁ, J. *Základy statistiky pro biomedicínské obory*. Vyd. 1. Praha: Karolinum, 2002, 218 s. ISBN 80-718-4786-0.
- [39] Tabulka 1: Kvantily Studentova rozdělení. In: *Ústav matematiky* [online]. 19.12.2006 [cit. 2014-05-27]. Dostupné z: http://www.umat.feec.vutbr.cz/~hlavicka/vyuka/kvantily_t.pdf

Seznam příloh

Příloha 1. Testovací dotazník

Příloha 2. DVD se zdrojovými soubory

Příloha 1

Testovací dotazník

Před Vámi je dotazník, jenž je součástí diplomové práce a téma 3D Autoškola. Cílem práce bylo navrhnout a poté vytvořit moderní 3D simulátor autoškoly s důrazem na kontrolu dodržování silničních pravidel. Dotazník je jedním z nástrojů určený pro zjištění úspěšnosti výsledku projektu. Děkuji Vám za jeho vyplnění. Prohlašuji, že tento dotazník je anonymní a bude využit pouze ke zpracování výsledku mé diplomové práce

Věková skupina *

Jaká je vaše věková skupina?

- 0 – 18
- 19 – 25
- 26 – 30
- 31 – 49
- 50 – 99

Pohlaví

- Muž
- Žena

Máte řidičské oprávnění třídy B? *

- Ano
- Ne

Spuštění aplikace

Měli jste problémy spustit aplikaci, případně nainstalovat Unity 3D plugin do vašeho prohlížeče?



Problematické spuštění

Bezproblémové spuštění

Uživatelské rozhraní *

Jak byste zhodnotil/a uživatelské rozhraní aplikace (menu, nastavení...)?



Neintuitivní

Intuitivní

Ovládání automobilu *

Jak byste zhodnotil/a ovládání automobilu?



Neovladatelné

Jednoduše ovladatelné

Reálnost chování automobilu *

Jak byste zhodnotil/a reálnost chování automobilu?

Velmi nereálné

Reálné

Model města *

Jak byste hodnotil/a model města a jeho dopravní systém?

Málo dopravního značení

Propracovaný dopravní systém

Hlášení o porušení silničních pravidel *

Jak byste ohodnotil/a hlášení o porušení pravidel během jízdy?

Rušivé

Adekvátní

Vyhodnocení jízdy *

Jak byste zhodnotil/a závěrečné vyhodnocení (počet přestupků a pokuty) po ukončení jízdy?

Malá vypovídající hodnota

Kvalitní hodnocení

Celkové hodnocení *

Jak byste zhodnotil/a aplikaci?

Nezajímavá

Vyjímečná

Potenciál aplikace *

Myslíte si, že má aplikace potenciál stát se učební pomůckou pro žáky autoškoly?

Ne

Ano

Další hodnocení

Pokud vám v předchozích otázkách nějaká chybí, zde můžete uvést další názory.

Nápady na zlepšení aplikace

Co byste uvítal/a v aplikaci 3D autoškola?

Našel/la jste nějakou chybu/defekt během užívání aplikace? Jakou?

Příloha 2

Obsah DVD

Builds – obsahuje sestavené aplikace 3D Autoškoly pro platformy Windows, Linux a webovou aplikaci.

Documentation – obsahuje dokumentaci k aplikaci.

Project – obsahuje celý projekt aplikace 3D Autoškoly včetně modelů a zdrojových kódů.

Thesis – obsahuje text diplomové práce.