



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

3D AUTOŠKOLA

3D DRIVING SCHOOL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR KAPOUN

VEDOUcí PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠÁTEK, Ph.D.

BRNO 2020

Zadání bakalářské práce



Student: **Kapoun Petr**
Program: Informační technologie
Název: **3D Autoškola**
3D Driving School
Kategorie: Modelování a simulace

Zadání:

1. Nastudujte problematiku simulace silničního provozu a metodiku kontroly dodržování pravidel silničního provozu. Analyzujte platná pravidla systému a vytvořte jejich seznam dle priority implementace.
2. Vytvořte architekturu systému simulátoru a uživatelského rozhraní.
3. Vytvořte (případně zajistěte z vhodných zdrojů) základní 3D modely a funkční 3D engine, které budou použity v simulátoru.
4. V simulátoru implementujte základní fyzikální model, sledování dodržování prioritních pravidel a reprezentaci pohybu účastníků provozu a jejich vzájemné interakce.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu.

Literatura:

- Zerbst, S.: 3D Game Engine Programming. Premier Press, 2004, ISBN 1-59200-351-6.
- Černovský, D., Bergmannová, M.: Autoškola. Vogel(Business Media CZ), 2016, ISBN 978-80-87388-31-0.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šátek Václav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 14. května 2020

Datum schválení: 31. října 2019

Abstrakt

Práce se zabývá návrhem a implementací simulátoru 3D autoškoly. Po analýze existujících řešení a návrhu simulátoru se práce zabývá problematikou kontroly pravidel silničního provozu a možností jejich implementace v simulátoru. Pro implementaci byl vybrán herní engine Unity 3D. Modely byly vytvořeny v programu Blender.

Abstract

This thesis deals with design and implementation of a 3D driving school simulator. After the analysis of existing solutions and the design of the simulator, this thesis discusses the issue of checking the rules on the road and the possibility of their implementation in the simulator. The Unity 3D game engine was chosen for this task. The models were created in Blender.

Klíčová slova

Autoškola, Simulace, Pravidla silničního provozu, Fyzikální model automobilu, Uživatelské rozhraní, Unity 3D, Blender

Keywords

Driving school, Simulation, Traffic rules, Physical model of a car, User interface, Unity 3D, Blender

Citace

KAPOUN, Petr. *3D Autoškola*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šátek, Ph.D.

3D Autoškola

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šátka, Ph.D. Další informace mi poskytl doc. Ing. Peter Chudý, Ph.D., MBA. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Petr Kapoun
27. května 2020

Poděkování

Děkuji vedoucímu práce, panu Ing. Václavu Šátkovi, Ph.D., za vřelý přístup a čas věnovaný při vedení práce. Dále bych rád poděkoval doc. Ing. Petru Chudému, Ph.D., MBA, který práci vedl předcházející akademický rok.

Obsah

1	Úvod	4
2	Simulace silničního provozu	6
2.1	Simulátor pro osobní počítač	6
2.2	Simulátor pro mobilní telefon	8
2.3	Platná pravidla silničního provozu	10
2.4	Kontrola dodržování pravidel silničního provozu	12
2.5	Seznam pravidel silničního provozu dle priority implementace	12
3	Návrh simulátoru	13
3.1	Herní engine	13
3.2	Architektura systému simulátoru a uživatelského rozhraní	15
3.2.1	Uživatelské rozhraní	15
3.2.2	Architektura simulátoru	16
3.3	3D modely	18
4	Fyzikální model	20
4.1	Přehled působících sil na vozidlo ve směru dopředu	20
4.2	Technická specifikace vozidla	21
4.3	Valivý odpor	21
4.4	Vzdušný odpor	22
4.5	Odpor stoupání	23
4.6	Moment a efektivní výkon motoru	23
4.7	Pilový diagram	23
4.8	Hnací síla motoru	25
4.9	Brždění	25
5	Implementace	27
5.1	Hlavní menu	27
5.2	Správa vstupů	29
5.3	Herní svět	29
5.4	Herní smyčka	29
5.5	Nekonečný svět	30
5.6	Cyklus střídání dne a noci	32
5.7	Vozovka	33
5.8	Objekt vozidla	33
5.8.1	Fyzikální model	35
5.9	Objekt hráče	35

5.10	Herní uživatelské rozhraní	37
5.10.1	Nápověda k ovládání	37
5.10.2	Hlášení jízdních událostí	37
5.10.3	Zobrazení stavu vozidla	38
5.10.4	Konec simulace	39
5.11	Ostatní objekty herního světa	40
5.12	Účastníci provozu	40
5.12.1	Začátek a konec života objektu	41
5.12.2	Směr jízdy vozidla a zapnutí směrových ukazatelů	42
5.12.3	Zastavení před překážkou	43
5.12.4	Přechod mezi dlaždicemi herního světa	44
5.13	Reprezentace křižovatek a jiných dopravních úseků s upravenou předností vozidel pro řízení provozu	44
5.14	Dodržení přednosti jízdy ostatních vozidel	49
5.15	Sledování dodržování pravidel silničního provozu	51
6	Testování	56
6.1	Stavba formuláře	56
6.2	Výsledky testování	57
7	Závěr	60
	Literatura	61
A	Obsah paměťového média	64
B	Návod na ovládání	65
C	Snímky výsledné podoby práce	66
D	Formulář pro uživatelské testování	69

Seznam zkratek

- 3D** Trojrozměrný prostor (Tri-dimensional space)
- AI** Umělá inteligence (Artificial Intelligence)
- CPU** Centrální procesorová jednotka (Central processing unit)
- GC** Způsob automatické správy paměti (Garbage collector)
- GPU** Grafický procesor (Graphic processing unit)
- GUI** Grafické uživatelské rozhraní (Graphic User Interface)
- HW** Hardware
- IT** Informační technologie (Information Technology)
- PC** Osobní počítač (Personal computer)
- UI** Uživatelské rozhraní (User Interface)
- SW** Software

Kapitola 1

Úvod

Vlastnit řidičský průkaz a umět řídit automobil se s postupem času stalo nutností, i proto automobilů na silnicích stále přibývá. Podle souhrnných statistik Ministerstva dopravy ČR [15] bylo v roce 2019 bylo registrováno 348 687 nových vozidel a 219 217 ojetých vozidel. Naopak vyřazeno bylo 245 287 vozidel. Celkem bylo k 1.1.2020 evidováno 9 342 673 vozidel [16]. Při pohybu na pozemních komunikacích je nutné myslet především na bezpečnost. I kvůli bezpečnosti existují pravidla silničního provozu, která musí každý jeho účastník dodržovat. V roce 2019 bylo evidováno 375 012 přestupků zakotvených v zákoně č. 361/2000 Sb. o provozu na pozemních komunikacích [17]. Neznalost pravidel silničního provozu může končit nebezpečnými situacemi, zraněními, doživotními následky nebo smrtí.

Žáci základních škol jezdí na dopravní hřiště, aby se seznámili se základními pravidly silničního provozu. Později většina lidí chodí do autoškoly, aby získala řidičský průkaz. Výuka v autoškolách probíhá mnoha způsoby. Krom jízdy s učitelem se musí žák naučit teorii, která může být vyučována čtením textu, řešením modelových obrázkových situací nebo jinými způsoby. Některé autoškoly používají také simulátory autoškol. Po složení závěrečných zkoušek se lidé jízdou na silnici stále učí a získávají zkušenosti, protože se na silnici dostávají do různých dalších neznámých situací. Velké množství lidí má se složením zkoušky problémy. Mají omezený počet jízd a poté nemají jak své dovednosti trénovat. Zde jim může pomoci simulátor autoškoly.

Před tvorbou libovolného projektu je vhodné prozkoumat danou oblast a existující řešení vytvořená jinými autory. Tvůrce se tímto průzkumem dostane na počáteční bod své práce. Existující řešení je nutno důkladně rozebrat a zaměřit se na různá řešení konkrétních problémů. I v této práci jsem začal hledáním a zkoušením jiných řešení ještě před návrhem vlastního simulátoru. Vyjasnil jsem si, co chci vytvořit i v čem se můj projekt má lišit od těch existujících. Důležitým zjištěním u simulátorů autoškol bylo, že jich moc nevzniká. Hledal jsem simulátory pro různé platformy a nejvíce jich je pro mobilní zařízení. Na desktopových zařízeních je stále hojně používán simulátor 3D *Fahrschule* [4] popsany v 2.1, který byl vydán v roce 2001. Z toho je patrné, že vydání nového produktu kýmkoliv je vzácné. Existují i jiná novější řešení, která sice jsou graficky propracovanější a přívětivější, ale kontrola pravidel silničního provozu zde působí druhořadě. Nejde o velké projekty, proto je zde malá paleta dopravních situací. Simulátorů autoškol existuje málo, a tak jsem zkoumal i jiné silniční simulátory, které nejsou autoškoly, ale jsou oproti simulátorům autoškol novější a propracovanější. Ze simulátorů autoškol jsem získal poznatky pro kontrolu dopravních předpisů a ze silničních simulátorů zase poznatky pro tvorbu herního světa, obzvláště pro tvorbu modelu vozidla a jeho ovládání.

Obecně se simulátory tvoří z různých důvodů. Existují simulátory výukové, ale také simulátory zábavné. Výukový simulátor má šetřit náklady na získání znalostí, zatím co zábavný má využít volný čas uživatele. Tvorba závodního simulátoru bez přelomového nápadu vzhledem k situaci na trhu by nedávala smysl. Tento simulátor lze označit jako čistě zábavný a takových simulátorů existuje obrovské množství. Na druhou stranu existující simulátory autoškol zdaleka nevyužily všechny možnosti, které tato problematika nabízí. K tvorbě se lze postavit mnoha různými způsoby. V případě simulátoru autoškoly může herní svět množstvím dopravních značení připomínat více dopravní hřiště než reálnou pozemní komunikaci. V simulátoru autoškoly je důležitější dodržovat pravidla silničního provozu než řídit vozidlo. Simulátor vytvoří uživateli modelové situace, kterých za stejnou dobu musí uživatel řešit více než v reálném provozu. Několikakilometrová silnice mezi městy působí neúčelně.

Cílem práce bylo vytvořit simulátor autoškoly, neboli vytvořit herní svět, který bude kontrolovat uživatelské dodržování pravidel silničního provozu. Jako zpětnou vazbu program sdělí detekované dopravní přestupky a případně může pochválit. Nejdříve jsem definoval, jaký simulátor chci vytvořit. Následoval návrh architektury simulátoru a implementace. Implementace začíná vytvořením a rozpohybováním hráčova vozidla, bez kterého by nebylo co detekovat. Potom lze tvořit herní svět od různých silnic a dopravních značek až po objekty mimo silnice, které jsou pro funkcionality nepotřebné. V každé dopravní situaci hraje roli několik aspektů, které je potřeba pro ekvivalentní modelovou situaci vytvořit pomocí modelů a skriptů. V návrhu herního světa hrají velkou roli ostatní účastníci provozu, kteří se v tomto světě musí zvládnout pohybovat, ale i dodržovat pravidla silničního provozu.

Text této práce popisuje postup splnění cíle práce v několika navazujících částech. Nejdříve jsou analyzována existující řešení a nastudována pravidla silničního provozu a problematika jejich kontroly. Další kapitola se zabývá návrhem architektury simulátoru a uživatelského rozhraní. Následují kapitoly popisující tvorbu 3D modelů a volbu herního enginu. V další kapitole je zjednodušeně fyzikálně popsán model auta. Vypočtena je hnací síla motoru a jízdní odpory. Následuje nejrozsáhlejší kapitola této práce, kterou je implementace. Na závěr je popsáno uživatelské testování vytvořené práce a jeho výsledky, které napomohly k jejímu vylepšení.

Kapitola 2

Simulace silničního provozu

Tato kapitola se zabývá rozbořem existujících simulátorů jízdy po pozemních komunikacích s kontrolou dodržování pravidel silničního provozu a dále samotnými pravidly silničního provozu.

2.1 Simulátor pro osobní počítač

Platforma osobních počítačů nabízí mnoho možností a je možné pro ni vytvořit komplikované simulátory. To je ale zároveň důvodem, proč detailně vypracovaných simulátorů autoškol není mnoho. Herní studia dávají přednost vývoji her pro mobilní platformy, kde jejich produkty nemusí být tak robustní. Vývoj menšího simulátoru je méně nákladný a studio lehčeji dosáhne očekávaného zisku. Z toho důvodu jsem na této platformě pro příklad zvolil pouze jeden simulátor autoškoly. Na druhou stranu existuje mnoho her, ze kterých si lze vzít ponaučení. Druhým příkladem bude světově známý simulátor bez výukového účelu, který demonstruje kvalitně zvládnuté ovládání vozidla.

3D Fahrschule

3D Fahrschule [4] je simulátor autoškoly od vydavatele Besier 3D-Edutainment Wiesbaden vydaný v roce 2001.

Simulátor je vhodný pro osvojení si základních jízdních schopností. Nabízí proměnlivý provoz, kdy je silnice v jednu chvíli prázdná a o chvíli později je na ní mnoho aut. Auta v provozu se chovají podle předpisů. Křižovatky v simulátoru jsou různorodé, ať už jde o různé kombinace odbočovacích pruhů nebo značení. Hráč ve hře postupuje úrovněmi a při chybě v jedné úrovni se nevrátí na začátek dané úrovně, ale jen před část trasy, kde chyboval. Kontroly dodržování většiny předpisů jsou striktní, což podporuje účel výukového programu.

Sporným bodem simulátoru je řidičovo rozhlížení se do stran. Nápad natáčení kamery doleva nebo doprava po stisku kláves je dobrý, ale uživatel nemá jak kontrolovat rychlost otáčení kamery. Tím se otáčení kamery za jízdy stává rizikovým, protože při pohledu doprava, není jisté co se děje nalevo a řidič může nabourat. Při pohledu z kabiny auta řidič nevidí pravý sloupek vedle čelního skla a pokud se nerozhledne doprava, tak může nabourat například do značky, která stojí vedle vozovky.

Nevýhodou simulátoru je jeho citlivost ovládání. Ta vede k nepřesnému projíždění zatáček a případně i ke srážkám. Snížení negativního dopadu na hru napomáhá fakt, že kontroly pozice auta nejsou tak striktní. To znamená, že když hráč přejede chodník, nebo jedním



Obrázek 2.1: **3D Fahrschule**. Hra od vydavatele Besier 3D-Edutainment Wiesbaden [4].

kolem vjede do protisměru, nic se nestane. Simulátor byl oficiálně vydán v roce 2001. Hry se rychle vyvíjí, a proto jeho grafika nepůsobí nejlépe. Na křižovatce simulátor sice kontroluje, zda hráč opustil křižovatku se zapnutým směrovým světlem, ale nekontroluje, že si řidič uprostřed křižovatky několikrát rozmyslel, na kterou stranu pojede.

Euro Truck Simulator 2

Euro Truck Simulator 2 [26] je hra od vydavatele SCS Software, která se úspěšně prosadila na herním trhu. Hra není postavena jako autoškola, ale je zde mnoho věcí, které v existujících simulátorech autoškol chybí. Důkazem pro toto tvrzení jsou recenze uživatelů na službě Steam [33], které jsou uvedené v tabulce 2.1. Mobilní verze hry Driving School Simulator z tabulky 2.1 je popsána v sekci 2.2.

Tabulka 2.1: Srovnání simulátorů autoškol a hry Euro Truck Simulator 2 na službě Steam ze dne 11.12.2018.

Titul	Vydavatel	% kladných recenzí
Driving School Simulator	Ovilex Software	20
City Car Driving	Forward Global Group	68
Euro Truck Simulator 2	SCS Software	96

Kontrola dodržování pravidel se projevuje udělováním pokut, protože hráč má účet s penězi i na vylepšování svého kamionu nebo kupování nových. Přestupky jsou podobné jako

v simulátorech autoškol, tedy například průjezd křižovatky na červenou nebo překročení maximální rychlosti.

V existujících simulátorech autoškol je problémem, že uživatel nemá informace o okolí. Tato hra nabízí uživateli pro přepínání 8 různých kamer, mezi které patří například: pohled z kabiny, pohled shora, pohled dozadu z levé nebo pravé strany vozidla. Uživatel se při zvolení vybraných kamer může navíc pohybovat v prostoru pohybem myši a podívat se přesně na místo, které potřebuje k úspěšnému provedení zamýšlené operace, například při parkování, při průjezdu křižovatkou, při přejíždění mezi pruhy nebo při zastavování na určeném místě. Dále si uživatel může, kromě klasických zpětných zrcátek na bocích kabiny, zapnout i zrcátka jako prvek uživatelského rozhraní, jak je zobrazeno na obrázku 2.2. Takto uživatel vždy vidí situaci na vozovce. Přestože u kamionu je těžší mít přehled o vozidle, tak v této hře má uživatel o vozidle lepší přehled než o autě v existujících simulátorech autoškol.



Obrázek 2.2: Euro Truck Simulator 2. Hra od vydavatele SCS Software [26].

2.2 Simulátor pro mobilní telefon

Mobilní zařízení přináší pro vývojáře simulátoru autoškol mnohá úskalí v oblasti vstupů. K ovládní simulátoru je i na desktopu potřeba velké množství kláves. Příkladem může být ovládní směrových světel nebo natočení kamer do stran před křižovatkami, aby se uživatel rozhlédl. I přes tuto obtíž spojenou s ovládním simulátoru vytvořila některá studia dobré simulátory. Nejdříve bude představen povedený simulátor autoškol Driving School 3D [20] a poté druhý simulátor autoškol Driving School 2016 [22].

Driving School 3D

Driving School 3D [20] je simulátor autoškol od vydavatele nullapp dostupný od roku 2018. Od roku 2020 je na platformě Google Play [12] uváděn pod názvem Driving School



Obrázek 2.3: **Driving School 3D**. Hra od vydavatele nullapp [20].

Simulator 2020, i když na stránkách vydavatele i v promočních materiálech i jako nainstalovaná aplikace je stále pod původním názvem.

V této hře najde uživatel různé herní světy. Mezi ty patří město, vesnice, hory a dálnice. Právě horské silnice, které jsou mnohdy i šterkové bych tomuto simulátoru vytkl, protože na nich se hráč nedostane do žádných dopravních situací. Po každém přechodu chodí dost chodců, ale na silnicích uživatel bohužel nepotká velké množství aut.

Unikátním řešením jsou průhledné obdélníky kolem křižovatek. Uživateli je jasné, že přesně na této hranici bude probíhat kontrola dodržení pravidel silničního provozu. Další předností tohoto produktu je zpětná vazba, která je poskytována uživateli. Je to hlavně kvůli časté pozitivní zpětné vazbě. Informace, že proběhla kontrola a jaká kontrola, jsou velmi cenné. Hráč nemusí díky spojení viditelných průhledných obdélníků a těchto pozitivních zpráv tápat, zda jede správně. Z těchto pozitivních zpráv je i patrné, že systém kontrol je v této hře velmi propracovaný.

Uživatelské rozhraní je přívětivé a snadno použitelné až na kamery, které nelze ovládat vůbec. Jsou zde pouze dvě k přepínání. Na křižovatkách je nelze natočit a hráč se tedy musí rozjet, i když může jet auto, které má přednost. Když hráč zastaví u semaforu, tak na světla na semaforu nevidí.

Driving School 2016

Driving School 2016 [22] je simulátor autoškoly od vydavatele *Ovilex Software* dostupný od roku 2018. Tento vydavatel má větší množství podobných produktů, jako je simulátor řízení taxi, autobusu, tahače, vlaku či lodě. Druhým podobným titulem je autoškola, ve které jezdí pouze starší vozidla, zatím co v této jezdí hráč v nových modelech. O rok později po vydání titulu *Driving School 2016* byla vydána novější a poslední verze této hry. Hru bylo bohužel možné hrát pouze s použitím gamepadu.



Obrázek 2.4: **Driving School 2016**. Hra od vydavatele Ovilex Software [22].

Hra je zaměřena spíše na požitek z řízení automobilu, než na dodržování pravidel. Je zde velká nabídka různých aut a map reprezentujících různé části světa. Modely silnic jsou jednoduché a některé křižovatky působí dost poslepaně. Křižovatek projede hráč za stejnou dobu v této hře výrazně méně než ve hře **Driving School 3D** [20]. I množství dopravních značek je malé. Na obrázku 2.4 je vidět ovládání hry. Tlačítka jsou na mobilní telefon malá. I proto mohl být v následující verzi povolen pouze gamepad jako jediný způsob ovládání simulátoru.

2.3 Platná pravidla silničního provozu

Existuje mnoho druhů pravidel pro silniční provoz. Je například přesně definováno, jakou rychlostí může řidič jet v obci, mimo obec nebo na rychlostí silnici. Velkou část pravidel řidiči stanovují dopravní značky, definované ve vyhlášce č. 294/2015 Sb. Tato pravidla jsou platná na daném úseku silnice. Dále jsou zde pravidla platná na každé veřejné komunikaci, jako jsou například pravidla pro přepravu osob a nákladu. Vytvořený seznam priority implementace je v sekci 2.5.

Svislé dopravní značky

Svislé dopravní značky existují stálé, přenosné a proměnné.

Stálé dopravní značky se upevňují na sloupky, budovy nebo jiné konstrukce pevně spojené se zemí. Červené a bílé pruhy na sloupku, na němž je umístěna přenosná dopravní značka, mají šířku 100 až 200 mm a retroreflexní provedení. Proměnné dopravní značky jsou značky, jejichž provedení se může měnit. Barevné provedení proměnných dopravních značek může být odlišné od stálých dopravních značek, a to tak, že podklad proměnných dopravních značek je tmavý a nápisy, symboly a ohraničení světlé. Pro zdůraznění významu a zlepšení viditelnosti lze svislou dopravní značku umístit na retroreflexním žlutozeleném

fluorescenčním podkladu. Takový podklad nelze použít v kombinaci se světelnými signály [2].

Existuje několik druhů svislých dopravních značek:

- značky upravující přednost,
- výstražné značky,
- zákazové značky,
- příkazové značky,
- informativní značky,
- dodatkové tabulky.

Vodorovné dopravní značky

Vodorovné dopravní značky se užívají buď samostatně nebo ve spojení se značkami svislými nebo s dopravními zařízeními, jejichž význam zdůrazňují nebo zpřesňují. Vodorovné dopravní značky jsou buď stálé nebo přechodné. Přechodné jsou vyznačený žlutou nebo oranžovou barvou. Vodorovné dopravní značky se rozdělují na:

- podélné čáry,
- příčné čáry,
- šipky,
- označení stání a parkovišť,
- označení zastávek a zákazů zastavení a stání,
- ostatní vodorovné dopravní značky [8].

Světelné signály

Existuje více druhů. Světelným signálem jsou například semaforey na křižovatkách, které řídí pořadí průchodu účastníků provozu křižovatkou a to ať už to jsou chodci nebo vozidla. Dále můžeme vidět světelný signál na železničním přejezdu, který udává maximální rychlost do vzdálenosti 50 metrů od přejezdu nebo zcela zastavuje dopravu v případě potřeby. Semaforey jsou také používány při dočasných dopravních omezeních, kdy se doprava obou směrů spojí do jednoho pruhu, ve kterém se směry střídají v průjezdu.

Dopravní zařízení

Dopravní zařízení doplňují dopravní značky, světelné a akustické signály při řízení dopravy. Často bývají použity při dočasných dopravních omezeních. Do dopravních zařízení řadíme například dopravní kužely, zábrany pro označení uzavírky, směrovací a vodící desky, směrové sloupky zastavovací terče pro policisty a další dopravní zařízení.

2.4 Kontrola dodržování pravidel silničního provozu

V reálném světě mají orgány pověřené kontrolou dodržování pravidel silničního provozu několik metod, jak kontroly provádět. Prvním typem kontroly je kamerové zařízení, které vyfotí řidiče při přestupku. Velmi často jsou tyto kamerové systémy používány pro kontroly rychlosti. Druhým typem kontrol je osobní kontrola, kdy pověřená osoba zaregistruje přestupek některého účastníka provozu.

Při simulaci silničního provozu jsou oba tyto způsoby neefektivní, a to jak z hlediska implementace, tak z hlediska počtu odhalených přestupků. V reálném světě nejsou odhaleny všechny spáchané přestupky a navíc je nutno dokázat podezřelé osobě spáchání konkrétního přestupku buď fotkou z kamery nebo chycením při činu. V simulátorech se také jen zřídka objevuje kontrola dodržování pravidel stanovených vodorovnými dopravními značkami, představenými v sekci 2.3, a dopravními zařízeními, představenými v sekci 2.3.

2.5 Seznam pravidel silničního provozu dle priority implementace

Při sestavování seznamu pravidel dle priority implementace je nutné vycházet z předcházejících částí této práce. Existuje mnoho pravidel stručně představených v části 2.3 této práce. Není možné všechna pravidla implementovat v simulátoru, příkladem může být kontrola obsahu alkoholu v krvi. Dalším faktorem je náročnost implementace jednotlivých pravidel i jejich vliv na celkový dojem aplikace. V neposlední řadě je dobré se poučit z existujících simulátorů. Lze zde odhalit problémy implementace některých pravidel, i zde najít možnosti pro jedinečnost vlastního simulátoru.

1. Kontrola dodržení maximální povolené rychlosti.
2. Průjezd křižovatkou bez značení.
3. Kontrola dodržení značky *Stůj, dej přednost v jízdě!*.
4. Průjezd křižovatkou se světelnými signály.
5. Průjezd křižovatkou a dání přednosti se značkami *Stůj* a *Dej přednost v jízdě*.
6. Kontrola dodržení zákazu zastavení a stání.
7. Kontrola správného průjezdu úsekem s dočasným dopravním omezením.
8. Kontrola správného průjezdu železničního přejezdu.

Kapitola 3

Návrh simulátoru

Před tvorbou projektu je nutné celou práci na projektu promyslet. Důležitá je volba nástrojů, které lze použít pro vytvoření práce. Správná volba nástroje může práci ulehčit. V této kapitole bude nejdříve popsána volba herního enginu v 3.1, která ovlivnila návrh architektury popsaný v . Nakonec bude představen zvolený nástroj pro tvorbu 3D modelů, pomocí kterého jsem sám vytvořil veškeré modely, které byly v této práci použity (auto, silnice, značky, budovy, chodníky, stromy, větrná elektrárna atd.).

3.1 Herní engine

Herní engine nejen ulehčuje práci na vývoji hry. Jeho hlavním úkolem je poskytnout úkony nižší rutiny jako je například základní fyzikální model, zpracování vstupů, zvuků, detekce kolizí nebo umělá inteligence. Tyto úkony jsou prováděny pomocí jednotlivých sub-enginů. Herní engine se skládá z mnoha sub-enginů, jako jsou například 3D engine, zvukový engine, engine pro správu vstupů, síťový engine, fyzikální engine, AI engine atd. Například zvukový engine se stará o nahrání, manipulaci a spouštění zvuků [36].

Při výběru enginů jsem se zaměřil na dva nejvíce používané herní enginy.

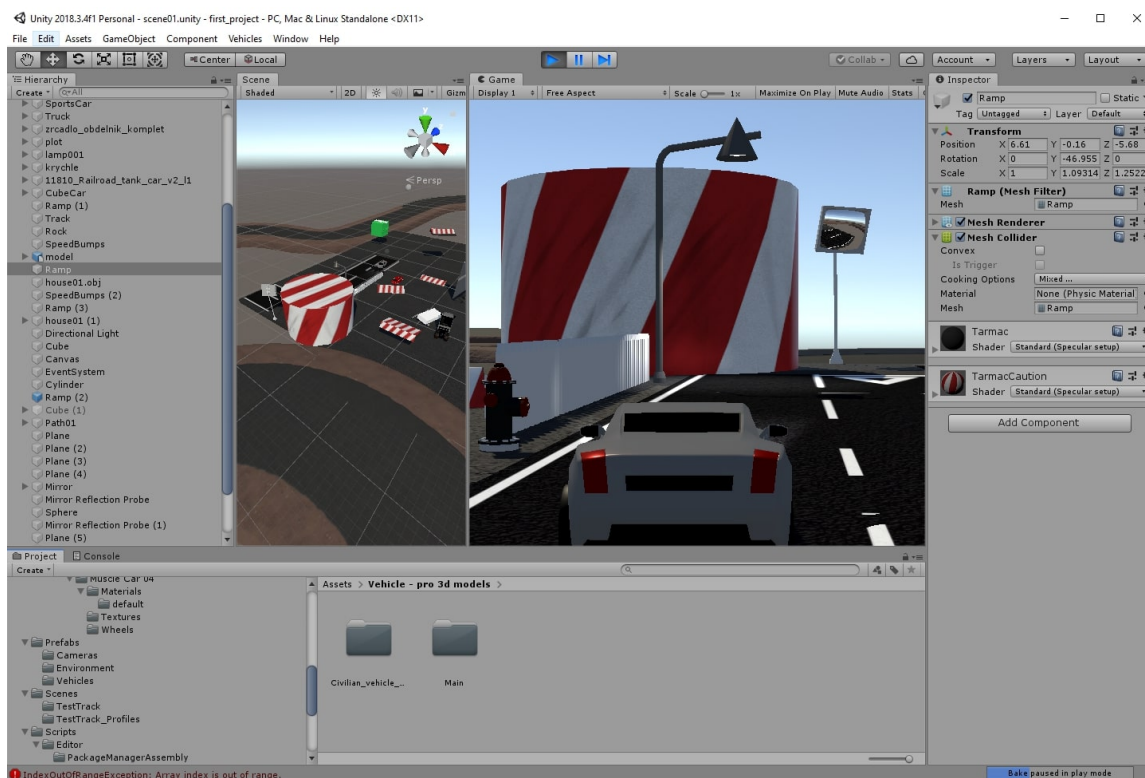
Unity 3D

Unity 3D [32] je herní engine vyvíjen společností Unity Technologies a byl poprvé vydán v roce 2005. Editor enginu je vidět na obrázku 3.1. Hlavním programovacím jazykem je C#, ale je možné psát skripty i v jazyce JavaScript a Boo. Editace kódu probíhá pomocí editoru třetí strany, lze použít například Visual Studio [14].

Editor scén je hlavní předností editoru, který obsahuje nástroje pro tvorbu terénu s aplikováním 2D a 3D textur, vkládání a správu objektů (přidání textur, kolizních bloků, komponentních skriptů s nastavením proměnných atd.). Během testování hry lze za běhu měnit, přidávat a odebírat jednotlivé komponenty a objekty.

Mezi další přenosti patří i početná komunita a velké množství studijních materiálů.

Práce s enginem Unity 3D [32] je velmi intuitivní a proto je doporučován začátečnickům. Je zároveň pomocí placených rozšíření a nástrojů dostatečně komplexní pro vytváření rozsáhlých a kvalitních her. Mezi nejznámější hry vytvořené pomocí tohoto enginu patří například Heartstone [6] společnosti Blizzard Entertainment, Assassin's Creed: Identity [29] společnosti Ubisoft, Pokemon Go [18] společnosti Niantic a Kerbal Space Program [27] společnosti Squad.



Obrázek 3.1: Herní engine Unity 3D [32].

Unreal Engine

Unreal Engine [11] je vyvíjen společností Epic Games a byl poprvé vydán v roce 1998. Editor engine je vidět na obrázku 3.2. Hlavní programovacím jazykem je c++. Pro jednodušší úkony je možné použít Blueprints Visual Scripting systém, což je rozhraní založené na uzlech. Ve skratce bývají tyto skripty nazývány Blueprints.

Unreal Engine [11] je již v základu velmi komplexní a jeho jednotlivé komponenty nabízí velkou variabilitu. To vytváří prostředí pro tvorbu jedinečné a do detailu vyladěné hry.

Engine má velmi kvalitní zpracování světla a stínů, které po vyrendrování vypadají velmi přesně a reálně. Textury nevytváří téměř žádné grafické závady a artefakty. Editor materiálů umožňuje bez velkého ladění kvalitně aplikovat a upravit textury, narozdíl od engine Unity 3D [32], ze kterého je nutné často přepínat do grafického editoru. Mezi hlavní přednosti engine patří i integrované nástroje pro Post processing, které výrazně zlepšují celkový grafický dojem. To jsou například objemové efekty, efekt objektivu nebo přechody.

Mezi nejznámější hry vytvořené pomocí tohoto engine patří například *Might and Magic Heroes VII* [30] společností Ubisoft, *Fortnite* [10] společností Epic Games, *Batman: Arkham City* [24] společností Rocksteady Studios nebo *Rocket League* [23] společností Psyonix.

Zvolený herní engine

Popsané enginey jsou nejvíce používanými herními enginey na světě. Oba jsou aktivně vyvíjeny, obsahují spoustu různých nástrojů a umožňují vytvořit kvalitní produkt, který lze



Obrázek 3.2: Herní engine **Unreal Engine** [11].

jednoduše publikovat pro mnoho platform. Hlavní předností engineu Unity 3D je jeho intuitivnost, která pro mě jako pro začátečníka je důležitá, na druhou stranu s Unreal Enginem je možné dosáhnout lepšího grafického výstupu.

Pro tuto práci jsem zvolil herní engine Unity 3D [32] a to hlavně kvůli jednodušší práci s engineem, menšímu rozsahu projektu a vzdělávacímu účelu projektu při kterém není realistický vzhled prostředí prioritou.

3.2 Architektura systému simulátoru a uživatelského rozhraní

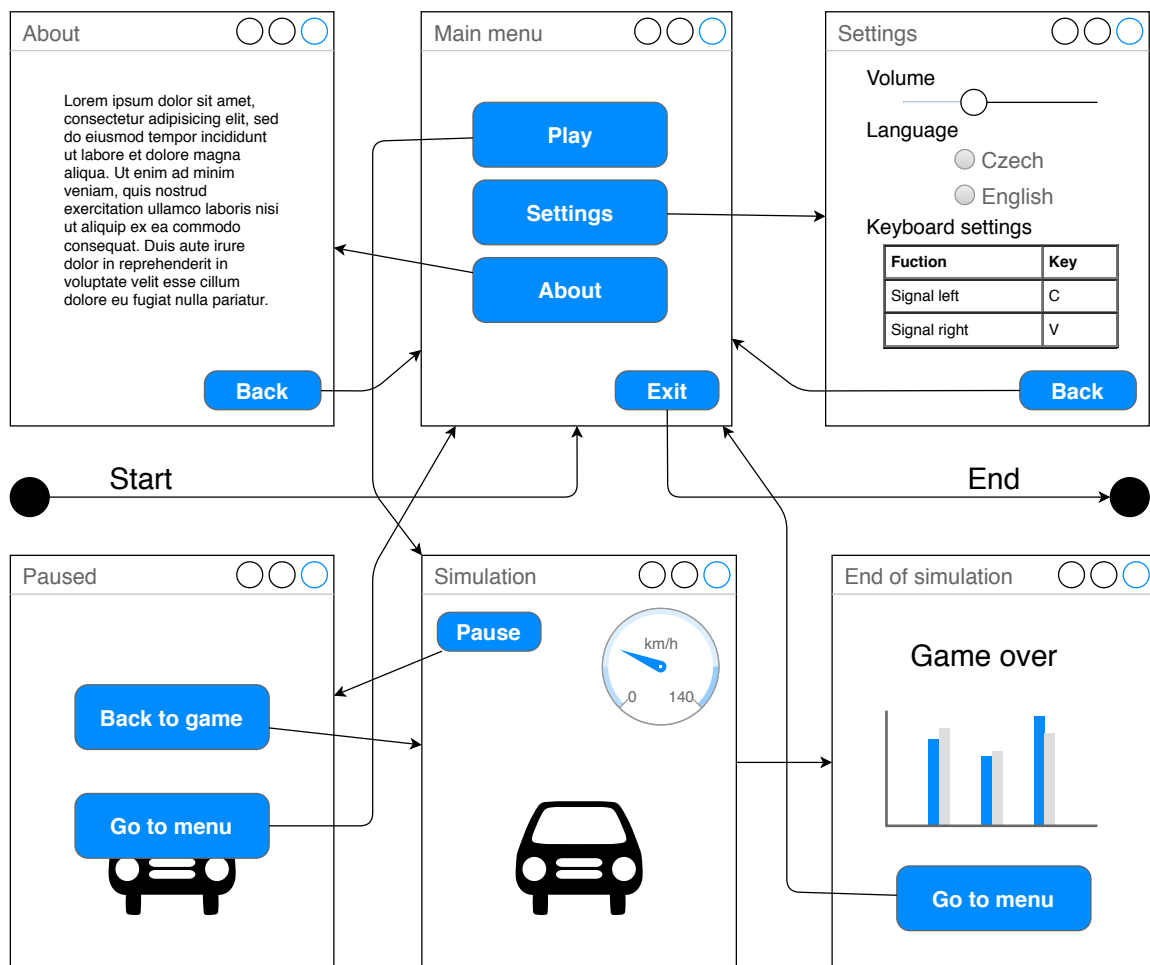
I ten nejmenší projekt může být velmi pracný. Pro programování neexistuje téměř žádná věc, která by byla příliš organizovaná. Když je vše dobře organizované, tak je v projektu mnohem jednodušší číst a psát. Konkrétní úkoly jsou přesné a někdy jednodušší [21].

V této kapitole bude popsán návrh základní struktury programu. Ten úzce souvisí i s výběrem herního engineu v kapitole 3.1.

3.2.1 Uživatelské rozhraní

Uživatelské rozhraní zajišťuje základní interakci s uživatelem, která se odvíjí od nároků na aplikaci. Uživatelské rozhraní hry je jiné než uživatelské rozhraní účetního programu. Po spuštění většiny her se zobrazí menu s malým množstvím voleb, jako je například otevření nastavení, spuštění samotné hry nebo vypnutí aplikace. Tento simulátor se bude držet tohoto jednoduchého schématu, aby bylo jeho uživatelské rozhraní pro uživatele srozumitelné a intuitivně použitelné.

Na obrázku 3.3 je zobrazena architektura uživatelského rozhraní, ze které vychází i návrh architektury simulátoru popsany v 3.2.2.



Obrázek 3.3: **Architektura uživatelského rozhraní.** Východím i koncovým bodem aplikace je hlavní menu.

3.2.2 Architektura simulátoru

Návrh simulátoru závisí na konkrétních nástrojích, které budou pro realizaci využity. V tomto případě jde o herní engine Unity 3D [32]. V této části bude popsána problematika návrhu v herním engine obsahující správu scén i doporučení pro architektonické oddělení jednotlivých částí hry.

Základní koncepce her v Unity 3D [32] je popsána v knize [21]. Samotný návrh začíná obalením základu hry, který by měl být pojmenován jako něco docela obecného, ale něco, co si lze pamatovat. Důvodem je rozlišení od ostatních her, které vytváříme. Vhodným základem hry je jmenný prostor, který slouží jako kontejner pro nejobecnější funkce a třídy projektu. Základ hry by se měl skládat z více tříd a funkcí, které jsou specifické pro danou hru, ale neměl by implementovat skutečné objekty hry. Základ hry by měl implementovat věci jako je spuštění hry, vytvoření scény a pozorování a ukončení hry. Uvnitř základu hry je možné implementovat více specifických jmenných prostorů pro správu scén, správu hráče, správu vstupu atd.

Správa hráče umístěna do hlavní části hry by znamenala existující identitu hráče i v hlavním menu. Podle návrhu uživatelského rozhraní na obrázku 3.3, je takový přístup u tohoto

simulátoru nevhodný, a proto se správa hráče přesune přímo do scény simulace (hry). Tak bude identita hráče vznikat se spuštěním simulace a zanikat s návratem do hlavního menu.

Správa scén v Unity

Komplikovaným prvkem je správa scén, která je velmi specifická pro herní engine Unity 3D [32]. Nejprve bude představena problematika přepínání jednotlivých scén, a poté organizace konkrétních scén.

Rus Scammell (*Product Evangelist, Unity*) [25] rozděljuje základní aplikační logiku na *Main Controller* a scény aplikace. *Main Controller*, který lze nazvat i jako *Scene Manager*, má `public static` funkce, aby k nim mohl přistupovat každý. Existuje po celou dobu běhu aplikace. Jeho základním úkolem je načítat jednotlivé scény a obnovovat jimi použitou paměť.

Main Controller je stavový automat a jeho hlavní funkcí podle Rusa Scammella je přepínání scén. Tato funkce může být nazvána například `switchState()`. Na úplném začátku spuštění aplikace *Main Controller* načte výchozí scénu. Při načítání výchozí scény a při přepínání scén prochází *Main Controller* několika fázemi:

1. *Reset* se pokusí pomocí GC obnovit část paměti.
2. *Preload* začne asynchronně načítat scénu.
3. *Load* čeká, dokud není načítání scény dokončeno.
4. *Unload* se zbaví nepoužívaných zdrojů.
5. *Postload* provede věci, které programátor chce, aby byly provedeny ihned po načtení, a poté nastaví scénu jako aktivní.
6. *Ready* se pokusí pomocí GC obnovit část paměti. Jsou provedeny věci, které mají být provedeny na začátku scény, jako je třeba čekání na vstup od uživatele.
7. *Run* je stav, ve kterém *Main Controller* zůstává, dokud není vyžadována další změna scény.

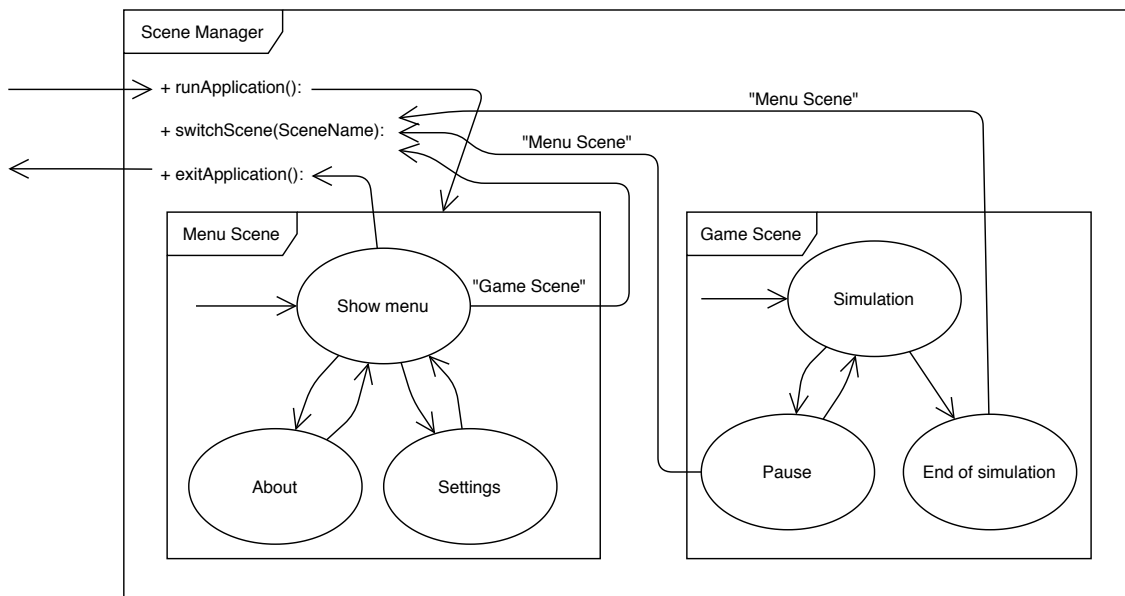
Jednotlivé scény jsou stavové automaty pro související části aplikace. Pro logičtější organizaci aplikace je vhodné mít například okna *Settings*, *About* a *Main menu* v jedné scéně. Dlaším důvodem pro toto spojení je i náročný přechod mezi jednotlivými scénami.

Pro návrh konkrétní scény a správného zahrnutí všech jejích částí slouží stavový automat. Každá scéna má vlastní stavový automat s jedním výchozím bodem. Tato problematika je podrobněji popsána v knize [19].

Správa scén, která tvoří základní kostru aplikace je na obrázku 3.4. Zobrazuje kombinaci správy scén se stavovými automaty jednotlivých scén.

Podpora více národních jazyků

Při návrhu architektury je potřeba zvážit i další požadavky. Na globální úrovni, tedy napříč scénami, jsou udržovány hodnoty (objekty) například pro podporu více národních jazyků a správu vstupů. Hlavním požadavkem pro správu národních jazyků je získat větu jednoho významu aktuálně nastaveného jazyka jedním způsobem. Pro dodržení tohoto požadavku bude jazyk implementován jako rozhraní (*interface*) a konkrétní jazyky (čeština, angličtina) jako třídy implementující toto rozhraní. Jedna funkce bude tvořit jednu větu. Právě



Obrázek 3.4: **Správa scén** navržena pro herní engine Unity 3D [32] se použitím funkce `switchScene()`.

implementace skrz funkce umožní jednoduše doplňovat do vět vlastní hodnoty (maximální povolená rychlost a rychlost hráče). V moment přepnutí jazyka je třeba myslet i na změnu textů v právě zobrazeném uživatelském rozhraní.

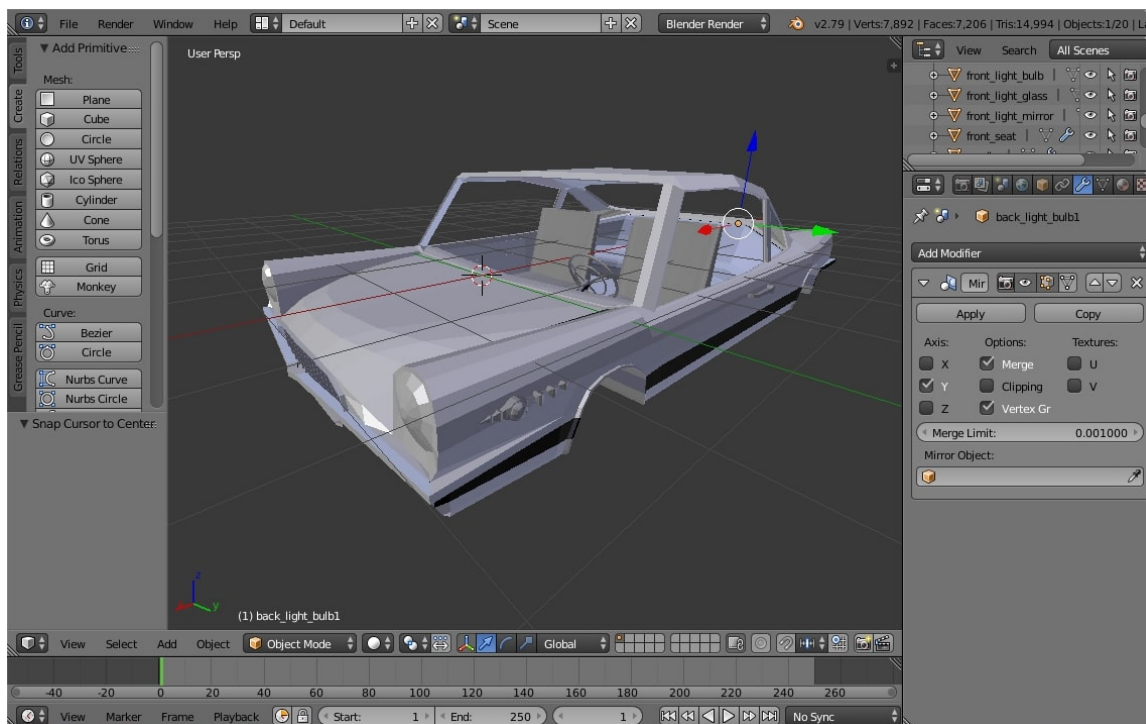
Správa vstupů

Specifické místo v architektuře aplikace má i nastavení vstupů, kterými uživatel řídí simulaci. Správce vstupů je objekt, který udržuje informaci o tom, která klávesa povede ke které akci při simulaci. V celém programu je vytvořen pouze jeden objekt, který musí být přístupný ve všech scénách. Scéna pro menu mění jeho hodnoty v nastavení a scéna hry hodnoty čte a používá.

3.3 3D modely

Modely a textury tvoří důležitou složku v celkovém dojmu aplikace. Existuje nepřehledné množství volně dostupných modelů na různých serverech, které jsou pěkně zpracované. Není problém sehnat i profesionálně vytvořený model, který autor zveřejní, aby si udělal reklamu. Přes tento fakt i potencionálně horší kvalitu vlastních modelů jsem se rozhodl si veškeré modely vytvořit sám. Důvodem pro to může být i různorodost cizích objektů. Například jednoduchý a složitý model by vedle sebe působily nesourodě. Dalším důvodem je model auta, u kterého jsem se rozhodl udělat proporciálně větší kabinu oproti skutečnosti z důvodu umístění kamery. Její vyšší pozice má za důsledek lepší viditelnost okolí z vozidla a lepší vnímání velikosti vozidla (hráč omylem nepřejede čáru u semaforů).

Pro modelování jsem vyzkoušel více jednoduchých editorů 3D modelů, jako je například `SketchUP` [28] nebo `Microsoft 3D Builder` [13]. Nakonec jsem se rozhodl pro open-source editor `Blender` [5], protože funkcionality jmenovaných editorů nebyly dostatečné. Ukázka prostředí programu je vidět na obrázku 3.5. Velké množství nástrojů a zkratk umožňují



Obrázek 3.5: Open-source software pro modelování a vykreslování **Blender** [5].

vytvořit a vykreslit různorodé modely pro mnoho účelů. Například grafici pro vytvoření poutavého videa k novému produktu nebo do filmu potřebují detailní modely a využijí i další funkce programu. Naopak vývojáři her upřednostňují modely jednodušší, kvůli HW náročnosti vykreslování hry. Materiály a textury může vývojář her aplikovat až v herním enginu, stejně tak jako nepotřebuje nasvěcovat model.

Negativem tohoto programu je pro nového uživatele právě velké množství zkratk i prostředí, které si může přepnout a bez správné zkratky neví, jak přepnout pohled zpět. Naučit se pracovat s Blenderem [5] zabere hodně času a opravdu kvalitní model vytvoří pouze profesionálové. Pro mě byla tato práce motivací se s editorem naučit, a proto jsem se rozhodl nepoužít cizí modely a ani je neupravovat.

Kapitola 4

Fyzikální model

Jednou z hlavních funkcionalit, které herní engine vývojářům nabízí, jsou i různé formy předimplementovaného fyzikálních modelů. Použití těchto vestavěných funkcionalit je jednoduché. Například přidáte komponentu `Rigid body` k hernímu objektu a nastavíte její vlastnosti. Objekt se poté při nárazu jiného objektu s komponentou `Rigid body` po nárazu odrazí a na objekt začne působit gravitace. Další důležitou funkcionalitou jsou kolize, které se opět přidávají jako komponenty k objektům v herní scéně.

Mnoho věcí lze jednoduše použít a naopak jejich vlastní implementace by byla velmi náročná a vedla by pravděpodobně k horším výsledkům než vestavěné fyzikální komponenty. Proto je důležité se zaměřit na to, co herní engine neposkytne. U auta máme mnoho fyzikálně vyjádřitelných pohybů a vlastností. Vývojáři engine samozřejmě mysleli na to, že v engine budou vytvářeny i hry s vozidly. Také proto v herním engine `Unity 3D` [32] existuje třída `WheelCollider`, která zajistí chování kol, jako je například prokluz kol a zatáčení. Odpružení neboli tlumiče jsou také předpřipravené herním engine. Jsou to věci obecné pro každé vozidlo. Herní engine neposkytuje fyzikální modely, které jsou závislé na konkrétním vozidle. Tyto hodnoty nemá herní engine kde vzít. Mezi tyto věci patří například převodovka vozidla a jízdní odpory. Při výpočtech těchto hodnot je potřeba mít technickou specifikaci vozidla.

4.1 Přehled působících sil na vozidlo ve směru dopředu

Rovnováhu sil ve směru osy x , tedy pohyb vozidla dopředu a dozadu, je možné zapsat:

$$F_x = O_f + O_s + O_{vz} + O_a + O_t \text{ [N]} \quad (4.1)$$

kde:

- F_x [N] hnací síla v podélném směru
- O_f [N] odpor valení
- O_s [N] odpor sklonu
- O_{vz} [N] odpor vzduchu
- O_a [N] odpor zrychlení
- O_t [N] odpor tahu přípojného vozidla

Nejprve budou popsány jízdní odpory při jízdě vozidla, potom bude vypočtena a graficky znázorněna síla motoru a z ní vypočteno zrychlení. Nakonec bude vypočtena síla při brzdění vozidla.

Hnací sílu vypočteme pomocí převodových poměrů převodovky, otáček motoru a výkonu motoru. Odpor valení se vypočte pomocí koeficientu valení. Odpor vzduchu je závislý na čelní ploše vozidla.

Odpor zrychlení představuje síly, které působí proti pohybu vozidla při změně rychlosti. Vzhledem k velikosti odporu bude tento odpor zanedbán. Také odpor tahu připojeného vozidla lze zanedbat, protože tato práce nebude obsahovat vleky a jiná přípojná vozidla.

4.2 Technická specifikace vozidla

Pro výpočty odporů, sil, rychlostí z otáček, zrychlení a dalších veličin při jízdě vozidla je nutné mít hodnoty z oficiální technické specifikace vozidla. Tyto data spolu s charakteristikou motoru nezveřejňuje každý výrobce automobilů, proto jsem vybral automobil Audi A3 2.0 TFSI.

Tabulka 4.1: Výběr dat z oficiální technické dokumentace vozu Audi A3 2.0 TFSI/147 kW z roku 2007 [3].

Veličina	Zkratka	Hodnota
Maximální rychlost [km/h]	v_{max}	236
Zrychlení 0-100 km/h [s]	a_{0-100}	7,1
Maximální výkon při otáčkách [kW/min ⁻¹ (HP)]	P_{max}	147 (200) / 5100–6000
Maximální točivý moment při otáčkách [Nm/min ⁻¹]	M_{max}	280 / 1800–5000
Pohotovostní hmotnost [kg]	m_p	1320
Celková hmotnost [kg]	m_c	1880
Součinitel čelního odporu	c_x	0,33
Čelní plocha [m ²]	S_x	2,13
Vnější průměr zatačení obrysový [m]	d_c	10,7
Převodové poměry pro rychlostní stupně i_p	i_1	3,357
	i_2	2,087
	i_3	1,469
	i_4	1,098
	i_5	1,108
	i_6	0,927
	i_r	3,990
Stálý převod i_s	i_{s1-4}	3,944
	$i_{s5,6,r}$	3,084

Pro výpočty jsou potřeba i další hodnoty a konstanty, které nelze vyčíst z dokumentace vozu.

4.3 Valivý odpor

Odpor valivý vzniká deformací pneumatiky a vozovky. Je-li vozovka tuhá, pak dochází jen k deformaci pneumatiky. Pneumatika se stýká s vozovkou v určité ploše, kterou nazýváme stopa. V přední části stopy ve směru valení dochází ke stlačování obvodu pneumatiky do

Tabulka 4.2: Další neoficiální hodnoty potřebné pro výpočty.

Veličina	Zkratka	Hodnota
Poloměr kola [m]	r	0,2855
Mechanická účinnost přev. ústrojí	η_{mech}	0,95
Gravitační zrychlení [m/s ²]	g	9,81
Součinitel valivého odporu	f	0,012

roviny vozovky a v zadní části se obvod opět vyrovnává do kruhového tvaru [34]. Valivý odpor vozidla O_f je dán součtem odporu jednotlivých kol:

$$O_f = \sum_i O_{fKi} = \sum_i Z_{ki} * f_{ki} \text{ [N]} \quad (4.2)$$

kde:

- O_{fKi} [N] - valivý odpor kola
- Z_{ki} [N] - radiální reakce vozovky na kole
- f_{ki} - součinitel valivého odporu na kole

Radiální reakce vozidla je rovna tíze vozidla. Dle [34] Dále předpokládáme-li stejné podmínky pro každé kolo, pak bude platit:

$$O_f = G_v * \cos\alpha * f \text{ [N]} \quad (4.3)$$

kde:

- G_v [N] - tíha vozidla
- α - úhel stoupání
- f - součinitel valivého odporu

Součinitel odporu valení se do rychlosti 80 km/h dá považovat za konstantu, poté se mírně zvyšuje, ale stále je valivý odpor menší než odpor vzdušný.

4.4 Vzdušný odpor

Celkový odpor vozidla se určuje z běžného aerodynamického vztahu [34]:

$$O_v = c_x * \frac{\rho}{2} * S_x * (v + v_v)^2 \text{ [N]} \quad (4.4)$$

kde:

- c_x - součinitel čelního odporu
 - S_x [m²] - čelní plocha vozidla
 - ρ [kg * m⁻³] - hustota vzduchu
 - v [m/s] - teoretická pojezdová rychlost
 - v_v [m/s] - relativní rychlost vzduchu vzhledem k vozidlu, působící proti směru vozidla
- Veličina v_v nebude v simulátoru uvažována.

4.5 Odpor stoupání

Při jízdě do kopce a z kopce se projeví odpor stoupání, který je určen rovnicí:

$$O_s = G * \sin \alpha \text{ [N]} \quad (4.5)$$

kde G [N] je celková tíha vozidla a α je úhel, který svírá rovina vozovky s vodorovnou rovinou, tzv. úhel stoupání [34].

4.6 Moment a efektivní výkon motoru

Při koupi vozidla výrobce udává výkon motoru v kW. Hodnota značí maximální výkon, při nižších otáčkách má motor nižší výkon. Výkon motoru je přímo závislý na točivém momentu motoru a přesné hodnoty těchto dvou veličin je nutné naměřit. Závislost efektivního výkonu motoru na točivém momentu je popsána následující rovnicí, do které je nutné převést otáčky motoru do základního tvaru, tedy vydělit šedesáti:

$$P_e = M_m * \omega = M_m * 2 * \pi * n \text{ [W]} \quad (4.6)$$

kde:

- P_e [m/s] - efektivní výkon motoru
- M_m [Nm] - moment motoru
- n [min^{-1}] - otáčky motoru

Konkrétní příklad pro motor Audi 2.0 TFSI je vidět na obrázku 4.1.

Z obrázku 4.1 je možné vyčíst konkrétní hodnoty a zapsat je do tabulky 4.3.

Tabulka 4.3: Točivý moment a efektivní výkon motoru.

Otáčky motoru [min^{-1}]	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
Točivý moment [Nm]	170	240	280	280	280	280	280	280	280	256	232
Efektivní výkon motoru [kW]	18	38	59	73	88	103	117	132	147	147	147

4.7 Pilový diagram

Dobrý přehled o rozvržení rychlostních stupňů dává tzv. pilový diagram $n - v$, který udává rychlost vozidla při jednotlivých rychlostních stupních v závislosti na otáčkách motoru [35]. Závislost těchto veličin popisuje následující rovnice:

$$v_t = \frac{2 * \pi * r * n}{i_s * i_n * 60} \text{ [m/s]} \quad (4.7)$$

2,0 Liter-Vierzylinder-Turbo-FSI-Motor im Audi A3

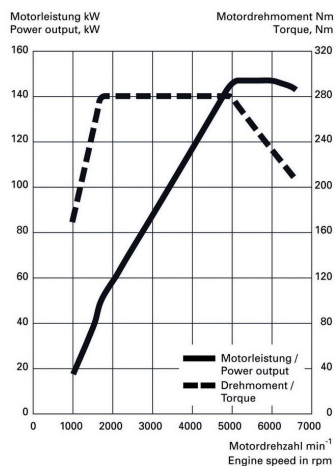


mit Benzindirekteinspritzung, Integral-Turbolader, zwei Ausgleichswellen
kontinuierliche Einlass-Nockenwellenverstellung und Kunststoff-Saugrohr

2.0 litre four cylinder Turbo FSI engine in the Audi A3
with fuel direct injection, integrated turbocharger, two balancer shafts,
variable inlet camshaft timing and plastic intake manifold

1,984 cm³ 147 kW (200 PS) von 5,100 bis 6,000 min⁻¹ 1,984 cc 147 kW (200 bhp) from 5,100 to 6,000 rpm
280 Nm von 1,700 bis 5,000 min⁻¹ 280 Nm from 1,700 to 5,000 rpm

11/07



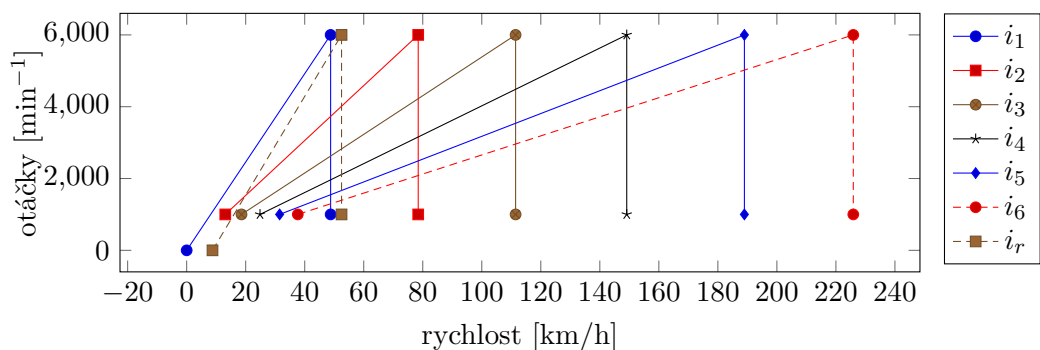
Obrázek 4.1: Charakteristika motoru Audi A3 2.0 Turbo FSI [7].

kde:

- v_t [m/s] - rychlost vozidla
- r [m] - poloměr kola
- n [min⁻¹] - otáčky motoru
- i_s - stálý převod
- i_n - převodový poměr pro rychlostní stupeň n

Tabulka 4.4: Teoretická rychlost v km/h v závislosti na převodových poměrech a otáčkách.

p_n	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
p_1	8,13	12,19	16,26	20,32	24,39	28,45	32,52	36,58	40,65	44,71	48,78
p_2	13,08	19,61	26,15	32,69	39,23	45,77	52,30	58,84	65,38	71,92	78,46
p_3	18,58	27,87	37,15	46,44	55,73	65,02	74,31	83,60	92,89	102,17	111,46
p_4	24,85	37,28	49,71	62,14	74,56	86,99	99,42	111,84	124,27	136,70	149,12
p_5	31,50	47,25	63,00	78,75	94,49	110,24	125,99	141,74	157,49	173,24	188,99
p_6	37,64	56,47	75,30	94,12	112,94	131,77	150,59	169,42	188,24	207,06	225,89
p_r	8,75	13,12	17,49	21,87	26,24	30,61	34,99	39,36	43,73	48,11	52,48



Obrázek 4.2: Pilový graf

Je-li poměr dvou po sobě jdoucích převodů stálý (převody jsou odstupňovány podle geometrické řady), budou ležet spodní špičky pilového diagramu na rovnoběžce s vodorovnou osou [35]. Znamenalo by to, že jde o převodovku s geometrickým odstupňováním. Z grafu 4.2 je vidět, že odstupňování převodovky je progresivní.

4.8 Hnací síla motoru

Se znalostí charakteristiky motoru a v ní obsaženého momentu motoru a dalších parametrů vozidla můžeme vypočítat hnací sílu. Tato síla bude muset překonat všechny jízdní odpory. Pro výpočet hodnot zobrazených v tabulce lze použít následující vztah:

$$F_H = \frac{M_m * i_s * i_n * n_{mech}}{r} \text{ [N]}. \quad (4.8)$$

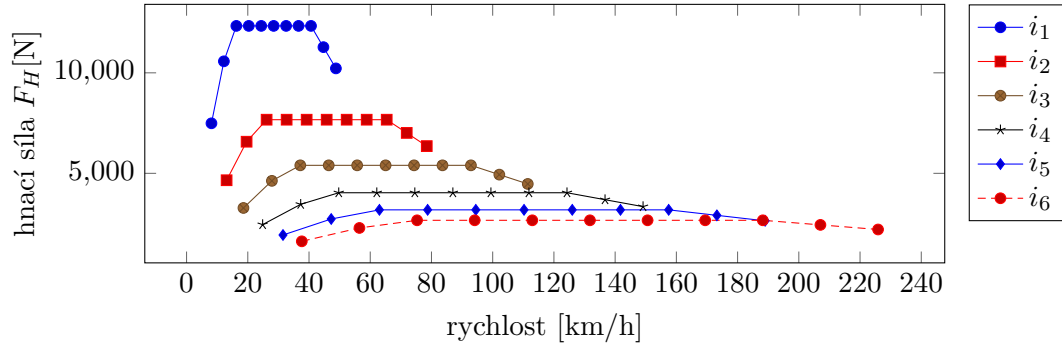
Tabulka 4.5: Hnací síla motoru v závislosti na momentu motoru a převodových poměrech.

n_m	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
M_m	170	240	280	280	280	280	280	280	280	256	232
p_1	7490	10573	12336	12336	12336	12336	12336	12336	12336	11278	10221
p_2	4656	6573	7669	7669	7669	7669	7669	7669	7669	7012	6354
p_3	3277	4627	5398	5398	5398	5398	5398	5398	5398	4935	4473
p_4	2450	3458	4035	4035	4035	4035	4035	4035	4035	3689	3343
p_5	1933	2729	3184	3184	3184	3184	3184	3184	3184	2911	2638
p_6	1617	2283	2664	2664	2664	2664	2664	2664	2664	2435	2207
p_r	6961	9827	11465	11465	11465	11465	11465	11465	11465	10482	9499

Diagram hnací síly motoru zobrazuje rychlost pro jednotlivé rychlostní stupně. Křivky hnací síly jsou vypočteny z vnější rychlostní charakteristiky momentu motoru, tzn. jsou to průběhy maximálních hnacích sil (plný plyn) pro jednotlivé rychlostní stupně. Hodnoty z tabulky 4.5 jsou zakresleny v grafu 4.3.

4.9 Brždění

V předchozích částech jsem se zabýval jízdními odpory a pohybem dopředu. Základní částí pro ovladatelnost a bezpečnost automobilu jsou také brzdy.



Obrázek 4.3: Diagram hnací síly motoru

Brždění vozidla se dosahuje zpravidla záměrně vyvolaným třením mezi rotujícími a pevnými částmi motorového vozidla, např. mezi brzdovým kotoučem a brzdovými čelistmi. Tím se pohybová energie mění ve třecích částech v energii tepelnou, kterou je nutno odvádět do ovzduší, aby nedošlo k poškození brzd [35].

Mezi dovednosti řidiče patří i odhad brzdné dráhy, která se zvětšuje s rychlostí vozidla. Při výpočtu lze vycházet z 2. Newtonova zákona. Obecně pro zrychlený a zpomalený pohyb platí.

$$F = m * a \text{ [N]} \quad (4.9)$$

$$\frac{F}{m} = a = \frac{dv}{dt} \quad (4.10)$$

$$\int \frac{F}{m} dt = \int dv \quad (4.11)$$

$$\frac{Ft}{m} + v_0 = v = \frac{dx}{dt} \quad (4.12)$$

$$\int dx = \int \left(\frac{Ft}{m} + v_0 \right) dt \quad (4.13)$$

$$x = \frac{Ft^2}{2m} + v_0t + x_0 \quad (4.14)$$

Na začátku je 2. Newtonův zákon (4.9). V rovnici (4.10) se k němu přidá vztah pro zrychlení. Integrací v rovnici (4.11) se získá vztah pro rychlost v v závislosti na čase t . Doba potřebná k zastavení se vypočítá ze vztahu pro rychlost (4.12), která se stanoví v konečném čase nulová. Hodnota v_0 se určí z počátečních podmínek. Další integrací (4.13) se vyjádří dráha x v rovnici (4.14), kterou vozidlo urazí. Počáteční dráha x_0 je nulová. Veličina F je součtem všech sil působících na vozidlo (jízdní odpory, brzdná síla).

Kapitola 5

Implementace

Základní kostra jakékoli hry se skládá z herního světa a herní smyčky[9]. Tyto pojmy budou vysvětleny v 5.3 a 5.4. Ve hře je velké množství objektů, kde většina z nich má svou smyčku podmíněných událostí, kterou nepřetržitě vykonává.

5.1 Hlavní menu



Obrázek 5.1: **Hlavní menu** programu. V pozadí je vidět scéna ze hry.

Podle návrhu architektury, existují tři kontrolery scén. Hlavní kontroler, kontroler scény hlavního menu a kontroler herní scény. Hlavní kontroler nemá vlastní scénu. Vytvoří se při spuštění programu a načte scénu hlavního menu, ve které je přítomný kontroler scény hlavního menu.

Na obrázku 5.1 jsou vidět možnosti, které má hráč v hlavním menu:

- spustit hru (Play),

- jít do nastavení (Settings),
- prohlédnout si informace o programu (About) a
- ukončit program (Exit).

Tlačítka pro volby mají černé pozadí, které je z částečně průhledné. Při najetí myší se pozadí tlačítek mění bílé, které je také částečně průhledné.

Do pozadí hlavního menu byla umístěna scéna, kterou lze vidět i v samotné hře. Nad scénou krouží kamera a míří dolů na střed tak, aby si uživatel mohl prohlédnout scénu, ale aby stačila jen malá část herního světa. Menší část herního světa znamená menší HW náročnost.

Na stránce **Nastavení**, které je vidět na obrázku 5.2, uživatel vidí nadpis a tlačítko pro návrat do hlavního menu. V nastavení je možné zvolit jazyk simulátoru (čeština, angličtina) a změnit ovládání simulátoru. Při změně jazyka se okamžitě změní všechny texty i přímo na stránce nastavení. Při změně klávesy ovládání uživatel klikne na tlačítko, které změní barvu. Následující stisknutá klávesa poté nahradí původní klávesu a tlačítko má po stisku klávesy opět původní barvu.



Obrázek 5.2: **Nastavení** v hlavním menu programu.

Na stránce 0 programu uživatel vidí nadpis, tlačítko pro návrat do hlavního menu a základní informace o programu. Mezi tyto informace patří například rok vytvoření nebo jméno autora.

5.2 Správa vstupů

U práce se vstupy je důležité, aby hodnoty byly nastaveny stejně ve všech částech, kde se s nimi pracuje. Herní engine Unity 3D [32] má vestavěnou správu vstupů, která se dá nastavit pouze před spuštěním hry. Ve stejném okno probíhá třeba i nastavení velikosti obrazovky. Implementoval jsem vlastní správu vstupů. Druhým místem, kde se čtou tyto hodnoty je hlavní menu hry. Třetím místem je samotná simulace. Hlavní menu a simulace jsou v jiných scénách, což zesložituje návrh. Napojení třídy `InputManager` na vozidlo přes třídu `CarUserControl` je vidět na obrázku 5.6.

5.3 Herní svět

To co dělá z her tak příjemnou formu zábavy je to, že můžeme prozkoumat imaginární svět a dělat tam věci, které bychom v reálném životě nejspíše nikdy nedělali. Tento imaginární svět ve kterém hrajeme hru se nazývá herní svět [9].

Herní svět může vypadat různě. Například herní svět karetních her na počítači je mnohem jednodušší než herní svět 3D hry. Herní svět této práce má jako základ jednoduchý rovný terén. Ten tvoří podklad a vymezuje hranice herního světa. Na terén se přidávají další objekty hry. To jsou třeba stromy, budovy, silnice, dopravní značky, odpadkový koš a hydranty. Krom nehybných částí herního světa v něm může být i více typů objektů, které se hráči jeví jako aktivní. Jako příklad lze uvést ostatní auta a postavy. Dále lze v herním světě nalézt spoustu objektů, které vidět nejsou, ale také mají svou reprezentaci pozice i jinou v herním světě. Speciální objektem ve hře je vždy objekt hráče, což může být třeba postava, ale v případě této práce je to auto.

Když je hra spuštěna na počítači, tak stroj udržuje vnitřní reprezentaci herního světa. Tato reprezentace nevypadá podobně jako to, co lze vidět na obrazovce při hraní hry. Skládá se převážně z čísel popisujících umístění objektů, kolik zásahových bodů může nepřítel dostat od hráče, kolik položek mají hráči ve svém inventáři atd. Pro uživatele stroj vytváří vizuálně příjemnou reprezentaci tohoto světa, která se zobrazuje na obrazovce [9].

5.4 Herní smyčka

Herní smyčka se zabývá všemi dynamickými aspekty hry. A během hry se děje mnoho věcí. Hráči stisknou tlačítka na gamepadu, nebo se dotknou obrazovky svého zařízení, neustále se mění svět her sestávající z úrovní, příšer a dalších postav, které je třeba nepřetržitě aktualizovat. Existují také speciální efekty, jako jsou výbuchy, zvuky a mnoho dalšího. Všechny tyto různé úkoly, které musí herní smyčka zvládnout, lze uspořádat do dvou různých kategorií:

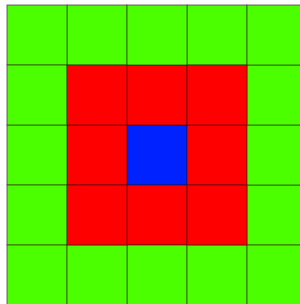
- úkoly související s aktualizací a udržováním herního světa a
- úkoly související s vizualizací herního světa hráči.

Herní smyčka tyto úkoly nepřetržitě provádí jeden po druhém [9].

Za části herní smyčky lze považovat akce, které jsou spuštěny nebo opakovány při hře. Příkladem v této práci mohou být kontroly dodržování předepsaných pravidel, přejezd hráče na jinou část herního světa popsány v 5.5 nebo automatické řízení ostatních aut v silničním provozu.

5.5 Nekonečný svět

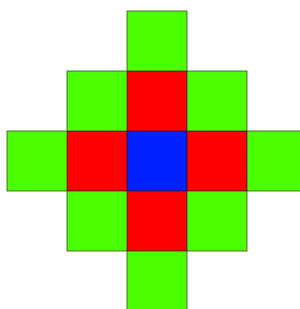
Ve hře jsem implementoval generování nekonečného světa. Základní jednotkou je čtvercová dlaždice. Základem je vytvářet svět před hráčem a odstraňovat svět za hráčem s ohledem na HW zdroje. Impulsem pro změnu herního světa je vstup hráče na jinou dlaždici světa, i proto je nutné udržovat ve hře současně více dlaždic. K určení generovaného okolí byly implementovány dvě varianty jeho tvaru, u kterých lze zvolit i velikost generovaného okolí neboli vzdálenost. První variantou je Moore a druhou von Neumann. Ve výsledném programu je pevně nastaveno von Neumannovské okolí s Manhattan vzdáleností 1.



Obrázek 5.3: Okolí typu Moore a vzdálenost Chebyshev hodnoty 1 pro červenou barvu a hodnoty 2 pro zelenou barvu.

Okolí typu Moore je na obrázku 5.3. Jeho velikost určuje vzdálenost Chebyshev, která je dána vzorcem (5.1). Je známá také jako šachovnicová vzdálenost, kde určuje nejmenší možný počet tahů krále, které musí udělat, aby se dostal na dané pole.

$$\max(|x_1 - x_2|, |y_1 - y_2|) \quad (5.1)$$



Obrázek 5.4: Okolí typu von Neumann a vzdálenost Manhattan hodnoty 1 pro červenou barvu a hodnoty 2 pro zelenou barvu.

Okolí typu von Neumann je na obrázku 5.4. Jeho velikost určuje vzdálenost Manhattan, která je dána vzorcem (5.2). Jeho výhodou je menší HW náročnost, ve scéně je menší množství objektů. Nevýhodou při vzdálenosti 1 jsou rohy. Při přejezdu auta přes roh může

být auto současně na území čtyř dlaždic. I podle obrázku 5.4 je patrné, že jedna z nich musí v danou chvíli chybět.

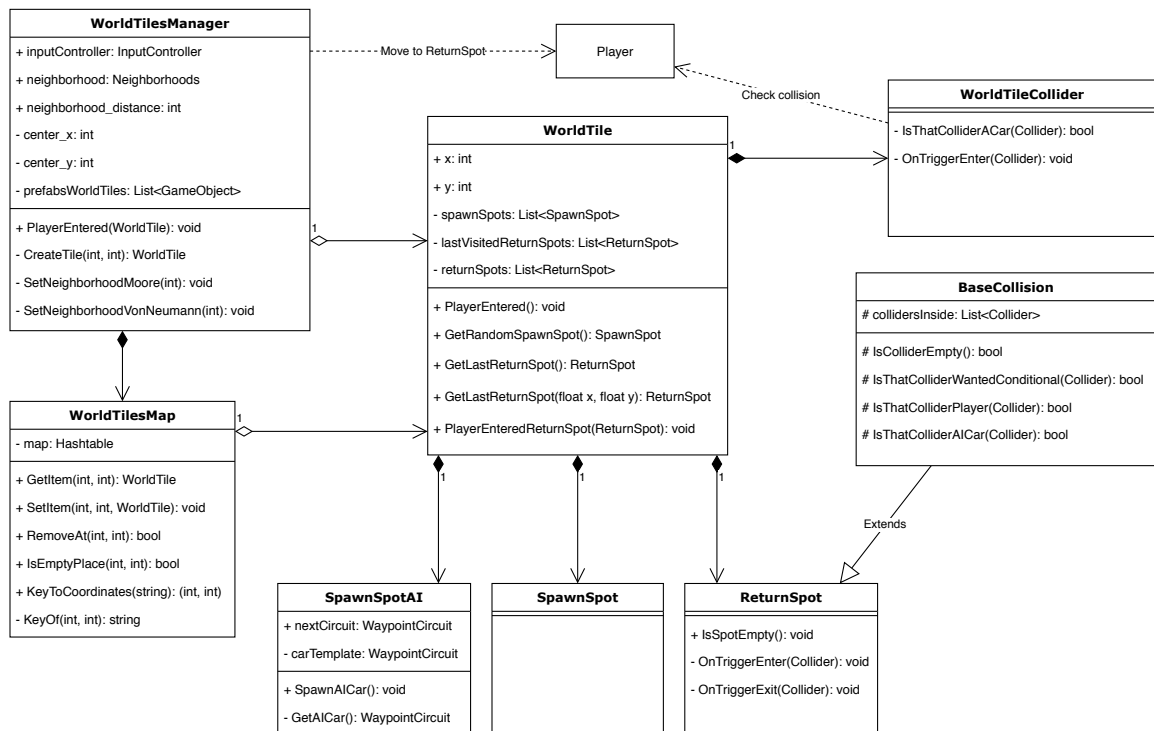
$$|x_1 - x_2| + |y_1 - y_2| \quad (5.2)$$

Způsob implementace nekonečného světa ovlivňuje i terén jeho jednotlivých částí. Obecně terén může být různě zakřivený, pod tím si lze představit kopce, údolí nebo koryta řek. Druhou možností je udělat terén jako rovnou plochu. Právě druhou možnost zvolily některé hry rozebrané v kapitole 2.

Volba rovného terénu má několik opodstatnění. Jednoduššími se stanou následující problémy:

- práce s kolizemi objektů a eliminace neočekávaných odrazů a styků objektů,
- tvorba vozovky,
- generování nekonečného světa při běhu hry (Dále popsáno v této kapitole).

Schéma objektů systému nekonečného světa



Obrázek 5.5: Diagram tříd systému dlaždic herního světa.

WorldTiles je řídicí třídou celého systému. Atribut neighborhood určuje typ okolí a neighborhood_distance jeho vzdálenost. Atributy center_x a center_y udržují souřadnici aktuální středové dlaždice. Atribut prefabsWorldTiles má uložený seznam všech dlaždic, které lze inicializovat a přidat do herního světa. Funkce CreateTile vytvoří herní dlaždici na zadaných souřadnicích. Funkce PlayerEntered dává informaci, že hráč se přemístil na jinou dlaždici s danými souřadnicemi, která bude od této chvíle středovou dlaždicí.

Funkce `SetNeighborhoodMoore` a `SetNeighborhoodVonNeumann` nastavují typ a vzdálenost okolí, pro které se bude generovat herní svět.

`SpawnSpot` je třída, které objekt určuje souřadnice, kam se může přemístit objekt hráče při startu hry, pokud je tato dlaždice první středovou. Fungování tohoto objektu bude více představeno v 5.9.

`SpawnSpotAI` je třída, jejíž objekty určují souřadnice, kde se tvoří instance ostatních účastníků provozu. Ostatní účastníci provozu poté pokračují po trase uvedené v atributu `nextCircuit`, kterého hodnota nahradí původní hodnotu atributu `carTemplate` objektu ostatního účastníka provozu. Fungování tohoto objektu bude více představeno v 5.12.1.

`ReturnSpot` je třída, jejíž objekty určují souřadnice, kam se může přemístit objekt hráče, pokud stiskne klávesu pro návrat na silnici. K ověření že je úsek silnice volný slouží funkce `IsSpotEmpty`. Pokud tímto místem projede hráč, tak je informace oznámena do příslušné instance třídy `WorldTile` pomocí její funkce `PlayerEnteredReturnSpot`. Fungování tohoto objektu bude více představeno v 5.9.

`BaseCollision` je obecněji používaná třída v tomto projektu. Právě funkce této třídy používá třída `ReturnSpot` pro držení informace, zda je úsek vozovky volný.

`WorldTilesMap` je pomocnou třídou třídy `WorldTiles` pro jednodušší správu existujících instancí `WorldTile`. Atributy `x` a `y` jsou souřadnicemi této dlaždice. Funkce `PlayerEntered` je informací, že hráč se přesunul na tuto dlaždici. Informace je předána do objektu třídy `WorldTiles`.

`WorldTile` je třída, která zastupuje jednu vytvořenou instanci herní dlaždice. Pokud je tato dlaždice středovou na začátku simulace, tak pomocí funkce `GetRandomSpawnSpot` se instance třídy `WorldTiles` dotáže, kam přesunout objekt hráče. V průběhu simulace se instance třídy `WorldTiles` pomocí funkce `GetLastReturnSpot` dotáže, na jaké místo vrátit hráče, pokud chce vrátit na vozovku. Vrací poslední navštívený `ReturnSpot`. Seznam posledních navštívených objektů třídy `ReturnSpot` si udržuje v atributu `lastVisitedReturnSpots`. Pokud dostane i souřadnice vozidla, může v případě prázdného seznamu vrátit nejbližší `ReturnSpot`, jinak vrátí náhodný ze seznamu `returnSpots`.

5.6 Cyklus střídání dne a noci

V reálném světě jsou jízdní podmínky v noci jiné než ve dne. Silnice bývají prázdnější, ale viditelnost horší, a to i přes možnou přítomnost pouličního osvětlení. Zrádné jsou průjezdy zatáček, kdy světla míří rovně, ale řidič jede do strany, která je méně osvětlená. Kvůli bezpečnosti existují pravidla pro používání světel popsaná v zákoně 361/2000 Sb. § 32 [1]. Přední světla mohou být obrysová, potkávací, pro denní svícení, mlhová a dálková. Mlhová světla lze používat pouze za mlhy, ta v simulátoru přítomny nebudou, a tedy je možno jejich popis zanedbat. Omezení použití dálkových světel je zjednodušeno pro simulátor na možné oslnění ostatních účastníků provozu a svícení na dostatečně a souvisle osvětlené vozovce. V této části je popsáno především použití předních a zadních světel, protože kontroly směrových světel jsou součástí kontrol průjezdu křižovatek.

Základem tohoto cyklu je slunce a měsíc. Měsíc má za úlohu alespoň trochu osvětit herní svět, aby nebyl absolutně černý v noci. Co se týče pozice v herním světě, tak potřebujeme, aby auto nemohlo dojet ke slunci nebo k měsíci. To znamená, že existuje vazba na pozici hráče. Zároveň, ale přímé přiřazení cyklu k hráči je nevhodně z několika důvodů. Prvním důvodem je naklání hráče například při průjezdu zatáčkou, najetí na retardér nebo jinou nerovnost. Druhým důvodem je zatáčení hráče, které by otočilo i slunce kolem hráče. Cyklus

tedy nesmí být ovlivněn rotací hráče a posunutím po svislé ose. Z toho vyplývá, že cyklu stačí znát vodorovné souřadnice hráče a po nich se posunovat.

Implementačně se cyklus skládá ze dvou světél na opačné straně kruhu, která míří na střed toho kruhu. Světla se po tomto kruhu posouvají ve stejném směru. Při každém posunutí se upraví rotace jednotlivých světél tak, aby světla mířila na hráče. Svítivost slunce je při klesání snižována. Objekt nadřazený světlům je na stejné úrovni jako objekt hráče a udržuje si stejnou pozici na vodorovných osách, jako má objekt hráče. Pro podpoření efektu cyklu, jsou při přechodu dne a noci měněny i globální rendrovací vlastnosti pomocí vestavěné třídy herního enginu `RenderSettings`¹.

5.7 Vozovka

Tvorbu vozovky jsem se rozhodl udělat jako sadu modelů, které se ve hře skládají vedle sebe. Základními modely silnic jsou:

- rovná silnice,
- zátačka s úhlem 90°,
- křižovatka se 4 výjezdy,
- železniční přejezd,
- most a
- kruhový objezd.

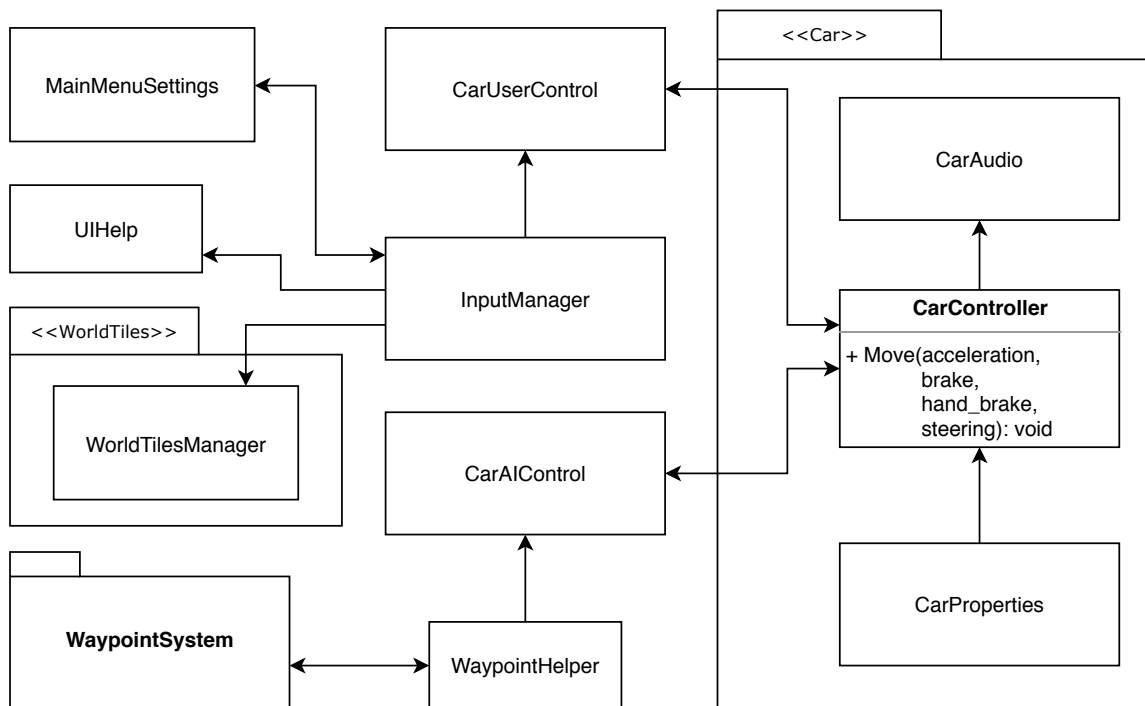
Každý z těchto modelů má několik variant. Liší se zejména v uspořádání pruhů na vozovce nebo v přidání odbočovacích pruhů. Další možnosti rozlišení křižovatek a jiných modelů silnic přímo ve hře je použití dalších doplňujících modelů, jako jsou semaforey, různé dopravní značky nebo dopravní zrcadlo. Tyto modely použijí stejný model vozovky, ale vytvoří jinou situaci pro hráče.

Charakter práce určuje i další specifické požadavky, které přímo ovlivňují i modely vozovky. Ve většině závodních her stačí jednoduché silnice bez složitějšího podélného značení. Pokud před křižovatkou chybí podélná čára, u které má hráč zastavit, pokud nemá přednost, tak hráč může omylem chtít zastavit až těsně za hranicí, kde probíhá kontrola dodržení pravidel. Z tohoto důvodu je u těchto značek dobré mít vyznačenou hranici křižovatky podélným značením. Stejně tomu je i u světelných křižovatek. Dalším faktorem ovlivňujícím tvorbu modelů vozovek je, že pro hráče by nemělo být těžké udržet auto na vozovce ve správném pruhu nebo zatáčet na křižovatce. Hráč má na práci mnoho dalších věcí. Příkladem může být sledování značek a ostatních vozidel na pozemních komunikacích. Pokud je vozovka širší a křižovatka nemá mezi výjezdy ostré hrany, hráč se může více věnovat pravidlům silničního provozu. V neposlední řadě je dobré myslet na různorodost těchto modelů a také jejich variabilitu. Stejný objekt křižovatky může mít různé podélné značení. Tyto věci mohou vytvořit ještě více odlišných situací pro hráče.

5.8 Objekt vozidla

Objekt vozidla se skládá z více podobjektů:

¹Třída `RenderSettings` je součástí `Scripting API` herního enginu Unity 3D [32]: <https://docs.unity3d.com/ScriptReference/RenderSettings.html>



Obrázek 5.6: **Diagram ovládní vozidla.** Vozidlo je ovládáno stejným způsobem buď hráčem nebo skriptem.

- model auta,
- WheelHubs obsahující WheelCollidery,
- kolizní bloky,
- Particles objekt tvořící dým od pneumatik.

Ve hře se objevuje jediný model auta, který jsem vytvořil v programu Blender. Tento objekt obsahuje i kola, ale po nich auto nejede. Jde pouze o vizuální efekt a neobsahují žádný kolizní blok. Kola v modelu jsou otáčena stejnou rychlostí jako jejich přidružené objekty třídy WheelCollider.

Jako základ implementace vozidla byly použity části balíčku **Standard Assets (for Unity 2017.3)** [31], který už byl společností označen za zastaralý a stažen. Jde o balík různých jednoduchších skriptů, materiálů, textur, zvuků, objektů a z nich vytvořených prefabů. Novější náhrada za tento balíček neexistuje. Naprostá většina částí balíčku byla z tohoto projektu odstraněna, protože byla nepotřebná. Použil jsem především některé skripty vztahující se k implementaci vozidla. Existující prefaby pak pro mě sloužily jako příklad, jak skripty použít. Použité části byly upraveny nebo kompletně přepsány tak, aby obsahovaly potřebné funkcionality, a odpovídali novější verzi herního engine Unity 3D [32]. Mezi důležité změny patřila implementace fyzikálního modelu popsána v 5.8.1. Bylo i hodně změn u ovládní aut, ať už hráčova nebo ostatních aut silničního provozu.

Samotné vozidlo, se skládá z více skriptů. Bezprostředně jsou na sebe navázány následující třídy:

- CarAudio,

- `CarProperties` a
- `CarController`.

Hlavní třídou vozidla je `CarController`. Tato třída implementuje fyzikální model a používá `CarProperties`. Oboje bude popsáno v 5.8.1. Třída `CarAudio` vychází z aktuálních hodnot v `CarController`.

5.8.1 Fyzikální model

Fyzikálního model popsaný v 4 je celý součástí třídy `CarController`. Jednotlivé rovnice jsou postupně uspořádané a ukládají hodnotu do této třídy. Rovnice potřebují pro výpočet hodnoty, které jsou technické specifikaci vozidla v 4.1. Na uchování těchto hodnot vznikla třída `CarProperties`.

Vypočtené hodnoty jsou při simulaci zobrazeny v uživatelském rozhraní popsaném v 5.10.3.

Chování auta v herním enginu je ovlivněno více fyzikálními nástroji herního enginu. Kola jsou reprezentována třídou `WheelCollider`². Při simulaci se mění hodnoty atributů `motorTorque` (moment motoru) a `brakeTorque` (brzdny moment) [Nm]. Třída obsahuje atribut `wheelDampingRate`, který není v dokumentaci dostatečně popsán. Po vlastním testování jsem zjistil, že jde o sílu působící proti pohybu vozidla. Může být vývojáři zamýšlena jako valivý odpor, ale nemá dostatečný popis a známé jednotky. Atributu je přiřazena nejnižší dovolená hodnota (0,0001), aby neovlivňoval rychlost vozidla.

Třída `Rigidbody`³ představuje tuhé těleso vozidla. Zde je přístupný vektor rychlosti, kterou se vozidlo pohybuje. Nejdůležitějším atributem je `mass` (hmotnost vozidla). Třída má atribut `drag`, který by bylo možné použít jako odpor vzduchu. Bohužel dokumentace tohoto atributu je velmi strohá, takže není jasné, co atribut přesně dělá a v jakých jednotkách je udávána hodnota. Proto je tento atribut nastavený na výchozí nulovou hodnotu. Důležitou funkcí je `AddForce`, která přidá další sílu působící na těleso. Kola (`WheelCollider`) pohánějí těleso dopředu nebo dozadu. Pomocí funkce `AddForce` je přidána síla působící proti směru pohybu vozidla. Funkci `AddForce` je možné použít případně i pro sílu působící shora, kterou lze vylepšit chování vozidla.

Třída `Collider`⁴ se stará o kolize, které se projeví srážkou dvou objektů nebo voláním funkce. Při přiřazení komponenty `Rigidbody` k objektu, který ve své hierarchii obsahuje objekty `Collider`, se váha umístí rovnoměrně do objektů `Collider`.

5.9 Objekt hráče

Hráč je nejdůležitějším objektem celé hry. Kolem něj se točí slunce, s jeho pohybem se generují a zanikají části herního světa. Jeho dopravní přestupky jsou detekovány a především pro hráčovo auta je implementován fyzikální model, který mu má přiblížit fungování a chování auta.

Objekt hráče je rozšířením vozidla a skládá se z následujících částí:

²Třída `WheelCollider` je součástí `Scripting API` herního enginu `Unity 3D` [32]: <https://docs.unity3d.com/ScriptReference/WheelCollider.html>

³Třída `Rigidbody` je součástí `Scripting API` herního enginu `Unity 3D` [32]: <https://docs.unity3d.com/ScriptReference/Rigidbody.html>

⁴Třída `Collider` je součástí `Scripting API` herního enginu `Unity 3D` [32]: <https://docs.unity3d.com/ScriptReference/Collider.html>



Obrázek 5.7: Objekt, na který se umístí hráčovo vozidlo při začátku simulace.

- model auta,
- WheelHubs obsahující objekty s komponentou WheelCollider,
- Colliders,
- Camera Manager a
- Particles (kouř od pneumatik).

Při začátku hry je hráčovo vozidlo s nulovou rychlostí umístěno do jízdního pruhu silnice ve správném směru na místo, kde neplatí žádná dopravní omezení. K tomu na herních dlaždicích existuje objekt `SpawnPoint`, který je vidět na 5.7. Ty se smažou po správném umístění hráčova vozidla nebo pokud hráčovo vozidlo už ve světě umístěné je. Pro umístění hráče na vozovku při zapnutí simulace jsou pečlivěji vybrána místa, kam je hráč umístěn. Narozdíl od objektů pro návrat na vozovku popsaných v 5.9 je hráč vždy před všemi značkami, které na dané dlaždici herního světa platí.

Kamery

Ve srovnání se závodní hrou musí tato práce mít propracovanější systém kamer. Základním rozdílem je natáčení kamer do stran, kdy hráč potřebuje vědět, že nejede žádné auto, kterému by musel dát přednost. Dále by měl hráč možnost pomocí přepínání kamer vidět auto ze všech úhlů. Pokud si třeba není jistý, kde auto končí, mohl by přejet podélnou čáru před křižovatkou, u které by měl zastavit.

Návrat na vozovku

V jakékoli hře může hráč uvíznout mimo trasu nebo i na trase. Auto se může zaseknout mezi kolizními bloky jiných objektů, jako je dopravní značka nebo strom. Ne vždy to musí být jeho chyba. Občas může jet tak rychle, že skončí mimo silnici. V těchto případech se mu hodí, když ho jedna klávesa vrátí zpět na vozovku.

Oproti závodní hře je návrat na vozovku v autoškole složitější. V jednotlivých částech herního světa máme situace, do kterých není vhodné hráče postavit. Příkladem může být křižovatka a její blízké okolí, přechod pro chodce nebo úsek dopravního omezení se střídavým provozem řízeným semaforem. Hráč musí být schopen z vloženého místa vyhodnotit dopravní situaci před ním a správně reagovat. Hráčovo vozidlo je důležité správně otočit do směru jízdního pruhu, do kterého je přesunut. Je nutné zkontrolovat, že daný úsek vozovky je prázdný. V neposlední řadě je nutné vynulovat rychlost auta tak, aby v nové situaci nemusel nečekaně rychle reagovat a brzdit.

Implementačně jde spíše o část správy herního světa než o komponentu modelu hráče. Důvodem jsou i dlaždice herního světa. Hráč je vždy přesunut pouze v rámci středové (aktuální) dlaždice herního světa. Na jednotlivých dlaždicích jsou ručně rozmístěny objekty s komponentou `ReturnPoint`. Každá dlaždice si najde všechny objekty s touto komponentou. Tyto objekty pak hlásí, když nimi hráč projede. Dlaždice si udržuje seznam těchto míst a hráče umísťuje na poslední z těchto navštívených míst. Pokud žádné není volné, tak je hráč umístěn na nejbližší `ReturnPoint` této dlaždice. Objekt s komponentou `ReturnPoint` obsahuje narázdíl od objektu s komponentou `SpawnPoint` kolizní blok, který kontroluje, zda je úsek silnice volný. Objekt `ReturnPoint` také existuje po dobu existence jeho nadřazené dlaždice herního světa. Ve hře je výrazně vyšší počet objektů s komponentou `ReturnPoint` než objektů s komponentou `SpawnPoint`.

5.10 Herní uživatelské rozhraní

Program obsahuje dva typy uživatelského rozhraní. Prvním je hlavní menu celého programu a obsahuje pouze pár tlačítek. Druhým typem jsou uživatelská rozhraní přímo v simulaci. Ty mají za úkol především informovat hráče o současném stavu simulace.

5.10.1 Náповěda k ovládní

Autoškola je ze své podstaty složitější na ovládní než běžná závodní hra. Krom ovládní směru a rychlosti jízdy jsou u auta další ovládní třeba na světla a směrová světla. Složitější ovládní má i kamera, která musí hráči umožnit správné rozhodnutí v dopravních situacích. To například znamená, že hráči dovolí se na křižovatkách rozhlednout do stran, aby věděl, že do křižovatky nejede jiné auto, které by mělo před ním přednost.

Pro snadnější ovládní jsem vytvořil jednoduchý přehled funkčnosti kláves jako náповědu při hře. Zobrazí se, když hráč drží tabulátor, a zmizí při jeho puštění.

5.10.2 Hlášení jízdních událostí

Hráče je potřeba informovat o dopravních přestupcích, kterých se dopustil. Je možné ho i upozornit, že například riskoval, když přešel přechod pro chodce v rychlosti 80 *km/h*. Toto hlášení má menší důležitost, než když se hráč opravdu poruší pravidla silničního provozu. Zásílané zprávy jsou rozděleny do tří skupin, vzhledem k jejich různosti: informativní (modré), varovné (žluté) a chybové (červené). Uživatelské rozhraní je vidět na [5.8](#).



Obrázek 5.8: Uživatelské rozhraní při hře.

5.10.3 Zobrazení stavu vozidla

Stav vozidla není vždy zřejmý pouze z pohledu na simulaci. Hráč potřebuje znát informace přesně a jasně, aby mohl dělat správná rozhodnutí. Pro správné dodržení rychlosti i nastavení světel auta potřebuje znát aktuální stav vozidla, proto uživatelské rozhraní zobrazuje následující hodnoty:

- rychlost vozidla,
- otáčky motoru,
- směrová světla,
- rychlostní stupeň,
- moment motoru,
- hnací síla motoru,
- zrychlení,
- jízdní odpory vozidla.

Mezi hodnotami je i více hodnot spojených s fyzikálním modelem popsaným v 4. Některým hráčům pomohou tyto hodnoty pochopit chování auta při různých rychlostech. Také lze

podle otáček a předvídat změnu rychlostního stupně převodovky a tedy i změnu chování sil motoru a potažmo možného aktuálního zrychlení vozidla. Uživatelské rozhraní je vidět na 5.8.

5.10.4 Konec simulace



Obrázek 5.9: Uživatelské rozhraní zobrazené po ukončení simulace.

Během simulace jsou události hlášeny hráči pomocí uživatelského rozhraní popsaného v 5.10.2. Na konci simulace se zobrazí přehled z celé simulace, který je na obrázku 5.9. Uživatel si v tomto rozhraní může veškeré situace ze simulace projít. Události lze filtrovat dle typu. Na obrázku 5.9 jsou nejsou zobrazeny informativní události. Ke každé zprávě jsou zaznamenány následující informace:

- typ (pozitivní, informativní, varovná, chybová),
- text,
- ikona,
- herní čas,
- ujetá vzdálenost v dobu zprávy a
- snímek obrazovky.

Právě snímek obrazovky může uživateli připomenout situaci a znázornit chybu. Na obrázku 5.9 je vidět situace, kdy uživatel nepoužil pravé směrové světlo při výjezdu z kruhového objezdu.

5.11 Ostatní objekty herního světa

Implementace všech kontrol dodržování pravidel silničního provozu je pro uživatele neviditelná. Nevidí umístění kolizních bloků a ani jejich funkcionalitu. Do hry bylo vytvořeno velké množství různých modelů, které mají hráči jednoznačně určit situaci tak, aby mohl správně reagovat. Mezi tyto objekty patří:

- semaforey,
- značky,
- směrové sloupky,
- vodící desky a
- podélné značení.

Krom objektů potřebných pro silniční provoz je zde i velké množství objektů, které mají vylepšit vzhled hry. I když hlavním úkolem je atraktivita hry, tak tyto objekty mohou nepřímo ovlivnit dopravní situace například snížením viditelnosti. Jde o nejrůznější objekty, jako jsou například:

- chodníky,
- stromy,
- budovy,
- ploty,
- zdi,
- svodidla,
- větrné elektrárny,
- hydranty,
- odpadkové koše a mnoho dalších.

hydrant, zeď, koleje, dopravní značky, pouliční lampy ...

5.12 Účastníci provozu

Důležitou součástí simulátoru jsou i ostatní vozidla, která hráče dostávají do konkrétních situací, které musí vyhodnocovat. Kdyby hráč jezdil sám, tak lze kontrolovat akorát maximální rychlost, jízdu na červenou a případně jízdu v protisměru.

Implementačně je celý systém ovládání vozidla napojený na stejné auto, jaké má hráč. Pro pohyb si vystačí s jednou funkcí, které předává hodnotu sešlápnutí pedálů plynu a brzdy,

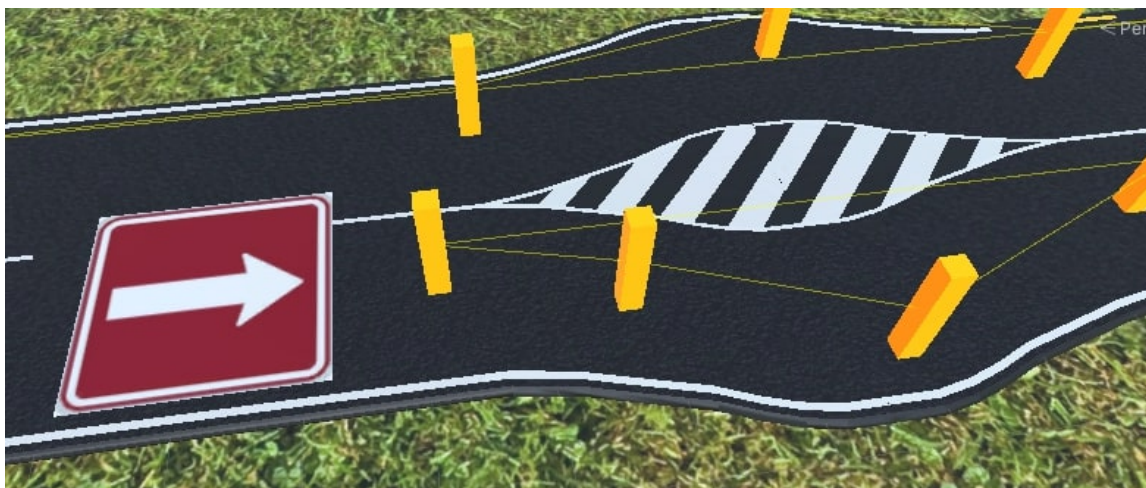
natočení volantu a hodnotu užití ruční brzdy. Dále ovládá světla vozidla, která nejsou přímo propojena s modelem vozidla. Ostatní logiku za naváděním na cílový bod řeší skripty napsané konkrétně pro tyto samoříditelná vozidla.

V této kapitole jsou popsány jednotlivé problematiky, které je nutno u samořízených vozidel řešit. U samotného navádění vozidel po trase je implementace rozdělena na dvě části:

- navádění směru vozidla a zapnutí ukazatelů směru,
- dání přednosti ostatním vozidlům na pozemní komunikaci.

Směr a zapnutí ukazatele směru je nezávislé na přibrždění nebo zastavení z důvodu dání přednosti jinému účastníkovi provozu.

5.12.1 Začátek a konec života objektu



Obrázek 5.10: Objekt, který tvoří nová auta pomocí své komponenty `SpawnPointAI`.

Objekt, který se má po herním světě pohybovat, by měl existovat minimálně po dobu, kdy je viditelný z pozice hráče. Z tohoto předpokladu vychází, že pokud hráč bude objekt následovat, tak by objekt ideálně měl vědět, co má dělat vždy a za každé situace. Z tohoto důvodu, jsem systém navádění ostatních aut navrhl tak, aby auto v nekonečném světě popsaném v 5.5 mohlo existovat stále.

Hlavním impulsem pro změnu herního světa a tedy i pro začátek tvorby objektů je přejetí hráče na jinou dlaždicí herního světa.

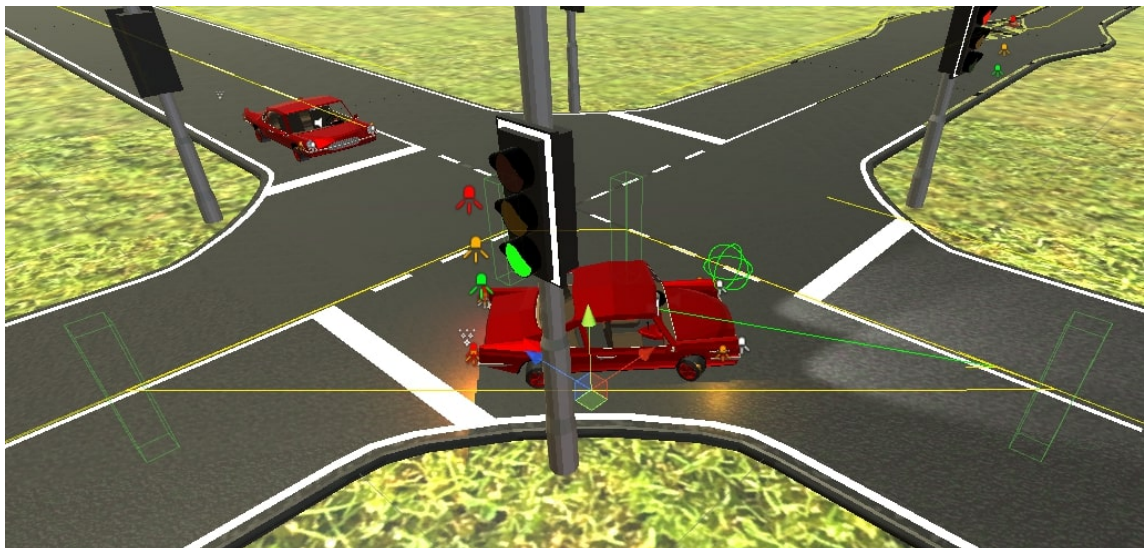
Objekty ostatních aut jsou vytvářeny při tvorbě dlaždic herního světa. Na dlaždicích se nachází objekty s komponentou třídy `SpawnPointAI`, který lze vidět i na 5.10. Ty vytváří novou instanci AI auta, a nastavují jeho pozici a rotaci v herním světě. Nakonec se autu přiřadí část trasy jízdy tak, jak je popsáno v 5.12.2.

Konec existence auta nastane pokud jeho výšková souřadnice klesne pod přesně danou hodnotu, která je pod úrovní herního světa. Tato situace nastane pouze ve dvou případech. Prvním případem je přejetí okraje herního světa. V tomto případě auto přepadne přes okraj herního světa a padá dolů. Druhým případem je přesun hráče na jinou herní dlaždicí. Při tomto přesunu se vygeneruje svět před hráčem, kde se vytvoří nová auta, ale zároveň se

odstraní dlaždice herního světa za hráčem, kde ostatní účastníci provozu začnou padat dolů až po požadovanou hodnotu pro jejich smazání.

5.12.2 Směr jízdy vozidla a zapnutí směrových ukazatelů

Z hlediska implementace lze průjezd křižovatkou rozdělit na dvě nezávislé části.



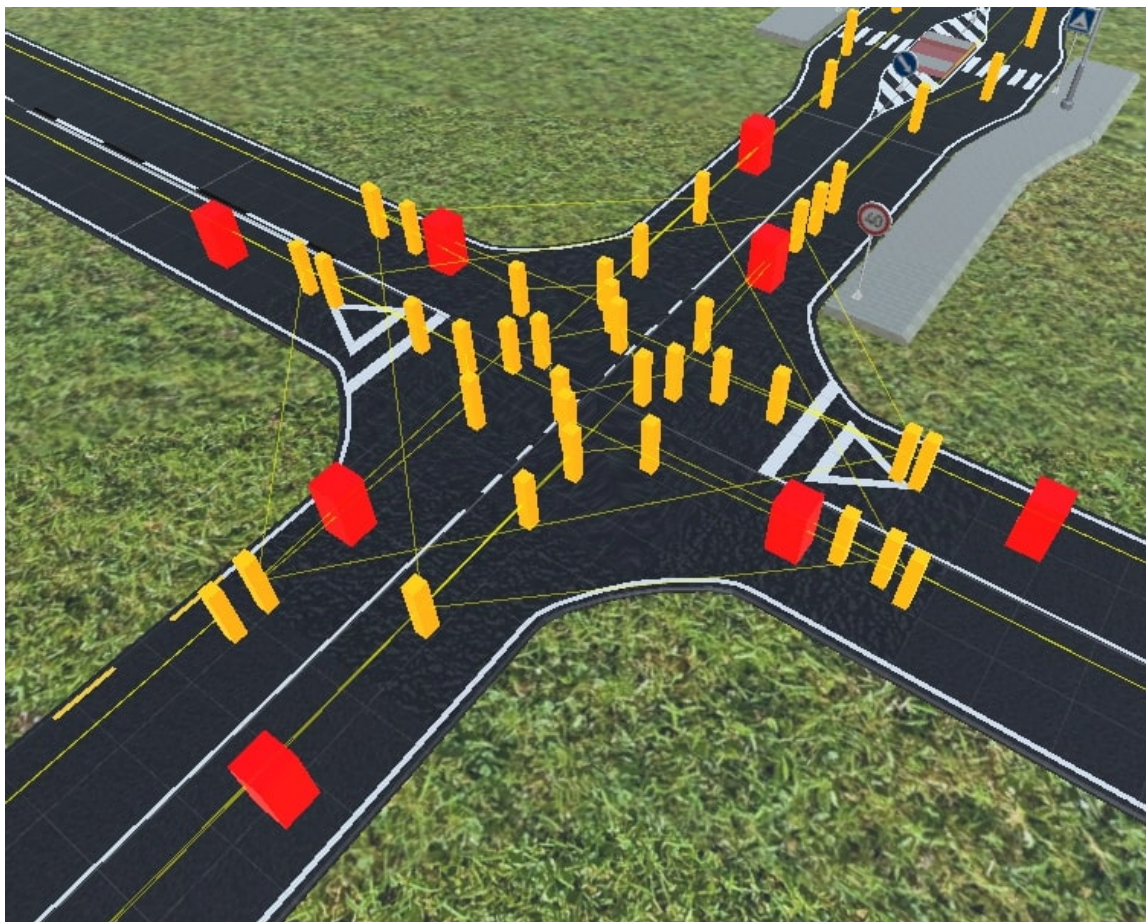
Obrázek 5.11: Jízda AI auta po okruhu složeném z bodů.

První částí je trajektorie průjezdu a zapnutí ukazatelů směru. Trasa průjezdu i ovládání světel auta nezávisí na přednostech, které musí dát ostatním autům, při průjezdu křižovatkou. Pokud auto musí dát na své trase přednost jinému autu, tak ho to na jeho trase pouze zpomalí nebo zastaví.

Zobrazení trajektorie průjezdu křižovatkou se čtyřmi výjezdy s hlavní silnicí vedoucí rovně je zobrazeno na obrázku 5.12. Zobrazení trajektorie průjezdu kruhovým objezdem se čtyřmi výjezdy je zobrazeno na obrázku 5.13. Na obrázcích 5.12 a 5.13 je vidět velké množství červených a žlutých objektů. Tyto objekty slouží k nekonečnému náhodnému navádění auta po různých pozemních komunikacích v herním světě.

Žluté objekty tvoří okruh, po kterém vozidlo jede. Jde vždy o seznam vybraných žlutých objektů. V tomto seznamu je důležité pořadí jednotlivých bodů. Při přiřazení nebo resetování stavu průjezdu aktuálním okruhem se vozidlo rozjede k prvnímu bodu z aktuálního okruhu. Na obrázku 5.11 je vidět navádění vozidla po okruhu ze žlutých objektů a také jeho délka v případě průjezdu křižovatkou. Obecně jsou ve hře okruhy, které vedou od konce herní dlaždice ke křižovatce. A poté krátké okruhy pro každý možný směr odbočení na křižovatce. Než dojde auto na konec aktuálního okruhu, je potřeba mu předat jiný okruh, po kterém má ve své cestě pokračovat. K tomu účelu slouží červené okruhy.

Červené objekty mají definovanou událost při kolizi se samořízeným autem. Mají tři různé hodnoty: okruh pro jízdu vlevo, okruh pro jízdu vpravo a okruh pro jízdu rovně. Náhodně se zvolí, který okruh auto dostane. Všechny hodnoty nemusí být vyplněny. Například uprostřed kruhového objezdu na obrázku 5.13 mají objekty přiřazené okruhy žlutých objektů pouze pro jízdu vpravo a rovně. Červené objekty na výjezdech z kruhového objezdu mají přiřazený pouze okruh pro jízdu rovně. Červený objekt vždy náhodně zvolí z dostupných možností jednu, kterou předá AI autu. Pokud je hodnotou jízda vpravo, zapne pravý ukazatel směru,



Obrázek 5.12: Trasy průjezdu AI auta křižovatkou.

pokud je hodnotou jízda vlevo, zapne levý ukazatel směru, a pokud je hodnotou jízda rovně, vypne ukazatele směru.

5.12.3 Zastavení před překážkou

Z různých důvodů se před autem může vyskytnout překážka. Hlavním případem, který je potřeba zvládnout je jiné auto před ním, které třeba jede nižší rychlostí nebo čeká, až bude moci vlet do křižovatky. Před autem je proto oblast, ve které když se objeví jiné auto, tak své auto zastaví. Oblast je implementována jako kolizní blok. V této oblasti by se také mohly objevit i jiné objekty herního světa. Například železniční přejezd má po silnici nájezd, který by mohl způsobit, že by se do kolizního bloku před autem dostala vozovka. To je druh objektu, který auto a jeho jízdu přes tento kolizní blok ovlivnit nesmí. Dalším příkladem mohou být obleky u zatáček vozovky. Dopravní značka směrové šipky doprava před ostrůvkem s přechodem pro chodce je ve středu obou jízdních pruhů. Díky své šířce by mohlo spadnout do kolizního bloku, ale auto v tu dobu jede doprava a je to spíše záležitostí správného navádění po trase popsaném v 5.12.2, aby ze své trasy auto příliš nevybočovalo.



Obrázek 5.13: Trasy průjezdu AI auta kruhovým objezdem.

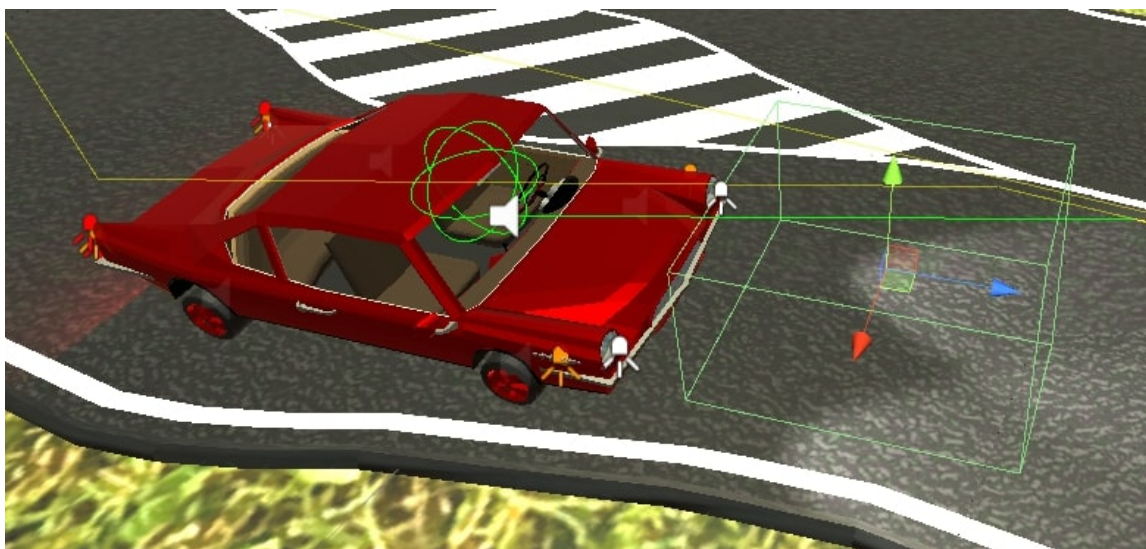
5.12.4 Přechod mezi dlaždicemi herního světa

Přejezd vozidel mezi dlaždicemi herního světa je zajištěn červenými objekty popsány v 5.12.2. Ty mají přiřazený okruh žlutých bodů pouze pro jízdu rovně. Výjezdové ukruhy na dlaždicích herního světa jsou úmyslně vedeny až za okraj dané dlaždice tak, aby při přejezdu došlo ke kolizi s červeným objektem sousední dlaždice, a tedy i k přiřazení dalšího okruhu žlutých objektů.

5.13 Reprezentace křižovatek a jiných dopravních úseků s upravenou předností vozidel pro řízení provozu

Jednou z nejdůležitějších částí práce je samotná kontrola dodržení pravidel silničního provozu. Speciálními a nejsložitějšími kontrolami jsou kontroly dodržení přednosti vozidel na křižovatkách. Právě kvůli těmto kontrolám je potřeba znát stav křižovatky. To zahrnuje pohyb ostatních aut uvnitř křižovatek, pohyb aut přijíždějících ke křižovatce, ale také zde může být jiný element, jako jsou například světelné signály.

Velkou roli při návrhu reprezentace křižovatek hrajou i ostatní účastníci provozu popsání v 5.12. Pro tyto účastníky provozu musí platit stejná pravidla jako pro hráče, jinak nelze po hráčovi dodržování těchto pravidel požadovat. Jak bylo popsáno v 5.12.2, tak ostatní účastníci provozu jsou implementačně rozdělení na dvě hlavní části. První je navádění směru a zapínání směrových světel. Druhou je dání přednosti na křižovatkách. Pro druhou část je potřeba auto přibrzdit nebo zastavit na křižovatkách a to je důležitý faktor při návrhu reprezentace křižovatek.



Obrázek 5.14: Kolizní blok před vozidlem.

K reprezentaci křižovatek je možných více přístupů. Ať jde o graf spojující všechny vstupní a výstupní jízdní pruhy, kde se spojovací čáry různě protínají, nebo o jednoduché kolizní bloky na vstupních a výstupních částech křižovatky. Oba tyto způsoby mají slepé místo uvnitř křižovatky. Křižovatka nezná pozici vozidel v této ploše, která může například prozradit, že už auto projelo část křižovatky, ve které překáželo jinému autu, které čekalo, až se bude moci rozjet.

Aby reprezentace křižovatky zvládla správně detekovat průjezd hráče i řídit ostatní účastníky provozu a vyhnula se problému s hluchým místem uvnitř křižovatky, tak jsem vytvořil systém zón, který je vždy použit v souladu se specifickými pravidly platnými pro konkrétní křižovatku. Nejdříve bude popsáno samotné fungování zón a poté konkrétní rozmístění zón na různých typech křižovatek. Jde o prerekvizitu pro kontroly dodržování pravidel silničního provozu popsané v 5.15 a pro řízení provozu ostatních vozidel popsaného v 5.14.

Základní implementace systému zón křižovatek

V systému zón křižovatek existují jednotlivé zóny. Implementace zóny je stejná pro všechny křižovatky včetně všech jejích použití v nich. Veškeré události ze zón jedné křižovatky zpracovává kontrolér událostí, který ví, jak s kterou zónou pracovat.

Zóna křižovatky

Třída `JunctionZone` vychází z třídy `BaseCollision`, která bude popsána jako první. V simulátoru má tato třída širší užití.

Třída `BaseCollision`. Vytváří abstrakci nad herním světem. Pro kontroly kolizí, správností nastavení některých hodnot nebo hlášení chyb je potřeba mít přístup k více objektům v herním světě. Většina z těchto objektů jsou uvnitř některého z jmenných prostorů, i to je důvod vytvoření této abstraktní vrstvy. Tato třída nemá implementované události kolizí, pouze poskytuje pomocné metody.

Třída `JunctionZone` je vytvořena pro systém křižovatek, který řídí kontrolér křižovatky. Základní úlohou je kolize zpracovávat, filtrovat a informaci o kolizi přeposlat kontroléru. V rámci třídy lze ke kontrétním typům kolizí přidružit některé z předdefinovaných událostí, které se pak automaticky provedou. Zóna si také udržuje informaci o tom, zda je právě rezervována. Rezervace se ukládá jako `Collider`, který rezervaci provedl, žádný jiný `Collider` nemůže rezervaci zrušit. Uvolnit blok tedy zvládne pouze stejné auto, které si ho rezervovalo.

Kontrolér křižovatky

Kontrolér křižovatky zpracovává události v jednotlivých zónách na základě následujících informací, které o událostech má:

- typ auta (AI auto, hráč),
- uložených informací o konkrétním autě (vstupní místo, očekávané výstupní místo),
- typ kolizního bloku (registrační, vstupní, průjezdový, výstupní),
- typ události v kolizním bloku zóny (enter, stay, exit),
- stavu zapnutí směrových světel (levé, pravé, žádné) a
- stavu dalších ovlivňujících elementů (světelné signály).

Většina uložených informací o stavu průjezdu konkrétního auta křižovatkou je používána jenom u hráče a je specifická pro konkrétní křižovátku.

Na obrázku [5.15](#), [5.17](#) a [5.18](#) je vidět více různých umístění objektů zón. Lze je rozdělit do skupin na:

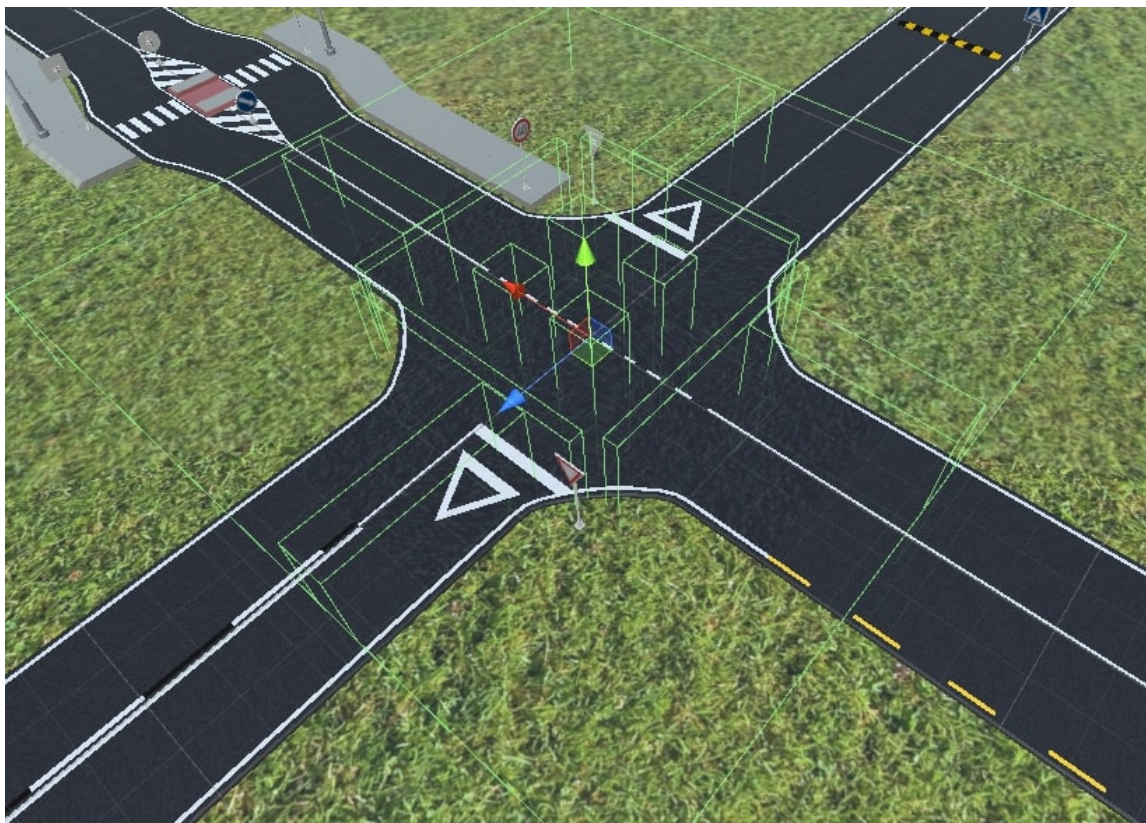
- registrační,
- vstupní,
- průjezdové,
- výstupní.

Registrační bloky mají za úkol brzdit AI auta v případě, že se v okolí nebo uvnitř křižovatky vyskytuje jiné auto, které má před tímto autem přednost. Používá se zde událost typu `stay`. Hráč by v tuto chvíli měl brzdit sám, aby nevjel do křižovatky. Rezervační blok hráč musí projít jako první, je to kontrola, aby do křižovatky nevjel z protisměru.

Vstupní bloky u hráče mimo jiné obvykle kontrolují, zda mohl vjet do křižovatky. U AI aut je implementace odlišná pro každou křižovátku.

Průjezdové bloky nejsou u každé křižovatky nutné, někde je jejich použití pouze velmi omezené.

Výstupní bloky uvolňují cestu ostatním autům, pokud nebyla její potřebná část uvolněna dříve v rámci průjezdových bloků. Probíhají zde další kontroly hráče a mazání uložených informací o průjezdu vozidla křižovatkou. U některých křižovatek lze tyto bloky spojit do jednoho.



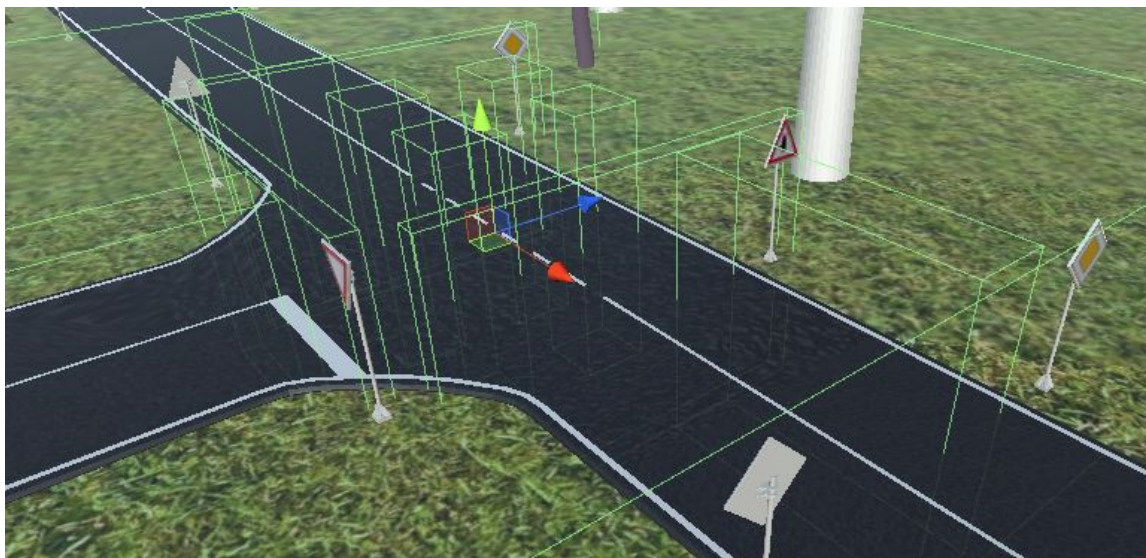
Obrázek 5.15: Rozmístění kolizních bloků na křižovatce se čtyřmi výjezdy a hlavní silnicí rovně.

Křižovatka s hlavní silnicí rovně

Křižovatka obsahuje tři skupiny zón, protože funkcionalita výstupních zón je přidružena ke vstupním zónám. Rozmístění zón lze vidět na 5.15. Tato křižovatka je implementačně nejsložitější, důvodem je existující rezervace průjezdových zón.

Základní kontrolou je kontrola možnosti vjezdu do křižovatky. Kontrola probíhá stejně pro AI auto i pro hráče, pouze se liší blok ve kterém se kontrolu a reakce na výsledek kontroly. Samotná kontrola se skládá ze dvou částí. První částí je přednost aut mimo křižovatku a druhou částí je kontrola volnosti potřebných zón uvnitř křižovatky. Kontroly jsou závislé i na směrovém světle auta. Při jízdě doprava se například kontroluje jenom jedna nejbližší průjezdní zóna, při jízdě rovně dvě zóny na plánované trase a při jízdě vlevo tři zóny. Při vjezdu probíhá rezervace těchto zón tak, aby ostatní auta věděla, že nesmí do křižovatky vjet. Kontrolér okamžitě uvolňuje rezervace zón a jejich předchozích zón, kterými auto projíždí. Pokud se rezervují 3 zóny a auto neprojde křižovatku ideálně a třeba projede zónou první a poté až třetí. Tak při průjezdu třetí se uvolní i druhá zóna, protože je jednou z předchozích zón třetí zóny. Při opuštění křižovatky se uvolní veškeré zabrané zóny tohoto auta.

Kontrola dodržení pravidel je psána tak, aby křižovatka mohla mít jednu nebo dvě vedlejší silnice. Křižovatky na obrázcích 5.15 a 5.16 jsou ovládány stejným skriptem. Při kontrolách se kontroluje, zda kolizní boky pro daný výjezd existují. Například pokud má



Obrázek 5.16: Rozmístění kolizních bloků na křižovatce se třemi výjezdy a hlavní silnicí rovně.

hráč zapnuté pravé směrové světlo a silnice doprava z křižovatky nevede, tak je to chyba. U ostatních aut je problém vyřešen v rámci Waypoint systému.

Světelná křižovatka

Křižovatka obsahuje tři skupiny zón, protože funkcionalita výstupních zón je přidružena ke vstupním zónám. Rozmístění zón lze vidět na 5.17. Jde o velmi podobnou křižovatku jako je křižovatka se čtyřmi výjezdy bez semaforů. Implementačně semaforey ulehčují kontroly dopravních přestupků i řízení ostatního provozu na tomto typu křižovatky. Pokud auto jede rovně nebo doprava, tak jedinou podmínkou je stav semaforu. Pokud jede auto rovně, tak podmínkou je stav semaforu a přednost protijedoucího auta.

Kruhový objezd

Křižovatka obsahuje čtyři skupiny zón. Rozmístění zón lze vidět na 5.18. Unikátnost křižovatky je v tom, že kontroly jednotlivých výjezdů nepotřebují znát žádné informace od ostatních výjezdů. Vždy jde o skupinu čtyř kolizních objektů. Průjezdový blok musí být prázdný, pokud auto chce vjet na kruhový objezd.

Odbočovací a připojovací pruh

Dopravní úsek s odbočovacím nebo připojovacím pruhem obsahuje dvě zóny, jednu lze označit jako vstupní a druhou jako výstupní. Rozmístění zón lze vidět na 5.19.

Úsek s dočasným dopravním omezením řízený semaforem

Jde o úsek se dvěma výjezdy. Implementačně je řízení tohoto úseku podobné řízení výše popsaných křižovatek, ale jsou zde i odlišnosti. Při vjezdu auto nejdříve projede registračním blokem, poté vjede vstupním blokem a na druhé straně vyjede vstupním blokem druhého vjezdu. K správnému určení chování v jednotlivých vjezdech si kontrolér úseku udržuje



Obrázek 5.17: Rozmístění kolizních bloků na křižovatce se světelnými signály.

informaci o tom, ze které strany které auto vjelo a čím projelo. Komplikací zejména při změnách směrových světel AI vozidel je to, že auta projíždějí velkou část stejným pruhem vozovky.

Při kontrole možnosti vjezdu do úseku se kontroluje stav semaforu a nepřítomnost vozidel, která do úseku vjela opačným vjezdem.

Zúžení vozovky

Kontrolér tohoto úseku je až na kontrolu velmi podobný kontroléru popsanému v 5.13. Přednost zde určují značky značící přednost protijedoucích vozidel.

Při kontrole možnosti vjezdu do úseku se kontroluje nepřítomnost vozidel, která do úseku vjela opačným vjezdem. Pokud se vjíždí ze směru, který nemá přednost, tak se kontroluje i prázdnota úseku vozovky před směrem, který má přednost.

5.14 Dodržení přednosti jízdy ostatních vozidel

Nutnou součástí implementace ostatních aut je i jejich správné chování na křižovatkách. Směr jízdy vozidla a správné zapnutí ukazatelů směru byly popsány v 5.12.2. Zbývá tedy auto podmíněčně zastavit především před křižovatkou. Jednotlivé křižovatky jsou v určení podmínek zastavení velmi odlišné. Pro příklad lze vzít kruhový objezd, kdy autu stačí vědět, že nevjede do cesty jinému autu na části kruhového objezdu, která je před jeho vjezdem. Naopak na světelné křižovatce musí mít auto na semaforu rozsvícené zelené světlo. Na nesvětelné křižovatce zase platí pravidla pro přednost zprava a právě tato křižovatka je implementačně z pohledu pravidel silničního provozu nejsložitější.



Obrázek 5.18: Rozmístění kolizních bloků na kruhovém objezdu.

Křižovatka s hlavní silnicí rovně

Křižovatka má implementovanou kontrolu možnosti vjezdu do křižovatky popsanou v 5.13. Registrační bloky zpomalují auto při kolizi typu `stay` na základě kontroly. Vstupní bloky se snaží opakovaně rezervovat průjezdové bloky pro toto auto, dokud se jim to nepodaří. Zde dochází k překryvu, auto je vzhledem k umístění bloků zároveň ve vstupním a v registračním bloku. Jakmile má auto rezervované bloky, tak ho registrační blok ho nebrzdí a může se rozjet. Průjezdové bloky mohou být rezervované nebo volné jak bylo popsáno v 5.13 o třídě `JunctionZone`. Výstupní bloky uvolní všechny tímto autem registrované průjezdové bloky.

Světelná křižovatka

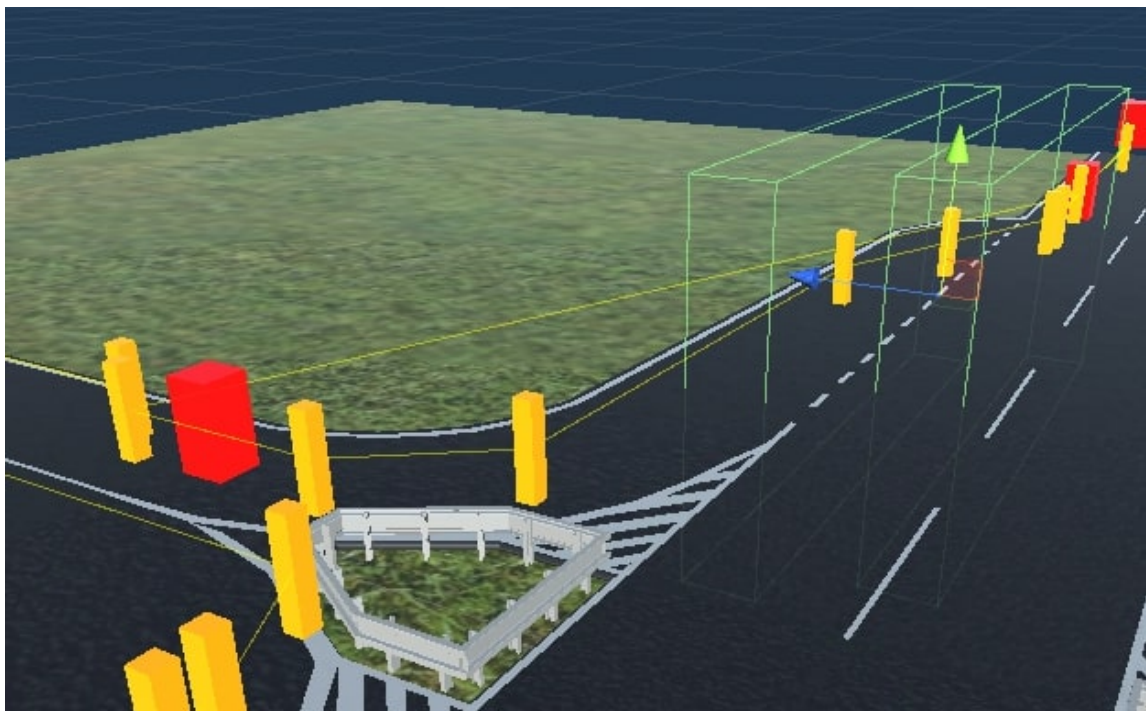
Na světelné křižovatce se AI auta chovají tak, že na oranžovou barvu brzdí. Kolizní bloky jsou stejné jako u předchozího typu křižovatky, ale neprobíhá zde rezervace zón. Rozmístění kolizních bloků bylo představeno v 5.13.

Kruhový objezd

Při vlezdu na křižovatku se kontroluje prázdnota průjezdního bloku, jak bylo napsáno i v 5.13. Vše ostatní je vyřešeno v rámci navádění vozidla popsaného v 5.12.2.

Úsek s dočasným dopravním omezením řízený semaforem

Na úseku s dočasným dopravním omezením řízeným semaforem, který je vidět na obrázku 5.20, jsou dva druhy kolizních bloků, které jsou obdobné kolizním blokům na křižovatkách.



Obrázek 5.19: Rozmístění kolizních bloků na odbočovacím pruhu vozovky.

Prvním auto projede registrační blokem, který zpomaluje AI auta. Při vjezdu do úseku auto projede vstupním blokem, který provádí kontroly. Ve vstupním bloku je nejvíce odlišné zapínání směrových světel. Pro Waypoint systém je obtížné nastavovat směrová světla v jízdním pruhu, ve kterém se jezdí oběma směry. Proto nastavení směrových světel provádí přímo kontroler úseku, který je zapíná pouze při vjezdu ze směru, který má svůj pruh neprůjezdný. Při vjezdu do vstupního bloku se zapne levé směrové světlo a při výjezdu z tohoto bloku se vypne. Obdobně na druhé se při výjezdu zapne pravé směrové světlo.

5.15 Sledování dodržování pravidel silničního provozu

U následujících křižovatek provedení kontroly znamená, že je hráči v případě platné konkrétní podmínky zobrazena zpráva v podčásti uživatelského rozhraní popsáno v 5.10.2.

Křižovatka s hlavní silnicí rovně

Křižovatka má implementovanou kontrolu možnosti vjezdu do křižovatky popsanou v 5.13. Při vjezdu do vstupního bloku se vyhodnotí kontrola na základě směrových světel hráčova auta. Také se uloží očekávaný výjezdní blok z křižovatky.

V simulátoru byla implementována složitá varianta kontroly průjezdu křižovatkou se zónami. Hráčovo auto rezervuje zóny stejným způsobem jako AI auto, aby mu AI auta dala přednost.

Při výjezdu se kontroluje zda hráč vyjel správným výjezdem.



Obrázek 5.20: Úsek s dočasným dopravním omezením řízený semaforem.

Zastavení před křižovatkou

Pokud je před křižovatkou na vedlejší silnici značka *Stůj, dej přednost v jízdě*, tak je zde přítomen i druhý skript. Ten používá stejné kolizní bloky, které byly popsány při popisu reprezentace křižovatky v 5.13. Na obrázku 5.21 jsou vidět 2 kolizní bloky použité pro kontrolu dodržení pravidla daného značkou *Stůj, dej přednost v jízdě*. První blok je registrační a druhý vstupní.

Pro kontrolu se zde používá ukládaná hodnota nejnižší prozatimní rychlosti. Při vjezdu do registračního bloku se nastaví tato hodnota na kladné nekonečno. Při kolizi se stále kontroluje hodnota aktuální rychlosti hráče a pokud klesne pod uloženou hodnotu, tak tuto hodnotu přepíše. Při vjezdu do druhého kolizního bloku se provede kontrola, zda hodnota nejnižší rychlosti za dobu průjezdu registračním blokem je v toleranci. Nekontroluje se vůči nulové hodnotě. Pokud hráč zastaví, zobrazí se mu pochvala.

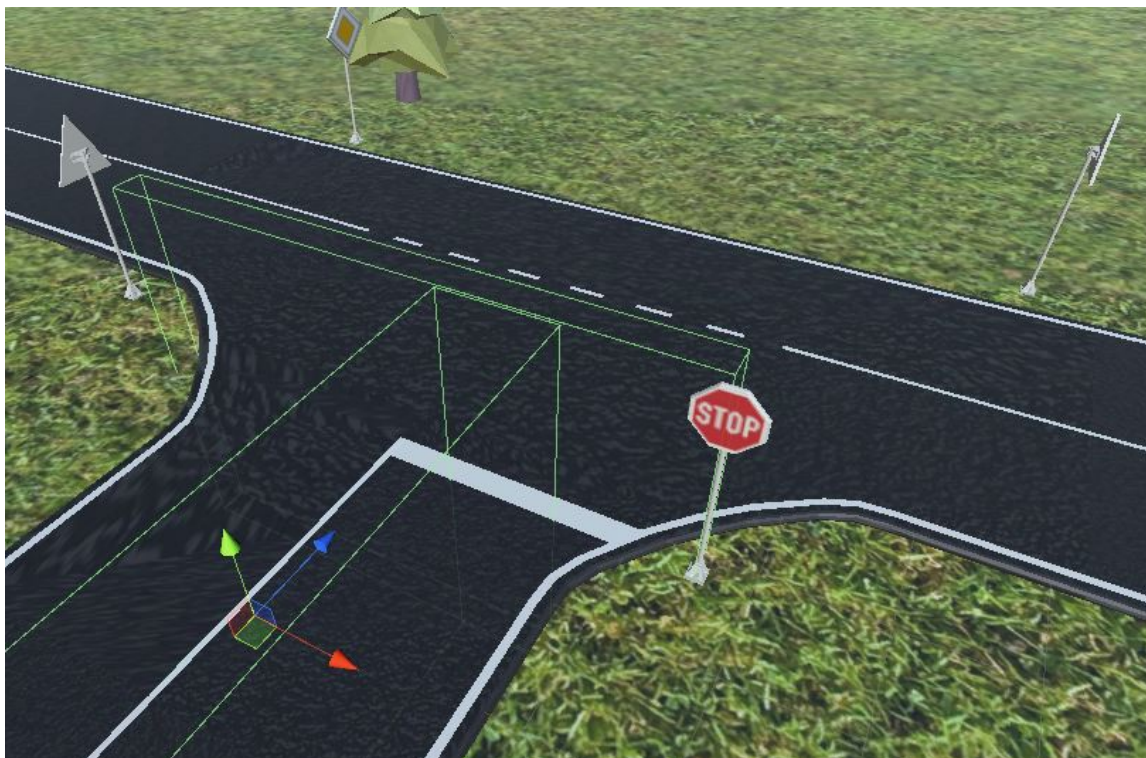
Křižovatka se světelným značením

Rozmístění kolizních bloků a kontrola možnosti vjezdu byla představena v 5.13. U hráče navíc probíhá kontrola vyjetí správným výjezdem, který je určen směrovým světlem a vstupním blokem.

Kruhový objezd

Při vjezdu na křižovatkou se kontroluje prázdnota průjezdního bloku, jak bylo napsáno i v 5.13. Dále se kontroluje, že hráč nemá zapnuta směrová světla.

V průjezdových blocích probíhá kontrola zapnutí pravého směrového světla. Pokud ho hráč zapne, tak se očekává následující výjezd. Tato kontrola má odchytnit případ, kdy by hráč jezdil s pravým směrovým světlem stále dokola na kruhovém objezdu.



Obrázek 5.21: Kolizní bloky použité pro kontrolu dodržení pravidla daného značkou *Stůj, dej přednost v jízdě*.

Při výjezdu je kontrolováno pravé směrové světlo dvakrát tak, aby ho hráč nevypínal moc brzo. Jde o kolize typu *enter* a *exit*.

Odbočovací a připojovací pruh

Kontroly průjezdu odbočovacím nebo připojovacím pruhem probíhají tak, jak jsou popsány v 5.13. Je k tomu využita funkcionality přidružených kontrol samotných zón.

Zákaz zastavení

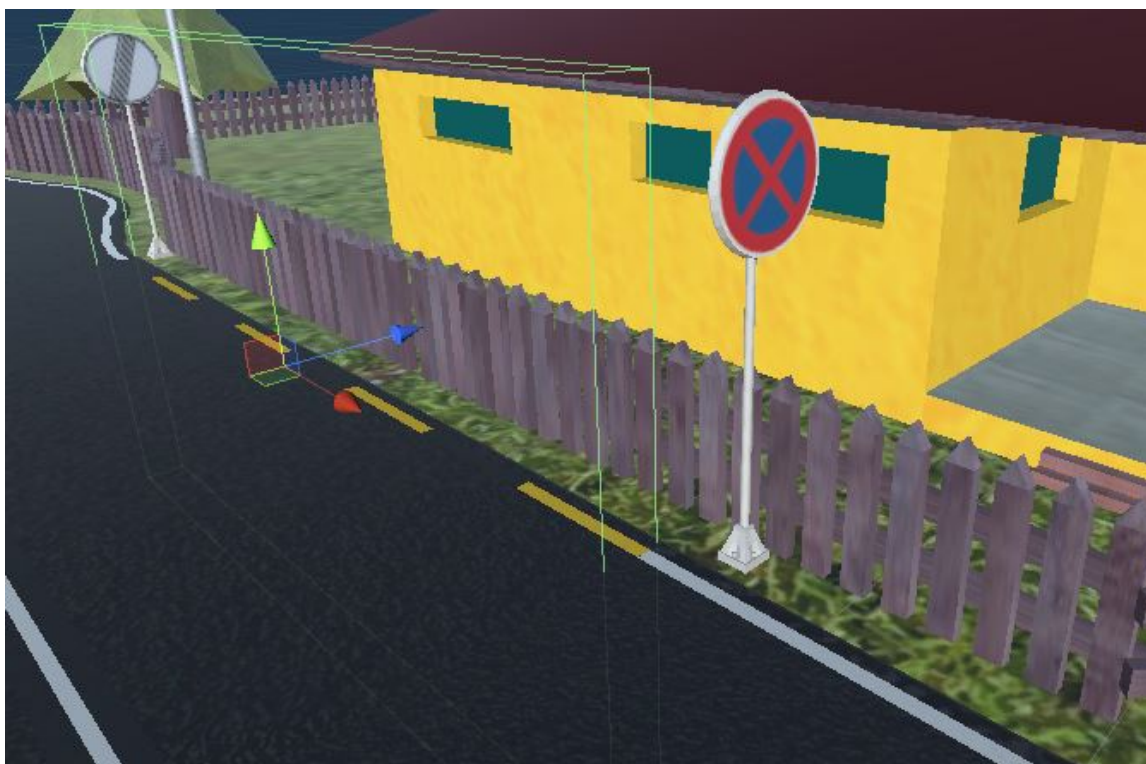
Na obrázku 5.22 je vidět jeden kolizní blok pro kontrolu zákazu zastavení. Na obrázku jsou vidět i značky *Konec všech zákazů*, *Zákaz zastavení* a podélné značení na vozovce. Při rychlosti blížící se nule je hráč upozorněn, že zde nesmí zastavit.

Železniční přejezd

Do hry byl vymodelován i železniční přejezd. Na tomto přejezdu se kontroluje pouze rychlost vozidla.

Dodržení maximální povolené rychlosti

Pro bezpečnou jízdu je nutné dodržovat maximální povolenou rychlost. Z hlediska rychlosti musí být vždy v rovnováze plynulost provozu, která s rychlostí stoupá, a bezpečnost, která



Obrázek 5.22: Kolizní blok pro kontrolu dodržení zákazu zastavení.

s rychlostí klesá. Na mnoha místech můžeme narazit i na značky s doporučenou maximální rychlostí.

Značení ovlivňujících rychlost provozu je mnoho. Mnohé z nich nemají přímo značku, na které je napsané číslo, ale vyplývají z typu a umístění vozovky. Příkladem může být silnice v obci, kde omezení začíná i končí značkami s názvem obce. Silnice pro motorová vozidla a dálnice mají také své značky bez čísla s konkrétní dovolenou maximální rychlostí, ale tato hodnota je tam zvýšena. Řidič musí znát pravidla pro různé typy pozemních komunikací daného státu.

Pro příklad lze uvést tyto:

- nejvyšší dovolená rychlost,
- nejnižší dovolená rychlost,
- obec a konec obce,
- silnice pro motorová vozidla,
- dálnice,
- železniční přejezd,
- zóna s dopravním omezením,
- pěší zóna.

Tuto funkcionalitu jsem v simulátoru implementoval jako oblasti, které mají nastavenou maximální dovolenou hodnotu. Všechny výše vypsané značky a situace nelze v simulátoru najít, i když implementačně je kontrola pro všechny situace stejná. Kontrolu nejnižší dovolené rychlosti jsem neimplementoval vzhledem k menší důležitosti a možným problémům s ostatními účastníky provozu. Maximální hodnota se kontroluje, pokud je vozidlo uvnitř oblasti. V případě zjištění překročení dané hodnoty se zobrazí chyba. V oblasti se chyba zobrazuje v daných časových intervalech s aktuální hráčovou rychlostí dokud hráč nespomalí pod maximální dovolenou rychlostní hodnotu.

Kapitola 6

Testování

Po implementaci funkčního simulátoru se základním menu začalo testování. Zvolenou formou testování bylo uživatelské testování, které je vzhledem k typu práce nejvhodnější. Důvodem je subjektivnost hodnocených faktorů.

Obvyklou úlohou testování je odhalit chyby. Dále měly názory uživatelů pomoci se zhodnocením práce a přinést návrhy pro menší okamžitá vylepšení nebo větší nápady do budoucna.

Samotný program do jisté míry znemožňuje vytvořit testovací scénář s konkrétními úkony, které by respondet postupně prováděl. Důvodem je například nekonečný svět, který je náhodně generován. Není známa křižovatka, kterou hráč projede, a ani směr, ze kterého hráč křižovatku projede. Ostatní účastníci provozu se také chovají do jisté míry náhodně a proto není přesně znám směr jejich jízdy. To znamená, že dvě auta jedoucí stejnou trasou, neprojedou danou trasu stejně. Směr a vybočování může nepřímo ovlivnit i rychlost. Na křižovatkách se vozidla náhodně rozhodují, kterým směrem pojedou. V neposlední řadě vozidla přejíždějí mezi dlaždicemi, jinými slovy nejsou vytvořena a spravována pouze danou dlaždicí, takže ke stejné křižovatce přijíždí pokaždé z různých směrů v různou dobu různá vozidla.

Každý z respondentů provedl následující úkony:

- stáhnul program,
- vyzkoušel program a
- vyplnil online formulář¹.

6.1 Stavba formuláře

Po dokončení testování programu vyplnil každý respondent formulář (dotazník), který měl zachytit jeho názory na různé části hry. Byl vytvořen tak, aby byl hodně strukturovaný. Účel struktury je pomoci respondentovi zaměřit se při odpovídání na konkrétní část programu. To ulehčí práci při analýze odpovědí, protože nejsou pomíchány názory na různé části programu. Formulář obsahuje velké množství otázek, kde respondent odpovídá textovými odpověďmi. Pomocí těchto otázek je možné oznámit chyby nebo popsat návrhy na vylepšení.

Strukturu formuláře jsem rozdělil do několika částí:

¹Formulář pro uživatelské testování byl vytvořen pomocí **Google Forms** a je dostupný na adrese: <https://forms.gle/tnDviMXB1YApuGbe8>

- profil respondenta,
- hlavní menu hry,
- vozidlo a kamery,
- uživatelské rozhraní při simulaci,
- nekonečný svět,
- HW náročnost,
- ostatní účastníci provozu,
- dopravní situace ve hře a
- závěr.

6.2 Výsledky testování

Po dokončení uživatelského testování následovalo jeho vyhodnocení. V této části budou rozebrány nejvýznamnější poznatky získané pomocí testování. Budou popsány případné opravy a vylepšení, která měla vylepšit dojem ze hry. Testování proběhlo z poloviny na lidech z oblasti IT a z poloviny na lidech mimo tuto oblast. Celkem se ho zúčastnilo 13 lidí. Většina respondentů patří do věkového rozmezí 15-25 let a má zkušenost s jiným simulátorem, kde uživatel řídí vozidlo (závodní hra, autoškola atd.). Otázky formuláře s výsledky netextových odpovědí jsou v příloze [D](#).

Hlavní menu hry

Hlavní menu hry bylo implementováno jako poslední. Respondenty bylo zhodnoceno jako jednoduché a přehledné. Nedostatky menu byl rozmazaný text tlačítek a malé množství možností nastavení.

Texty tlačítek a nadpisy v hlavním menu byly nahrazeny jinými s nerozmazaným fontem. Tlačítko *Exit* bylo zvětšeno. Byla zvýšena rychlost posouvání nastavení kolečkem myši. Také byla opravena chyba s neaktuální hodnotou jazyka v nastavení po návratu ze simulace.

Vozidlo a kamery

Velmi dobře byly hodnoceny kamery. Občasnou připomínkou bylo rozhlížení do stran, které by mohlo být rychlejší. Jeden respondent navrhl natáčení kamery pomocí pohybu myši.

Navrženo bylo jiné ovládání světel. Při vypnutí dálkových světel by měla být zapnuta normální světla. Klávesy pro ovládání směrových světel občas nereagovaly. Světla vozidla občas vytvoří přsvětlené bílé plochy, to bylo vyřešeno jejich ztlumením.

Z výsledků testování vyplynulo, že se vozidlo nechová příliš realisticky. Konkrétně bylo uvedeno, že naklánění v zatáčkách by mohlo být menší, a že nereálně působí naražení do jiného objektu. Při volnoběhu při pohybu vzad působily jízdní odpory příliš slabě.

V reakci na tyto problémy byla výrazně zvýšena přítlaková síla, která je implementována pomocí funkce `Rigidbody.AddForce()`². To vyřešilo problémy s nakláněním při

²Třída `Rigidbody` je součástí `Scripting API` herního engine Unity 3D [\[32\]](https://docs.unity3d.com/ScriptReference/Rigidbody.html): <https://docs.unity3d.com/ScriptReference/Rigidbody.html>

brždění i v zatáčkách. zároveň kolize s jinými objekty vypadají reálněji, protože auto se tolik neodrazí. Mělo to pozitivní vliv i na různé nerovnosti nebo jízdu přes retarder. Implementace jízdních odporů byla vylepšena použitím normalizovaného směrového vektoru vozidla místo použití jednotkového vektoru vpřed nebo vzad.

Uživatelské rozhraní při simulaci

Někteří respondenti si myslí, že je zobrazeno moc informací o vozidle. Jinými respondenty byly tyto informace naopak také oceněny. Do budoucna by bylo vhodné je volitelně skrývat.

Stav světel a dálkových světel by mohl být zobrazen jako stav směrových světel.

Nápověda k ovládání je pro uživatele skrytá, proto byla přidána do simulace věta, ve které je uživateli sdělena klávesa zabrazující nápovědu. Byla zvýšena rychlost posouvání nápovědy kolečkem myši.

Nekonečný svět

V této části formuláře šlo o zhodnocení systému dlaždic herního světa. Zhodnocení konkrétních dlaždic herního světa měla na starost další část formuláře popsaná v 6.2. Většinu respondentů zhodnotila volbu implementace nekonečného světa jako správnou.

Generování bylo vylepšeno tak, aby se negenerovaly dvě stejné dlaždice vedle sebe. Svět díky tomu vypadá věrohodněji.

HW náročnost

S menším sekáním se setkali dva respondenti. Nikomu se hra nesekala příliš.

Ostatní účastníci provozu

Respondenti byli spokojeni s chováním ostatních aut na silnici. Naprostá většina respondentů uvedla že vozidla dodržovala dopravní předpisy. Většina lidí by v simulátoru uvítala více ostatních vozidel. Navrženo bylo vytvoření míst s frekventovanějším provozem.

Ostatní vozidla obsahují náhodnostní faktor pro vybočování na trase. Ten způsobuje, že dvě vozidla neprojedou stejnou trasu po stejné trajektorii. Chování ostatních účastníků, má díky němu vypadat věrohodněji. V extrémních případech vybočení na trase mohou zaseknout o okolní texturu. Z toho důvodu formulář obsahoval otázku, která se dotazovala na počet vozidel zaseknutých o texturu. Z otázky vyplynulo, že necelá polovina respondentů potkala ve hře vozidlo zaseknuté o jiný objekt herního světa. Odpovědi z této otázky lze spojit s odpověďmi na otázku, kde je uveden čas strávený testováním.

Dopravní situace ve hře

Podle respondentů je ve hře dostatek dopravních situací. Hodnocení všech dopravních situací dopadlo dobře.

Menší výhrady byly ke kontrole na kruhovém objezdu, která upozorňuje hráče při ježdění dokola se zapnutým pravým směrovým světlem. Při zapnutém pravém směrovém světle se při vjezdu do středového kolizního bloku očekává, že hráč vyjede dalším výjezdem. Při zapnutí na poslední chvíli, téměř v dobu kolize, si uživatelé mysleli, že projeli křižovatku správně. Dále byla opravena chyba v kontrolách u připojovacího a odpojovacího pruhu.

V otázce zabývající se možným dalším rozšířením byly uvedeny následující možnosti:

- přidání chodců,
- přidání jednosměrek,
- přidání parkování,
- rozšíření železničního přejezdu o průjezd vlaku,
- vytvoření situace pro využití výstražných světel.

Závěr

V závěrečné sekci formuláře byly otázky mířeny na celkový dojem ze simulátoru a na zjištění jeho nejslabších a nejsilnějších stránek. Simulátor byl hodnocen kladně.

Konkrétně hra byla hodnocena jako velmi rozmanitá (rozmanitost jednotlivých bloků). Často byl zmiňován princip nekonečného světa (dynamicky generovaný svět). Líbily se kontroly dopravních přestupků a zpětná vazba o přestupcích a chybách.

Mezi negativa hry respondenti zařadili absenci zvuku, hlavní menu nebo kolize vozidla s objekty, které působí nereálně. Podle odpovědí v předchozích částech dotazníku bylo možné hlavní menu a chování vozidla vylepšit a opravit chyby v jejich implementaci.

Kapitola 7

Závěr

Cílem práce bylo vytvořit simulátor autoškoly, který bude kontrolovat dodržování pravidel silničního provozu. Záměr se podařilo splnit ve všech bodech.

Nejdříve jsem studoval pravidla silničního provozu, jejich kategorie a způsob kontroly dodržování. Následoval průzkum existujících simulátorů 3D autoškol. Poté jsem vytvořil seznam pravidel dle priority implementace. Navrhl jsem architekturu simulátoru a uživatelského rozhraní. Pro implementaci jsem zvolil herní engine **Unity 3D** [32] a modely vytvořil v programu **Blender** [5]. Při vytváření fyzikálního modelu jsem se zaměřil zejména na sílu motoru a jízdní odpory. Implementoval jsem pohyb ostatních účastníků provozu. Reprezentace křižovatek a jiných řízených úseků byla vytvořena na základě potřeby kontroly dodržování pravidel silničního provozu uživatelem a potřeby řízení ostatních účastníků provozu tak, aby také dodržovali pravidla silničního provozu. Ostatní účastníci provozu reagují i na jiná auta v herním světě a brzdí, pokud mají některé v cestě. Se zhodnocením dosažených výsledků mi pomohlo uživatelské testování. V následujících odstavcích budou nastíněny některé možnosti rozšíření projektu.

Práce mě naučila mnoho nových věcí. Záměrně jsem si zvolil zadání, které nerozšiřuje existující projekt, a které je rozmanité. Nikdy dříve jsem nedělal v herním enginu **Unity 3D** [32] ani v žádném jiném. Programovacím jazykem herního enginu je **C#**, který jsem znal velmi málo. Mohl jsem použít cizí 3D modely, protože i volně dostupných existuje velké množství. Rozhodl jsem se však veškeré modely sám vytvořit. Vyzkoušel jsem více nástrojů pro modelování a nakonec použil **Blender** [5]. Sám jsem vytvořil i několik textur. Při vytváření fyzikálního modelu vozidla jsem musel nastudovat mně neznámou problematiku.

V práci by bylo možné pokračovat přidáním podpory pro více platforem. Například pro mobilní zařízení by stačilo upravit uživatelské rozhraní přidáním tlačítek pro ovládání uvnitř simulace. Při podpoře mobilních zařízení by se musela ověřit viditelnost hlášených zpráv o dodržování pravidel silničního provozu, protože by mohly být moc malé. Musela by se otestovat i HW náročnost. Přidání dalších modelových dopravních situací je jednoduché díky systému nekonečného světa. K tomu by se případně dotvořily chybějící modely silnic a nové kontrolní skripty a vše by se umístilo do herní dlaždice. Samotný systém herních dlaždic nabízí rozšíření. Bylo by možné přidat možnost skupin více propojených dlaždic. Třeba dvě spojené čtvercové dlaždice nebo tři dlaždice tvaru písmene L by vytvořili více prostoru pro objekty, které by vypadaly na větším prostoru lépe. Příkladem může být železnice nebo dálnice. Skupina dlaždic by v herním světě existovala, pokud by alespoň jedna dlaždice byla uvnitř aktuálního okolí nekonečného světa.

Literatura

- [1] *Zákon č. 361/2000 Sb., o provozu na pozemních komunikacích a o změnách některých zákonů.* In: *Sbírka zákonů.* 2000. ISSN 1211-1244.
- [2] *Vyhláška č. 294/2015 Sb., kterou se provádějí pravidla provozu na pozemních komunikacích.* In: *Sbírka zákonů.* 2015. ISSN 1211-1244.
- [3] AUDI. *Technische Daten Audi A3 2.0 TFSI* [Audi]. March 2007. [Online; navštíveno 30.09.2019]. Dostupné z: https://www.audi.de/dam/nemo/customer-area/more-information/predecessor-models/a3/pdf/a38p/2007_03_a3_ab2_20tfsi.pdf.
- [4] BESIER 3D-EDUTAINMENT WIESBADEN. *3D Fahrschule.* [Online; navštíveno 11.12.2018]. Dostupné z: <http://www.3dfahrschule.de/>.
- [5] BLENDER FOUNDATION. *Blender.* [Online; navštíveno 26.01.2019]. Dostupné z: <https://www.blender.org/>.
- [6] BLIZZARD ENTERTAINMENT. *Heartstone.* [Online; navštíveno 23.09.2019]. Dostupné z: <https://playhearthstone.com/>.
- [7] CARSCOOPS. *2008 Audi A3 Cabriolet – Full Image Gallery.* November 2007. [Online; navštíveno 30.09.2019]. Dostupné z: <https://www.carscoops.com/2007/11/2008-audi-a3-cabriolet-full-image/>.
- [8] D. Černovský a BERGMANNOVÁ, M. *Autoškola.* Vogel(Business Media CZ), 2016. ISBN 978-80-87388-31-0.
- [9] EGGES, A., FOKKER, J. D. a OVERMARS, M. H. *Learning C# by Programming Games.* Springer, 2013. ISBN 978-3-642-36579-9.
- [10] EPIC GAMES. *Fortnite.* [Online; navštíveno 23.09.2019]. Dostupné z: <https://www.epicgames.com/fortnite/>.
- [11] EPIC GAMES. *Unreal Engine.* [Online; navštíveno 26.01.2019]. Dostupné z: <https://www.unrealengine.com/>.
- [12] GOOGLE, INC.. *Google Play.* [Online; navštíveno 26.04.2020]. Dostupné z: <https://play.google.com/store>.
- [13] MICROSOFT. *Microsoft 3D Builder.* [Online; navštíveno 21.05.2020]. Dostupné z: <https://www.microsoft.com/cs-cz/p/3d-builder/9wzdnrcfj3t6>.
- [14] MICROSOFT. *Visual Studio.* [Online; navštíveno 26.01.2019]. Dostupné z: <https://visualstudio.microsoft.com/>.

- [15] MINISTERSTVO DOPRAVY ČR. *Souhrnné statistiky - 2018, 2019* [Centrální registr vozidel]. January 2020. [Online; navštíveno 24.04.2020]. Dostupné z: <https://www.mdcr.cz/Statistiky/Silnicni-doprava/Centralni-registr-vozidel/Souhrnne-statistiky-2018?returl=/Statistiky/Silnicni-doprava/Centralni-registr-vozidel>.
- [16] MINISTERSTVO DOPRAVY ČR. *Statistika II. pol./2019 (k 1.1.2020)* [Centrální registr vozidel]. January 2020. [Online; navštíveno 15.05.2020]. Dostupné z: <https://www.mdcr.cz/Statistiky/Silnicni-doprava/Centralni-registr-vozidel/Statistika?returl=/Statistiky/Silnicni-doprava/Centralni-registr-vozidel>.
- [17] MINISTERSTVO DOPRAVY ČR. *Statistiky přestupků a trestných činů za rok 2019* [Statistiky přestupků a trestných činů]. February 2020. [Online; navštíveno 24.04.2020]. Dostupné z: [https://www.mdcr.cz/getattachment/Statistiky/Silnicni-doprava/Statistiky-k-bodovemu-hodnoceni/Statistiky-prestupku-a-trestnych-cinu/Statistiky-prestupku-a-trestnych-cinu-za-rok-2-\(4\)/Bodovane-prestupky-a-trestne-ciny.pdf.aspx](https://www.mdcr.cz/getattachment/Statistiky/Silnicni-doprava/Statistiky-k-bodovemu-hodnoceni/Statistiky-prestupku-a-trestnych-cinu/Statistiky-prestupku-a-trestnych-cinu-za-rok-2-(4)/Bodovane-prestupky-a-trestne-ciny.pdf.aspx).
- [18] NIANTIC. *Pokemon Go*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://www.pokemongo.com/>.
- [19] NORTON, T. *Learning C# by Developing Games with Unity 3D Begginer's Guide*. Packt Publishing, 2013. ISBN 978-1-84969-658-6.
- [20] NULLAPP. *Driving School 3D*. [Online; navštíveno 26.04.2020]. Dostupné z: <https://www.nullapp.com/details.php?id=40>.
- [21] OKITA, A. *Learning C# Programming with Unity 3D*. CRC Press, 2014. ISBN 978-1466586529.
- [22] OVILEX SOFTWARE. *Driving School 2016*. [Online; navštíveno 11.12.2018]. Dostupné z: <http://www.ovilex.com/app/driving-school-2016/>.
- [23] PSYONIX, INC.. *Rocket League*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://www.rocketleague.com>.
- [24] ROCKSTEADY STUDIOS. *Batman: Arkham City*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://www.warnerbros.com/games-and-apps/batman-arkham-city/>.
- [25] SCAMMELL, R. *Architecting Games in Unity: Common Patterns for Developing Games in Unity [přednáška]* [Casual Connect Asia]. May 2014. [Online; navštíveno 10.12.2018]. Dostupné z: <https://www.youtube.com/watch?v=64u0Vmq5R1k>.
- [26] SCS SOFTWARE. *Euro Truck Simulator 2*. [Online; navštíveno 11.12.2018]. Dostupné z: <https://eurotrucksimulator2.com/>.
- [27] SQUAD. *Kerbal Space Program*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://www.kerbalspaceprogram.com/>.
- [28] TRIMBLE INC.. *SketchUp*. [Online; navštíveno 21.05.2020]. Dostupné z: <https://www.sketchup.com/>.
- [29] UBISOFT. *Assassin's Creed: Identity*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://assassinscreed.ubisoft.com/>.

- [30] UBISOFT. *Might and Magic Heroes VII*. [Online; navštíveno 23.09.2019]. Dostupné z: <https://mmh7.ubi.com/>.
- [31] UNITY TECHNOLOGIES. *Standard Assets (for Unity 2017.3)*. [Online; navštíveno 11.12.2019]. Dostupné z: <https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351>.
- [32] UNITY TECHNOLOGIES. *Unity 3D*. [Online; navštíveno 26.01.2019]. Dostupné z: <https://unity3d.com/>.
- [33] VALVE CORPORATION. *Steam*. [Online; navštíveno 11.12.2018]. Dostupné z: <https://store.steampowered.com/>.
- [34] VLK, F. *Dynamika motorových vozidel*. Nakladatelství a vydavatelství VLK, 2000. ISBN 80-238-5273-6.
- [35] VLK, F. *Lexikon moderní automobilové techniky*. Nakladatelství a vydavatelství VLK, 2005. ISBN 80-239-5416-4.
- [36] ZERBST, S. *3D Game Engine Programming*. Premier Press, 2004. ISBN 1-59200-351-6.

Příloha A

Obsah paměťového média

Paměťové médium přiložené k práci obsahuje kompletní projekt herního enginu Unity, ve kterém jsou veškeré zdrojové kódy, vytvořené 3D modely, použité textury a jiné soubory.

Obsah paměťového média má následující strukturu:

text Zdrojové kódy textu práce v **Latexu**.

project Projekt **Unity** (verze **Unity 2019.3.6f1**).

build_win64 Spustitelný simulátor pro platformu **Windows 10** (architektura **x86_64**).

readme.txt Popis struktury paměťového média.

Příloha B

Návod na ovládání

V tabulce B.1 je vidět výchozí nastavení ovládání simulátoru po spuštění. Pokud je uvedena alternativní klávesa, tak fungují obě klávesy současně.

Tabulka B.1: Výchozí nastavení ovládání simulátoru.

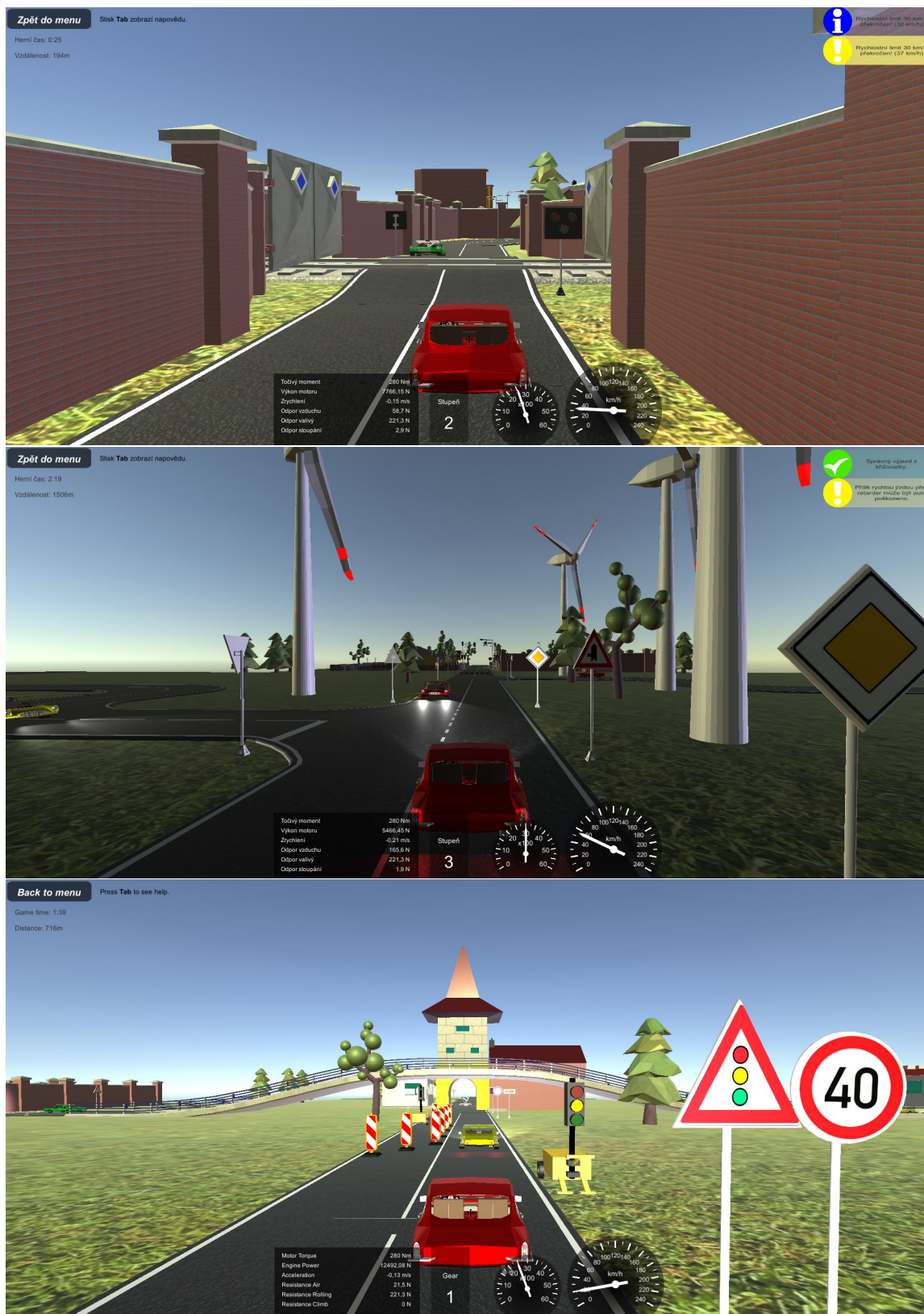
Popis	Klávesa (alternativní klávesa)
Plyn	Šipka nahoru (W)
Brzda	Šipka dolů (S)
Zatáčení doleva	Šipka doleva (A)
Zatáčení doprava	Šipka doprava (D)
Ruční brzda	Mezerník
Levý blinkr	F
Pravý blinkr	G
Výstražná světla	H
Světla	L
Dálková světla	K
Změna kamery	C
Natočení kamery vlevo	Q
Natočení kamery vpravo	E
Návrat na silnici	R
Náповěda k ovládání	Tabulátor
Pozastavení hry	P

Příloha C

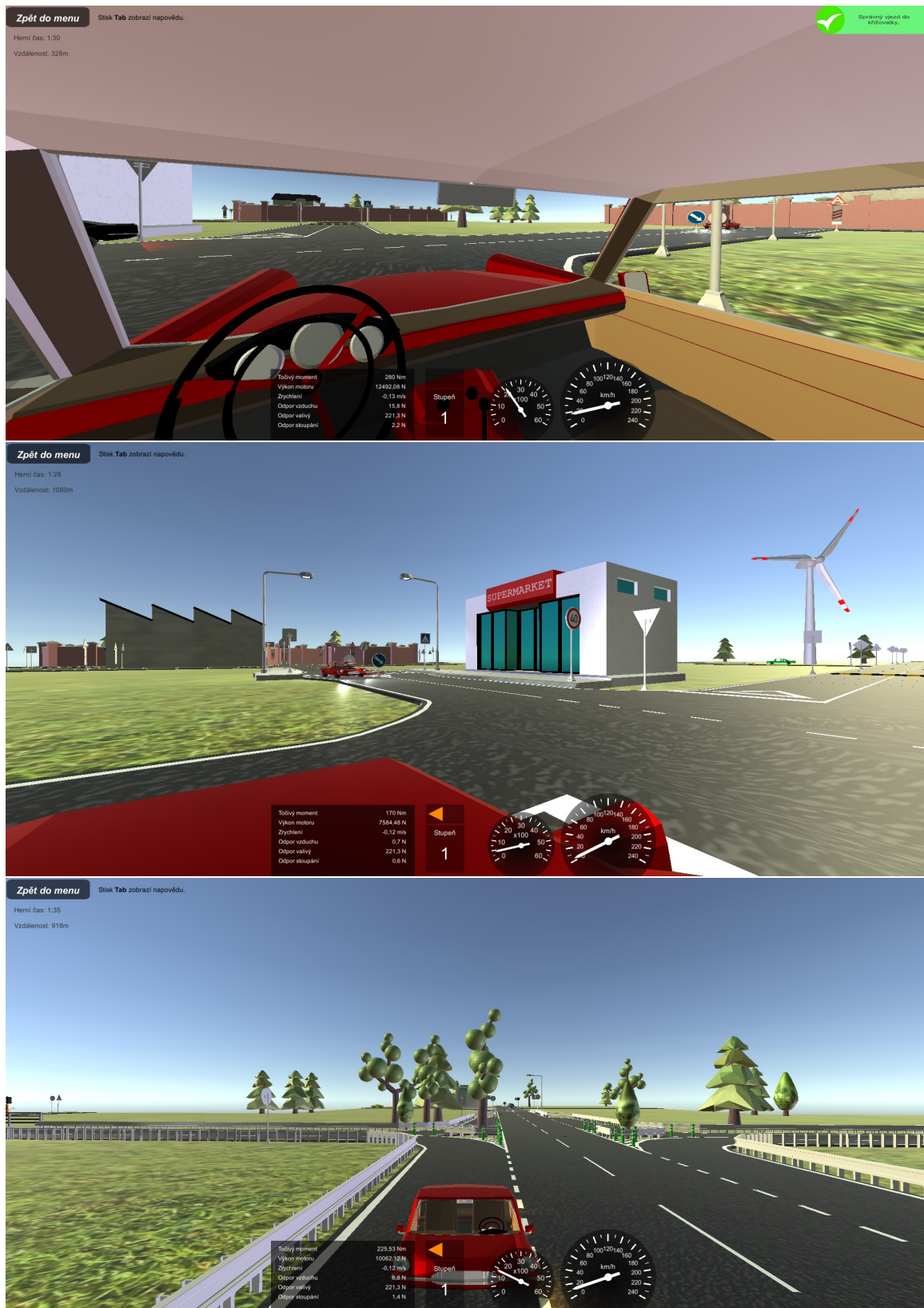
Snímky výsledné podoby práce



Obrázek C.1: Na prvním obrázku je vidět křižovatka se světelnými signály ve dne. Na druhém obrázku je vidět pohled z horní kamery na kruhový objezd v noci.



Obrázek C.2: Na prvním obrázku je vidět překročení rychlosti při jízdě k železničnímu přejezdu. Na druhém obrázku je vidět křižovatka se třemi výjezdy. Na třetím obrázku je vidět dočasné dopravní omezení řízené světelnými signály.



Obrázek C.3: Obrázky ukazují použití různých kamer pro získání přehledu o dopravní situaci.

Příloha D

Formulář pro uživatelské testování

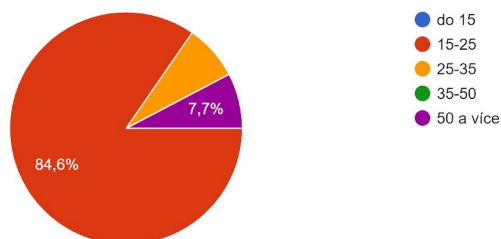
3D Autoškola

Bakalářská práce
FIT VUT v Brně
2020
Petr Kapoun

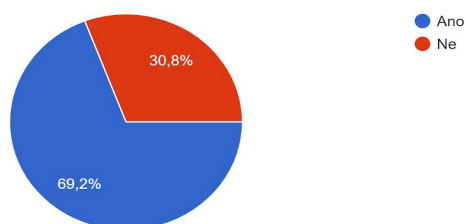
Vytvořil jsem simulátor 3D autoškoly (hru). Simulátor kontroluje dodržování dopravních předpisů.

Profil respondenta

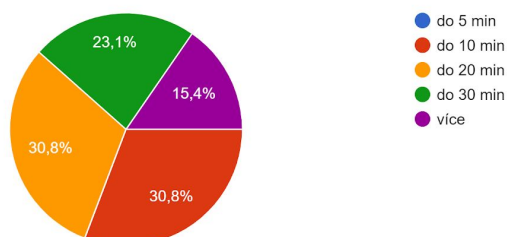
Věk
13 odpovědí



Máte zkušenost s nějakým 3D simulátorem (hrou), kde hráč řídí vozidlo? (závodní hra, autoškola atd.)
13 odpovědí

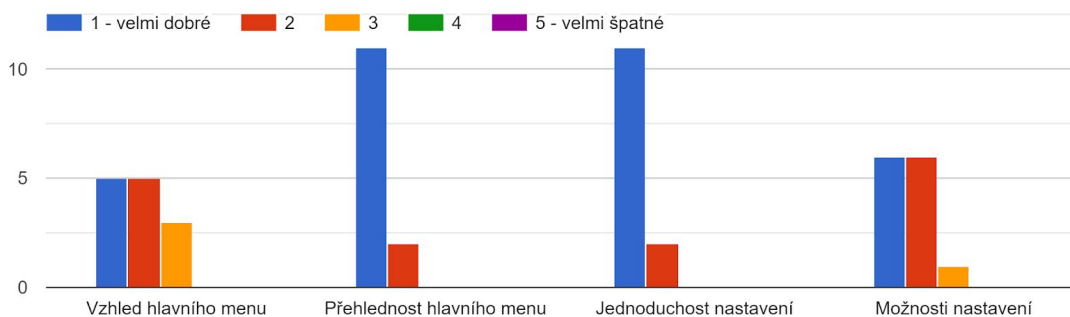


Doba strávená testováním vytvořené práce
13 odpovědí



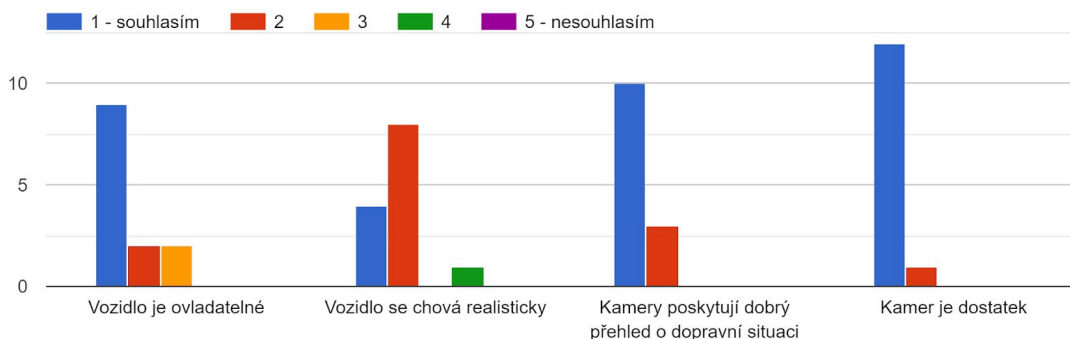
Hlavní menu hry

Po spuštění hry se zobrazí hlavní menu, ve kterém jsou 4 tlačítka (hrát, nastavení, o programu a ukončit).
V nastavení lze změnit jazyk a klávesy pro ovládání vozidla.



Co by šlo v hlavním menu hry vylepšit?

Vozidlo a kamery



Dalo by se něco zlepšit u ovládání vozidla?

Dalo by se něco zlepšit u kamer?

Uživatelské rozhraní při simulaci

V samotné hře jsou 4 typy uživatelského rozhraní:

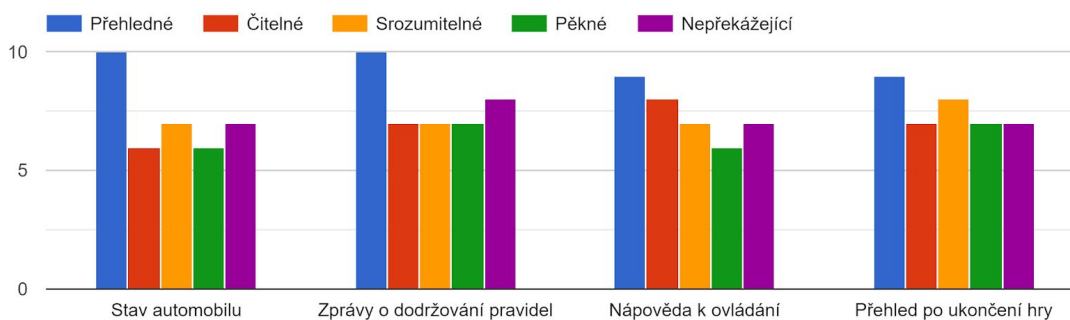
* zprávy o stavu automobilu (rychlost, otáčky, blinkry, ...),

* zprávy o kontrolách dodržování pravidel silničního provozu,

* nápověda k ovládání,

* přehled události po ukončení hry (simulace).

Vlastnosti jednotlivých uživatelských rozhraní v simulaci



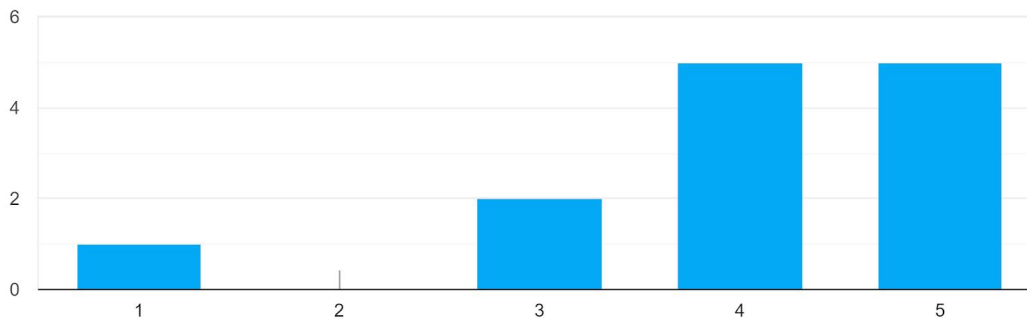
Co lze u uživatelských rozhraní vylepšit? Je v nich nějaká chyba?

Nekonečný svět

V rámci tvorby herního světa byl vytvořen nekonečný herní svět z herních dlaždic. V jednu chvíli existuje jedna středová dlaždice a dlaždice okolní. Při přejezdu hráče na některou z okolních dlaždic se z této dlaždice stane dlaždice středová. Následně se vytvoří dlaždice okolní k této středové a smažou se dlaždice příliš vzdálené, které nespádají do okolí nové středové dlaždice.

Opakem nekonečného světa by byl jeden konkrétní svět. Byla volba nekonečného světa správná?

13 odpovědí

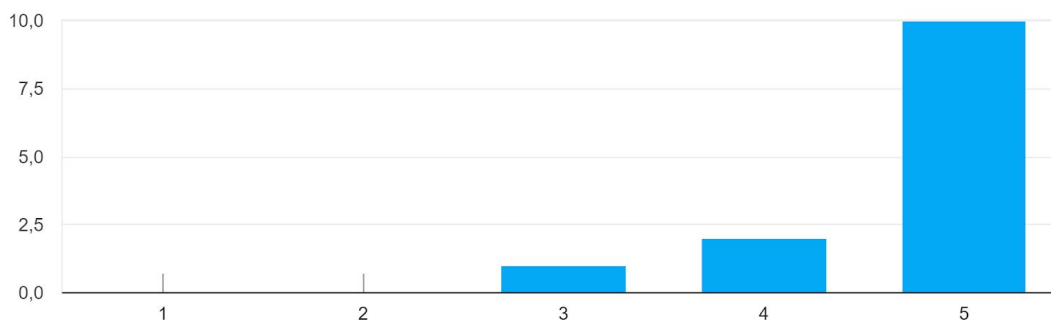


1 - Ne, 5 - Ano

HW náročnost

Zvládalo Vaše zařízení hru?

13 odpovědí



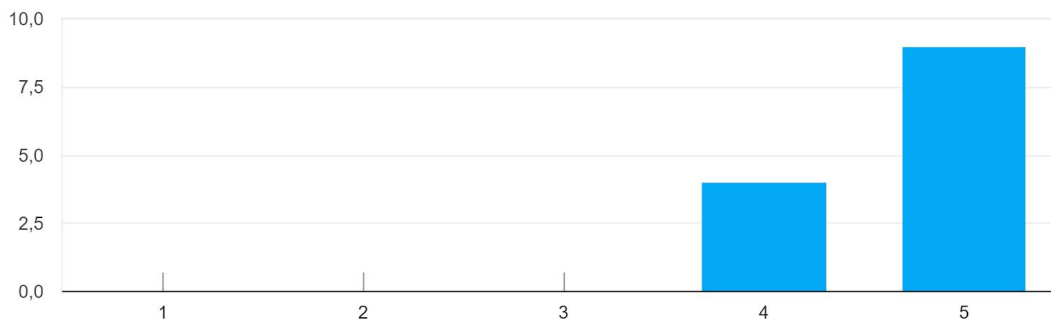
1 - Hra se velmi sekala nebo zamrzla, 5 - Hra běžela plynule

Ostatní účastníci provozu

Ve hře jezdí i jiná auta. Na křižovatkách náhodně vybírají, kterým směrem pojedou a podle toho zapínají blinkry. Na křižovatkách dávají přednost.

Byl jsem spokojený s chováním ostatních aut na silnicích.

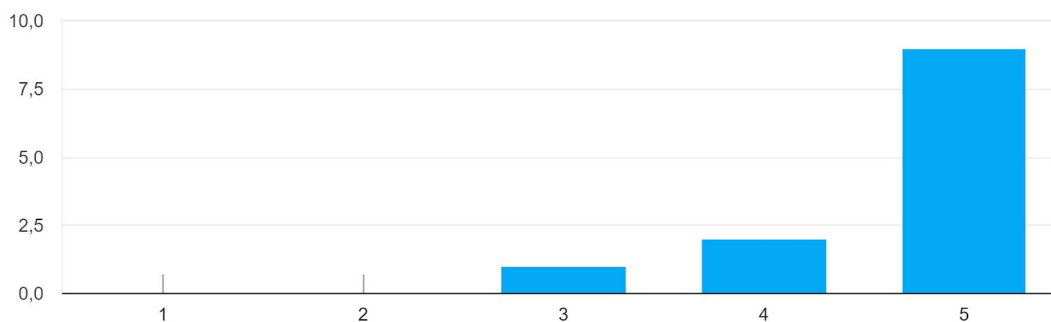
13 odpovědí



1 - Ne, 5 - Ano

Dodržovala ostatní auta dopravní předpisy?

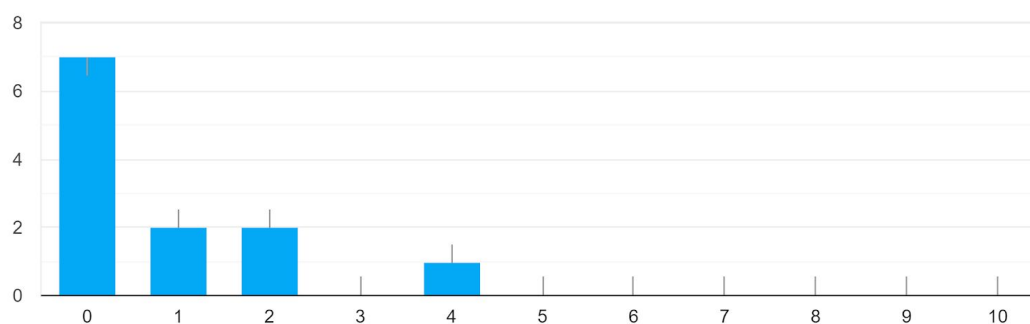
12 odpovědí



1 - Ne, 5 - Ano

Ve hře jsem potkal přibližně tento počet ostatních aut zaseknutých o jiný objekt.

12 odpovědí



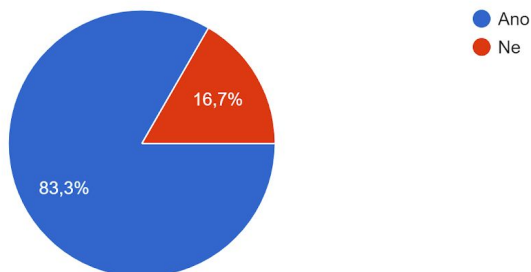
Co by bylo vhodné na ostatních účastnících provozu vylepšit? Případně zde uveďte konkrétní chyby.

Dopravní situace ve hře

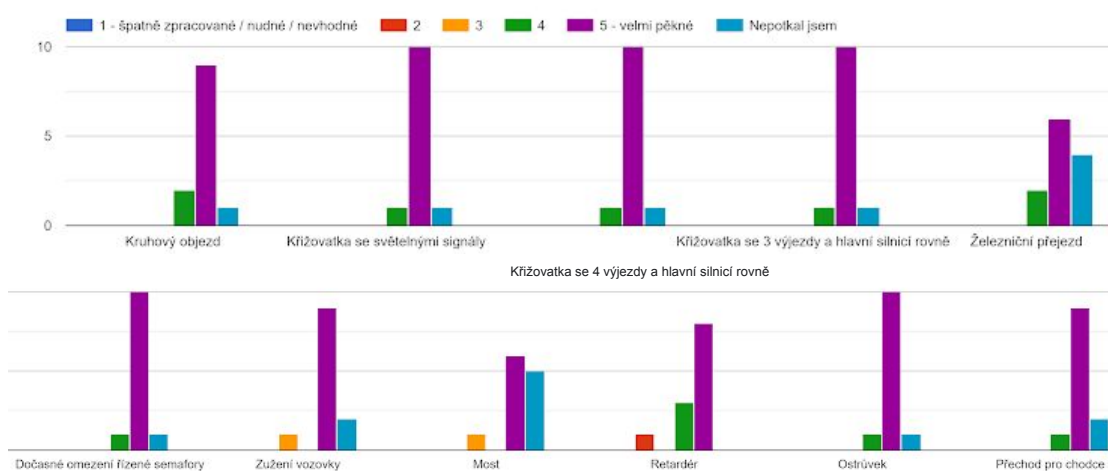
Nejdůležitější částí práce je kontrola dodržování pravidel silničního provozu.

Byl dopravních situací byl dostatek?

12 odpovědí

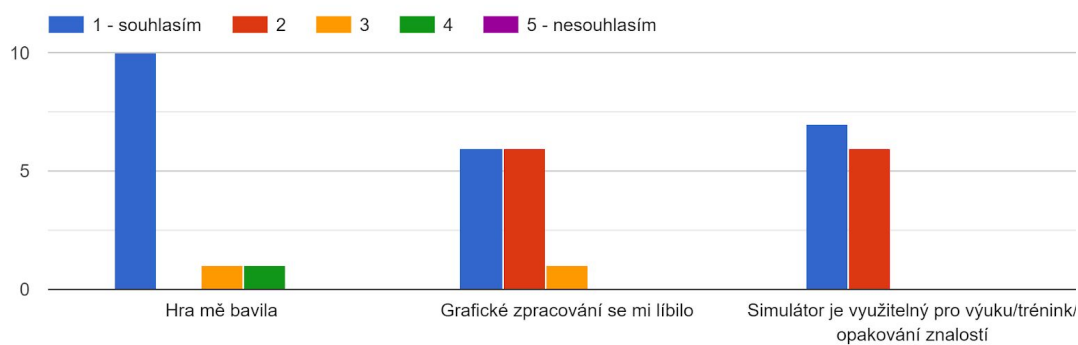


Jak se vám líbily jednotlivé dopravní situace?



Jakou další jednu dopravní situaci by měla autoškola obsahovat?

Závěr



Co se Vám na hře líbilo nejvíce?

Co je největší slabina hry?