ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Tomáš Červenka

VÝVOJ MIKROSIMULAČNÍHO MODELU CHOVÁNÍ VOZIDEL PRO VOZIDLOVÝ SIMULÁTOR

Bakalářská práce

2019

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

K616.....Ústav dopravních prostředků

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Tomáš Červenka

Kód studijního programu a studijní obor studenta:

B 3710 – DOS – Dopravní systémy a technika

Název tématu (česky): **Vývoj mikrosimulačního modelu chování vozidel pro vozidlový simulátor**

Název tématu (anglicky): Development of microsimulation model of vehicle behavior for vehicle simulator

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte osnovou uvedenou v následujících bodech:

- Popište kinematické chování motorových vozidel
- Prostudujte a porovnejte typy mikrosimulačních dopravních modelů
- Prozkoumejte způsoby generování provozu v současných vozidlových simulátorech
- Za využití nejvhodnějších nástrojů vytvořte mikrosimulační model chování pro pohyb vozidel v extravilánu do vozidlového simulátoru
- Vytvořte scénář do vozidlového simulátoru, ve kterém ověříte funkčnost mikrosimulačního modelu

Rozsah grafických prací: dle pokynů vedoucího práce

Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: Vlk, F.: Dynamika motorových vozidel 2. vyd. Brno. 2003. 432 s. ISBN 80-239-0024-2.

PŘIBYL, Pavel a Miroslav SVÍTEK. Inteligentní dopravní systémy. Praha: BEN - technická literatura, 2001. ISBN 80-7300-029-6.

Vedoucí bakalářské práce: **Ing. Adam Orlický**

Ing. Josef Mík, Ph.D.

Datum zadání bakalářské práce: **22. června 2017**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce: **26. srpna 2019**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

doc. Ing. Petr Bouchner, Ph.D.
vedoucí
Ústavu dopravních prostředků

doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.

Tomáš Červenka
jméno a podpis studenta

V Praze dne ..... 3. prosince 2018

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Adamovi Orlickému za odborné vedení a za hodnotné rady a připomínky při psaní této bakalářské práce. Také bych chtěl poděkovat své rodině za podporu, kterou mi při vypracovávání této práce poskytli.

Prohlášení

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

V Děčíně dne 26. 8. 2019 Tomáš Červenka

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Tomáš Červenka

VÝVOJ MIKROSIMULAČNÍHO MODELU CHOVÁNÍ VOZIDEL PRO VOZIDLOVÝ SIMULÁTOR

Bakalářská práce

2019

Abstrakt

Cílem této bakalářské práce je návrh mikrosimulačního modelu chování vozidel pro vozidlové simulátory.

V teoretické části se nejdříve zaměřím na vozidlové simulátory, jaké jsou jejich funkce a význam. Poté si jeden z nich vyberu a popíši, na jakém principu se v něm generuje provoz. Pro vytvoření modelu chování vozidla také musím prozkoumat kinematiku a dynamiku jízdy vozidel. Následně se vrátím zpět k vozidlovým simulátorům. Konkrétně k makrosimulačním a mikrosimulačním modelům provozu v dopravních simulátorech.

V praktické části začnu tvorbou scénáře, ve kterém budu provádět testování svého modelu chování. Poté vytvořím mikrosimulační model chování vozidla, zaměřím se na vytvoření takového modelu, který bude možné použít pro větší množství vozidel. V závěru vytvořím scénář pro ověření funkčnosti výsledného modelu chování.

Klíčová slova

dopravní mikrosimulace, vozový simulátor, model chování vozidla, C# scripty, Unity

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF TRANSPORTATION SCIENCES

Tomáš Červenka

DEVELOPMENT OF MICROSIMULATION MODEL OF VEHICLE BEHAVIOR FOR VEHICLE SIMULATOR

Bachelor's thesis

2019

Abstract

The subject of this bachelor‘s thesis is to develope microsimulation model of vehicle behavior for vehicle simulators.

In the theoretical part I'm firstly going to focus on vehicle simulators what are their functions and purpose. Then I'm going to choose one of them and describe what principle of traffic generating is used in it. I also need to explore the kinematics and dynamics of vehicle motion to create a vehicle behavior model. Then I'm going to return to the vehicle simulators. Especialy to macrosimulation and microsimulation models of traffic simulation.

In the practical section I'm going to begin by creation of a scenario in which I will test my behavior model. Then I'm going to create a microsimulation model of vehicle behavior, I'm going to focus on creating a model that can be used for a larger number of vehicles. Last I'm going create a scenario to verify the functionality of the resulting behavior model.

Keywords

traffic microsimulations, vehicle simulator, vehicle behavior model, C # scripts, Unity

Obsah

Seznam použitých zkratek ... 7

Úvod ... 8

TEORETICKÁ ČÁST ... 10

1 Vozidlové simulátory ... 10

1.1 Hardware ... 11

1.2 Software ... 11

1.3 Simulátor FD K616 ... 13

2 Kinematika a dynamika ... 15

2.1 Základní pohybové rovnice kinematiky ... 15

2.2 Základní vzorce dynamiky ... 16

2.3 Jízdní odpory ... 19

2.4 Hnací síla ... 21

2.5 Průměrné hodnoty zrychlení a brzdění ... 22

2.6 Kovaříkův vzorec ... 23

3 Simulační modely ... 25

3.1 Makrosimulace ... 25

3.2 Mikrosimulace ... 25

3.3 „Car-following“ model ... 26

3.4 Psycho-fyzikální modely ... 27

3.5 Model chování vozidla Nikoly Čemusové ... 28

PRAKTICKÁ ČÁST ... 31

4 Tvorba testovacího scénáře ... 31

5 Vývoj modelu chování vozidla ... 33

5.1 Model jízdy v jízdním pruhu ... 34

5.2 Model chování v závislosti na časovém odstupu ... 34

5.3 Model chování v závislosti na detekci pomocí triggerů ... 36

5.4 Model chování v závislosti na detekci pomocí dynamických triggerů 41

6 Experiment pro ověření funkčnosti 44

Závěr 47

Použité zdroje 49

Seznam obrázků 50

Zdroje obrázků 51

Seznam příloh 52

Seznam použitých zkratek

CGI	Computer-Generated Imagery
PC	Personal Computer
ABS	Anti-lock Brake System
3D	trojrozměrný
C#	C Sharp

Úvod

Žijeme v době, kdy si život bez výpočetní techniky už nedokážeme představit. Počítače jsou nedílnou součástí našeho každodenního života. Provází nás od ranního probuzení až po večerní odpočinek u televizního vysílání, popřípadě u počítačové hry. Výpočetní technika nám také pomáhá při řešení složitých výpočtů, řešení grafických úkolů například CGI efektů (Computer-generated imagery neboli počítačová animace) ve filmech, nebo při tvorbě virtuálních prostředí pro účely učení, zábavy, nebo simulačních testů například pro demolice budov nebo „crash“ testů. V dnešní době si většina lidí pod pojmem počítač představuje osobní počítač (PC) nebo laptop. Tento pojem je však mnohem širší. Počítače řídí činnosti zařízení všude kolem nás. Jsou součástí dopravních prostředků, mobilních telefonů, televizí, přehrávačů, průmyslových strojů, kuchyňských spotřebičů, klimatizací, zámků atd.

Každý z nás používá počítač denně, někteří z nás si to ani neuvědomují. Když ráno vstaneme a otevřeme ledničku, používáme počítač. Kdykoli nám někdo zavolá nebo pošle SMS, používáme počítač. U novějších automobilů počítač ovládá jejich funkci. Počítače jsou všude kolem nás.

Filmy a televizní seriály dnes již hojně využívají počítače při samotném natáčení a následné postprodukci. Ať už jen chceme, aby herečka měla hebčí pleť, nebo potřebujeme odstranit letící letadlo ze středověkého filmu, nebo chceme vytvořit živě vypadající mimozemskou rasu, počítače nám vždy pomohou. V dnešní době je mnohdy důležitější zpracování počítači a přidání CGI efektů než samotné natáčení.

Počítačové (popřípadě konzolové) hry, jsou dnes největším tahounem počítačové grafiky. Každá společnost zabývající se tvorbou her, chce svou novou hrou překonat grafiku ve hrách všech svých rivalů. V tomto vývoji jim však nejvíce bráníme my uživatelé, protože společnost vytvářející hry musí také brát ohled na uživatele a jejich počítače, společnost chce na své hře především vydělat, a proto musí grafické nároky svých her limitovat tak, aby se hra dala spustit i na starších počítacích.

První počítače byly vytvářeny proto, aby lidem usnadnily práci. Dobře naprogramovaný stroj dokážu pracovat s nesrovnatelně vyšší efektivitou než zkušený zaměstnanec. Naneštěstí stejně jako lidé jsou i stroje poruchové. Problém u strojů nastává, když udělá chybu. Stroj nepřemýšlí, pouze opakuje předem nastavený program. Lidé si své chyby všimnou a pokusí se ji opravit. Proto i přes vysokou efektivitu práce strojů, je vždy nutné, aby daný stroj byl pod dohledem zaměstnance.

Počítače v lékařství zjednodušují doktorům složité operace (operace mozku) díky nim mohou pracovat s vyšší přesností. Také mohou vytvářet virtuální modely kostry v případě zlomenin, díky kterým mohou s větší přesností určit nejlepší způsob fixace dané zlomeniny pro její co nejlepší zahojení. Automatizace v tomto oboru zatím není možná, protože každý člověk je jinak stavěný a k jeho vyléčení je zapotřebí jisté dávky improvizace.

Doprava se v současnosti stále rozvíjí. Silniční doprava je díky své dostupnosti nejrozšířenějším druhem dopravy. Ve městech zejména v jejich centrech pak nastávají problémy s kapacitou, ať už se jedná o parkovací místa nebo komunikace samotné. Při takto zvýšené frekvenci dopravy však dochází k dopravním nehodám. Tyto nehody jsou většinou způsobeny nepozorností nebo chybnými rozhodnutími řidičů. Díky počítačům a dopravním simulacím těmto nehodám můžeme předcházet. Využití počítačů v dopravě je široké. Všechny novější automobily mají svůj palubní počítač, který ovládá jednotlivé funkce automobilu od vstřikování paliva do motoru, přes posilovače brzd, popřípadě ABS (Anti-lock Brake System) a podobných systémů až po ovládání autorádia. Ve vyspělejších státech počítače ovládají dopravu ve městech a rozhodují o snížení rychlosti na dálnicích v případě mimořádné události. Počítače je také možné využít při simulacích a testech („crash“ testy). Další využití nalezneme ve výuce, jedná se především o letecké (ale i jiné) simulátory. Může se také jednat o vozidlové simulátory, které neslouží k výuce, ale testují nové automobily nebo nový úsek pozemní komunikace. Tyto simulace mnohdy ukážou nedostatky, které na první pohled nebyly viditelné. Jejich pomocí také lze zkoumat chování řidiče jako jednotlivce, chování dopravního proudu nebo studovat komplexnější problémy jako jsou například vliv dopravy na životní prostřední (vliv emisí) nebo výzkum vlivu hluku na životní prostředí.

Nejdůležitější na těchto simulacích je, aby co nejlépe odpovídaly realitě. Nestačí však pouze vizuální kvalita, důležitým prvek také je reálné chování ostatních účastníků provozu (ostatních vozidel, chodců atd.). Proto v této práci vytvořím co nejpřesnějšího modelu chování vozidla v extravilánu pro vozidlový simulátor. Jelikož se velké množství simulací provádí na dálnicích, rozhodl jsem se vytvořit model chování vozidla právě pro směrově rozdělené komunikace.

TEORETICKÁ ČÁST

1 Vozidlové simulátory

V První řadě se musíme zeptat: Co jsou vozidlové simulátory? Jak fungují? A z čeho se skládají? Jedná se o zařízení, která slouží k co nejpřesnější simulaci řízení reálného vozidla. Jejich cílem je vytvoření virtuálního prostředí, ve kterém se řidič bude cítit jako v reálném provozu. Využívá se při tom nejen grafického vyobrazení scény (virtuální scéna), ale i zvukových efektů (zvuky motoru, jízda pneumatik po vozovce atd.) nebo umístění na pohyblivou plošinu simulující síly, které na řidiče během jízdy působí (akcelerace, decelerace, jízda zatáčkou atd.). Základními dvěma složkami, které takovýto simulátor tvoří je hardware (user interface a výpočetní jednotka) a software (engine, scénář a jeho jednotlivé moduly). [1]

Obrázek 1 Vozidlový simulátor

1.1 Hardware

Slovo hardware označuje veškeré fyzicky existující vybavení počítače. Může být interní nebo externí. Všechny moderní počítače užívají Von Neumannovy architektury. Každý počítač má svou výpočetní jednotku a uživatelské rozhraní neboli user interface (vstupní a výstupní zařízení).

Výpočetní jednotka

Výpočetní jednotkou v tomto případě je počítač samotný (případně více počítačů). Stejně jako všechny moderní počítače má svůj procesor, grafickou kartu, operační paměti, základní desku a pevný disk. Zde je uložen software vozidlového simulátoru.

User interface

Jako ostatní počítače i vozidlový simulátor má k počítači připojené základní vstupní a výstupní zařízení: myš, klávesnice a monitor. Avšak tato zařízení při samotné simulaci nehrají velkou roli.

Vozidlové simulátory se mohou svým uživatelským rozhraním lišit. Může se jednat o křeslo z automobilu (lehký simulátor) nebo o celé vozidlo (plnohodnotný simulátor), které navíc může být doplněno pohyblivou plošinou. Všechny vozidlové simulátory však mají několik prvků společných. Vstupními zařízeními zde vždy jsou volant a pedály (mohou být rozšířeny o řadicí páku). Výstupní zařízení se opět mohou lišit. Může se jednat o jednoduchý monitor postavený na stole, obraz promítaný na plochu (např: stěnu), nebo o monitory připevněné před čelní sklo vozidla. Jedno však mají společné, cílem je vytvořit iluzi skutečné jízdy. Čím lepší provedení výstupních zařízení, tím lepší iluze skutečné jízdy a zároveň i přesnější výsledky experimentů. [1]

1.2 Software

Software obecně označuje programové vybavení počítače. Lze dělit na systémový software, který zajišťuje chod počítače a jeho styk s okolím, a na aplikační software, se kterým buď přímo pracuje uživatel, nebo slouží k řízení nějakého přístroje. V případě vozidlových simulátorů záleží především na kvalitě aplikačního softwaru.

Engine

Je hlavním softwarem skládajícím se z fyzikálního modulu, grafického modulu, zvukového modulu, modulu chování účastníků dopravy a propojení k uživatelskému rozhraní vozidla popřípadě dalších. Jednotlivé moduly mohou a nemusí běžet na svém počítači.

Scénář

Jedná se o kombinaci specifického 3D modelu a naplánovaného chování řidiče. Například se může jednat o jízdu nepřehledným úsekem pozemní komunikace, nebo po dálnici, na které vzniká dopravní kolona. 3D model se skládá z jednotlivých modelů, které mohou mít přiřazenou animaci. Cílem scénářů je sledování chování řidiče v těchto situacích.

Animace ve scénáři se skládají z jednotlivých souborů, které popisují chování animace (čas, poloha, natočení v prostoru), pohyb mezi těmito soubory se určuje interpolací (dopočítává). Dále se jedná o soubory „trigger“ (spoušť), tyto soubory spustí určitou animaci na základě její aktivace (triggeru). Například: Vozidlo se blíží k přejezdu, vjede do oblasti, ve které se trigger nachází, čímž spustí animaci závor. Nakonec se jedná o animace jako takové (mohou být řízeny triggerem), například animace provozu. Scénář pro své fungování využívá jednotlivých modulů.

Vizualizační (grafický) modul

Slouží k zobrazování předem připravené 3D grafiky dle informací, které získává od fyzikálního modulu. Jedná se o aktuální polohu vozidla v simulátoru (poloha viewportu ve virtuální realitě).

Fyzikální modul

Sám se skládá ze dvou modulů. Prvním je modul vozidla samotného, který popisuje jeho chování, funkci vstupních zařízení (volant a pedály vozidla v reálném světě atd.). Druhým je modul okolního provozu (resp. animací). Data z fyzikálního modulu se dají ukládat po předem určených intervalech (řádově v milisekundách). Tato data se poté vyhodnocují. [1]

Modul chování účastníků dopravy

Může být součástí fyzikálního modulu. Vozidlový simulátor potřebuje mít vygenerovaný provoz, který zajistí, aby se řidič cítil jako na skutečné komunikaci. Nejedná se však pouze o silniční vozidla, součástí tohoto modulu mohou být také chodci, případně i jiné dopravní prostředky jako jsou např. tramvaje, vlaky nebo dokonce lodě. [1]

1.3 Simulátor FD K616

Tento simulátor využívá vlastní, proprietární engine, který společně vytvořili doc. Bouchner a doc. Novotný. Jeho základ vznikal již v letech 2003-2005, jeho vývoj se však do dnešního dne nezastavil. Některé jeho části (např: fyzikální modul) využívají převzatých součástí, ale jako celek je tento engine jedinečný. Na jednu stranu to poskytuje svobodu, ale aktuálně je relativně zastaralý, proto se jeho tvůrci pokouší experimentovat s jinými enginy především pak s Unity.

Obrázek 2 Simulátor FD 616

Animace provozu jsou v tomto simulátoru stejně jako ostatní animace matice časů a souřadnic. Kromě toho v každém výpočetním kroku simulátoru zjišťují kolize mezi sebou a také s řidičem a umí se zastavit (avšak ne moc plynule ani realisticky). Tyto matice se vytváří tak, že je někdo z týmu, který tento simulátor vytváří „najede“, jede stejně jako zamýšlené vozidlo v simulátoru. Data jeho jízdy se vyexportují, převedou se na křivku jízdy a pomocí scriptu v programu matlab se z těchto dat mohou generovat trajektorie více aut (na stejné křivce, v jiných časech). Animace vozidel v tomto simulátoru nemají svoji fyziku (rychlost, zrychlení atd.), mají jen časy a souřadnice, na kterých se v těchto časech mají nacházet. Umí se zastavit v případě, že detekují nebezpečí kolize. Nevýhodou tohoto modelu je, že simulovaná vozidla nejsou schopna v případě potřeby předjíždět. Proto se ve své práci zaměřím hlavně na schopnost předjíždět v případě, že vozidlo není ohroženo tento manévr vykonat. [2]

2 Kinematika a dynamika

Pro vytvoření mikrosimulačního modelu chování vozidla potřebujeme také pochopit funkci vozidla samotného. Především pak musíme pochopit způsoby, jakými se může pohybovat a síly, které na něj během tohoto pohybu působí. Ve zkratce musíme pochopit kinematiku a dynamiku jeho jízdy.

2.1 Základní pohybové rovnice kinematiky

Existuje několik základních druhů pohybu, kterými se tělesa mohou pohybovat v prostoru. Mezi tyto základní druhy patří:

Pohyby Translační

Pohyb rovnoměrný

Při tomto pohybu ujede automobil za každou jednotku času stejnou vzdálenost. Mezi dráhou *(s)* rychlostí *(v)* a časem *(t)* platí následné vztahy:

$$s = v * t \qquad v = s/t \qquad t = s/v$$

Pohyb zrychlený

Při tomto pohybu se rychlost automobilu s narůstajícím časem zvyšuje. To nazýváme zrychlení nebo akcelerace *(a)*. Pokud vozidlo z nulové *(v_0)*neboli počáteční rychlosti zrychluje konstantním zrychlením, jeho rychlost po určitém čase bude možné vypočíst ze vztahů:

$$v = a * t = 2s/t \qquad v = v_o + a * t = v_o + 2s/t$$

Pokud neznáme čas, ale známe dráhu můžeme použít úpravu vztahu změny kinetické energie *(E_k)* a vykonané práce *(W_s)*:

$$E_k = W_s \Longrightarrow \frac{1}{2}m * v^2 - \frac{1}{2}m * v_0^2 = m * a * s \qquad v = \sqrt{v_0^2 + 2 * a * s}$$

Pro výpočet času potřebného pro změnu rychlosti při konstantním zrychlení můžeme použít jeden z následujících vztahů:

$$t = \frac{v}{a} = \sqrt{\frac{2s}{a}} = \frac{2s}{v} \qquad t = \frac{v - v_0}{a} = \frac{2s}{v + v_0}$$

Pokud potřebujeme vypočítat zrychlení, kterého dosahujeme v závislosti na čase nebo dráze, použijeme jeden z těchto vztahů:

$$a = \frac{v}{t} = \frac{v^2}{2s} = \frac{2s}{t^2} \qquad a = \frac{v - v_0}{t} = \frac{v^2 - v_0^2}{2s}$$

Potřebujeme-li vypočítat ujetou dráhou za konstantního zrychlení, je možné použít jeden z následujících vztahů:

$$s = \frac{a * t^2}{2} = \frac{v^2}{2a} = \frac{v * t}{2} \qquad s = v_0 * t + \frac{a * t^2}{2} = \frac{v^2 - v_0^2}{2a} = \frac{v + v_0}{2} * t$$

Pohyb rotační

Při rotačním pohybu rovnoměrném se vozidlo každou sekundu otočí o stejný úhel. Mezi úhlem pootočení *(φ)*, úhlovou rychlostí rotace *(ω)* a časem platí následující vztahy.

$$\varphi = \omega * t \qquad \omega = \varphi / t \qquad t = \varphi / \omega$$

Rychlost otáčení se většinou udává v jednotkách radiány za sekundu (rad/s). 360° je 2π radiánů což je zhruba 6,28 rad.

Při rotačním pohybu rovnoměrně zrychleném bereme v úvahu ještě úhlové zrychlení (ε), které obdobně jako zrychlení translačního pohybu zrychleného má jednotku (rad/s^2). Zároveň zde platí obdobné vztahy jako u translačního pohybu rovnoměrně zrychleného. [10]

2.2 Základní vzorce dynamiky

V reálném světě se za každým pohybem skrývá nějaká síla (popřípadě moment síly), tyto síly mohou na vozidlo působit z okolí (např: tíhová síla), nebo mohou mít svůj původ přímo v daném vozidle (např: tažná síla).

Síla

Je veličina, která má svou velikost, směr a orientaci, jedná se tedy o vektor. Velikost síly je obecně dána vztahem mezi silou *(F)*, hmotností tělesa *(m)* a zrychlením *(a)*:

$$F = m * a$$

Moment síly

Pokud síla působí na rameni vzniká moment síly. Moment působící po určitou dobu vyvolá rotační pohyb. Čím je síla *(F)* větší nebo rameno *(n)* na, kterém síla působí větší, tím je větší moment síly *(M)*. Platí následující vztah:

$$M = F * n$$

Tíhová síla

Je to základní síla působící na každé těleso. Vzniká působením gravitační (tíhové) síly země na vozidlo. Zrychlení je zde zastoupeno gravitačním zrychlením *(g)*, které činí cca $9{,}81\ m/s^2$. Je dána vztahem:

$$F_G = m * g$$

Síla setrvačná

Podle Newtonova zákonu síla, která se snaží udržet tělo v konstantním stavu, tzn. buď v klidu, nebo v rovnoměrném pohybu. Setrvačná síla *(F_S)* je stejně velká jako síly zrychlující (zpomalující), má stejný směr, ale opačný smysl (opačnou orientaci).

$$F_S = m * a$$

Odstředivá síla

Je výsledkem setrvačné síly, která se vozidlo snaží udržet v přímočarém pohybu. Odstředivá síla *(F_O)* vyvolává moment působící v příčném směru, v hraničním případně může dojít k přerušení kontaktu mezi vnitřními koly a vozovkou, případně k převracení vozidla. Její velikost kromě hmotnosti vozidla a rychlosti vozidla závisí také na poloměru zatáčení, dá se vyjádřit vztahem:

$$F_O = m * \frac{v^2}{R}$$

Adhezní síla

Je síla přilnavosti *(F_T)* pneumatik k povrchu vozovky. Součinitel adheze *(μ)* neboli přilnavost je schopnost materiálu přilnout k jinému materiálu. Odborněji je to schopnost přenosu tečných sil ve styku dvou povrchů bez zřetelného pohybu. Lze vyjádřit vztahem:

$$F_T = F_G * \mu$$

Součinitel adheze závisí především na těchto faktorech: jakost a stav povrchu vozovky, jakost a stav povrchu pneumatiky, rychlost jízdy, velikost prokluzu kola, výskyt vody a jiných materiálů na vozovce.

Součinitel adheze na různých druzích povrchu:

Povrch vozovky		M	Povrch vozovky		μ
beton	Suchý	0,8 - 1,0	polní cesta	Suchá	0,4 – 0,6
	Mokrý	0,5 - 0,8		Mokrá	0,3 - 0,4
asfalt	Suchý	0,6 - 0,9	Tráva	Suchá	0,4 - 0,6
	Mokrý	0,3 - 0,8		Mokrá	0,2 - 0,5
dlažba	Suchá	0,6 - 0,9	písek, sníh		0,2 - 0,4
	Mokrá	0,3 - 0,5	Náledí	0°C	0,05 - 0,10
makadam	Suchý	0,6 - 0,8		-10°C	0,08 - 0,15
	Mokrý	0,3 - 0,5		-20°C	0,15 - 0,20

Třecí síla

Třecí síla *(F_T)* vzniká při vzájemném posuvu dvou těles. Součinitel smykového tření *(f)* je fyzikální veličina vyjádřená poměrem třecí síly a kolmá tlakové síly (v našem případě tíhové síly). Závisí především na povrchu a drsnosti pneumatiky a vozovky. Smykové tření vzniká mezi tělesy při jejich posuvném pohybu. [3]

$$F_T = F_G * f$$

2.3 Jízdní odpory

Při pohybu vozidel vznikají jízdní odpory, které působí proti směru pohybu vozidla a snaží se ho přibrzdit popřípadě zastavit.

Valivý odpor

Vzniká deformací pneumatiky a vozovky. Počítáme-li s vozovkou jako dokonale tuhým tělesem, pak se bude deformovat pouze pneumatika. K této deformaci dochází na obvodu pneumatiky, ve směru valení. Valivý odpor lze vyjádřit jako vztah mezi radiální reakcí vozovky *(Z_K)* a součinitelem valivého odporu *(f_K)*:

$$O_{fK} = Z_K * f_K$$

Aerodynamický odpor

Při jízdě vozidla dochází k proudění vzduchu kolem jeho karoserie. Větší část proudí kolem vrchní částí karoserie, zbylá část se musí protlačit mezi podvozkem a povrchem vozovky. Proudění vzduchu se za vozidlem neuzavírá, ale přechází ve víření. Celkový aerodynamický odpor můžeme vyjádřit jako vztah mezi čelní plochou vozidla *(S_x)*, součinitelem aerodynamického odporu *(c_x)*, hustoty vzduchu *(ρ)*, rychlosti vozidla *(v)* a rychlosti vzduchu *(v_v)*:

$$O_V = \frac{1}{2} S_x * c_x * \rho * (v + v_v)$$

Odpor zrychlení

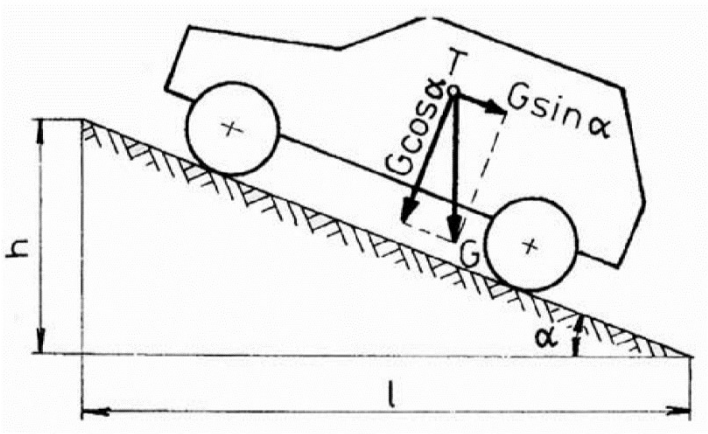
Tento odpor vzniká, když se vozidlo snaží překonat setrvačné síly vozidla. Jedná se především o setrvačnost vozidla jako takového a setrvačnosti otáčejících se částí uvnitř vozidla. Odpor zrychlení lze vyjádřit vztahem mezi mementem setrvačnosti rotujících částí motoru *(J_m)*, celkovým převodem mezi motorem a hnacími koly *(i_c)*, momentem setrvačnosti rotujících částí převodovky *(J_p)*, převodem rozvodovky *(i_r)*, momentem setrvačnosti kol *($\sum J_{Ki}$)*, mechanickou účinností *(η)*, hmotností *(m)*, zrychlením *(a)* a poloměrem kol *(r_d)*l:

$$O_Z = \left[1 + \frac{\left(J_m * i_c^2 + J_p * i_r^2\right) * \eta + \sum_i J_{Ki}}{m * r_d^2}\right] m * a$$

Odpor stoupání

Pokud je vozidlo nuceno překonávat převýšení, působí na něj také odpor stoupání, v tomto případě má tíhová síla *(F_G)* i složku rovnoběžnou s vozovkou, která s rovinou svírá úhel α. Výsledný odpor stoupání je dán vztahem:

$$O_S = F_G * sin\alpha$$



Obrázek 3 Odpor stoupání

Celkový jízdní odpor

Celkový jízdný odpor získáme prostým součtem všech jízdních odporů, tento celkový jízdní odpor vyjadřuje potřebnou hnací sílu *(F_H)*, kterou musíme vynaložit, abychom vozidlo uvedli do pohybu, případně ho v pohybu udrželi: [3]

$$O_C = O_{fK} + O_V + O_S + O_Z = F_H$$

2.4 Hnací síla

Proti všem těmto jízdním odporům působí tzv hnací síla. Tato síla se obecně stanovuje ze zjednodušeného strukturního modelu pohonu vozidla. Zdrojem výkonu silničních vozidel je ve většině případů spalovací motor, jehož výkon je definován krouticím momentem a otáčkami hřídele. Tyto dvě veličiny jsou na nápravy silničních vozidel přenášeny převodovkou. Při tomto převodu sil dochází ke ztrátám výkonu, počítá se s tzv. účinností převodovky, která nikdy není stoprocentní. [4]

Hnací moment je dán vztahem:

$$M_n = M_m * i_p * \eta_p$$

M_n [Nm] = hnací moment na obvodu náprav

M_m [Nm] = kroutící moment motoru

i_p [-] = převodový poměr převodovky

η_p [-] = účinnost převodovky

Hnací síla je dána vztahem:

$$F_{Hi} = \frac{M_{ni}}{ri}$$

F_{Hi} [N] = hnací síla na jednotlivých nápravách

M_{ni} [Nm] = hnací moment na jednotlivých nápravách

ri [m] = dynamický poloměr jednotlivých náprav

Celková hnací síla je poté dána jako suma hnacích sil na jednotlivých nápravách

$F_H = \sum_i F_{Hi}$ [N]

2.5 Průměrné hodnoty zrychlení a brzdění

Pro tvorbu simulátoru je dobré výše zmíněné síly znát, není však nutné s nimi při vytváření počítat. Při tvorbě scénáře je důležitá především kinematika vozidla, tedy popis jednotlivých pohybů vozidla. Díky hnací síle na vozidlo působí podélné zrychlení, s tímto zrychlením budu v simulátoru operovat. I přestože u spalovacích motorů po určité dosažené rychlosti začne vozidlo ztrácet výkon, tudíž není schopné pokračovat se stejným zrychlením, budeme pro zjednodušení modelu počítat s konstantním zrychlením.

Na vozidla v silniční dopravě však působí více sil než jen těch fyzikálních, které jsou pro vytvoření modelu chování důležitější než ty fyzikální. Jednou z těchto sil je síla legislativy. I když jsme schopni vyrobit automobily, které dosáhnou rychlosti i přes 300 km/h, zákony a předpisy nám umožní na pozemních komunikacích dosáhnout znatelně nižších rychlostí. Nejdůležitějším omezením se tedy stává dopravní značení. V obcích v ČR můžeme jet rychlostí maximálně 50 km/h, mimo obce 90 km/h a na dálnicích až 130 km/h. Některá vozidla však nemají povoleno se na dálnicích pohybovat. Pokud vozidlo není schopné vyvinout rychlost alespoň 80 Km/h, musí dálnici na nejbližším výjezdu opustit. Další silou, která na vozidlo působí, je řidič samotný. Řidič je stále jen člověk, proto na něj mohou působit různé vlivy okolí. Tyto vlivy pak mohou zhoršovat jeho schopnost soustředit se na řízení, čímž se zvyšuje jeho reakční doba. Tyto rušivé elementy mohou být například hlasité rádio, šero nebo tma, únava nebo dokonce otravný spolujezdec. [8]

Zrychlení

Pro zrychlení při pohybu po křivce (nenarušeném jinými vozidly) budu využívat komfortního zrychlení z intervalu 0-3 m/s^2. Pro mé potřeby zvolím hodnotu 1,5 m/s^2. Tuto hodnotu bude v případě potřeby možné změnit. Například v případě, kdy pro bezpečný předjížděcí manévr bude zapotřebí vyššího zrychlení. V mém případě však postačí zrychlení komfortní. [7]

Brzdění

U vozidel jde u pohybu zpomaleného zejména o brzdění. Může se také jednat o dojezd do zastavení nebo o náraz. Jedná se o pohyb analogický s pohybem zrychleným. Rozdílem je zde, že směr zrychlení působí proti směru pohybu. Ve výpočtech se tedy vyskytuje s opačným znaménkem. Brzdné zpomalení (decelerace) je vyvolána adhezní silou působící mezi vozovkou a pneumatikami. Tato síla je závislá na několika faktorech především na povrchu vozovky a pneumatiky, výskytu vody a jiných materiálů na vozovce atd. Na suché rovné vozovce musí vozidla splňovat předepsané minimální zpomalení:

Osobní automobil: $5{,}8\ m/s^2$

Autobusy: $5{,}0\ m/s^2$

Ostatní automobily: $5{,}0\ m/s^2$

Skutečné zpomalení však bývá vyšší, na ideálním povrchu s dobrými pneumatikami a rovnoměrným zatížením náprav může dosahovat $9\ m/s^2$ a více. [3]

2.6 Kovaříkův vzorec

V této práci vytvořím model chování vozidla, který bude umět na směrově rozdělených komunikacích sledovat křivku (udržet se na komunikaci), přibrzdit a v případě potřeby zastavit před překážkou, nebo se překážce vyhnout. Pro tento výhybný manévr využiji tzv. Kovaříkův vzorec. Kovaříkův vzorec je způsob výpočtu nejkratšího příčného přemístění mezi dvěma jízdními pruhy použitím jednoho nebo dvou oblouků. Tento vzorec má různé formy v závislosti na druhu vozidla (především záleží na rychlosti, jakou vozidlo dokáže vyvinout, a také na tom, zda se jedná a vozidlo jednostopé či dvoustopé).

V zásadě se Kovaříkův vzorec bez přechodnic používá při výpočtu u pomalu jedoucích vozidel a s přechodnicemi při výpočtu u rychle jedoucích vozidel.

Kovaříkův vzorec bez přechodnic:

1 oblouk

$$t_y \geq 1{,}41\sqrt{\frac{y}{a_y}}$$

2 oblouky

$$t_y \geq 1{,}57\sqrt{\frac{y}{a_y}}$$

Kovaříkův vzorec s přechodnicemi:

1 oblouk

$$t_y \geq 2\sqrt{\frac{y}{a_y}}$$

2 oblouky

$$t_y \geq 3{,}13\sqrt{\frac{y}{a_y}}$$

t_y [s] = doba příčného přemístění

y [m] = příčné přemístění

a_y [m/s2] = příčné zrychlení

Příčné přemístění jedním obloukem se používá při snaze odvrátit střet s překážkou nezávisle na tom, kam bude pokračovat pohyb vozidla.

Příčné přemístění dvěma oblouky se také používá při vyhýbání se překážce, avšak v tomto případě druhý oblouk slouží k navrácení se do původního směru pohybu (rovnoběžného s původním směrem). Proto pro naše potřeby budeme využívat Kovaříkův vzorec s přechodnicemi s 2 oblouky. [5]

3 Simulační modely

Další položkou, kterou pro vytvoření mikrosimulačního modelu chování vozidla nezbytně musíme, znát jsou simulační modely, které jsou v současné době používány. Ty se v základu dělí na dvě skupiny makrosimulační a mikrosimulační. Základní rozdíl mezi těmito dvěma skupinami je měřítko, v jakém problémy zkoumají. Zatímco mikrosimulace se zaměřují na vozidla jako jednotlivce, makrosimulace se zabývají chováním dopravních proudů.

3.1 Makrosimulace

Makrosimulace zkoumají pohyb dopravních proudů v dopravní síti. Jedná se zde především o veličiny, jako jsou intenzita a hustota. Příkladem zkoumané situace by mohlo být zatížení dopravní sítě ve městě, pokud by se uzavřela jedna z křižovatek z důvodu rekonstrukce. Výsledkem těchto simulací jsou predikce (předpovědi), zda vytvořené řešení situace nebude znamenat kolaps dopravní sítě. Pro vozidlové simulátory nemají velký význam, zde jsou důležité mikrosimulační modely, které zkoumají chování jednotlivých vozidel. [6]

3.2 Mikrosimulace

Podstatou mikrosimulací je modelování chování jednotlivých vozidel po pozemních komunikacích, přičemž zohledňujeme všechny parametry komunikace i vozidla, a to včetně chování řidiče.

Mikrosimulace zpravidla využívají obecně známé parametry. Základem je detailní geometrická definice posuzovaného prostředí tzv. scénář (úsek komunikace, křižovatka). U vozidel je nejdůležitější získat přesné údaje o jejich rozměrech, dosahované rychlosti, zrychlení a zpomalení, hmotnosti vozidla atd. Součástí vstupů jsou dále předem určené modely chování vozidel a ostatních účastníků provozu (chodci, cyklisté, tramvaje atd.).

Na závěr je nutné doplnit údaje o provozu na komunikaci (intenzita dopravy, podíl nákladních vozidel atd.). Tyto vstupy jsou údaje, které jsou standardně získávány z dopravních průzkumů. [7]

3.3 „Car-following“ model

Nejčastěji používaným typem mikrosimulačních modelů je „car-following“ model (model sledu vozidel), který popisuje pohyb a chování vozidla v dopravním proudu v závislosti na předcházejícím vozidle. Vyvíjeny jsou od 50. let 20. století.

Pro vytvoření funkčního „Car-following“ modelu je zapotřebí stanovit závislost zrychlení vozidla na okolních podmínkách. V jednodušších případech to znamená stanovit závislost na stavu vozidla před vozidlem následovaným. Obecně je možné zrychlení vozidla vyjádřit následovně: [7]

$$a = a(v, \Delta v, \Delta x)$$

a zrychlení vozidla

v rychlost vozidla

Δv relativní rychlost v porovnání s následovaným vozidlem a

Δx odstup od následovaného vozidla.

V průběhu času bylo postupným zkoumáním podrobných způsobů a míry ovlivňování vozidel mezi sebou vytvořeno několik modelů. V zásadě existují dva základní přístupy při vytváření těchto modelů.

První způsob se zaměřuje především na pozorování fyzikálních parametrů jako je zrychlení a brzdění (akcelerace a decelerace). Základním předpokladem je, že každý řidič bude dodržovat bezpečnou vzdálenost mezi svým vozidlem a vozidlem před ním regulací své rychlosti. Změnu provede pouze, když následované vozidlo změní rychlost. Relativní rychlost (rozdíl rychlostí jeho vozidla a následovaného vozidla) již není nulová.

Druhý způsob sleduje vzdálenost mezi vozidly. Předpokládáme, že relativní rychlost závisí na změně vzdálenosti mezi vozidly. Regulace rychlosti se provádí ve chvíli, kdy překročí krajní hodnoty intervalu bezpečné vzdálenosti. Pokud vozidlo před ním zvýší svou rychlost, překročí se horní hranice a řidič zvýší svou rychlost. Pokud následované vozidlo sníží svou rychlost, překročí se tím dolní hranice a řidiči sníží svou rychlost.

První generace modelů byly schopné popsat pouze stavy, kdy vozidlo následovalo vozidlo jiné. Moderní modely jsou komplexnější a jsou schopné popsat chování vozidel za jakékoli situace. Lze je nazvat multirežimové modely (jízda po volné komunikaci, jízda s režimem nouzového brzdění atd.), u komplexnějších modelů (VISSIM) může být počet těchto režimů vyšší.

GHR (Gazis - Herman - Rothery) model

GHR model je velmi známým „car following“ modelem. Model je založen na pozorování rychlosti následovaného vozidla, relativní rychlosti, vzdálenosti mezi vozidly a reakční dobou řidiče.

Lineární model

Jeho základní podoba vychází z původního GHR modelu. Avšak modifikuje vztahy mezi zrychlením a rychlostí následovaného vozidla, minimální rozestupy mezi vozidly a reakční dobou řidiče.

Model „bezpečné vzdálenosti“

Model „bezpečné vzdálenosti“ („Safety distance model“) je klíčovým komponentem systému prevence kolizí. Tento model se zaměřuje na přísné dodržování vzdálenosti mezi vozidly. Novější verze tohoto modelu sledují také relativní rychlost mezi vozidly. Jakákoli změna vzdálenosti mezi vozidly vyvolá zpomalení nebo zrychlení pohybu vozidla.

Pravděpodobně nejznámější aplikací modelu „bezpečné vzdálenosti“ je „Gippsův“ model a jeho použití v prostředí AIMSUN 2. Nejsilnější stránkou modelu je jeho reálné zobrazení chování dvou a více vozidela relativně jednoduché kalibrování parametrů. [7]

3.4 Psycho-fyzikální modely

Psycho-fyzikální modely, jak už jejich název napovídá, kombinují psychologické a fyziologické vlastnosti řidiče. Základní myšlenkou modelu je předpoklad, že řidič bude reagovat na podněty, pokud přesáhnou určitou hranici. Může se jednat například o rychlost (zrychlení/zpomalení) následovaného vozidla nebo změna vzdáleností mezi vozidly. Pokud se řidič cítí být ohrožen některým ze sledovaných podnětů, reaguje změnou rychlosti (nebo směru), řidič bude upravovat svou rychlost, dokud se nedostane zpět do své komfortní zóny.

„Wiedemannův model“

Nejznámější aplikací psycho–fyzikálního modelu je „Wiedemannův“ model a jeho použití v programu VISSIM. Na diagramu vzdálenosti mezi vozidly a jejich relativní rychlostí je naznačeno chování řidiče. Sníží-li odstup mezi vozidly na určitou hodnotu, začne snižovat svou rychlost. Překročí-li však hodnota odstupu mezi vozidly hodnotu na horním okraji tohoto intervalu, začne svou rychlost naopak zvyšovat. Řidič se vždy snaží dostat zpět do své komfortní zóny (zóny bez reakce). [7]

3.5 Model chování vozidla Nikoly Čemusové

Tento model chování vozidla je založen na časovém odstupu mezi vozidly. Vozidla jsou v něm uzpůsobena tak, aby si neustále udržovala odstup 2 sekund. Každému vozidlu je zde přiřazeno jedno vozidlo, jako vozidlo sledované.

Obrázek 4 Časový odstup mezi vozidly

V zásadě je použit „Wiedermannův model“. Avšak chování vozidla samotného je rozděleno do více intervalů. Pokud je odstup vozidla mezi 2 a 4 sekundami, vozidlo následuje vozidlo před sebou bez jakékoli změny rychlosti. Překročí-li však tato hodnota 4 sekundy, vozidlo začne zrychlovat. Čím větší je časová mezera mezi vozidly, tím větší zrychlení na automobil působí až do limitní hodnoty, od které je zrychlení konstantní. Mnohem zajímavější je však případ, kdy se vozidlo začne blížit k překážce. V tomto případě je krásně nasimulováno chování skutečného řidiče. Čím menší časová mezera se mezi vozidly nachází, tím větší zpomalení (brzdění) na vozidlo působí. V čase blížícímu se nule se používá maximálního zpomalení. Ve skutečném světě by se v tomto případě dalo mluvit o silnějším sešlápnutí brdového pedálu. Jednotlivé intervaly jsou popsány jako:

Accelerate (časový odstup >6 s)

Vozidlo se pohybuje s maximálním zrychlením až do bodu, kdy dosáhne své maximální rychlosti. Poté se zrychlení sníží na 0 a vozidlo již dále pokračuje svou maximální rychlostí.

Follow accelerate (časový odstup 4-6 s)

Pohyb vozidla je stále zrychlený, ale už ne maximálním zrychlením, ale hodnotou z intervalu komfortních zrychlení. Zvyšuje svou rychlost, dokud se nedostane do blízkosti sledovaného vozidla.

Follow (časový odstup 2-4 s)

V tomto časovém intervalu si vozidlo udržuje svou rychlost na stabilní hodnotě. V ideálním případě se jeho rychlost rovná rychlosti sledovaného vozidla.

Follow brake (časový odstup 1,5-2 s)

Vozidlo začíná zpomalovat se zpomalením z intervalu komfortních zpomalení a to až do chvíle, kdy se dostane do časového odstupu většího než 2 sekundy.

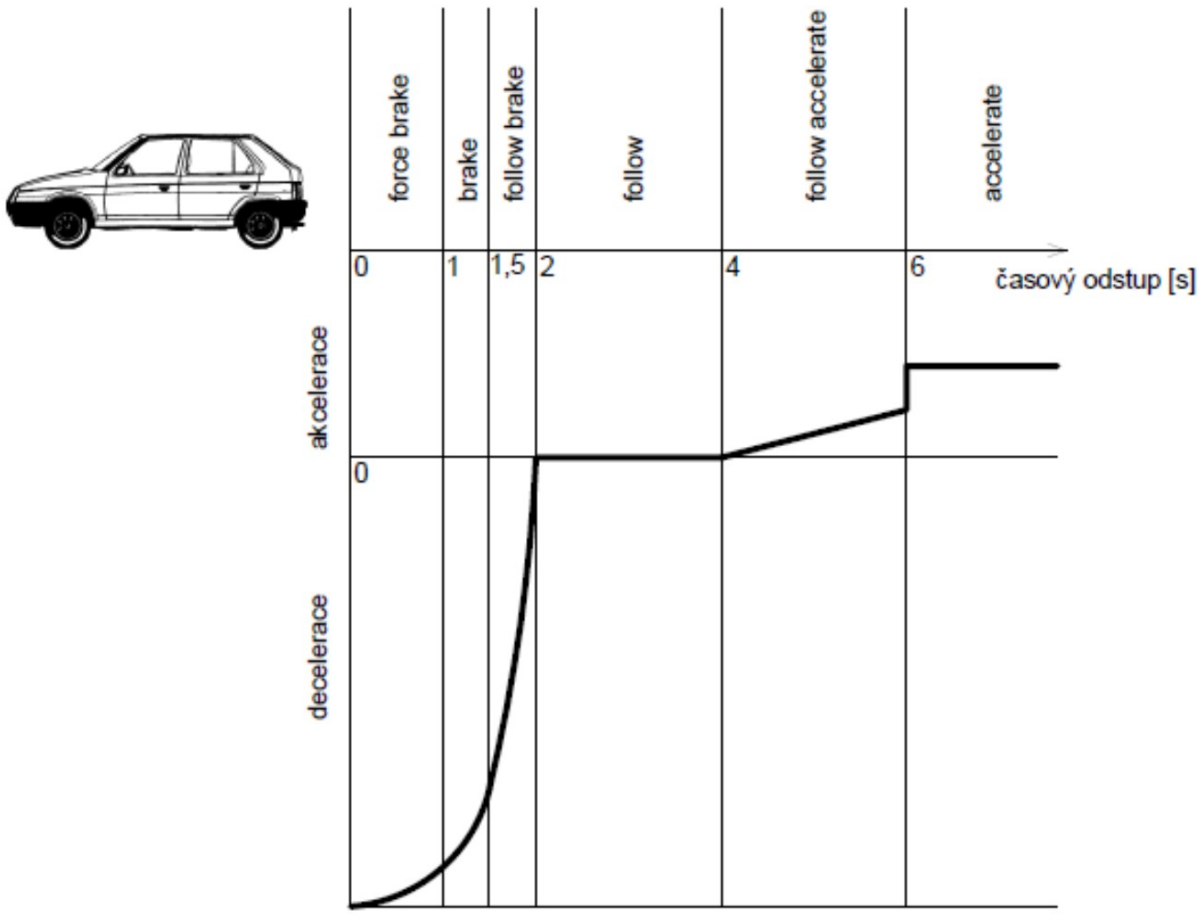
Brake (časový odstup 1-1,5 s)

Jestliže však brzdění z intervalu „Brake“ není dostačující a sníží se časový odstup mezi vozidly pod 1,5 sekundy, začne vozidlo brzdit se zvýšeným zpomalením.

Force Brake (časový odstup <1 s)

V tomto intervalu již simulované vozidlo brzdí s maximální účinností brzd, čímž se snaží zamezit případné srážce vozidel.

Tento model chování vozidla je pojatý skvěle v případě, že se jedná o provoz, ve kterém není možné předjíždění. Proto jsem se rozhodl, že vytvořím model chování, který tento doplní o schopnost předjíždět. [7]



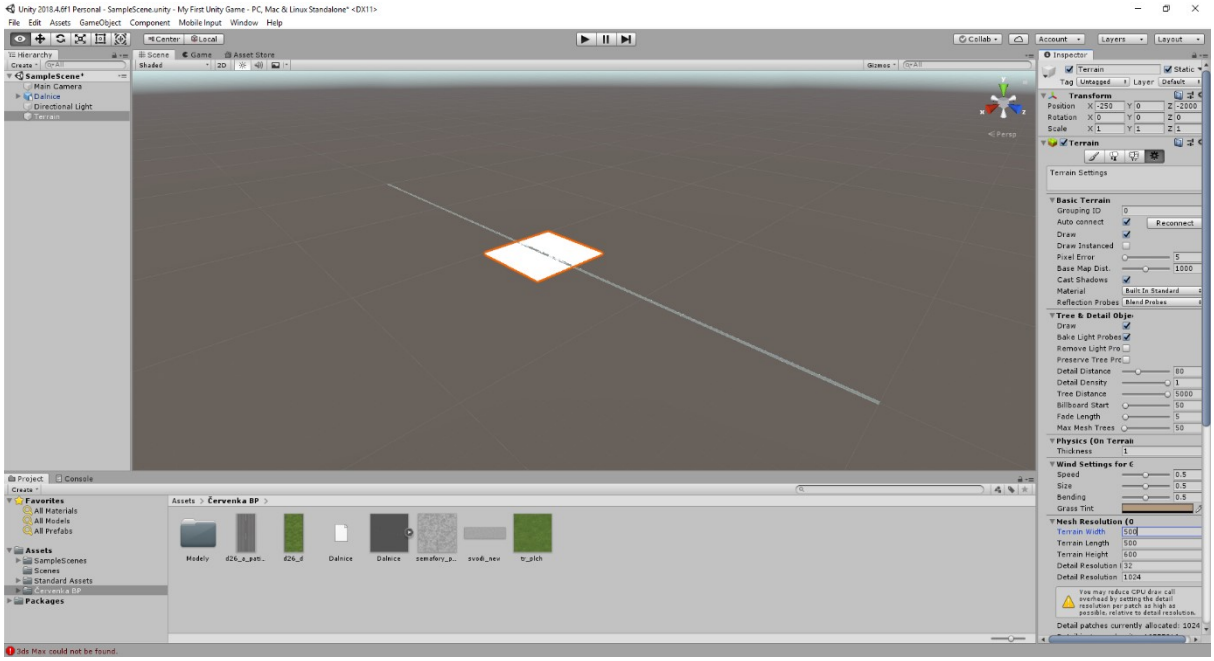
Obrázek 5 Časové intervaly popisující chování vozidla

PRAKTICKÁ ČÁST

4 Tvorba testovacího scénáře

Pro testování mého modelu chování vozidla jsem si vytvořil provizorní scénář, který při testování bez problémů postačí.

V programu rhinoceros jsem si vytvořil 4 km dálnice, které pro naše potřeby postačí. Tuto komunikaci jsem poté vložil do scény herního enginu Unity, ve kterém jsem se rozhodl svůj model chování vozidla vyvinout a posléze i testovat.

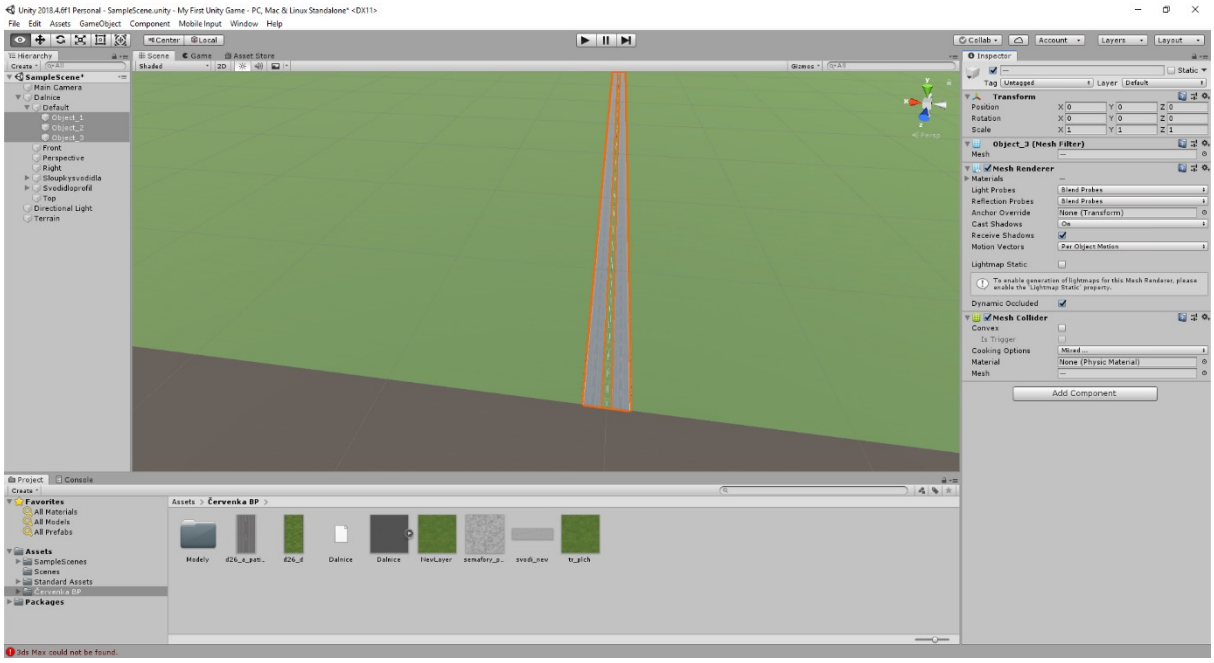


Obrázek 6 Dálnice 1

Následně jsem vytvořil terén (vložil jsem herní objekt „terrain“, zvětšil ho do velikosti komunikace a použil texturu trávy pro obarvení), aby scénář nebyl pouze jedna směrově rozdělená komunikace.

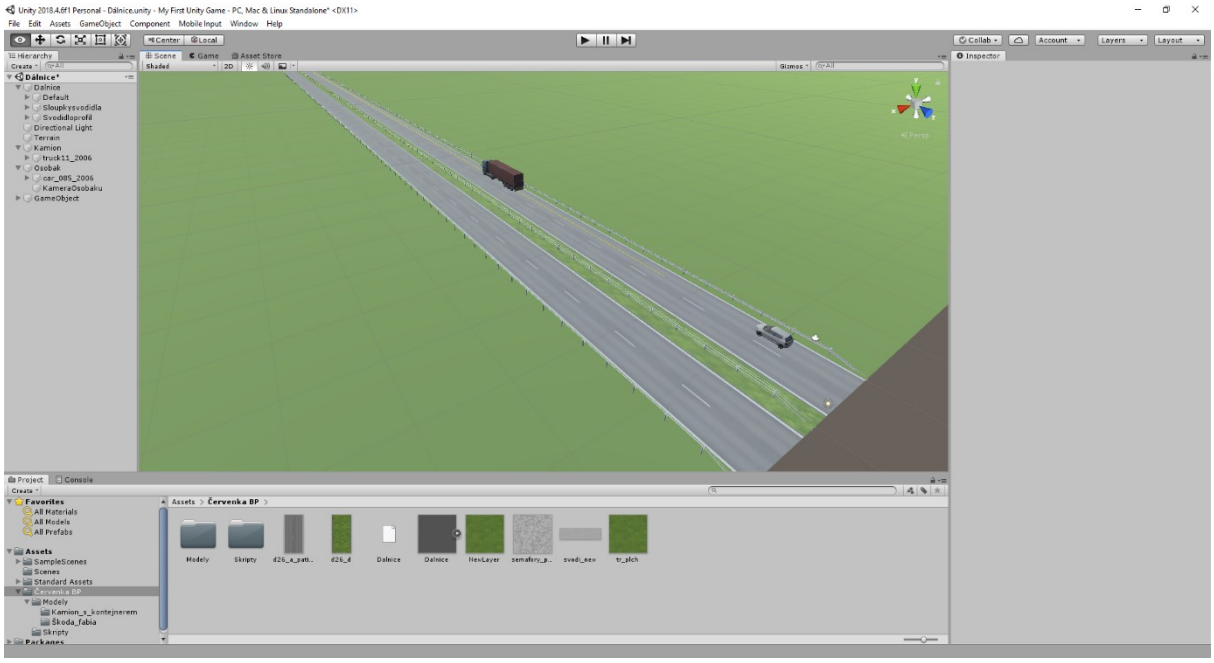
Dálnici jsem rozložil na jednotlivé položky, aby se dala jednodušeji upravovat. V mém případě se jednalo o úpravu textury svodidel. Protože se jedná jen o testovací část dálnice, další úpravy jsem nepovažoval za nutné.

V Případě potřeby však není problém přidat do scénáře další objekty, jako jsou stromy, budovy dráty elektrického vedení a podobně.



Obrázek 7 Dálnice 2

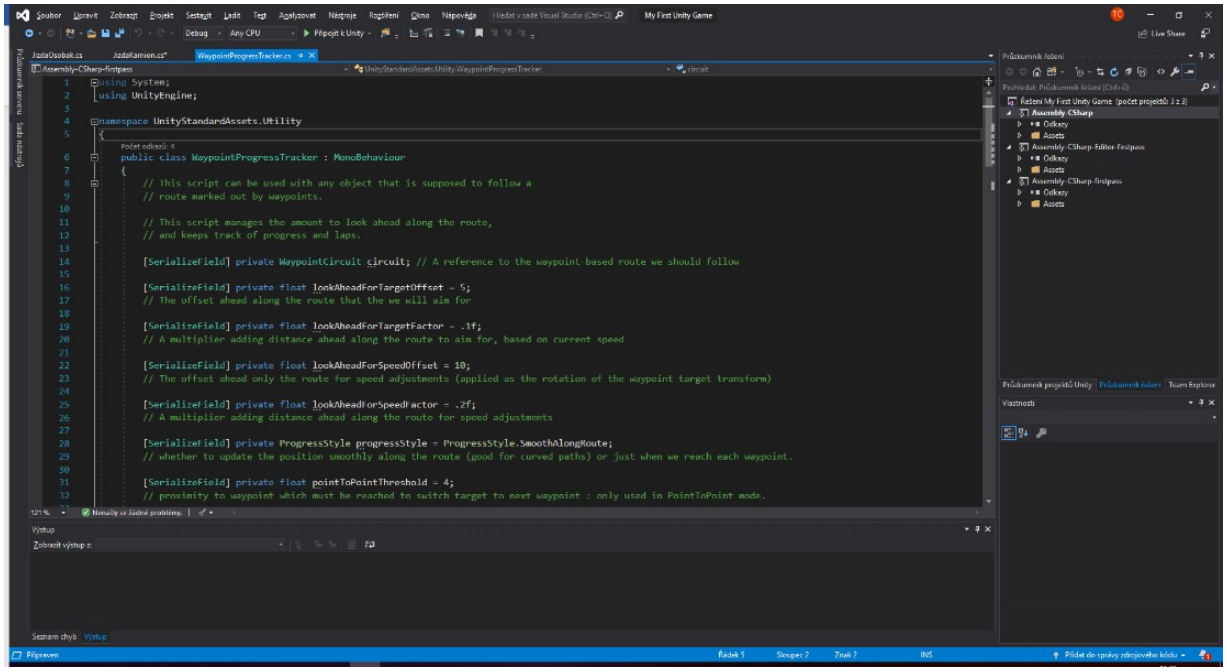
Na závěr jsem do scénáře přidal dva automobily, které poslouží při testování vytvořených scriptů. Podle komplikovanosti scriptů postupně budou další automobily přibývat. Aby bylo možné vytvořit co nejpřesnější model chování, bude zapotřebí simulovat co nejvíce vozidel.



Obrázek 8 Dálnice 3

5 Vývoj modelu chování vozidla

V této části se zaměřím na samotnou tvorbu modelu chování vozidla. Pro tuto část jsem využil program Microsoft Visual Studio, který při tvorbě scriptů používá programovací jazyk C#.



Obrázek 9 Microsoft Visual Studio

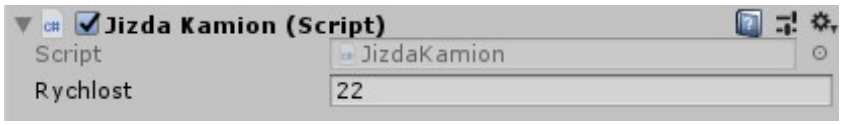
Pro sledování určité křivky budu v průběhu vytváření scriptů používat opensource (volně přístupnou) funkci „WaypointProgressTracker“, díky které se vozidlo bezpečně udrží na komunikaci a umožní mi tím zaměřit se na samotný model chování vozidla. [9]

5.1 Model jízdy v jízdním pruhu

Nejdříve jsem si vytvořil prosté chování vozidla, které mi poslouží jako vozidlo následované. Toto vozidlo má jednu prostou funkci: Sledovat předem definovanou křivku („WaypointProgressTracker“). A udržet si konstantní rychlost.

Pseudokód

1. public float Rychlost = rychlost automobilu, kterou si uživatel může sám zvolit
2. **void update** (proveď následující výpočty pro každý nový snímek)
3. Teleso.velocity = provádí výpočet nové pozice objektu v závislosti na rychlosti a směru pohybu (předem definované křivky)
4. transform.LookAt = zajistí, aby vozidlo bylo vždy natočeno rovnoběžně s křivkou
5. **opakuj od bodu 2.**



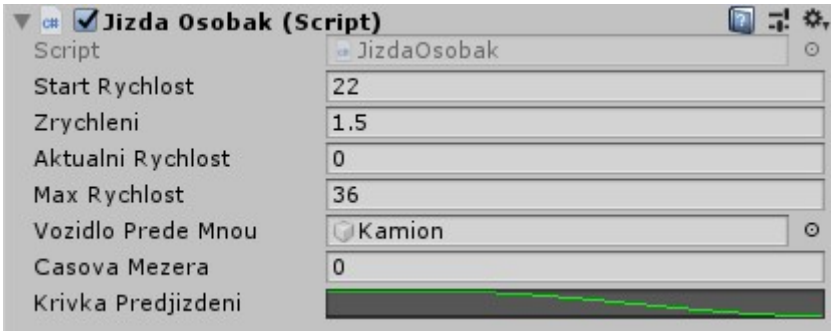
Obrázek 10 Detail scriptu "Jizda Kamion" v rozhraní Unity

5.2 Model chování v závislosti na časovém odstupu

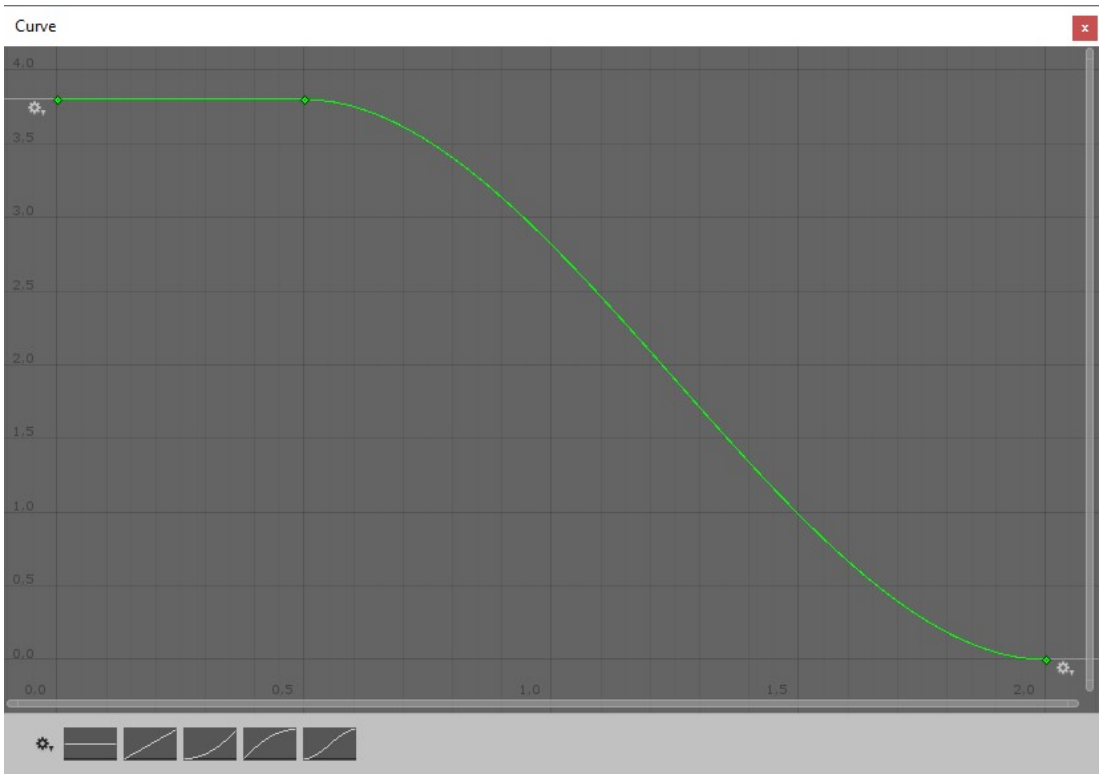
Poté jsem se rozhodl vytvořit model vozidla, který by fungoval na základě detekce časového odstupu mezi ním a vozidlem, které se nachází před ním. Můj záměr byl tento script použít jako doplnění do modelu chování vozidla Nikoly Čemusové. V případě, že se vozidlo blíží ke sledovanému vozidlu a časový odstup mezi nimi se sníží na předem definovanou hodnotu (v tomto případě 2 sekundy), začne sledované vozidlo předjíždět. Pokud tato podmínka není splněna (pokud je časová mezera větší než 2 sekundy, nebo se před vozidlem žádné vozidlo nenachází), bude se vozidlo nadále pohybovat po sledované křivce a je-li to možné (není na své maximální rychlosti), bude zároveň zvyšovat svou rychlost.

Pseudokód

1. StartRychlost = počáteční rychlost zvolená uživatelem
2. Zrychleni = hodnota zrychlení (při testování scriptu 1,5 m/s^2)
3. AktualniRychlost = aktuální hodnota rychlosti (tuto hodnotu nelze zvolit, ale lze ji během testování monitorovat)
4. MaxRychlost = zvolená maximální rychlost
5. VozidloPredeMnou = objekt zvolený jako sledované vozidlo
6. CasovaMezera = aktuální hodnota časového odstupu mezi vozidly
7. KrivkaPredjizdeni = křivka, kterou si uživatel sám nadefinuje (vozidlo ji bude používat při předjíždění)
8. **void update** (proveď následující výpočty pro každý nový snímek)
9. CasovaMezera = výpočet časové mezery na základě polohy a rychlosti
10. if (AktualniRychlost<MaxRychlost)
11. AktualniRychlost = AktualniRychlost + Zrychleni * čas
12. if (CasovaMezera< 2)
13. Teleso.velocity = výpočet nové pozice na základě „KrivkaPredjizdeni“
14. transform.LookAt = vozidlo sleduje svůj nový směr pohybu
15. else (pokud podmínka splněna není)
16. Teleso.velocity = vozidlo se pohybuje po původní křivce
17. transform.LookAt = vozidlo sleduje původní křivku
18. **opakuj od bodu 8.**



Obrázek 11 Detail scriptu "JizdaOsobak" v rozhraní Unity



Obrázek 12 Detail nastavení křivky předjíždění

Tento model funguje skvěle v případě, že se v simulaci nacházejí pouze dva automobily. V případě, že by se v simulaci nacházel další automobil, byla by jeho detekce příliš obtížná. Proto jsem se rozhodl udělat druhý model předjíždění, který by byl schopný detekovat libovolné vozidlo.

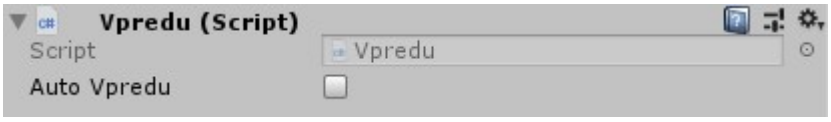
5.3 Model chování v závislosti na detekci pomocí triggerů

Pro svůj další model jsem se rozhodl využít funkci enginu Unity tzv. triggerů. Trigger (neboli spoušť) je objekt v herním enginu, kterému je přidělena funkce „BoxCollider“ (kolizní krabice). V této funkci musí být zároveň vybrána vlastnost „IsTrigger“ (je spouští). Všem vozidlům, které jsem v této simulaci potřeboval detekovat, jsem také přidal kolizní krabice a označil jsem je „Tagem“ (značkou) „CarCollider“, aby vozidlo nemohlo detekovat kolizi s ostatními triggery. Protože jsem ve svém dalším modelu chtěl naučit vozidlo nejen detekovat libovolné vozidlo před sebou, ale zároveň také detekovat vozidlo, které by při předjíždění mohlo být hrozbou, potřeboval jsem pro tento model triggery dva. Jeden pro detekci vozidla vpředu a druhý pro detekci vozidla vlevo. Pro tyto triggery jsem si vytvořil dva scripty. Scripty jsou téměř identické, proto zde popíši pouze jeden z nich.

Pseudokód

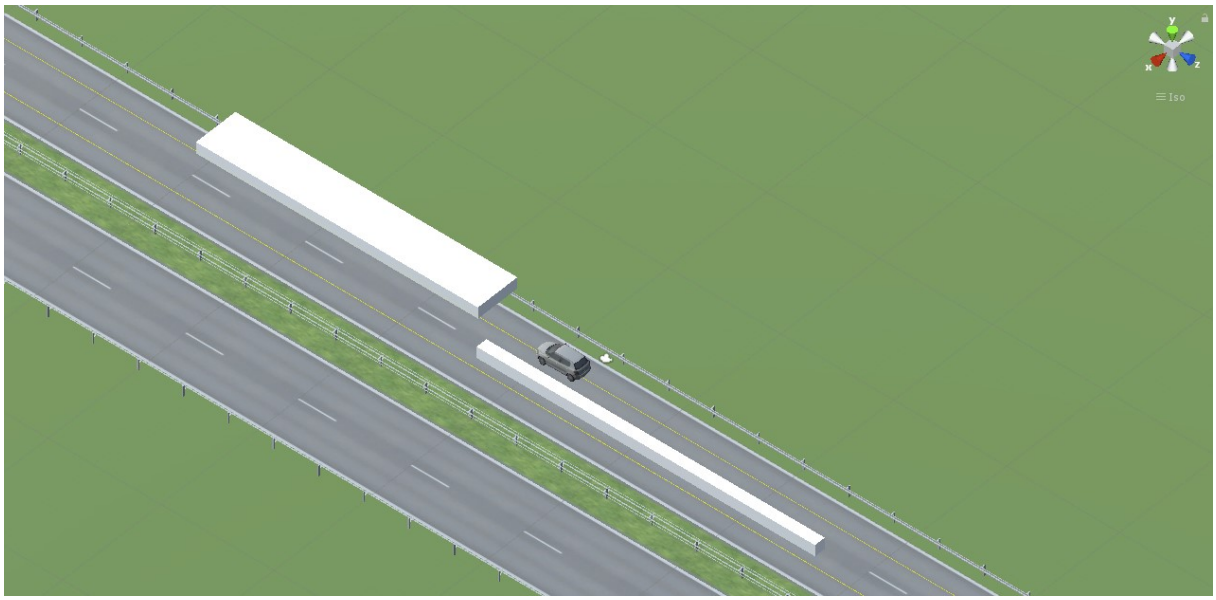
1. AutoVpredu = logická proměnná (zaznamenává pouze dva stavy true/false) na začátku nastavená jako false
2. **void OnTriggerEnter** (pokud do tohoto triggeru vstoupí jiný „BoxCollider“)
3. if(má objekt „Tag“ = „CarCollider“)
4. AutoVpredu = true
5. **void OnTriggerStay** (pokud se uvnitř tohoto triggeru nachází „BoxCollider“)
6. if (má objekt „Tag“ = „CarCollider“)
7. AutoVpredu = true
8. **void OnTriggerExit** (pokud tento triggeru opustí jiný „BoxCollider“)
9. if (má objekt „Tag“ = „CarCollider“)
10. AutoVpredu = false

Detekce, zda se uvnitř triggeru stále něco nachází, byla nutná, protože se při testování stávalo, že tento trigger opustilo vozidlo jiné než to, které bylo právě předjížděné, což způsobovalo ukončení předjížděcího manévru a srážku vozidel. Ve scriptu se také nachází příkazy na změny barev, které pro samotnou funkci nejsou důležité, ale posloužily při testování triggerů. V uživatelském rozhraní unity je pak pouze vidět, zda je nebo není detekovaná přítomnost jiného vozidla.



Obrázek 13Detail scriptu "Vpredu" v rozhraní Unity

V tuto chvíli jsem mohl začít s vývojem modelu chování vozidla samotného. Malou část tohoto modelu jsem převzal z modelu předchozího (model chování v závislosti na časovém odstupu). Ke schopnostem sledovat křivku, zvyšovat svou rychlost do doby dosažení maximální rychlosti a předjíždění (v tuto chvíli již libovolných) vozidel, jsem navíc přidal schopnost přibrzdit, popřípadě i zastavit, pokud se v levém jízdním pruhu nachází jiné vozidlo, které by mohlo předjížděcí manévr ohrozit. Vozidlo jsem také naučil setrvat v levém jízdním pruhu, dokud návrat do pravého jízdního pruhu není bezpečný (již se v něm nenachází žádné další vozidlo).

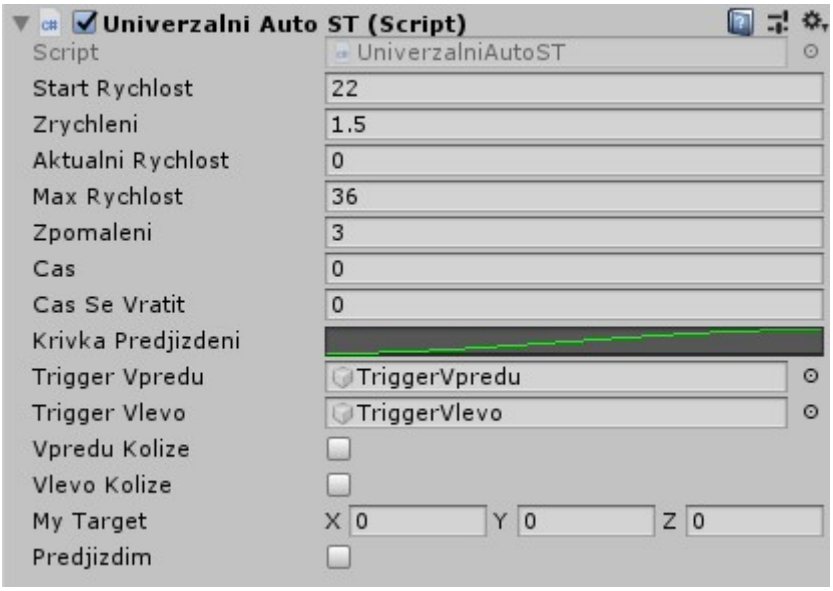


Obrázek 14 Statické triggery

Pseudokód

1. StartRychlost = počáteční rychlost zvolená uživatelem
2. Zrychleni = hodnota zrychlení (při testování scriptu 1,5 m/s^2)
3. AktualniRychlost = aktuální hodnota rychlosti (tuto hodnotu nelze zvolit, ale lze ji během testování monitorovat)
4. MaxRychlost = zvolená maximální rychlost
5. Zpomaleni = hodnota zpomalení (při testování scriptu 3 m/s^2)
6. Cas = čas, který se začne přičítat s každým snímkem při předjíždění
7. CasSeVratit = čas, který se začne odečítat při návratu do pravého jízdního pruhu
8. KrivkaPredjizdeni = křivka, kterou si uživatel sám nadefinuje (vozidlo ji bude používat při předjíždění)
9. TriggerVpredu = detekce vozidla vpředu
10. TriggerVlevo= detekce vozidla v levém jízdním pruhu
11. VpreduKolize = pomocná proměnná pro detekci vozidel
12. VlevoKolize = pomocná proměnná pro detekci vozidel
13. MyTarget = pomocný vektor pro výpočet směru pohybu
14. Predjizdim = logická proměnná pro snazší výpočet křivky návratu po předjíždění
15. **void update** (proveď následující výpočty pro každý nový snímek)
16. if (VpreduKolize == true „a zároveň“ VlevoKolize == true)
17. AktualniRychlost = AktualniRychlost – Zpomaleni * čas
18. Cas = 0 (proměnná „Cas“ se v tomto kroku nastaví na hodnotu 0)
19. else (pokud podmínka splněna není)
20. if (VpreduKolize == true)
21. Cas = Cas + čas (nová hodnota je stará hodnota plus skutečný čas)
22. MyTarget = nová hodnota se vypočte v závislosti na proměnné „Cas“
23. Predjizdim = true (nová hodnota proměnné)
24. if (VpreduKolize == false „a zároveň“ Vlevokolize == false)
25. if (Predjizdim == true)
26. if (CasSeVratit> 0)
27. CasSeVratit = CasSeVratit – čas
28. MyTarget = nová hodnota v závislosti na proměnné „CasSeVratit“
29. else (po návratu do pravého pruhu je „CasSeVratit“=0)
30. Predjizdim = false
31. CasSeVratit = 4 (neznámá se musí připravit na další předjíždění)
32. if (AktualniRychlost<MaxRychlost)
33. AktualniRychlost = AktualniRychlost + Zrychleni * čas

34. Teleso.velocity = vozidlo se pohybuje po křivce závislé na „MyTarget“
35. transform.LookAt = vozidlo sleduje křivce závislé na „MyTarget“
36. **opakuj od bodu 15.**

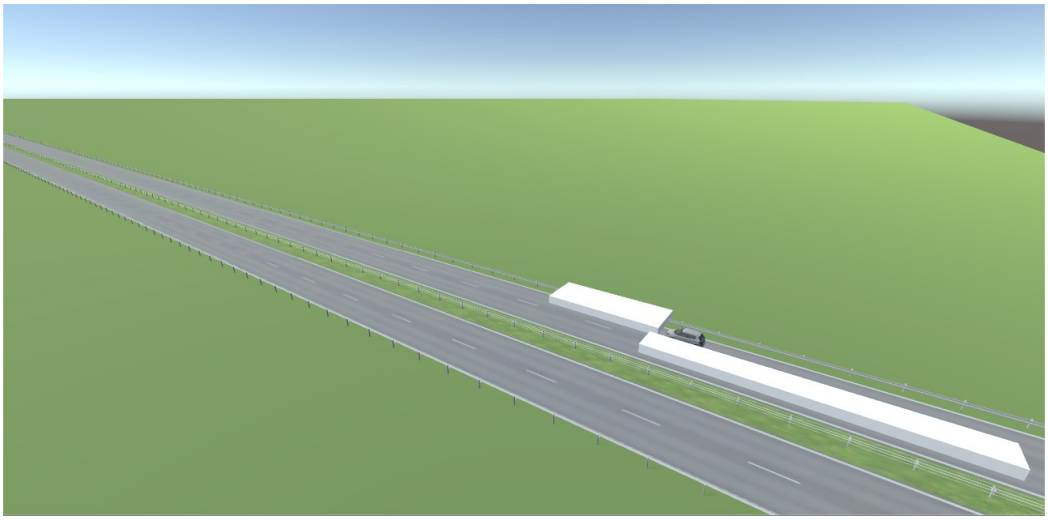


Obrázek 15 Detail scriptu "Univerzalni Auto ST" v rozhraní Unity

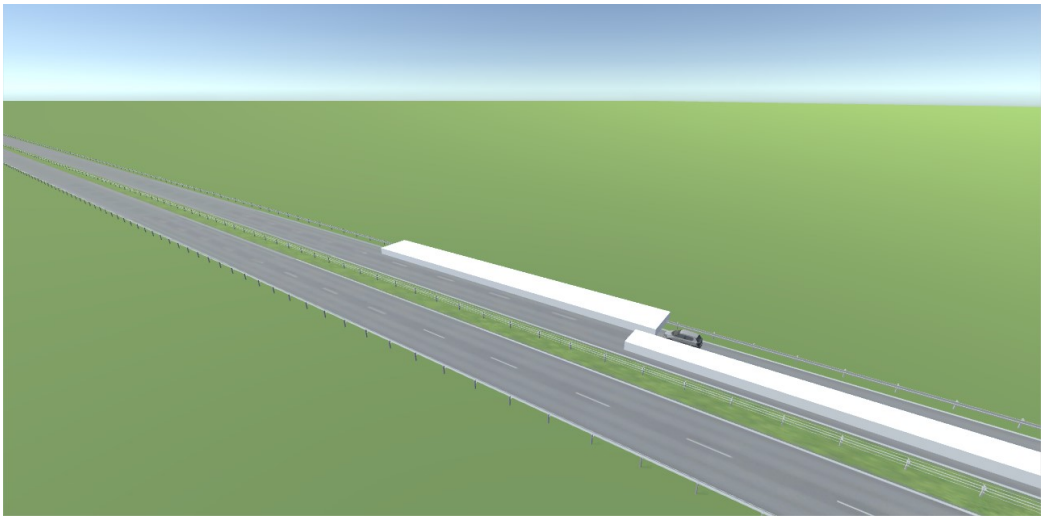
U tohoto modelu jsem se však setkal s problémem, kdy triggery nevyhovovaly použitým rychlostem. Byly buď moc dlouhé, což způsobovalo, že vozidla začínala zpomalovat se zbytečně velikým odstupem, nebo naopak nezvládala v čas zpomalit, či se překážce vyhnout. Proto jsem se rozhodl zaměřit se na rychlosti vozidel.

5.4 Model chování v závislosti na detekci pomocí dynamických triggerů

Mým dalším a již posledním modelem chování vozidla je model založený na detekci pomocí triggerů. V tomto případě však kolizní krabice, které jsou triggery, mají schopnost měnit svou velikost na základě rychlosti pohybu vozidla. Navíc si uživatel u tohoto modelu může sám zvolit, zda chce, aby jednotlivá vozidla měla schopnost předjíždět, nebo pouze následovat a upravovat svou rychlost na základě spuštění triggeru „Vpredu“. Pro tuto funkci jsem si vytvořil pro každý trigger tzv. „Parrent Game Object“ (rodičovský objekt), díky kterému je možné ovládat nejen růst jednotlivých triggerů, ale také směr jejich růstu.



Obrázek 16 Dynamické triggery před spuštěním simulace

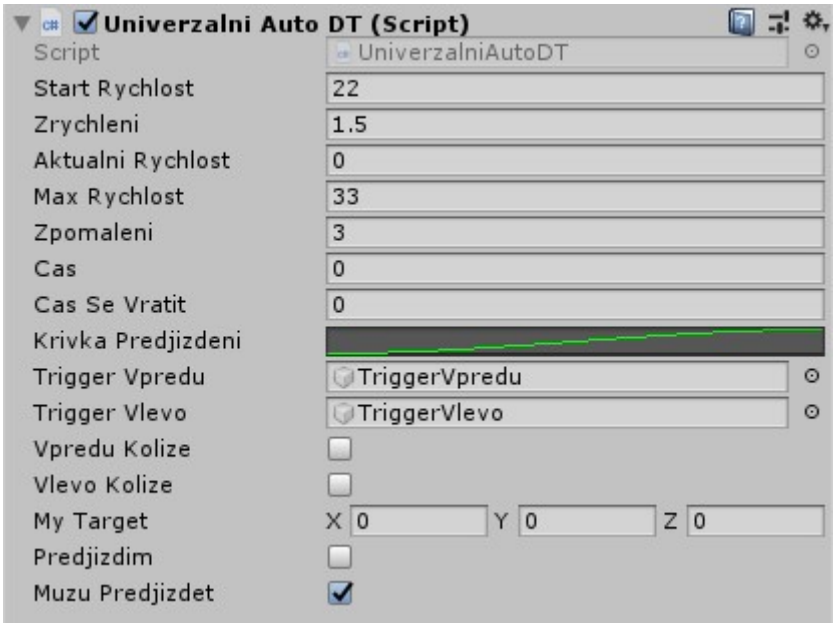


Obrázek 17 Dynamické triggery po spuštění simulace

Pseudokód

1. StartRychlost = počáteční rychlost zvolená uživatelem
2. Zrychleni = hodnota zrychlení (při testování scriptu 1,5 m/s^2)
3. AktualniRychlost = aktuální hodnota rychlosti (tuto hodnotu nelze zvolit, ale lze ji během testování monitorovat)
4. MaxRychlost = zvolená maximální rychlost
5. Zpomaleni = hodnota zpomalení (při testování scriptu 3 m/s^2)
6. Cas = čas, který se začne přičítat s každým snímkem při předjíždění
7. CasSeVratit = čas, který se začne odečítat při návratu do pravého jízdního pruhu
8. KrivkaPredjizdeni = křivka, kterou si uživatel sám nadefinuje (vozidlo ji bude používat při předjíždění)
9. TriggerVpredu = detekce vozidla vpředu
10. TriggerVlevo = detekce vozidla v levém jízdním pruhu
11. VpreduKolize = pomocná proměnná pro detekci vozidel
12. VlevoKolize = pomocná proměnná pro detekci vozidel
13. MyTarget = pomocný vektor pro výpočet směru pohybu
14. Predjizdim = logická proměnná pro snazší výpočet křivky návratu po předjíždění
15. MuzuPredjizdet = logická proměnná, díky které si uživatel může sám zvolit, zda chce, aby vozidlo mělo schopnost předjíždět
16. **void update** (proveď následující výpočty pro každý nový snímek)
17. if (MuzuPredjizdet == true)
18. if (VpreduKolize == true „a zároveň“ VlevoKolize == true)
19. AktualniRychlost = AktualniRychlost – Zpomaleni * čas
20. Cas = 0 (proměnná „Cas“ se v tomto kroku nastaví na hodnotu 0)
21. else (pokud podmínka splněna není)
22. if (VpreduKolize == true)
23. Cas = Cas + čas (nová hodnota je stará hodnota plus skutečný čas)
24. MyTarget = nová hodnota se vypočte v závislosti na proměnné „Cas“
25. Predjizdim = true (nová hodnota proměnné)
26. if (VpreduKolize == false „a zároveň“ Vlevokolize == false)
27. if (Predjizdim == true)
28. if (CasSeVratit> 0)
29. CasSeVratit = CasSeVratit – čas
30. MyTarget = nová hodnota v závislosti na proměnné „CasSeVratit“
31. else (po návratu do pravého pruhu je „CasSeVratit“=0)
32. Predjizdim = false

33. CasSeVratit = 4 (neznámá se musí připravit na další předjíždění)
34. if (AktualniRychlost<MaxRychlost)
35. AktualniRychlost = AktualniRychlost + Zrychleni * čas
36. if (MuzuPredjizdet == false)
37. if (VpreduKolize == true)
38. AktualniRychlost = AktualniRychlost – Zpomaleni * čas
39. else (pokud podmínka splněna není)
40. if (AktualniRychlost<MaxRychlost)
41. AktualniRychlost = AktualniRychlost + Zrychleni * čas
42. TriggerVpredu.localScale = TriggerVpredu mění svou velikost (AktualniRychlost/10)
43. TriggerVlevo.localScale = TriggerVlevo mění svou velikost (AktualniRychlost/10)
44. Teleso.velocity = vozidlo se pohybuje po křivce závislé na „MyTarget“
45. transform.LookAt = vozidlo sleduje křivce závislé na „MyTarget“
46. **opakuj od bodu 16.**

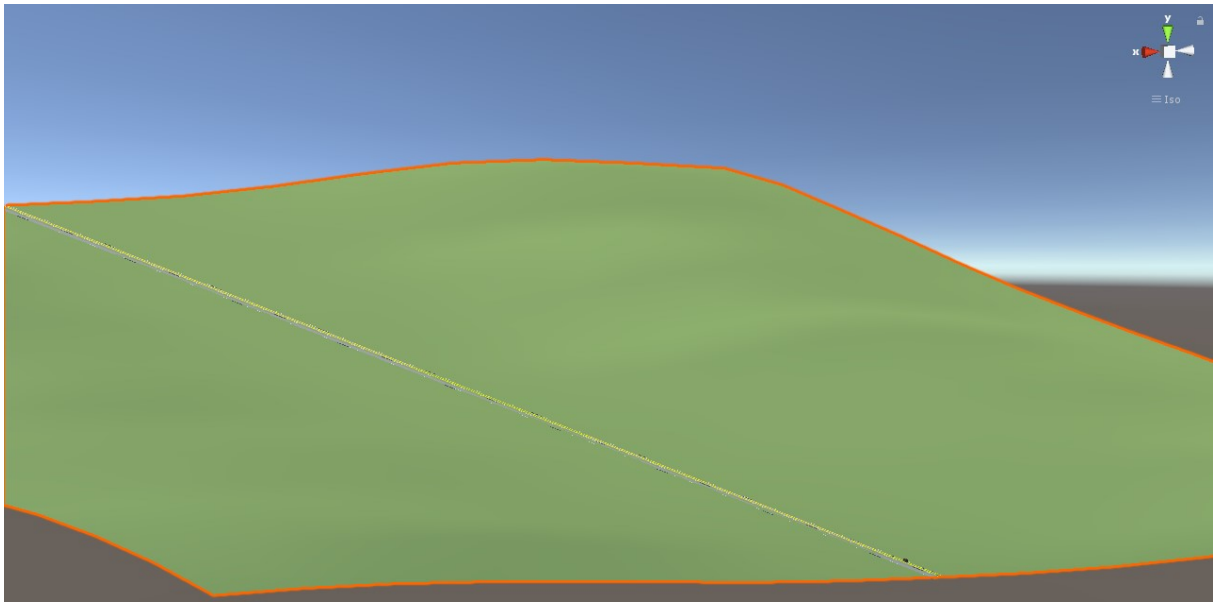


Obrázek 18 Detail scriptu "Univerzalni Auto ST" v rozhraní Unity

Tento model vozidla je nejvíce multifunkční. Po zvolení jednotlivých hodnot pro každé vozidlo v simulátoru a správném umístění jednotlivých triggerů a jejich rodičů je možné nasimulovat relativně přesný a živý provoz do simulátorů. Díky dynamickým triggerům by neměl být problém nasimulovat provoz téměř v jakékoli rychlosti. Model je možné použít pro neomezené množství vozidel. Díky způsobu detekce nezáleží, jestli se jedná o simulaci 2 nebo 30 vozidel, vozidla budou vždy schopna se vzájemně detekovat.

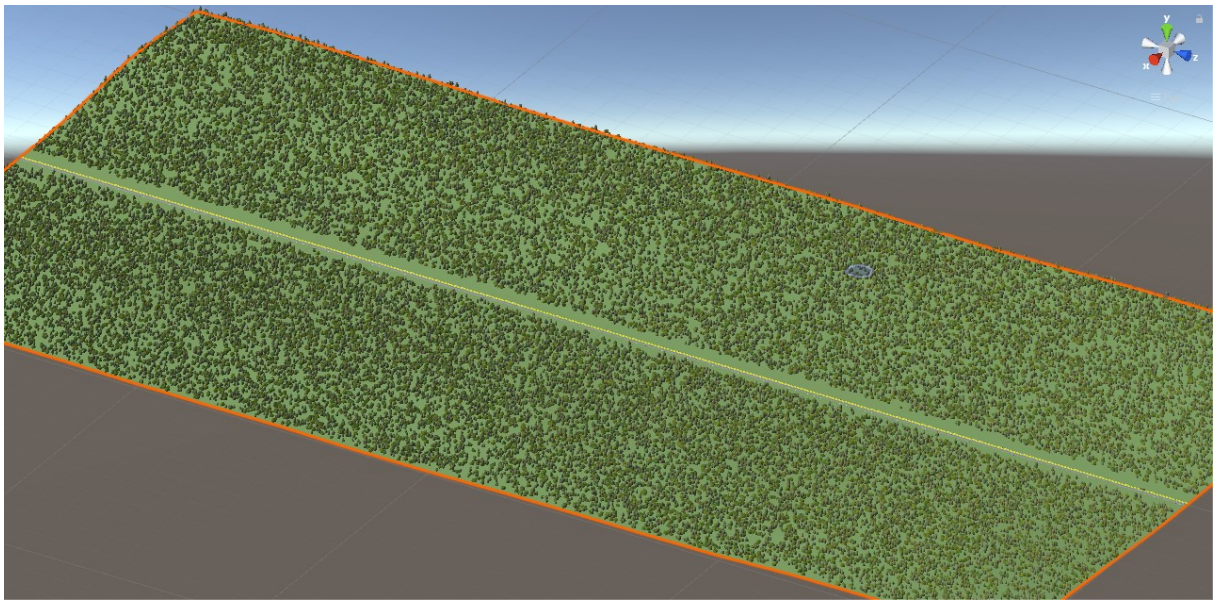
6 Experiment pro ověření funkčnosti

V této poslední kapitole vytvořím kompletní scénář, který využiji při závěrečném experimentu pro ověření funkčnosti modelu chování vozidla v závislosti na detekci pomocí dynamických triggerů. Jako základ tohoto scénáře použiji svůj scénář původně vytvořený pro testování modelů chování. V původním scénáři je terén pouze rovná zelená plocha, proto jsem tuto plochu upravil tak, aby doopravdy připomínala terén.



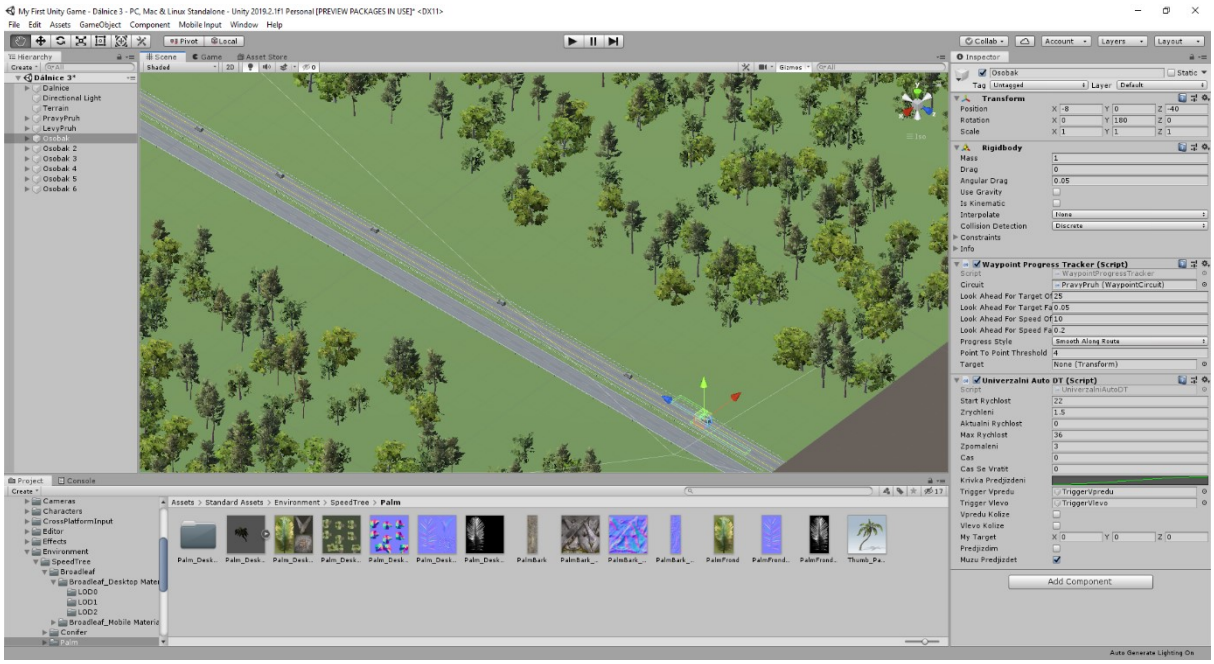
Obrázek 19 Vytvarování terénu

Poté co jsem upravil terén, jsem se rozhodl přidat do scény stromy. Aby celý les nebyl tvořen pouze jedním druhem stromů, vybral jsem si z balíčku „Standart Assets“, který je volně stažitelný, tři různé stromy. Ty jsem pomocí příkazu „Mass Placement“ nechal náhodně rozmístit po celém terénu. Některé z těchto stromů se objevily i přímo uprostřed dálnice. Tyto stromy jsem štětcem vymazal a vytvořil tím bezpečný prostor po obou stranách dálnice.



Obrázek 20 Vygenerování lesa

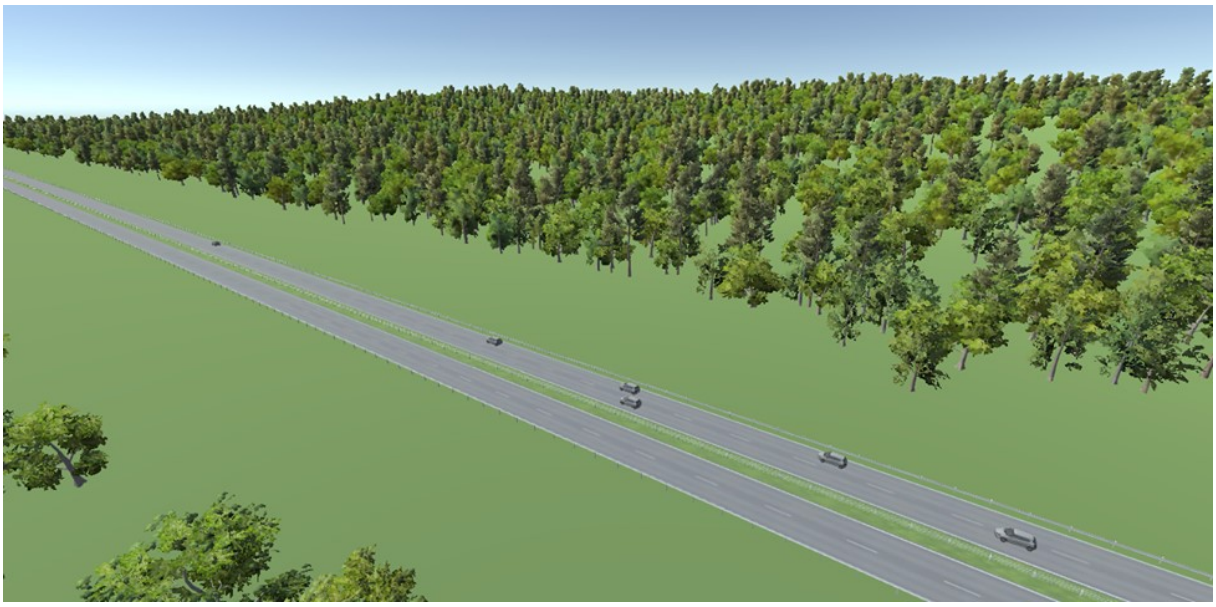
Na závěr jsem upravil i vozidla v simulaci. Odstranil jsem všechna vozidla, která sloužila jako pomocná (nákladní vozidlo a osobní vozidlo jedoucí v levém jízdním pruhu) a přidal jsem další vozidla s modelem chování v závislosti na detekci pomocí dynamických triggerů. Některým z těchto vozidel jsem dal zákaz předjíždění. Nakonec jsem jen upravil kameru, aby zobrazovala co největší množství vozidel během simulace.



Obrázek 21 Výsledný scénář

Každému vozidlu v simulaci jsem také nastavil jinou maximální rychlost, aby k předjíždění vůbec mohlo dojít. Během experimentu samotného se nevyskytly žádné problémy. Vozidla, která měla povoleno předjíždět, bez problému předjela všechna ostatní vozidla a pomalu se ztrácela z dohledu. Vozidla, která tuto schopnost povolenou neměla, se po zbytek simulace pohybovala společně s pevnými rozestupy.

Na obrázku 22 je vidět, že vozidlo s nejvyšší rychlostí mizí ze snímku. Dále zde vidíme, že pomalejší vozidlo zbrzdilo dvě vozidla, která neměla povolení předjíždět. Vozidlo, na kterém byla kamera přichycena, se udržuje v levém jízdním pruhu a poslední vozidlo v simulaci právě začíná svůj předjížděcí manévr.



Obrázek 22 Průběh experimentu

Závěr

Cílem této bakalářské práce bylo vytvoření mikrosimulačního modelu chování vozidel pro vozidlový simulátor. Model chování vozidla jsem vytvořil za použití scriptů vytvořených v programovacím jazyce C# a herního enginu Unity, ve kterém jsem poté daný model chování vozidla i otestoval.

Jako první jsem se zaměřil na vozidlové simulátory, jaká je jejich funkce a význam. Z čeho se tyto simulátory skládají (hardwarové i softwarové součásti). Popsal jsem nedostatky simulátoru FD ústavu K616. Po zjištění, že v tomto simulátoru není možné předjíždění vozidel, jsem se zaměřil především na vytvoření modelu, který by tuto schopnost měl.

Pro vytvoření modelu chování vozidla jsem však nejdříve musel prozkoumat kinematiku a dynamiku vozidel. Začal jsem od základních pohybových rovnic, pokračoval jsem přes jízdní odpory, dotknul jsem se hnací síly vozidla a nakonec jsem došel až ke Kovaříkově vzorci, který mi pomohl určit hodnoty potřebné pro vytvoření křivky předjíždění.

Po prozkoumání kinematiky a dynamiky vozidel jsem se vrátil zpět simulacím konkrétně ke stávajícím makrosimulačním a mikrosimulačním modelům. Zjistil jsem, že některé z nich fungují na jednoduchých principech, jako jsou časové odstupy. Některé však jsou složitější a napodobují psychiku řidičů. Z těchto modelů byl nejzajímavějším modelem „Wiedermannův model“. Na závěr jsem se zaměřil na model chování vytvořený mou kolegyní z Prahy, Nikolou Čemusovou, který byl podle mého názoru úžasný. Avšak i tomuto modelu stále chybělo něco, co mě pronásledovalo už od samého začátku, neschopnost vozidel předjíždět pomalejší vozidla.

Protože žádný z modelů chování vozidel, na které jsem se při své práci narazil neměl schopnost předjíždět, rozhodl jsem se vytvořit model chování, který by tuto schopnost měl. Nejdříve jsem se pokusil navázat na svou kolegyni a vytvořit model založený na sledování časového odstupu mezi vozidly. Tento model jsem vytvořil, ale nakonec jsem ho shledal neúčinným, protože vozidlo bylo schopné sledovat pouze jeden cíl. Tím pádem by tento model byl v praxi použitelný, pouze pokud by se v simulaci nacházela dvě vozidla. V jiném případě by vozidlo svůj cíl předjelo a hned do dalšího by narazilo, protože by ho nebylo schopné detekovat.

Zaměřil jsem se tedy na schopnost vozidla detekovat libovolné překážky. Pro tento model jsem využil tzv. triggerů (spouští) a založil jsem chování vozidla na detekci vozidel pomocí box colliderů (kolizních krabic). Tento model už byl znatelně lepší než předešlý, dokázal detekovat libovolná vozidla. Ale ukázalo se, že rychlosti pohybu musí v modelu chování vozidla hrát roli, protože při testování se stávalo, že vozidla buď nestíhala zastavit dostatečně rychle, nebo brzdila se zbytečně velkým odstupem.

Proto jsem do modelu chování přidal příkaz, kterým si vozidlo samo mění velikost svých triggerů na základě své rychlosti. Díky tomu byl předešlý problém eliminován. Na závěr jsem do modelu přidal „quality of life“ (pro zvýšení pohodlí) příkaz, kterým si uživatel může sám zvolit, zda chce, aby vozidlo mělo schopnost předjíždět nebo ne.

V Poslední části jsem vytvořil scénář v hernímu enginu Unity pro ověření funkčnosti tohoto modelu chování, aby se nejednalo pouze o dálnici na zelené ploše. Použil jsem svůj testovací scénář a doplnil ho o pár prvků, díky kterým nevypadá tak prázdný. Poté jsem do scény přidal několik dalších vozidel, abych ověřil všechny funkce modelu chování vozidla.

Se svým posledním modelem chování vozidla jsem velmi spokojen. Lze ho použít pro libovolné množství vozidel v simulaci. Uživatel si sám může zvolit jednotlivé parametry chování vozidla, jako jsou rychlost, zrychlení, schopnost předjíždět atd. Model je vytvořený pouze pro směrově rozdělené komunikace, ale při dostatku času a nápaditosti by neměl být problém modifikovat ho i pro ostatní komunikace.

Použité zdroje

1. PUDOVÁ, Bc. Klára. Tvorba počítačových 3D modelů segmentů komunikací pro vozidlové simulátory. Praha, 2017. Vedoucí práce doc. Ing. Stanislav Novotný, Ph.D., Ing. Adam Orlický.
2. VÁLEK, DiS. Jan. Základní info o simulátoru k616
3. VLK, František. Dynamika motorových vozidel 2. vyd. Brno, 2003
4. http://homen.vsb.cz/~s1i95/mvd/Moodle/2_4.pdf
5. FRIEDEL, Bc. Daniel. Analýza doby příčného přemístění motocyklů. Brno, 2013. Vedoucí práce Ing. Bc. Marek Semela, Ph.D.
6. https://www.cdv.cz/file/metodika-pro-tvorbu-a-hodnoceni-makroskopickych-dopravnich-modelu/
7. ČEMUSOVÁ, Nikola. Návrh mikrosimulačního modelu chování provozu do vozového simulátoru. Praha, 2019. Vedoucí práce doc. Ing. Stanislav Novotný, Ph.D., Ing. Adam Orlický.
8. https://cs.wikipedia.org/wiki/Omezen%C3%AD_rychlosti_na_pozemn%C3%ADch_komunikac%C3%ADch_v_%C4%8Cesku
9. https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-32351
10. http://radek.jandora.sweb.cz/f01.htm

Seznam obrázků

Obrázek 1 Vozidlový simulátor 10
Obrázek 2 Simulátor FD 616 13
Obrázek 3 Odpor stoupání 20
Obrázek 4 Časový odstup mezi vozidly 28
Obrázek 5 Časové intervaly popisující chování vozidla 30
Obrázek 6 Dálnice 1 31
Obrázek 7 Dálnice 2 32
Obrázek 8 Dálnice 3 32
Obrázek 9 Microsoft Visual Studio 33
Obrázek 10 Detail scriptu "Jizda Kamion" v rozhraní Unity 34
Obrázek 11 Detail scriptu "JizdaOsobak" v rozhraní Unity 35
Obrázek 12 Detail nastavení křivky předjíždění 36
Obrázek 13Detail scriptu "Vpredu" v rozhraní Unity 37
Obrázek 14 Statické triggery 38
Obrázek 15Detail scriptu "Univerzalni Auto ST" v rozhraní Unity 40
Obrázek 16 Dynamické triggery před spuštěním simulace 41
Obrázek 17 Dynamické triggery po spuštění simulace 41
Obrázek 18 Detail scriptu "Univerzalni Auto ST" v rozhraní Unity 43
Obrázek 19 Vytvarování terénu 44
Obrázek 20 Vygenerování lesa 45
Obrázek 21 Výsledný scénář 45
Obrázek 22 Průběh experimentu 46

Zdroje obrázků

1. https://www.autobild.de/bilder/fahrsimulator-bei-daimler-eingeweiht-1267781.html#bild1
2. https://www.cbcsd.cz/udrzitelna-mestska-mobilita/
3. JURÁČEK, Petr. Jízdní odpory vozidel. Brno, 2007. Diplomová práce. Mendelova zemědělská a lesnická univerzita v Brně. Vedoucí práce Ing. Jiří Čupera, Ph.D.
4. ČEMUSOVÁ, Nikola. Návrh mikrosimulačního modelu chování provozu do vozového simulátoru. Praha, 2019. Vedoucí práce doc. Ing. Stanislav Novotný, Ph.D., Ing. Adam Orlický.
5. ČEMUSOVÁ, Nikola. Návrh mikrosimulačního modelu chování provozu do vozového simulátoru. Praha, 2019. Vedoucí práce doc. Ing. Stanislav Novotný, Ph.D., Ing. Adam Orlický.
6. Vlastní obrázek
7. Vlastní obrázek
8. Vlastní obrázek
9. Vlastní obrázek
10. Vlastní obrázek
11. Vlastní obrázek
12. Vlastní obrázek
13. Vlastní obrázek
14. Vlastní obrázek
15. Vlastní obrázek
16. Vlastní obrázek
17. Vlastní obrázek
18. Vlastní obrázek
19. Vlastní obrázek
20. Vlastní obrázek
21. Vlastní obrázek
22. Vlastní obrázek

Seznam příloh

1. JizdaKamion.cs
2. JizdaOsobak.cs
3. Vpredu.cs
4. Vlevo.cs
5. UniverzalniAutoST.cs
6. UniverzalniAutoDT.cs