ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Ústav dopravních prostředků

GENEROVÁNÍ 3D ANIMACÍ DO INTERAKTIVNÍCH VOZIDLOVÝCH SIMULÁTORŮ

Generation of 3D animations for interactive vehicle simulators

Bakalářská práce

Studijní program: Technika a technologie v dopravě a spojích

Studijní obor: Dopravní systémy a technika

Vedoucí práce: Ing. Adam Orlický

.

Rodrigo Vidarte Vargas

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):

Rodrigo Vidarte Vargas

Kód studijního programu a studijní obor studenta:

B 3710 – DOS – Dopravní systémy a technika

Název tématu (česky): **Generování 3D animací do interaktivních vozidlových simulátorů**

Název tématu (anglicky): Generation of 3D animations for interactive vehicle simulators

Zásady pro vypracování

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

- Prostudujte možnosti tvorby animací ve 3D virtuálním prostředí
- Prozkoumejte dostupné metody tvorby animací a provozu pro interaktivní vozidlové simulátory
- Vytvořte metodu automatického generování 3D animací a provozu pro vozidlové simulátory FD ČVUT za pomoci nejvhodnějších nástrojů
- Za pomoci této metody vytvořte scénář pro vozidlový simulátor a následně proveďte experiment pro ověření funkčnosti

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: KERLOW, Isaac Victor. Mistrovství 3D animace Brno: Computer Press, 2011. Mistrovství. ISBN 978-80-251-2717-9.

CULEK, Jakub. Porovnání vhodných grafických enginů pro interaktivní vozidlové simulátory. Praha, 2016. Bakalářská práce. ČVUT FD.

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.

Rodrigo Vidarte Vargas
jméno a podpis studenta

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

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na ČVUT v Praze Fakultě dopravní.

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.

Nemám závažný důvod proti užití 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).

V Děčíně dne 26.08. 2018 Rodrigo Vidarte Vargas

Poděkování

Touto cestou bych rád poděkoval vedoucímu bakalářské práce Ing. Adamovi Orlickému. za odborné vedení, za hodnotné rady a připomínky při psaní této bakalářské práce. Dále bych rád poděkoval všem zúčastněným, kteří se podíleli na analýze a na tvorbě přípravných prací pro analýzu a dalších činnostech.

Abstrakt

Cílem této bakalářské práce je Generování 3D animací do interaktivních vozidlových simulátorů pomocí rychlé a snadné metody automatického vytváření všech nutných souborů k formulaci animace.

Teoretická část je věnována počítačové grafice, jejímu dělení a vysvětlení základních pojmů. Další kapitola se zabývá interaktivními vozidlovými simulátory, jejich druhy, popisem a dále tvořením virtuální reality pro jejich správnou funkčnost. Na tuto kapitolu navazuje popis animací, druhy animací provozu do vozidlových simulátorů a popis chování vozidla během jízdy, který je důležitý pro správnou definici pohybu vozidla během animace.

V praktické části je navázáno na stávající plug-in vytvořením nadstavby, která rozšiřuje tvorbu animací o nové možnosti, jako jsou spuštění těchto animací v závislosti na čase a aktivací pohybem řidiče skrz konkrétní místo ve vozidlovém simulátoru. V práci se nachází rozbor původního algoritmu plug-inu a popis nově vytvořeného algoritmu s vysvětlením principu fungování. V závěru práce je proveden experiment testující funkčnost a kooperaci těchto dvou algoritmů.

Klíčová slova: 3D modelování, vozidlové simulátory, provoz, animace, trigger, Rhinoceros, RhinoScript

Abstract

The subject of this bachelor's thesis is to generate 3D animations into interactive vehicle simulators using fast and easy methods of automatic creation of all necessary files for animation formulation.

The theoretical part is devoted to computer graphics, its division and explanation of basic concepts. The next chapter deals with interactive vehicle simulators, their types, description and creation of virtual reality for their correct functionality. This chapter is followed by a description of the animations, types of animations of traffic to the vehicle simulators and a description of the vehicle behavior while driving, which is important for the correct definition of the movement of the vehicle during animation.

In the practical part the existing plug-in is extended by creation of a superstructure that enables wider options of creating animations with new options, such as running these animations depending on time and activation by driver moving through a specific place in the vehicle simulator. The thesis contains an analysis of the original algorithm of the plug-in and a description of the newly created algorithm with an explanation of the principle of operation. At the end of the work there is an experiment testing the functionality and cooperation of these two algorithms.

Key words: 3D modelling, driving simulators, traffic, animation, trigger, Rhinoceros, Rhinoscript

Obsah

Seznam použitých zkratek 9

1 Úvod 10

TEORETICKÁ ČÁST 12

2 Počítačová grafika 12

2.2 Využití počítačové grafiky 12

2.3 Počítačová grafika v dopravě 13

2.4 Dělení počítačové grafiky 13

2.4.1 2D grafika 13

2.4.2 3D grafika 14

3 Interaktivní vozidlové simulátory 15

3.1 Lehké simulátory 16

3.2 Plnohodnotné simulátory 17

3.3 Interaktivní vozidlový simulátor na ČVÚT v Děčíně 17

4 Tvorba virtuální reality pro interaktivní vozidlový simulátor 19

4.1 Virtuální scéna 19

4.2 Virtuální scénář 23

4.2.1 Animace ve virtuálním scénáři 23

4.2.2 Druhy animací v dopravních simulátorech 23

4.2.3 Automatická tvorba provozu ve scénářích pro vozidlové simulátory 24

4.3 Import do simulátoru na ČVÚT 25

5 Pohyb vozidel v simulacích 29

5.1 Popis pohybu vozidla 29

5.2 Dynamika jízdy vozidla 29

5.3 Pohyb vozidla při animaci 31

PRAKTICKÁ ČÁST 32

6 Plug-in pro generování provozu pomocí triggeru 32

6.1 Tvorba vrstvy ..... **Chyba! Záložka není definována.**

Pseudokód *33*

6.2 Vykreslení rychlostní křivky 33

Pseudokód *36*

6.3 Vytvoření složky s animačním souborem 38

Pseudokód *39*

7 Experiment 41

8 Závěr ..... **Chyba! Záložka není definována.**

9 Použité zdroje 45

10 Seznam obrázků 47

11 Seznam příloh 48

Seznam použitých zkratek

2D	Dvourozměrný
3D	Trojrozměrný
ČVÚT	České vysoké učení technické
LCD	Liquid crystal display
NURBS	Non-uniform rational basis spline
TIN	Triangulated irregular network
HTML	Hyper text markup language

1 Úvod

Grafika je odvětvím výtvarného umění. Obecně jako grafiku můžeme označit jakékoli umělecké dílo, které umělec vytvořil jakoukoli grafickou technikou. Své dílo poté podepíše. Od této chvíle se z vytvořeného díla stává originál.

První grafické tisky se objevily již v 6. Století v Číně. Použitou technikou v tomto případě byl dřevořez, jehož pomocí se vytvářely náboženské amulety. První kompletní dochovaná kniha, která byla vytištěna pomocí této techniky, je již z roku 868. Postupem času však technika dřevořezu nebo dřevorytu přestala vyhovovat. Umělci hledali materiál, který by byl schopný přesněji přenášet jemnější tvary. Zde se začaly vyskytovat první náznaky tzv. tisku z hloubky. Umělci náměty vyrývali do ocelových a měděných destiček. Mezi další grafické techniky, které v této době začaly vznikat patří například tisk z výšky a tisk z plochy. Dále existují techniky mimo tyto skupiny jedná se především o techniky jako sítotisk, monotyp a papírotisk.

Počátky počítačové grafiky se datují již v 15. století, kdy v Evropě vznikaly a dále se rozvíjely první tiskařské technologie. Postupem času se tento obor zdokonaloval, ale vždy navazoval na předchozí poznatky a dodnes nám zanechal jeden z nejdůležitějších obsahů používaným každým textovým editorem, písmo. Některé fonty, které do dnes používáme, se téměř od 18. století nezměnily a jsou stále pojmenovány po svých tvůrcích.

Dvacáté století však nastartovalo největší rozvoj grafiky v historii lidstva. První počítače nebyly moc výkonné, takže první grafické výtvory nedosahovaly příliš vysoké kvality. Postupem času se však počítače zdokonalily. V dnešní době je již počítačové grafika na tak vysoké úrovni, že za použití dostatečně výkonných nástrojů dokážeme vytvořit iluzi reality nebo dokonce přivést mrtvé herce zpět k životu.

Virtuální realita existuje již řadu let, ale nyní jsou již i osobní počítače dostatečně výkonné, že pro běžné lidi stačí pouze zakoupit si 3D brýle, které připojí k počítači nebo k herní konzoli, a mohou si užívat virtuální reality u sebe v obývacím pokoji.

Virtuální realita a virtuální prostředí mají velmi široké využití. Zábavní průmysl je jednoznačně tahounem počítačové grafiky. Jednotlivý tvůrci her se neustále předhání v tom, který z nich vytvoří hru s nejlepší grafikou. A akční, sci-fi nebo fantas filmy bez speciálních CGI efektů si už dnes ani nejsme schopni představit.

Virtuální prostředí a počítačovou grafiku je také možné využít v dopravních simulátorech

při simulacích a experimentech. Virtuální realitu je také možné použít při výuce, může se jednat například o letecké (ale i jiné) simulátory. Některé simulátory však nemusí sloužit k výuce, ale k předvádění nových automobilů nebo třeba nových úseků pozemních komunikací. Zákazníci a manageři nemusí vždy úplně dokonale rozumět technickým dokumentacím, ale určitě ocení, když jim nový automobil představím jako 3D model ve virtuální realitě.

TEORETICKÁ ČÁST

2 Počítačová grafika

Počítačová grafika je z technického hlediska oborem výpočetní techniky, který se rozvíjí velmi rychle. Bylo tomu teprve nedávno, kdy nebylo možné na osobním počítači editovat v přijatelném grafickém režimu obrázek. V minulosti to bylo možné pouze v rámci grafických, televizních a reklamních studií se „superpočítači“.

Počátky počítačové grafiky se datují již v 15. století, kdy vznikaly a dále se rozvíjely první tiskařské technologie. Postupem času se tento obor zdokonaloval, ale vždy navazoval na předchozí poznatky a dodnes nám zanechal jeden z nejdůležitějších obsahů používaným každým textovým editorem, písmo. Některé fonty, které do dnes používáme, se téměř od 18. století nezměnily a jsou stále pojmenovány po svých tvůrcích.

Počítače v současnosti jsou již pokrokovější a obsahují výkonné procesory s dostatečným množstvím operační paměti, které umožňují, aby každý uživatel osobního počítače mohl pomocí příslušného softwaru vytvářet celé virtuální světy a jiné grafické zázraky. V posledních letech jsou také programy pro práci s grafikou dostupnější. Není-li uživatel příliš náročný a nevyžaduje-li špičkové nástroje, dokonce i poloprofesionální software, který pro domácí použití zcela dostačuje, je možné zakoupit již za řádově tisíce korun.

2.2 Využití počítačové grafiky

Počítačová grafika má velmi široké využití, a to hned v mnoha oborech. Jedním z prvních příkladů jsou tiskoviny, kam spadají prakticky veškeré časopisy, noviny, knihy apod. Nedílnou součástí novin a časopisů jsou také reklamy, které využívají grafiku na každém kroku. Dalším příkladem jsou multimediální CD, televizní efekty, titulky, grafické obrázky, schémata a grafy ve večerních zprávách, jenž se používají v médiích a multimédiích. Další velkou kapitolou je prostorové modelování, to umožňuje vytvářet nové objekty a následně i celé nové světy. Pod tím si můžeme představit vytváření nových výrobků, které lze vidět dříve, než jsou vyrobeny. Jako například nové modely automobilů, či návrhy interiérů. Jedno ze specializovaných odvětvích počítačové grafiky pracující na odlišném principu, než ta výše zmíněná se nazývá CAD/CAM projektování. Díky tomuto počítačovému projektování lze například konstruovat celé budovy a poté vytvořit jejich prostorové scény tak, aby projektant vše viděl a mohl libovolně do projektu zasahovat. Za zmínku určitě stojí počítačová grafika v zábavním průmyslu, jedná se samozřejmě o hry, ať už na počítačích, či jiných herních platformách. Kromě těchto uvedených odvětví existuje samozřejmě ještě mnoho dalších oblastí, kde se můžeme setkat s počítačovou grafikou. [6]

2.3 Počítačová grafika v dopravě

V dopravě se můžeme s počítačovou grafikou setkat u návrhu řešení infrastruktury, kam také neodmyslitelně patří hledání vhodných alternativ či analýza nehod. Zařadit sem můžeme dopravní značky, billboardy apod., které potkáváme při řízení automobilu a jsou běžnou součástí provozu.

Velkou roli hraje počítačová grafika používaná v dopravních simulátorech při návrhu virtuální scény, vytváření scénářů a návrhu kokpitu. Realističnost a kvalita těchto prvků musí být co nejvyšší, abychom získali správné a použitelné výsledky experimentů.

2.4 Dělení počítačové grafiky

Počítačovou grafiku lze rozdělit do několika kategorií podle různých kritérií. Z hlediska prostorového vnímaní můžeme počítačovou grafiku rozdělit na 2D (dvourozměrná grafika) a 3D (trojrozměrná grafika). Existují také 1D a 4D, ale tyto dva typy nemají při vytváření virtuální reality vozidlového simulátoru význam, proto v této práci nebudou rozebírány do hloubky.

2.4.1 2D grafika

Jak už napovídá název, dvourozměrná grafika je definována dvěma rozměry. Dvourozměrná grafika pracuje hlavně s obrázky a fotografiemi, dělí se na grafiku rastrovou (bitmapovou) a vektorovou. Viditelný rozdíl, po přiblížení (8x) obrázku fotoaparátu, mezi grafikou rastrovou a vektorovou je znázorněn na obrázku (Obrázek 1).

Obrázek 1: Příklad rastrové a vektorové grafiky

- **Rastrová grafika** – Slovo raster se dá volně přeložit jako síť. V tomto případě je to síť pixelů (drobné čtverce), které když se poskládají do mřížky vytvoří obraz. Pixel nese informaci o jeho jasu, barvě, poloze a kontrastu. Čím více jich je, tím víc je obraz kvalitnější, kvalitu obrazu nazýváme rozlišení. Hlavní nevýhodou je ztráta kvality při zvětšování obrázků.
- **Vektorová grafika –** Vektorová grafika vykresluje obraz úplně jiným způsobem než rastrová. Neukládá body do mřížek, ale obrázky skládá z geometrických tvarů (body, kruhy, křivky, mnohoúhelníky). Jejich vzájemná vzdálenost je vypočítávána na základě rovnic, díky tomu neztrácejí kvalitu obrazu, při deformování, či při zvětšování. Proto je taky tato grafika využívána u počítačového modelování a projektování. Běžnou fotografii však nelze uložit do programů využívající vektorovou grafiku.

2.4.2 3D grafika

3D počítačová grafika pracuje s trojrozměrnými objekty, liší se tedy od 2D grafiky tím, že se k ní přidá ještě jeden rozměr, nejčastěji označován jako hloubka tělesa. Převod 3D objektů do 2D zobrazení se nazývá renderování. Jeden z nejznámějších využití počítačové 3D grafiky je vytváření animací, a to jak pro tvorbu filmů, počítačových her, tak pro počítačové simulace.

Samotné modelování prostorového tělesa je reprezentováno několika způsoby. Každé z nich pracuje s odlišnými principy. Podle těchto principů si vybíráme ze tří základních reprezentací. Tyto tři základní druhy reprezentací se nazývají objemová reprezentace, hraniční reprezentace a procedurální reprezentace. [8]

- **Hraniční reprezentace** – Pomocí 3D modelovacího softwaru tvoříme tzv. obálku či „skořápku“ objektu. To znamená, že jsme vytvořili viditelnou hranici mezi objektem a okolním prostorem. Tato hranice, respektive povrch tělesa se popisuje množinou hraničních geometrických prvků (pomocí bodů, úseček, částí rovin, křivek či ploch). Hraniční reprezentace je nejběžnější, protože je pro lidi nejvíc přirozená. Při kreslení nejčastěji kreslíme objekty pomocí obrysů (=podmnožina hranice).

  S hraniční reprezentací pracuje většina 3D modelovacích programů, mnoho jich však využívá různé metody popisu povrchu objektů. Mezi nejrozšířenější metody pro tvoření scénářů do vozidlových simulátorů patří metody TIN (triangulated irregular network = nepravidelná síť trojúhelníků), NURBS (non-uniform rational B-spline = neuniformní racionální B-spline) a T-Spline (rozšíření NURBS).

- **Procedurální reprezentace** – V procedurální reprezentaci jsou tělesa definovány algoritmy. Uživatel je nucen upřesnit několik parametrů a program následně automaticky vytvoří objekt. Výsledné tvary mohou být velice komplexní díky fraktálům, které lze volně použit při generování daného objektu.
- **Objemová reprezentace** – Objemová reprezentace prostorových objektů pracuje s informací, zda je určitý bod 3D prostoru součástí daného objektu či nikoliv. Kromě povrchu tedy popisuje i objem tělesa. Objem je popisován obsazením jednotlivých prostorových částí. Čím více je tato síť prostoru hustá, tím je těleso popsáno kvalitněji (podobně jako u rastrové grafiky). Tato reprezentace se používá nejčastěji v technických oborech případně v medicíně, protože je až příliš složitá a ve většině případů postačí pouze reprezentace hraniční. Nejznámější zástupci této reprezentace jsou BSP strom, buněčný strom, oktanový strom. [8]

3 Interaktivní vozidlové simulátory

Interaktivní vozidlový simulátor je zařízení, jehož podstatou je co nejpřesněji simulovat řízení reálného automobilu. Proto je jednou z nejlepších cest, jak zkoumat chování řidiče. Chování je následně vyhodnocováno, je-li k dané situace adekvátní či nikoliv. Řidič ovládá vozidlo v simulátoru a pohybuje se ve virtuální scéně bez jakýchkoliv skutečných následků. Právě proto jsou vozidlové simulátory vhodné k nejrůznějším pokusům, které by měly v reálném světě fatální následky. Nejdůležitějšími součástmi interaktivního vozidlového simulátoru jsou virtuální scéna a virtuální scénář, kterým se věnuji v kapitole tvorba scénářů.

Nejvíce používaný typ vozidlového simulátoru je simulátor s projekcí typu CAVE. V tomto simulátoru je vozidlo, nebo jeho část, obklopeno velkoplošnými LCD obrazovkami, případně promítacími plátny, která jsou sestavena do tvaru kvádru nebo krychle, kam se řidiči promítá virtuální scéna. Jak už bylo zmíněno výše, cílem je tedy vytvořit takové virtuální prostředí, aby řidič nabýval pocitu, že se nachází v reálném provozu. Toho můžeme dosáhnout kromě virtuální scény také pohybovou plošinou a zvukovými efekty. Pohybová plošina simuluje dynamiku pohybu vozidla (např. akceleraci), ale také naklánění vozidla.

Obrázek 2: Simulátor typu CAVE

Jedním ze základních prvků simulátoru je kokpit, nebo i vozidlo samotné, které je se simulačním systémem propojeno oboustranně. Do výpočetní jednotky simulátoru jsou odesílány data s informacemi o pohybu volantu a pedálů, také informace o rychlosti a zařazení rychlostního stupně. Zpět do kokpitu to jsou pak informace o rychlosti, otáčkách či nastavení vlastností volantu (pružnost, tuhost, moment). K softwarovému vybavení simulátoru patří řízení běhu scén, generátor událostí, simulované prostředí virtuální reality, řízení pohybu plošiny, záznam jízdy, nástroje na vyhodnocení řidičova výkonu a matematický model chování vozidla.

Z konstrukčního hlediska můžeme interaktivní vozidlové simulátory rozdělit na tzv. lehké a plnohodnotné.

3.1 Lehké simulátory

Součást lehkého simulátoru může tvořit kabina vozidla nebo její část. Virtuální scéna bývá promítána na monitorech před vozidlem či na promítacích plátnech před vozidlem. Mezi výhody lehkého simulátoru patří jednoduchá operativnost s jednotlivými částmi, dá se tedy lehce přestavět a během chvíle přizpůsobit požadavkům konkrétního experimentu nebo i dovybavit přídavnými zařízeními. K dalším výhodám se také řadí cena simulátoru, která je výrazně nižší než u simulátoru plnohodnotného. K nevýhodám patří fakt, že se nejedná o celé vozidlo, proto se zde může řidič cítit nepřirozeně a nabývat dojmu, že není v reálném provozu, což se může negativně odrazit na výsledcích měření. Proto je nutné si před výběrem simulátoru uvědomit, že výsledky mohou být nepřesné, a dobře si rozmyslet, jaký typ experimentu provádíme a jaké relevantní informace chceme získat.

3.2 Plnohodnotné simulátory

Na rozdíl od lehkého simulátoru je u plnohodnotného simulátoru využito celého vozidla, kde v rámci experimentu řidič sedí v kokpitu a cítí se jako za normálního provozu v reálném vozidle. Tím se tedy opět přibližujeme co nejblíže reálné jízdě. Projektování virtuální scény probíhá na projekčních plátnech, která jsou u tohoto typu simulátoru situována před vozidlem, ale i po jeho stranách a také i zezadu. Řidič v kokpitu vnímá virtuální scénu zepředu, zezadu i zboku a je tedy opět o něco blíže reálné jízdě. Nevýhodou plnohodnotného simulátoru však je to, že se nedá lehce přestavět nebo přizpůsobit jinému experimentu.

3.3 Interaktivní vozidlový simulátor na ČVÚT v Děčíně

Jeden z projektů na fakultě dopravní ČVUT s názvem „Simulace a vizualizace“ se bude zbývat sestrojením lehkého simulátoru v prostorách pracoviště Děčín. Mělo by se jednat právě o simulátor typu CAVE, který je zmíněn v úvodním odstavci o interaktivních vozidlových simulátorech.

Je plánováno sestavit pohyblivý simulátor spolu s třemi FullHD projektory a jejich projekcí na promítacích plátnech. Simulátor by měl využívat lineárního pojezdu a simulovat síly, které za normální jízdy působí na řidiče. Samotná kabina simulátoru se plánuje uchytit na platformu s 6 stupni volnosti (6LOF).

Obrázek 3: Vizualizace vozidlového simulátoru na FD v Děčíně (laboratoř)

Obrázek 4: Vizualizace vozidlového simulátoru na FD v Děčíně (kokpit)

4 Tvorba virtuální reality pro interaktivní vozidlový simulátor

4.1 Virtuální scéna

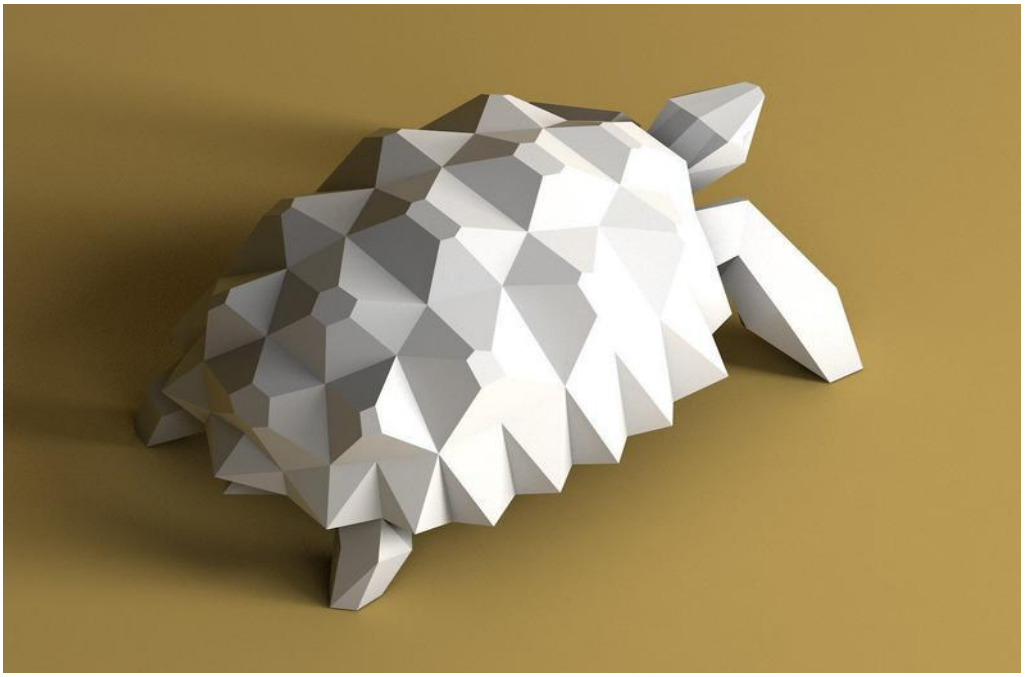
Jeden ze stěžejních prvků interaktivního vozidlového simulátoru je virtuální scéna. A to právě proto, že samotný simulátor bez virtuální scény nemá příliš smysluplné využití. Bavíme se tedy o prostředí, které je vymodelováno pomocí 3D počítačové grafiky a ve kterém se řidič i s kokpitem pohybuje. Samozřejmě opět platí, že vymodelované 3D prostředí se musí co nejvíce podobat reálnému prostředí, abychom získali co nejpřesnější informace a data. Abychom dosáhli ještě větší realističnosti a aby se i scéna dostatečně oživila, můžeme přidat animace s různými modely (např. automobily, chodce nebo i divokou zvěř).

Obrázek 5: Příklad virtuální scény

Požadavky 3D objektů do virtuální scény

Virtuální scéna se tvoří pomocí 3D objektů. Jedním z důležitých faktorů 3D modelů ve virtuálních realitách je jejich dostatečně rychlé vykreslování a zároveň realistická podoba. To znamená, že musí existovat určitý kompromis mezi kvalitou a náročností 3D objektů.

U většiny simulačních programů se nejvíce tomuto kompromisu blíží definice objektů známá jako TIN reprezentace. U této reprezentace je objekt složen a definován z co nejméně trojúhelníků. Snížení výsledné náročností na hardware lze dosáhnout jednotnou texturou u více objektů. Objekty, které obsahují nízký počet trojúhelníků, se nazývají „low poly objekty“.



Obrázek 6: Low poly object [11]

Tvorba virtuální scény

Abychom vytvořili adekvátní scénu pro interaktivní vozidlové simulátory, potřebujeme ve 3D prostředí vytvořit vhodnou silniční infrastrukturu. Na vytvoření silniční infrastruktury lze použít dvě rozdílné metody.

Snadnější alternativou je metoda za použití softwarů specializovaných na vytváření silniční infrastruktury. V těchto softwarech se navrhuje a plánuje nové silniční komunikace v reálných lokalitách po světě, proto mají i funkce pro detailní vykreslení a tvorbu projektové dokumentace. Programy jsou často velice sofistikované, což se odráží mimo jiné na jejich vysokých cenách. Za nevýhodu se dá považovat neschopnost exportovat do formátů používaných pro vozidlové simulátory.

Složitější alternativou je tvorba scén pomocí 3D modelovacích programů nedisponujících funkcemi pro generování silničních komunikací. Některé tyto programy využívají pár základních funkcí, což ale není dostačující, protože neumožňují tvořit silniční komunikace dle norem. Jednou z výhod je například bezproblémový export modelů.

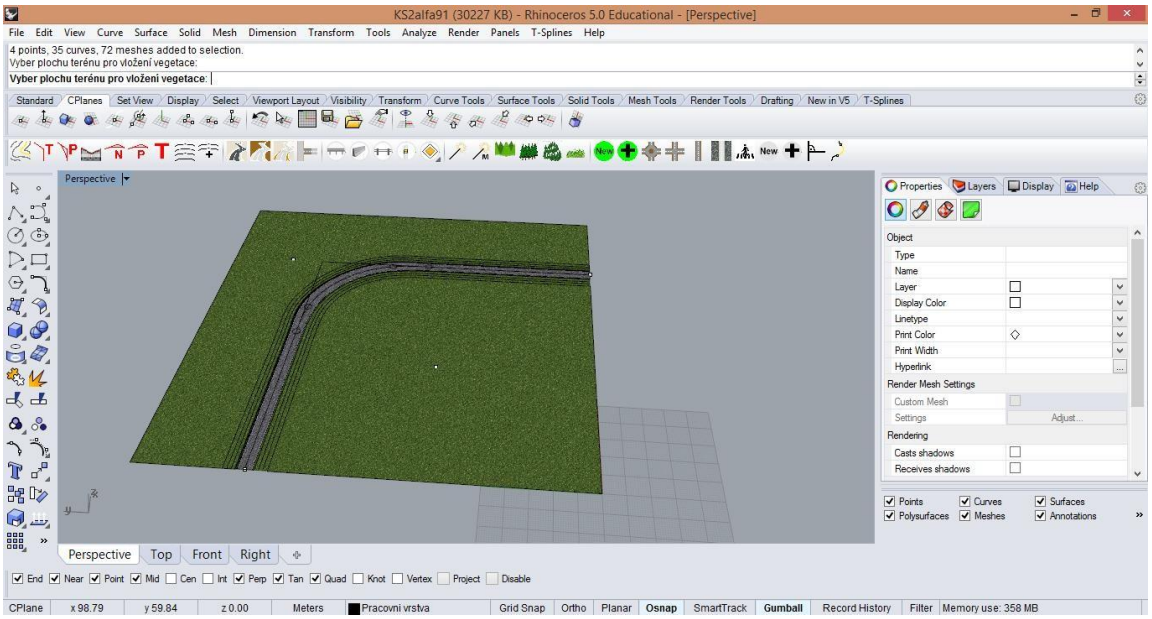
Řada moderních programů má dnes možnost rozšířit své funkce díky zásuvným modulům, též známým jako „plug-in“. Zásuvné modely se dají vytvořit ve speciálních vývojářských programech (např. Visual Studio), některé programy však mají vlastní nástroje pro skriptování, díky kterým si mohou vytvářet vlastní plug-iny. Plug-iny sice nefungují samostatně, ale v součinnosti s hlavním programem a samotná syntax skriptů je uživatelsky celkem jednoduchá. Jakmile naskriptujeme vlastní funkce na tvoření silniční infrastruktury dle norem, můžeme ho již vložit do 3D modelovacího programu pro který byl vytvořen.

K nejznámějším a nejpoužívanějším 3D modelovacím programům, ve kterých můžeme vytvářet zásuvné modely za pomocí skriptů, patří: CINEMA 4D, Rhinoceros V6, Blender, AutoCAD 3DS Max a Sketchup. Samotné programy se pak od sebe liší především množstvím, funkcí a cenou.

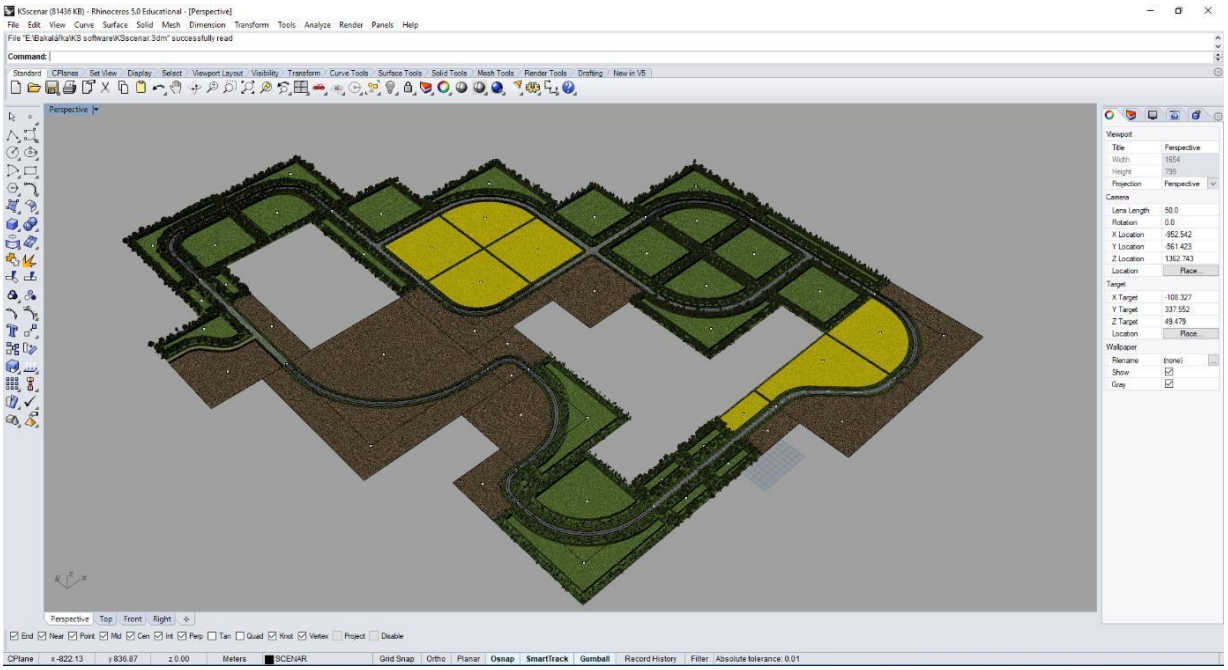
Jeden z plug-inů do 3D modelovacího programu Rhinoceros, byl vytvořen na ČVÚT v rámci bakalářské práce „Tvorba počítačových 3D modelů segmentů komunikací pro vozidlové simulátory“.

Plug-in pro generování segmentů komunikací

Tento plug-in usnadňuje vytváření scénářů. Jeho princip spočívá ve tvoření scén skládajících se ze silničních komunikací v extravilánu a okolní vegetace. Plug-in využívá předpřipravených segmentů komunikací, tzv. dlaždic. Uživatel tyto dlaždice může skládat k sobě stejným principem jako puzzle. Tímto způsobem může uživatel jednoduše generovat silniční infrastrukturu v libovolné velikosti i s okolní vegetací. K dispozici je zde i databáze obsahující základní dlaždice, které jsou předem vymodelovány. Aby tato metoda fungovala správně, musí mít všechny dlaždice přesně dané rozměry a správně definované výstupy a vstupy silničních komunikací. Pokud jsou tyto podmínky dodrženy, uživatel docílí přesné návaznosti mezi jednotlivými segmenty. Mezi jednoznačné výhody tohoto plug-inu patří rychlost, s jakou může uživatel scény tvořit a ušetřený čas pak věnovat jiným detailům, které vyžaduje konkrétní experiment. K správnému používání plug-inu není ani zapotřebí velká znalost 3D modelování. Protože je uživatelsky velice přívětivý, tvorbu scény zvládne i začátečník, což se také řadí k velkým výhodám. [3]



Obrázek 7: Ukázka základní dlaždice [3]



Obrázek 8: Propojení dlaždic v celou scénu [3]

Plug-in RoadCreator for Rhino [1]

Tento plug-in byl vytvořen v bakalářské práci Ing. Adama Orlického . Je vytvořen tak, aby bylo možné funkcemi generovat silniční infrastrukturu dle českých norem. Vyskytují se zde funkce, které zjednodušují dotváření scén, především funkce pro dotváření krajiny. Program Rhinoceros pracuje na principu vrstev, což je v tomto plug-inu využito.

Jednou z významných funkcí plug-inu RoadCreator je Vrstevnice. Vrstevnice jsou křivky, které spojují místa v terénu se stejnou nadmořskou výškou. Díky zobrazení vrstevnic má uživatel významně usnadněný návrh směrového řešení komunikace v terénu.

Kromě této funkce plug-in disponuje dalšími dílčími funkcemi jako jsou: přechodnice, podélný profil, tvorba nivelety, parabolický oblouk výškového vedení trati, a nakonec i samotné vytvoření trasy v prostoru.

Jakmile jsou pomocí plug-inu vytvořeny křivky trasy a příčné profily, je možné začít vytvářet 3D modely silnic. V této fázi plug-in opět disponuje dalšími funkcemi, které jsou určené pro generování 3D modelů silniční infrastruktury a zakomponování silnic do terénu.

Plug-in kromě těchto funkcí obsahuje i doplňkové funkce. Uživatel má možnost obohatit silniční infrastrukturu vytvořeného modelu o různé druhy bezpečnostních zařízení nebo osazovat silnice dopravními značkami.

Všechny tyto funkce dělají z tohoto plug-inu ideálního pomocníka u tvorby virtuálních scén pro interaktivní vozidlové simulátory.

4.2 Virtuální scénář

Virtuální scénář se dá definovat jako souhrn událostí v dané simulaci, na které je řidič nucen při jízdě reagovat. V těchto událostech je zahrnuta definice trasy, po které se řidič s vozidlem pohybuje, definice libovolných překážek na trase (např. železniční přejezdy, přechody pro chodce, zúžení), definice dynamických objektů (např. animované billboardy, světelné a svislé dopravní značení) a simulované události (např. provoz vozidel). Tyto zmíněné události napomáhají dotvářet scénu. Stejně tak mohou být zahrnuté zde i animace pro potřeby různých experimentů a oživit tak statickou virtuální scénu.

4.2.1 Animace ve virtuálním scénáři

V dnešní době se tvorba animací rozvíjí velice rychle, a proto se s nimi setkáváme stále častěji a častěji. Mnoho animací můžeme dnes vidět v počítačových hrách nebo animovaných filmech. Vývoj má za příčinu také neustálý zájem a zvyšující se nároky diváků-hráčů, kteří kladou čím dál tím větší důraz na co nepřirozenější pohyb daného objektu.

Jestliže chceme dosáhnout dokonalé animace, je nutné si už v počáteční fázi uvědomit, jak by měla naše animace vypadat na konci. Klíčové jsou zde jednotlivé snímky, ze kterých je animace složena. Budeme-li animovat lidské tělo, je nutné znát jeho pohybový aparát. Musíme prostudovat jeho kostru, kde se nachází svaly nebo jak se pohybují klouby. Taktéž to funguje u tvorby animací do vozidlových simulátorů. Vytvořené animace vozidel by měli odpovídat fyzikálním zákonům a vozidla by se měla pohybovat stejně jako v reálném světě.

4.2.2 Druhy animací v dopravních simulátorech

V dopravních simulátorech je jednou z priorit, aby se řidič při jízdě cítil jako v reálném prostředí, proto zde animace hrají velice důležitou roli. Scénu lze dále doplnit o animace pohybu mraků, pohyb chodců nebo pohyby ostatních vozidel. Tato vozidla pak tvoří provoz ve virtuálním scénáři. Pro simulace ve vozidlovém simulátoru je provoz velice důležitý prvek.

Při tvoření těchto animací se dá provoz rozdělit podle druhu chování na tři druhy, na interaktivní, neinteraktivní a autonomní.

Provoz interaktivní

U tohoto typu provozu je pohyb vozidel řízen složitějšími algoritmy, které spolupracují s podněty okolí a následně jim pak přizpůsobují chování a jízdu. Vozidla mají předem danou trasu po křivce. Když se však tato křivka střetne s trajektorií jízdy řidiče simulátoru, přizpůsobí se dané situaci. Pokud řidič v simulátoru zpomalí, s naprogramovaným vozidlem těsně za ním nedojde ke střetu, ani neprojede skrz něj, ale bude udržovat vzdálenost a zpomalí taky.

Provoz neinteraktivní

U provozu neinteraktivního má každé vozidlo v provozu předem danou přesně určenou

trasu a parametr jízdy. Na rozdíl od provozu interaktivního vozidla zde nereagují na okolí a ani se nemění v čase.

Trasa vozidla je daná křivkou, po které jede. V každém bodě, ze kterého je tato křivka složena, jsou zadány konkrétní parametry. Parametry udávají rychlost a pohyb vozidla. Podrobněji se jedná o tyto parametry: čas (udáván s milisekundách, ve kterém bodě se vozidlo nachází), souřadnice polohy (osy x, y, z určují polohu vozidla), souřadnice natočení (úhel natočení v osách x, y a z, které určují klonění, klopení a stáčená vozidla).

Tento typ simulace provozu se nejčastěji v simulátorech využívá jako oživení okolí, například pro vozidla jedoucí po jiné silnici než řidič simulátorů. A to z důvodu, aby nedocházelo ke střetu, na který tyto simulace neumí reagovat.

Provoz autonomní

Autonomní režim simulace je specifický svou vlastní umělou inteligencí. Je tedy již z názvu jasné, že se vozidlo rozhoduje zcela samo, kudy a jak pojede. Trasa není popsána křivkou, jako tomu bylo u dvou předešlých simulací, ale vozidlo si vše určuje samo. Vozidlo dokáže rozpoznat silnici podle materiálu nebo barvy a zvolí si tedy i jízdní pruh samo. Jakmile se vozidlo ocitne na křižovatce, je opět na umělé inteligenci se rozhodnout, kterým směrem se vydá, k čemuž ji napomáhají samoučící algoritmy. Autonomnost je natolik vysoká, že vozidla dokáží samy reagovat na řidiče v simulátoru, jako tomu je při simulaci interaktivní.

Částicové systémy

Tyto specifické animace napomáhají v dopravních simulátorech k větší realističnosti, a to jak okolím (sněhová bouře, mlha atd.), tak i efekty jízdy (např. prach při akceleraci nebo při průjezdu pískem).

Částicové systémy jsou určeny pro popis objektů, jejichž tvary jsou příliš složité, členité, nebo se mění takovým způsobem, že je nelze definovat jako povrch.

Nejčastější si pod tímto pojmem lze představit atmosférické efekty jako je kouř, sníh, mlha, déšť ale i tvary, které se v čase mění a nelze je tedy definovat jedním tvarem. Částicové systémy jsou tedy tvořeny větším množstvím malých částí, které se chovají jako jeden objekt. Částice vznikají a zanikají v čase, nejsou statické.

4.2.3 Automatická tvorba provozu ve scénářích pro vozidlové simulátory

Automatickou tvorbu provozu ve scénářích se na ČVÚT zabývala studentka ve své bakalářské práci a vytvořila i plug-in do 3D modelovacího programu Rhinoceros ve skriptovacím jazyce Rhinoscript. Plug-in byl rozdělen do čtyř hlavních funkcí – výběr trasy, vykreslení rychlostní křivky, vytvoření animačního souboru.

V této práci bylo dbáno na důležitost realističnosti scénářů a scén, proto lze s tímto plug-inem tvořit kompletní statickou virtuální scénu použitelnou pro nejrůznější experimenty v interaktivním vozidlovém simulátoru. Aby realističnost experimentů pro řidiče v kokpitu simulátoru byla opravdu co nejvyšší, plug-in doplňuje scénu o animace jízdy ostatních vozidel. Plug-in generuje provoz v co nejkratším čase a uživatelsky je celkem jednoduchá a intuitivní (znalost 3D modelování a exportování je opravdu začátečnická).

Mezi jediné nevýhody patří fakt, že tento plug-in navazoval na další plug-in tvořící silniční infrastrukturu pomocí segmentů (dlaždic), který jsem podrobně rozebral v kapitole 4.2.1. Plug-in pro generování segmentů komunikací a jsou tedy provázány navzájem. Tvorba provozu je závislá na křivkách provozu kopírující jízdní pruhy, které jsou doplněny pod každým segmentem. Tyto křivky definují trasu, po které se animace vozidel pohybuje a křivky musí být také navzájem propojené, aby vznikla výsledná trasa jízdy animovaného vozidla. [5]

Já jsem se rozhodl v této práci tento nedostatek vyřešit, aby nebyl plug-in závislý na segmentech a aby byla možnost tvořit provoz vozidel i na tzv. trigger, který otevírá tomuto plug-inu zcela nové možnosti experimentů.

Trigger

Trigger (česky spoušť) je místo definované v prostoru, které při aktivaci spouští určitou činnost. V praxi to pak ve virtuální scéně znamená, že některým animacím ve vozidlových simulátorech trigger definuje souřadnice spuštění – např. můžeme umístit trigger před železniční přejezd a když do něj řidič v simulátoru vjede, spustí se animace závor.

4.3 Import do simulátoru na ČVÚT

Na Fakultě dopravní ČVÚT v Praze je dostupný simulátor, který pracuje se speciálně vyvinutým 3D grafickým enginem (vizualizační software). U toho enginu je ale velice odlišné importování modelů, než u enginů stávajících a volně dostupných.

Samotný průběh tvorby scénáře provádíme takto: za prvé si vymodelujeme scénu v kterémkoliv dostupném 3D modelovacím softwaru. Dbáme na to, aby se menší modely, ze kterých se skládá vytvořený model scény, se svými normálami ploch nacházely ve správném směru a náležitě se vykreslovaly. Je také nutné, aby veškeré textury byly ve formátu dds (Microsoft Direct Draw Surface – formát grafiky), protože již zmíněný engine na ČVÚT jiné formáty nepodporuje.

Dalším krokem po vytvoření modelu je vytvořit kolizní a vizualizační soubory. Tyto soubory jsou definovány ve formátech SNT A PBT. SNT soubor je textový soubor nesoucí v sobě

informaci o výčtu bodů, ke kterým pak následně přiřazuje trojúhelníky. Jeho funkcí je popsat kolizní objekty, tedy objekty, které řidič nemůže fyzicky projet a hrozí tedy naražení auta (např. povrch silnice, pevné objekty kolem silnice). O grafickou (vizualizační) část se stará soubor PBT, který je textovým souborem obsahujícím texturované UV koordináty textur.

Následuje tvorba dvou adresářů. První adresář pojmenujeme názvem Data a zahrneme zde tři složky: Models (zkomprimované soubory PBTz), Textures (textury s formátem dds) a Animations (soubor Anim). Tento adresář zahrnuje taktéž textový soubor Models (názvy objektů), Textures (názvy textur) a World (fixace a vizuální chování – jas,mlha). Druhý adresář má jméno Mesh a obsahuje soubor SNT.

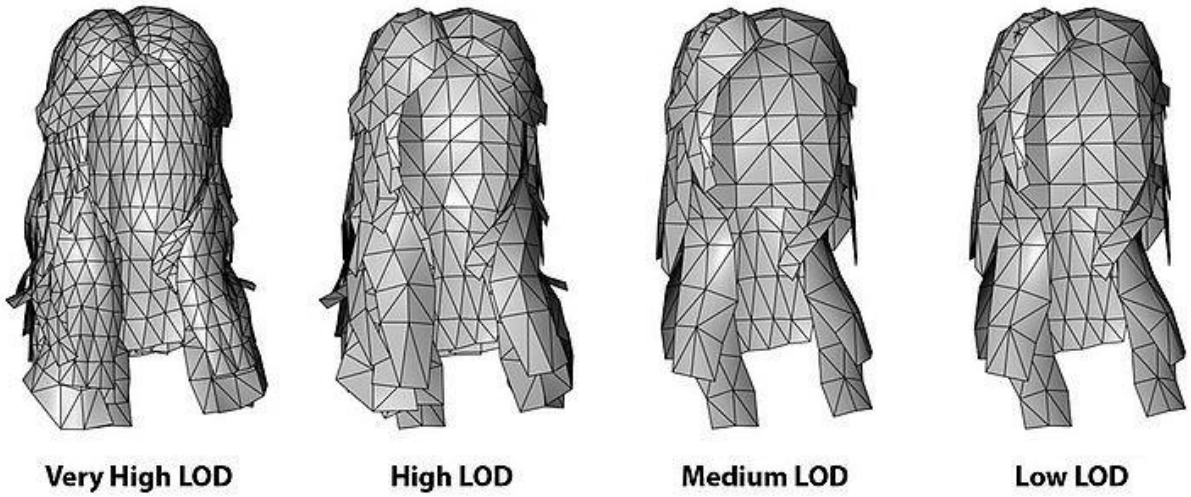
Jakmile máme tyto dva adresáře vytvořené, nahrají se do počítačů simulátorů. Do počítačů s vizualizacemi vložíme adresář Data a druhý adresář Mesh do počítače s fyzikou. Jedním z hlavních nevýhod enginu používaným na ČVÚT je fakt, že si neumí sám generovat stíny objektů, a to i když je postup importu zřetelně zdlouhavý a složitý.

Existují však jiné, modernější enginy, jako je například Unreal Engine nebo Unity, kde je postup viditelně snazší. Nejprve je model prostředí vytvořen v 3D modelovacím softwaru (3DS Max, Rhinoceros apod.), s možností uložení v několika formátech (dae, obj, 3ds, fbx). Také textury mají možnost být uloženy v různých formátech a to tga, dds, jpeg. Jakmile je engine spuštěn, jsou nahrány všechny textury a soubory. Samotný model se rozdělí na jednotlivé objekty vytvořené dříve ve 3D modelovacím programu a všem objektům je pak možné definovat, zdali jsou kolizní či nikoliv. Mezi výhody modernějších enginů také patří možnost úpravy materiálu objektu.

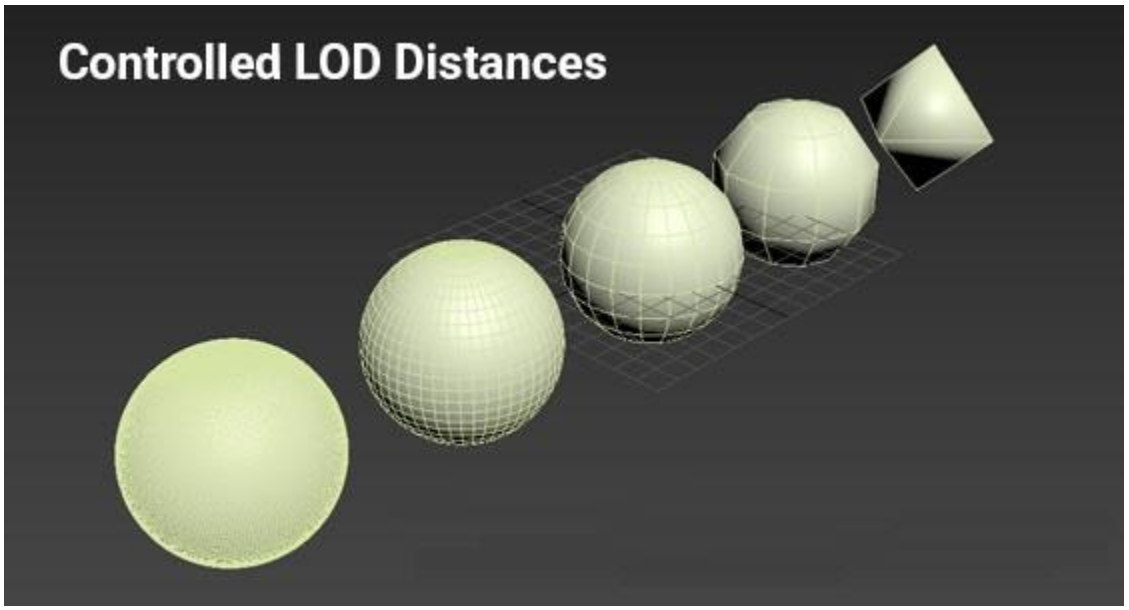
Mezi další výhody zmíněných enginů patří neodmyslitelně také množství integrovaných funkcí. Je zde možnost používat částicové modelování pro tvorbu kouře, stíny jsou generovány automaticky v závislosti na osvětlení a lze vcelku uživatelsky jednoduše tvořit různé globální prostředí díky skyboxu (šestistěnná krychle potažená texturou, tvořící krajinu nebo oblohu). [7]

Obrázek 9: Skybox [10]

Integrace LOD – level of details je další integrací, která stojí určitě za zmínění. Při této metodě se integrace snaží o co největší zjednodušení objektů v modelu scény a to v závislosti na vzdálenosti uživatelé od daného objektu. Takto je dosažena velká úspora výpočetní techniky. Co zde ale chybí, je tvoření provozu, které je k určitým dopravním experimentům nezbytné a je tedy nutné jej naprogramovat zvlášť. [10]



Obrázek 10: Integrace LOD [10]

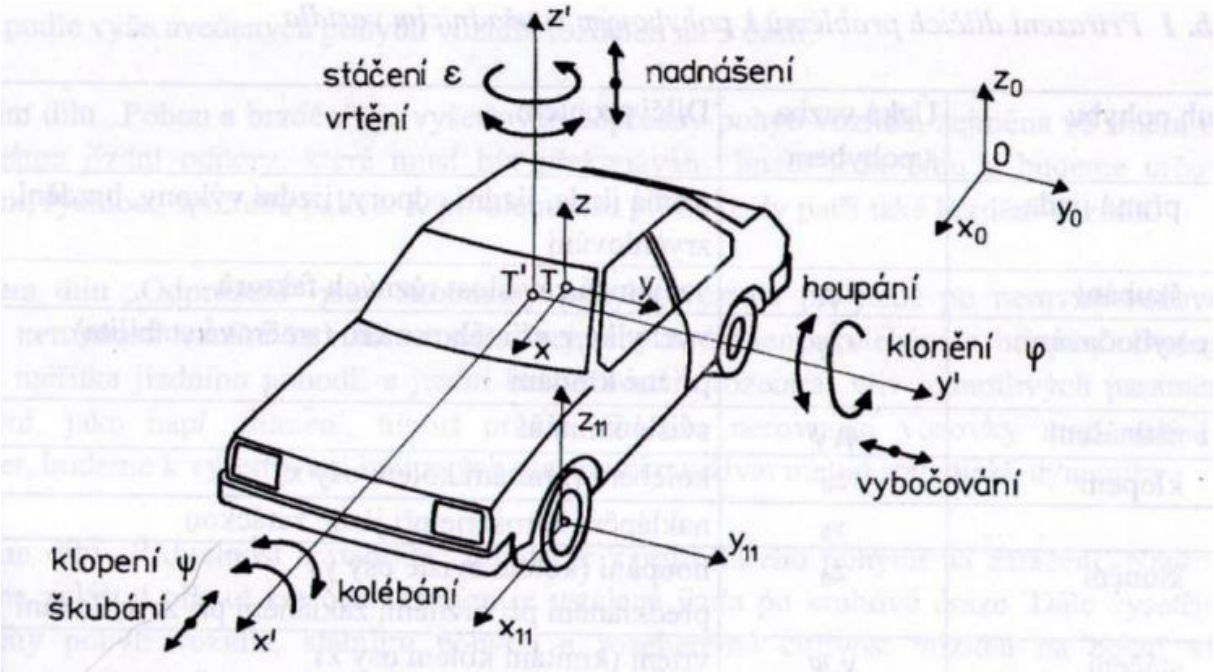


Obrázek 11: Integrace LOD (v závislosti na vzdálenosti) [10]

5 Pohyb vozidel v simulacích

5.1 Popis pohybu vozidla

Díky osám x, y a z můžeme popsat pohyb vozidla v průběhu času, a to polohou a natočením v těchto třech osách. Tento souřadnicový popis lze využit i při simulaci pohybu vozidla ve vozidlových simulátorech



Obrázek 12: Souřadnice určující polohu a natočení vozidla [4]

.

5.2 Dynamika jízdy vozidla

Chování vozidla během jízdy můžeme vyjádřit pomocí dynamiky jízdy vozidla. Pro větší přehlednost a přesnost dělíme dynamiku jízdy na 3 druhy.

První z nich je podélná dynamika, ta se zabývá rozjezdem a brzděním. U podélné dynamiky má na vozidlo v pohybu vliv jízdní odpor, který je určený součtem všech dílčích odporů (valivý odpor, odpor vzduchu, odpor stoupání, odpor zrychlení). Tento výsledný odpor určuje potřebný výkon hnacího mechanizmu vozidla k pohybu vpřed či vzad.

Akcelerace a decelerace

Akcelerace je definována jako aktuální zrychlení závisející na technických parametrech vozidla, aktuální rychlosti a míře sešlápnutí plynového pedálu. Mezi technické parametry vozidla patří výkon motoru, čelní plocha vozidla, součinitel vzdušného odporu, hmotnost vozidla. Parametry se většinou zjišťují přímo od výrobce vozidla.

Řidiči ve vozidlech ne vždy využívají maximální akceleraci při běžné jízdě, ale většinou zrychlují s tzv. komfortní akcelerací. Komfortní akcelerace je maximální zrychlení, při kterém

posádka jedoucího vozidla nepociťuje extrémní změnu rychlosti. Jízda je tedy pro ně komfortní neboli pohodlná. Typická hodnota komfortní jízdy pro osobní automobil je stanovena takto:

- Komfortní zrychlení – akcelerace = 1,0 m/s^2**,**
- Komfortní zpomalení – decelerace = 1,5 m/s^2**.** [2]

Každé vozidlo má své zrychlení limitované technickými parametry, a z toho vyplývá, že při dosažení určité rychlosti je zrychlení nižší, než je hodnota komfortní akcelerace. Na základě předchozích vztahů je možné vypočítat hodnotu maximálního zrychlení konkrétního vozidla pro danou hodnotu aktuální rychlosti.

Aby simulace pohybu vozidla v interaktivním vozidlovém simulátoru byla realistická, je zapotřebí zjistit hodnotu zrychlení pro každou hodnotu aktuální rychlosti. Jakmile tyto hodnoty zjistíme, je nutné určit bod, od kterého se hodnota maximálního zrychlení snižuje a je nižší než hodnota komfortního zrychlení. Každý typ vozidla má jiné technické parametry, na kterých závisí i různá hodnota rychlosti, od které jíž nejde zrychlovat s komfortním zrychlením.

Jako příklad znázornění závislosti zrychlení vozidla na jeho aktuální rychlost, jsem se rozhodl uvést vozidlo značky Škoda Octavia.

Tabulka 1: Parametry vozidla Škoda Octavia [9]

Škoda Octavia	
P [W]	85 000
m [kg]	1 775
S_x [m^2]	2,17
c_x [-]	0,28

Obrázek 13:Znázornění závislosti zrychlení na aktuální rychlosti vozidla Škoda Octavia [9]

Jak vidíme na grafu, hodnota, ve které se protínají maximální zrychlením s komfortním je cca 117 km/h. To znamená, že v tomto bodě nemůže vozidlo Škoda Octavia zrychlovat s komfortním zrychlením, ale už jen se zrychlením menším než 1 m/s^2.

Další z dynamik jízdy vozidla je dynamika příčná, ta popisuje vozidlo při průjezdu směrovým obloukem. Tak jako u dynamiky předchozí, i zde působí na vozidlo několik sil. Mezi ně patří síla odstředivá a dostředivá, další faktory ovlivňující jízdu vozidla je odstředivé či dostředivé zrychlení a boční síla větru.

Poslední ze zmíněných dynamik je dynamika svislá. Ta se zabývá silami, které působí na vozidlo ve svislém směru, což je logicky síla gravitační. Na komfort posádky má vliv odpružení vozidla, pomocí pérování kola vozidlo lépe kopírují podložku, ale i šetří namáhání součástek vozidla. [4]

Také kmitání vozidla se dá tlumit, a to díky tlumičům, které mění frekvence kmitání. Mimo tuto funkci také zajišťují dobrý kontakt pneumatiky s vozovkou a dostačující přítlačnou sílu kola na vozovku.

5.3 Pohyb vozidla při animaci

Z této kapitoly nám vyplývá, že pro definici pohybu vozidla při animaci jsou nejdůležitější hlavně souřadnice os x, y, z určující natočení a polohu vozidla v prostoru, hodnoty zrychlení a zpomalení při různých situacích a rychlost vozidla při jízdě.

PRAKTICKÁ ČÁST

6 Plug-in pro generování provozu pomocí triggeru

V praktické části své bakalářské práce se věnuji rozšíření plug-inu, který byl vytvořen na ČVÚT v bakalářské práci s názvem „Automatická tvorba provozu ve scénářích pro vozidlové simulátory“. Tuto práci jsem blíže popsal v kapitole Automatická tvorba provozu ve scénářích pro vozidlové simulátory. Všechny změny stávajícího plug-inu jsem tvořil pro 3D modelovací program Rhinoceros ve skriptovacím jazyce Rhinoscript.

Jak jsem již načal v kapitole 4.2.3, rozhodl jsem se plug-in upravit tak, aby nebyl závislý na segmentech (dlaždicích). Také jsem rozšířil plug-in o spouštění animací provozu pomocí tzv. triggeru a to protože v provozu je jeden z nejdůležitějších bodů také čas, kdy se v dané scéně vozidla v provozu začínají pohybovat nebo reagovat a k tomu nám dopomůže právě zmíněny trigger. Mnou vytvořený plug-in je složen ze tří dílčích funkcí – příprava vrstvy pro generování provozu, vykreslení rychlostní křivky a vytvoření složky s animačním souborem. Jednotlivé skripty funkcí jsem se rozhodl popsat pseudokódy, což jsou neformální přepisy kódu, které zjednodušeně popisují algoritmy funkce. K jejich čtení není zapotřebí znalost daného programovacího jazyka a jeho syntaktických pravidel.

6.1 Příprava vrstvy pro generování provozu

Před použitím plug-inu, je samozřejmě nutné vytvořit virtuální prostředí. Já jsem se rozhodl vytvořit prostředí pomocí plug-inu RoadCreator for Rhino, který jsem blíže popsal v kapitole Virtuální scéna.

První mnou vytvořená funkce RS_TvorbaVrstvy má za úkol vytvořit vrstvu RS_silnice v hladinách programu Rhinoceros, abychom definovali silnice v námi vymodelovaném prostředí. Tato funkce je velice důležitá pro tvoření animačního souboru a následného vytvoření triggeru. Aby se zvolená animace pohybovala po vymodelované silnici, a ne mimo ní.

Po spuštění funkce je uživatel vyzván k výběru plochy silnic, po kterých se budou pohybovat vozidla. Což lze provést jednoduchým klikem na vytvořený model silnic. Následně se nám mezi vrstvami objeví nová vrstva s názvem RS_silnice.

Obrázek 14: Vytvoření vrstvy silnic pomocí funkce RS_TvorbaVrstvy

Pseudokód

1 silnice = Výběr plochy ve virtuálním prostředí, které představují silnice

2 Vlož objekty Silnice do vrstvy RS_Silnice

6.2 Vykreslení rychlostní křivky

Jakmile máme definovanou silnici, může uživatel přistoupit k vykreslení grafu průběhu rychlosti. Graf nám vykreslí další část plug-inu s názvem RS_KrivkaRychlosti.

K dispozici má uživatel databázi vozidel, které jsou umístěny v systémové složce programu Rhinoceros. Složka obsahuje pro všechny typy vozidel jejich modely ve formátu pbtz a textury pro jednotlivé barevné variace připravené pro export do simulátoru. Samotný export těchto modelů a textur je popsán v kapitole 4.3. Import do simulátoru na ČVÚT a byl použit i pro tuto práci.

Než se samotný graf vykreslí je uživateli nabídnuto z výběru několika typů vozidel. Výběr vozidla se učiní přes HTML rozhraní a na výběr je zde ze 7 osobních automobilů, 2 dodávek, 2 kamionů a autobusu. Uživatel má také na výběr z několika barev vozidel, které lze vybrat opět prostřednictvím HTML rozhraní. Součástí této funkce je také možnost zadání hodnoty komfortního zrychlení a zpomalení, výchozí hodnoty jsou nastaveny na 1,0 m/s^2 pro zrychlení a 1,5 m/s^2 pro zpomalení.

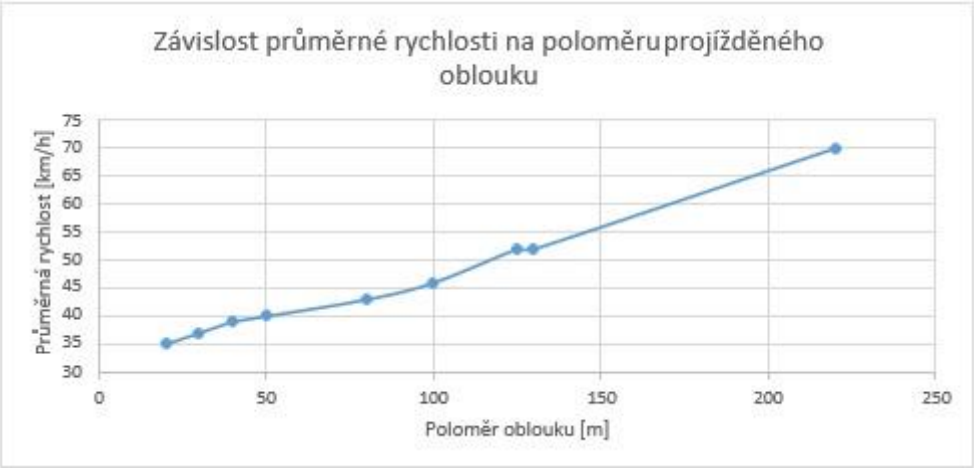
Obrázek 15: HTML rozhraní pro výběr vozidla

Obrázek 16: HTML rozhraní pro výběr barvy vozidla

Graf průběhu rychlosti se skládá z osy x a osy y. Osa x reprezentuje vzdálenost získanou z údajů o délce vybrané trasy provozu a osa y reprezentuje hodnoty rychlosti. Funkce také vykresluje pod grafem směrové vedení trasy, které znázorňuje informace o tom, kdy vozidlo projíždí obloukem nebo kdy jede přímo. Od těchto informací se odvíjí rychlost vozidla – pro přímé úseky je nastavena rychlost 90 km/h, pro oblouky je rychlost zvolená na základě rovnice, vycházející z naměřených hodnot Kláry Pudové a její diplomové práce „Automatická tvorba provozu ve scénářích pro vozidlové simulátory“ [5]. Tato rovnice je funkcí grafu na obrázku 17 a je definovaná jako: $y = 0.1692x + 31.051$ kde y je průměrná rychlost a x je poloměr.

Tabulka 2: Průměrné rychlosti při průjezdu oblouky [5]

Poloměr	Průměrná rychlost [km/h]
R20	35
R30	37
R40	39
R50	40
R80	43
R100	46
R125	52
R130	52
R220	70



Obrázek 17:Grafické znázornění závislosti průměrné rychlosti na poloměru projížděného oblouku [5]

Vykreslování rychlostní křivky začíná nejprve úsečkou pro každý úsek s její odpovídající hodnotou rychlosti na daném úseku. Následně jsou jednotlivé úseky propojovány parabolou, které reprezentují zrychlení a zpomalení vozidla.

Křivka decelerace se vykresluje na základě známých počátečních a koncových rychlostí s přihlédnutím na dané komfortní zpomalení.

Křivka akcelerace kromě počáteční, koncové rychlosti a komfortního zrychlení, bere v potaz i technické parametry vozidla, které uživatel zvolil na začátku. Důvod, proč jsou brány v potaz technické parametry vozidla, je vysvětlen v kapitole 5.2 Dynamika jízdy vozidla, konkrétně pod nadpisem akcelerace a decelerace.

Jakmile jsou původní úsečky oříznuty, vloženy propojovací křivky a odstraněny všechny přebytečné části křivek, vznikne nám nová křivka, která je sloučena do výsledné rychlostní křivky. Pro větší přehlednost je graf popsán podle vybraného typu a barvy vozidla.

Tato funkce byla vytvořená v diplomové práci, kterou jsem se rozhodl rozšířit. Jelikož není závislá na předchozích funkcích ani na tvoření segmentů, rozhodl jsem se ji použit beze změn i v mé práci.

Pseudokód

```
// vykreslení křivky zrychlení a zpomalení - pomocí funkcí
// funkce pro deceleraci - Krivkarychlosti (nezávislá na výkonu motoru)
(vstupní hodnoty: a0 - zrychlení [m/s2], v0 - počáteční rychlost [m/s], v1 - požadovaná rychlost [m/s])
time = (v1-v0) / a0 // celkový čas nutný ke změně rychlosti
i = 0
distance = 0
Do While i < 21 // rozdělení křivky zrychlení body na 20 částí
speed = v0 + a0 * time * (i/20)
dis = speed * time * (1/20) // přírůstek vzdálenosti pro 1/20 křivky
distance = distance + dis // aktuální celková vzdálenost
přidej bod o souřadnicích (distance - v0 * time * (1/20), speed - v0, 0)
Loop
Krivkarychlosti = prolož vzniklé body křivkou
// funkce pro akceleraci - Krivkarychlosti1 (závislá na výkonu motoru)
(vstupní hodnoty: a0 - zrychlení [m/s2], v0 - počáteční rychlost [m/s], v1 - požadovaná rychlost [m/s], P - výkon vozidla [W], m - hmotnost vozidla [kg], Sx - čelní plocha vozidla [m2], cx - součinitel vzdušného odporu [-])
ro = 1.225 // hodnota hustoty vzduchu [kg/m2]
f = 0.01 // hodnota součinitele valivého odporu pro asfalt
eta = 0.9 // hodnota účinnosti motoru
time = (v1-v0) / a0 // celkový čas nutný ke změně rychlosti
i = 0
distance = 0
v = v0
Do While i < 21 // rozdělení křivky zrychlení body na 20 částí
Of = v * m * 9.81 * f // odpor valivý
```

```
25 Ov = 0.5 * Sx * cx * ro * v3 // odpor vzduchu
26 Oz = P * eta - Of - Ov // odpor zrychlení
27 a = Oz / (v * m) // max. zrychlení
28 If a > a0 Then // výkon motoru je dostatečný pro toto zrychlení
29 speed = v0 + a0 * time * (i/20)
30 Else // výkon motoru není dostatečný - je dosáhnuto maximálního zrychlení
31 time = (v1-v0) / a
32 speed = v0 + a * time * (i/20)
33 End If
34 dis = speed * time * (1/20) // přírůstek vzdálenosti pro 1/20 křivky
35 distance = distance + dis // aktuální celková vzdálenost
36 přidej bod o souřadnicích (distance - v0 * time * (1/20), speed - v0, 0)
37 v = speed
38 i = i + 1
39 Loop
40 Krivkarychlosti1 = prolož vzniklé body křivkou
41 // vykreslení grafu rychlostní křivky
42 vozidlo, barva = uživatel vybere typ a barvu vozidla z nabídky přes HTML rozhraní
43 vytvoř novou vrstvu Rychlost
44 krivka = uživatel vybere křivku trasy provozu
45 points = rozděl křivku krivka body po 1 metru
46 pocatek = uživatel vybere bod, kam chce vykreslit graf rychlosti
47 akcelerace, decelerace = uživatel zadá hodnotu komfortního zrychlení a zpomalení v m/s2
48 vykresli osy grafu: osa x = vzdálenost [m] (délka podle délky trasy), osa y = rychlost [m/s] (0- 50)
49 přidej popis os grafu: název osy, hodnoty (staničení, rychlost)
50 For Each point In points
```

51 radius = změř poloměr křivky krivka v bodě point

52 podle hodnoty radius vykresli pod graf směrové vedení trasy

53 **Next**

54 k jednotlivým úsekům přidej popis úseku (přímá, oblouk o daném poloměru)

55 rychlosti = nad každý úsek (přímá, oblouk) vykresli v grafu rychlosti úsečku ve výšce odpovídající hodnotě rychlosti pro daný úsek

56 **For Each** rychlost In rychlosti

57 If výška úsečky rychlosti(i) > výška úsečky rychlosti(i+1) Then // zpomalení

58 vykresli spojovací křivku pomocí funkce Krivkarychlosti a přesuň ji na konec úsečky rychlosti(i)

59 **Else** // zrychlení

60 vykresli spojovací křivku pomocí funkce Krivkarychlosti1 (s parametry podle zvoleného vozidla) a přesuň ji na konec úsečky rychlosti(i)

61 **End If**

62 **Next**

63 ořež protínající se křivky a smaž všechny přebytečné části

64 sluč zbylé křivky ve výslednou křivku rychlosti

65 pojmenuj výslednou křivku podle zvoleného typu a barvy vozidla

6.3 Vytvoření složky s animačním souborem

Poslední části plug-inu je generování složky s animačním souborem a dalšími soubory nutnými k úspěšnému exportu do simulátoru. Tato část se dá považovat za nejdůležitější, protože se zde vygeneruje samotná animace a je definován i trigger.

Hlavním výstupem této funkce je už zmíněny animační soubor, který obsahuje všechny údaje definující animaci jízdy vozidla. V jednotlivých řádcích souboru je popsána pozice vozidla v daném čase. Díky těmto údajům generovaným ve sloupcích vzniká samotná animace pohybu vozidla.

Pro větší přehlednost souboru je v souboru vytvořena funkce s názvem Decimal, která zajišťuje stejný počet desetinných míst, aby měla čísla v tomto souboru stejnou délku. Znamená to, že pokud má číslo více desetinných míst je zaokrouhleno a pokud má míst méně, prázdná místa jsou doplněny nulami.

Uživatel je při spuštění této funkce nejprve dotázán, do jakého adresáře chce umístit složku animation. Do této složky jsou pak ukládány všechny potřebné soubory. Funkce je vytvořena tak, aby bylo možné do jednoho scénáře vkládat více animačních souborů pro různá vozidla. Aby byla jednotlivá vozidla od sebe odlišné, ukládají se postupně pod názvem auto1.anim, auto2.anim, atd.

Funkce se skládá ze dvou části. První je tvořena původní funkcí již zmíněné diplomové práce a druhou jsem doplnil svými skripty, které vytváří animace po aktivování určitého triggeru. Pro lepší orientaci jsem při tvorbě pseudokódu převzanou část označil šedým písmem a doplňující funkce základní černou barvou.

Pseudokód

1 // funkce Decimal (pro jednotnou délku zapisovaných čísel - větší přehlednost výstupního souboru)

2 (vstupní hodnoty: číslo, počet desetinných míst)

3 roundcislo = zaokrouhli číslo na daný počet desetinných míst

4 textcislo = převeď číslo roundcislo na text

5 rozděl textcislo v místě desetinné čárky

6 **If** počet částí = 2 **Then** // jedná se o desetinné číslo

7 doplňuj 0 k části za desetinnou čárku, dokud nedojdeš k požadovanému počtu desetinných míst

8 Decimal = první část čísla + desetinná čárka + druhá část čísla + přidané nuly

9 **Else** // jedná se o celé číslo

10 doplňuj 0, dokud nedojdeš k požadovanému počtu desetinných míst

11 Decimal = číslo + desetinná čárka + přidané nuly

12 **End If**

13 // vytvoření složky animation a souborů v ní

14 folder = uživatel vybere adresář pro uložení složky animation

15 newfolder = vytvoř složku animation v adresáři folder

16 vytvoř v newfolder soubor auto(i).anim (i = pořadí vytvářeného souboru) // např. pokud již ve složce existuje soubor auto1.anim, vytvoří se auto2.anim

17 trigger = Rhino.GetBoolean("Má se animace spouštět na trigger?", Array("Trigger", "ne", "ano"), Array(False)) //volba, zda se má animace spouštět na trigger

```
18 If (trigger = True) Then
19 Rhino.CurrentView "Top" //přepne pohled na pohled shora
20 triggerpoint = Rhino.GetPoint("Vyber umístění triggeru")
21 TPoint = Rhino.AddPoint(triggerpoint) //výběr místa triggeru
22 triggerprompoint = Rhino.ProjectPointToMesh(triggerpoint, plochysilnic, array(0, 0, 1))
23 If Isnull(triggerprompoint) Then
24 triggervyska = 0
25 Else
26 triggervyska = triggerprompoint(0)(2)
27 End If
28  cas = uživatel zadá počáteční čas pro spuštění animace
29  repeat = uživatel zvolí, zda se má animace opakovat
30  krivka = uživatel vybere křivku trasy provozu
31  krivkapoly = rozděl křivku krivka body (na přímé po 5 metrech, v obloucích hustěji)
32  points = body na křivce krivkapoly
33  krivkarychlost = uživatel vybere křivku rychlosti
34  plochysilnic = vyber všechny objekty ve vrstvě RS_Silnice
35  vykresli pomocný kříž pro určení náklonu vozidla (Pitchline: ±1.3 v ose x, Rollline: ±0.1
v ose y)
36  For Each point In points
37  Yaw = směrový úhel křivky v bodě point
38  prompoint = promítni bod point na všechny plochysilnic, pokud se promítne, ulož danou
plochu
39  distance = vzdálenost mezi bodem prompoint a minulým bodem
40  speed = rychlost podle grafu rychlosti - hodnota v bodě určeném pomocí distance
41  time = čas v minulém bodě + distance/speed
42  zkopíruj Pitchline a Rollline na bod point, orotuj je podle úhlu Yaw
```

43 promítni počáteční a koncový bod Pitchline a Rollline na plochysilnic

44 vypočítej úhel natočení Pitch a Roll

45 zapiš nový řádek do souboru auto(i).anim: "time, souřadnice x, y, z bodu prompoint, úhly

Roll, Yaw, Pitch" - hodnoty uprav pomocí funkce Decimal

46 Next

47 vytvoř v newfolder soubor animation.ini

48 zapiš do něj potřebné údaje pro mraky, auto(i)

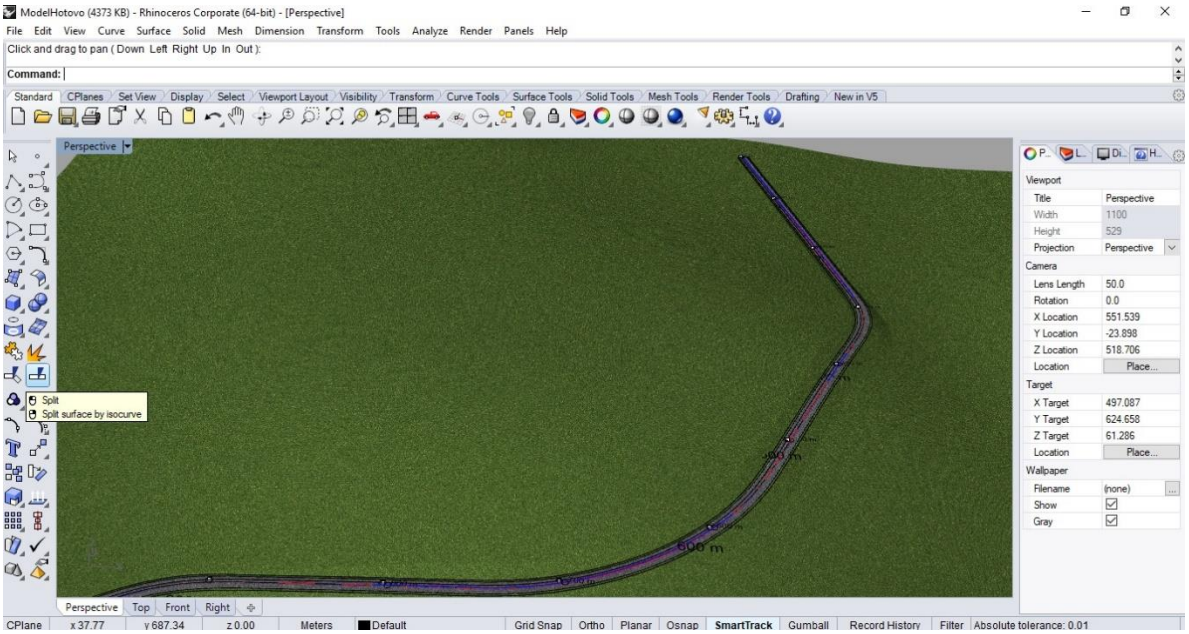
49 vytvoř trigger soubor

50 zkopíruj do složky newfolder soubor mraky.anim, model zvoleného vozidla (soubor pbtz), texturu pro zvolený typ a barvu vozidla (soubor dds)

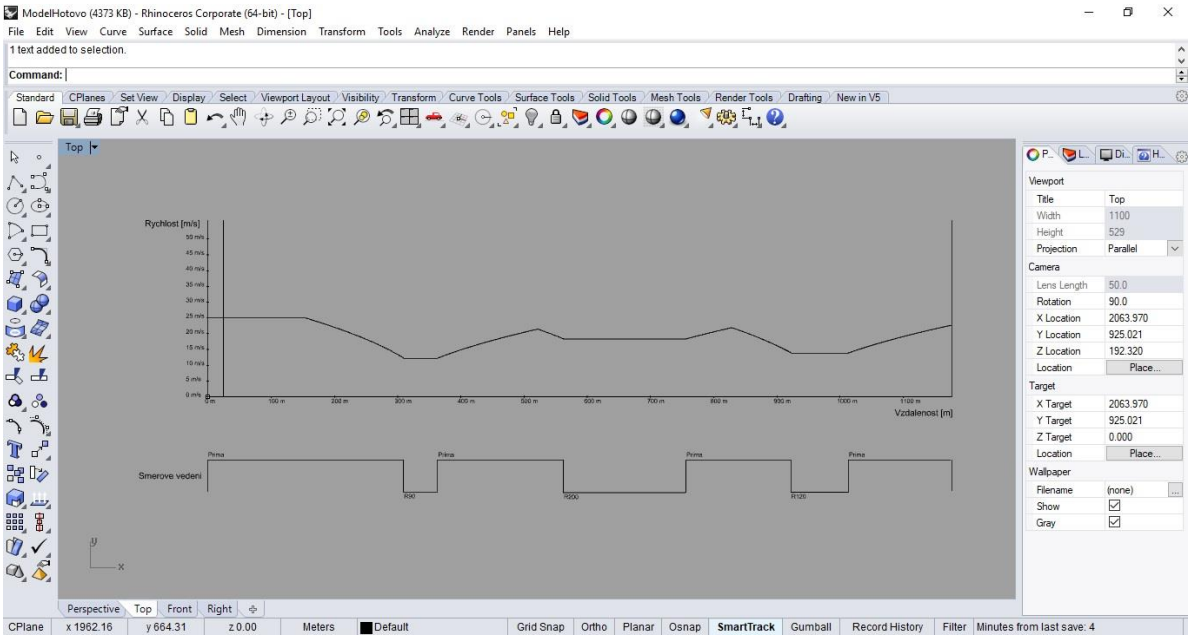
7 Experiment

Při využití všech dílčích funkcí mého plug-inu, nám vznikla složka animation, která obsahuje všechny potřebné soubory pro správné spuštění animace ve vozidlovém simulátoru. Mezi tyto soubory patří animační soubor auto1.anim a animační soubor se spuštěním na trigger auto1.trigg. Jakmile spustíme grafický engine, spustí se i animace provozu ve vymodelovaném scénáři, a to podle toho, jak si ji uživatel nastavil.

Virtuální scénu jsem vytvořil pomocí plug-inu RoadCreator v programu Rhinoceros.



Obrázek 18: Scéna vytvořená pomocí plug-inu RoadCreator



Obrázek 19: Rychlostní křivka pro vybranou trasu

Díky mému doplnění triggeru do funkce můžu určit bod ve scéně, na který bude animace vozidla reagovat. Jako uživatel mám také možnost opětovného použití funkcí plug-inu a mohu tedy snadno přidat libovolný počet dalších vozidel. Také si mohu zvolit pořadí vozidel a jaká bude časová mezera mezi nimi. Scéna na obrázku 20 byla obohacena o vegetaci a bezpečnostní prvky silnice pro větší realističnost.

Obrázek 20: Animace vozidla reagující na určený bod triggeru

Je pouze na uživateli, aby zvolil animaci její parametry – typ nebo barva vozidla, rychlostní křivka, čas začátku, podmínky animace, umístění trigger bodu nebo třeba i odlišnou trasu.

8 Závěr

Cílem této bakalářské práce bylo rozšířit metodu pro automatické generování provozu do scénářů pro vozidlové simulátory. Původní plug-in pracoval pouze se scénáři tvořenými pomocí segmentů.

V teoretickém úvodu byly popsány vozidlové simulátory jejich funkčnost a také využití v nejrůznějších experimentech. Jedná se o nenahraditelné nástroje pro testování řidičů v situacích, které by v reálném světě měly fatální následky. Díky vozidlovým simulátorům můžeme v bezpečných podmínkách virtuálního prostředí zkoumat reakce řidičů a následně je vyhodnocovat a dále s nimi pracovat. Byly zde také popsány jednotlivé druhy simulátorů, jejich části, výhody i nevýhody a principy tvorby virtuálního prostředí do nich. To se vytváří s využitím 3D i 2D počítačové grafiky, a to vždy podle požadavků daného experimentu.

Popsal jsem zde již existující plug-in, který generuje animace provozu na předem vytvořených segmentech. Já jsem se rozhodl tento plug-in rozšířit a tím i usnadnit tvorbu nových scénářů i jiným způsobem, než za pomocí segmentů. Tímto vzniká nová možnost tvoření silniční infrastruktury, která může být nyní zase blíže realitě, o což nám ve vozidlových simulátorech jde hlavně. Tato nově nalezená metoda proces urychli, zjednoduší, i usnadní práci na Fakultě Dopravní. Za tímto účelem jsem nejprve nastudoval původní plug-in a možné současné metody tvorby provozu do vozidlových simulátorů a následně tyto znalosti aplikoval do nově mnou vytvořeného plug-inu.

Další část bakalářské práce jsem věnoval správné definici animace jízdy vozidla, která vychází ze zákonitosti podélné, příčné a svislé dynamiky. K této definici nám napomáhají souřadnice určující polohu a natočení vozidla v prostoru. Dále bylo taky vysvětleno, že na animaci má vliv i rychlost jízdy vozidla a hodnoty jeho komfortního a maximálního zrychlení.

V dalších kapitolách byl popsán původní plug-in pro 3D modelovací program Rhinoceros, který umožňuje generování provozu do scénářů pro vozidlové simulátory. Po důkladných studiích tohoto plug-inu jsem se rozhodl ho rozšířit o novou funkci s názvem Trigger. Tato funkce hraje v generování provozu velkou roli, jelikož dává možnost tvoření animací v určitém čase, kdy řidič simulátoru projede určitým místem a animace je následně spuštěna. Trigger tedy otevírá nové možnosti experimentů pro vozidlový simulátor. Jako příklad experimentu může být třeba vytvořený bod ve scéně, který spustí animaci vozidla jedoucího přímo do cesty řidiče simulátoru. Zde můžeme pak pozorovat reakci řidiče, která by mohla v reálném světě být fatální.

Všechny úpravy původního plug-inu jsem vytvořil pomocí editoru RhinoScript Edtior pro 3D modelovací program Rhinoceros, pomocí skriptovacího jazyka VBScript. Všechny úpravy a nově vytvořené skripty jsou popsány formou pseudokódu, kde je srozumitelně popsán

algoritmus, aniž by bylo třeba znát syntaxi daného jazyka. Kódy jsou také okomentovány, aby se v nich dobře orientovalo a byla hned jasná funkčnost algoritmů. Díky jednoduché srozumitelnosti je samozřejmě také možné v budoucnu tento plug-in dále rozšiřovat.

Celý plug-in je navržen tak, aby byl uživatelsky co nejjednodušší a dostatečně přívětivý.

9 Použité zdroje

[1] ORLICKÝ, Adam. Automatická tvorba silniční infrastruktury ve 3D pro vozidlové simulátory. Praha, 2016. Diplomová práce. Fakulta dopravní, ČVUT. Vedoucí práce Doc. Ing. Stanislav Novotný, Ph.D.

[2] APELTAUER, Tomáš. Dopravní inženýrství - Modul 2: Modelování dopravního proudu.Brno: VUT, Fakulta stavební, 2007.

[3] PUDOVÁ, Klára. Tvorba počítačových 3D modelů segmentů komunikací pro vozidlové simulátory. Praha, 2017. Bakalářská práce. Fakulta dopravní, ČVUT. Vedoucí práce Doc. Ing. Stanislav Novotný, Ph.D, Ing. Adam Orlický

.

[4] VLK, František. Dynamika motorových vozidel: jízdní odpory : hnací charakteristika : brzdění : odpruženost : řiditelnost, ovladatelnost : stabilita. Brno: VLK, 2000. ISBN 80- 238-5273-6.

[5] PUDOVÁ, Klára, Automatická tvorba provozu ve scénářích pro vozidlové simulátory. Praha, 2019, diplomová práce. Fakulta dopravní ČVÚT, Vedoucí práce Doc. Ing. Stanislav Novotný, Ph D, Ing. Adam Orlický

[6] ŽÁRA, Jiří, Bedřich BENEŠ a Petr FELKEL. Moderní počítačová grafika. Praha: Computer Press, 1998. ISBN 80-7226-049-9.

[7] CULEK, Jakub. Porovnání vhodných grafických enginů pro interaktivní vozidlové simulátory. Praha, 2016. Bakalářská práce. ČVÚT FD.

[8] KERLOW, Isaac Victor. Mistrovství 3D animace Brno: Computer Press, 2011. Mistrovství. ISBN 978-80-251-2717-9.

[9] Škoda Octavia: katalog. Autobible.cz [online]. Dostupné z: https://autobible.euro.cz/wp-content/uploads/2017/01/Technick%C3%A9-%C3%BAdaje-%C5%A0koda-Octavia-2017-z%C3%A1%C5%BEehov%C3%A9-motory.pdf

[10] Trainz portal. Americké fórum pro nadšence 3D modelování. [online]. Dostupné z: http://www.trainzportal.com/blog/view/view/controlling-level-of-detail-distance

[11] Instructables craft. Americký portál s nejrůznějšími grafickými postupy. [online]. Dostupné z https://www.instructables.com/id/Make-Low-Poly-Objects/

10 Seznam obrázků

Obrázek 1: Příklad rastrové a vektorové grafiky 12
Obrázek 2: Simulátor typu CAVE 15
Obrázek 3: Vizualizace vozidlového simulátoru na FD v Děčíně (laboratoř) 16
Obrázek 4: Vizualizace vozidlového simulátoru na FD v Děčíně (kokpit) 17
Obrázek 5: Příklad virtuální scény 18
Obrázek 6: Low poly object [11] 19
Obrázek 7: Ukázka základní dlaždice [3] 20
Obrázek 8: Propojení dlaždic v celou scénu [3] 21
Obrázek 9: Skybox [10] 26
Obrázek 10: Integrace LOD [10] 26
Obrázek 11: Integrace LOD (v závislosti na vzdálenosti) [10] 27
Obrázek 12: Souřadnice určující polohu a natočení vozidla [4] 28
Obrázek 13:Znázornění závislosti zrychlení na aktuální rychlosti vozidla Skoda Octavia [9] 30
Obrázek 14:Vytvoření vrstvy silnic pomocí funkce RS_TvorbaVrstvy 32
Obrázek 15: HTML rozhraní pro výběr vozidla 33
Obrázek 16: HTML rozhraní pro výběr barvy vozidla 33
Obrázek 17:Grafické znázornění závislosti průměrné rychlosti na poloměru projížděného oblouku [5] 34
Obrázek 18: Scéna vytvořená pomocí plug-inu RoadCreator 40
Obrázek 19: Rychlostní křivka pro vybranou trasu 41
Obrázek 20: Animace vozidla reagující na určený bod triggeru 41

11 Seznam příloh

Obsah CD

Plugin

Skripty pro jednotlivé funkce:

- RS_TvorbaVrstvy.rvb
- RS_KrivkaRychlosti.rvb
- RS_TvorbaSouboru.rvb

Soubory v databázi vozidel:

- model vozidla: vozidlo.3dm, vozidlo.pbt, vozidlo.pbtz (pro všechny typy a barvy vozidel)
- textura vozidla: vozidlo.dds (pro všechny typy a barvy vozidel)

Scénář vytvořený pro experiment:

- RS_ModelScénáře.3dm
- Složka se všemi soubory potřebnými pro spuštění scénáře v grafickém enginu (spouštěcí soubor: AutoSimt.exe)

Textury

Bakalářská práce

- F6D-BP-2019-Vidarte-Vargas-Rodrigo-Bakalarska-Prace

Veškeré obrázky, tabulky a přílohy, které nemají uvedený zdroj, byly vytvořeny autorem práce.