



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMOBILNÍHO A DOPRAVNÍHO INŽENÝRSTVÍ

INSTITUTE OF AUTOMOTIVE ENGINEERING

NÁVRH VIRTUÁLNÍHO SVĚTA PRO TESTOVÁNÍ AUTONOMNÍCH VOZIDEL

DESIGNING A VIRTUAL WORLD FOR TESTING AUTONOMOUS VEHICLES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Turek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Pavel Kučera, Ph.D.

BRNO 2024

Zadání bakalářské práce

Ústav: Ústav automobilního a dopravního inženýrství
Student: **Martin Turek**
Studijní program: Základy strojního inženýrství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **doc. Ing. Pavel Kučera, Ph.D.**
Akademický rok: 2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh virtuálního světa pro testování autonomních vozidel

Stručná charakteristika problematiky úkolu:

Důležitou součástí vývoje autonomních vozidel je jejich testování nejen v reálném prostředí, ale také ve virtuálním světě. Úkolem je vytvořit virtuální svět vybrané lokality v ČR pomocí softwaru RoadRunner, který bude detailně odpovídat reálné předloze. Dále je potřeba ověřit funkčnost a export do prostředí pro testování autonomních vozidel.

Cíle bakalářské práce:

Rešerše k dané oblasti.
Návrh virtuálního světa vybrané lokality v ČR.
Ověření funkčnosti.

Seznam doporučené literatury:

DIETSCHKE, Karl-Heinz a Konrad REIF. Automotive Handbook. 11. Karlsruhe: Robert Bosch, 2022. ISBN 978-1-119-91190-6.

ČERNÝ, David, Ondřej VACULÍN a Petr ZÁMEČNÍK. Automatizované řízení vozidel a autonomní doprava: technické a humanitní perspektivy. Praha: Academia, 2022, 389 stran: ilustrace, grafy. ISBN 978-80-200-3358-1.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

prof. Ing. Josef Štětina, Ph.D.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

ABSTRAKT

Bakalářská práce se zabývá návrhem virtuálního světa pro testování autonomních vozidel, jehož předlohou je reálná lokalita v ČR. Vozidlo řízené chybně navrženým algoritmem autonomního řízení nebo ADAS systémem potenciálně představuje velké bezpečnostní riziko pro všechny účastníky silničního provozu. Z toho plyne nutnost vše velmi důkladně odzkoušet, přičemž virtuální simulační prostředí představuje ekonomicky výhodnou alternativu k nákladnému testování prototypů ve skutečném světě. Detailní virtuální reprezentace vybrané lokality byla vytvořena v softwaru RoadRunner na základě digitálních mapových a fotografických podkladů. Funkčnost vytvořeného světa byla ověřena v Unreal Engine prostřednictvím jednoduché simulace navržené v prostředí Simulinku. Výstupem práce je funkční virtuální prostředí, jehož předlohou bylo blízké okolí FSI v Brně. Součástí vytvořeného světa je veškerá důležitá dopravní infrastruktura a vizuální autenticita okolí silnic je zajištěna umístěním 3D modelů reálných objektů.

KLÍČOVÁ SLOVA

Virtuální svět, autonomní vozidla, RoadRunner, simulace, Simulink, Unreal Engine

ABSTRACT

The bachelor's thesis deals with a design of a virtual world for testing autonomous vehicles, which is based on a real location in the Czech Republic. A vehicle controlled by a flawed autonomous driving algorithm or ADAS system potentially poses a major safety risk to all road users. Because of that everything needs to be tested very thoroughly, with a virtual simulation environment being a cost-effective alternative to the expensive testing of prototypes in the real world. A detailed virtual representation of the selected location was created in the RoadRunner software using digital maps and photographs. The functionality of the created world was verified in Unreal Engine through a simple simulation designed in the Simulink environment. The output of this work is a functional virtual environment, representing the immediate surroundings of the FME in Brno. The created world contains all important road infrastructure, and the visual authenticity of road surroundings is achieved by the placement of 3D models of real objects.

KEYWORDS

Virtual world, autonomous vehicles, RoadRunner, simulation, Simulink, Unreal Engine

BIBLIOGRAFICKÁ CITACE

TUREK, Martin. *Návrh virtuálního světa pro testování autonomních vozidel*. Online, bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automobilního a dopravního inženýrství, 2024. Vedoucí práce Pavel Kučera. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/157059>.



ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Pavla Kučery, Ph.D. a s použitím informačních zdrojů uvedených v seznamu.

V Brně dne 24. května 2024

.....

Martin Turek

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své bakalářské práce doc. Ing. Pavlu Kučerovi, Ph.D. za jeho cenné rady a připomínky a za velkou vstřícnost a ochotu během konzultací. Velké poděkování dále patří mé rodině za jejich podporu v průběhu mého dosavadního studia.

OBSAH

Úvod	11
1 Autonomní vozidla a virtuální svět	12
1.1 Stupně automatizace	12
1.2 Senzory	13
1.2.1 Kamera.....	13
1.2.2 Radar.....	14
1.2.3 Lidar	15
1.2.4 Ultrazvukové snímače	15
1.2.5 GNSS	16
1.2.6 Inerciální měřicí jednotka.....	16
1.2.7 Rozmístění senzorů na vozidle	16
1.3 Software k testování vozidel ve virtuálním prostředí.....	17
1.3.1 Formáty ASAM OpenX	17
1.3.2 MATLAB RoadRunner	18
1.3.3 Truevision Designer	18
1.3.4 Automated Driving Toolbox	19
1.3.5 CARLA.....	20
1.3.6 NVIDIA DRIVE Sim	20
1.3.7 CarMaker	21
2 Návrh virtuálního světa vybrané lokality v ČR.....	23
2.1 Volba lokality	24
2.2 Mapové podklady	24
2.3 Tvorba silniční sítě	25
2.3.1 Silnice	25
2.3.2 Křižovatky	29
2.3.3 Parkoviště	32
2.4 Tvorba ostatních ploch v okolí silnic	35
2.4.1 Chodníky	35
2.4.2 Okolní terén	39
2.5 Dopravní značení.....	40
2.5.1 Vodorovné dopravní značení.....	40
2.5.2 Svislé dopravní značení	41
2.6 Modely.....	42
3 Ověření funkčnosti	46
3.1 Export z RoadRunneru a import do Unreal Engine.....	46
3.2 Propojení se Simulinkem.....	48
3.3 Nastavení Simulace	48
3.4 Stanovení trajektorie herců.....	49
3.5 Nastavení kamery a pořízení videozáznamu	50



Závěr	51
Použité informační zdroje	52
Seznam použitých zkratk a symbolů	58
Seznam příloh.....	59

ÚVOD

Autonomní řízení je v současné době jednou z nejdynamičtěji se rozvíjejících oblastí automobilového průmyslu. Hnacímotorem tohoto rozvoje je potenciál autonomních vozidel zásadně zvýšit bezpečnost provozu na silničních komunikacích díky schopnosti počítačů mnohem rychleji a spolehlivěji reagovat na aktuální ději na silnici a možnosti neustálé komunikace s okolními vozidly. V neposlední řadě pak také autonomní vozidla umožňují cestujícím lidem se místo řízení věnovat například práci nebo odpočinku.

Z výše zmíněných důvodů proto automobilky, ale i jiné technologické firmy, do jejich vývoje investují nemalé finanční prostředky. V současné době jsou už pokročilé asistenční systémy (ADAS) zcela běžnou součástí výbavy nových automobilů a mnoho společností již disponuje prototypy vozidel, které jsou za daných podmínek schopné autonomní jízdy. Doba, kdy se se zcela autonomními vozidly budeme moci zcela běžně setkávat i na veřejné silnici, je ale ještě stále relativně vzdálená. Před jejich vpuštěním do reálného provozu je totiž nutné důkladně ověřit, že jejich systémy vykazují maximální možnou spolehlivost, a tato vozidla tedy nebudou místo kýženého zvýšení bezpečnosti naopak představovat hrozbu jak pro posádku, tak i pro ostatní účastníky silničního provozu.

Najezdit s reálnými prototypy miliony kilometrů v mnoha různých prostředích za všech představitelných světelných a povětrnostních podmínek by bylo ekonomicky neúnosné a možná i zcela nemožné. Proto hrají při vývoji systémů autonomního řízení klíčovou roli počítačové simulace, které jednak umožňují simulace provádět v podstatě kdekoli a kdykoli, ale taktéž zcela eliminují riziko poškození drahého prototypu vozidla.

Aby byly výsledky prováděných simulací co nejrelevantnější, je potřeba aby používaný simulátor disponoval věrnými výpočtovými modely jak jízdní dynamiky vozidla, tak především virtuálním světem a senzory, které slouží k získávání potřebných dat při jejich iteraci. Jednotlivé druhy senzorů používané v autonomních vozidlech, jejichž chování je v simulaci třeba napodobit, jsou popsány v první kapitole této práce. Dále se tato kapitola věnuje vybraným simulátorům, používaným k virtuálnímu testování, a softwarům, které slouží k tvorbě samotného virtuálního prostředí.

Hlavním cílem této bakalářské práce je vytvořit virtuální svět vybrané lokality v České republice, který bude detailně odpovídat své reálné předloze. K tomu byl využit software RoadRunner od společnosti MathWorks, přičemž postup tvorby virtuálního prostředí je podrobně popsán ve druhé kapitole.

Závěrečná kapitola se věnuje poslednímu cíli, kterým je ověření funkčnosti vytvořeného virtuálního světa. Je třeba vyzkoušet, že je možné svět nahrát do vybraného simulačního prostředí a že se v něm virtuální vozidlo ovládané simulátorem může bez problémů pohybovat.

1 AUTONOMNÍ VOZIDLA A VIRTUÁLNÍ SVĚT

Pod pojmem autonomní vozidlo se rozumí mechatronický systém, který je schopný zcela samostatné jízdy bez jakýchkoliv lidských zásahů [1]. Tato vozidla jsou zpravidla opatřena velkým množstvím různých senzorů a snímačů, které získávají velké množství informací o prostředí, ve kterém se vozidlo pohybuje. Data ze snímačů jsou neustále přijímána řídicí jednotkou (ECU), která z nich prostřednictvím různých metod, jako mohou být třeba neuronové sítě, rozpoznává okolní objekty a zpracovává informace například o poloze vozidla s tímto systémem a o poloze a pohybu dalších vozidel a jiných účastníků silničním provozu v okolí [1; 2]. Na základě takto zpracovaných dat, informací o chování vozidla a požadavků uživatele řídicí jednotka pomocí pokročilých algoritmů vyhodnocuje optimální trajektorii vozidla a ovládá řízení, brzdění a akceleraci [1]. Z důvodu obrovských datových toků a nutnosti provádět úkony popsané výše v reálném čase musí vozidlo disponovat dostatečně velkým výpočetním výkonem [3].

1.1 STUPNĚ AUTOMATIZACE

Vozidla, která se v současné době pohybují po silnicích, se výrazně liší co do schopností autonomní jízdy. Je tedy vhodné zavést dělení podle míry jejich autonomie. Celosvětovým standardem se stala klasifikace do šesti stupňů automatizace podle SAE International [3].

Jako první se uvádí tzv. **nultý stupeň**, kdy je řízení zcela v rukou řidiče. Spadají sem tedy zcela obyčejná vozidla vyráběná po desetiletí, u kterých se žádná automatizace nevyskytuje [4].

Nejnižším stupněm automatizace je tzv. **první stupeň**. V tomto případě asistenční systémy zajišťují buď zatáčení, jako to dělá například asistent držení v jízdních pruzích, nebo ovládají rychlost jízdy vozidla prostřednictvím zrychlování či zpomalování, jako je tomu u již zmíněného adaptivního tempomatu [4]. Vozidlo ale do této kategorie spadá pouze v případě, že tyto systémy nejsou schopny pracovat souběžně, v opačném případě se již jedná o vyšší stupeň automatizace [4]. V případě tohoto stupně slouží systémy zmíněné výše výhradně jako podpora řidiče, který zůstává zcela zodpovědný za jízdu vozidla a musí chování asistenčních systémů neustále kontrolovat [4].

To samé platí i pro **druhý stupeň** [4]. Sem vozidlo spadá, je-li součástí jeho výbavy systém, který je schopný vozidlo souběžně kontrolovat jak v podélném, tak i v příčném směru [3]. Mezi systémy druhého stupně spadá například již zmíněný asistent pro jízdu po dálnici [3]. Dále se sem řadí i automobily společnosti Tesla, u kterých je inzerována schopnost zcela samostatného řízení (Full-Self Driving) [5]. Podle SAE se nicméně o zcela autonomní řízení nejedná. Veškerá zodpovědnost za chování vozidla totiž stále leží na bedrech řidiče, který musí být kdykoliv připraven do řízení zasáhnout [5].

Velmi odlišná je situace u **třetího stupně** automatizace. Takový systém je za určitých podmínek již schopen převzít veškeré úkony spojené s řízením vozidla [3]. Osoba za volantem už za těchto specifických podmínek nemusí na systém neustále dohlížet a není považována za řidiče [4]. Stále ale platí, že musí být dotyčný připraven převzít řízení, jestliže si to systém vyžádá [4]. Do tohoto stupně spadá například systém Drive Pilot dostupný ve vrcholných modelech značky Mercedes-Benz, který je za stanovených podmínek možné aktivovat na vybraných úsecích dálnic v Kalifornii a Nevadě [6].

Systemy **čtvrté úrovně** jsou schopné zcela autonomního řízení vozidla za určitých podmínek [4]. Na rozdíl předchozí úrovně si ale systém ani v případě poruchy nevyžádá zásah řidiče, a tudíž už nese veškerou zodpovědnost za chování vozidla [3]. Do této úrovně lze zařadit například autonomní taxi od společnosti Waymo, která operují ve vybraných lokalitách v Arizoně [7].

Cílem vývoje autonomních vozidel je dosažení **páté úrovně**, kam spadají vozidla schopná zcela autonomní jízdy za jakýchkoliv podmínek. Uživatel vozidla, které už ani nemusí být vybaveno volantem a pedály, pouze zadá cílovou destinaci a veškerá zodpovědnost za chování vozidla je na řídicím systému, který i v případě poruchy nesmí nikterak ohrozit posádku ani okolní účastníky silničního provozu. [3; 4]

1.2 SENZORY

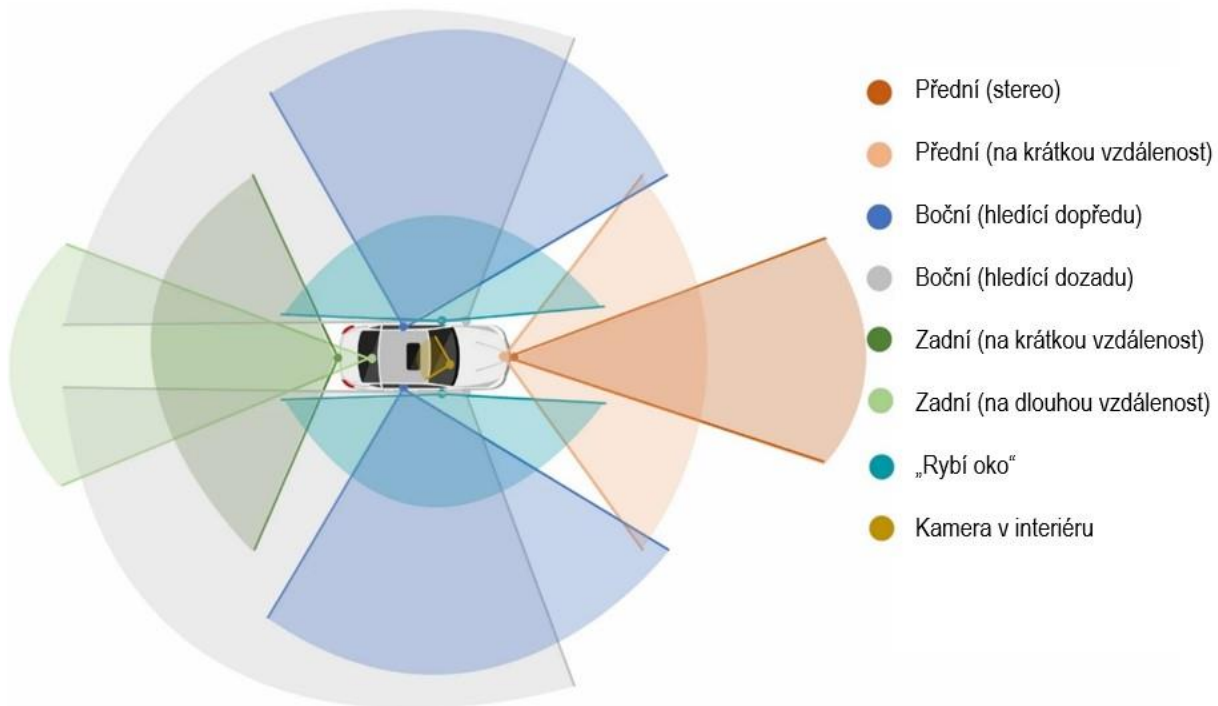
Aby se autonomní vozidlo nebo vozidlo vybavené ADAS systémy mohlo samostatně pohybovat, musí disponovat informacemi o tom, kde přesně se nachází, jak vypadá jeho bezprostřední i vzdálené okolí a co se kolem něj v daném okamžiku děje. Kvůli tomu se vozidla osazují senzory, které jsou schopny tyto informace získat. Každý typ používaných senzorů má svoje přednosti a nevýhody, a proto jsou vozidla zpravidla vybavena kombinací různých senzorů, které se navzájem doplňují.

1.2.1 KAMERA

Neodmyslitelnou součástí výbavy autonomních vozidel jsou kamery. Ty se používají k rozpoznávání a sledování objektů v okolí vozidla, jako například jízdní pruhy, dopravní značky, semaforey nebo chodci [8]. Velkou výhodou kamer je relativně nízká cena, kompaktní rozměry a schopnost získat velké množství informací o okolních objektech jako jsou nejen rozměry ale i jejich barva nebo textura povrchu [9]. Na druhou stranu jsou ale citlivé na počasí a na intenzitu osvětlení [9]. Jinak řečeno, klasická barevná kamera nedokáže při snížené viditelnosti v noci nebo za špatného počasí poskytovat tolik relevantních informací jako za jasného dne.

Digitální kamera se skládá ze dvou hlavních částí, z objektivu a z polovodičového optického snímače [3]. Optický snímač rozhoduje o rozlišení kamery [3], přičemž trendem posledních let je postupný nárůst rozlišení kamer [3]. To sice umožňuje přesněji rozeznávat objekty v okolí vozidla, na druhou stranu ale rostou nároky na výpočetní výkon potřebný pro zpracování obrazu. Objektiv prostřednictvím čoček přivádí světlo z okolí na optický senzor uvnitř kamery a jeho konstrukce společně s rozměry optického snímače určuje velikost zorného pole [3]. Kamery určené na velké vzdálenosti mívají zorné pole užší, zatímco u tzv. rybích ok (viz *obr. 1*) může zorný uhel přesahovat i 180° [3]. Těch bývá na vozidlech několik a používají se k vytvoření 360stupňového zobrazení blízkého okolí vozidla. Takové zobrazení je velmi užitečné například při parkování nebo popojíždění v kolonách [7].

Kamery se mohou kromě velikosti zorného pole lišit i počtem objektivů. Nevýhodou kamer s jedním objektivem je jejich neschopnost vnímat rozdílnou vzdálenost pozorovaných objektů. Tento problém řeší tzv. stereo kamery, které disponují schopností prostorového vidění. Z podstaty své konstrukce mají ale tyto kamery užší zorné pole (viz *obr. 1*), a jsou tedy vhodné především pro použití na delší vzdálenosti. Zobrazení zorného pole stereo kamery na *obr. 1* je pouze schématické a nezohledňuje to, že tato kamera má ve skutečnosti optické snímače dva. [7]



Obr. 1 Příklad rozmístění kamer a schématické znázornění jejich zorných polí ([61], upraveno)

1.2.2 RADAR

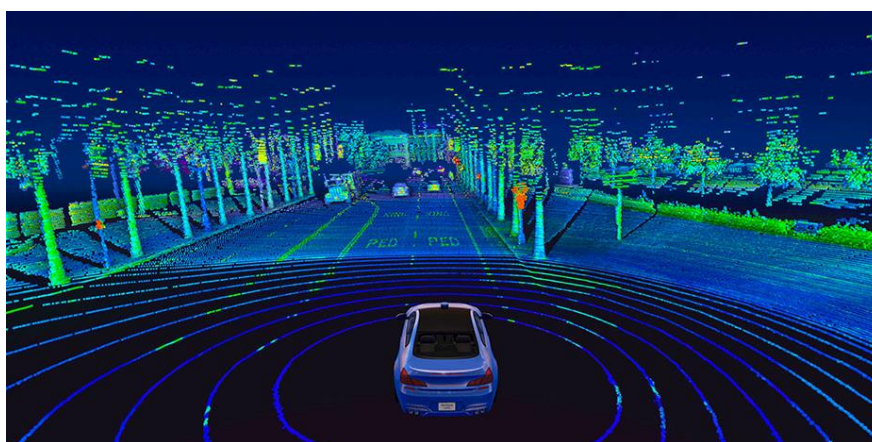
Dalším důležitým senzorem je radar. Ten vysílá do svého okolí v pravidelných intervalech elektromagnetické vlny, které se odrazí od okolních překážek a část z nich se vrací zpět k radaru [3]. Z doby, která uplynula mezi vysláním a přijetím signálu, lze následně určit vzdálenost dané překážky od vozidla [3]. Pomocí Dopplerova jevu je dále možné určit relativní rychlost dané překážky [7]. Azimut překážky se určuje z úhlu, pod kterým se odražená vlna vrací k přijímacím anténám radaru [8]. Aby systém odhadu úhlu fungoval, musí být přijímací antény alespoň dvě, ale v praxi jich za účelem dosažení vyšší přesnosti používá i více [8].

Velkou výhodou radarů je schopnost pracovat spolehlivě za zhoršených světlených či povětrnostních podmínek [9]. Tma nebo například mlha pro ně tedy na rozdíl od kamer nepředstavují problém. Další výhodou je, že být umístěny např. za nárazník, díky čemuž jsou pak lépe chráněny před okolními vlivy [7]. Na druhou stranu ale nejsou kvůli relativně nízkému rozlišení vhodné k rozpoznávání konkrétních objektů [7]. Jinak řečeno, radar je schopný velmi spolehlivě detekovat přítomnost překážky, ale má problém určit, o co konkrétně se jedná. Problém může představovat i rušení od radarů na jiných vozidlech [9].

Vozidla mohou být vybavena větším počtem radarů, které se mohou lišit vzdáleností, na kterou jsou určeny (viz obr. 3). Existují jednak radary s dlouhým (LRR) a s krátkým dosahem (SRR), které se navzájem liší frekvencemi, se kterými pracují [8]. První zmíněné mají dosah přibližně 300 m a šířku zorného úhlu do 100° a využívá je například adaptivní tempomat k hlídání vzdálenosti od vpředu jedoucího vozidla [3; 7]. Naopak například systémy, které při vyjíždění z parkovacích varují před projíždějícími vozidly, používají radary s krátkým dosahem, který se typicky pohybuje od 20 do 50 m, přičemž zorný úhel může mít až 160° [3; 7]. Aktuálně se ve vozidlech také čím dál častěji objevují radary na střední vzdálenost (MRR), které jsou do určité míry schopné zastat funkci obou výše zmíněných typů [3].

1.2.3 LIDAR

Dalším senzorem, který bývá součástí výbavy autonomních vozidel, je lidar. Lidar pracuje na principu ToF (time of flight) [8], kdy se měří doba od vyslání laserového paprsku do jeho návratu po odražení od objektu v okolí vozidla. Z této doby je pak na základě rychlosti šíření světla určena vzdálenost bodu, ve kterém došlo k odrazu paprsku [8]. Tímto způsobem zařízení postupně určí polohu tisíců bodů, čímž vzniká 3D reprezentace okolí vozidla ve formě tzv. mračna bodů (viz obr. 2) [7]. Lidary se tak mohou například ve spojení s přesnými mapovými podklady používat k přesné lokalizaci vozidla, což je velmi účinné v městském prostředí, kde nemusí určování polohy pomocí GNSS fungovat spolehlivě [8].



Obr. 2 Mračno bodů z lidarů [62]

Výhodou lidarů je, že se vyznačují vysokou přesností i na dlouhé vzdálenosti (200 až 300 m) a na rozdíl od kamery funguje v noci stejně spolehlivě jako ve dne. Jeho výhodou oproti radaru je, že dokáže rozeznávat objekty téměř jakýchkoliv tvarů a velikostí bez ohledu na materiálové a fyzikální vlastnosti jejich povrchu. V porovnání s ním je ale citlivější na špatné počasí a negativní vliv může mít i ostrý sluneční svit. V současné době představuje problém také jejich vysoká cena, a to především u mechanických lidarů. [3]

Co se dosahu týče, existují lidary určené pro použití jak na krátké, tak dlouhé vzdálenosti. V praxi se používají především ty s delším dosahem, jelikož na krátké vzdálenosti je možné je spolehlivě nahradit levnějšími technologiemi jako jsou radary či ultrazvukové snímače [3]. Lidary lze dále dělit podle toho, jakým způsobem je řešeno směřování laserového paprsku. Tzv. mechanický lidar je vybaven vysoce kvalitní optikou a otočnou sestavou, díky které se dokáže otáčet o 360°, a lidar je tak schopný skenovat celé okolí vozidla [8]. Naproti tomu lidar typu SSL používá ke směřování laserových paprsků mikroelektromechanická zrcadla [8]. Absence velkých pohyblivých částí vede k nižší ceně, nicméně šířka zorného pole v horizontálním směru běžně dosahuje pouze 120° [7; 8].

1.2.4 ULTRAZVUKOVÉ SNÍMAČE

Dalšími běžně používanými senzory jsou ultrazvukové snímače, které vysílají pro člověka neslyšitelné zvukové vlny, jež se odrážejí od objektů v okolí [3]. Vzdálenost jednotlivých objektů se stejně jako v případě lidarů určí pomocí principu ToF [8]. Ultrazvukové snímače typicky bývají integrovány do nárazníků vozidla (viz obr. 3) a zpravidla jich bývá několik [3]. Díky tomu, že naráz jednu překážku detekuje více snímačů, je možné metodou triangulace dopočítat její přesnou polohu [3].

Ultrazvukové snímače se na krátké vzdálenosti vyznačují velmi vysokou přesností [10]. Další důvodem, proč jsou využívány, je jejich příznivá cena a to, že dobře odolávají vlivům nepříznivého počasí [3]. Problém může nicméně představovat akustické rušení [10]. Dalším omezením je již zmíněný relativně krátký dosah, který se řádově pohybuje v nižších jednotkách metrů [11]. Z toho důvodu se tyto snímače používají především v situacích, kdy se vozidlo pomalu pohybuje v těsné blízkosti nějakých překážek [11]. Typicky se jedná o parkování. Moderní ultrazvukové snímače mohou být v nízkých rychlostech dále využívány i systémy nouzového brždění [3]. Naopak ve vyšších rychlostech je možné je použít pro hlídání mrtvého úhlu [3].

1.2.5 GNSS

Autonomní vozidla bývají dále vybavena zařízením, které slouží k určování polohy na zemském povrchu pomocí globálního družicového polohového systému (GNSS). Poloha vozidla je určována na základě znalosti polohy jednotlivých družic na oběžné dráze Země a doby, za kterou družicemi vysílaný signál k přijímači doputoval [3]. Pro určení zeměpisné šířky a délky je potřeba signál z alespoň tří družic, k určení nadmořské výšky je třeba ještě družice čtvrtá [3]. V současné době globálně fungují čtyři družicové systémy. Těmi jsou americký systém GPS, evropský Galileo, ruský GLONASS a čínské BeiDou [12].

Při použití pouze signálů z družic lze dosáhnout přesnosti v řádu jednotek metrů [8], nicméně existuje několik metod k jejímu zvyšování. Tou nejpřesnější je metoda RTK, s jejímž využitím lze dosáhnout přesnosti v řádu centimetrů [12]. GNSS je tak možné použít například i k určování, ve kterém konkrétním pruhu se vozidlo nachází. K tomu však musí mít vozidlo k dispozici přesné mapové podklady, kde jsou tyto informace zaneseny [12]. Při určování polohy vozidla nicméně nelze spoléhat pouze na GNSS, jelikož nemusí být vždy dostupné. Aby totiž tento systém fungoval, musí být zajištěna přímá viditelnost mezi přijímačem a družicí [8]. Problémy mohou nastat, pohybuje-li se vozidlo například mezi vysokými budovami a v tunelech tento systém nefunguje vůbec [8]. Je tedy nutná spolupráce s ostatními senzory, kterými je autonomní vozidlo vybaveno [12].

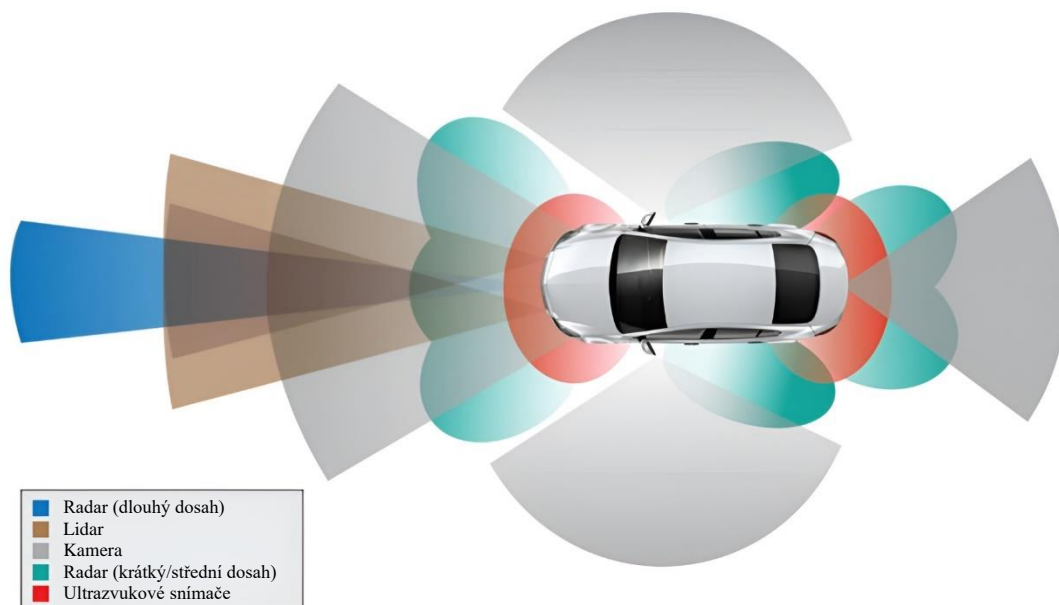
1.2.6 INERCIÁLNÍ MĚŘICÍ JEDNOTKA

Dalším senzorickým zařízením, které bývá součástí výbavy vozidel je inerciální měřicí jednotka (IMU). Součástí IMU jsou akcelerometr a gyroskop [8], které poskytují informace o příčném zrychlení vozidla a jeho náklonech vůči gravitačnímu poli Země, a to v obou případech ve třech osách [13]. V současné době jsou v automobilech díky svým kompaktním rozměrům inerciální používány měřicí jednotky typu MEMS [14].

Společně s GNSS se data z IMU používají k preciznímu určování polohy vozidla, přičemž klíčovou roli hrají v situacích, kdy dojde k přerušení spojení s družicemi [15]. Data o poloze z IMU a dalších senzorů, kterými je vozidlo vybaveno, jsou pak po dobu výpadku spojení využívána k výpočtu relativní změny polohy, díky čemuž je vozidlo schopné dále pokračovat v jízdě po předem stanovené trajektorii [15].

1.2.7 ROZMÍSTĚNÍ SENZORŮ NA VOZIDLE

Množství a rozmístění senzorů na vozidle závisí na konkrétním výrobcu. I tak je ale možné často nalézt jednotlivé typy senzorů na podobných místech. Příklad umístění senzorů na vozidle včetně schématického znázornění jejich zorných polí je uveden na *obr. 3*.



Obr. 3 Příklad umístění senzorů na vozidle a jejich zorná pole ([63], upraveno)

Autonomní vozidlo nicméně nemusí být vybaveno všemi typy senzorů, které jsou na obr. 3 znázorněny. Například společnost Waymo u svého autonomního taxi, postaveného na bázi vozu Jaguar I-Pace, používá pro detekci překážek v bezprostřední okolí místo ultrazvukových snímačů čtveřici lidarů, které jsou umístěny po obvodu vozidla. Toto vozidlo dále disponuje 360stupňovým lidarem umístěným na střeše, šesti radary namířenými do všech směrů a skupinou 29 kamer. [16]

1.3 SOFTWARE K TESTOVÁNÍ VOZIDEL VE VIRTUÁLNÍM PROSTŘEDÍ

Jelikož mohou autonomní vozidla potenciálně představovat velké nebezpečí jak pro posádku vozidla, tak i pro ostatní účastníky silničního provozu, je nutné je před vpuštěním do reálného provozu nejprve důkladně otestovat ve virtuálním prostředí. V následujících podkapitolách jsou popsány vybrané softwary využívané jak pro tvorbu virtuálních světů, tak i ty používané pro samotné testování ve virtuální prostředí, které spojuje možnost v nich využít světy vytvořené v softwaru RoadRunner.

1.3.1 FORMÁTY ASAM OPENX

Souhrnně jsou takto označovány otevřené datové formáty vyvíjené Asociací pro standardizaci automatizačních a měřících systémů (ASAM), určené k popisu virtuálních prostředí, určených k testování ADAS systémů a systémů autonomního řízení. V současné době se jedná o celosvětově rozšířené standardy [17]. Patří mezi ně například standard **OpenDRIVE**, který je určen k popisu silničních sítí, dále **OpenCRG** sloužící k detailnímu popisu povrchu silnic anebo **OpenSCENARIO**, který se používá k definování dynamického obsahu jízdních simulací, jako je například popis trajektorie vozidel [17]. Dále bude pozornost věnována jen prvnímu zmíněnému standardu, který nejvíce souvisí s tématem této práce.

Formát ASAM OpenDRIVE představuje způsob popisu silničních sítí, kdy uložená data nesou informaci o geometrii silnic, jízdních pruzích a objektech, které se v blízkosti silnice nacházejí [17]. Díky této standardizaci je možné data jednoduše přenášet mezi jednotlivými simulátory, což vede k zjednodušení vývoje, a tudíž i k poklesu nákladů [17]. Na druhou stranu jsou ale možnosti modelování prostředí v okolí silnic značně omezené a tento formát

navíc nenese žádnou konkrétní informaci o vzhledu daného prostředí [18]. Použitý simulátor tedy musí být schopný data vizuálně interpretovat.

Pomyslnou páteř silnic tvoří v tomto formátu tzv. referenční linie. Jedná se o matematicky popsanou křivku, která určuje tvar a výškový profil silnice. Kromě samotného výškového profilu silnice je dále možné definovat i klopení vozovky či detailní tvar jejího příčného průřezu. V případě jízdních pruhů je OpenDRIVE schopný nést informaci o jejich počtu, konfiguraci, šířce, směru jízdy či vodorovném dopravním značení na jejich okraji. Co se křižovatek týče, tak OpenDRIVE nedefinuje pouze místo, kde se jednotlivé silnice stýkají, ale zároveň i trajektorie, po kterých se mají vozidla danou křižovatkou pohybovat. Tato informace neslouží primárně pro testované vozidlo, ale pro simulování okolní dopravy. Dále je pomocí tohoto formátu možné stanovit polohu a velikost různých nepohyblivých objektů v prostoru silnice a jejím okolí, jako jsou například přechody pro chodce, parkovací místa, svodidla, semaforey či dopravní značky. Poloha jednotlivých objektů může být definována buď relativně vůči referenční linii nebo absolutně v globálním souřadnicovém systému. [19]

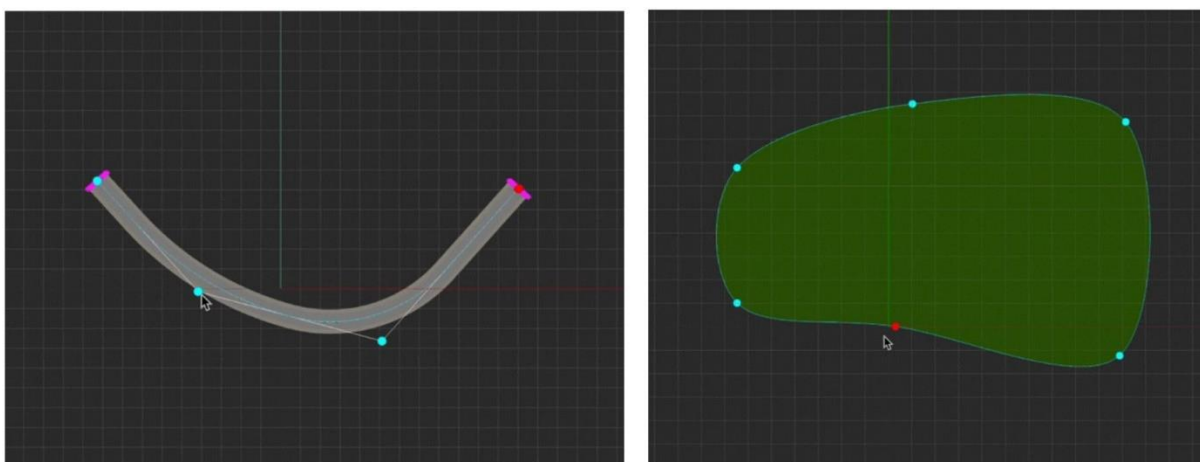
1.3.2 MATLAB ROADRUNNER

RoadRunner je placený editor od společnosti MathWorks, který slouží k tvorbě trojrozměrných virtuálních prostředí určených k simulaci a testování systémů autonomního řízení. Do pracovního prostředí RoadRunneru lze naimportovat vizuální podklady ve formě například leteckých snímků či výškových map, na základě kterých je uživatel schopen v tzv. „Scene Editoru“ pomocí různých nástrojů vytvářet realistické virtuální světy. To bylo předmětem této práce a je tomu věnována celá kapitola 2. Ne všechny dostupné funkce ale byly v rámci práce využity. RoadRunner dále nabízí například nástroje k tvorbě světelně řízených křižovatek či je zde nástroj pro vytváření mostů. Tvorbu realistických virtuálních prostředí usnadňuje vestavěná knihovna assetů, ve které je zahrnuto relativně velké množství 3D modelů od budov přes vegetaci až po dopravní značení. Silniční síť je možné buď vytvořit ručně pomocí dostupných nástrojů, nebo ji do pracovního prostředí naimportovat ze souboru ve formátu ASAM OpenDRIVE. Alternativně je možné použít nástroj „RoadRunner Scene Builder“, který je schopen silniční síť zrekonstruovat podle dat z podporovaných map ve vysokém rozlišení, tzv. HD map. RoadRunner dále umožňuje navrhovat scénáře pro simulace řízené MATLABem či Simulinkem. K tomu slouží prostředí RoadRunner Scenario. [20]

RoadRunner nabízí spoustu možností exportu vytvořených virtuálních světů. Jednak lze vytvořené světy exportovat jako sémantické mapy v datových formátech, jako je již zmíněný ASAM OpenDRIVE, jednak jako 3D geometrii ve formátu STL, OBJ či FBX [21]. Dále RoadRunner nabízí specializované nástroje určené k exportu vytvořených scén pro použití v nějakém konkrétním simulátoru, jakým může být například CARLA, nebo v herním enginu, kterým může být buď Unity nebo Unreal Engine [21]. Například výsledkem exportu do Unreal Enginu, což byl způsob využitý v této práci, pak byla kombinace souborů v různých formátech.

1.3.3 TRUEVISION DESIGNER

Dalším editorem, který lze využít k tvorbě virtuálních světů, je TrueVision Designer. Jeho předností je, že je pro nekomerční použití volně dostupný [22]. Samotné pracovní prostřední i nástroje jsou velmi podobné těm v RoadRunneru (viz *obr. 4*). Dostupných nástrojů je nicméně méně a jejich celková funkcionálna je v porovnání s RoadRunnerem omezená. To samé platí i pro možnosti exportu. Virtuální svět vytvořený tímto editorem je totiž možné vyexportovat pouze ve formátu ASAM OpenDRIVE [22].

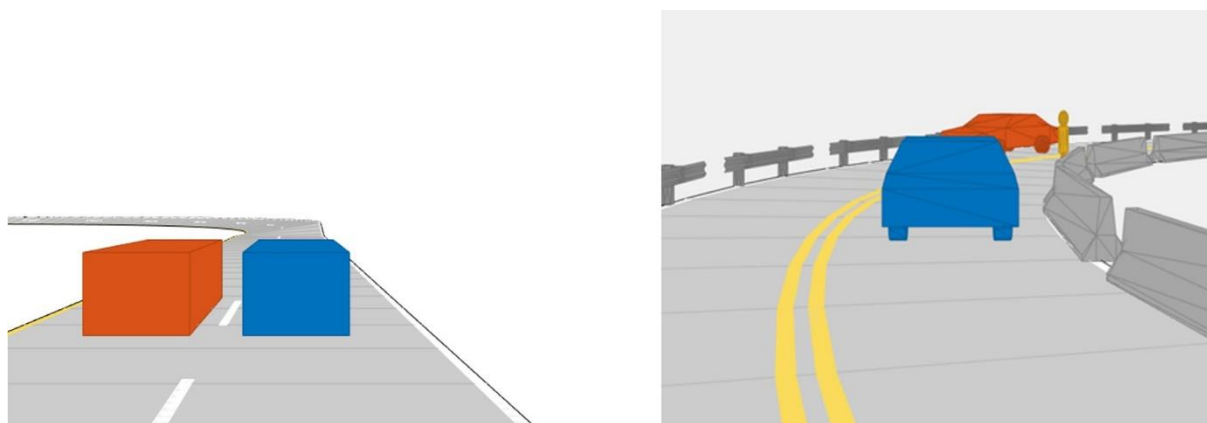


Obr. 4 Uživatelské rozhraní softwaru Truevision Designer [22]

1.3.4 AUTOMATED DRIVING TOOLBOX

Automated Driving Toolbox je doplněk k softwarům MATLAB a Simulinku, který obsahuje nástroje určené k vývoji, simulaci a testování ADAS systémů a systémů autonomního řízení [23]. Tento doplněk dokáže generovat kód v programovacích jazycích C a C++ [23], díky čemuž je možné navržené algoritmy využít i mimo ekosystém programů od MathWorks. Automated Driving Toolbox umožňuje uživateli provádět simulace s virtuálními senzory, jako jsou kamery, radar a lidar, a to ve třech různých simulačních prostředích [24].

Prvním simulačním prostředím je tzv. „**cuboid simulation environment**“, které se vyznačuje velmi jednoduchou grafikou. Jednotlivá vozidla jsou zde reprezentována barevnými kvádry o různých velikostech (viz obr. 5 vlevo) [24]. Je-li použit lidar, je geometrie vozidel složitější (viz obr. 5 vpravo) [24]. Virtuální prostředí a scénáře lze buď naprogramovat pomocí dostupných funkcí, nebo vytvořit interaktivně pomocí vestavěné aplikace „Driving Scenario Designer“. Alternativně lze silniční síť získat importem ze souboru ve formátu OpenDRIVE, nebo z podporovaných online map [25]. Do světa je pak možné umístit různé herce (vozidla, chodce, cyklisty), nastavit, kudy se mají pohybovat a vozidla opatřit senzory [26]. Vytvořené světy a scénáře je pak z této aplikace možné vyexportovat například ve formátech OpenX [26]. Navržené scénáře s vozidly, která mají nadefinovány trajektorie, je možné taktéž využít například v simulacích prováděných v Simulinku [26].



Obr. 5 Simulační prostředí Cuboid [64; 65]

Druhou možností je využít jako simulační prostředí **Unreal Engine**, který se propojí se Simulinkem. Unreal Engine dokáže renderovat detailní, realisticky vypadající trojrozměrné virtuální prostředí, ve kterém lze pomocí virtuálních senzorů získávat mnohem detailnější data o okolním prostředí [24], což v porovnání s prostředím „cuboid“ umožňuje provádět simulace, které se více blíží realitě. Virtuální vozidlo se zde opět může pohybovat po předem stanovené trajektorii, nebo může být řízeno algoritmem autonomního řízení podle dat z virtuálních senzorů [27]. Na druhou stranu je oproti prostředí „Cuboid“ náročnost na výpočetní výkon počítače výrazně vyšší. K testování v Unreal Engine je možné využít buď předpřipravené světy, které jsou součástí doplňku, nebo si vytvořit vlastní [27]. To lze teoreticky udělat přímo v editoru, který je součástí Unreal Engine, ale praktičtější je použít specializovaný program, jakým je právě RoadRunner.

Třetí možností je simulaci, kterou řídí MATLAB nebo Simulink, provádět v simulačním prostředí **RoadRunner Scenario** [24]. V něm je možné do virtuálního prostředí vytvořeného v „Scene Editoru“ umístit herce ve formě různých vozidel, kterým lze naprogramovat chování a opatřit je virtuálními senzory [28]. Jakmile simulace proběhne, je možné využít MATLABu k analýze získaných dat [28].

1.3.5 CARLA

CARLA je open source simulátor určený k vývoji, trénování a validaci funkčnosti systému autonomního řízení [29]. Simulace je zde řízena kódem v programovacích jazycích Python nebo v C++ [30]. Kód může uživatel upravovat a ovládat tak veškeré aspekty simulace, jako je například chování okolního provozu, chodců, počasí či senzorů, kterými jsou vozidla vybavena [29]. Samotná simulace pak probíhá ve virtuálním prostředí, které je renderováno pomocí Unreal Engine [30].

Středobodem simulace v simulátoru CARLA je tzv. „Ego Vehicle“. Jedná se o vozidlo, které uživatel vybaví virtuálními senzory a které následně projíždí virtuálním prostředím [31]. To se může pohybovat buď po předem stanovené trajektorii a získávat syntetická data pro strojové učení, nebo může být přímo řízeno algoritmem autonomního řízení [31]. Co se samotných senzorů týče, vozidlo může být opatřeno například různými druhy kamer, radarem, lidarem, IMU či GNSS senzorem [32].

Základem virtuálního světa, tzv. mapy, jsou silnice definované ve formátu ASAM OpenDRIVE, jejichž okolí je opatřeno 3D modely [33]. K testování je možné využít jednu z mnoha předpřipravených map, které jsou součástí simulátoru a které může uživatel dále podle potřeby upravovat, nebo si lze vytvořit zcela novou mapu a následně ji do CARLY nainportovat [33]. K vytváření nových map je doporučeno používat RoadRunner [34]. CARLA nabízí i experimentální nástroj „Digital Twin Tool“, který je schopen na základě dat z OpenStreetMap vygenerovat digitální dvojče vybrané reálné lokality [35].

1.3.6 NVIDIA DRIVE Sim

NVIDIA DRIVE Sim je platforma určená k vývoji a testování ADAS systémů a systémů autonomního řízení. Simulace v tomto případě probíhá na dvou oddělených serverech [36]. Na prvním z nich z nich běží samotný DRIVE Sim, který v reálném čase simuluje virtuální prostředí (viz obr. 6) a generuje výstupní data ze senzorů (kamery, radar, lidar, IMU atd.) umístěných na virtuálním vozidle [36; 37]. Tato data jsou následně odesílána na druhý server, jehož hardware a software odpovídá tomu, kterým je vybaven skutečný prototyp autonomního vozidla. Druhý server data vyhodnocuje a posílá zpět na první server instrukce k řízení

virtuálního vozidla [36]. Virtuální svět používaný při simulaci může být získán buď skenováním reálného prostředí, nebo může být vytvořen pomocí specializovaného editoru, jakým je například právě RoadRunner [38]. V průběhu simulace je bráno v potaz i to, že v reálném světě mají na data získávaná senzory vliv i další faktory, jako jsou například náklony karoserie při brzdění nebo zatáčení [37]. K tomu lze například použít výpočtový model jízdní dynamiky vozidel od společnosti IPG Automotive [39]. Použity ale mohou být i modely od jiných partnerských společností, jelikož DRIVE Sim je, co se týče výpočtových modelů, koncipován jako otevřená platforma, což platí nejen pro modely jízdní dynamiky, ale i pro výpočtové modely jednotlivých senzorů, provozu atd [39].



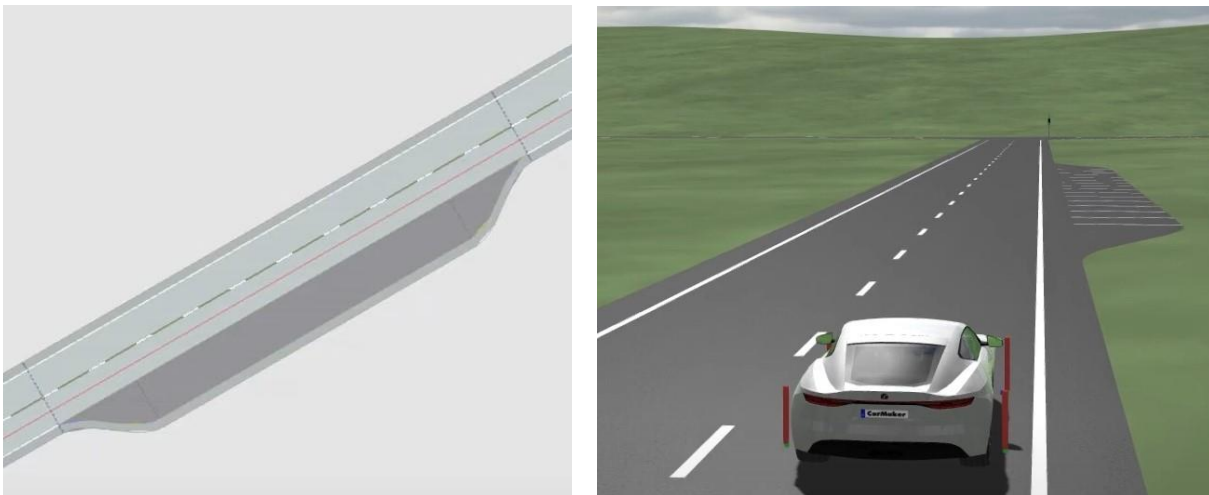
Obr. 6 Vzhled virtuálního světa v NVIDIA DRIVE Sim [66]

1.3.7 CARMAKER

CarMaker je komerční software vyvíjený společností IPG Automotive, který je určený nejen k vývoji a testování ADAS systémů a systémů autonomního řízení, ale i k simulaci chování pohonného traktu a jízdní dynamiky vozidel [40]. Předností tohoto softwaru je jeho škálovatelnost, tedy možnost provádět optimalizované simulace jak na méně výkonném hardwaru, jako jsou třeba notebooky, tak i na lokálních výpočetních clustrech nebo v cloudu [41]. Této flexibility lze dosáhnout díky tomu, že CarMaker disponuje výpočtovými modely o různých stupních složitosti [42]. Stejně jako DRIVE Sim je i CarMaker schopen věrně simulovat dynamiku vozidla, okolní provoz a samozřejmě také všechny běžně používané senzory včetně GNSS [43]. K vizualizaci virtuálního prostředí zde slouží grafický engine zvaný Movie NX, který zajišťuje realistické osvětlení virtuálního prostředí za jakéhokoliv počasí a denní doby [40]. Co se týče samotného virtuálního prostředí, lze jej do CarMakeru nahrát buď ve formátu OpenDRIVE, nebo si vytvořit vlastní pomocí vestavěného editoru, jehož součástí je i knihovna 3D modelů [43].

V tomto editoru je možné vytvářet silniční sítě, upravovat vzhled a typ povrchu (asfalt, dlažba atd.) silnice a vytvářet na něm nerovnosti [44]. Dále lze umisťovat vodorovné i svislé dopravní značení a semaforey a za účelem zkrášlení světa vytvářet terén v okolí silnic

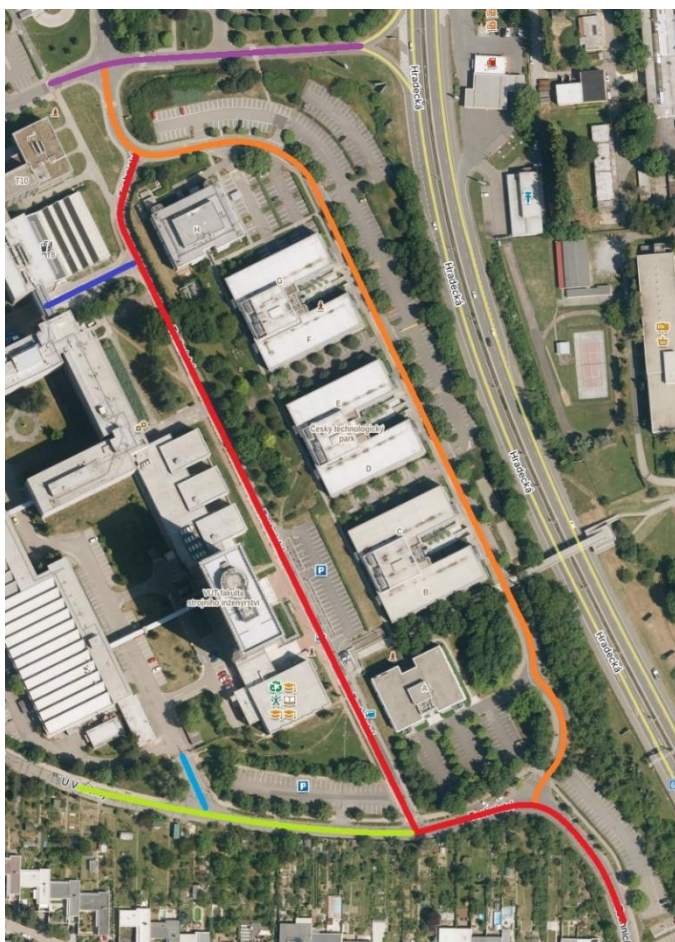
a umisťovat do něj různé objekty [44]. Na rozdíl od RoadRunneru se zde svět vytváří v 2D pohledu shora (viz *obr. 7*), přičemž vytvářené virtuální prostředí je v tomto pohledu po vizuální stránce značně zjednodušené. Kvůli fixnímu pohledu shora tedy musí být například výškový profil silnice vytvářen ručním nastavením číselné hodnoty výšky v jednotlivých bodech podél referenční linie [44]. V odděleném okně si je pak možné zapnout náhled výsledné podoby světa ve 3D, který je nicméně potřeba vždy po provedení změn v editoru aktualizovat [44]. Jsou tu i další rozdíly oproti RoadRunneru. Například při umisťování objektů jako jsou domy či lampy nestačí pouze kliknout do bodu, kde se má daný objekt nacházet, ale je třeba nadefinovat jeho přesnou relativní polohu vůči referenční linii vybrané silnice [44]. Terén je v tomto editoru vytvářen automaticky kolem umístěných silnic a dalších objektů, uživatel může nastavit maximální a minimální výšku v rámci celého vygenerovaného terénu [44]. V tomto ohledu jsou nástroje RoadRunneru mnohem flexibilnější. Na druhou stranu ale editor CarMakeru disponuje i funkcemi, které RoadRunner vůbec nenabízí. Jednou z nich je třeba možnost vytvářet tunely [44].









Obr. 7 CarMaker – Scenario Editor

2 NÁVRH VIRTUÁLNÍHO SVĚTA VYBRANÉ LOKALITY V ČR

Hlavním cílem této závěrečné práce bylo pomocí softwaru MATLAB RoadRunner vytvořit virtuální kopii zvolené reálné lokality na území České republiky, která bude své předloze detailně odpovídat. Vybráno bylo blízké okolí Fakulty strojního inženýrství Vysokého učení technického v Brně. Na *obr. 8* je letecký snímek této lokality, na němž jsou vyznačeny ulice, které jsou součástí vytvořeného virtuálního světa. V podkapitole 2.1 jsou uvedeny důvody, proč byla zvolena právě tato lokalita. K tvorbě virtuálního světa byl použit RoadRunner ve verzi R2023b.



	Technická – horní větev
	Technická – dolní větev
	U Vodárny
	příjezdová komunikace do areálu FSI z ulice U Vodárny
	příjezdová komunikace do areálu FSI mezi budovami A4 a T8
	ulice spojující Hradeckou s křižovatkou Technické a Podnikatelské

Obr. 8 Rozsah zvolené lokality ([67], upraveno)

Aby byla virtuální reprezentace zvolené lokality co nejvěrnější, byly při tvorbě práce využity vizuální podklady z mnoha různých zdrojů. Velmi užitečné byly letecké snímky dostupné na internetovém mapovém portálu Mapy.cz, nicméně nejšířší využití našla webová aplikace Google Maps, která kromě obyčejných satelitních snímků dává uživateli možnost si dané místo prohlédnout i ve 3D z ptáčích perspektiv. Součástí Google Maps je i funkce Street View, která byla nesmírně užitečná, jelikož je možné si pomocí ní prohlédnout sekvenci sférických fotografií, pořízenou speciálním vozidlem při průjezdu danou ulicí. Velkou výhodou je, že Street View kromě nejnovějších dostupných snímků z března roku 2020, umožňuje zobrazit i snímky z předchozích let. Ty sice již nejsou aktuální, nicméně dávají dobrou představu o tom, jak dané místo vypadá v letních měsících, což je období, do kterého je vytvářený virtuální svět zasazen. V samotném rozhraní softwaru RoadRunner byly velkou oporou

výšková mapa (viz *obr. 9*) a letecký snímek (viz *obr. 10*) z Geoportálu Českého úřadu zeměměřičského a katastrálního (ČÚZK), které si lze v pracovním prostředí RoadRunneru promítnout na pozadí. Nicméně i přes tuto poměrně širokou nabídku internetových podkladů bylo třeba lokalitu navštívit osobně a pořídit fotografie aktuálního stavu, jelikož daná lokalita se neustále vyvíjí a nedávné změny nemusí být na internetu vůbec zaznamenány.

2.1 VOLBA LOKALITY

Hlavním důvodem volby okolí FSI bylo to, že se jedná o známé a velmi dobře dostupné prostředí, kde bylo možné kdykoliv jednoduše zjistit, jak jednotlivá místa napodobovaná v RoadRunneru aktuálně vypadají. Další výhodou je, že se jedná o poměrně různorodé městské prostředí s velkým množstvím různých parkovacích ploch a výjezdů z areálů jednotlivých budov. Relativně úzkými komunikacemi této lokality dále pravidelně projíždí vozidla městské hromadné dopravy a prostředí je uzpůsobeno nejen pohybu motorových vozidel, ale i relativně velkému množství chodců a cyklistů. Vytvořené virtuální prostředí tedy lze potenciálně využít například k tvorbě širokého spektra možných scénářů, zahrnujících pohyb těchto osob. Jako příklad lze uvést situaci, kdy autonomní vozidlo jede za autobusem, který zastaví na zastávce a lidé z něj začnou vystupovat a přecházet silnici. Dalším příkladem by mohla být modelová situace, kdy cyklista jede po pravé straně silnice v jízdním pruhu pro cyklisty a v protisměru jede jiné vozidlo. Vozovka ale není dostatečně široká a autonomní vozidlo tedy musí počkat na správný okamžik, kdy může cyklistu bezpečně předjet.

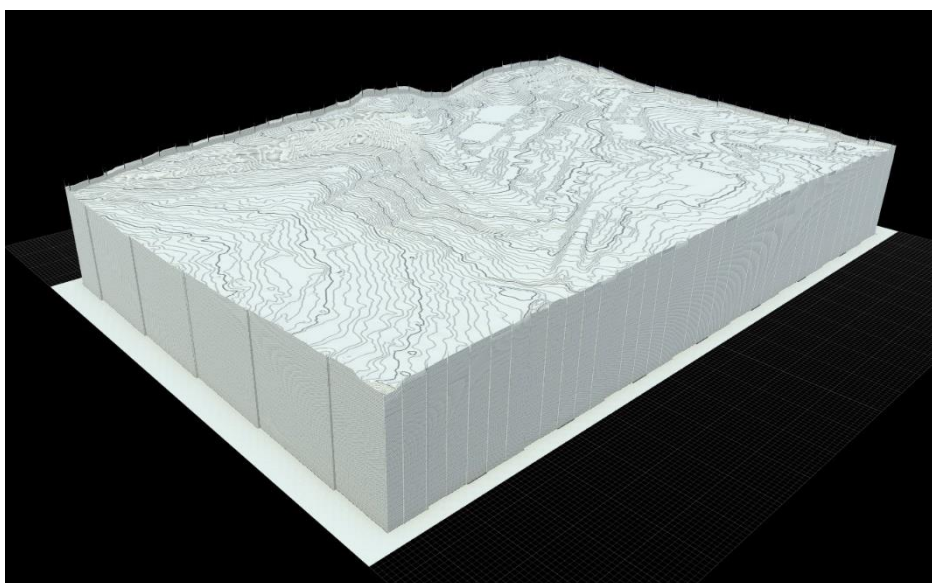
Volba této lokality s sebou ale nese i jisté nevýhody. Jelikož se jedná o městské prostředí, bylo pro dosažení požadované autentičnosti virtuálního světa potřeba relativně velké množství vlastních 3D modelů, což se ukázalo být časově náročnou činností. Vybraná lokalita taktéž nezahrnuje například žádné složité křižovatky či kruhové objezdy, což do jisté míry omezuje množství typů scénářů, které lze testovat.

2.2 MAPOVÉ PODKLADY

Jak již bylo zmíněno v úvodu této kapitoly, při vytváření světa byl využit letecký snímek a výšková mapa z Geoportálu ČÚZK [45]. Zde jsou jednotlivé produkty rozděleny na výdejní jednotky o velikosti v nižších jednotkách kilometrů čtverečních. K vypracování této práce byly využity produkty „Ortofotosnímky“ z roku 2022 ve formátu JPEG 2000 a produkt „Nadmořská výška (EL) – GRID“, což je digitální model reliéfu ve formátu TIFF. Tyto produkty byly dostupné zdarma ke stažení.

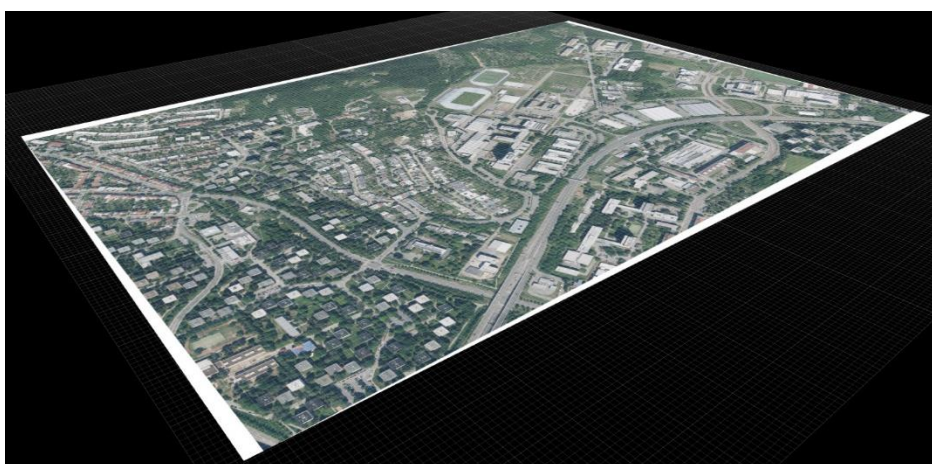
Před samotným importem stažených dat do pracovního prostředí bylo pomocí nástroje „World Settings Tool“ nejprve potřeba zvětšit pracovní prostor tak, aby se tam celé podklady vešly, jelikož výdejní jednotky z Geoportálu jsou výrazně větší než defaultní pracovní prostor.

Import výškové mapy (viz *obr. 9*) byl proveden pomocí nástroje „Elevation Map Tool“. Nejprve bylo třeba změnit typ souboru stažené mapy na „Elevation Map“, aby RoadRunner věděl, že se nejedná o obyčejnou texturu, ale právě o výškovou mapu. Stažené podklady jsou georeferencované, což znamená, že obsahují metadata nesoucí informaci o projekci mapy a o tom, kde na zemském povrchu se daná lokalita nachází. Díky tomu stačilo soubor s výškovou mapou jen přesunout z knihovny do pracovního prostoru a RoadRunner automaticky mapu umístil na správné místo. Poté už jen stačilo nastavit rozlišení mapy, které je v tomto případě 5 metrů na pixel [46].



Obr. 9 Výšková mapa

Import leteckého snímku (viz obr. 10) proběhl obdobným způsobem, tentokrát ale pomocí nástroje „Aerial Imagery Tool“. Defaultní typ souboru byl v tomto případě změněn na „Aerial Image“ a rozlišení bylo nastaveno na 0,125 metrů na pixel [47]. Na rozdíl od výškové mapy neobsahoval soubor leteckého snímku informaci o projekci. Vyřešeno to bylo převzetím této informace ze souboru výškové mapy. Kvůli tomu bylo poté třeba ověřit, že si obě mapy navzájem odpovídají, což se potvrdilo.



Obr. 10 Letecký snímek

2.3 TVORBA SILNIČNÍ SÍTĚ

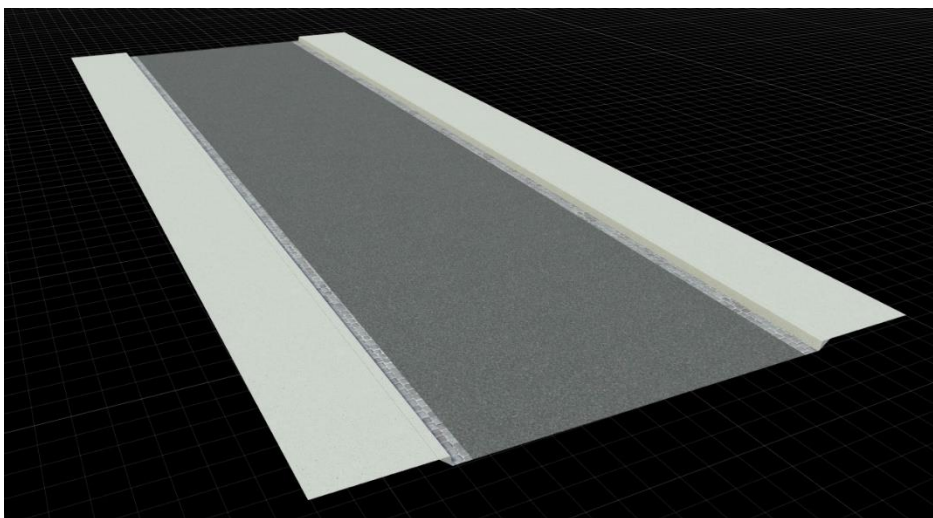
2.3.1 SILNICE

Umístění silnic je úplně prvním krokem v procesu tvorby virtuálního světa v softwaru RoadRunner. Jedná se totiž o páteřní prvky, kolem kterých celé virtuální prostředí vzniká. Je proto velmi důležité je umístit na správné místo, jelikož větší změna jejich polohy v pozdější části vývoje může být kvůli dalším prvkům umístěným kolem nich dost pracná a časově náročná.

Pro správné umístění silnic do prostoru byl klíčový podkladový letecký snímek. Ten není jen obyčejným pozadím, ale je programem automaticky promítán přes jednotlivé objekty ve světě, jako jsou právě silnice nebo třeba terény či budovy. Podstatné bylo, že takto promítaný snímek se prolíná s texturami umístěných objektů, díky čemuž je bylo možné do prostoru umisťovat s maximální přesností.

Aby bylo možné vytvořit novou silnici, je nutné si vybrat z knihovny styl, který definuje, jak bude tato nová silnice vypadat. To znamená, jaké je uspořádání jízdních pruhů, jaký je směr jízdy v těchto pružích, zda jsou určeny pro motorová vozidla nebo pro cyklisty a jestli je mezi nimi nějaký dělicí pás, případně jaký. Styl dále říká, zda je silnice lemována krajnicí či obrubníkem s chodníkem. Styl udává i to, jaká je šířka všech výše zmíněných částí tvořících danou komunikaci, jak je vysoký obrubník a jak má vypadat vodorovné dopravní značení. Je možné si vybrat buď styl z knihovny RoadRunner nebo použít vlastní. Vzhledem k tomu, že žádný defaultní styl nebyl vhodný, bylo třeba se vydat cestou tvorby vlastního. Ta spočívá ve vytvoření segmentu silnice v libovolném existujícím stylu a následné úpravě tohoto segmentu do požadované podoby pomocí dostupných nástrojů. Po provedení všech změn se nová podoba uloží jako nový styl.

Styl vytvořený pro použití při tvorbě základní sítě silnic je na *obr. 11*. Jeho základem jsou dva jízdní pruhy o šířce 2,7 m. Pro každý směr jízdy je určen jeden, přičemž na každý z nich navazuje krajnice široká 30 cm. To dohromady dává šířku silnice 6 m, což je hodnota získaná pomocí nástroje měření vzdálenosti v Google Maps. Krajnice je zde z toho důvodu, že v některých místech je asfalt od obrubníku oddělen pásem dlažebních kostek. Tuto skutečnost lze ve virtuálním světě reflektovat právě díky samostatné krajnici, které je možné nezávisle na povrchu vozovky změnit texturu. V místech, kde dlážděný pás není, zůstala zachována původní asfaltová textura. Krajnice je na obou stranách lemována betonovým obrubníkem. Součástí stylu jsou také 1,6 metru široké chodníky na obou stranách silnice.

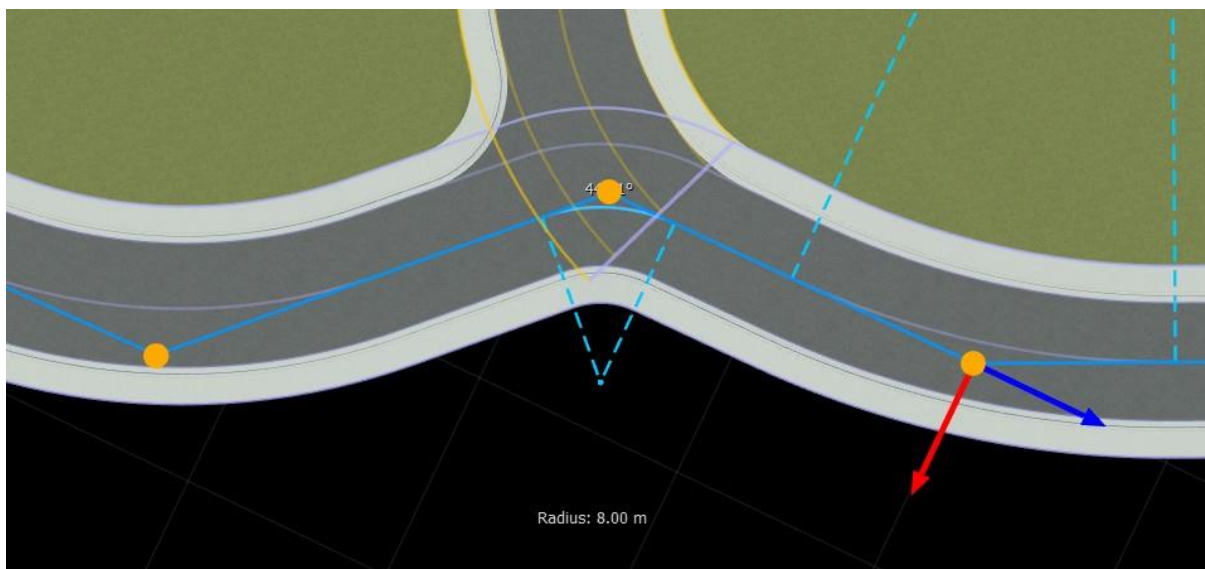


Obr. 11 Styl použitý pro základní síť silnic

K umisťování nových silnic slouží nástroj „Road Plan Tool“. Obecně se v RoadRunneru nové objekty do virtuálního světa přidávají kliknutím pravým tlačítkem do bodu, kde se má nový objekt nacházet. U tohoto nástroje se pravým kliknutím vkládají řídicí body, které definují tvar referenční křivky silnice. Tu je možné vidět například na *obr. 12* a *obr. 13*, kde je vyobrazena jako fialová křivka jdoucí středem silnice. Podle ní poté RoadRunner silnici

automaticky vygeneruje. Nejprve byly vytvořeny přímé segmenty silnic. K tomu stačí určit pouze dva řídicí body, počáteční a koncový. Přímé segmenty byly poté propojeny segmenty zakřivenými, jejichž referenční křivka musí být v automatickém režimu definována alespoň třemi body. Vytvořením segmentů v tomto pořadí bylo zamezeno nechtěnému zakřivení přímých úseků silnic z důvodu toho, že se RoadRunner vždy snaží docílit toho, aby na sebe referenční křivky sousedních úseků tečně navazovaly, kvůli čemuž někdy mění geometrii již existujících segmentů. Takto byla na základě podkladového leteckého snímku vytvořena hrubá kostra silniční sítě vybrané lokality, jež byla následně precizně upravena tak, aby co nejdříve odpovídala reálné předloze, čehož bylo dosaženo buď posouváním a otáčením celých segmentů, nebo úpravou polohy individuálních řídicích bodů.

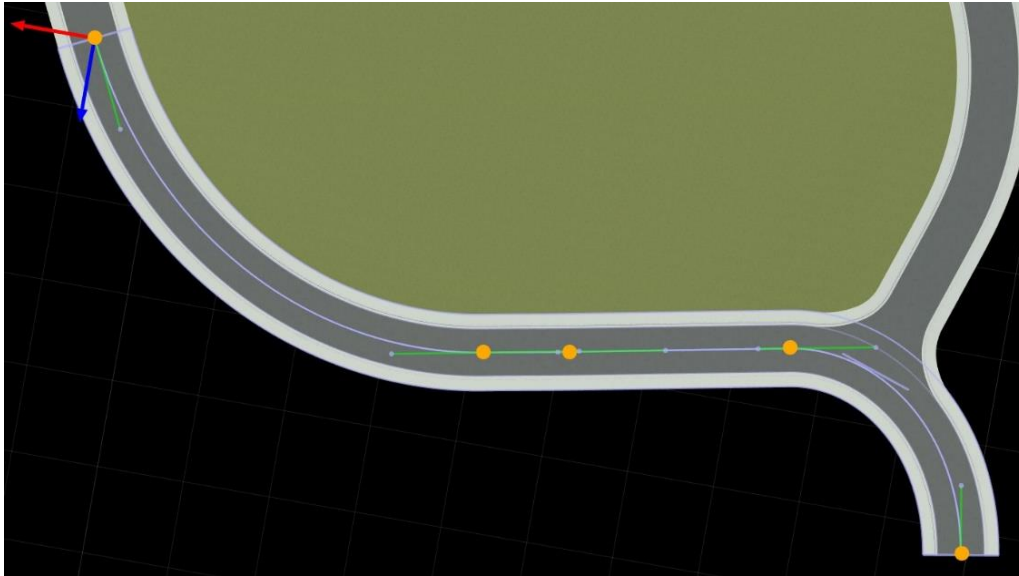
Nástroj „Road Plan Tool“ silnice defaultně vytváří v tzv. automatickém režimu. V tomto režimu RoadRunner spojí sousední řídicí body (na *obr. 12* vyznačeny oranžově) přímkami, čímž vznikne lomená čára. Referenční křivka silnice tuto čáru následuje, přičemž zakřivení vzniká v okolí řídicích bodů (s výjimkou počátečního a koncového), kde vytváří v podstatě zaoblení rohů lomené čáry. Jsou-li řídicí body blízko u sebe, jednotlivá zaoblení na sebe plynule navazují. V opačném případě mezi nimi vznikne přímý úsek. Po kliknutí na konkrétní řídicí bod, lze v panelu atributů nastavit maximální poloměr zaoblení, čímž je možné vytvořit prudší zatáčku. Toho bylo využito například v místě, kde se na ulici Technickou napojuje příjezd k parkovišti u kancelářské budovy č. 2851/15 (viz *obr. 12*). Zde byl maximální poloměr nastaven na 8 metrů.



Obr. 12 Segment silnice v automatickém režimu

Ve většině případů byly výše popsané úpravy dostačující pro dosažení uspokojivé přesnosti tvaru silnice. V některých případech, jako například u esovitě zakřivené části ulice Technické (viz *obr. 13* a *obr. 47-49* v příloze A), to nestačilo a bylo třeba provést konverzi daného segmentu silnice do tzv. explicitního režimu. Provede-li se tato operace, RoadRunner na daném segmentu silnice vytvoří nové řídicí body tak, aby byl zachován původní tvar silnice. Nové body leží už přímo na referenční křivce, která již není definovaná poloměry zaoblení, ale směrnicemi tečen v jednotlivých řídicích bodech. Směrnice tečen je možné libovolně upravovat a dále lze i manuálně přidávat nové řídicí body či odebírat body existující, což umožňuje vytvářet zcela libovolné tvary silnic.

Problém nastává, je-li třeba provést konverzi zpět do automatického režimu, jelikož to ve většině případů způsobí nežádoucí změnu tvaru silnice, a to i v případě, že v automatickém režimu nedošlo k žádným změnám. Proto byla konverze do explicitního režimu prováděna až tehdy, když bylo jasné, že již nebudou potřeba žádné další výraznější změny obecného tvaru daného segmentu a budou provedeny pouze mírnější korekce tvaru referenční křivky.

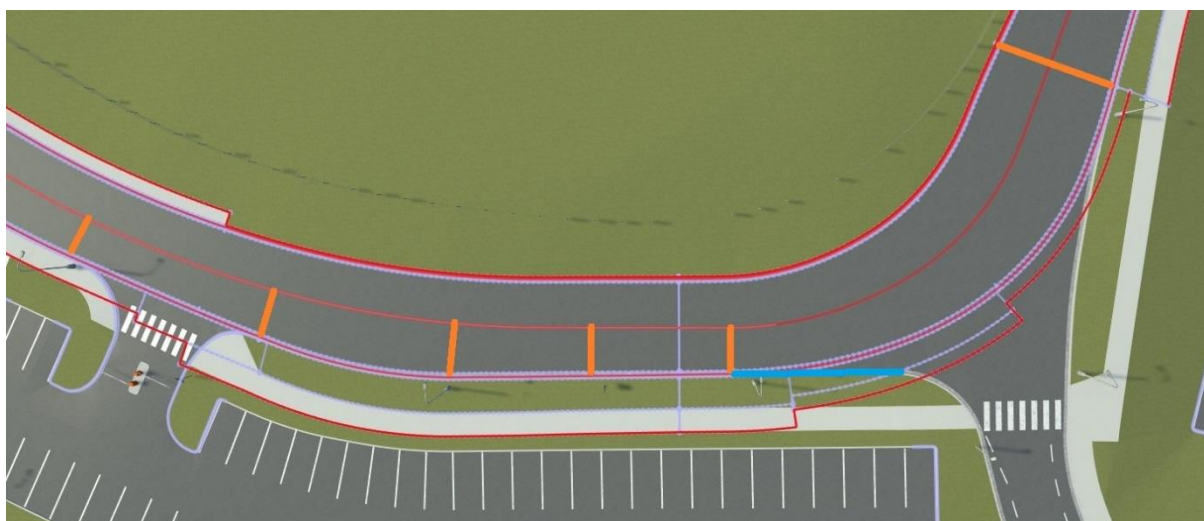


Obr. 13 Segment silnice v explicitním režimu

Způsobem popsaným v odstavcích výše byla vytvořena základní síť silnic (viz obr. 8), tvořená ulicí Technickou od křižovatky s ulicí Horskou až po její konec na křižovatce s ulicí Podnikatelskou. Zahrnuta je jak její horní větev Technické vedoucí kolem výškové budovy A1 (viz obr. 41-46 v příloze A), tak dolní větev, vedoucí níže podél ulice Hradecké (viz obr. 37 a obr. 47-52 v příloze A). Dále byla součástí základní sítě ulice spojující křižovatku Technické a Podnikatelské s ulicí Hradeckou a část ulice U Vodárny od jejího počátku na křižovatce s horní větví Technické (viz obr. 39 v příloze A) po křížení s příjezdovou komunikací do areálu FSI (viz obr. 40 v příloze A), která byla taktéž zahrnuta. Dále byla součástí základní sítě i příjezdová komunikace mezi budovou A4 FSI a budovou T8 Fakulty elektrotechniky a komunikačních technologií (viz obr. 45 v příloze A).

Jakmile byla základní síť hotová v rovině, bylo potřeba všem řídicím bodům nastavit správnou výšku, což byl díky stažené výškové mapě jednoduchý proces. Stačilo označit všechny řídicí body a v rámci „Road Plan Tool“ použít funkci „Project Roads“, která automaticky přiřadila všem řídicím bodům výšku na základě dat z naimportované výškové mapy. Poté bylo ještě nicméně potřeba u některých bodů upravit výšku manuálně pomocí nástroje „Road Height Tool“, jelikož rozlišení výškové mapy je v porovnání s leteckým snímkem znatelně nižší, a tudíž i přesnost dat není nejlepší. V důsledku toho byly některé silnice nepřirozeným způsobem mírně zvlněné, což bylo potřeba ručně opravit. Pomocí „Road Height Tool“ je dále možné nastavit, zda bude k vytvoření výškového profilu silnice mezi jednotlivými řídicími využita kvadratická nebo kubická interpolace. Použití kubické interpolace umožňuje měnit směrnici tečny k výškovému profilu v jednotlivých řídicích bodech, díky čemuž se lze více přiblížit realitě. Na druhou stranu ale častěji vede ke vzniku nepřirozené vlnitosti, a je tudíž méně vhodná pro použití při nižší kvalitě výškových dat [48]. Proto byla využita především interpolace kvadratická.

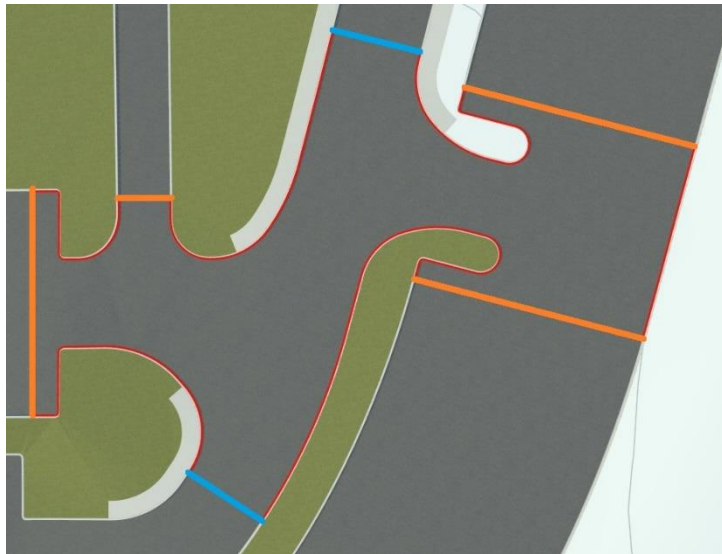
V neposlední řadě bylo ještě třeba upravit šířku silnic tam, kde se liší od výchozí šířky ve stylu použitém pro tvorbu základní sítě. K tomu byl využit nástroj „Lane Width Tool“, kterým lze měnit šířku jednotlivých pruhů, a v důsledku toho i celé silnice. Pro změnu šířky pruhu po celé délce jednoho segmentu silnice stačilo kliknout na přerušovanou čáru na vnějším okraji daného pruhu a v panelu atributů změnit jeho šířku. Tímto způsobem byla například ulice U Vodárny rozšířena na 7 metrů, a naopak příjezdová komunikace mezi budovami A4 a T8 zúžena na 5 metrů. Naproti tomu šířka dolní větve ulice Technické byla změněna přidáním pruhů pro cyklisty o šířce přibližně 1 metr na obě strany silnice pomocí nástroje „Line Add Tool“ a následným zúžením původních jízdních pruhů. Šířka silnic byla změřena buď pomocí nástroje měření vzdálenosti, který je součástí Google Maps, nebo přímo na promítnutém podkladu v pracovním prostředí RoadRunneru pomocí nástroje „Measure Tool“. V případě, že se šířka pruhu, potažmo silnice, podél jednoho segmentu mění, je možné opět pomocí nástroje „Lane Width Tool“ individuálně změnit šířku buď v krajích bodech, nebo lze kliknutím pravého tlačítka vytvořit uvnitř segmentu nové řezy a změnit šířku v těchto řezech. (na obr. 14 jsou tyto řezy vyznačeny oranžově). Tato metoda byla využita například v okolí křižovatky, kde se dolní větev ulice Technické napojuje na větev horní. Zde se ulice před křižovatkou ve směru of fakulty plynule rozšiřuje. Naopak na druhé straně křižovatky dochází nejprve k rozšíření a poté k následnému zúžení jízdního pruhu, který je blíže k přilehlému parkovišti. (viz obr. 14 a obr. 38 v příloze A).



Obr. 14 Okolí křižovatky, kde se dolní větev ulice Technické napojuje na větev horní

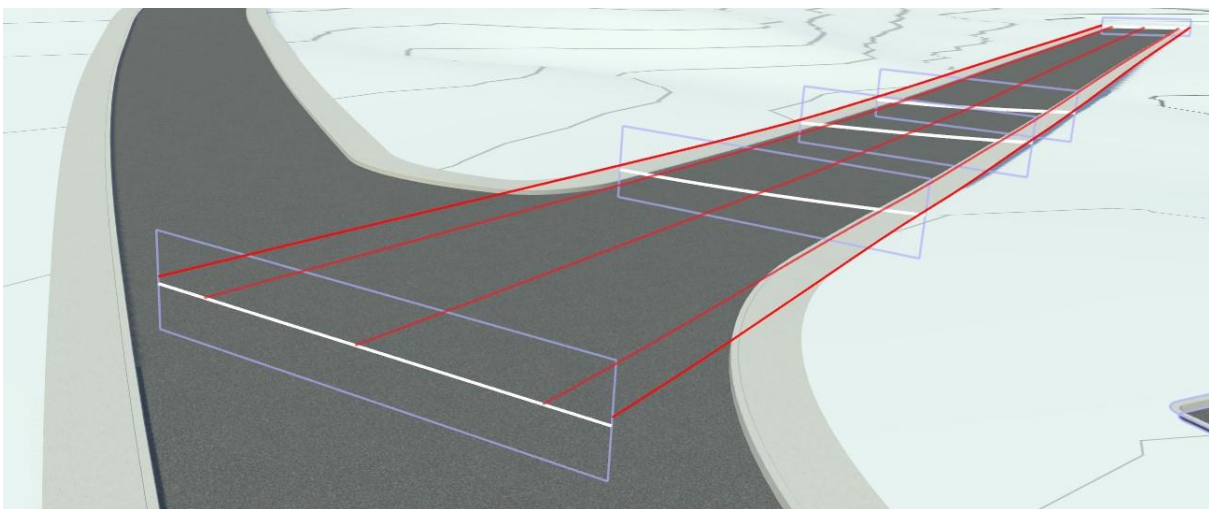
2.3.2 KŘÍŽOVATKY

Křižovatky RoadRunner vytváří automaticky, pokud se překrývají dva různé segmenty silnic s žádným nebo malým výškovým rozdílem. Některé křižovatky tedy vznikly již společně se základní sítí silnic. Ty ale bylo nicméně ještě potřeba dále upravit tak, aby co nejvíce odpovídaly realitě. Křižovatky je také možné vytvářet manuálně pomocí nástroje „Custom Junction Tool“. Ten umožňuje křižovatkou propojit buď volné konce jednotlivých segmentů silnic (v obr. 15 vyznačeny oranžově), nebo napojit volné konce na výseč již existující silnice (v obr. 15 vyznačena modře). Tento nástroj byl využit především při tvorbě křižovatek složitých tvarů, jakými byla například napojení parkovišť na průběžné silnice. Příklad takového napojení je právě na obr. 15.



Obr. 15 Příklad manuálně vytvořené křižovatky

Před úpravou samotný tvar křižovatky, je potřeba u křižovatek ve svahu zajistit, že po napojení příčné silnice bude povrch křižovatky dostatečně hladký a nevzniknou žádné schody či ostré hrany. To byl problém křižovatky ulice Technické s ulicí U Vodárny (viz obr. 39 v příloze A) a sousedního napojení příjezdové komunikace do areálu FSI (viz obr. 40 v příloze A). V případě první zmíněné křižovatky stačilo pro dosažení uspokojivého výsledku pomocí nástroje „Road Superelevation Tool“ natočit koncový průřez příčné silnice o 2° , jelikož se tato silnice napojuje pod úhlem, který není příliš ostrý. V případě druhé silnice, která se napojuje pod ostřejším úhlem, již toto nestačilo a bylo kromě natočení koncového průřezu potřeba pomocí nástroje „Cross Section Tool“ změnit tvar příčného průřezu vozovky v několika bodech koncového segmentu příčné silnice (viz obr. 16).

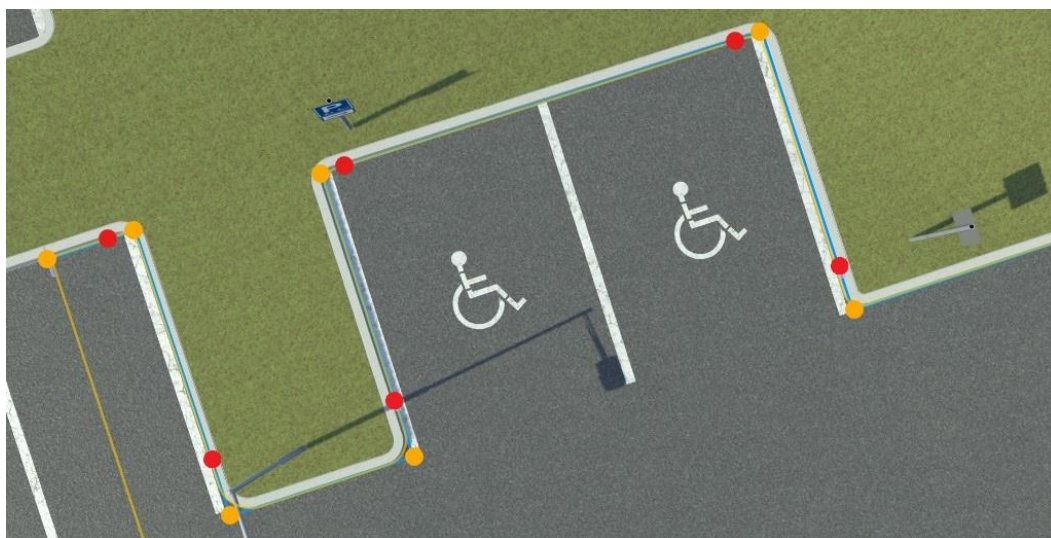


Obr. 16 Napojení příjezdové komunikace do areálu FSI na ulici U Vodárny

Dalším nástrojem, který byl využit k vyhlazení povrchu křižovatek napříč celým vytvářeným virtuálním světem, byl „Junction Surface Tool“, který umožňuje u jednotlivých ploch v křižovatce v panelu atributů nastavit, zda bude tato plocha při triangulaci povrchu křižovatky zohledněna či nikoliv.

Změna samotného tvaru křižovatek byla prováděna úpravou tvaru jejich rohů pomocí nástroje „Corner Tool“, který lze použít jak na křižovatky automatické, tak na ty vytvořené manuálně. Prvním krokem v procesu změny bylo nastavení poloměru zaoblení rohů na požadovanou hodnotu v metrech. V některých případech, jako například u napojení příjezdové komunikace do areálu FSI na ulici U Vodárny (viz *obr. 40* v příloze A), byla pouhá změna poloměru rohů dostačující k dosažení požadovaného tvaru. Nicméně u valné většiny křižovatek bylo třeba využít i druhou funkci tohoto nástroje, kterou je možnost upravovat řídicí body rohu, jenž definují jeho tvar. Obdobně jako u silnic lze přidávat nové, rušit existující a měnit jejich polohu, ale je zde více omezení. Mezi krajními body musí být v automatickém režimu alespoň další dva, přičemž řídicí body nejbliže ke kraji křižovatky je možné posouvat pouze po tečně k příslušnému segmentu silnice v krajním bodě křižovatky. To může být v některých případech omezující, nicméně nastaly i situace, kdy se toho, společně s možností posouvat krajní bod křižovatky podél silnice, dalo velice dobře využít. Příkladem takové situace je křižovatka na *obr. 14*. Zde byl modře vyznačený přímý úsek obrubníku vytvořen posunutím krajního bodu rohu křižovatky až k prvnímu oranžově vyznačenému řezu. Tímto způsobem bylo dosaženo mnohem rychleji výrazně lepšího výsledku, než kdyby měly být vytvářeny další řezy pomocí „Lane Width Tool“.

Dalším omezením nástroje „Corner Tool“ je, že na rozdíl od silnic není možné individuálně měnit poloměr zaoblení v jednotlivých řídicích bodech a míra zaoblení se tak řídí pouze vzdáleností od sousedních řídicích bodů. Platí, že čím blíže k sobě tyto body jsou, tím má zaoblení menší poloměr. Kvůli tomu ani nebylo možné některé potřebné tvary v automatickém režimu vytvořit, a bylo nutné poměrně často přepínat do režimu explicitního, který zde funguje stejným způsobem jako u silnic. To bylo třeba především u manuálně vytvořených křižovatek, které slouží jako napojení parkovišť na průběžné silnice. Zde se nachází rohy s poměrně velkými poloměry zaoblení, které nebylo možné v automatickém režimu vytvořit. Vyskytla se ale i jedna specifická situace, kdy bylo možné toto omezení využít ku prospěchu. Konkrétně na parkovištích bylo často potřeba vytvořit ostrý roh, k čemuž stačilo pouze pravým kliknutím vytvořit jeden nový řídicí bod blízkosti bodu, kde se má ostrý roh nacházet. Na *obr. 17* jsou tyto nové body vyznačeny červeně. Ostatní řídicí body jsou vyznačeny oranžově.



Obr. 17 Příklad obrubníku s ostrými rohy

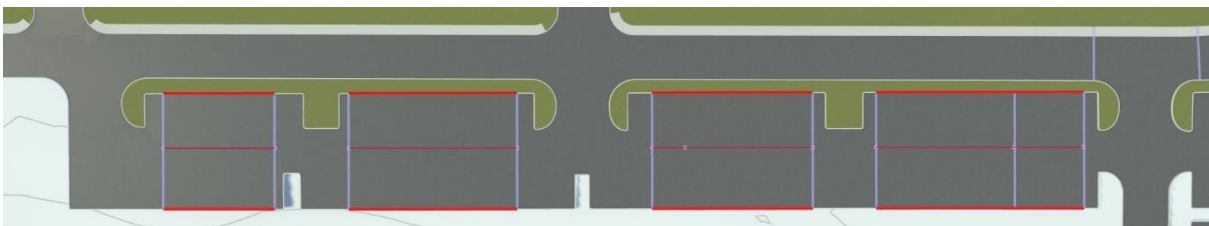
2.3.3 PARKOVIŠTĚ

Ve většině případů tvorba parkovišť spočívala ve spojování relativně krátkých úseků silnic pomocí manuálně vytvořených křižovatek, k čemuž byly opět v hojné míře využívány nástroje a metody popsány v předchozích podkapitolách. Ale vzhledem k tomu, že parkoviště poměrně často nabývají relativně složitých tvarů, bylo potřeba použít i metody nové.

Parkovacích ploch je ve zvolené lokalitě velké množství, a vyplatilo se tedy vytvořit nový styl silnice. Jako výchozí byl použit styl vytvořený pro základní síť, na kterém byly provedeny dvě změny. Za prvé byl po obou stranách silnice mezi jízdní pruh a krajnicí přidán 5 metrů široký pás asfaltu tvořící parkovací místa a za druhé mezi průběžnými pruhy přibyl asfaltový dělicí pás. Jeho defaultní šířka byla nastavena na 50 cm, nicméně ta byla ve většině případů podle potřeby změněna. Vzhledem tomu, že dělicí pás má stejnou texturu jako průběžné pruhy, není na první pohled vidět. Přesto ale hraje důležitou roli, které bude věnován prostor dále v rámci této podkapitoly.

Prvním krokem při tvorbě parkovišť bylo zhotovení samotných parkovacích ploch pomocí výše popsaného nového stylu, přičemž v této fázi nebyl brán ohled na křižovatky či na výklenky pro stromy a další podobné detaily. Dále byla parkovacím plochám pomocí podkladového leteckého snímku nastaven správný tvar a jednotlivým řídicím bodům přiřazena správná výška podle výškové mapy. Toto bylo provedeno metodami, které již byly detailně popsány v předešlých podkapitolách věnovaných silnicím a křižovatkám. Dále byly pomocí nástroje „Lane Width Tool“ podle potřeby upraveny šířky jak pásů dělicích, tak těch, na kterých se nachází parkovací místa.

V druhém kroku byl využit nástroj „Road Chop Tool“, který umožňuje v libovolném bodě již existujícího segmentu silnice vytvořit řez, který původní segment rozdělí na dva menší. Takovéto řezy byly vytvořeny v místech, kde se následně budou nacházet hranice křižovatek a dále jimi byly vymezeny úseky, kde mají obrubníky složitý tvar, tedy například místa, kde se nacházejí již zmíněné výklenky pro stromy. Vytvořené řezy jsou na *obr. 18* vyznačeny fialově. Následně byly vymezené úseky smazány, přičemž poloha, tvar ani výška zbylých segmentů se nezměnily. Na *obr. 18* jsou tyto segmenty zvýrazněny červeně.



Obr. 18 Příklad parkovací plochy

V následujícím kroku byly takto vzniklé separátní segmenty propojeny s průběžnými silnicemi a samy se sebou pomocí již popsaného nástroje „Custom Junction Tool“, přičemž tvar těchto nově vzniklých křižovatek byl upraven opět tak, aby co nejvěrněji reflektoval realitu. Jednoduché spojky mezi jednotlivými separátními segmenty rozhodně nejsou křižovatkami v pravém slova smyslu, nicméně RoadRunner je za ně považuje, a tudíž i přímé obrubníky po okrajích spojek vnímá jako rohy křižovatky. To znamená, že je možné libovolně měnit jejich tvar a vytvářet tak již zmíněné výklenky pro stromy, ale i jakékoliv jiné tvary. V této fázi byl velice užitečný nástroj „Measurement Tool“, který umožňuje ve vytvářeném

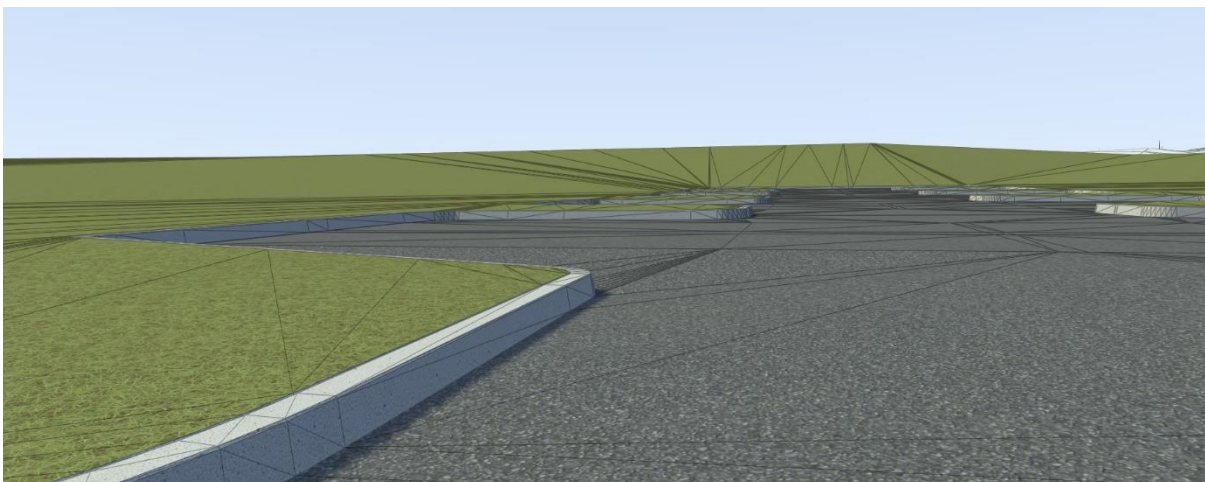
virtuálním světě detailně měřit vzdálenosti a úhly. Naměřené rozměry mají podobu fialových čar, u kterých je zobrazen příslušný rozměr (viz *obr. 19*). Existující čáry je dále možné kopírovat a vytvořit tak například rovnoběžky. Zásadní je, že tyto čáry zůstávají viditelné i poté, co uživatel „Measurement Tool“ vypne a pokračuje v práci s jinými nástroji, a je tedy možné je použít jako vodítka. *Obr. 19* byl pořízen v okamžiku, kdy už byl výklenek v pravém horním rohu podle vodítek namodelován. Tvar těch zbývajících byl zatím pouze vodítky vyznačen na asfaltové ploše.



Obr. 19 Použití vodítek při tvorbě parkovací plochy

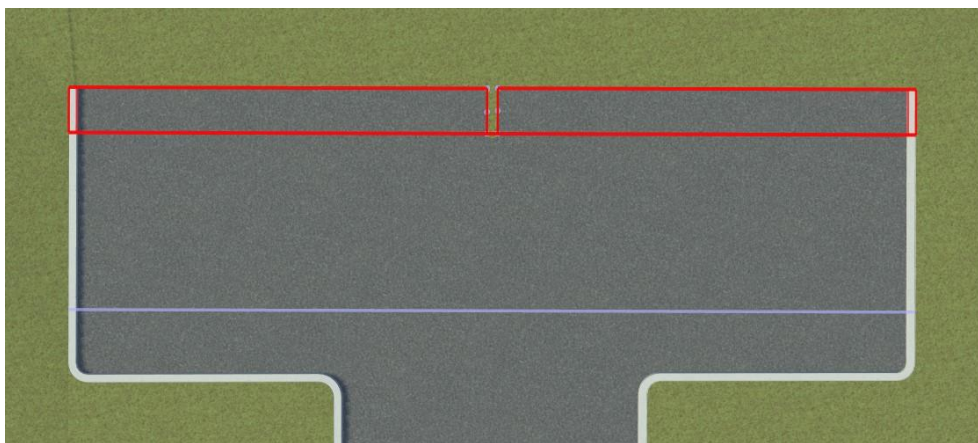
U parkovacích ploch umístěných ve svahu bylo nutné po úpravě tvaru obrubníků do požadované polohy provést ještě jeden dodatečný krok, a to pomocí nástroje „Road Height Tool“ správně nastavit jejich výšku. V okolí částí obrubníků, jejichž řídicí křivka má podobný směr jako vrstevnice svahu v daném místě, totiž dochází ke vzniku nepřirozených nerovností asfaltového povrchu v důsledku toho, že obrubník buď vyčnívá nad okolní asfalt, nebo se naopak pod něj noří (viz *obr. 20*). K vyrovnaní povrchu většinou nestačilo jen upravit výšku jednotlivých již existujících řezů po délce daného obrubníku, ale bylo třeba vhodně umístit řezy nové nebo odstranit ty původní. Nicméně aby bylo možné výšku tímto způsobem měnit, bylo nutné nastavit, že výškový profil obrubníku má být vytvářen pomocí kubické interpolace, a nikoliv kvadratické. Druhá jmenovaná, která je v RoadRunneru nastavená jako výchozí, totiž výše popsané úpravy neumožňuje.

Nakonec bylo potřeba vyřešit, jakým způsobem na parkovištích vytvořit slepé konce, kterých je ve vybrané lokalitě velké množství. Je-li segment silnice vytvořen klasickým způsobem a z jedné strany na něj nebude nic napojeno, tak sice vznikne slepý konec, a RoadRunner kolem něj dokonce vytvoří i terén, nicméně obrubník tam už automaticky nevznikne. A jelikož vozovka a horní hrana obrubníku jsou v různé výšce, tak na konci slepého segmentu v místech, kde končí obrubníky, vzniknou v terénu průhledné díry. S využitím výhradně nástroje „Custom Junction Tool“ toho taktéž nebylo možné dosáhnout, jelikož pro manuální vytvoření nové křižovatky jsou potřeba alespoň dva nezávislé okraje silnic.



Obr. 20 Příklad obrubníku zanořeného pod okolní povrch

Slepý konec tedy bylo nutné vytvořit následujícím způsobem. Nejprve byla při pohledu shora v ortografickém režimu, který nezkrsluje vzdálenosti, pomocí nástroje „Measurement Tool“ vytvořena vodítka mající požadovaný tvar daného slepého konce. Poté byl využit nástroj „Slip Road Tool“, který umožňuje silnici rozdělit tak, že se každý pruh stane samostatným segmentem (viz obr. 21). Tyto segmenty nemusí být pro tento účel nikterak dlouhé. Délka 1 m byla zcela dostačující. Krátké segmenty totiž slouží pouze k tomu, aby vznikly dva volné okraje, které již manuálně vytvořenou křižovatkou propojit lze, což bylo učiněno pomocí již zmíněného nástroje „Custom Junction Tool“.



Obr. 21 Krátké segmenty vytvořené nástrojem „Slip Road Tool“

Následně byl tvar obrubníku takto vytvořené „křižovatky“ podle nachystaných vodítek upraven do požadované podoby. Aby nevznikly nerovnosti podobné těm na obr. 20, byl zapnut režim pohledu zvaný „Shaded Wireframe“, jenž zvýrazní obrys jednotlivých polygonů, které tvoří povrch silnice. V tomto režimu pohledu je lépe vidět, kde vznikají v okolí obrubníku nerovnosti, a je tak jednodušší je změnou výšky obrubníku odstranit. Manuálním vytvořením křižovatky dále vznikne mezi krátkými segmenty uzavřená smyčka, kterou RoadRunner automaticky vyplní terénem. Defaultně má terén uvnitř této smyčky texturu trávníku. Ta byla vyměněna za stejnou asfaltovou texturu, jakou mají silnice, čímž s nimi smyčka vizuálně dokonale splynula.

2.4 TVORBA OSTATNÍCH PLOCH V OKOLÍ SILNIC

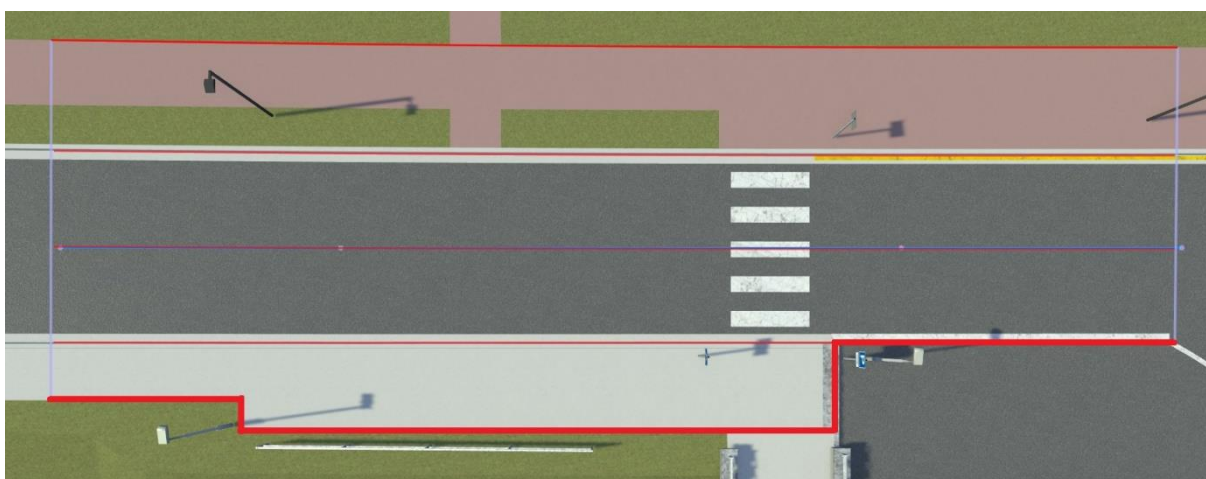
2.4.1 CHODNÍKY

Chodníky jsou součástí stylu silnice použitého k tvorbě základní silniční sítě, a vznikly tedy už v předešlých fázích tvorby virtuálního světa. Ve vybrané lokalitě ale není příliš mnoho míst, kde by ve skutečnosti vypadaly stejně jako v tomto stylu. Často tedy bylo třeba provést úpravy tak, aby chodníky svým vzhledem co nejvíce odpovídaly své reálné předloze, nebo naopak bylo nutné celý chodník smazat, pokud se v daném místě vůbec nenachází.

V nejjednodušších případech stačilo pro dosažení požadované podoby pouze změnit šířku chodníku. Například chodník, lemující příjezdovou cestu do areálu FSI u budovy A4 (chodník vpravo na *obr. 45* v příloze A), stačilo pouze pomocí nástroje „Line Width Tool“ po celé délce rozšířit na 2,9 m, což dohromady s obrubníkem dává šířku přibližně 3 m, změřenou v Google Maps.

Často ale není šířka chodníku po celé délce segmentu silnice stejná. V takových případech bylo nutné využít nástroj „Lane Chop Tool“, který umožňuje v libovolných bodech pruhu (chodníky považuje RoadRunner za pruhy typu „Sidewalk“) vytvořit řezy a rozdělit ho tak na několik samostatných segmentů, které mohou být následně individuálně upravovány.

Například na dolní větvi ulice Technické u budovy č. 2935/23 končí chodník u přechodu pro chodce (viz *obr. 45* v příloze A). Zde bylo nutné vytvořit jeden řez a druhý v místě napojení horní větve ulice Technické, jelikož za touto křižovatkou už chodník opět pokračuje. Část původního chodníku mezi těmito řezy byla následně smazána. Příkladem dalšího místa, kde byl nástroj „Lane Chop Tool“ velmi užitečný, je zastávka autobusu před budovou A1 ve směru na konečnou zastávku Technologický park. Zde bylo také potřeba vytvořit dva řezy, jeden v místě, kde se chodník kvůli zastávce skokově rozšiřuje a druhý v místě přechodu pro chodce, kde chodník končí a začíná vyhrazené parkoviště (viz *obr. 22*). Část původního chodníku v místě vyhrazeného parkoviště byla následně smazána s využitím nástroje „Line Tool“.



Obr. 22 Chodník v místě autobusové zastávky

V okolí FSI je dále poměrně běžná i situace, kdy chodník přímo nenavazuje na obrubník, ale je od vozovky oddělen travnatým nebo štěrkovým pásem (viditelné např. na *obr. 41, 50 a 51* v příloze A) V tomto případě bylo třeba za již existující chodník přidat ještě jeden pruh typu

„Sidewalk“ pomocí nástroje „Line Add Tool“ a pak změnit texturu původního, nyní už vnitřního, chodníku na požadovaný materiál (šterk nebo trávník). Dále byla ještě podle situace pomocí nástroje „Lane Width Tool“ upravena šířka jak travnatého/šterkového pásu, tak i samotného chodníku.

Názorným příkladem je chodník před budovami FSI táhnoucí se v podstatě podél celé horní větve ulice Technické (viz obr. 41-45 v příloze A). Chodník zde má šířku přibližně 2 m a dělicí pás je široký cca 1,2 m. Celý tento rovný úsek se skládá ze čtyř nezávislých segmentů silnice, přičemž na každém z nich bylo vytvořeno relativně velké množství řezů. A to proto, že dochází jednak ke změnám materiálu dělicího pásu, ale i kvůli tomu, že před budovou A1 chodník zasahuje až k samotné silnici (viz obr. 23). Toho bylo dosaženo tak, že dělicímu pásu i chodníku byl přiřazen identický materiál, čímž vznikla vizuálně spojitá plocha. Dále je chodník přerušen dvěma vyasfaltovanými přístupovými cestami ke vchodům do budovy A4. Místa jejich napojení byla vymezena řezy a materiál jak chodníku, tak dělicího pásu, byl změněn na příslušnou variantu asfaltu. Dále bylo ještě potřeba změnit barvu povrchu chodníku z původní šedé na červenou. Taková textura v knihovně RoadRunner dostupná není, a bylo tedy potřeba si vytvořit vlastní. K tvorbě nové textury byla jako základ využita původní šedá textura. Ta byla zduplikována a přejmenována a následně byla změněna barva textury na požadovaný odstín červené.



Obr. 23 Chodník vedoucí podél budov FSI

Při přidávání druhého chodníku se někdy stalo, že se nový chodník nezarovnal s horní hranou obrubníku, ale s povrchem silnice. Nastal-li takový problém, byl k jeho vyřešení použit nástroj „Sidewalk Height Tool“, který umožňuje nastavit výšku odsazení zvoleného chodníku od povrchu vozovky na požadovanou hodnotu. Tento nástroj byl dále velmi užitečný při tvorbě dlážděných přechodů pro chodce na dolní větvi ulice Technické, které jsou vyvýšeny nad okolní vozovku a slouží tak zároveň i jako retardéry (viz obr. 24 a obr. 50-52 v příloze A), které byly vytvořeny následujícím způsobem.

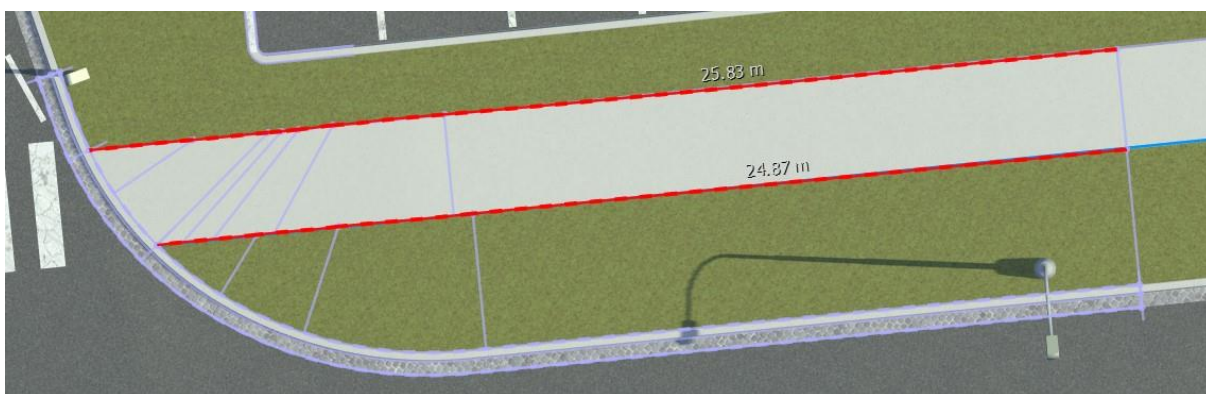
Nejprve byly nástrojem „Road Chop Tool“ na silnici v místě přechodu pro chodce vytvořeny čtyři řezy. První dva byly vytvořeny na úplných okrajích retardéru na rozhraní mezi asfaltem a dlažbou, zatímco zbylé dva vymezyly vyvýšenou vodorovnou plochu v prostřední části retardéru, po které se pohybují lidé a která je zarovnaná s horními hranami obrubníků. V RoadRunneru byla tato střední část pomocí nástroje „Road Height Tool“ zvednuta přesně o výšku obrubníků. Zároveň s ní byly nicméně zvednuty i samotné obrubníky. Právě v tomto okamžiku byl využit nástroj „Sidewalk Height Tool“, pomocí kterého byla výška odsazení obrubníků nad vozovkou v místě vyvýšení nastavena na nulu, čímž bylo dosaženo zarovnání obrubníku s vozovkou a zároveň i plynulého napojení na obrubník před a za retardérem (viz obr. 24). Nakonec ještě bylo potřeba změnit texturu povrchu vozovky v místě retardéru z asfaltu na dlažbu.



Obr. 24 Přečhod pro chodce sloužící zároveň jako retardér

Poměrně složitá pak byla situace na křižovatkách, kde se vedlejší silnice bez chodníků napojuje na průběžnou silnici s chodníkem. Typicky jedná o výjezdy z parkovišť na průběžné silnice. Zde bylo potřeba vytvořit ukončení chodníku v místě zaobleného rohu, což bylo v závislosti na situaci provedeno jedním z následujících dvou způsobů.

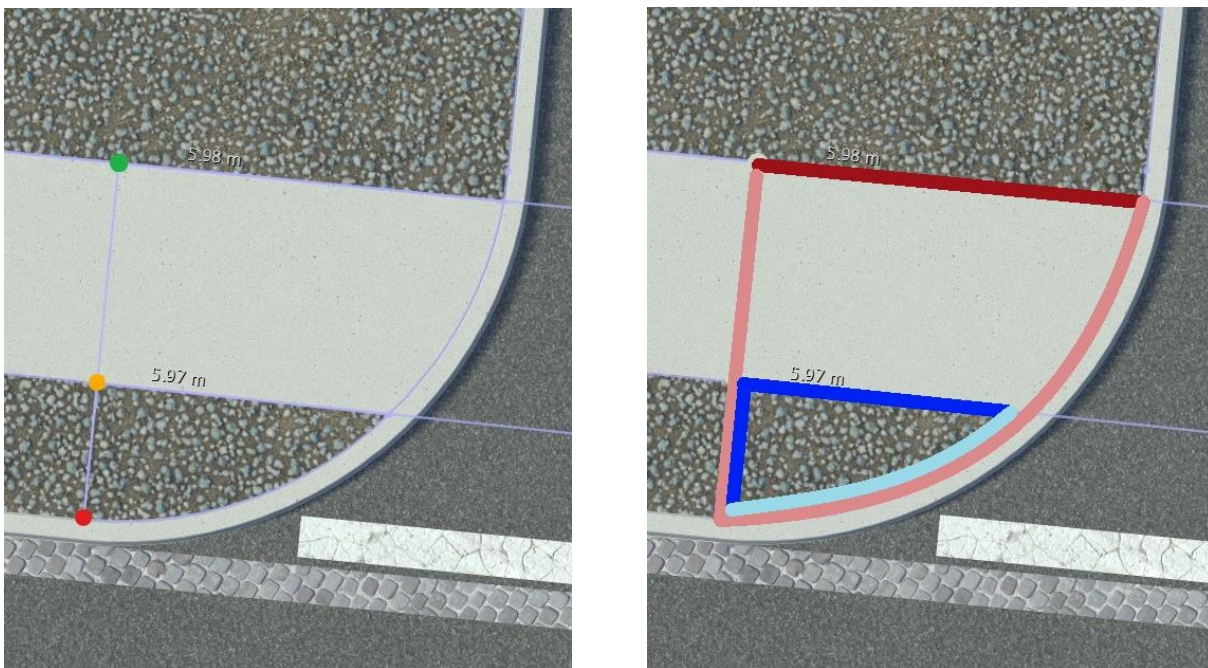
Prvním způsobem bylo vytvoření zakončení chodníku pomocí nástroje „Line Width Tool“. Tento způsob byl využit u křižovatky, kde se ulice Technická rozděljuje na dvě větve. Příjezd k této křižovatce je na obr. 37 a 40 v příloze A. Zde byla nejprve pomocí nástroje „Measurement Tool“ umístěna vodítka, určující jak bude zakončení vypadat (viz obr. 25). Následně byly pomocí nástroje „Lane Chop Tool“ vytvořeny řezy v bodech, kde končí hrany chodníku (v obr. 25 hrany vyznačeny červeně). Poté byly pomocí „Lane Tool“ smazány přebytečné úseky chodníku i travnatého dělicího pásu. Následně byla v několika místech jak chodníku, tak travnatého pásu nástrojem „Lane Width Tool“ změněna šířka a upraven směr tangent tak, aby výsledný tvar jejich okrajů kopíroval nachystaná vodítka (viz obr. 25). Těmito místy se myslí fialové řezy na obr. 25. Výhodou tohoto postupu oproti tomu následujícímu je, že chodník lépe kopíruje výškový profil a klopení silnice u křižovatek ve svahu či takových s nějak jinak deformovaným povrchem. Na druhou stranu je ale mnohem obtížnější vytvořit přímý okraj, jelikož dochází ke vzniku vlnění v horizontálním směru. V neposlední řadě je tento způsob i mnohem pracnější.



Obr. 25 Ukončení chodníku vytvořené nástrojem "Line Width Tool"

Druhý způsob je postaven na využití nástroje „Surface Tool“, sloužícího k úpravě terénů, místo použití nástrojů k úpravě silnic a jejich jednotlivých pruhů. První krok, spočívající ve vytvoření vodítek, byl stejný jako u předešlého postupu. Následně byly pomocí „Line Tool“ smazány veškeré automaticky vytvořené chodníky v rozích křižovatky. Poté byl nástrojem „Surface Tool“ chodník či travnatý/štěrkový pás podle vodítek vytvarován do požadovaného tvaru propojením okraje obrubníku křižovatky s uzlem v rohu chodníku na přechodu mezi křižovatkou a sousedním segmentem silnice (tento uzel je na *obr. 26* vyznačen zeleně). Takto vznikla uzavřená smyčka, která je na *obr. 26* vyznačena červeně, přičemž tmavší odstín znázorňuje hranici smyčky vytvořenou ručně a světlejší hranici tvořenou okrajem silnice. Textura terénu uvnitř této smyčky byla následně podle potřeby změněna. Tento postup je rychlý a přesný u křižovatek v rovině. Použitelný je i ve svazích u jednoduchých zakončení chodníků, kde se nevyskytuje dělicí pás. Na druhou stranu je ale u křižovatek se složitějším výškovým profilem povrchu náročnější dosáhnout toho, aby vytvořený chodník tento profil dobře kopíroval.

Určitý problém při použití tohoto způsobu nastává v místech, kde je chodník od vozovky oddělen dělicím pásem, jako je tomu například u vjezdů a výjezdů do a z garáží pod budovami na dolní větvi ulice Technické (viz *obr. 26*). V takovémto případě není na rozhraní chodníku a dělicího pásu uzel a nejbližší použitelný je až u samotného obrubníku (na *obr. 26* vyznačen červeně). Pak je potřeba smyčku vést až z tohoto bodu a v místě rozhraní chodníku a dělicího pásu je třeba vytvořit uzel navíc (na *obr. 26* vyznačen oranžově). Tato smyčka je znázorněna modře, přičemž tmavší odstín opět znázorňuje hranici vytvořenou ručně, zatímco světlejší odstín hranici tvořenou okrajem silnice. Problém tkví v tom, že jakmile dojde později z nějakého důvodu k posunutí celé křižovatky, tak žlutě vyznačený uzel zůstane na svém původním místě. To může vést ke vzniku zvláštních stínů, dostane-li se uzel do prostoru chodníku automaticky generovaného podél segmentu silnice, který s křižovatkou sousedí. V opačném případě se přidávaný uzel přiblíží k rohu křižovatky a dělicí pás rozdělí klín, mající texturu chodníku. V obou případech je pak nutné polohu problémového uzlu ručně opravit.



Obr. 26 Složitější ukončení chodníku vytvořené nástrojem "Surface Tool"

2.4.2 OKOLNÍ TERÉN

RoadRunner terén vytváří automaticky, dojde-li k uzavření smyčky, jejíž hranici může tvořit buď okraj silnice nebo křivka vytvořená uživatelem pomocí nástroje „Surface Tool“. Uživatelem vytvořené křivky, které tvoří okraj plochy, jsou definovány řídicími body a tečnami v těchto bodech a jejich tvar lze měnit úplně stejným způsobem jako tvar silnic v explicitním režimu. Tímto způsobem byly vytvořeny veškeré terény po okrajích virtuálního světa. Dále lze pomocí tohoto nástroje dělit již existující smyčky, čehož bylo využito především při tvorbě chodníků, které nejsou souběžné se silnicemi. Nové smyčky je možné vytvářet i uvnitř již existujících terénů a změnou výšky řídicích bodů těchto nových smyček měnit celkový výškový profil terénu. Alternativně je možné tyto smyčky vysunout nad okolní terén, přičemž vršek vysunutého terénu může být automaticky zarovnan do roviny bez ohledu na to, jak vypadal výškový profil po obvodu původní smyčky. Tato funkce byla využita při tvorbě vchodů do budov A1 a A4, které nejsou v úrovni země a je potřeba k nim vyjít po schodech. Mezi schody a dveřmi jsou nicméně ještě vyvýšené plochy (viz obr. 27) a právě ty byly vytvořeny výše popsaným způsobem.



Obr. 27 Vyvýšená plocha před vstupem do budovy A4

Výšku řídicích bodů, které definují tvar smyček, lze stejně jako v případě silnic měnit dvěma způsoby. První možností je zvolit konkrétní řídicí body a použít funkci „Project Nodes“, která těmto bodům výšku automaticky nastaví podle výškové mapy. Problémem ale je, stejně jako u silnic, její relativně nízká přesnost. Některé detaily, jako právě třeba chodníky, byly zcela pod její rozlišovací schopnost, jelikož jejich šířka byla ve většině případů menší než rozlišení výškové mapy. V mnoha případech proto musela být výška jednotlivých bodů upravena druhým způsobem, tedy ručně. Vyrovnány musely být například chodníky ve svazích, a to především ty, které kopírují vrstevnici. Jedním z míst, kde se ve vybrané lokalitě takové chodníky nacházejí, je okolí budovy č. 2851/15.

Výšku terénu lze nicméně měnit i mimo řídicí body obvodové křivky. Je-li na vybranou smyčku aplikována funkce „Sample Global Elevation“, RoadRunner automaticky rozdělí terén uvnitř smyčky na pravidelně se opakující trojúhelníky a na základě dat z výškové mapy nastaví vrcholům těchto trojúhelníků výšku. Tyto trojúhelníky nicméně vznikají až v určité vzdálenosti od okraje smyčky, v důsledku čehož byla tato funkce využita minimálně, jelikož většina vytvořených smyček byla prostě příliš malá. Naopak u velkých terénů tímto způsobem vzniká velké množství polygonů navíc, a rostou tak nároky na hardware počítače. Ve výsledku tak byla tato funkce využita pouze jednou, a to v případě terénu v místě zahrad podél ulice U Vodárny.

2.5 DOPRAVNÍ ZNAČENÍ

2.5.1 VODOROVNÉ DOPRAVNÍ ZNAČENÍ

Vodorovné značení lze v RoadRunneru vytvářet několika způsoby. Tím nejrychlejším je použití nástroje „Lane Marking Tool“, kterým lze přidávat čáry na rozhraní pruhů, jak na obyčejných segmentech silnic, tak i v křižovatkách. Při tvorbě vodorovného značení tímto způsobem se nejdříve ve virtuálním světě vybere dané rozhraní a následně se mu přiřadí požadovaný styl čáry z knihovny. Stejně jako u stylů silnic je možné si vybrat buď styl od vývojářů RoadRunneru, anebo si vytvořit vlastní. To ale nebylo potřeba, jelikož knihovna RoadRunneru obsahuje poměrně širokou paletu čar, přičemž mnohé z nich jsou těm českým velmi podobné, a je tedy možné je použít. Tento nástroj byl uplatněn na více místech, především ale na dolní větvi ulice Technické. Zde byly s jeho pomocí vytvořeny přerušované čáry oddělující pruhy pro cyklisty od středního pruhu pro automobily (viz obr. 37 a 47-52 v příloze A).

Čáry je dále možné vytvářet pomocí nástroje „Marking Curve Tool“. Ten umožňuje tvořit křivky libovolných tvarů obdobným způsobem jako silnice a následně těmto křivkám přiřadit libovolný styl čáry. Takto byla vytvořena například klikatá žlutá čára zakazující zastavení před budovou A1 (viz obr. 28). Zde byla využita ještě jedna funkcionality tohoto nástroje, kterou je možnost ve vybraném řídicím bodě rozpojit tečny a vytvořit tak ostré zalomení čáry.

K umístování piktogramů na vozovku byl využit nástroj „Marking Point Tool“. Piktogram lze pomocí něj umístit do středu libovolného jízdního pruhu s tím, že RoadRunner jej automaticky správně natočí podle nastaveného směru jízdy v daném pruhu. Tímto způsobem byly například do pruhů pro cyklisty vloženy piktogramy jízdního kola. Alternativně lze piktogram umístit na libovolné místo ve virtuálním světě jako jakýkoliv jiný model. Pak je ale nutné jeho přesnou polohu a natočení nastavit ručně. Takto byly například umístěny piktogramy vozíčkářů značící parkovací stání vyhrazená pro invalidy.

Součástí RoadRunneru je dále nástroj „Parking Tool“ určený k tvorbě parkovacích míst. Pomocí něj byla vytvořena naprostá většina parkovacích míst ve virtuálním světě, ať už ta pro příčné parkování, kterých byla většina, nebo ta pro podélné před budovou A1. Tento nástroj funguje obdobně jako „Marking Curve Tool“ s tím rozdílem, že podél řídicí křivky RoadRunner místo obyčejné čáry automaticky vytvoří přesně nadefinovaná parkovací místa (viz obr. 28). Ve vytvářeném virtuálním světě byly jako čáry oddělující jednotlivá parkovací místa zvoleny plné čáry ze setu stylů pro Německo, které vypadají stejně jako ty reálné ve vybrané lokalitě. Při zadávání rozměrů parkovacích stání bylo vycházeno z normy ČSN 73 6056 (viz tab. 1). Tyto hodnoty nicméně nebyly striktně dodržovány a když bylo třeba, tak byla velikost stání upravena tak, aby co nejlépe reflektovala realitu.

Tab. 1 Rozměry parkovacích stání [49]

Typ stání	Šířka (m)	Délka (m)
Příčné	2,50	5,00
Příčné pro invalidy	3,50	5,00
Podélné	2,40	5,75

Je nutné poznamenat, že skutečná šířka příčných parkovacích míst a skutečná délka podélných se může ve virtuálním světě od hodnot zadaných do nástroje „Parking Tool“ mírně lišit. RoadRunner totiž v závislosti na délce křivky, podél které jsou stání vytvářena, upravuje jejich velikost tak, aby vznikl celočíselný počet stejně širokých parkovacích míst.



Obr. 28 Parkoviště před budovou A1

2.5.2 SVISLÉ DOPRAVNÍ ZNAČENÍ

Jako svislé dopravní značky lze v RoadRunneru do virtuálního prostředí vkládat obrázky ve formátu SVG, tedy ve formátu škálovatelné vektorové grafiky. Díky tomu je možné například přímo v RoadRunneru měnit velikost značek bez ztráty kvality. Po vložení se značky chovají jako jakékoliv jiné modely, a je tedy možné standardním způsobem upravovat jejich polohu a natočení. Zároveň ale mají i některé sloupky z knihovny tu vlastnost, že se na ně značka při vkládání automaticky přichytí a pak už ji jen stačí natočit do požadovaného směru.

Knihovna objektů v RoadRunneru žádné české značky neobsahuje. Nabízí ale poměrně velké množství značek německých. Některé z nich, jako například značky „Dej přednost v jízdě!“ (viz obr. 29 vlevo) nebo značky stanovující maximální dovolenou rychlost jsou vizuálně téměř identické jako ty české, a bylo tedy možné je použít. Před jejich vložení do virtuálního světa byla pouze změněna jejich velikost tak, aby byla v souladu se vzorovým listem Ministerstva dopravy [50]. Potřebnou značku bylo možné získat i úpravou již existující značky z knihovny. Takto byla například vytvořena značka zakazující vjezd vozidlům na zkapalněný propan-butan, kdy byl na značku „Zákaz vjezdu všech vozidel“ z německého setu ve vestavěném 2D editoru dopravních značek přidán nápis „LPG“ (viz obr. 29 uprostřed).

V mnoha případech nicméně knihovna vhodné značky neobsahovala, a bylo tak třeba potřebný soubor SVG získat jiným způsobem. Některé značky jako například „Jednosměrný provoz“ (viz obr. 29 uprostřed) nebo „Slepá pozemní komunikace“ byly staženy přímo ve formátu SVG získány z internetového zdroje [51]. V jiných případech byl využit bezplatný nástroj od společnosti Adobe, který dokáže převést bitmapový obrázek ve formátu JPG nebo PNG do formátu SVG. Takto byla opatřena například značka „Parkoviště“ na obr. 29. Aby měl vygenerovaný vektorový obrázek dostatečnou kvalitu, musel mít původní obrázek dostatečně vysoké rozlišení. Takové obrázky byly získány z webu e-shopu

„safetyshop.cz“ [52]. Konverzí získaný soubor SVG byl podle potřeby upraven ve volně dostupném editoru vektorové grafiky Inkscape. Jednalo se například o nutnost provést korekci barev, aby byl odstín modrého pozadí stejný jako na ostatních značkách. Nastaly ale i případy, kdy bylo nutné si potřebné značky vytvořit ručně. Jednalo se především o dodatkové tabulky. Prázdné tabulky požadovaných rozměrů byly vytvořeny podle vzorového listu [50] pomocí již zmíněného softwaru Inkscape a text či piktogramy (viz dodatková tabulka vpravo na *obr. 29*) byly následně přidány v RoadRunner pomocí 2D editoru. Oficiálním fontem, používaným na českých dopravních značkách, je font DIN 1451 [50]. Ten sice v nabídce 2D editoru není, nicméně je tam font „Bahnschrift“, což je jeho digitalizovaná verze [53]. Konkrétně byly použity jeho varianty „Semibold“ a „Semibold Condensed“, které nejvíce odpovídají realitě.



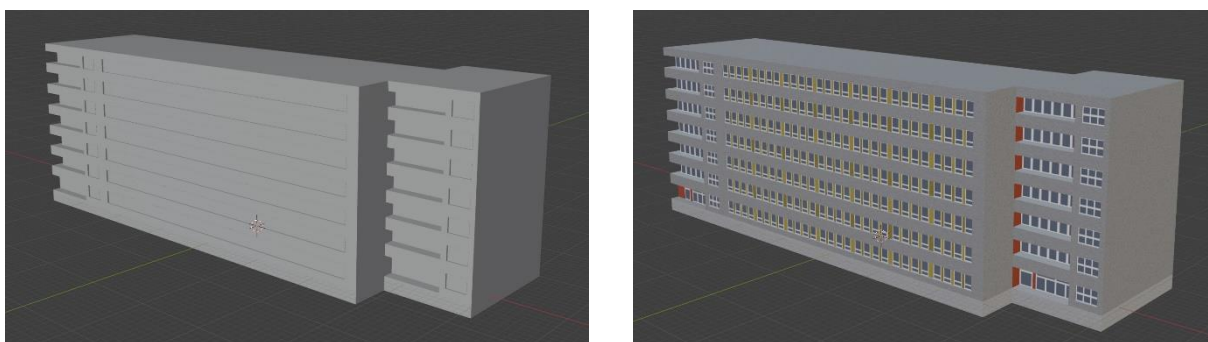
Obr. 29 Vybrané dopravní značky

2.6 MODELY

Modely byly do virtuálního prostředí umisťovány pomocí nástrojů „Prop Point Tool“, „Prop Span Tool“ a „Prop Polygon Tool“. První zmíněný nástroj umožňuje umístit samostatný model na konkrétní místo. Polohu a natočení vložených modelů je následně možné libovolně upravovat. Tento nástroj byl využit především při umisťování budov, které se většinou ve vybrané lokalitě nachází pouze jednou. Druhý zmíněný nástroj umožňuje vytvářet křivky, ke kterým může být následně přiřazen nějaký model z knihovny. Potom je možné v panelu atributů nastavit velikost rozestupů mezi jednotlivými modely a jejich relativní natočení vůči směru této křivky. Toto bylo užitečné především při umisťování lamp, které jsou podél silnic. Těmto křivkám je možné kromě obyčejných modelů přiřadit i objekt typu „Extrusion Style“. Ten je definován modelem, který se v pravidelných intervalech opakuje a profilem tažení, které kopíruje tvar křivky a které opakující se modely navzájem propojí. Objektem tohoto typu například dřevěný plot u křižovatky ulic Technické a U Vodárny (viz *obr. 39* v příloze A). V tomto případě byl použit již nadefinovaný plot z knihovny assetů. Jindy bylo ale třeba si vytvořit vlastní „Extrusion Style“. Příkladem takové situace je zelený plot podél horní větve ulice Technické (viz *obr. 44* v příloze A), u kterého byly použity jak vlastní sloupky, tak i vlastní textura tažení. Nástroj „Prop Polygon Tool“ umožňuje vytvářet ve virtuálním světě mnohoúhelníky, kterým je možné z knihovny přiřadit buď jeden konkrétní model nebo předem nadefinovanou skupinu různých modelů. RoadRunner poté do prostoru vymezeného mnohoúhelníkem náhodně umístí několik kopií zvoleného modelu, přičemž je možné nastavit jejich hustotu. Takto byly vytvářeny především různé porosty.

Knihovna RoadRunneru obsahuje relativně velké množství modelů, nicméně ty byly využity jen minimálně, jelikož se do vybrané lokality většinou příliš nehodí. Z toho vyplynula nutnost si potřebné modely vytvořit. K tomu byl použit bezplatný software Blender ve verzi 4.0.2.

Při tvorbě modelů budov, lamp, betonových prvků apod. byly nejprve základní objekty, jako jsou krychle či válce, přetvořeny do požadovaného tvaru (viz *obr. 30* vlevo), přičemž bylo vždy třeba najít kompromis, co se týče množství detailů, tak aby model vypadal dobře, ale zároveň aby nebyl tvořen příliš velkým počtem polygonů a nekladl tak příliš velké nároky na výkon počítače. Vytvořený model bylo následně nutné opatřit texturami (viz *obr. 30* vpravo). V mnoha případech pouze stačilo změnit barvu jednotlivých ploch, které vytváří povrch modelu. Jindy bylo třeba jako textury použít obrázky. Typickým příkladem jsou okna. Ta byla nejprve nakreslena v Inkscapu a poté byla v dostatečném rozlišení vyexportována jako bitmapový obrázek, který byl v Blenderu použit jako textura. U některých modelů nicméně vůbec nebylo potřeba textury vytvářet, jelikož mohly být použity přímo textury z knihovny objektů v RoadRunneru. To byl případ například různých betonových prvků. Zde byly využity textury, které jsou sice primárně určené k použití jako povrch chodníků, nicméně i pro výše zmíněnou aplikaci byly zcela dostačující.

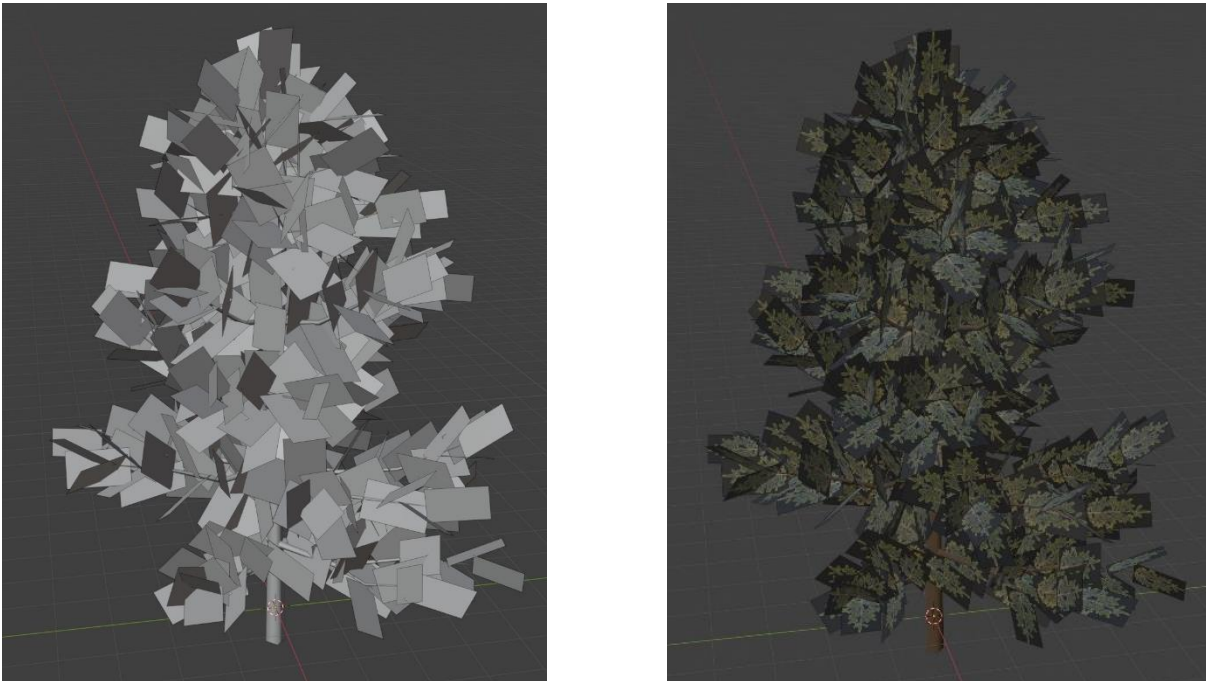


Obr. 30 Model budovy A4 bez textur (vlevo) a s texturami (vpravo)

Modelů stromů a keřů je v rámci knihovny objektů poměrně hodně, nicméně v praxi byly použitelné pouze keře, jelikož modely stromů z knihovny jsou často tvořeny velmi vysokým počtem polygonů. Obvykle se jedná o nižší desítky tisíc, zatímco v Blenderu nebyl problém vytvořit dostatečně dobře vypadající modely stromů s méně jak dvěma tisíci polygony.

Proces tvorby modelů stromů byl výrazně odlišný od procesu tvorby budov. Stromy totiž nebyly vytvářeny ručně z jednoduchých objektů, nýbrž byl použit bezplatný doplněk do Blenderu zvaný „Sapling Tree Gen“, který umí stromy automaticky generovat podle zadaných vstupních parametrů. Při zadávání těchto parametrů bylo nejprve třeba si z nabídky zvolit základní tvar stromu, tzn. zda je jedná například o listnatý, či jehličnatý strom. Dále bylo možné nastavit výšku stromu, šířku koruny, průměr kmene a větví, jak vysoko nad zemí mají větvě začít vyrůstat z kmene a mnoho dalších detailů. Obzvláště důležité bylo přiměřeně zvolit počet větví a míru jejich větvení. Tyto parametry totiž zásadním způsobem ovlivňovaly počet polygonů vytvářeného modelu. Nakonec bylo potřeba vytvořit listy. Doplněk je schopný kolem větví automaticky vygenerovat zvolený počet listů, přičemž je možné si zvolit libovolný objekt, který bude listy představovat. Modelování samostatných listů by vedlo ke vzniku obrovského množství polygonů. Jako model listu tedy byla použita relativně velká obdélníková rovina, opatřená texturou větvě ve formátu PNG (viz *obr. 31*). V tomto případě byl formát PNG zásadní, jelikož je schopný nést informaci o tom, které části daného obrázku mají být průhledné a které ne. Pokud je tedy jako textura použit obrázek větve, kde jsou mezi

jednotlivými listy průhledné mezery, vznikne ve virtuálním prostředí iluze, že listy jsou samostatnými objekty, bez toho aniž by musela být vytvářena jakákoliv geometrie navíc. V případě jehličnatých stromů, byla taková textura opatřena z internetového zdroje [54] a v případě listnatých stromů byla převzata ze staženého 3D modelu [55]. Ze stejného modelu byla převzata i textura použitá pro kmeny listnatých stromů. V případě stromů jehličnatých byla využita textura z knihovny objektů.



Obr. 31 Model jehličnatého stromu bez textur (vlevo) a s texturami (vpravo)

Problém při tvorbě modelů obecně představovaly rozměry modelovaných objektů. U velkých objektů, jako jsou například budovy, bylo možné rozměry půdorysu určit například pomocí nástroje měření vzdálenosti v Google Maps. Problém ale představovalo určení výšky modelovaných objektů. V případě budovy A1 ji bylo možné dohledat. U ostatních budov ji bylo třeba odhadnout. K tomu posloužila informace o výšce budovy A1, která měří 74 metrů a má 19 pater [56]. Vezme-li se v potaz to, že vstup do budovy je vyvýšen nad okolní terén a že nad posledním patrem je ještě strojovna výtahů, vyjde, že jedno patro je vysoké přibližně 3,5 metru. Na základě toho byla výška ostatních budov hrubě odhadnuta tak, že byla tato hodnota vynásobena počtem pater dané budovy. Určité rozměrové nepřesnosti nicméně velký problém nepředstavují, jelikož z pohledu jízdy vozidla po silnici není výška okolních budov nikterak zásadní. V případě menších objektů byla výška odhadnuta tak, aby na pohled proporcionálně odpovídala okolním objektům, jako jsou třeba zaparkovaná auta, nebo právě budovy. Rozměry bylo navíc ještě možné doladit přímo v prostředí RoadRunneru, kde lze modely podle potřeby škálovat nezávisle ve třech osách.

Jakmile byl model v Blenderu hotov, bylo ho třeba vyexportovat ve vhodném formátu tak, aby mohl být ve virtuálním světě použit. RoadRunner podporuje mnoho různých formátů. Za účelem zajištění kompatibility byl zvolen formát FBX, což je ten samý formát, ve kterém je následně vytvořený svět exportován do Unreal Engine.

Umístění vytvořených modelů do světa v RoadRunneru bylo posledním krokem při tvorbě samotného virtuálního prostředí. Na *obr. 32 a 33* je vytvořený virtuální svět v celém jeho rozsahu. Další snímky světa včetně jejich porovnání s realitou jsou součástí přílohy A.



Obr. 32 Pohled shora na vytvořené virtuální prostředí



Obr. 33 Pohled na vytvořené virtuální prostředí z ptačí perspektivy

3 OVĚŘENÍ FUNKČNOSTI

RoadRunner posloužil k vytvoření samotného virtuálního světa. Za účelem ověření jeho funkčnosti ho bylo třeba vyexportovat takovým způsobem, aby mohl být použit ve vybraném softwaru pro testování autonomních vozidel. Tím může být jakýkoliv simulátor zmíněný v rešeršní části práce. CarMaker nebo DRIVE Sim nicméně v rámci této práce nebylo možné použít, jelikož by bylo nutné k nim vlastnit licenci. Použitím simulačního prostředí RoadRunner Scenario by došlo k určitému omezení využitelnosti vytvořeného světa, kvůli nutnosti pro použití světa vlastnit kromě licence k softwarům MATLAB a Simulink i licenci k samotnému RoadRunneru, což v případě použití vyexportovaného světa v Unreal Engine nebo v simulátoru CARLA neplatí. Nakonec byla zvolena varianta exportu do Unreal Engine, který je pro nekomerční využití volně dostupný a který je možné pomocí nástrojů z doplňku „Automated Driving Toolbox“ propojit se Simulinkem, jenž je dostupný v rámci akademické licence na produkty od společnosti MathWorks. Další výhodou je to, že se jedná o tu samou společnost, která vyvíjí RoadRunner, díky čemuž jsou dostupné jednak nástroje určené přímo pro export do Unreal Engine, tak i podrobná online dokumentace popisující tento proces.

V Simulinku pak byla vytvořena jednoduchá simulace, jejíž cílem bylo ověřit, že se virtuální svět zobrazuje tak, jak má, Simulink dokáže ovládat pohyb herců, kteří se v něm pohybují a tento pohyb je realistický, tedy že například nedochází k přerušení kontaktu vozidla s vozovkou. Důležité bylo rovněž otestovat, že simulace probíhá na použitém hardwaru dostatečně plynule. Tím byl stolní počítač vybavený procesorem Intel Core i7-14700K, grafickou kartou NVIDIA GeForce RTX 4070 Ti a 64 GB RAM.

3.1 EXPORT Z ROADRUNNERU A IMPORT DO UNREAL ENGINE

Před samotným exportem bylo ještě nutné nastavit, co přesně se má vyexportovat. Scénu lze vyexportovat buď celou v jednom kuse, nebo ji rozdělit na menší dlaždice o stanovené velikosti. To dává smysl především u velkých scén, kdy někdy není potřeba celý virtuální svět, ale stačí pouze jeho část. Rozložení těchto dlaždic je možné před exportem vizualizovat pomocí nástroje „Scene Export Preview Tool“, pomocí kterého lze zjistit i již několikrát zmíněný počet polygonů, a to jak celého světa, tak i jednotlivých modelů. Možnost rozdělit svět na menší dlaždice nicméně využita nebyla, jelikož vytvořený svět není nikterak rozsáhlý.

Před zahájením importu bylo nutné zkontrolovat, že nainstalovaná verze Unreal Engine je kompatibilní s použitou verzí R2023b softwarů MATLAB a Simulink. Tím je Unreal Engine ve verzi 4.27 [57]. Dále bylo třeba nainstalovat Microsoft Visual Studio 2019. Po spuštění Unreal Engine byl nejprve vytvořen nový projekt. Aby propojení se Simulinkem fungovalo, musí jednat o projekt typu C++ [58]. Dále bylo třeba ze webu MathWorks stáhnout pluginy „RoadRunnerImporter“, „RoadRunnerRuntime“ a „RoadRunnerMaterials“. Ty slouží k usnadnění procesu importu vytvořeného virtuálního světa do Unreal Editoru, což je pracovní prostředí Unreal Engine, ve kterém lze vytvářet a upravovat virtuální světy. Stažené pluginy bylo třeba vložit do příslušné složky v rámci vytvořeného projektu a následně v Unreal Editoru aktivovat.

Samotný proces importu byl zahájen vložením souboru FBX, vytvořeného při exportu z RoadRunneru, do prohlížeče obsahu v Unreal Editoru. Díky nainstalovaným pluginům se v tomto okamžiku objevilo dialogové okno, které umožnilo detailně nastavit, jakým způsobem má import proběhnout. Aby vše proběhlo v pořádku, bylo v záložce „Static Meshes“ provedeno několik změn. Na základě doporučení z dokumentace byla vybrána

možnost „Import Normals“ v kolonce „Import Normal Method“ a dále byly deaktivovány parametry „Remove Degenerates“ a „Generate Lightmap UVs“ [58]. Ostatní parametry byly ponechány na výchozích hodnotách.



Obr. 34 Vzhled virtuálního prostředí v Unreal Engine po importu

Po importu byly ještě v rámci Unreal Editoru provedeny drobné úpravy. Textury listů a jehličí, použité u vlastních modelů stromů, byly nastaveny jako oboustranné. Ve výchozím nastavení jsou totiž textury jednostranné, a viditelné jsou tudíž pouze čelní strany polygonů, ze kterých je objekt složen. Naprostá většina modelů má nějaký objem a jednostranné textury tedy nepředstavují problém, jelikož jsou vždy viditelné pouze čelní strany polygonů. To ale neplatí u modelů stromů, kde je listí a jehličí tvořeno pouze rovinnými útvary. Koruna stromu se proto ve výchozím nastavení zdá řídká, jelikož ty roviny, které jsou ke kameře natočeny zadní stranou, nejsou viditelné. Použití oboustranné textury tento problém vyřešilo. Dále byly provedeny změny osvětlení, aby byl výsledný vizuální dojem z virtuálního prostředí a z videozáznamu co nejlepší. Ve výchozím stavu po importu byly stíny příliš tmavé, a zanikalo v nich tak značné množství detailů (viz obr. 34). Byla proto snížena intenzita osvětlení sluncem a změněn směr, ze kterého slunce svítí, a zvýšena intenzita nepřímého osvětlení. Těmito úpravami bylo dosaženo stavu na obr. 35.



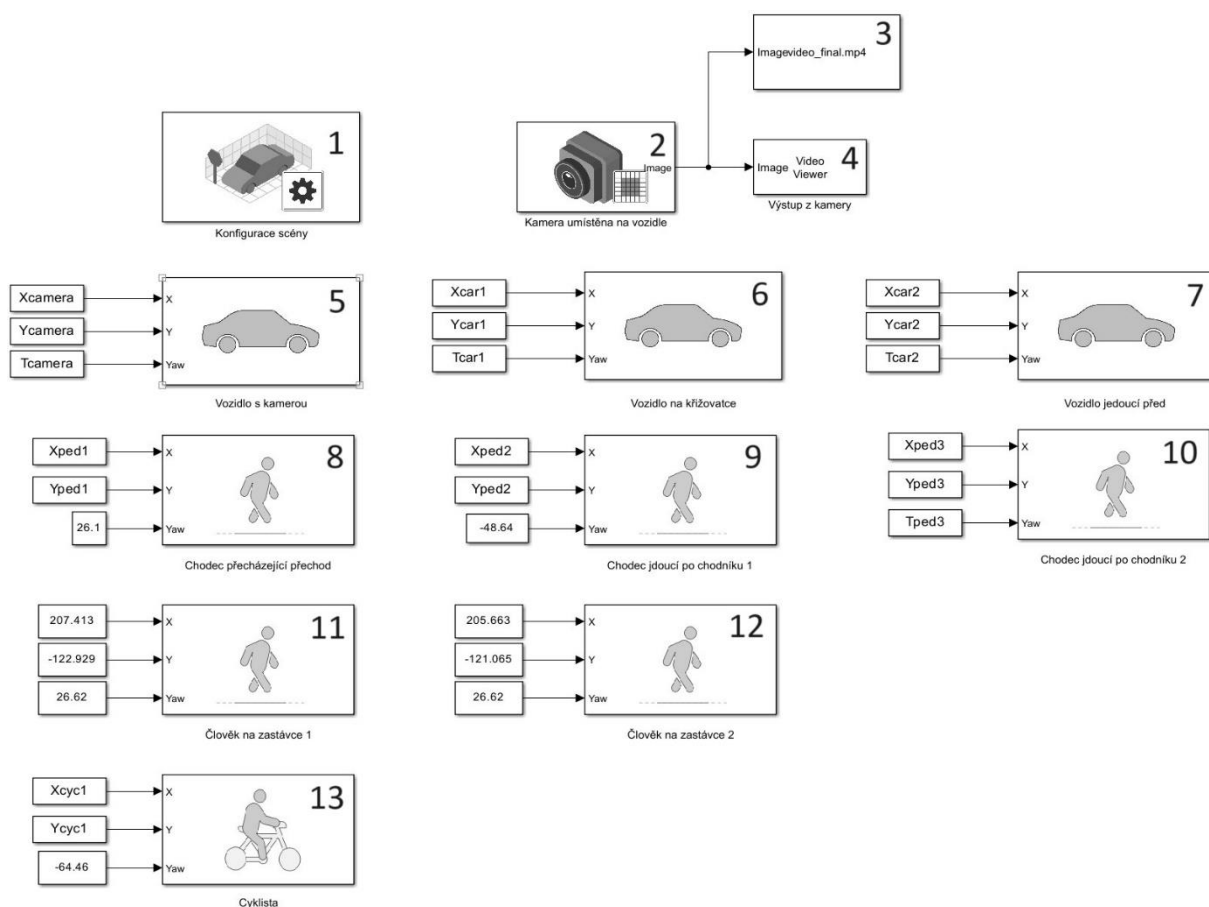
Obr. 35 Vzhled virtuálního prostředí po úpravě osvětlení

3.2 PROPOJENÍ SE SIMULINKEM

K propojení Simulinku s Unreal Engine i k vytvoření jednoduché simulace, sloužící k ověření funkčnosti vytvořeného virtuálního světa, byly použity nástroje z v rešerši zmíněného doplňku do Simulinku „Automated Driving Toolbox“. Aby byla zajištěna komunikace mezi Simulinkem a Unreal Engine, bylo třeba stáhnout, doinstalovat a aktivovat ještě další dva pluginy, konkrétně „MathWorksSimulation“ a „MathWorksAutomotiveContent“. Samotné propojení zajišťuje v prostředí Simulinku blok „Simulation 3D Scene Configuration“ (blok 1 na obr. 36). Součástí tohoto bloku je parametr „Scene Source“, který určuje zdroj scény, ve které bude simulace probíhat. Zde byla vybrána možnost „Unreal Editor“ a zadána cesta k projektu s virtuálním světem ve formátu UPROJECT. Aby mohlo být spojení mezi Simulinkem a Unreal Engine navázáno, nesmí být Unreal Editor spuštěn konvenčním způsobem, ale přes Simulink prostřednictvím výše zmíněného bloku „Simulation 3D Scene Configuration“.

3.3 NASTAVENÍ SIMULACE

Funkčnost vytvořeného virtuálního prostředí byla otestována pomocí simulace, jejíž část je vidět na obr. 36. Za účelem vyšší přehlednosti byly z obrázku vypuštěny bloky některých herců, které se od těch na obrázku liší pouze vstupními hodnotami. Celá simulace je obsažena v příloze C. V průběhu simulace se ve virtuálním světě pohybují jednotliví herci po předem stanovených trajektoriích. Herci jsou myšlena virtuální vozidla, chodci a cyklisté, přičemž chování každého jednoho z nich je řízeno samostatným blokem (bloky 6-13 na obr. 36).



Obr. 36 Část schématu simulace

Pro vozidla byl použit blok „Simulation 3D Vehicle with Ground Following“ se třemi nezávislými vstupy (viz bloky 5-7 na *obr. 36*). Těmi jsou poloha vozidla v ose x a y a jeho natočení kolem svislé osy, přičemž tyto vstupy mohou být buď konstantní nebo časově proměnné. Vstupní data mají formát matice o dvou sloupcích, kdy v prvním sloupci je čas od počátku simulace, kterému jsou ve sloupci druhém přiřazeny hodnoty dané souřadnice v metrech nebo natočení ve stupních. Polohu vozidla ve svislém směru a jeho natočení kolem os x a y Simulink v čase automaticky upravuje tak, aby vozidlo zůstávalo v neustálém kontaktu s povrchem pod ním [59].

V rámci bloků „Simulation 3D Vehicle with Ground Following“ byly upraveny následující parametry. Byla nastavena počáteční pozice a natočení vozidla, přičemž platilo, že hodnoty souřadnic x a y a natočení kolem svislé osy muselo být stejné jako vstupní hodnoty do bloku na počátku simulace v čase 0. Počáteční pozice v ose z musela být zvolena mírně nad povrchem vozovky, aby byl Simulink schopen povrch nalézt a vozidlo na něj umístit. Dále bylo možné v parametrech bloku dané vozidlo libovolně pojmenovat a zvolit jeho typ a barvu. Například vozidlo, na kterém je umístěna kamera, je typu „muscle car“ a je červené. To ale na průběh samotné simulace nemá žádný vliv. Kromě vozidla s kamerou (blok 5 na *obr. 36*), byl blok „Simulation 3D Vehicle with Ground Following“ použit i pro umístění dalších vozidel do scény, která slouží k jejímu oživení. Za tímto účelem byli do scény vloženi i další herci ve formě chodců a cyklisty, jejichž pohyb je řízen bloky „Simulation 3D Pedestrian“ (bloky 8-12 na *obr. 36*), resp. „Simulation 3D Bicyclist“ (blok 13 na *obr. 36*). Ty fungují stejným způsobem jako výše popsany blok „Simulation 3D Vehicle with Ground Following“.

Dále bylo třeba nastavit vzorkovací frekvenci celé simulace. Systémy autonomního řízení musí pracovat v reálném čase a vzorkovací perioda, se kterou pracují, musí mít pevnou velikost a musí být dostatečně krátká, aby byl systém schopný držet krok s pohybem vozidla a bleskově reagovat na dění v jeho okolí [8]. Vzorkovací perioda byla proto v simulaci nastavena na pevnou hodnotu 20 ms, což odpovídá vzorkovací frekvenci 50 Hz.

3.4 STANOVENÍ TRAJEKTORIE HERCŮ

Při stanovování trajektorií jednotlivých herců byla v RoadRunneru nejprve zvolena místa, kterými má vozidlo nebo cyklista projet nebo chodec projít. Do těchto míst byly umístěny provizorní modely, které byly následně natočeny tím směrem, kterým se má herec v tomto bodě pohybovat. Vzhledem k tomu, že RoadRunner používá stejný souřadnicový systém jako Simulink, bylo možné souřadnice a natočení těchto modelů přímo použít jako vstupní data do bloků. V případě herců, kteří se pohybují po přímce, pak už jen stačilo doplnit časy, ve kterých se mají herci ve zvolených bodech nacházet, a data byla připravena k použití. U vozidel, která se pohybují po zakřivených trajektoriích, to bylo složitější. Simulink sice nabízí možnost hodnoty mezi jednotlivými body interpolovat, nicméně problémem je, že vozidlo se pak mezi jednotlivými body nepohybuje po realistických křivkách, ale po přímkách.

Jednou z možností, jak toto vyřešit, bylo použít v rešeršní části práce zmíněný nástroj „Driving Scenario Designer“. Do něj by bylo možné silnice z vytvořeného světa naimportovat a následně na ně potřebné trajektorie zakreslit. Tato metoda nicméně využita nebyla a místo toho byl v softwaru MATLAB napsán skript, který je schopný ze zadaných hodnot realistickou trajektorii vypočítat (viz příloha B). Důvodem tohoto rozhodnutí byla snaha nevyužívat pouze nástroje vytvořené někým jiným, ale pokusit se najít vlastní řešení daného problému, které povede k dosažení uspokojivých výsledků.

Vstupními hodnotami do tohoto skriptu jsou souřadnice vozidla v osách x a y , hodnota natočení kolem svislé osy ve zvolených bodech a dále i rychlost v m/s, kterou se má vozidlo v daných bodech pohybovat. Skript nejprve pomocí funkce „smoothPathSpline“ vytvoří mezi každými dvěma v RoadRunneru zvolenými sousedními body určitý počet nových vložených bodů, které leží na křivce, která je v krajních dvou bodech tečná k směru jízdy vozidla. Skript následně spočítá přes vložené body délku dráhy pohybu vozidla mezi dvěma krajními body. Ze znalosti této dráhy a zadané rychlosti vozidla v krajních bodech je následně dopočteno zrychlení na daném úseku. Za předpokladu, že je zrychlení na tomto úseku konstantní, skript dopočítá časy, ve kterých se má vozidlo v jednotlivých vložených bodech a v koncovém bodě nacházet tak, aby bylo zrychlování či zpomalování vozidla plynulé. Je-li na daném úseku zrychlení nulové, skript pouze spočítá, za jak dlouhou dobu vozidlo dorazí do koncového bodu, a přiřadí k jednotlivým bodům čas tak, aby byla rychlost konstantní. Stojí-li vozidlo na místě, například v okamžiku když pouští chodce na přechodu, skript žádné vložené body nevytváří.

3.5 NASTAVENÍ KAMERY A POŘÍZENÍ VIDEOZÁZNAMU

K získávání obrazových dat ze simulace posloužil v Simulinku blok „Simulation 3D Camera“ (blok 2 na *obr. 36*). Jedná se o virtuální kameru, která se ve virtuálním prostředí nachází na předem specifikovaném místě, ze kterého zaznamenává, co se v průběhu simulace ve světě, který renderuje Unreal Engine, děje. Kamera byla umístěna za čelní sklo do oblasti u zpětného zrcátka vozidla, řízeného blokem 5, které v průběhu simulace projede většinou vytvořeného virtuálního světa. Toto umístění bylo zvoleno proto, že se jedná o místo, kde lze kamery v automobilech často nalézt. Dále bylo zvětšeno zorné pole kamery a rozlišení zaznamenávaného obrazu bylo nastaveno na 1920x1080 pixelů, což je maximální možná hodnota [60]. Cílem těchto úprav bylo dosáhnout co nejlepšího vizuálního dojmu z výsledného videozáznamu.

Blok „Simulation 3D Camera“ slouží pouze k získávání dat. Aby bylo možné si záznam z kamery prohlédnout, byla tato obrazová data použita jako vstup do bloku „Video Viewer“ (blok 4 na *obr. 36*), což je v podstatě virtuální displej, na kterém je možné v reálném čase sledovat, co kamera v průběhu simulace snímá. Toto bylo užitečné především při nastavování simulace k ověřování toho, že se jednotliví herci ve virtuálním světě chovají tak, jak mají. Jakmile měla simulace finální podobu, byl k výstupu z kamery připojen i blok „To Multimedia File“ (blok 3 na *obr. 36*), který záznam z kamery uložil jako video ve formátu MP4.

Tento výsledný videozáznam slouží jako důkaz toho, že vytvořený virtuální svět bylo možné do Unreal Engine úspěšně nahrát a ověřovací simulace probíhá na použitém hardwaru plynule a bez problémů. Záznam dále dokazuje, že je možné z vytvořeného světa pomocí virtuálního senzoru, kterým byla v tomto případě kamera, získávat relevantní data. Z toho plyne, že vytvořený virtuální svět je funkční a je možné ho využít k tomu, k čemu je určený.

ZÁVĚR

Bakalářská práce se zabývá vývojem virtuálního světa pro testování autonomních vozidel. V rešeršní části práce jsou rozebrány vybrané programy, určené k návrhu samotného virtuálního prostředí, zmíněny používané datové formáty sloužící k popisu virtuálních světů. Dále jsou popsány simulátory, ve kterých lze světy vytvořené v softwaru RoadRunner použít, přičemž zmíněny jsou i jejich vlastní nástroje pro tvorbu a úpravu virtuálních světů. Pozornost je věnována i sensorům, jejichž funkci je třeba při virtuálním testování simulovat.

Hlavním cílem práce bylo vytvořit virtuální svět vybrané lokality v České republice pomocí softwaru RoadRunner, který bude detailně odpovídat reálné předloze. Zvoleno bylo okolí Fakulty strojínského inženýrství v Brně, což je městské prostředí, které skýtá pro autonomní vozidla množství výzev a umožňuje testovat relativně široké spektrum různých scénářů. Svět byl vytvořen pomocí nástrojů dostupných v RoadRunneru na základě referencí, jako jsou fotografie, satelitní snímky či výšková mapa, a opatřen vytvořenými modely, které detailně odpovídají svým skutečným protějškům. Vytvořený svět dále obsahuje relevantní svislé a vodorovné dopravní značení jehož umístění přesně odpovídá skutečnosti. Je tedy možné říci, že vytvořený svět v rámci možností RoadRunneru dostatečně věrně reflektuje realitu, a tudíž lze cíl prohlásit za splněný.

Dalším cílem práce bylo ověření funkčnosti vytvořeného světa. Za tímto účelem byl svět vyexportován ve formě 3D geometrie a nahrán do Unreal Engine, který posloužil k renderování virtuálního prostředí v průběhu simulace, zatímco Simulink, který lze s Unreal Engine propojit, řídí chování vozidel, chodců a cyklistů. Simulinkem řízené vozidlo, které je středobodem simulace, se pohybuje po předem stanovené trajektorii. Nejedná se tedy o autonomní vozidlo, jelikož cílem simulace bylo ověřit, že se svět během simulace správně zobrazuje, simulace probíhá plynule, simulátor dokáže ovládat pohyb vozidla ve vytvořeném světě a tento pohyb je realistický. To znamená, že se vozidlo například nenorí do asfaltu nebo nad ním naopak nelevituje. Vše zmíněné se potvrdilo. Vytvořením videozáznamu pomocí virtuální kamery pak bylo dokázáno, že je možné z vytvořeného prostředí získávat použitelná sensorická data.

Pokud jde o potencionální použití výsledků práce, mohl by být vytvořený svět využit například k ověřování funkčnosti různých navržených algoritmů autonomního řízení či ADAS systémů, nebo naopak k získávání sensorických dat potřebných k jejich vývoji. Alternativně by mohly být při tvorbě jiných virtuálních světů použity některé vytvořené univerzální modely, jako jsou například stromy nebo dopravní značky.

Vytvořený svět by mohl být do budoucna vylepšen například tak, že by byl rozšířen. Ve své aktuální podobě totiž vůbec nezahrnuje například tramvajové koleje, víceproudé komunikace nebo mimoúrovňové, světelné či okružní křižovatky. Výhodou je, že všechny tyto zmíněné prvky by bylo možné do světa relativně jednoduše přidat, jelikož se v realitě nacházejí poměrně blízko vybrané lokality. Dále by mohla být například vytvořena varianta světa určená k testování v noci. K tomu by bylo třeba upravit modely lamp a budov tak, aby emitovaly světlo.

POUŽITÉ INFORMAČNÍ ZDROJE

- [1] ROUSE, Margaret. *Autonomous Vehicle*. Online. In: Technopedia. 5 July 2023. Dostupné z: <https://www.techopedia.com/definition/30056/autonomous-vehicle>. [cit. 2024-04-11].
- [2] PAREKH, Darsh; PODDAR, Nishi; RAJPURKAR, Aakash; CHAHAL, Manisha; KUMAR, Neeraj et al. A Review on Autonomous Vehicles: Progress, Methods and Challenges. Online. *Electronics*. 2022, vol. 11, iss. 14, p. 1-18. ISSN 2079-9292. Dostupné z: <https://doi.org/10.3390/electronics11142162>. [cit. 2024-05-22].
- [3] DIETSCHKE, Karl-Heinz a REIF, Konrad. *Automotive handbook*. 11th edition. Karlsruhe: Bosh, 2022. ISBN 978-1119911906.
- [4] SAE INTERNATIONAL. *SAE Levels of Driving Automation™ Refined for Clarity and International Audience*. Online. SAE INTERNATIONAL. SAE international Website. 2021. Dostupné z: <https://www.sae.org/blog/sae-j3016-update>. [cit. 2024-04-11].
- [5] *Autopilot and Full Self-Driving Capability*. Online. Tesla. © 2024. Dostupné z: <https://www.tesla.com/support/autopilot>. [cit. 2024-04-11].
- [6] *Drive Pilot*. Online. Mercedes-Benz USA. © 2024. Dostupné z: <https://www.mbusa.com/en/owners/manuals/drive-pilot>. [cit. 2024-04-11].
- [7] YEONG, De Jong; VELASCO-HERNANDEZ, Gustavo; BARRY, John a WALSH, Joseph. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Online. *Sensors*. 2021, vol. 21, iss. 6, p. 1-37. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s21062140>. [cit. 2024-04-11].
- [8] ČERNÝ, David; VACULÍN, Ondřej a ZÁMEČNÍK, Petr (ed.). *Automatizované řízení vozidel a autonomní doprava: technické a humanitní perspektivy*. Praha: Academia, 2022. ISBN 978-80-200-3358-1.
- [9] BILIK, Igal; LONGMAN, Oren; VILLEVAL, Shahar a TABRIKIAN, Joseph. The Rise of Radar for Autonomous Vehicles: Signal Processing Solutions and Future Research Directions. Online. *IEEE Signal Processing Magazine*. 2019, vol. 36, iss. 5, p. 20-31. ISSN 1053-5888. Dostupné z: <https://doi.org/10.1109/MSP.2019.2926573>. [cit. 2024-04-11].
- [10] ZHU, Li. Analyze the Advantages and Disadvantages of Different Sensors for Autonomous Vehicles. Online. *Advances in Economics, Business and Management Research*. 2022, iss. 215, p. 1020-1024. Dostupné z: <https://doi.org/10.2991/aebmr.k.220405.171>. [cit. 2024-04-12].
- [11] VARGAS, Jorge; ALSWEISS, Suleiman; TOKER, Onur; RAZDAN, Rahul a SANTOS, Joshua. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Online. *Sensors*. 2021, vol. 21, iss. 16, p. 1-22.

- ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s21165397>. [cit. 2024-04-12].
- [12] JOUBERT, Niels; REID, Tyler G. R. a NOBLE, Fergus. Developments in Modern GNSS and Its Impact on Autonomous Vehicle Architectures. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas: IEEE, 2020, p. 2029-2036. ISBN 978-1-7281-6673-5. Dostupné z: <https://doi.org/10.1109/IV47402.2020.9304840>.
- [13] *Inertial measurement unit*. Online. Bosch Mobility. © 2024. Dostupné z: <https://www.bosch-mobility.com/en/solutions/sensors/inertial-measurement-unit/>. [cit. 2024-05-10].
- [14] *Bosch MEMS Sensors*. Online. Bosch Mobility. © 2024. Dostupné z: <https://www.bosch-mobility.com/en/solutions/electronic-components/mems-sensors/>. [cit. 2024-05-19].
- [15] *Localization for automated driving*. Online. Bosch Mobility. © 2024. Dostupné z: <https://www.bosch-mobility.com/en/solutions/automated-driving/localization-for-automated-driving/>. [cit. 2024-05-19].
- [16] WAYMO. *Designing the Waymo Driver*. Online, video. 2020-03-26. Dostupné z: YouTube, <https://www.youtube.com/watch?v=o8rCOKSDMcg>. [cit. 2024-05-10].
- [17] *ASAM OpenDRIVE*. Online. ASAM. © 2024. Dostupné z: <https://www.asam.net/standards/detail/opendrive/>. [cit. 2024-05-11].
- [18] DILLMANN, Nicco. *ASAM OpenDRIVE Webinar: Part 1*. Online, PDF. In: ASAM. 14th October 2020. Dostupné z: <https://www.asam.net/index.php?eID=dumpFile&t=f&f=3820&token=3f7d12b0573405baf8a9adac449551262d72968d>. [cit. 2024-05-11].
- [19] ASAM. *OpenDRIVE® 1.8.0, Static Road Network Description*. Online. 2023. Dostupné z: https://publications.pages.asam.net/standards/ASAM_OpenDRIVE/ASAM_OpenDRIVE_Specification/latest/specification/index.html. [cit. 2024-05-11].
- [20] *RoadRunner*. Online. MathWorks. © 1994-2024. Dostupné z: https://www.mathworks.com/help/roadrunner/index.html?s_tid=CRUX_lftnav&status=SUCCESS. [cit. 2024-05-11].
- [21] *Export Scenes*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/roadrunner/export-scenes.html>. [cit. 2024-05-11].
- [22] *Truevision Designer*. Online. Truevision. © 2024. Dostupné z: <https://www.truevision.ai/designer>. [cit. 2024-05-11].
- [23] *Automated Driving Toolbox*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/>. [cit. 2024-05-12].

- [24] *Scenario Simulation*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/scenario-simulation.html>. [cit. 2024-05-12].
- [25] *Cuboid Scenario Simulation*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/cuboid-scenario-simulation.html>. [cit. 2024-05-19].
- [26] *Driving Scenario Designer*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ref/drivingscenariodesigner-app.html>. [cit. 2024-05-20].
- [27] *Unreal Engine Simulation for Automated Driving*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ug/3d-simulation-for-automated-driving.html>. [cit. 2024-05-19].
- [28] *Overview of Simulating RoadRunner Scenarios with MATLAB and Simulink*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ug/overview-of-co-simulating-roadrunner-with-matlab-and-simulink.html>. [cit. 2024-05-19].
- [29] *CARLA Simulator*. Online. © 2024. Dostupné z: <https://carla.org/>. [cit. 2024-05-12].
- [30] *Introduction*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/start_introduction/. [cit. 2024-05-12].
- [31] *First steps*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/tuto_first_steps/. [cit. 2024-05-12].
- [32] *Sensors and data*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/core_sensors/. [cit. 2024-05-12].
- [33] *Maps*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/core_map/. [cit. 2024-05-12].
- [34] *Generating Maps in RoadRunner*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/tuto_M_generate_map/. [cit. 2024-05-12].
- [35] *Digital Twin Tool*. Online. CARLA Simulator. Dostupné z: https://carla.readthedocs.io/en/latest/adv_digital_twin/#building-the-osm-renderer. [cit. 2024-05-12].
- [36] *NVIDIA DRIVE CONSTELLATION*. Online, PDF. NVIDIA. © 2019. Dostupné z: <https://resources.nvidia.com/en-us-auto-constellation/drive-constellation>. [cit. 2024-05-12].
- [37] GREENSTEIN, Zvi. *A Path for Safe Self-Driving: NVIDIA Opens DRIVE Constellation Platform to Simulation Partners*. Online. NVIDIA. September 12, 2018. Dostupné z: <https://resources.nvidia.com/en-us-auto-constellation/nvidia-opens-drive->

- constellation. [cit. 2024-05-12].
- [38] GOEL, Neha. *Building an Autonomous Vehicle (AV) Simulation Toolchain with Simulink, RoadRunner and NVIDIA DRIVE Sim*. Online. In: MathWorks. 2021. Dostupné z: <https://blogs.mathworks.com/student-lounge/2021/04/26/building-an-autonomous-vehicle-av-simulation-toolchain-with-simulink-roadrunner-and-nvidia-drive-sim/>. [cit. 2024-05-12].
- [39] GREENSTEIN, Zvi. *The Test Fleet of the Future Is Virtual: DRIVE Constellation Now Available*. Online. NVIDIA. March 18, 2019. Dostupné z: <https://resources.nvidia.com/en-us-auto-constellation/virtual-fleet>. [cit. 2024-05-12].
- [40] *CarMaker*. Online. IPG Automotive. © 2024. Dostupné z: <https://www.ipg-automotive.com/en/products-solutions/software/carmaker/>. [cit. 2024-05-12].
- [41] *Software*. Online. IPG Automotive. © 2024. Dostupné z: <https://www.ipg-automotive.com/en/products-solutions/software/>. [cit. 2024-05-12].
- [42] *Autonomous vehicles: Development of autonomous driving functions with simulation*. Online. IPG Automotive. © 2024. Dostupné z: <https://www.ipg-automotive.com/en/applications/autonomous-vehicles/>. [cit. 2024-05-12].
- [43] *Perception and localization*. Online. IPG Automotive. © 2024. Dostupné z: <https://www.ipg-automotive.com/en/applications/autonomous-vehicles/perception-localization/>. [cit. 2024-05-12].
- [44] *CarMaker User's Guide*. PDF. IPG Automotive, 2020.
- [45] *Poskytování geodat*. Online. Geoportál ČÚZK. © 2010. Dostupné z: [https://geoportal.cuzk.cz/\(S\(khiodgep2qpvligezesf5uoo\)\)/Default.aspx?mode=eShop&head_tab=sekce-01-gp&menu=13](https://geoportal.cuzk.cz/(S(khiodgep2qpvligezesf5uoo))/Default.aspx?mode=eShop&head_tab=sekce-01-gp&menu=13). [cit. 2024-05-23].
- [46] *INSPIRE téma Nadmořská výška (EL) - GRID*. Online. Geoportál ČÚZK. © 2010, poslední aktualizace informací: 2023-07-03. Dostupné z: [https://geoportal.cuzk.cz/\(S\(1jvngr5gvuyc3jfwj1wbgnpp\)\)/Default.aspx?lng=CZ&mode=TextMeta&side=vyskopis&metadataID=CZ-CUZK-EL&mapid=8&menu=304](https://geoportal.cuzk.cz/(S(1jvngr5gvuyc3jfwj1wbgnpp))/Default.aspx?lng=CZ&mode=TextMeta&side=vyskopis&metadataID=CZ-CUZK-EL&mapid=8&menu=304). [cit. 2024-05-12].
- [47] *Ortofoto České republiky - úvod*. Online. Geoportál ČÚZK. © 2010, poslední aktualizace: 02.04.2024. Dostupné z: [https://geoportal.cuzk.cz/\(S\(1jvngr5gvuyc3jfwj1wbgnpp\)\)/Default.aspx?mode=TextMeta&text=ortofoto_info&side=ortofoto](https://geoportal.cuzk.cz/(S(1jvngr5gvuyc3jfwj1wbgnpp))/Default.aspx?mode=TextMeta&text=ortofoto_info&side=ortofoto). [cit. 2024-05-12].
- [48] *Road Height Tool*. Online. Mathworks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/roadrunner/ref/roadheighttool.html?status=SUCCESS>. [cit. 2024-05-13].

- [49] ÚŘAD PRO TECHNICKOU NORMALIZACI, METROLOGII A STÁTNÍ ZKUEŠBNICTVÍ. ČSN 73 6056, *Odstavné a parkovací plochy silničních vozidel*.
- [50] MINISTERSTVO DOPRAVY ČESKÉ REPUBLIKY. VL 6.1, *Svislé dopravní značky*. 2019.
- [51] *Seznam dopravních značek v Česku*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2024. Dostupné z: https://cs.wikipedia.org/wiki/Seznam_dopravn%C3%ADch_zna%C4%8Dek_v_%C4%8Cesku. [cit. 2024-05-13].
- [52] *Safetyshop.cz*. Online. © 2020. Dostupné z: <https://www.safetyshop.cz/>. [cit. 2024-05-15].
- [53] *Bahnschrift font family*. Online. In: Microsoft. 03/30/2022. Dostupné z: <https://learn.microsoft.com/en-us/typography/font-list/bahnschrift>. [cit. 2024-05-18].
- [54] *Fir Tree Branch Texture Clipart*. Online, obrázek. In: PikPng. © 2019. Dostupné z: https://www.pikpng.com/pngvi/ixomowT_share-this-image-fir-tree-branch-texture-clipart/. [cit. 2024-05-15].
- [55] *Low Poly Tree Oak*. Online, 3D model. In: Free3D. 2017, last updated Nov 20, 2019. Dostupné z: <https://free3d.com/3d-model/lowpolytree-oak-33972.html>. [cit. 2024-05-15].
- [56] RŮŽIČKOVÁ, Kateřina. *120xFSI: První brněnský „mrakodrap“*. Online. Vysoké učení technické v Brně, Fakulta strojního inženýrství. 28. 5. 2020. Dostupné z: <https://www.fme.vutbr.cz/fakulta/120/67880>. [cit. 2024-05-13].
- [57] *Unreal Engine Simulation Environment Requirements and Limitations*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ug/unreal-engine-simulation-environment-requirements-and-limitations.html>. [cit. 2024-05-13].
- [58] *Export to Unreal Using Filmbox (.fbx) File*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/roadrunner/ug/export-to-unreal-fbx.html>. [cit. 2024-05-13].
- [59] *Simulation 3D Vehicle with Ground Following*. Online. MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ref/simulation3dvehiclewithgroundfollowing.html>. [cit. 2024-05-13].
- [60] *Simulation 3D Camera*. Online. MathWorks. © 1994-2024. Dostupné z: https://www.mathworks.com/help/driving/ref/simulation3dcamera.html?s_tid=doc_ta. [cit. 2024-05-13].

- [61] WANG, Chaoyang; WANG, Xiaonan; HU, Hao; LIANG, Yanxue a SHEN, Gang. On the Application of Cameras Used in Autonomous Vehicles. Online. *Archives of Computational Methods in Engineering*. 2022, vol. 29, iss. 6, p. 4319-4339. ISSN 1134-3060. Dostupné z: <https://doi.org/10.1007/s11831-022-09741-8>. [cit. 2024-05-14].
- [62] *A Guide to Lidar Wavelengths for Autonomous Vehicles and Driver Assistance*. Online, obrázek. In: Velodyne Lidar. 2018, Updated 3/26/2021. Dostupné z: <https://velodynelidar.com/blog/guide-to-lidar-wavelengths/>. [cit. 2024-05-14].
- [63] MORRIS, Brendan. *Identifying E/E Architecture Requirements for Autonomous Vehicle Development*. Online, obrázek. In: EE Times Asia. 2021. Dostupné z: <https://www.eetasia.com/identifying-e-e-architecture-requirements-for-autonomous-vehicle-development/>. [cit. 2024-05-14].
- [64] *Create Driving Scenario Programmatically*. Online, obrázek. In: MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ug/create-driving-scenario-programmatically.html>. [cit. 2024-05-14].
- [65] *Create Driving Scenario Interactively and Generate Synthetic Sensor Data*. Online, obrázek. In: MathWorks. © 1994-2024. Dostupné z: <https://www.mathworks.com/help/driving/ug/create-driving-scenario-interactively-and-generate-synthetic-detections.html>. [cit. 2024-05-14].
- [66] TIDEMAN, Martijn. *NVIDIA DRIVE Sim Ecosystem Creates Diverse Proving Ground for Self-Driving Vehicles*. Online, obrázek. In: NVidia. April 12, 2021. Dostupné z: <https://blogs.nvidia.com/blog/drive-sim-ecosystem-diverse-proving-ground-self-driving-vehicles/>. [cit. 2024-05-14].
- [67] SEZNAM.CZ. *Mapy.cz*. Online. Brno. Letecká. 2022. Dostupné z: <https://mapy.cz/letecka?x=16.5773850&y=49.2248473&z=18>. [cit. 2024-05-14].

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

ADAS	Advanced driver-assistance system
ASAM	Association for Standardisation of Automation and Measuring Systems
ČSN	Česká technická norma
ČÚZK	Český úřadu zeměměřičský a katastrální
DIN	Deutsches Institut für Normung
ECU	Electronic Control Unit
FSI	Fakulta strojního inženýrství
GNSS	Global navigation satellite system
IMU	Inertial measurement unit
LRR	Long-range radar
MEMS	Micro-electromechanical system
MRR	Medium-range radar
RAM	Random-access memory
RTK	Real time kinematic
SAE	Society of Automotive Engineers
SRR	Short-range radar
SSL	Solid state lidar
ToF	Time of flight

SEZNAM PŘÍLOH

Přílohy zahrnuté v tomto dokumentu

Příloha A: Porovnání vytvořeného světa s realitou	I
Příloha B: Skript k výpočtu trajektorie	XVII

Samostatné přílohy

Příloha C: Ověřovací simulace v Simulinku

PŘÍLOHA A: POROVNÁNÍ VYTVOŘENÉHO SVĚTA S REALITOU

Na následujících stranách je vždy dvojice obrázků, kdy horní obrázek je fotografie pořízená 30. 4. 2024 a dolní obrázek je screenshot z virtuálního světa pořízeny v Unreal Engine.



Obr. 37 Přijezd k místu napojení dolní větve ulice Technické na větev horní



Obr. 38 Ulice Technická před rozdělením do dvou větví



Obr. 39 Křižovatka horní větve Technické s ulicí U Vodárny



Obr. 40 Vjezd do areálu FSI z ulice U Vodárny



Obr. 41 Horní větev ulice Technické



Obr. 42 Horní větev ulice Technické – prostranství před budovou A1



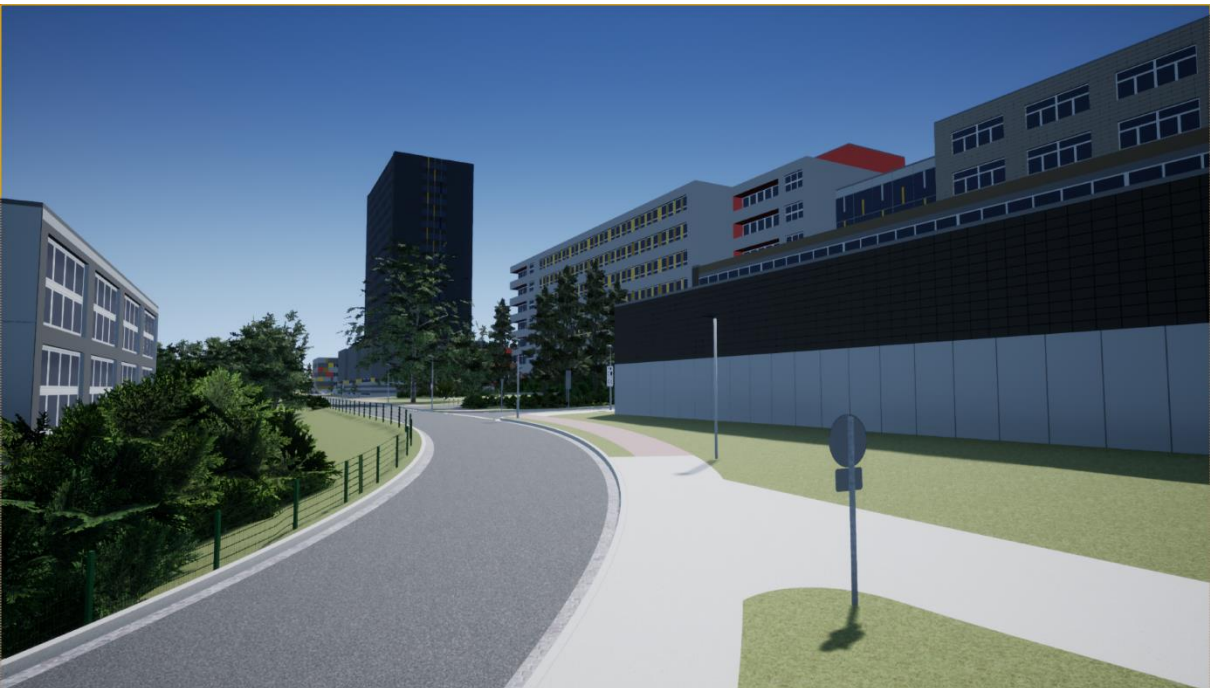
Obr. 43 Horní větev ulice Technické – parkoviště před budovou A1



Obr. 44 Horní větev ulice Technické



Obr. 45 Prostranství před budovou A4 a vjezd do areálu FSI z horní větve ulice Technické



Obr. 46 Horní větev ulice Technické



Obr. 47 Dolní větev ulice Technické



Obr. 48 Dolní větev ulice Technické



Obr. 49 Dolní větev ulice Technické



Obr. 50 Dolní větev ulice Technické



Obr. 51 Dolní větev ulice Technické



Obr. 52 Dolní větev ulice Technické – přechod u lávky přes ulici Hradeckou

PŘÍLOHA B: SKRIPT K VÝPOČTU TRAJEKTORIE

Níže je uveden skript, který byl použit ke stanovování trajektorie vozidel, která se pohybují vytvořeným světem v průběhu ověřovací simulace. Použit by ho ale bylo možné i k výpočtu trajektorie chodců a cyklistů. Skript byl naprogramován v softwaru MATLAB ve verzi R2023b. Aby byl skript funkční, musí být nainstalován doplněk Automated Driving Toolbox.

VSTUPNÍ HODNOTY

Vstupem do skriptu je matice o pěti sloupcích. Tu zde reprezentuje *tab. 2*. Konkrétně se jedná o výřez ze vstupních dat, která byla použita k výpočtu trajektorie vozidla s kamerou.

Tab. 2 Příklad vstupních hodnot

Bod	1. souřadnice	2. souřadnice	Natočení	Rychlost	Doba stání
	x (m)	y (m)	φ (°)	v (m/s)	t (s)
35	242,790	-187,867	-64,29	30	0
36	254,435	-211,907	-64,29	5	0
37	255,040	-213,151	-64,29	0	2
38	255,040	-213,151	-64,29	0	0
39	255,777	-214,687	-64,29	5	0
40	263,125	-219,898	6,76	10	0
41	288,978	-214,066	18,08	30	0
42	309,193	-207,458	18,08	30	0
43	334,200	-215,506	-47,81	40	0

V **prvních dvou sloupcích** matice jsou souřadnice zvolených bodů, kterými má vozidlo projet. Ty jsou vztaženy k počátku virtuálního světa.

Ve **třetím sloupci** je požadované natočení vozidla kolem svislé osy (anglicky „yaw“) v jednotlivých zvolených bodech.

Ve **čtvrtém sloupci** je rychlost, kterou se má vozidlo ve zvolených bodech pohybovat.

V **pátém sloupci** je doba, na kterou má vozidlo v daném bodě zastavit. Má-li vozidlo v nějakém bodě zastavit, musí být tento bod do matice zapsán dvakrát, přičemž doba zastavení se uvádí pouze v prvním z těchto dvou řádků.

VÝSTUPNÍ HODNOTY

Výstupem je časový vektor popisující trajektorii vozidla. Tato data se použijí jako vstupy do bloků v Simulinku:

- `positionX` – x-ová složka časového vektoru trajektorie vozidla
- `positionY` – y-ová složka časového vektoru trajektorie vozidla
- `positionT` – časový vektor natočení

SKRIPT

```
points = size(data,1);
time = zeros(points,1);
vel = data(:,4)/3.6;
positionX = [0,data(1,1)];
positionY = [0,data(1,2)];
positionT = [0,data(1,3)];

for i = 1:(points-1)
    if data(i,1) == data(i+1,1) && data(i,2) == data(i+1,2)
        dist(i) = 0;
        vel(i) = 0;
        time(i+1) = time(i)+data(i,5);
    else
        refPoses = data([i,i+1],[1,2,3]);
        refDirections = ones(2,1);
        sn = 3*sqrt((data(i+1,1)-data(i,1))^2 + (data(i+1,2)-data(i,2))^2);
        vn = (data(i+1,4)+data(i,4))/(2*3.6);
        n = round((sn/vn)*60)+1;
        numSmoothPoses = n;
        [poses,directions] =
smoothPathSpline(refPoses,refDirections,numSmoothPoses);

        X = poses(:,1);
        Y = poses(:,2);
        dist(i) = sum(vecnorm(diff([X(:),Y(:)]),2,2));
        time(i+1) = time(i)+(2*dist(i))/(vel(i+1)+vel(i));
        acc(i) = (vel(i+1)-vel(i))/(time(i+1)-time(i));
    end
end

for i = 1:(points-1)
    if data(i,1) == data(i+1,1) && data(i,2) == data(i+1,2)
        posX = [time(i)+data(i,5),data(i+1,1)];
        posY = [time(i)+data(i,5),data(i+1,2)];
        posT = [time(i)+data(i,5),data(i+1,3)];
    else
        refPoses = data([i,i+1],[1,2,3]);
        refDirections = ones(2,1);
        numSmoothPoses = n;
        [poses,directions] =
smoothPathSpline(refPoses,refDirections,numSmoothPoses);

        if data(i,4) == data(i+1,4)
            timeVector = linspace(time(i),time(i+1),n);
            timeVector = timeVector';
        end
    end
end
```

```
else
    step = dist(i)/(n-1);
    timeVector = zeros(n,1);
    timeVector(1) = time(i);
    timeVector(n) = time(i+1);
    for j = 1:(n-2)
        timeVector(j+1) = time(i)+((-
vel(i)+sqrt((vel(i))^2+2*acc(i)*step*j))/acc(i));
    end
end

posX = [timeVector(2:n),poses(2:n,1)];
posY = [timeVector(2:n),poses(2:n,2)];
posT = [timeVector(2:n),poses(2:n,3)];
end
positionX = [positionX;posX];
positionY = [positionY;posY];
positionT = [positionT;posT];
end
```