

3D Vizualizace současného Zlína s pomocí modulu BlenderGIS

Tomáš Bukový



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Tomáš Bukový
Osobní číslo:	A19009
Studijní program:	B3902 Inženýrská informatika
Studijní obor:	Softwarové inženýrství
Forma studia:	Prezenční
Téma práce:	3D Vizualizace současného Zlína s pomocí modulu BlenderGIS
Téma práce anglicky:	3D Visualisation of Contemporary Zlín with the Help of BlenderGIS

Zásady pro vypracování

1. Seznamte se s softwarovými nástroji, které budou použity v praktické části této práce.
2. Za pomocí programu Blender nástroje BlenderGIS vygenerujte model současného Zlína.
3. Ručně domodelujte chybějící části města a aplikujte vhodné materiálové vlastnosti.
4. Proveďte export tohoto modelu z Blenderu do Unreal Engine.
5. V Unreal Engine vytvořte jednoduchou vizualizační aplikaci a navrhnete možné grafické vylepšení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Blender 2.93 Reference Manual — Blender Manual. Blender Documentation — blender.org [online]. Copyright © [cit. 2021-9-26]. Dostupné z: <https://docs.blender.org/manual/en/latest/index.html>
2. Unreal Engine 4 Documentation. Unreal Engine Documentation — unrealengine.com [online]. Copyright © [cit. 2021-9-26]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US>
3. Home · domlysz/BlenderGIS Wiki. BlenderGIS Wiki — github.com [online]. Copyright © [cit. 2021-9-26]. Dostupné z: <https://github.com/domlysz/BlenderGIS/wiki>
4. BRITO, Allan. Blender 2.9: The beginner's guide. Independently published, 2020. ISBN 979-8676661700.
5. NIXON, David. Unreal Engine 4 for Beginners. Luquinox, 2017. ISBN 978-0692841983.

Vedoucí bakalářské práce:

Ing. Pavel Pokorný, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 18.5.2022

Tomáš Bukový, v.r.
podpis studenta

ABSTRAKT

Cílem této práce je za použití rozšíření BlenderGIS do aplikace Blender vytvořit a ručně dopravit model současného Zlína, aplikovat materiály na budovy a následně v prostředí programu Unreal Engine vytvořit vizualizační aplikaci.

Klíčová slova: Vizualizace, Krajina, Budovy, Blender, Unreal Engine, Zlín, 3D model

ABSTRACT

The goal of this thesis is to create and correct the model of contemporary Zlín with a help of BlenderGIS, manually apply materials to the building and then create a visualisation in the Unreal Engine environment.

Keywords: Visualisation, Landscape, Buildings, Blender, Unreal Engine, Zlín, 3D model

Chtěl bych vyjádřit poděkování panu Ing. Pavlu Pokornému, Ph.D. za jeho cenné připomínky a rady při vypracovávání této práce. Dále bych chtěl poděkovat rodině za jejich podporu po celou dobu studia.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 BLENDER.....	11
1.1.1 Historie	11
1.1.2 Popis prostředí.....	13
1.1.3 Funkce	14
1.1.3.1 Renderování	14
1.1.3.2 Modelování	14
1.1.3.3 Grease Pencil	15
1.1.3.4 Sculpting.....	15
1.1.3.5 Animování	15
1.1.3.6 Simulace.....	15
1.1.3.7 Video editace	15
1.1.3.8 Pipeline	16
1.1.3.9 Skriptování.....	16
1.1.3.10 Exportování a importování.....	16
1.1.4 Materiály a textury	16
1.2 BLENDERGIS	17
1.2.1 Popis.....	17
1.2.2 Funkce	18
1.2.3 Web Geodata	18
1.2.3.1 Base map.....	18
1.2.3.2 OSM.....	19
1.2.3.3 SRTM.....	20
1.2.4 Import.....	21
1.2.4.1 Shapefile	21
1.2.4.2 Georaster.....	21
1.2.4.3 OpenStreetMap	21
1.2.5 Export.....	21
2 UNREAL ENGINE	22
2.1.1 Historie	22
2.1.2 Epic Games Launcher	24
2.1.3 Popis uživatelského prostředí.....	24
2.1.3.1 Úvodní obrazovka.....	24
2.1.3.2 Pracovní plocha projektu	26
2.1.4 Funkce	27
2.1.4.1 Import a export.....	27
2.1.4.2 Budování světa.....	27
2.1.4.3 Editor	27
2.1.4.4 Animace	28
2.1.4.5 Světlo a vykreslování.....	28
2.1.4.6 Materiály.....	28
2.1.4.7 Simulace.....	28
2.1.4.8 Herní prvky	28
2.1.4.9 Platformy	29
2.1.4.10 Blueprints	29

II PRAKTICKÁ ČÁST	30
3 VYGENEROVÁNÍ MODELU Z BLENDERGIS.....	31
3.1 BASEMAPA	31
3.2 OSM	32
4 DODĚLÁNÍ CHYBĚJÍCÍCH ČÁSTÍ MODELU V BLENDERU.....	35
4.1 PŘIDÁNÍ REFERENČNÍHO OBRÁZKU.....	36
4.2 TVORBA DETAILU BUDOVY.....	37
4.2.1 Základní tvar budovy	37
4.2.2 Vytvoření vodících bodů na budově	39
4.2.3 Materiály	40
4.2.4 Výsledná budova	41
4.3 TVORBA SILNIC A CHODNÍKU.....	42
4.4 ZMĚNA TEXTURY MAPY NA MATERIÁL	43
4.5 TVORBA KOLEJÍ.....	43
4.6 KORYTO ŘEKY.....	45
4.7 STROMY	46
4.8 SEMAFORY A ZASTÁVKY PRO MHD	47
5 EXPORT DO UE A OPRAVY CHYB MODELU	49
5.1 OPRAVA NORMÁLOVÉ MAPY	49
5.2 EXPORTOVÁNÍ	50
5.2.1 Vyčištění modelu	50
5.2.2 Zmenšení modelu	50
5.2.3 Vybrání částí modelu	51
5.2.4 Exportování	51
6 TVORBA APLIKACE V UNREAL ENGINE	52
6.1.1 Vytvoření nového projektu	52
6.1.2 Importování	52
6.1.3 Vytvoření BlueprintClass pro modely	52
6.1.4 Vložení modelu do scény	53
6.1.5 Kolizní model terénu a silnice.....	53
6.1.6 Opravy kolizního modelu u určitých budov.....	54
6.1.7 Vložení zeleně do modelu.....	55
6.1.8 Tvorba hlavního menu	56
6.1.9 Tvorba pauzy.....	57
6.1.10 Vytvoření tratě a automatické jízdy	58
6.1.11 Volný pohled a zrychlení vizualizační aplikace.....	59
6.1.12 Zabalení do spustitelného souboru.....	59
7 VÝSLEDKY PRÁCE	60
ZÁVĚR	61
SEZNAM POUŽITÉ LITERATURY.....	62
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	65
SEZNAM OBRÁZKŮ	66
SEZNAM TABULEK.....	68
SEZNAM PŘÍLOH.....	69

ÚVOD

Z důvodu nezastavitelného pokroku v moderních technologiích v posledním desetiletí se 3D počítačová grafika stala součástí života každého z nás. Ubylo omezení, s nedostatkem výkonu počítačů minulých a díky tomu můžeme tvořit realistické scény a rozsáhle otevřené světy. Dnešní počítače zvládají úkoly, o kterých se před 20 lety lidem ani nesnilo. A díky dostupnosti kvalitních programů, mnohé z nich jsou zdarma, si můžeme vytvářet vlastní virtuální světy a uvolnit svoji fantazii z pohodlí domova, stačí nám k tomu výkonný počítač, jehož vlastnictví není již privilegium pro pár vyvolených a programy jako je Blender a Unreal Engine nám tuto tvorbu ulehčí.

Kombinace programů Blender a Unreal Engine nám dává neomezené možnosti, co se týče tvorby projektů, můžou to být vizualizační aplikace jako je tato bakalářská práce anebo rovnou celé počítačové nebo mobilní hry. A díky rozšíření jako je BlenderGIS si můžeme nechat vygenerovat jakékoliv město, které nás napadne.

Pro potřeby bakalářské práce byly využity programy Blender ve verzi 2.93.4 a Unreal Engine ve verzi 4. V případě Blenderu se nejedná o nejnovější verzi z důvodu problému kompatibility mezi Blenderem a rozšířením BlenderGIS, které bylo stěžejným nástrojem pro vytvoření této práce.

Teoretická část popisuje výše zmíněné programy a rozšíření. Popisuje je včetně jejich historie, demonstruje jejich funkce a možnosti a také popisuje uživatelské prostředí. V neposlední řadě zmiňuje také instalaci rozšíření BlenderGIS.

Praktická část se zabývá tvorbou modelu města Zlín v prostředí aplikace Blender za použití dříve zmiňovaného rozšíření a poté popíše tvorbu dodatečných modelů a úpravu budov. K této části mi byla poskytnuta i část hotových modelů panem Ing. Pavlem Pokorným, Ph.D. [30]

Poté se zde popisuje práce exportování výsledného modelu z Blenderu a jeho importování do Unreal Engine a řešení problému, které vznikly při tomto procesu. Finální částí je tvorba vizualizační aplikace.

I. TEORETICKÁ ČÁST

1 BLENDER

Blender je známý a volně dostupný, multiplatformní a open-source program určený pro práci ve 3D, jehož předností je modelování a animování ve 3D prostředí. V základu je program vybaven nástroji pro modelování, animování, simulaci nebo sledování pohybu. Další nástroje a rozšíření si může uživatel stáhnout formou add-onů. Díky volné dostupnosti a nulové pořizovací ceně je populární jak mezi profesionály tak také mezi obyčejnými lidmi.

Mezi klíčové vlastnosti patří také využití rozhraní OpenGL, které zaručuje totožné a konsistentní prostředí na všech podporovaných platformách. Blender umožňuje využití portable verze formou spustitelného souboru .exe, který můžete mít třeba na flash disku a spouštět ho bez nutnosti instalace. [1]

1.1.1 Historie

V roce 1988 se Ton Roosendaal spoluúčastnil na založení nizozemského animačního studia NeoGeo. Společnost se poté stala největším animačním studiem v celém Nizozemsku a vůdčí společností Evropy. Ton byl zodpovědný jak za umělecký směr, tak také za vývoj softwaru. Po zvážení se rozhodl, že současná 3D sada nástrojů byla zastaralá a obtížná na údržbu a proto ji bylo nutné přepsat od základu. V roce 1995 začal vytvářet nový software, který známe dnes pod názvem Blender. Postupem času jak NeoGeo rozšiřovalo a vylepšovalo Blender bylo zřejmé pro Tonyho, že by se Blender dal využít i mimo studio NeoGeo a být dostupný ostatním umělcům.

V roce 1988 založil společnost Not a Number, zkráceně NaN, jako vedlejší produkt společnosti NeoGeo s úkolem propagovat a vyvíjet Blender. Cílem NaN bylo vytvořit a distribuovat kompaktní, multiplatformní 3D aplikaci zdarma. Na tuto dobu to byl revoluční koncept, jelikož většina podobných produktů stála v řádech tisíců amerických dolarů. NaN chtělo dostat do rukou veřejnosti 3D modelovací a animační software na profesionální úrovni. V roce 1999 se NaN zúčastnilo SIGGRAPH konference s cílem propagovat Blender. První konvence Blenderu měla obrovský úspěch a získala obrovský zájem z řad tisku a účastníků.

Díky úspěchu na konferenci SIGGRAPH si v roce 2000 společnost NaN zajistila finanční obnos ve výši 4,5 milionů eur. Díky tak vysokému finančnímu obnosu mohla společnost NaN zvýšit pole působnosti. Chvilí nato až 50 zaměstnanců pracovalo po celém

světě, aby vylepšovali a promovali Blender. V létě 2000 vyšla verze 2.0, která přidala integraci do herních enginů do své aplikace. Koncem roku bylo na stránkách NaN zaregistrováno přes 250 000 uživatelů.

Schopnosti společnosti NaN nebyly na tak vysoké úrovni, aby se vyrovnali jejich vlastním ambicím a trhu té doby, a proto se v roce 2001 uskutečnil restart celé společnosti s novým investorem a menším týmem. O šest měsíců později vydala společnost NaN první komerční produkt, Blender Publisher, tento produkt cílil na rychle rostoucí trh s interaktivními 3D medii. V reakci na nepříznivé prodeje se nový investor rozhodl ukončit společnost NaN, což také znamenalo ukončení vývoje a práce na Blenderu. Sice současná verze Blenderu měla řadu problému, nedokončené funkce nebo nestandardní implementaci GUI, tak i přes obrovskou komunitu lidí podporující Blender jako takový, tak zákazníků, kteří si koupili Blender Publisher se Ton Roosendaal rozhodl v květnu roku 2002 založit nevýdělečnou organizaci *Blender Foundation*. Cílem bylo najít způsob jak vyvíjet Blender jako komunitní open-source aplikaci.

Kampaň *Free Blender* se snažila vybrat 100 000 euro, aby mohla od společnosti NaN odkoupit práva a zdrojový kód a následně vydat Blender jako komunitní open-source aplikaci. Díky obrovskému zájmu z řad dobrovolníků, mezi nimi byly i zaměstnanci společnosti NaN, se povedla tato částka vybrat pouze za 7 dní. V říjnu roku 2002 byl Blender vydán pod licencí GNU GPL. [2]

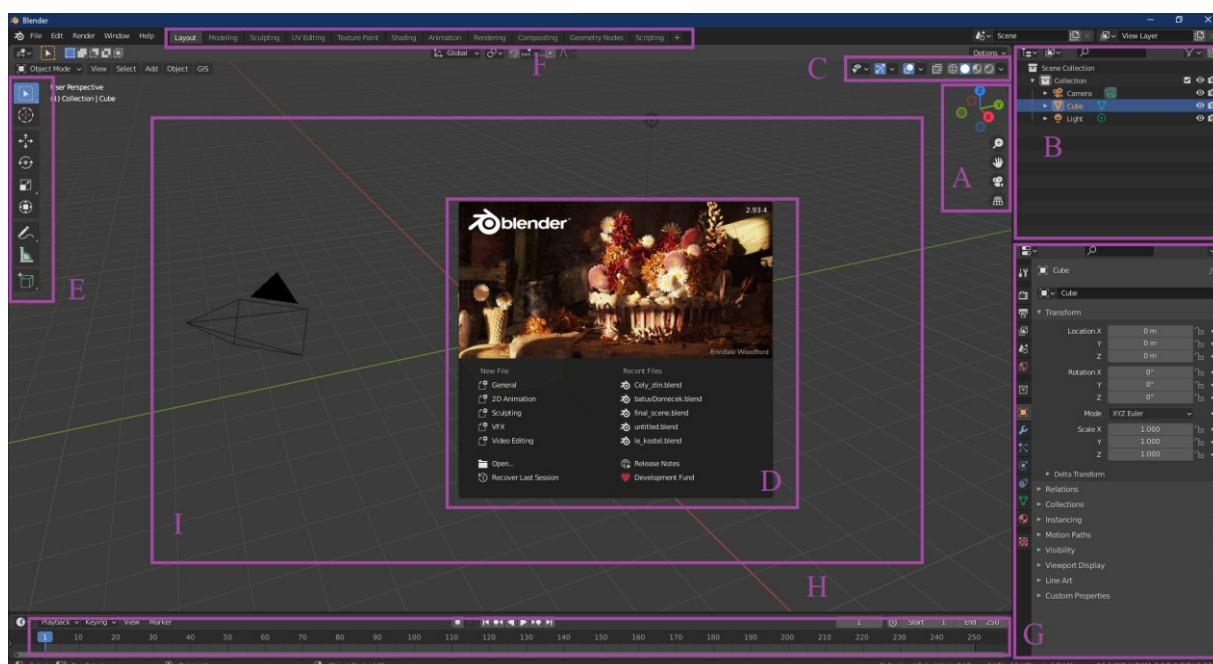
Tabulka 1. Hlavní verze

Verze 1.0	Leden 1994
Verze 2.0	Srpen 2000
Verze 3.0	Prosinec 2021

1.1.2 Popis prostředí

Prostředí aplikace Blender se dá rozdělit na 9 částí, v mém případě jsem využil označení od A do I. Příslušná označení s písmeny jsou zobrazeny v obrázku číslo 1. Označení nesou následovné vysvětlení: [29 – Kapitola 1, Strana 16]

- A. Rotace po osách a změna pozice kamery, přiblížení a perspektivy
- B. Seznam objektů, které obsahuje scéna – lze skládat do složek, přejmenovávat nebo dočasně skrýt
- C. Přepínání režimu zobrazení Viewportu – Změna mezi wireframe, materiály nebo zobrazení pomocí renderovacího režimu.
- D. Úvodní okno, které se zobrazí při zapnutí aplikace – Možnost vytvořit nový projekt nebo vybrat z posledně otevřených
- E. Nástroje pro práci – Mění se v závislosti na vybraný režim mezi Editačním či Objektovým
- F. Záložky pro přepínání pracovního režimu
- G. Okno pro podrobnější nastavení vlastností materiálu, renderu, objektů nebo třeba kamery
- H. Časová osa sloužící k tvorbě animací
- I. Pracovní plocha



Obrázek 1. Uživatelské prostředí

1.1.3 Funkce

Blender je program s kompletním balíčkem funkcí vhodnou jak pro modelování tak animování. Mezi nejpodstatnější části patří

1. Renderování
2. Modelování
3. Grease Pencil
4. Sculpting
5. Animování
6. Simulace
7. Video Editace
8. Pipeline
9. Skriptování

1.1.3.1 *Renderování*

Uživatel má k dispozici dva renderovací enginy, Eevee a Cycles. Eevee je realtime renderovací engine využívající OpenGL. Oproti druhému jmenovanému je méně náročnější a hodí se na renderování rozsáhlejších scén nebo na náhled materiálů ve 3D viewportu. Eevee není raytrace engine a tudíž nepočítá každý paprsek světla zvlášť, využívá místo toho proces rasterizace. Tento proces vypočítává dopad jednotlivých paprsků na objekt, a jak světlo interaguje s daným objektem pomocí algoritmů [9]

Oproti tomu Cycles podporuje raytracing, vypočítávání každého paprsku světla zvlášť, a díky tomu dokáže vyprodukovat mnohem kvalitnější výstupy za cenu vyšší náročnosti na výkon a vyšší časovou náročnost. Z tohoto důvodu také podporuje akceleraci pomocí grafické karty a to buď pomocí CUDA jader u karet značky NVIDIA nebo OPENCL od značky AMD. [10] [8]

1.1.3.2 *Modelování*

Uživatel má k dispozici rozsáhlou sadu nástrojů pro tvorbu, transformování a editaci modelů. Blender nám práci usnadňuje ale i urychluje díky implementaci klávesových zkratk. Přítomnost mřížky v prostředí programu také umožňuje zpřesnit naši práci. Můžeme manipulovat jak s celým objektem tak jeho částí, například pouze s jednou hranou. Program také podporuje modifikátory, které jsou automatické operace, které ovlivňují objekt v nedestruktivní formě. Modifikátory ulehčují operace, které by byly velmi náročné vytvo-

řit manuálně, například dělení povrchu, bez toho aniž bychom ovlivnili základní geometrii objektu. Při odstranění modifikátoru bude objekt v původním stavu. [5]

1.1.3.3 Grease Pencil

Jedná se o nástroj, který umožňuje kreslit přímo ve 3D viewportu programu Blender. Umožňuje tvořit 2D animace nebo skicování [6]

1.1.3.4 Sculpting

Nástroje v této poskytují možnost upravovat objekty formou sochařství. Samotný proces lze přirovnat k modelování s hlinou. Dá se využít pro pracování na dizajnu postav nebo na tvorbu terénu. V základu má Blender k dispozici 20 štětců. [7]

1.1.3.5 Animování

Blender disponuje s balíčkem nástrojů, který umožní uživateli ze statické postavy vytvořit animovanou. Mezi funkce patří například editor pro animaci póz postav, nelineární animace pro nezávislé pohyby nebo synchronizaci zvuku s animací. Animaci tvoříme umístěním snímků (*keyframes*) na časovou osu a tím se jim přiřadí vlastnost. Z jednotlivých rozdílů mezi *keyframes* se poté tvoří animace. [11]

1.1.3.6 Simulace

V Blenderu je k dispozici částicový systém umožňující přidání do scény simulaci ohně, kouře nebo například látky. [12]

1.1.3.7 Video editace

Součástí Blenderu je také základní editor videa a zvuku. Zvládá jednoduché úkoly jako stříhat klipy ale také složitější operace jako maskování nebo úpravu barev. Editor zvládne živý náhled, mixovat a synchronizovat zvuky, anebo vizualizaci zvukových vln. Editor také podporuje 32 slotů na přidávání videí, obrázků, mask a nebo efektů. Zvládne také upravovat vrstvy, přechody, rychlost a práci s filtry. [13]

1.1.3.8 Pipeline

Blender má podporu pro importování a exportování pro velkou škálu programu. Obrázky zvládne importovat/exportovat v obyčejném JPEG nebo PNG ale také OpenEXR, DPX nebo SGI iris. Video poté zvládá AVI, MPEG a Quicktime, který je využíván platformou OSX. [14]

1.1.3.9 Skriptování

Díky obrovské komunitě má Blender k dispozici velký obsah různých rozšíření a skriptů. Blender podporuje jak oficiální tak neoficiální rozšíření. Uživatel si je může kdykoliv aktivovat v nastavení aplikace. Můžou to být generátory stromů nebo jiné zeleně, terénu anebo oblohy. Rozšíření se můžou také týkat nových způsobů importování a exportování. [15]

1.1.3.10 Exportování a importování

Blender podporuje exportování a importování jak 2D tak 3D objektů. Ze základu podporuje řadu formátů jako je ABC, USD, OBJ nebo třeba FBX a další můžou být přidány formou rozšíření jako je například formát .osm pro geografická data. Mezi poslední přidané formáty byl Alembic (.abc), který byl přidán v aktualizaci 2.78. Formát je zaměřen pro ukládání. Pro potřeby práce byl při exportování objektu z Blenderu využíván formát FBX. [14]

1.1.4 Materiály a textury

Materiály v Blenderu se starají o to, aby jak bude následný objekt vypadat. Starají se o to, jak bude objekt reagovat na světlo, jakou bude mít barvu.

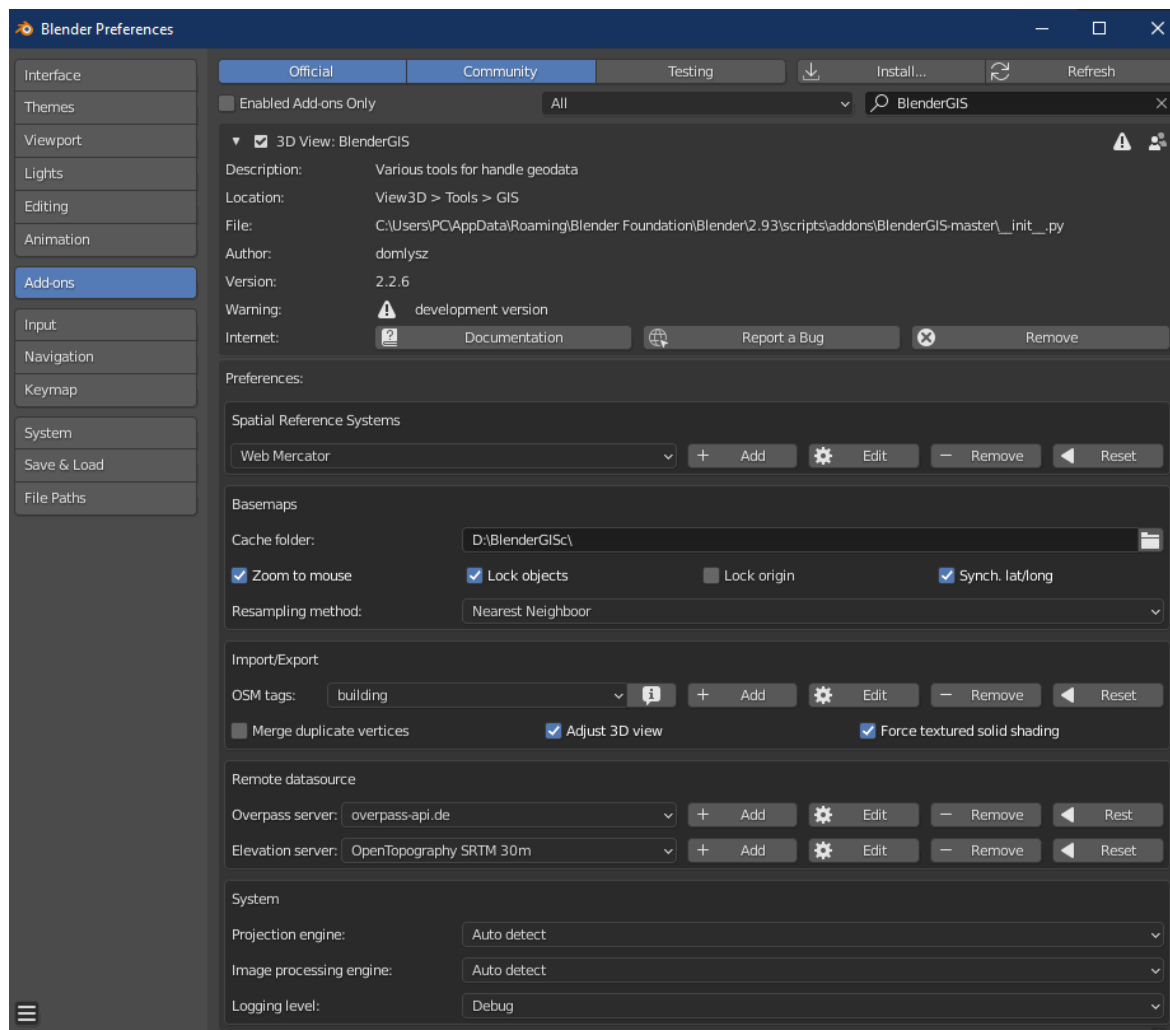
Materiály můžeme tvořit v Materiálových vlastnostech objektu. Zde pomocí různých posuvníků a změn hodnot docílíme požadované výstupu. Každý objekt může mít jeden nebo více materiálů. Pomocí tzv. *Nodes (Uzlů)* můžeme vytvořit pokročilé pokročilé materiály, ty můžou imitovat například sklo nebo kovový povrch. Textura je materiál, který má na sobě nanesený 2D obrázek. Procesu nanášení textury na materiál se říká mapování. [26]

1.2 BlenderGIS

1.2.1 Popis

Jedná se o neoficiální rozšíření, slouží k importování geografických dat do prostředí programu Blender. Uživatel si díky tomuto rozšíření může do svého projektu importovat mapu například z Google map nebo z alternativ jako je open street map jako plane objekt, ať už formou souboru .shapefile, geografickým rastrem nebo přímo souborem open street map .osm. Další možností je přímo mapu vygenerovat v samotném programu Blender. Rozšíření poté disponuje řadou nástrojů pro práci s touto mapou. V této části se budu věnovat možnostem rozšíření. [16]

BlenderGIS není oficiální rozšíření a proto se instalace liší oproti oficiálním rozšířením. Na oficiální stránce rozšíření [16], která je na Githubu se dá rozšíření stáhnout. Pro stažení není potřeba žádný účet a jediné co potřebujeme je kliknout na svítivě zelené tlačítko „Code“ a vybrat položku stáhnout jako zip. Druhým způsobem je použít příkazový řádek a pomocí příkazu „git clone“ s dosazenou adresou a stáhnout si ho takto. Po stažení si zapneme program Blender a v listě nástrojů vybereme položku Edit a v ní Preferences. Vyskočí nám okno s nastavením Blenderu a vybereme položku Add-ons. V pravém horním rohu bude tlačítko „Install...“, na to klikneme a v průzkumníku souborů vybereme náš stažený zip soubor. Po nainstalování tlačítkem pro obnovení si obnovíme seznam addonů a vyhledáme BlenderGIS. Pro jeho používání je ho nutné aktivovat a to uděláme tak, že klikneme na zaškrkávací tlačítko vedle jména addonu. Pro správnou funkčnost si musíme vytvořit Cache složku. Ta bude sloužit pro ukládání dočasných a neměl by se v ní tvořit projekt. V nastavení addonu ji najdeme v kategorii Basemaps.



Obrázek 2. BlenderGIS nastavení

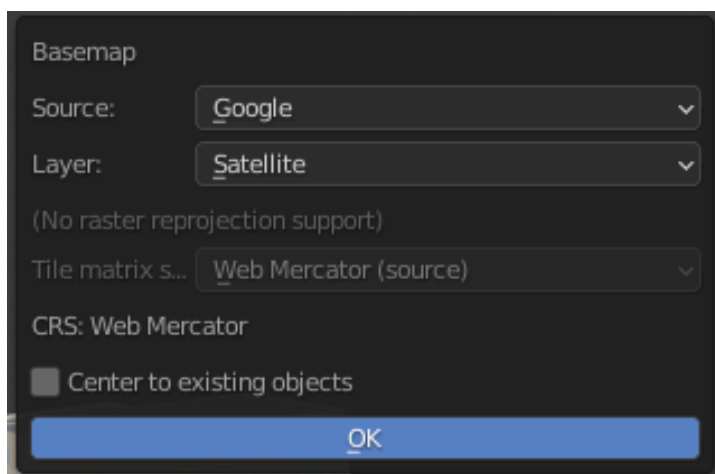
1.2.2 Funkce

1.2.3 Web Geodata

Jedná se o kategorii funkcí, která se zabývá získáváním geografických informací pomocí stahování potřebných dat ze vzdálených serverů.

1.2.3.1 Base map

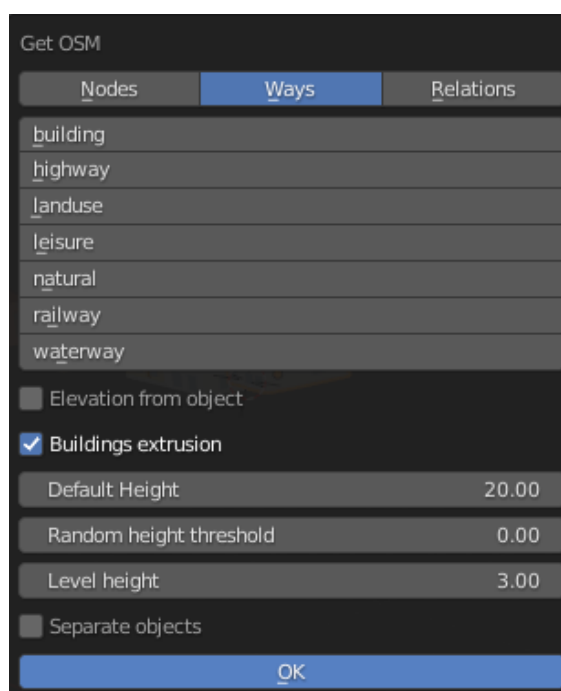
Tato funkce vygeneruje mapu jako objekt a vloží ji přímo do *viewportu*. Uživatel si může vybrat, ze kterého serveru budou data získávána. Je na výběr mezi Google, OSM, Bing, Ersi a OSM WMS. Další možností je výběr typu mapy. Pro funkčnost je nutné si nastavit složku, do které bude BlenderGIS vkládat Cache data. (viz obrázek č.2 – položka Basemaps). [17]



Obrázek 3. Nastavení Base mapy

1.2.3.2 OSM

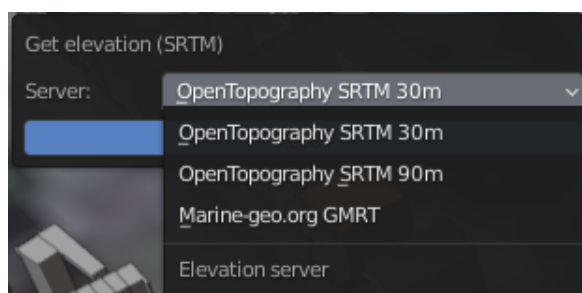
BlenderGIS umožňuje do existující mapy importovat OpenStreetMap data jako jsou domy, řeky, silnice nebo například parky. Existují 2 způsoby jak tyto data do projektu nahrát. Pomocí předem staženého souboru ve formátu *.osm* anebo přímo vygenerováním v aplikaci. Aby generování v aplikaci fungovalo je nutno mít vytvořenou Basemapu. Délka generování závisí na výkonu počítače a samotné velikosti generované oblasti a počtu vybraných částí. [18]



Obrázek 4. OSM

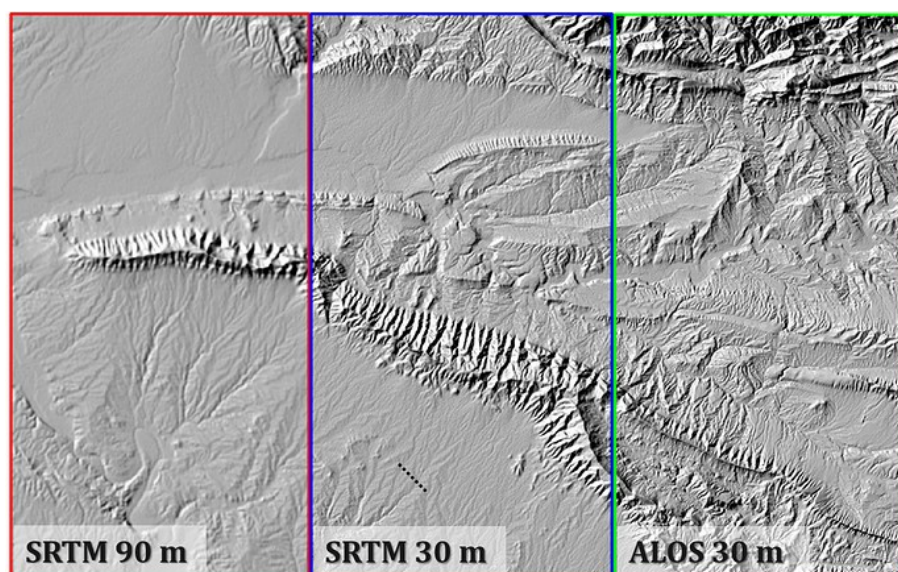
1.2.3.3 SRTM

SRTM je zkratka pro *Shuttle Radar Topography Mission*, který odkazuje na světový Digitální výškový model vytvořený společností NASA. Samotná databáze, ze které získává funkce data, není celosvětová a pokrývá plochu mezi 56°S jižní paralelou a 60°N severní paralelou. Pro veřejnost jsou dostupné datové balíčky v rozlišení 90m na pixel a pro Spojené státy americké to je 30m. Funkce SRTM nezvládne poskytnout data ve vyšším rozlišení. Funkce nám dává na výběr mezi dvěma servery, jedním je OpenTopography a druhým je Marine-geo.org.



Obrázek 5. STRM

Při využití serverů OpenTopograhya nám funkce dává na výběr mezi SRTM 30m a 90m. U verze s rozlišením 90m se jedná o převzorkovanou verzi o rozlišení 30m, která má za cíl zvýšení detailu u výškových rozdílů u údolí za cenu ztráty detailů u hřebenů pohoří. Tato verze poté působí hladším dojmem, jelikož se převzorkování chová jako *low pass filtr*. [19]



Obrázek 6. Rozdíly mezi SRTM 90m,30m a ALOS 30m z pohledu satelitu

1.2.4 Import

1.2.4.1 Shapefile

Rozšíření umožňuje importovat tak exportovat geografická data. Mezi nejpopulárnější formáty patří Shapefile. Jedná se geoprostorový vektorový využívaný geografickými informačními systémy. Díky tomuto formátu můžeme do Blenderu importovat většinu shapefilů obsahující geometrii i s osou Z. BlenderGIS nepodporuje objekty obsahující *multipatch*. *Multipatch* je GIS objekt, který uchovává informace o krajích objektu formou jednoho řádku v databázi. Atributy lze využít pro vytvoření výškových hodnoty Z nebo také hodnoty pro extrusion. Ve výchozím nastavení budou shapefile soubory importovány jako jeden objekt, je tu ale možnost je importovat jako separátní objekty. Tato volba není ale doporučovaná jelikož Blender není stavěný na to, aby zvládal tisíce objektů. Importování tímto způsobem může být pomalé.

1.2.4.2 Georaster

Jedná se o způsob importu, kdy uživatel vybere obyčejný obrázek a k němu přidá *worldfile*, který obsahuje georeference obrázku jako je například: lokace, měřítko nebo otočení na mapě. Druhou možností je použití geotiff souboru, který tyto informace má v sobě zabudované formou tiff tagů.

1.2.4.3 OpenStreetMap

Jedná se o způsob importování geografických dat formou souboru .osm, je to xml soubor, který můžeme získat ze stránek OpenStreetMap.org.

1.2.5 Export

Jediným způsobem exportu pro rozšíření BlenderGIS je za použití převodu *mesh* dat do souboru shapefile. Uživatel si může definovat, jestli se typy budou exportovat jako body, úsečka nebo polygony. Export podporuje 2 režimy, jak bude pracovat s vlastnostmi objektu. První možností je *Mesh to feature*, kdy se každý bod, úsečka nebo polygon *meshe* převedou jako nová vlastnost shapefilu. Druhá možnost je exportování celého objektu jako novou shapefile vlastnost. Pokud tedy objekt obsahuje víc než jednu plochu, úsečku nebo bod bude vytvořena vlastnost obsahující více částí.

2 UNREAL ENGINE

Unreal Engine (UE) je herní engine, který byl vytvořen a je stále aktivně vyvíjen společností EpicGames. Jedná se o sadu nástrojů, které pomáhají ve tvorbě herního obsahu. Za necelých 20 let získal obrovskou popularitu na trhu. Samotný Unreal Engine je k dispozici zdarma, pouze pokud vaše hra generuje vysoký zisk, je nutné platit poplatek za samotný engine. Ten činí 5% ze zisku aplikace, která překročila hrubý příjem 1 milion dolarů za sobu svojí existence. Engine je napsán v jazyce C++ a díky tomu má vysokou kompatibilitu mezi počítači, telefony, konzolami nebo virtuální realitou. V této části práce se budu zabývat historií a funkcemi tohoto programu.[21] [27]

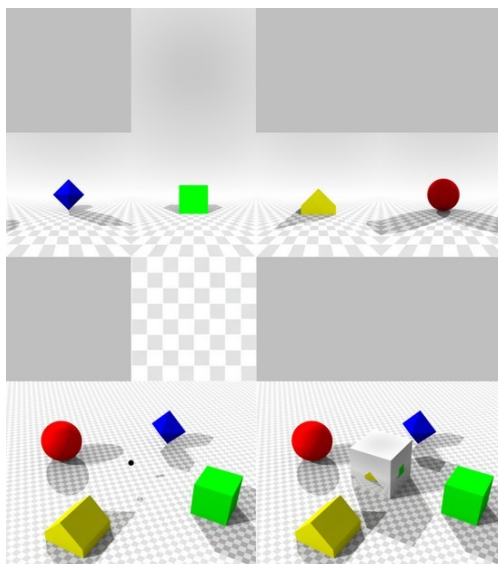
2.1.1 Historie

V roce 1995 začal Tim Sweeney, zakladatel Epic Games, vyvíjet herní engine pro novou hru Unreal. Hra Unreal měla být novým kouskem ve vzkvétajícím žánru her, byly to FPS střílečky jako DOOM, Wolfenstein 3D, Duke Nukem 3D a nebo Quake. Unreal byla vyvíjena jako hlavní konkurent hry od studia id Quake II. Hra patřila mezi nejlépe vypadající střílečky tohoto žánru té doby, s neobvyklou lehkostí zvládala pracovat s velkými a detailními lokacemi ať už vnitřních prostorů nebo venkovních. Samotný George Broussard, jeden z autorů serie Duke Nukem, byl tak překvapen hrou Unreal, že se rozhodl změnit engine u hry Duke Nukem Forever na Unreal Engine. Oproti konkurenci měla hra Unreal vlastní skriptovací jazyk *UnrealScript* i editor úrovní *UnrealEd*, což žádná hra té doby neměla. Po dobu roku 1998 dostával Unreal Engine nespočet opravných balíčků, které byly vydávány skoro každý měsíc. Každý update přidal nové technologie. [21][22]

Aktualizace Unreal Engine 1 se dělí na 2 části, první částí jsou sestavení od čísla 1 po sestavení 266, ty byly nazývány „Unreal“ a sestavení od 300 po 613 se nazývaly „Unreal Tournament“. UE1 podporoval několik renderovacích technologií, které byly postupně přidávány jednotlivými aktualizacemi. Od sestavení 216 UE1 podporoval DirectX ve verzi 5, DirectX 6 a od sestavení 226 DirectX 7. Po sestavení 500 přibyla podpora pro DirectX ve verzi 8. Podporoval také OpenGL, jehož podpora byla přidána v sestavení 209. UE podporuje klasické odrazy zrcadlových ploch, tak následně i od polotmavých ploch a ještě k tomu přidává možnost vidět přes teleporty, jak vypadá krajina na druhé straně. [20]

Vývoj UE2, nástupce UE1, probíhal mezi lety 2001 a 2005. První UE se dal rozdělit na 2 části, u UE2 si vývoj můžeme rozdělit na 4 období podle čísla sestavení. Prvním

obdobím je *Unreal Warfare Build*, jedná se sestavení, jenž přímo navazovalo na UE1 a přinášelo řadu vylepšení, mezi hlavní patří tzv. *cube mapping*. Jedná se o způsob zobrazování okolí, které se promítá na vnitřní straně krychle, kde se promění na 2D obraz. Scéna je tedy vyrenderována v každý moment 6x, přičemž každý pohled je otočen o 90°. V roce 2002 přišla změna podnázvu na *Unreal Tournament 2003*. Specifickou etapou byl *Pure Unreal Engine 2*. Jednalo se o spojení prvků z *Unreal Warfare Build* a *Unreal Tournament 2003*. Zajímavostí je, že se žádná sestavení neobjevují v *Unreal Tournament 2003* ikdyž s *Pure Unreal Engine* sdílí stejná čísla sestavení. Tato etapa trvala do roku 2004. Největším skokem byl *Unreal Engine 2.5*, ten kromě lepší komprese a podpory textur v rozlišení 4096x4096 přinesl možnost spolupráce s AI či podporů botů. Další novinkou byla podpora DirectX9 a OpenGL2.[20]



Obrázek 7. Cube mapping

Unreal Engine 3 byl představen v roce 2004 a jeho vývoj skončil o 5 let později, v roce 2009. Pro tuto verzi bylo zásadní představení Unreal development kitu, jehož SDK bylo volně dostupné na stránkách výrobce. Znamenalo to tedy, že si kdokoliv mohl tuto verzi zdarma stáhnout a začít vyvíjet vlastní hru. V případě, že by chtěl uživatel hru komerčně vydávat, musel by zaplatit. Tato verze UE byla založena na Direct3D 8 nebo 9 a na OpenGL 2 nebo 3. Přidána byla také podpora renderovací pipeline a grafické efekty pro cyklus dne a noci. Byla přidána řada efektů *postprocessingu* jako *motion blur*, *depth of field*. [20]

Na začátku roku 2012 byl představen Unreal Engine 4. Velkou změnou byl způsob monetizace Unreal development kitu, kdy se pouze platilo komerční využití určitým procentem z ceny, ta činila 5%, pokud hra překročila za dobu svoji existence hrubý příjem přesahující 1 milion dolarů. Tohoto využili hlavně malí vývojáři, kteří mohli vyvíjet herní tituly zadarmo. Díky tomuto kroku se na internetu začali objevovat amatérské videa demonstrující možnosti UE. Změna byla také u vydávání verzí, při každé aktualizaci se zvýšilo desetinné číslo za číslovkou 4. UE4 podporoval DirectX 11 a možnost vyrenderovat odrazy v celé scéně v HDR. [20] [27]

Nejnovějším Unreal Enginem je verze 5, ta ve verzi Preview vyšla 22. února roku 2022. Největší novinkou je nový systém, který usnadňuje práci s mikropolygonovou geometrií, *Lumen* jako nový systém na globální nasvícení se zaměřením na nové herní konzole. První dostupnou hrou na předvedení UE5 byl *The Matrix Awakens: An Unreal Engine 5 experience*. [23]

2.1.2 Epic Games Launcher

Pro využívání UE je nutné mít staženou aplikaci Epic Games Launcher a v ní vytvořený účet. Jedná se o platformu pro nákup digitálních her a aplikací, podobně jako Steam či Origin a je ke stažení zdarma. Zajímavým zpestřením pro uživatele Epic Games Launcheru oproti konkurenci je, že každý týden dává nějakou hru zadarmo. Může se jednat o indie hry nebo i Triple-A hry. Obdobně jako Steam či jiné platformy má uživatel knihovnu vlastní her, svůj seznam přání a přátele. Unreal Engine má poté svoji záložku přes kterou můžou uživatelé nahlédnout na video tutoriály nebo si rozkliknout marketplace a v něm získat například různé assety nebo modely, a to buď zadarmo, nebo jednorázový poplatek.

V roce 2019 začal Epic Games Launcher nabízet aplikaci Twinmotion. Jedná se o aplikaci, která je v nekomerčním použití zdarma a slouží k vytváření prostředí a panoramat.

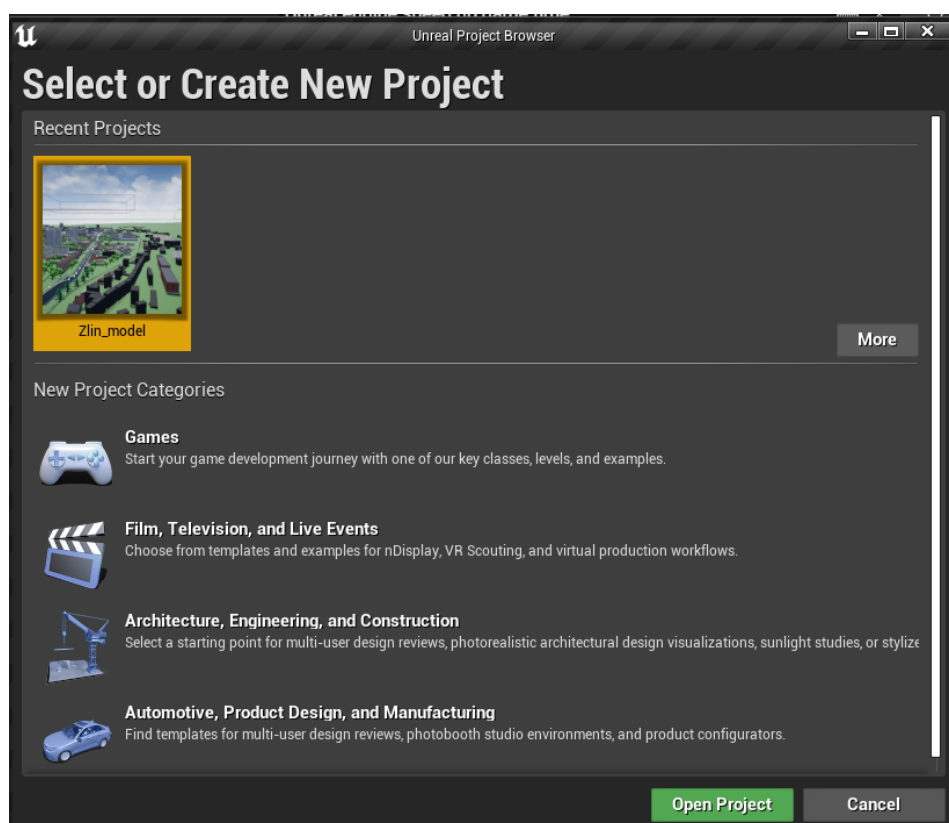
2.1.3 Popis uživatelského prostředí

2.1.3.1 Úvodní obrazovka

Při spuštění programu Unreal Engine se otevře úvodní obrazovka, na které jsou k dispozici poslední projekty a také přednastavené šablony pro vytvoření nových projektů. Při vybrání jedné z posledních položek se rovnou otevře hlavní okno programu s projek-

tem. Při přejetí myši přes poslední projekty se zobrazí okno, které ukáže podporované platformy tohoto projektu a jeho místo uložení na disku počítače. Při vytváření projektu se vybere jedna z šablon a poté následují další stránky s podrobnějším nastavením projektu a výběrem šablon.

Přednastavená šablona může například u projektu „*Games*“ být, jestli je hra z první osoby nebo ze třetí osoby, jestli bude zaměřená na VR nebo AR. Každá z těchto možností ovlivňuje načtení potřebných assetů a komponentů. Možností je také vytvoření prázdného projektu, tam se už očekává, že si uživatel dodá své vlastní komponenty a assety. Poté následuje stránka, kde si uživatel může nastavit větší podrobnosti jako je výběr mezi používáním *Blueprints* nebo C++, jestli uživatel chce, aby aplikace byla pouze stavěná na maximální kvalitu anebo aby se dala kvalita měnit od nejslabšího zařízení po to nejsilnější anebo jestli zařízení na, kterém projekt poběží, bude pro počítače nebo konzole anebo pro mobilní zařízení.

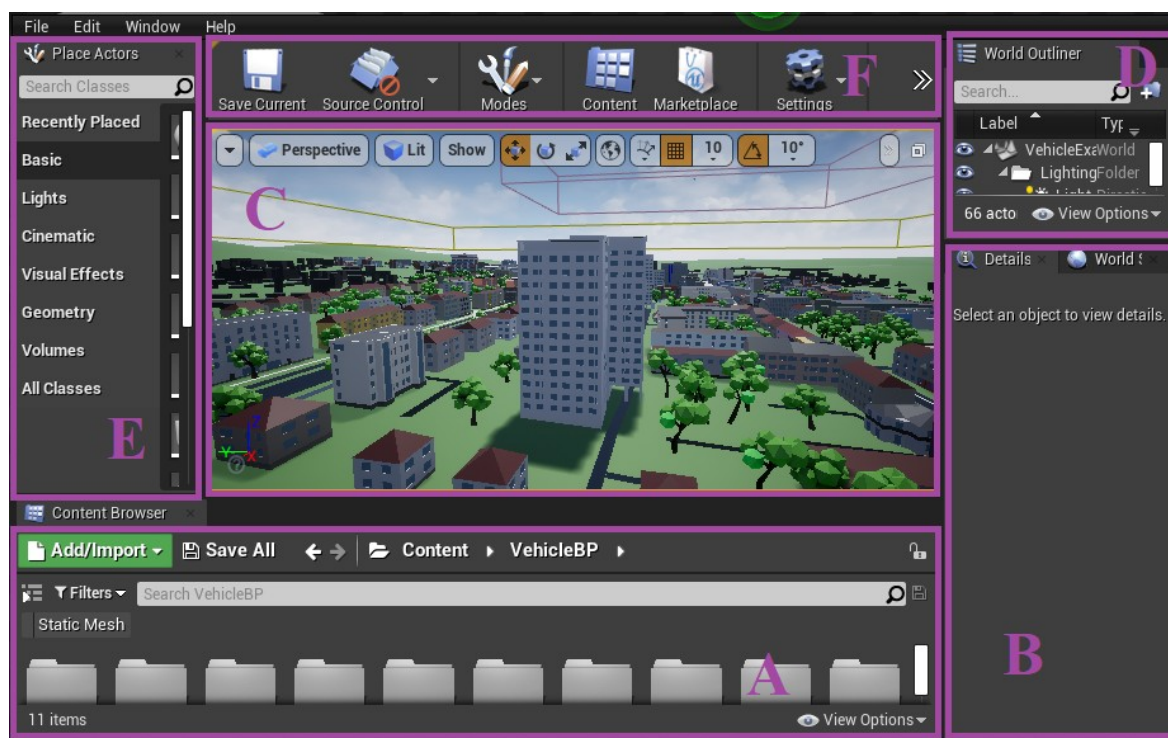


Obrázek 8. Úvodní obrazovka pro výběr šablony a projektu

2.1.3.2 Pracovní plocha projektu

Prostředí Unreal Engine 4 se dá rozdělit 6 částí, části jsou znázorněny na obrázku číslo 9. Každé písmeno odpovídá jedné sekci v aplikaci [28 – strana 55 až 65]

- A. Prohlížeč assetů – v tomto okně si můžeme prohlížet assety, které jsou spojeny s projektem
- B. Okno pro detaily objektu - v tomto okně se zobrazí detaily právě vybraného objektu
- C. Náhled projektu – interaktivní náhled v reálném čase
- D. Prohlížeč aktérů – zobrazuje aktéry, které jsou použity v projektu
- E. Knihovna dostupných aktérů – z této knihovny může přetáhnout a vložit aktéry do náhledu projektu
- F. Panel rychlých akcí – panel, který obsahuje nejčastěji používané akce, mezi ně například patří možnost spuštění aplikace formou náhledu nebo sestavení



Obrázek 9. Pracovní plocha programu Unreal Engine

2.1.4 Funkce

2.1.4.1 Import a export

Importovat do programu UE lze ze spousty 3D programů. Pro největší seznam podporovaných formátů je nutné doinstalování pluginu *datasmith*, který je zdarma. Bez problému zvládne průmyslově standardizované formáty jako je FBX, USD nebo Alembic. Asi nejzajímavější formát je USD, ten má výhodu v tom, že ho UE dokáže číst přímo ze souboru a není díky tomu nutný import. Díky tomu ušetříme čas, který by byl zabrán plnohodnotným importem. Unreal Engine také podporuje Python scriptování.

Unreal Engine také podporuje reimportování assetů z disku. Tato možnost se hodí, pokud máme object na požadované pozici a smazání a nový import by zabral moc času. Můžeme tedy provést změny například v Blenderu a pokud nezměníme název assetu a přepíšeme jej v adresáři počítače, tak si při reimportu Unreal Engine nahraje nový asset a jeho pozice zůstane zachována.

Uživateli může také ušetřit čas jednoduchý *drag-and-drop* assetů přímo do programu, bez potřeby používat kontextové menu a hledat soubor v počítači. Stačí pouze vzít podporovaný soubor a přetáhnout jej do prohlížeče assetů (obrázek 9. – písmeno A). [24]

2.1.4.2 Budování světa

Unreal Engine obsahuje nástroje pro tvorbu světů obrovských rozsahů. Za pomoci *Landscape* systému, zvládnete vytvořit hory, údolí ale i jeskyně. Můžete mít více výškových map a pracovat s každou nezávisle na druhé nebo nedestruktivně pracovat na terénu pomocí systému *Splines*. *Grass tool* dovoluje vyplnit venkovní lokace trávou, květinami, anebo kamením. *Procedural Foliage Tool* umožňuje simulovat růst lesů za několik let. *Water systém* umožňuje přidat realistické řeky a jezera a díky zabudované simulaci tekutin budou postavy a objekty realisticky reagovat na vodu. [24]

2.1.4.3 Editor

UE obsahuje základní editor objektů, slouží primárně k malým úpravám bez nutnosti si objekt upravovat v původním programu a poté jej znovu importovat. [24]

2.1.4.4 Animace

V UE můžete tvořit animace pomocí *Animation Blueprints*. Jedná se o speciální blueprint, který ovládá animaci *mesh* kostry. Grafy se upravují přímo v *Animation Blueprint editoru*, kde můžeme provádět směřování animací, přímo ovládat části kostry nebo nastavit logiku, která bude definovat finální animaci pro kostru. UE umožňuje také pracovat s animacemi z motion capture pomocí *Take Recorder*. [24]

2.1.4.5 Světlo a vykreslování

Unreal Engine umožňuje vytvářet realistické nasvícení vnitřních a venkovních lokací v reálném čase za použití široké škály nástrojů, mezi tyto nástroje patří atmosférické efekty slunce a oblohy, volumetrická mlha a mraky nebo volumetrické světelné mapy.

K dispozici je také podpora sledování paprsků v reálném čase, pomocí kterých lze vytvářet světelné efekty na úrovni Hollywoodu. [24]

2.1.4.6 Materiály

Pomocí *Material editoru* zvládnete ovládat vzhled a působení postav a objektů, můžete vrstvit materiály a na pixelovou přesnost nastavit veškeré hodnoty a tím dosáhnout cíleného vzhledu. UE umožňuje rozdělení velmi velkých textur na menší díly a načítat pouze viditelné části. [24]

2.1.4.7 Simulace

Pro tvorbu simulací je v UE k dispozici systém *Niagara*, který umožňuje v reálném čase tvořit efekty pro oheň, kouř, prach anebo vodu s kompletní možností nastavení. Své scény můžeme oživit o světelné efekty, vytvářet komplexní pohyby světelných částic pomocí vektorových polí anebo nechat částice reagovat na hudbu. [24]

2.1.4.8 Herní prvky

K práci na uživatelském rozhraní máme k dispozici *Unreal Motion Graphics UI designer*, pomocí něj můžeme vytvářet menu našich aplikací, různé HUDy nebo libovolné prvky uživatelského rozhraní. Pro usnadnění práce zde existují *Blueprint Widgets*, které mají předpřipravené funkce jako tlačítka, posuvníky nebo různé indikátory.

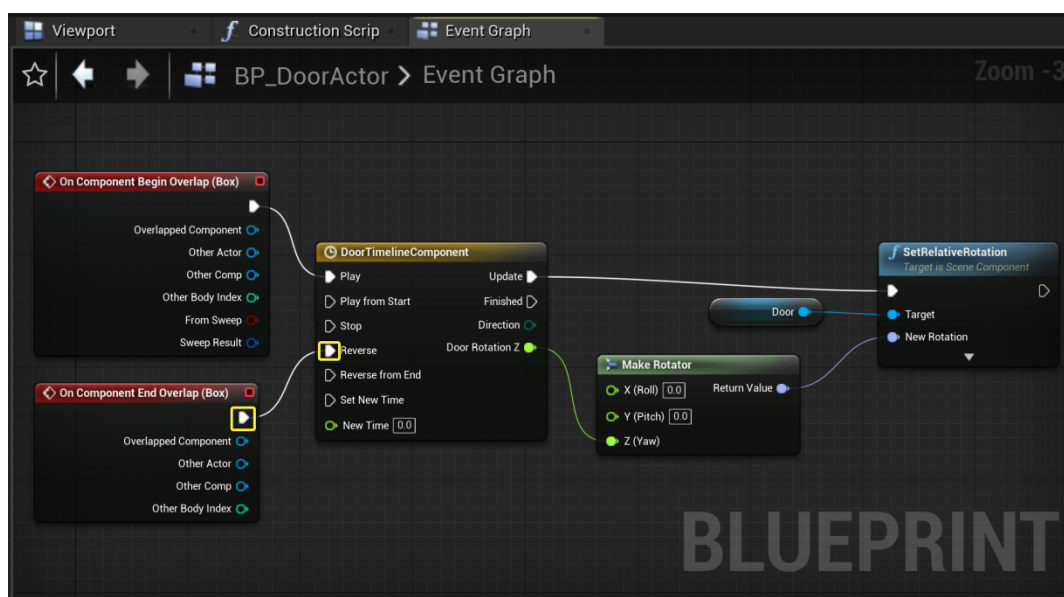
UE disponuje pokročilých systémem umělé inteligence, vaše vytvořené postavy díky tomuto systému mají povědomí o sobě a svém okolí. Systém se dá použít buď formou *Blueprintů* nebo pomocí *Behaviour Tree* (stromu chování). [24]

2.1.4.9 Platformy

Unreal Engine podporuje všechny hlavní počítačové platformy, výstup bude fungovat jak na platformách Windows a MacOS tak i na platformě Linux. Podpora nekončí jen u počítačů, UE podporuje vlajkové konzole nové generace jako je Xbox Series X a Playstation 5. Podporu mají i starší konzole z řad Playstation 4 nebo Xbox One ale i Nintendo Switch. Ochuzení o podporu nejsou také mobilní zařízení s operačním systémem iOS a Android nebo také zařízení pro virtuální realitu jako je například Oculus Quest nebo HoloLens 2. [24]

2.1.4.10 Blueprints

Jedná se o interní skriptovací jazyk uvnitř editoru, který využívá Unreal Engine. Je založen na konceptu využívající uzly a díky tomu nevyžaduje žádné programovací znalosti, jelikož se uživatel nemusí setkat s kódem a může využívat části, které byly bez *Blueprints* dostupné pouze programátorům. Vše funguje na základě spojování uzlů, událostí a proměnných pomocí *wire*. [25]



Obrázek 10. Ukázka Blueprintu v prostředí Unreal Engine

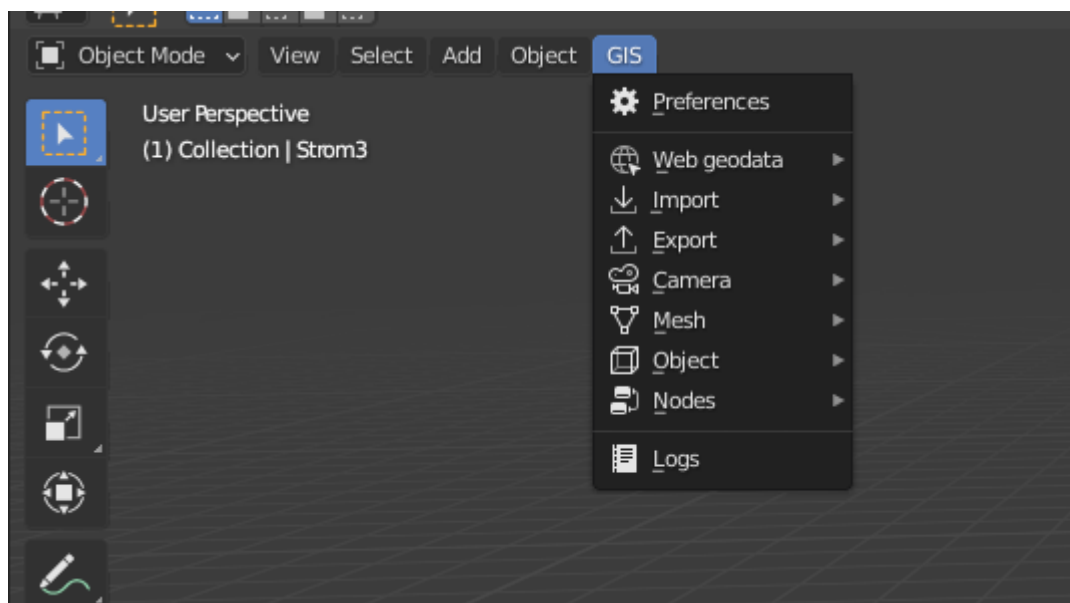
II. PRAKTICKÁ ČÁST

3 VYGENEROVÁNÍ MODELU Z BLENDERGIS

Po nainstalování a nakonfigurování rozšíření BlenderGIS si vytvoříme nový Blender soubor typu *General*. Jelikož si BlenderGIS při práci vytváří svoje podpůrné soubory po boku .blend souboru, je velice vhodné mít vytvořenou složku, do které se budou ukládat veškeré soubory spojené s projektem. Bez vytvoření a uložení do složky hrozí přepsání podpůrných souborů, jelikož je BlenderGIS pojmenovává stejným názvem, který nerespektuje název .blend souboru.

3.1 Basemapu

Abychom si mohli nechat vygenerovat budovy, potřebujeme prvně vytvořit *Basemapu*, bez této části nelze vygenerovat samotné modely budov. To uděláme tak, že v kontextovém menu v hlavičce aplikace vybereme nově přidanou možnost díky addonu a to *GIS*. Při rozkliknutí se nám otevře menu možností a vybereme položku *Web geodata*, tady vybereme již zmiňovanou *Basemapu*. Po kliknutí na položku *Basemap* nám vyskočí okno (obrázek 3). Tady si vybereme možnost zdroje jako Google a vrstvu mapy vybereme jako satelitní. Po potvrzení se nám zobrazí v Blenderu mapa světa. Poté si můžeme manuálně najít Zlín a ten vybrat nebo použít klávesové zkratky jak jsem učinil já. Prvně zmáčkneme klávesu G, po zmáčknutí se nám zobrazí okno a do textového pole zadáme „Zlín“ a jako zoomlevel, neboli hodnotu jak vzdálený pohled z vrhu bude na mapu (0 značí nejvyšší vzdálenost a hodnota 25 zase nejnižší). Pro potřeby práce jsem zvolil hodnotu 15, díky této hodnotě pokryji celý Zlín, hodnota o jedno vyšší již zasahuje do Otrokovic a pojme více okolních vesnic. Výběr mapy je *WYSIWYG* a proto dokud nejsem spokojený s výslednou mapou, kterou vidím ve *Viewportu*, si na mapě můžu pomocí levého tlačítka, anebo stlačením kolečka u myši pohnout a tím změnit výsledný obraz mapy. Výška lze také měnit pomocí točení kolečka na myši nahoru nebo dolů. Jakmile jsem spokojený tak zmáčknou klávesu E a výběr se přenesení na *Plane* objekt a ten se vloží do středu *viewportu*. V současném stavu ale chybí výšková mapa, kterou musíme vygenerovat zvlášť. Půjdeme tedy do *GIS* → *Web geodata* a vybereme možnost *Get elevation*, zobrazí se nám nové okno (obrázek 5.) a zde vybereme prostřední možnost a potvrdíme. Blender začne generovat výškový terén a při tomto procesu může aplikace působit jako by se sekla a na krátkou chvíli si může objevit „Neodpovídá“.

Obrázek 11. Rozbalené menu *BlenderGIS*

3.2 OSM

Po vytvoření Basemapy a degenerování výškové mapy jsem přidal budovy. V kontextovém menu vybereme GIS → Web geodata → Get OSM. Blender nám otevře okno, zde jsem nechal výchozí nastavené hodnoty pro *Ways* a *Building extrusion* a k tomu přidal navíc položku *Buildings*, která vygeneruje samotné budovy. Zaškrtnul jsem také *Elevation from object*, která zaručí, že se budou budovy generovat dle výškové mapy a poté poslední možností je *Seperate objects*. Pomocí této možnosti se každá budova vygeneruje, jako separátní objekt což mi poté ulehčí samotou práci na detailech budovy.

Proces generování OSM je velmi časově náročný, v mém případě to zabralo 3 hodiny a 44 minut. Po celou dobu je aplikace Blender ve stavu „neodpovídá“ a nejlepší volba je to tedy nechat dělat v době, kdy počítač nepotřebujeme. Využití procesoru v tomto stavu pro moji sestavu s čtyřjádrovým procesorem Intel Core i5 - 6500 nepřesáhlo 30%, takže samotný proces není náročný na hardware. Po vygenerování jsem měl k dispozici kompletní model Zlína, který obsahoval 19 477 objektů.

Tabulka 2. Hodnoty po vygenerování

Objekty	19 477
Vertexy	246 307
Faces	144 612

Z důvodu dříve zmiňované časové náročnosti jsem se rozhodl vygenerovat silnice až po vygenerování budov. Postup byl stejný jako pro generování budov, akorát místo *buildings* se zaškrtnula možnost *highway*. BlenderGIS nevytvořil silnice, jako plane objekt ale pouze jako vodící čáru, kudy silnice vede. Jako součást silnic byly i vodící čáry po chodnících, které do sebe zakomponovali i místa pro přechody pro chodce. Ty byly vidět jako propojení vodících čar silnic a chodníků přes vertex. Výhoda vygenerování silnic tímto způsobem je taková, že nemusíme využívat Google mapy nebo samotnou mapu v Blenderu a podle ní tvořit silnice. Ubude nám také starost o to, aby silnice respektovala výškové rozdíly *Basemapy*.

Budovy *BlenderGIS* vygeneruje jako jednoduché bezbarevné objekty, které respektují geometrii budov. Po osách *X* a *Y* byly přesné, ale některých modelů pokulhávala výška samotného objektu a objekt jsem musel manuálně natáhnout po ose *Z*.



Obrázek 12. Vygenerované modely budov

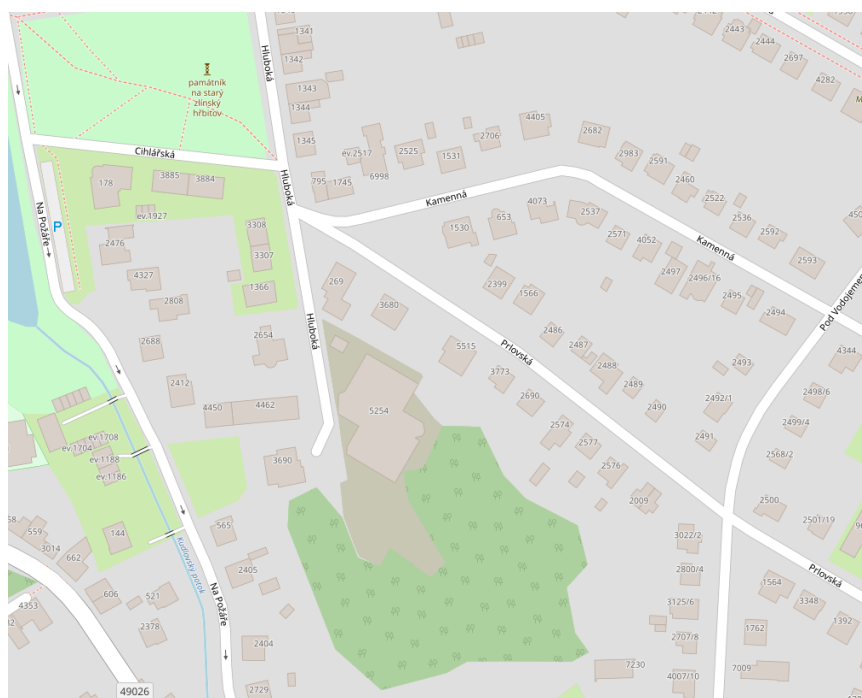
Některé budovy byly složeny z více vygenerovaných objektů, například na obrázku 11 to jsou Baťovské domky. Ty byly vygenerovány jako dvě části a pro lepší možnosti úpravy jsem je musel spojit dohromady. To proběhlo tak, že jsem vybral obě části objektu a zmáčkne klávesovou zkratku *Ctrl + J*. Ta docílila toho, že se oba objekty spojily do jednoho. Poté jsem v *editačním režimu* umazal stěny, které byly proti sobě.

4 DODĚLÁNÍ CHYBĚJÍCÍCH ČÁSTÍ MODELU V BLENDERU

Aktuálnost budov ovlivňují data, která mají v sobě nahrané servery od *Openstreetmap*, ze kterých rozšíření *BlenderGIS* získává svá data. Proto se může stát, že některé změny nebyly aktualizované na straně *OSM* a proto je modul nevygeneruje.

Následující obrázky 12. a 13. demonstrují porovnání stejné lokace z databáze *Openstreetmap* v porovnání s generovaným modelem pomocí *BlenderGIS*. Z obrázku lze vidět, jakým způsobem objekty generuje. Podle hran budov z mapy vygeneruje objekt, který se těmito hranami řídí. Lze také vidět, že zvládá vytvořit i cesty vedoucí k domům tak některé části chodníků v parku, který můžeme vidět v levém horním rohu obrázku 12.

V této kapitole byly také využity již existující modely budov [30], jednalo se přesněji o 213 hotových objektů v okolí centra města Zlín.



Obrázek 13. Mapa z OSM



Obrázek 14. Vygenerovaná část pomocí rozšíření

4.1 Přidání referenčního obrázku

Pro vytváření detailů budov jsem si vytvořil nový *.blend* soubor. Tímto způsobem jsem zabránil chybám a také jsem měl vždy původní objekt nedotčený a bez změn v původním souboru. Další výhodou pracování ve vedlejším souboru byla plynulost prostředí. Při snaze upravovat model přímo v souboru s vygenerovaným Zlínem jsem se setkal se sníženou snímkovací frekvencí celého *Viewportu* a také nespočtem záseků.

Pro ulehčení nahrávání referenčních obrázku do projektu v Blenderu jsem si nainstaloval rozšíření „*Import Image As Plane*“. Toto rozšíření nahraje obrázek do projektu, jako *Plane object* ve správném poměru stran vůči obrázku. Pro nahrání poté použiji stejnou zkratku jako pro přidání nového objektu, tedy *Shift + A* a vyberu položku *Image* a v ní položku *Image as Plane*. Na 3D kurzoru se mi poté vytvoří *Plane* s nastaveným obrázkem. Druhým způsobem jak obrázek dostat do Blenderu pomocí tohoto rozšíření je v kontextovém menu aplikace vybrat položku *File* → *Import* a v ní položku *Image as Plane*.

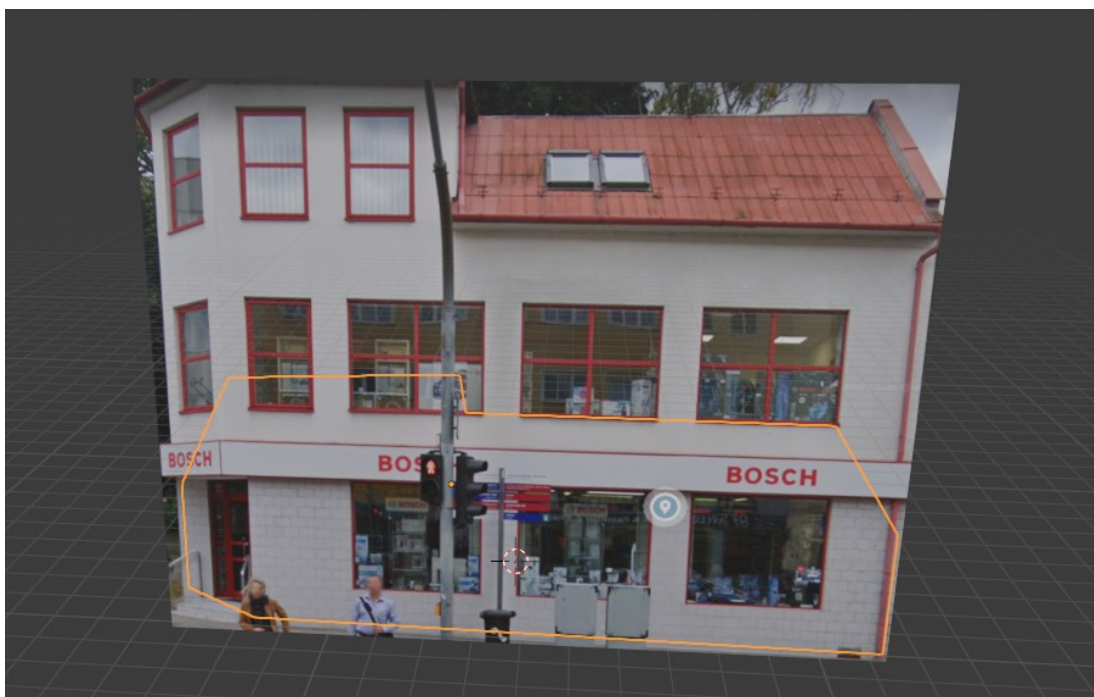
Obrázky pro toto použití získám při použití *Google street view* a zabudovaného nástroje na *výstřižky* v operačním systému Windows. Při nalezení požadovaného objektu si pomocí zkratky *Windows + Shift + S* vytvořím výstřižek a ten uložím do složky, ve které ukládám referenční obrázky.

4.2 Tvorba detailu budovy

Mezi nejčastější nástroje pro tuto část patří nástroj *Loopcut*. Pomocí možnosti vytvoření více *Loopcutů* za použití kolečka myši mi ubyla starost vytvářet všechny *Loopcuty* jeden po druhém. V následující části demonstruji postup úprav budov na jedné budově. Tento postup nebo jemu podobný se opakoval pro každou budovu, která dostala materiál. Některé úpravy byly prováděny v odděleném souboru, jiné, méně náročné budovy jako byly například garáže se upravovaly přímo v souboru, kde byl výsledný model.

4.2.1 Základní tvar budovy

Ze souboru ve kterém mám vygenerovaný model pomocí *Ctrl-C* zkopíruji požadovanou budovu a v novém souboru ji pomocí *Ctrl-V* vložím. Poté si přidám *Plane* s referenčním obrázkem a pomocí klávesy *S* jej roztáhnu tak aby referenční obrázek pokryl šířku budovy, jelikož v několika případech BlenderGIS ji nesprávně vygeneroval a tudíž ji nemůžeme využít. Pro přesný pohled zmáčkne na numerické části klávesnice číslovku *1*. Takto si uzamkneme *pohled po ose X* což nám ulehčí nanášení obrázku na šířku budovy.



Obrázek 15. Nanesení obrázku na budovu

Po úspěšném nanesení si můžeme upravit výšku modelu tak aby byla totožná jako na referenčním obrázku. Přepneme se do editačního režimu a pomocí *Face select modu* vybereme vrchní část modelu. Zamkneme se pomocí klávesy *1* na osu *X* a zmáčkne klávesu *E*. Pomocí nástroje *Extrude* vytáhneme horní část modelu do úrovně střechy. Poté se pře-

pneme do *Edge select modu*, poté klikneme levým tlačítkem na původní *Edge* a u toho držíme *Shift*, takto ho vybereme celý naráz a nemusíme klikat všechny jeho části zvlášť. Poté zmáčkne klávesu *X* a vybereme možnost *Dissolve edge*. Takto původní *Edge* smažeme. Pomocí *Loopcutu* na výšku a *Extrude* vytvoříme i 3. patro budovy (obrázek 12.).



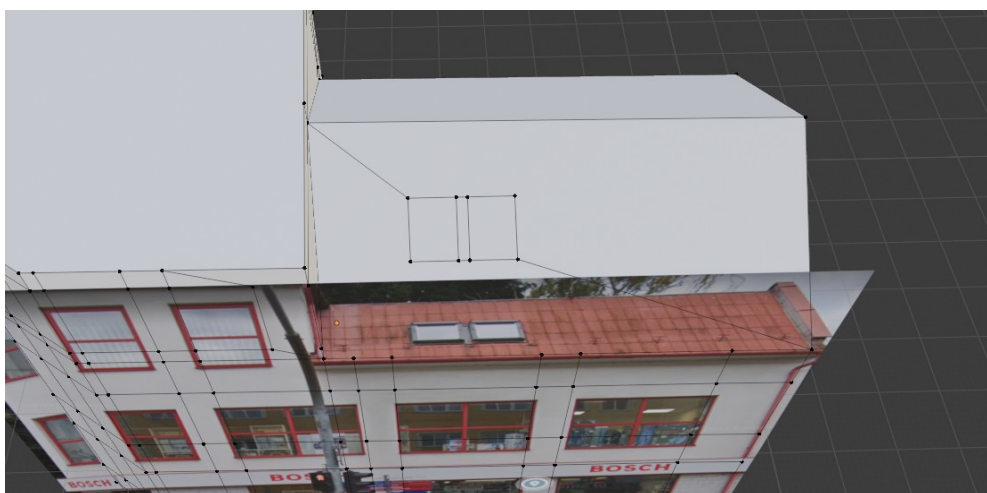
Obrázek 16. Úprava horní části modelu

4.2.2 Vytvoření vodících bodů na budově

Pro vytvoření oken si musím vytvořit referenční body, pro tyto body použiji také stejný obrázek jako pro nastavení výšky budovy. Pro okna využiji kombinaci nástrojů *Knife* a *Loopcut*. Na vytvoření pater použiji nástroj *Loopcut* a jelikož má budova 2 patra tak vytvořím 4 horizontální řezy a potvrdím. Poté jednotlivé řezy pomocí dvojitého zmáčknutí klávesy G pohybuji tak aby šli přes hrany oken. Stejný postup poté zvolím pro vytvoření vertikálních řezů. Pro vytvoření střešních oken využiji nástroje *Knife* a okna vyřežu.



Obrázek 17. Referenční body



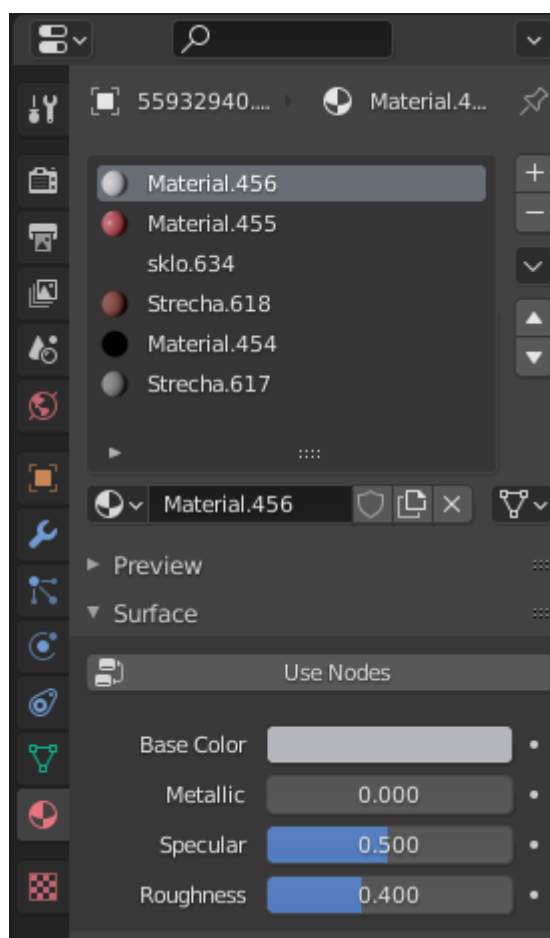
Obrázek 18. Střešní okna

4.2.3 Materiály

Jakmile mám vytvořené referenční body na budově, začnu aplikovat materiály. Pro získání potřebné barvy využiji *Kapátko*, které je dostupné v aplikaci Blender. Kapátkem kliknu na potřebnou část v referenčním obrázku.

Kapátko bylo využito z důvodu jeho jednoduchosti a také, protože jsem měl již zmiňovaný referenční obrázek nahraný v programu. Tímto způsobem jsem mohl vybrat barvu objektu přímo v aplikaci a nemusel pro to pouštět externí programy nebo kapátka. Výsledná barva nemusela být vždy přesná, proto bylo nutné ji upravit. Nejvíce problému bylo se světlostí samotného materiálu. Kapátko z Blenderu udělalo u šedých barev výslednou barvu vždy o trochu tmavější.

Po získání všech potřebných barev aplikuji hlavní materiál celé budovy, na obrázku to je materiál s šedým odstínem. Poté již pomocí *Face select modu* vybírám části, jako jsou okna nebo střecha a aplikuji na ně požadovaný materiál. V případě oken využiji *Extrude* aby byly zapuštěné.



Obrázek 19. Kápátko v Blenderu

4.2.4 Výsledná budova

Po dokončení aplikace všech materiálů mám vytvořenou kompletní budovu. Tuto budovu poté již překopíruji zpět do hlavního a nahradím tak vygenerovanou budovu. Referenční obrázek poté smažu.

Budova se vložila do středu modelu, takže jsem ji musel vždy dostat do požadované lokace. Abych budovu dostal na požadovanou lokaci, musel jsem si ji prvně vybrat. Budově jsem nastavil střed na její geometrický střed. To jsem udělal tak, že na vybranou budovu jsem klikl pravým tlačítkem myši a vybral možnost *Set Origin* → *Origin to geometry*. Jakmile byla tato část hotová tak jsem pomocí pravého tlačítka myši s držením klávesy *Shift* umístil kurzor na požadovanou lokaci. Poté jsem pomocí kombinace kláves *Shift* + *S* vybral možnost *Selection to cursor*. Takto se mi povedlo dostat objekt do požadované lokace. Poté jsem musel akorát zajistit, aby nebyl ve vzduchu nebo naopak nebyl v zemi. Toho jsem docílil tak, že jsem vybral objekt a zmáčkl klávesu *G*. Po zmáčknutí jsem zmáčkl klávesu *X* pro zamknutí pohybu pouze na osu *X* a s držením klávesy *Shift* jsem objektem pohyboval. Klávesa *Shift* zajistila, že byly pohyby menší a pomalejší. Takto se pozice lépe trefovala.



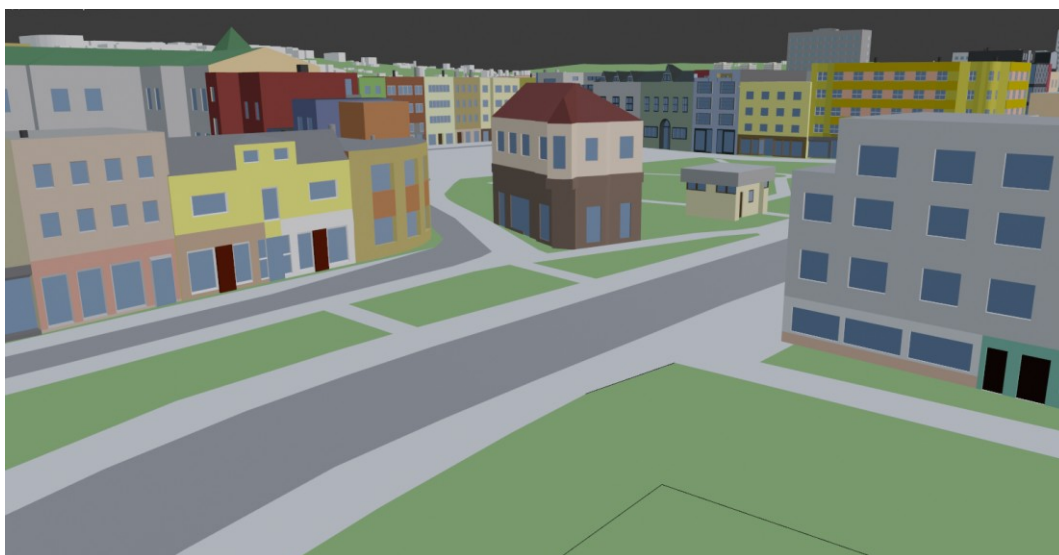
Obrázek 20. Výsledná budova

4.3 Tvorba silnic a chodníků

Pro vytvoření silnic a chodníků jsem využil vygenerované vodící čáry od rozšíření BlenderGIS. Tyto čáry jsem pomocí klávesy *E Extrudoval* a posouval tak aby utvořili cestu, respektive chodník. Různé segmenty jsem poté na sebe napojoval, kdy jsem ve *Vertex modu* pomocí klávesy *F* spojil 2 vertexy a pomocí *Edge select modu* vybral kraje a pomocí klávesy *F* ji vyplnil materiálem. Pro nastavení tloušťky a také zamezení problému s normálami byl na model silnic a chodníků aplikován modifikátor *Solidify*. Ten byl ponechán na výchozí hodnotě.

Kruhový objezd u autobusového nádraží byl tvořen pomoc nového kruhu, který byl spojen poté s vygenerovaným kruhem. Pomocí klávesové zkratky *Shift + A* jsem vytvořil nový kruh, ten pomocí klávesy *S* na změnu velikosti přizpůsobil velikosti vygenerovaného kruhu. Poté jsem oba kruhy vybral a pomocí *Edge select modu* vybral jeho části a pomocí klávesy *F* je spojil.

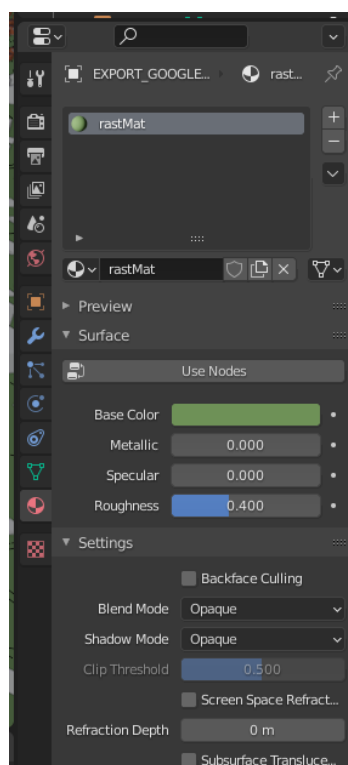
Silnice a chodníky měli poté nastaveny rozdílný materiál, aby šli od sebe rozeznat, v tomto případě jsem použil tmavý odstín šedé pro silnice a chodníky jsou poté světlejším odstínem té stejné barvy.



Obrázek 21. Chodníky a silnice v modelu v Blenderu

4.4 Změna textury mapy na materiál

BlenderGIS vytvořil podkladovou mapu jako texturu, tu budu muset smazat a také budu muset převést vygenerovanou mapu na *Mesh*. V okně pro nastavení vlastností objektu (obrázek 1. písmeno G) vyberu druhou možnost od spodu s názvem *Material properties*. Původní materiál s nastavenou texturou smažu stiskem tlačítka s mínusem. Pro tuto operaci nesmím být v *editačním režimu* jinak *Blender* vyhodí chybovou hlášku. Po smazání zmáčknou naopak klávesu +, pod oknem ve, kterém se zobrazí materiál, vyberu nové tlačítko *New* a poté kliknu na modré tlačítko *Use nodes*. Poté v položce *Base color* vyberu zelenou barvu, pro kterou použiji formou HEX čísla 6E9157. Hodnotu *Specular* snížím na 0.

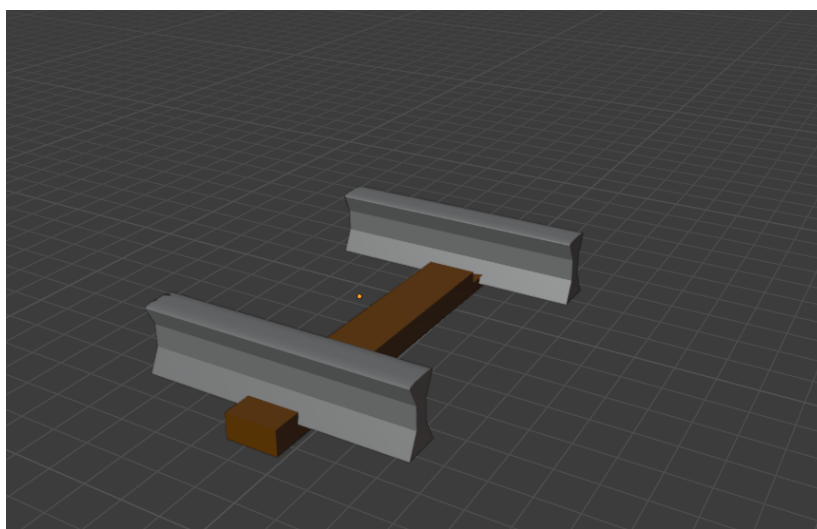


Obrázek 22. Nastavení barvy pro terénu

4.5 Tvorba koleji

Koleje v modelu byly vytvořeny pomocí modifikátorů *Array* a *Curve*, prvně se v novém souboru vytvoří krátký úsek koleje, na kolej se poté nanese materiál imitující kolejnice a pražce. V souboru se samotným modelem Zlína se poté vytvořila nová křivka, na kterou bude železnice nanášena. Pro tvorbu modelu byly využity dva *Cube* objekty, ty byly poté pomocí dvou *řezů* vertikálně rozděleny a vybrané řezy jsem poté pomocí klávesy *S* zmenšil, tím jsem docílil tvaru koleje. Na horní čas koleje jsem poté použil nástroj *Bevel*,

pomocí kterého jsem vytvořil jemné zakulacení. Pražec byl vytvořen, jako plane objekt, který byl vložen pod kolej. Poté byl pomocí *Extrude* roztažen na potřebnou výšku a pomocí nástroje pro *Loopcuty* byl rozpůlen horizontálně a vertikální řezy byly využity pro vyznačení částí, do kterých bude kolej vložena. Poté pomocí *Face select modu* byly části vybrány a smazány. Nakonec se vybraly koleje a byly posunuty tak aby zapadly do vytvořeného místa. Následně se objekt pomocí *Ctrl + J*, byly objekty spojeny do jednoho. Poté následovalo přetažení modelu koleje do hlavního modelu.



Obrázek 23. Model koleje

Pomocí klávesové zkratky *Shift + A* se v položce *Curve* vybrala možnost *Bezier*. Po přepnutí do *editačního režimu* si vyberu jeden konec a pomocí klávesy *7* si uzamknu pohled z vrchu. Poté pomocí držení klávesy *Ctrl* a klikání *pravým tlačítkem* myši budu extrudovat křivku do požadované délky. Po vytvoření křivky se přepnu do *Objektového modu* a vyberu si model koleje. V nastavení modelu vyberu položku pro modifikátory s názvem *Modifier properties*. V nastavení poté vyberu modifikátor *Array*, změním položku *Fit type* na *Fit Curve* a poté si v položce *Curve* vyberu vytvořenou křivku pro kolejiště. Druhý modifikátor bude *Curve*, ten zaručí, že bude kolej respektovat samotnou křivku, v tomto modifikátoru pouze vyberu křivku v položce *Curve object*.

Pro správnou funkčnost je nutné zachovat správné pořadí, kdy modifikátor *Array* bude vždy nad modifikátorem *Curve*.

Před dokončením práce na kolejišti se musí zkontrolovat, jestli se kolej v žádném místě nevznáší nad zemí nebo naopak není pod zemí. Pokud tato situace nastane, vyberu

křivku a přepnu se do *editačního režimu*. V editačním režimu vyberu *vertex*, u kterého je kolej ve špatné pozici a pomocí zmáčknutí klávesy *G* jej posunu.

Nakonec poté převedu samotnou kolej na mesh. V objektovém režimu vyberu kolej a v menu, které vyvolám kliknutím pravého tlačítka myši, vyberu možnost *Convert to → Mesh*. Nakonec můžu křivku smazat.



Obrázek 24. Hotová kolej v modelu

4.6 Koryto řeky

Pro vymodelování řeky bude využito *Sculpting* nástrojů, které jsou dostupné v aplikaci Blender. Pro správnou funkčnost musíme na terénu modelu ve směru řeky přidat více vertexů. To provedeme tak, že si vybereme terén modelu a přepneme se do *editačního režimu*. Poté ve *Face select modu* vybereme části ve směru řeky a v menu, které vyvoláme pravým tlačítkem myši stiskneme možnost *Subdivide*, tu použijeme ještě 5x.

Poté se přepneme v panelu záložek do *Sculpting* režimu a v panelu nástrojů vybereme si nástroj s názvem *Crease*. Pomocí tohoto nástroje vytvoříme základní tvar koryta řeky a poté pomocí nástroje *Flatten* zarovnáme dno.

Vodu vytvoříme tak, že do modelu přidáme nový *Plane*, ten roztáhneme do velikosti koryta řeky aby pokryl veškerou část a poté jej posuneme pod terén. Bude tedy pouze viditelný v korytu. Nakonec aplikujeme modifikátor *Boolean*, necháme výchozí možnost *Dif-*

ference a poté v položce *Object* vybereme terén. Výsledkem bude to, že plane, bude pouze v korytu řeky. Poté akorát aplikujeme materiál imitující vodu a převedeme *Plane* na *Mesh*, tím se zbavíme přebytečné části a nebudeme muset spoléhat na modifikátor.



Obrázek 25. Dřevnice

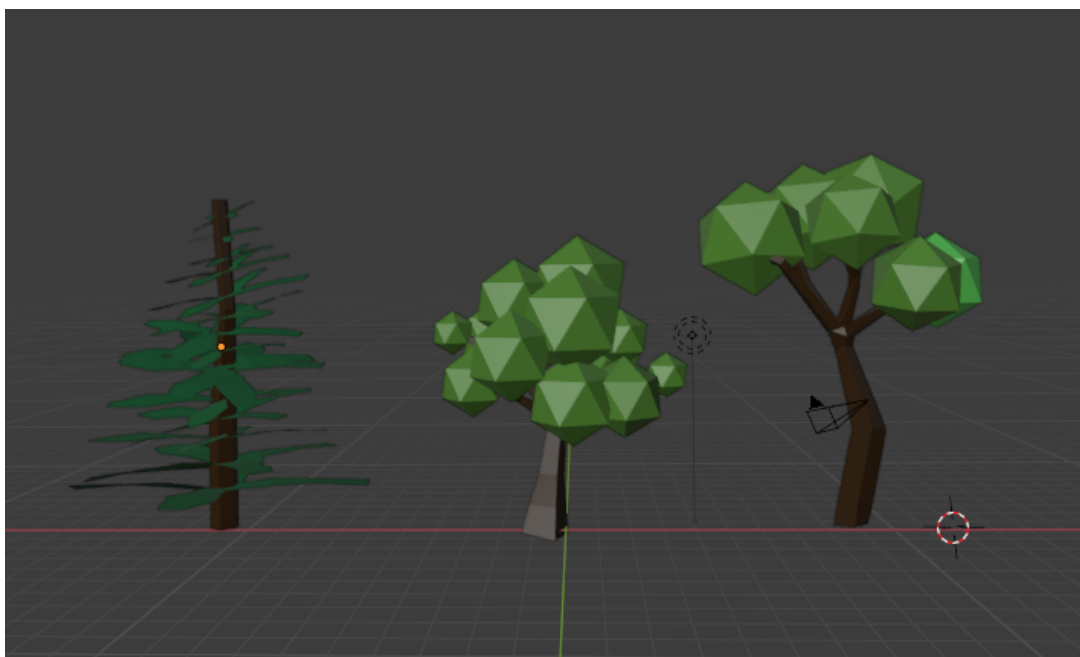
4.7 Stromy

BlenderGIS nevygeneruje stromy a proto je budeme muset vymodelovat zvlášť, jejich implementaci do modelu ale uděláme až v aplikaci Unreal Engine, jelikož má Unreal Engine přímo nástroj na tuto činnost. Pro vytvoření stromu budeme využívat modifikátoru *Skin*.

Pro modelování si vytvoříme nový soubor *.blend*, ve kterém vytvoříme 3 různé stromy. Při exportování bude mít každý strom svůj *.fbx* soubor. No nového souboru si vložíme *Cube* a přepneme se do *editačního režimu*. V editačním režimu poté vybereme *vertex select mode*, pomocí klávesy *A* vybereme všechny *vertexy* a stiskem klávesy *M*, to vyvolá *Merge* a vybereme možnost *At center*. Nyní máme z kostky pouze jeden *vertex*, ten si poté vybereme a pomocí extrudování z něj budeme tvořit další body, budeme se snažit napodobovat tvar stromu. Aby byl strom objekt a nikoliv pouze pár spojených vertexů, použijeme modifikátor *Skin*. Nyní model připomíná strom a provedeme zmenšení větví oproti kmenu.

To provedeme tak, že si zapneme *X-RAY mode*, ten zapneme klávesovou zkratkou *Alt + Z*. Po zapnutí si vybereme horní částí modelu, kde budou větve a zmáčkneme *Ctrl + A*, čím budeme zmenšovat větve modelu. Stejný postup poté použijeme na kmen stromu, ten naopak zvětšíme. Jakmile náš objekt připomíná kostru stromu, v modifikátoru *Skin* vybereme možnost *Apply*. Tento krok je nevratný a poté již nebudeme moc zmenšovat nebo zvětšovat části stromu.

Pro listy přidáme do scény nový objekt, tentokrát to bude *Ico sphere* a nastavíme počet *Subdivisions* na 1, tento objekt poté posuneme na konec větve, Poté zmáčkneme klávesovou zkratkou *Shift + D* a duplikujeme objekt a posuneme jej na další větev. V průběhu budeme poté náhodně měnit velikost pomocí klávesy *S* a rotaci pomocí klávesy *R*, aby listy nebyly vždy stejně orientované. Po dokončení přidávání listů aplikujeme materiály na části, pro kmen použijeme tmavší odstín hnědé barvy a pro listů světle zelenou. Nakonec všechny části spojíme do jednoho objektu pomocí klávesové zkratky *Shift + J*.

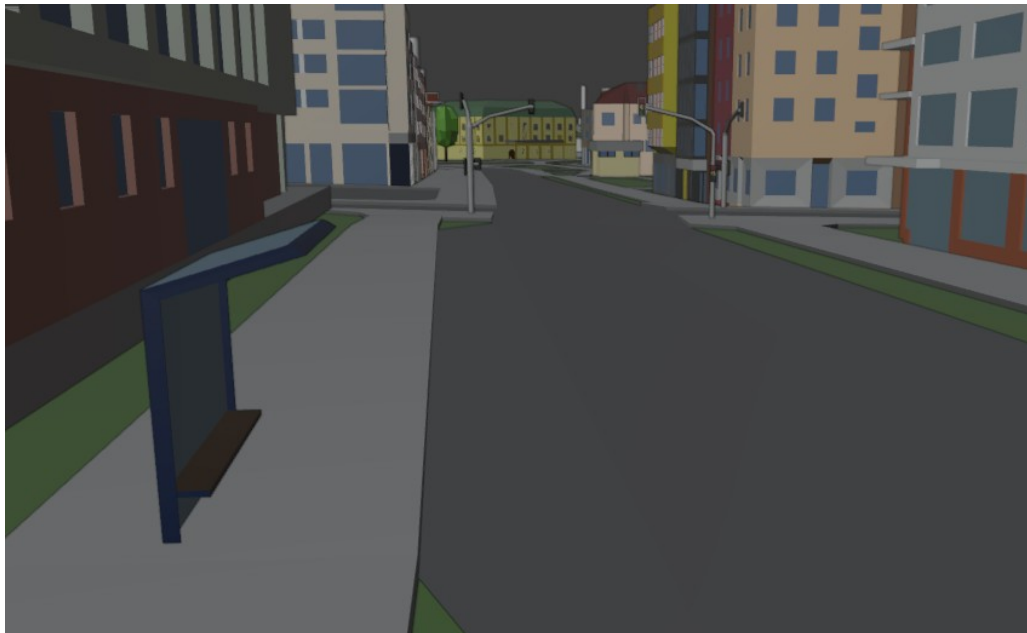


Obrázek 26. Modely stromů v prostředí Blenderu

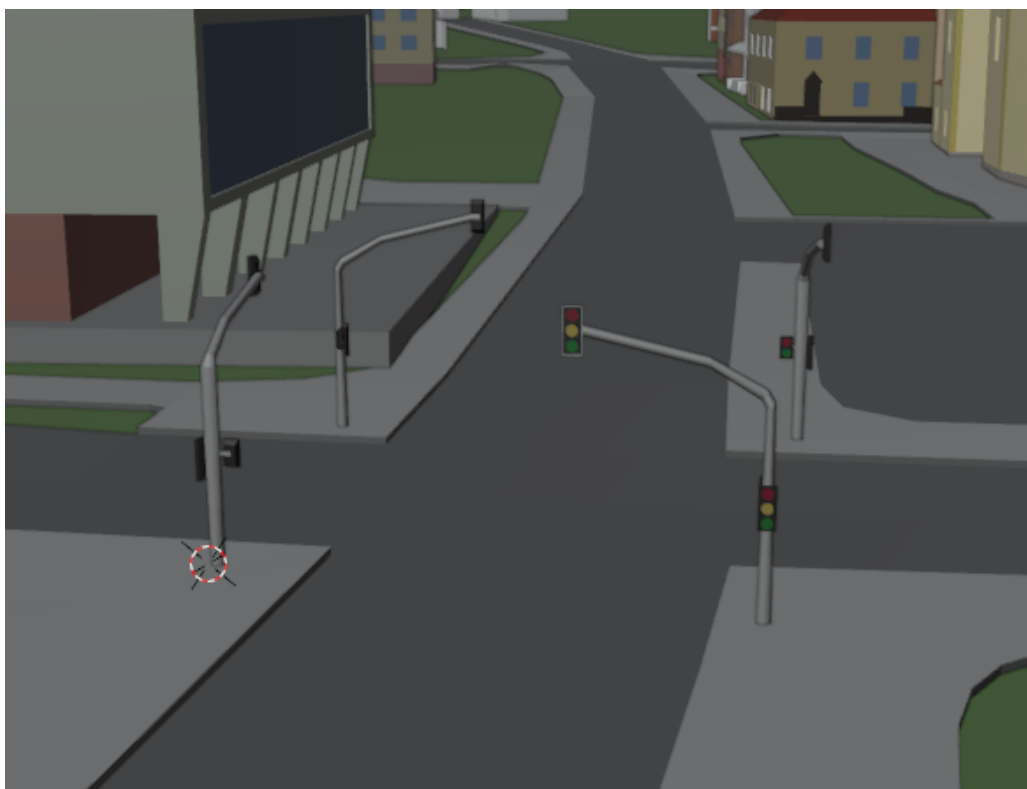
4.8 Semaforey a zastávky pro MHD

Do modelu byly přidány jednoduché modely zastávek pro městskou hromadnou dopravu a na světelných křižovatkách byly přidány semaforey, pro semafor se prvně vytvořil jednoduchý model sloupu, ten byl vytvořen pomocí objektu *Cylinder*, ten byl poté pomocí *Extrudování* uzpůsoben tvaru sloupu a poté byla použita *Cube*, ze které se vymodeloval

samotný semafor. Ten byl poté duplikován a bylo mu přidáno bílé ohrazení a byl použit na horní části sloupu. Na vytvoření semaforu pro chodce byla poté uříznuta spodní část a chybějící části byli vyplněni pomocí klávesy *F*.



Obrázek 27. Model autobusové zastávky



Obrázek 28. Model semaforu

5 EXPORT DO UE A OPRAVY CHYB MODELU

5.1 Oprava normálové mapy

Před exportem do Unreal Engine musela proběhnout oprava normálových map, kdy určité části budov, v mém případě se primárně jednalo o střechy budov, mají opačně normálovou mapu. Ve výchozím nastavení *viewportu* si této chyby nelze všimnout a chybu spatříte až po exportování do Unreal Engine. Nejjednodušší způsob jak tuto chybu opravit a také zamezit jejímu přehlédnutí je v nastavení *Viewportu* zapnout možnost *Backface culling*.

To se udělá tak, že v přepínači režimu *Viewportu* (obrázek 1. – písmeno C) máme vybraný *Solid mode* a klikneme na šipku pro zobrazení více možností vpravo tohoto přepínače. Zde v poslední části v kategorii *Options* uvidíme zmiňovaný *Backface culling*. Tato možnost vykresluje pouze přední stranu části objektu a vnitřní zůstane „průhledná“. Oprava následně probíhá, že rozbitý model vybereme a přepneme se do *editačního modu*. V něm stiskneme klávesu *A* a vybereme všechny části objektu, alternativně lze v kontextovém menu vybrat položku *Select* a v ní první možnost *All*. Druhým krokem je oprava samotných normálových map. Pro to použijeme klávesovou zkratku *Shift + N*. Toto provede rekalkulaci normálů a předek *Mesh* bude směřovat směrem ven od objektu. Tato možnost ale nemusí vždy fungovat a proto je dalším řešením pouze otočení normál. Místo výběru celého objektu si označíme pomocí *Face select modu* rozbité části a v kontextovém menu vybereme *Mesh* → *Normals* → *Flip*. Tato možnost nám otočí pouze dotčené části a ostatní nechá ve stejném stavu jako předtím.

Oprava normálových map lze také provést přímo v programu Unreal Engine ale oproti Blenderu neumožňuje rekalkulaci ale pouze manuální otáčení. Tudíž je nutné, vybrat vždy veškeré části manuálně což zabere mnohem více času. Ve výchozím nastavení Unreal Engine tuto možnost nenabízí a je nutné pro projekt aktivovat plugin „*Polygon Editing*“.

5.2 EXPORTOVÁNÍ

5.2.1 Vyčištění modelu

Před tím než samotný model vyexportuji, musím udělat pár změn, mezi ty patří oddělení silnic a chodníků od předgenerovaných částí. To udělám tak, že si vyberu objekt, ve kterém mám silnice a chodníky s názvem *Ways:Highway* a přepnu se do *editačního režimu*. V editačním režimu za použití *Face select modu* vyberu libovolnou část silnice kde je materiál, při držení kláves *Shift* vyberu i část na chodníku. Po vybrání obou materiálů jak pro silnici tak chodník, zmáčknu klávesovou zkratku *Shift + G*. Objeví se mi kontextové menu pro výběr podobných prvků *Select similar* a v tomto menu vyberu hned první možnost, kterou je *Material*. Po zmáčknutí budu mít ve výběru vybrané všechny vymodelované silnice a chodníky. Poté pouze zmáčknu klávesu *P*, která otevře kontextové menu pro rozdělení *Separate* a vyberu první možnost s názvem *Selection*. Jakmile tak udělám, budu mít ve výběru oddělené silnice a chodníky od vodících čar. Přepnu se do *Objektového režimu* a vyberu si oddělené části pomocí klávesy *X* smažu.

5.2.2 Zmenšení modelu

Jelikož výběr mapy není nejpresnější a zvyšování či snižování rozsahu mapy pouze po jednotkách znamenalo, že mapa, která byla vygenerovaná, byla moc velká pro potřeby této práce. Proto bylo nutné ji zmenšit. To se provedlo následovně, Prvně jsem si vybral terén a přepnul se do *editačního režimu*. Poté jsem si vybral místo kde budu chtít mapu zkrátit a použil nástroj *Loopcut* na rozdělení. Poté pomocí *Face select modu* vybral části mapy, které již nepotřebuji a smazal je.

Pro objekty to bylo obdobné, vybral jsem veškeré objekty, které v současném stavu neměli pod sebou žádný terén a smazal. Jelikož objektů bylo hodně, tak tento proces znamenal seknutí aplikace do doby, než bude vše smazáno. V mém případě to bylo pár desítek sekund.

Poté následovalo zmenšení velikosti modelu. Model po nahrání do UE byl moc veliký a proto je vhodné jej zmenšit přímo v prostředí Blenderu. Vybral jsem veškeré objekty, který v sobě model má. V objektovém modu jsem vybral všechny objekty. Poté zmáčknu klávesu *S* a zadal hodnotu a na klávesnici napsal *.5*, to zmenšilo vygenerovaný model o polovinu.

5.2.3 Vybrání částí modelu

Při nepovedených pokusných importech v Unreal Engine jsem musel vybrat pouze určitou část modelu, která bude sloužit jako podklad vizualizační aplikace. Celkový počet itemů kompletního modelu atakoval hranici 13 000. Což je pro import velmi náročné a aplikace vždy spadla. Proto jsem vybral pouze část modelů s materiály s nejbližší okolím objektů bez materiálů. Nakonec jsem celkový počet objektů snížil na 2 900, což zrychlilo čas importu ale také exportu a celkovou plynulost aplikace.

5.2.4 Exportování

Pro potřeby exportu do modelu do programu Unreal Engine použiji formát FBX. Samotný export provedu tak, že v panelu nástrojů vyberu záložku *File*, v ní položku *export* a vyberu *FBX (.fbx)*. Otevře se mi okno s nastavením exportu, veškeré nastavení nechám na výchozích hodnotách a pouze si vyberu složku, kam výsledný soubor chci uložit. Poté použiji název souboru jako *export.fbx*. Tento název bude poté v aplikaci Unreal Engine použit jako prefix modelů, které jsem vytvořil. Místo *Plane_047* bude *export_Plane_047* a proto je vhodné nepoužívat diakritiku. Z důvodu velikosti samotného modelu se po kliknutí na export aplikace zasekne a je lepší aplikaci nechat běžet do doby, než se sama odsekne, ukončením na sílu pomocí správce úloh přerušíme export a model bude nekompletní nebo nefunkční. Po dokončení exportu budu mít na určené lokaci v mém počítači soubor s mým modelem.

Tento postup platil pro exportování celého modelu, ale z důvodu vysoké náročnosti na paměť počítače jsem musel prvně vybrat všechny objekty, které jsem chtěl exportovat a poté v okně pro nastavení exportu vyberu možnost, aby exportoval pouze vybrané objekty.

Plný soubor má oproti *.blend* souboru, *64MB* kdy *.blend* soubor má *250MB*, osekkaná verze má poté *32MB*.

6 TVORBA APLIKACE V UNREAL ENGINE

Součástí bakalářské práce je vytvoření vizualizační aplikace. Ve vizualizační aplikaci bude uživatel uživatel sedět v autě, které pojede po předem definované trati. Nebude se tedy muset věnovat řízení a bude mít možnost se rozhlížet pomocí kamery a zrychlovat herní čas.

6.1.1 Vytvoření nového projektu

V úvodní obrazovce aplikace Unreal Engine (obrázek č.9) si vybereme *kategorii Game* a klikneme na *Next*. V sekci pro *předdefinované šablony* si vybereme *Vehicle* a dáme *Next*. Na následující stránce necháme výchozí hodnoty a pouze změníme jméno projektu, to provedeme ve spodní části okna a poté klikneme na *Create project*.

6.1.2 Importování

V prohlížeči assetů (obrázek 9. - písmeno A) si vytvoříme novou složku, jelikož počet naimportovaných objektů bude veliký. To provedeme tak, že na volné místo klikneme pravým tlačítkem myši a vybereme první položku, kterou je *New Folder*. Poté se do nové složky přesuneme a .fbx soubor, který jsme vytvořili exportováním v Blenderu sem přetáhneme. Unreal Engine začne importovat objekty, tento proces může zabrat i několik minut. Po importu všech objektů do prostřední Unreal Engine začne proces kompilace stínů. Samotný proces v mém případě zabral něco přes 3 hodiny. Ale i přes kompilaci stínů je možné stále v aplikaci pracovat, jen jsem se osobně setkával s menší plynulostí aplikace, která byla způsobena kompilací stínů na pozadí. Samotná kompilace má ve správci úloh v operačním systému Windows *nižší než normální prioritu*, takže kompletně nezahltí počítač a procesor.

6.1.3 Vytvoření BlueprintClass pro modely

Pro vkládání modelů do scény si vytvoříme *BlueprintClass* typu *Actor*. To nám pomůže nahrát všechny modely Zlína jako jeden objekt do světa. Vytvoření provedeme tak, že klikneme na volné místo v prohlížeči assetů a vybereme možnost *Blueprint Class*, jako typ vybereme *Actor*, v seznamu assetů se nám zobrazí vytvořená *Blueprint Class* a tu budeme muset pojmenovat. V mém případě jsem zvolil *Model_Zlín*. Do této třídy poté budeme vkládat naše modely. Model pro silnici a terén do této třídy ale nevložíme. Ty budeme vybírat z prohlížeče assetů a myši vždy vybereme modely, při vybírání držíme *Shift*,

to zajistí, že vybereme víc než jen jeden model. Pro vkládání jsem si určil, že to budu vkládat po přibližně 500 modelech, počítač by určitě zvládl víc modelů naráz, ale takto zajistím, že mi při přetahování modelů do *BlueprintClass* aplikace nespadne.

6.1.4 Vložení modelu do scény

Jelikož jsme si při vytváření projektu vybrali jeden ze základních přednastavených šablon, musíme původní vytvořenou arénu na testování jízdy autem smazat. Auto si ale necháme. V prohlížeči aktéru (obrázek 9. písmeno D) si smažeme tedy vše až na složku *RenderFX* a *Lighting* a *SkyShereBlueprint*. Tímto nám ubude starost vytvářet nasvícení modelu a také tvorba oblohy, obloha je v základu animovaná. Poté si v prohlížeči assetů nalezneme náš *Zlin_Model*, a přesuneme ho pracovní plochy. Chvilu po vložení se nám tam zobrazí. Poté v prohlížeči assetů vyberu silnici a terén a vložím je do scény.

V detailech objektu pro silnici a terén použiji stejné hodnoty pro lokaci pomocí souřadnic XYZ, tím zaručím, že bude silnic s terénem ve stejné úrovni jako *BlueprintClass*. A také to bude vypadat stejně jako v Blender modelu.

6.1.5 Kolizní model terénu a silnice

Vytvořené budovy si v sobě při importu přenesli jednoduchou kolizní mapu. Ta pro potřeby vizualizační aplikace stačí, jelikož se uživatel nebude moct pohybovat postavou a tudíž se k budovám nepřiblíží. Auto bude jezdit ale po silnici a ta je položená na terénu. Z tohoto důvodu budeme muset nastavit *komplexní kolizní model* pro tyto 2 objekty, jinak by hrozilo, že se auto začne například vznášet, jelikož kolizní model je jednoduchý a nerespektuje hrany modelu. Oprava kolizního probíhá tak, že si v prohlížeči assetů najdeme náš terén, respektive silnici a klikneme na ní dvakrát. Unreal Engine nám otevře okno ve kterém, čárka můžeme upravovat samotný objekt. V panelu nástrojů si vybereme možnost *Collision* a dáme možnost *Simple Collision*, po zmáčknutí nic neuvidíme, jelikož se v modelu nenachází. V menu baru našeho okna vybereme možnost *Collision* a v ní dáme první možnost, tou je *Add Sphere simplified collision*. Nyní v náhledu vidíme něco, co připomíná kolizi ale je jednoduchá a vůbec nerespektuje model a proto ji musíme změnit. V právě části okna máme detaily týkající se našeho objektu. Ve vyhledávacím poli napíšeme *Collision* a najdeme možnost *Collision Complexity*, místo výchozího *Project Defaults* vybereme možnost *Use complex collision as simple*. V náhledu poté vidíme novou kolizní mapu. Model bude vypadat, jako kdyby měl na sobě pavučinu. Poté v panelu ná-

strojů vybereme možnost *Save* a zavřeme okno. Stejný proces poté použijeme i pro kolizní mapu terénu.

6.1.6 Opravy kolizního modelu u určitých budov

Některé budovy díky svému jednoduchému koliznímu modelu způsobí, že jejich kolizní model zasahuje přímo do silnice. Auto poté při jízdě do tohoto kolizního modelu nabourá, ikdyž není nic vidět. Tento problém je hlavně u budov, které mají předsazenou část. Ta je například vidět poblíž zastávky Dlouhá.



Obrázek 29. Předsazená část budovy

Pro opravení budeme muset najít veškeré budovy, které se s tímto problémem můžou potýkat. Pokud předsazená část nezasahuje do silnice tak ji měnit nemusíme. Postup je stejný jako přidávání kolizního modelu. Tentokrát model nemusíme vůbec přidávat, jelikož ho budova má a pouze ho změníme na komplexní. Jelikož má budova číselné označení, bylo by těžké ji hledat v prohlížeči assetů a proto si v prohlížeči Aktorů vybereme náš *Model_Zlína* a poté klikneme na požadovanou budovu. V detailech je nám poté zobrazí naše hledaná budova a pouze 2x klikneme na náhled modelu u položky *Static Mash*. Poté se nám otevře stejné okno, jako když jsme přidávali kolizní model, ale tentokrát pouze vyhledáme *Collision* a změníme *Project Defaults* na *Use complex collision as simple*.

Objekt uložíme a jako otestování si spustíme aplikaci a pokusíme se autem projet k budově. V tomto případě bude možné dojet autem k budově a nabourat do stěny. Už nás tedy neblokuje.

6.1.7 Vložení zeleně do modelu

Pro vkládání využijeme zabudovaných nástrojů, které má Unreal Engine k dispozici. V panelu nástrojů vybereme přepínač režimů *Modes* a v něm vybereme možnost *Foliage*, nebo můžeme využít klávesové zkratky *Shift+3*. V levé části okna Unreal Engine se nám otevře na místo knihovny dostupných aktérů nová záložka přímo pro *Foliage*, do které poté přetáhneme zeleň. Jejich výběr provedeme klikáním levého tlačítka myši, kdy zároveň budeme držet klávesu *Ctrl*. V tomto okně poté vidíme naše přetažené stromy.

U každého stromu poté vidíme číslovku, která reprezentuje kolikrát se pomocí malování každý strom do scény vložit



Obrázek 30. Foliage

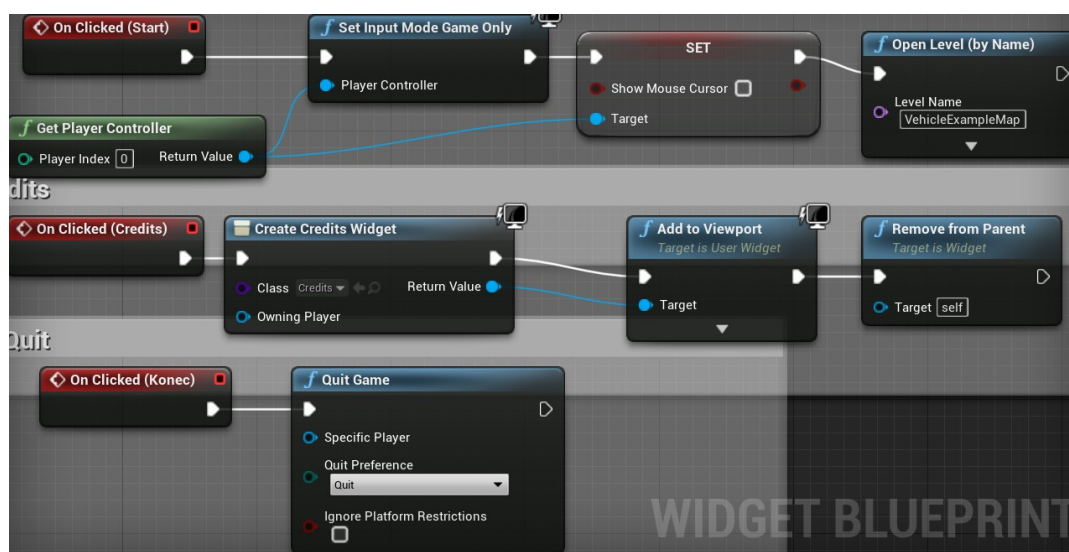
Poté využijeme nástroj *Paint* a budeme nanášet zeleň na terén. Hodnotu *Size* nastavíme na 500 a ostatní hodnoty necháme ve výchozím nastavení. Poté si ve *Filter* musíme odškrknout *Static mesh*, jinak bychom nemohli nanášet stromy na terén. V místech kde chceme pouze jeden strom, si přepneme nástroj na *Single* a pouze klikáme myší, na místě kliknutí se objeví strom.

6.1.8 Tvorba hlavního menu

Pro vytvoření hlavního menu bylo využito prvku *Widget Blueprint*, samotný prvek byl poté vložen do nové úrovně. V samotném *Widget Blueprintu* jsem přidal prvky pro zapnutí aplikace tlačítkem Start, tlačítko pro zobrazení autorů a tlačítko pro vypnutí aplikace. Jako pozadí hlavního menu byl použit vyrenderovaný obrázek z modelu v prostředí *Blender*. Pro vyrenderování byl použit *engine Cycles* se *vzorkováním 128*. Na horní část aplikace byl poté přidán název aplikace.

K vytvoření funkcionality hlavního menu byly využity uzly v oblasti grafů události. Kdy každé tlačítko obsluhovalo svoji událost. Tlačítko start otevře hlavní úroveň s modelem, a začne automatická jízda autem.

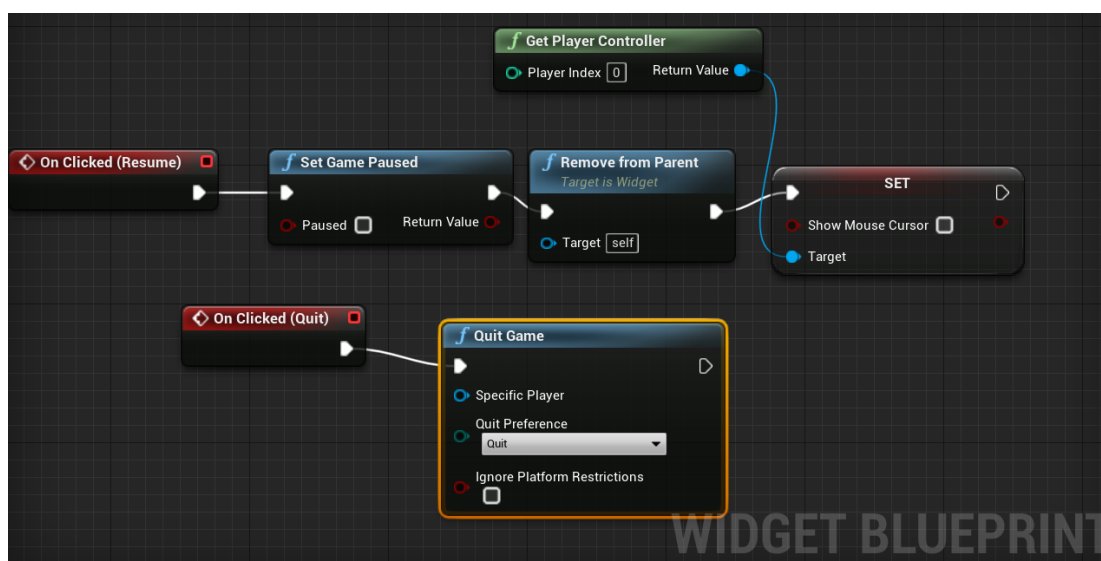
Tlačítko Autor má svůj vlastní *Widget*, který má základní informace o autorovi a tlačítko pro zpět, které nás přesměruje zpátky do hlavního menu. Událost *Remove from Parent* zajistí to, že se předchozí menu uzavře.



Obrázek 31. Blueprint pro hlavní menu

6.1.9 Tvorba pauzy

Součástí vizualizační aplikace je také menu, které se zobrazí po zmáčknutí klávesy ESC. Pro vytvoření menu pro pauzu bylo použito *Blueprint widgetu*. Menu pauzu obsahuje tlačítko pro návrat do hry a také tlačítko pro ukončení aplikace, to funguje stejně jako v případě hlavního menu a rovnou aplikaci ukončí. Po stisknutí klávesy ESC se také zobrazí kurzor. Jako pozadí menu využijeme možnost jemného rozostření a tím docílíme, že hra bude v pauze rozostřená.



Obrázek 32. Blueprint pro pauzu

6.1.10 Vytvoření tratě a automatické jízdy

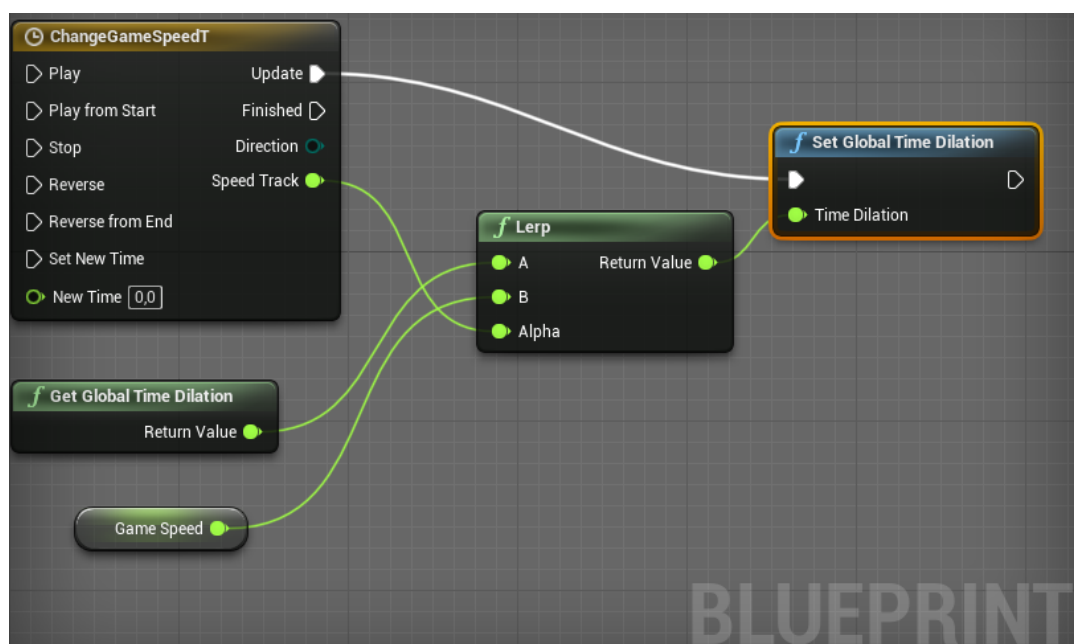
Pro vytvoření simulace automatické jízdy využiji výchozího sedanu, který je k dispozici v naší šabloně, který jsem si pro účely této práce vybral. Auto bude následovat jednoduchou trať, která je tvořená pomocí *Blueprint Class* typu *Actor*, která obsahuje pouze *Splines*. Ty poté pomocí posouvám a klávesou *Alt* tvořím nové body. Otevřu si graf událostí pro výchozí Sedan a smažu části týkající se zatáčení a akcelerace. Akceleraci nastavím na pevnou hodnotu, tedy 0,5 a ovládání zatáčení bude mít na starost výstup funkce *getPath*, kterou si vytvořím. Ta si bude hlídat lokaci auta oproti *Splines* a upravovat tak natáčení kol. Ta bude napojena na *Get All Actor Of Class*, kde jako parametr použijeme třídu pro *Splines*, tedy *Cesta*. Výstup funkce půjde přímo do funkce, která se stará o ovládání.



Obrázek 33. Volání funkce *GetPath*

6.1.11 Volný pohled a zrychlení vizualizační aplikace

Pro potřeby vizualizační aplikace byla přidána možnost, si samotnou aplikaci zrychlit. V aplikaci jsou implementovány 3 různé rychlosti. Aktivují se stisknutím numerických kláves 1, 2 a 3. Kdy hodnota 1 značí výchozí rychlost aplikace, zatímco hodnota 3 je nejvyšší zrychlení aplikace. Pro implementaci byl upraven graf událostí výchozího sedánu, do kterého byla přidána nová proměnná *GameSpeed*, která nabírá hodnoty dle stisku kláves a upravuje tím herní čas.



Obrázek 34. Uzly pro změnu herního času

Volný pohled se aktivuje, pokud uživatel bude držet kolečko myši. V ten moment se pohybem myši bude měnit směr kamery, která je za vozidlem.

6.1.12 Zabalení do spustitelného souboru

K vytvoření spustitelného .exe souboru pro Vizualizační aplikaci je nutné aplikaci zabalit. Unreal Engine má tuto funkcionalitu zabudovanou v sobě a nám stačí vybrat položku File a Package project, kde jako výslednou platformu vybereme Windows 64-bit. Unreal Engine nám poté otevře okno, ve kterém si vybereme, kam chceme výslednou aplikaci uložit. Po vybrání lokace začne proces balení, který může trvat několik hodin. Záleží na komplexitě scény a také na hardwaru samotného počítače.

Výstupem je poté složka, ve které kromě spustitelného souboru .exe nalezneme složku engine ve které se nachází archivační soubor .pak.

7 VÝSLEDKY PRÁCE

Výsledkem bakalářské práce je vizualizační aplikace v prostředí současného Zlína, která umožňuje se projet samořízeným autem po vytvořené trati. Model Zlína byl vygenerován za pomoci rozšíření BlenderGIS a budovy byly ručně upravovány a byly na ně aplikované vhodné materiály.

Oproti vygenerovanému Modelu Zlína zde byly přidány stromy, semaforey a zastávky, které byly ručně vymodelovány a poté importovány do Unreal Engine, aby model i přes jednoduchou grafiku působil živě. Bylo doděláno koryto řeky Dřevnice a vytvoření materiálu imitující vodu. Do modelu byla přidána také železnice, která byla vytvořena v prostředí Blender za pomoci modifikátorů a v neposlední řadě byla vytvořena silnice, která využívá vygenerovaných vertexových čar. Aplikace obsahuje 800 vymodelovaných budov a 556 stromů.

Po spuštění aplikace se objeví menu, ve kterém se můžeme dostat ho hlavní scény, zobrazit si informaci o autorech anebo ukončit aplikaci. Při zapnutí hlavní scény se automaticky zapne jízda, kdy vozidlo následuje předem určenou trať. Auto je možné poté zastavit použitím mezerníku nebo se pomocí zmáčknutí kolečka myši volně rozhlížet po scéně. Za běhu aplikace je možné aplikaci kdykoliv pozastavit a to zmáčknutím klávesy ESC, to otevře menu pro pauzu, ve které můžeme aplikaci správně ukončit anebo se pomocí tlačítka vrátit zpět do hlavní scény, která bude pokračovat v místě původního přerušení. Pozastavení můžeme také zrušit obětovným zmáčknutím klávesy ESC. V samotné aplikaci můžeme zrychlit herní čas a ve třech stupních tím docílit rychlejšího průběhu jízdy.

Výslednou podobu práce lze vidět v příloze (Příloha P IV)

ZÁVĚR

Cílem práce bylo za použití modulu BlenderGIS vygenerovat model současného Zlína, ručně domodelovat chybějící části a vymodelovat budovy. Výsledný model poté převést do aplikace Unreal Engine a v něm vytvořit jednoduchou vizualizační aplikaci.

Teoretická část se věnuje programům, které byly využity v praktické části. První polovina teoretické části se věnuje aplikaci Blender. Byla popsána historie, popis uživatelského prostředí a poté se věnuje funkcím, které Blender nabízí. Součástí této části byl také popis rozšíření BlenderGIS, zde byla popsána jeho instalace a funkce a využití, které toto rozšíření umí.

Druhá polovina teoretické části se věnuje aplikaci Unreal Engine, která byla použita pro tvorbu vizualizační aplikace, tato část se věnuje jeho historii, popisu funkcí a také popisu uživatelského prostředí.

Praktická část začíná generováním podkladové mapy a generováním samotných objektů města Zlín, popisuje postup úprav, které byly provedeny na modelech, vytvoření železnice, stromů, silnice, zastávky, semaforů a také práci s materiály. Poté se věnuje zmenšování výsledného modelu a vyčištění. Dále se věnuje exportování výsledného modelu a popisu a opravám chyb spojených s exportem u normálových map objektů a jejich řešení jak na straně Unreal Engine tak na straně Blenderu.

Druhá část se věnuje importu modelu, práci na tvorbě vizualizační aplikace, zde je popsán proces importování modelů do prostředí programu Unreal Engine a jejich vládání do scény pomocí BlueprintClass. Poté bylo popsáno řešení kolizního modelu budov, které zasahovali do cesty a tvorba různých prvků aplikace jako je Hlavní menu a menu pro Pauzu, v neposlední řadě se věnuje tvorbě automatické jízdy vozidla, které následuje předem vytvořenou trasu po modelu a výsledném zabalení aplikace do spustitelného souboru.

Vytvořený projekt může být dále rozšiřován a postup lze aplikovat na libovolné město. Díky své jednoduché grafice a absenci komplexních textur by mohl být projekt vylepšen za použití virtuální reality, kde by uživatel měl pohled přímo z kabiny auta a mohl se rozhlížet. Vylepšení by mohl dostat terén, jenž by mohl využít Landscape tools, dostupných v programu Unreal Engine.

SEZNAM POUŽITÉ LITERATURY

- [1] *Introduction. Blender 3.1 Reference Manual* [online]. [cit. 2022-02-25]. Dostupné z: https://docs.blender.org/manual/en/3.1/getting_started/about/introduction.html
- [2] *History. Blender 3.1 Reference Manual* [online]. [cit. 2022-02-25]. Dostupné z: https://docs.blender.org/manual/en/3.1/getting_started/about/history.html
- [3] *Features - blender.org* [online]. [cit. 2022-02-26]. Dostupné z: <https://www.blender.org/features/>
- [4] *FREY, Shawn. Blender 3.0: Low Poly Tutorial. All3DP - all3dp.com* [online]. 2021, Dec 11, 2021 [cit. 2021-12-31]. Dostupné z: <https://all3dp.com/2/blender-low-poly-tutorial/>
- [5] *Modeling - blender.org* [online]. [cit. 2022-02-26]. Dostupné z: <https://www.blender.org/features/modeling/>
- [6] *Grease Pencil - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/grease-pencil/>
- [7] *Sculpting - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/sculpting/>
- [8] *Rendering - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/rendering/>
- [9] *Eevee - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://docs.blender.org/manual/en/latest/render/eevee/introduction.html>
- [10] *Cycles- Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://docs.blender.org/manual/en/latest/render/cycles/introduction.html>
- [11] *Animation - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/animation/>
- [12] *Simualtion - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/simulation/>
- [13] *Video Editing - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/video-editing/>
- [14] *Pipeline- Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/pipeline/>

- [15] *Scripting - Blender.org* [online]. [cit. 2022-03-03]. Dostupné z: <https://www.blender.org/features/scripting/>
- [16] *BlenderGIS Addon* [online]. [cit. 2022-03-03]. Dostupné z: <https://blender-addons.org/blendergis-addon/>
- [17] *Basemaps - BlenderGIS/wiki* [online]. [cit. 2022-03-03]. Dostupné z: <https://github.com/domlysz/BlenderGIS/wiki/Basemaps>
- [18] *OSM - import - BlenderGIS/wiki* [online]. [cit. 2022-03-03]. Dostupné z: <https://github.com/domlysz/BlenderGIS/wiki/OSM-import>
- [19] *Difference between using a dem of 30m and 90 m to INSAR* [online]. [cit. 2022-03-03]. Dostupné z: <https://forum.step.esa.int/t/difference-between-using-a-dem-of-30m-and-90-m-to-insar/13128/5>
- [20] *Unreal Engine a postupny vyvoj* [online]. [cit. 2022-03-21]. Dostupné z: <https://www.svethardware.cz/unreal-engine-a-postupny-vyvoj/41366-2>
- [21] *History of the unreal engine* [online]. [cit. 2022-03-21]. Dostupné z: <https://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine>
- [22] *Better with age a history of epic games* [online]. [cit. 2022-03-21]. Dostupné z: <https://www.polygon.com/2012/10/1/3438196/better-with-age-a-history-of-epic-games>
- [23] *Unreal engine 5* [online]. [cit. 2022-03-24]. Dostupné z: <https://www.unrealengine.com/en-US/unreal-engine-5>
- [24] *Unreal Engine - features* [online]. [cit. 2022-03-25]. Dostupné z: <https://www.unrealengine.com/en-US/features>
- [25] *Docs Unrealengine - Blueprints* [online]. [cit. 2022-03-29]. Dostupné z: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/>
- [26] *Material - Blender.org* [online]. [online]. [cit. 2022-05-10]. Dostupné z: <https://docs.blender.org/manual/en/latest/render/materials/introduction.html>
- [27]] FREQUENTLY ASKED QUESTIONS (FAQ) [online]. [cit. 2022-05-08]. Dostupné z: <https://www.unrealengine.com/en-US/faq>
- [28] NIXON, David. *Unreal Engine 4 for Beginners*. Luquinox, 2017. ISBN 978-0692841983.

- [29] BRITO, Allan. *Blender 2.9: The beginner's guide*. 2020. ISBN 979-8676661700.
- [30] DUREC Marián, Markéta MAZÁČOVÁ a Kamil STOKLÁSKA. *3D vizualizace centra současného Zlína*. Zlín, 2014. Univerzita T. Bati ve Zlíně, Fakulta aplikované informatiky.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

WYSIWYG Akronym z anglického *What you see is what you get*, která označuje, že to co vidíš, to dostaneš.

UE Unreal Engine

SEZNAM OBRÁZKŮ

Obrázek 1. Uživatelské prostředí	13
Obrázek 2. BlenderGIS nastavení	18
Obrázek 3. Nastavení Base mapy	19
Obrázek 4. OSM	19
Obrázek 5. STRM	20
Obrázek 6. Rozdíly mezi SRTM 90m,30m a ALOS 30m z pohledu satelitu	20
Obrázek 7. Cube mapping	23
Obrázek 8. Úvodní obrazovka pro výběr šablony a projektu	25
Obrázek 9. Pracovní plocha programu Unreal Engine	26
Obrázek 10. Ukázka Blueprintu v prostředí Unreal Engine	29
Obrázek 11. Rozbalené menu <i>BlenderGIS</i>	32
Obrázek 12. Vygenerované modely budov	33
Obrázek 13. Mapa z OSM	35
Obrázek 14. Vygenerovaná část pomocí rozšíření	36
Obrázek 15. Nanesení obrázku na budovu	37
Obrázek 16. Úprava horní části modelu	38
Obrázek 17. Referenční body	39
Obrázek 18. Střešní okna	39
Obrázek 19. Kápatko v Blenderu	40
Obrázek 20. Výsledná budova	41
Obrázek 21. Chodníky a silnice v modelu v Blenderu	42
Obrázek 22. Nastavení barvy pro teré	43
Obrázek 23. Model koleje	44
Obrázek 24. Hotová kolej v modelu	45
Obrázek 25. Dřevnice	46
Obrázek 26. Modely stromů v prostředí Blenderu	47
Obrázek 27. Model autobusové zastávky	48
Obrázek 28. Model semaforu	48
Obrázek 29. Předsazená část budovy	54
Obrázek 30. Foliage	55
Obrázek 31. Blueprint pro hlavní menu	56
Obrázek 32. Blueprint pro pauzu	57

Obrázek 33. Volání funkce GetPath	58
Obrázek 34. Uzly pro změnu herního času	59

SEZNAM TABULEK

Tabulka 1. Hlavní verze.....	12
Tabulka 2. Hodnoty po vygenerování.....	33

SEZNAM PŘÍLOH

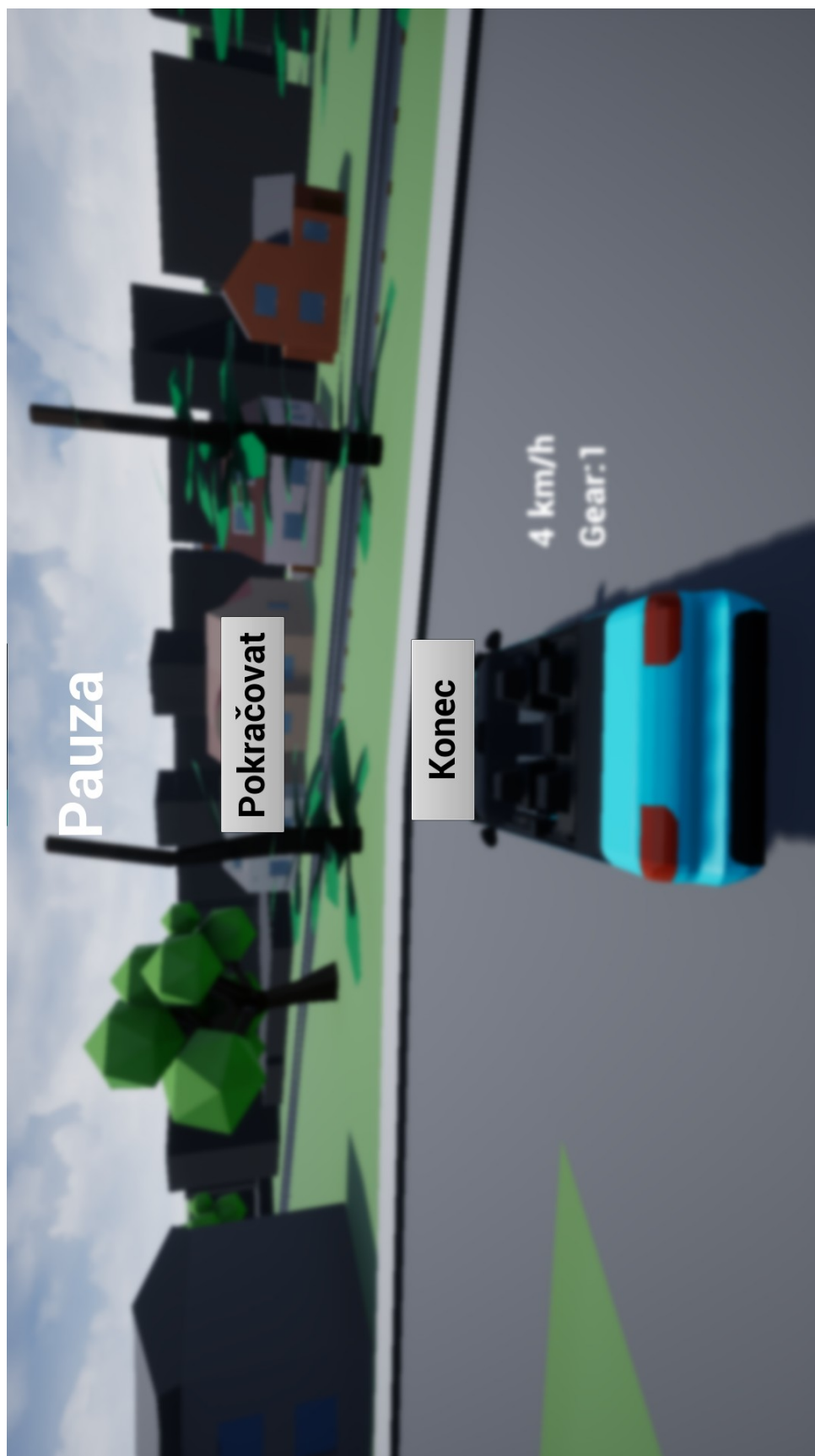
- P I Obsah přiloženého disku
- P II Menu pauzy aplikace
- P III Hlavní menu aplikace
- P IV Pohled po oddálení

PŘÍLOHA P I: OBSAH PŘILOŽENÉHO DISKU

Adresová struktura přiloženého DVD je následující:

- Soubor **fulltext.pdf** obsahuje elektronickou verzi práce ve formátu PDF/A
- Adresář **Blender** obsahuje .blend soubor s modelem Zlína
- Adresář **Aplikace** obsahuje spustitelnou aplikaci
- Adresář **UE_projekt** obsahuje projektové soubory aplikace Unreal Engine

PŘÍLOHA P II: MENU PAUZY APLIKACE



PŘÍLOHA P III: HLAVNÍ MENU APLIKACE



PŘÍLOHA P IV: POHLED PO ODDÁLENÍ

