

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Virtuální prohlídka kampusu Mendelovy univerzity v Brně

Diplomová práce

Vedoucí práce:
Ing. David Procházka, Ph.D.

Bc. Michal Korolkov

Brno 2016

Zde bude vloženo zadání práce.

Chtěl bych poděkovat panu Ing. Davidu Procházkovi, PhD., za odborné vedení práce, cenné rady, trpělivost a ochotu, kterou mi v průběhu zpracování diplomové práce věnoval. Dále bych chtěl poděkovat rodině a přátelům za podporu, kterou mi po celou dobu studia poskytovali.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Virtuální prohlídka kampusu Mendelovy univerzity v Brně**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

Brno 9.5.2016

.....

Abstract

Bc. Michal Korolkov, Virtual visitation of campus Mendel University in Brno. Diploma thesis. Brno: MENDELU, 2016

The aim of this thesis is to create a virtual Mendel university campus walkthrough that can be viewed in the *Oculus Rift*. The virtual walkthrough is based on the *Unity* engine, models are made with the *SketchUp Make* program. Model interaction is implemented using animations or *C#* scripts. In this work, a *SketchUp* module has been used for that serves to delete surplus material, edges, surfaces and other. This work also describes techniques that have been used in *Unity* to decrease computing power requirements (e.g. LOD, culling, bump mapping and other).

Keywords

Unity, Oculus Rift, SketchUp, virtual walkthrough, Mendel University in Brno campus, optimization techniques.

Abstrakt

Bc. Michal Korolkov, Virtuální prohlídka kampusu Mendelovy univerzity v Brně. Diplomová práce. Brno: MENDELU, 2016

Cílem této práce je vytvoření virtuální prohlídky kampusu Mendelovy univerzity, kterou lze zobrazit v *Oculus Rift*. Virtuální prohlídka je vytvořena v herním enginu *Unity* a jednotlivé modely v grafickém programu *SketchUp Make*. Interakce s modelem je zajištěna animací nebo skripty v jazyce *C#*. V práci je použit modul programu *SketchUp* pro úpravu modelů, který slouží k odstranění přebytečných materiálů, hran, ploch a jiné. Dále jsou v práci popsány techniky použité v *Unity* pro snížení nároků na výpočetní výkon (například *LOD*, *culling*, *bump mapping* a další).

Klíčová slova

Unity, Oculus Rift, SketchUp, virtuální prohlídka, areál Mendelovy univerzity v Brně, optimalizační techniky.

Obsah

1	Úvod a cíl práce	11
1.1	Úvod	11
1.2	Cíl práce	11
2	Virtuální realita	12
2.1	Uplatnění VR v praxi	12
2.2	VR v roce 2016	13
3	Vývoj aplikací	15
3.1	Herní engine	15
3.2	Výběr herního enginu	15
3.3	Osvědčené postupy pro VR	17
4	3D Modelování	19
4.1	Prostorová data	19
5	Optimalizace grafického výkonu	21
6	Metodika	24
7	Použité aplikace a technologie	25
7.1	ArcGIS	25
7.2	SketchUp	25
7.3	Unity	26
7.4	Oculus Rift	27
8	Vlastní práce	28
8.1	Vytvoření terénu	28
8.2	Modelování	33
8.3	Scény	35
8.4	Interakce s modelem	36
8.5	Optimalizace výkonu aplikace	39
9	Diskuze	50
9.1	Zhodnocení obtížnosti vývoje	50
9.2	Doporučené rozšíření	51
9.3	Využití řešení v podnikové praxi	51
10	Závěr	53
11	Reference	54
	Přílohy	57

OBSAH	7
A Modely	58
B Scéna	60
C Obsah DVD	62

Seznam obrázků

Obrázek 1: Simulace mírové operace v Bosně	13
Obrázek 2: Virtuální helmy (zleva) – HTC Vive, Oculus Rift, PlayStation VR	14
Obrázek 3: Zobrazení Avatara ve spodní části obrázku	18
Obrázek 4: Základní myšlenka skládání vrstev, které obsahují různá data	19
Obrázek 5: Ukázka objektu v úrovni LOD0 (vlevo) a LOD1	21
Obrázek 6: Použití techniky bump mapping u textury	22
Obrázek 7: Aplikovaný atlas textur (vpravo) na postavu	22
Obrázek 8: Znázornění scény v <i>Unity</i> (vlevo) a výsledek při použití <i>frustum culling</i> (vpravo nahoře) a <i>occlusion culling</i> (vpravo dole)	23
Obrázek 9: Tvorba polygonového <i>shapefile</i> pomocí nástroje <i>Editor</i>	29
Obrázek 10: Výsledná polygonová vrstva budov zobrazená v ArcScene	30
Obrázek 11: Znázornění 3D modelu terénu ve SketchUpu	31
Obrázek 12: Výsledný terén generovaný skriptem <i>object2terrain</i> (v pozadí obrázku) a terén vytvořený ve SketchUpu (podkladový terén)	32
Obrázek 13: Znázornění neupraveného terénu zasahujícího do budovy (vlevo) a jeho úprava	33
Obrázek 14: Úvodní obrazovka aplikace	35
Obrázek 15: Zjednodušené schéma popisující fungování skriptu na otevírání/zavírání dveří	37
Obrázek 16: Zobrazení areálu a možností volného pohybu (zeleně značená oblast)	39
Obrázek 17: Interiér budovy Q po vstupu do budovy (červeně jsou zaznačena místa, kam není umožněn vstup)	39

Obrázek 18: Interiér budovy Q a schodiště vedoucí ke „skriptárně“ (červeně jsou zaznačena místa, kam není umožněn vstup)	40
Obrázek 19: Modul CleanUp ³ a část jeho nastavení	42
Obrázek 20: Textura TER_stena v rozlišení 256 × 128, 512 × 256, 2048 × 1024	42
Obrázek 21: Umístění textury normálové mapy do standardního <i>shaderu</i> v kartě <i>Inspector</i>	44
Obrázek 22: Zobrazení rozdílů mezi běžnou texturou a texturou s normálovou mapou (vpravo) u vybraného objektu	44
Obrázek 23: Budova Q zobrazená v LOD2 (vlevo) a LOD1	45
Obrázek 24: Budova Q zobrazená v LOD1 (vlevo) a LOD0	45
Obrázek 25: Nastavení vlastnosti <i>Static</i> u herního objektu	46
Obrázek 26: Nastavení úrovně kvality pro <i>Oculus Rift</i> v <i>Unity</i>	47
Obrázek 27: Model budovy X v porování se skutečností	58
Obrázek 28: Model budovy M v porování s realitou	58
Obrázek 29: Model budovy Q v porování se skutečností	58
Obrázek 30: Model lavičky s košem a její skutečná podoba	59
Obrázek 31: Výsledný model areálu Mendelovy univerzity	59
Obrázek 32: Ukázka aplikace – pohled z budovy M	60
Obrázek 33: Ukázka aplikace – pohled od budovy X	60
Obrázek 34: Ukázka aplikace – pohled z ulice Lesnická	61
Obrázek 35: Ukázka aplikace – pohled od hlavního vchodu do areálu	61

Seznam tabulek

Tabulka 1: Závislost velikosti souboru na rozměru textury	43
Tabulka 2: Závislost počtu hran na úrovni detailu	45

1 Úvod a cíl práce

1.1 Úvod

Virtuální realita je znovuobjeveným fenoménem současné doby. Jedná se o nové uživatelské rozhraní, jehož cílem je co nejvíce přiblížit počítačové prostředí skutečnosti tak, jak ji zachycují naše smysly. Prozatím virtuální realita využívá sluch, zrak a částečně hmat. S dalšími smysly jako je čich a chuť se experimentuje. Virtuální realita si dokázala najít uplatnění nejen v počítačových hrách, ale také v dalších oblastech jako například architektura, vzdělávání, zdravotnictví a další. Lze předpokládat, že se v blízké době bude oblast uplatnění ještě dále rozšiřovat.

V současné době zažívá virtuální realita rozmach, během kterého se na trhu objevila celá řada řešení, která umožňují uživateli nahlédnout do virtuálních 3D světů. Tuto vlnu zájmu odstartovala společnost *Oculus VR* se svojí virtuální helmou *Oculus Rift*, cenově dostupnou pro veřejnost. Tím byl odstraněn hlavní negativní faktor, tedy cena, která bránila masivnímu používání. To, že *Oculus Rift* měl důležitou roli při rozmachu virtuální reality, dokazuje i jeho integrace do herních enginů jako jsou *Unity*, *Unreal Engine*, *CryEngine* a další.

Jedná se o první závěrečnou práci na Mendelově univerzitě zaměřenou na tuto novou technologii *Oculus Rift*. Přínosem práce pro univerzitu by mohlo být její použití v rámci prezentace na různých akcích (Dny otevřených dveří, Noc vědců apod.).

1.2 Cíl práce

Cílem diplomové práce je vytvoření zjednodušeného modelu areálu Mendelovy univerzity. Model bude obsahovat budovy umístěné v areálu, stromy a další významné vizuální prvky (lavičky, koše, cedule a jiné). Následně bude model zobrazen a uživateli umožněn volný průchod.

Stanovení dílčích cílů

Pro dosažení cíle byl stanoven tento postup:

- nastudování potřebných informací pro zpracování tématu,
- zvolení modelovacího a zobrazovacího programu,
- vytvoření modelu (tvorba budov a dalších objektů umístěných v areálu),
- import vytvořených modelů do zobrazovacího programu.

2 Virtuální realita

V knize *Virtual Reality Technology* (Burdea a Coiffet, 2003) je virtuální realita definována jako počítačové vysokoúrovňové uživatelské rozhraní, které zahrnuje simulaci v reálném čase a interakci pomocí smyslů jako jsou zrak, sluch, hmat, čich a chuť.

Virtuální realita (dále jen VR) není jen médium nebo vysokoúrovňové uživatelské rozhraní, ale také aplikace, které zahrnují řešení skutečných problémů ve strojirenství, zdravotnictví, armádě apod. (Burdea a Coiffet, 2003)

2.1 Uplatnění VR v praxi

Zdravotnictví

VR v oblasti zdravotnictví je forma lékařské informatiky, která zaznamenala za posledních 10 let stabilní, byť pomalý růst. Tento proces byl zbrzděn nedostatkem norem v datovém formátu, ověřovacích opatření a drahými VR technologiemi. VR přináší do oboru řadu výhod jako je lepší vzdělávání lékařů (modelování neobvyklých a vzácných chorob, trénování zákroků na virtuálních modelech spíše než na skutečných pacientech), realističtější postupy pro vydávání osvědčení (například objektivní měření chirurgické dovednosti), příjemnější léčba (v rámci virtuální rehabilitace).

Ve zdravotnictví jsou používány aplikace jako *Anatomic VisualizeR* (zaměřena na vzdělávání), *BioSimMER* (tréninkový nástroj pro simulaci teroristických útoků), *CathSim* (simulátor intravaskulárního cévkování) a další, které se převážně zaměřují na různé druhy zákroků a rehabilitace. (Burdea a Coiffet, 2003)

Vzdělávání, umění a zábava

VR začíná být způsobem, respektive učební pomůckou, ke zprostředkování poznání dětem i dospělým. Využitím interaktivity, se kterou se stává VR poutavá a zábavná, může mít rovněž vzdělávací složku (výuka umění, historie, zeměpisu a tak dále). Oblast zábavy dosahuje nejrychlejšího růstu VR, především díky zapojení gigantů jako *Sega*, *Disney*, *Paramount Pictures*, *LucasArts*, *Microsoft* a další.

Kromě velkého počtu her a různě tematicky zaměřených virtuálních prohlídek zde může být uvedena aplikace *CyberMath*, spadající do oblasti vzdělávání, která slouží jako sdílené prostředí pro poznání matematiky. (Burdea a Coiffet, 2003)

Vojenství

Současný trend, vzhledem k vyšší technologické složitosti a kratší životnosti vojenské techniky, vyžaduje simulátory, které jsou flexibilní, rozšiřitelné a méně nákladné. Virtuální realita odpovídá potřebám vojenské simulace, a proto našla i zde své uplatnění.

Většina aplikací se v této oblasti zabývá simulací různých dopravních prostředků a výcvikem (*VESUB* – simulace řízení ponorky, *VECTA* nebo *DMT* – letecké simulátory a jiné). (Burdea a Coiffet, 2003)



Obrázek 1: Simulace mírové operace v Bosně (Burdea a Coiffet, 2003)

Další oblasti

Své použití VR našla i v **automobilovém průmyslu**, zejména designu. Použití *Oculusu Rift* a virtuálního prostředí umožňuje americké nadnárodní korporaci *Ford Motor Company* (známé také jako *Ford*) rychle vytvářet prototypy aut a zároveň je vyhodnocovat. Tento způsob společnosti výrazně snižuje náklady. Již dříve designeři a inženýři této společnosti využívali virtuální zvukové rozhraní pro ladění zvuku motorů a výfukových systémů. (Carr, 2014)

Také *Mozilla* v rámci svého **webového prohlížeče** začala experimentovat s podporou VR, které sekunduje *Google* se svým prohlížečem *Chrome*. Mimo integraci podpory vývojový tým řeší i způsob předávání informací ve VR. (O'Reilly Media, 2015)

VR si slibuje zvětšení popularity v **rámci koncertů, zpravodajství a dalších živých akcí**. Hudebníci Paul McCartney a Jack White patří mezi několik hudebníků, kteří již vysílají své živé vystoupení v podobě přizpůsobené pro virtuální realitu. (O'Reilly Media, 2015)

2.2 VR v roce 2016

V roce 2016 se na trhu začaly prodávat virtuální helmy. Vše odstartoval 28. března *Oculus Rift* (nyní ve vlastnictví společnosti *Facebook*), kterého následovalo *HTC Vive* (*HTC/Valve*) a *Sony PlayStation VR* (*Sony*). (Gilbert, 2016)

Virtuální helmy nemohou samostatně fungovat a potřebují výpočetní výkon z vnějšího zdroje (počítač nebo herní konzole). *Oculus VR* nabízí řešení v podobě

Oculus Ready počítačů s vysokoúrovňovou vestavbou v ceně přibližně 36 000 Kč¹. Dalším řešením, se kterým přišla společnost *Sony*, je zakoupení herní konzole *PlayStation 4*. Toto řešení platí pouze pro virtuální helmu *Sony PlayStation VR*. Oproti tomu *Oculus Rift* a *HTC Vive* pracuje s jakýmkoliv vysokoúrovňovým počítačem. (Gilbert, 2016)

Omezení virtuální helmy od *Sony* je částečně vykompenzováno cenou (pod hranicí 11 000 Kč²). V případě koupi helmy *HTC Vive* si zákazníci připlatí. Ta stojí více než *Oculus Rift*, kde je cena stanovena přibližně na 14 000 Kč³. Cena virtuální helmy od společnosti *HTC* je více než 21 000 Kč⁴. (Gilbert, 2016)

Současný stav virtuálních helem má několik omezení. Jedním z nich je do jisté míry omezenost pohybu. Virtuální helmy sice disponují poměrně dlouhým drátem, takže uživatel může chodit po místnosti, ale díky jejich konstrukci je blokováno celé zorné pole uživatele. S odstraněním tohoto nedostatku je prozatím nejdále *HTC Vive*. To umístilo kameru na vnější stranu helmy pracující se sledujícím systémem tak, aby byla schopna po nasazení helmy, ukázat stěny a jiné překážky v reálném světě. Dalším nedostatkem může být systém sledující pohyb, který vyžaduje samostatnou kameru nebo senzory, a to za předpokladu, že by někdo bral přidání kamery do obývacího pokoje, kanceláře nebo vyhrazeného VR pokoje jako narušení soukromí. (Gilbert, 2016)

Momentálně mohou VR helmy při jejich užívání u uživatelů vyvolat nevolnost. Tento problém by mohl být postupně odstraněn zlepšujícími se vlastnostmi helem a aplikacemi, které budou založeny na osvědčených postupech. (Gilbert, 2016)



Obrázek 2: Virtuální helmy (zleva) – HTC Vive, Oculus Rift, PlayStation VR (Gilbert, 2016)

¹Cena sestavy *Oculus Ready* je \$ 1500, cena v české měně je uvedena k datu 20.4.2016.

²Uváděná cena výrobcem pro *Sony PlayStation* je 399 Eur, převod na české koruny je uveden k datu 20.4.2016.

³Prodejní cena zařízení, určená výrobcem, je stanovena na částku \$ 599, převod ceny na českou měnu je uveden k datu 20.4.2016.

⁴Uváděná cena zařízení *HTC Vive* výrobcem je \$ 899, převod na českou měnu je uveden k datu 20.4.2016.

3 Vývoj aplikací

3.1 Herní engine

Herním enginem je označován program, který je rozšiřitelný a může být použit, bez větších úprav, jako základ pro různé hry. Herní enginy jsou převážně určené pro konkrétní žánr hry z důvodu jejich specifických požadavků. Najdou se však také výjimky, například *Unreal Engine*, který je úspěšně použit i v jiných žánrech, než pro který byl původně navržen (tj. akční počítačové hry hrané z pohledu první osoby). Obsahem herního enginu, kromě základních částí (například správa paměti, matematická knihovna, vlastní datové struktury a algoritmy aj.), jsou i další systémy zaměřující se na zvuk, vykreslování, animace, fyzikální vlastnosti a další. Požadavkem na většinu herních enginů je, aby mohly být používány na více hardwarových platformách. (Gregory, 2014)

3.2 Výběr herního enginu

V počáteční fázi vývoje aplikace je důležité, kromě promyšlení samotné aplikace a jejího žánru, vhodný výběr herního enginu. Po provedeném průzkumu na oficiálním fóru *Oculus Rift* (Oculus VR, 2015) bylo zjištěno, že se nejčastěji používají následující enginy: *Unreal Engine*, *Unity* a *CryEngine*. Na stránkách výrobce virtuální helmy je uvedena podpora pouze prvních dvou zmíněných enginů.

Unreal Engine 4 nabízí velkou podporu komunity, širokou nabídku lehce ovladatelných nástrojů (Jasani, 2014), skvělé výsledné zpracování a systém *Blueprints* (Masters, 2015). Nevýhodou mohou být vyšší nároky na učení pro začátečníky (tzn. vyšší křivka učení). (Jasani, 2014)

Unity má nejlepší licenční podmínky v herním průmyslu, dobrou podporu komunity a nejnižší křivku učení (Jasani, 2014). Nevýhodou je omezenost používaných nástrojů (v případě modelování složitějších objektů je nutnost využít software třetí strany). (Masters, 2015)

CryEngine je srovnatelný s *Unreal Engine* zejména v grafických schopnostech (Masters, 2015), vyšší křivce učení a možností vizuálního skriptování (*FlowGraph*). Tím ovšem podobnost víceméně končí. Jelikož je *CryEngine* poměrně nový v herním průmyslu, nemá dostatečně velkou podporu, komunitu a s tím související dokumentaci. (Jasani, 2014)

Z tohoto důvodu bude dále provedeno srovnání ve vybraných oblastech pouze u prvních dvou uvedených herních enginů.

Unity 5 vs. Unreal Engine 4

Časy rychlé obnovy/kompilace

Časy kompilace jsou u *Unity* zdaleka nejrychlejší (o několik vteřin oproti 20–100 sekundám *Unreal Engine 4* (dále jen UE4 nebo pouze UE), podle toho, kolik souborů

zdrojového kódu potřebuje UE4 generovat – to se ve velkém zvyšuje při použití GUI vzhledem k jeho závislosti na vnořených makrech).

UE4 poměrně často selhává při rychlé obnově, což vede k nutnosti restartování editoru. V *Unity* se skutečně podařila jak rychlá obnova, tak persistence dat. (Unity, 2014)

Programovací jazyky

Unreal Engine nabízí dvě možnosti pro vytváření nových herních prvků – *C++* nebo *Blueprints*. V případě *C++* pracují uživatelé ve svých oblíbených vývojových prostředích (obvykle *Microsoft Visual Studio* nebo *Apple Xcode*). Pro ty, kteří se rozhodnou pro vizuální skriptování, slouží *Blueprint Editor*. (Unreal Engine Technology, 2004-2015)

Unity nabízí možnost programovat hry pomocí skriptovacích jazyků *JavaScript*, *C#* nebo *Boo* (programovací jazyk podobný *Pythonu*). Skripty je možné naprogramovat v již v zabudovaném editoru zdrojového kódu *MonoDevelop-Unity* nebo si zvolit vlastní vývojové prostředí v nastavení projektu. (Van Oosten, 2012)

Vizuální skriptování

Vizuální skriptování v *Unity* je dostupné pouze z *Asset Store*.

UE4 nabízí již zmíněný systém *Blueprints*, který je poměrně robustní (může například vykonávat *for* a *for-each* cykly) a rychlý (převedení existující funkce na uzel zabere čas v řádu sekund). (Unity, 2014)

Přidávání kódu/vlastního chování

Z hlediska přidávání kódu/vlastního chování u objektů je *Unity* stále nejrychlejší. Jakmile je chování přidáno, jedná se už pouze o čas potřebný k rychlé obnově/kompilaci, ve kterém, jak již bylo zmíněno, *Unity* nemá konkurenci.

Postup pro přidání kódu/vlastního chování v UE4 není tak přímočarý jako v *Unity*. Pokud projekt při načítání nemůže najít definované třídy (například kvůli přejmenování nebo smazání) a jsou zaregistrovány jako chybějící, tak UE4 vymaže všechny související objekty. Chování *Unity* je v tomto ohledu mnohem lepší - pouze odkazuje na chybějící třídu, ale danou vlastnost (komponentu) u všech objektů zanechá. (Unity, 2014)

Uvolňování paměti

K uvolňování paměti v UE4 může dojít i v méně vhodné chvíli. Pokud je během hry vytvořena instance, odvozená z *UObjectu* (což je skoro vše, co není hodnotový typ), dojde k jejímu uvolnění po čase uvedeném v relevantním *.ini* souboru. Tento problém lze opravit, nicméně řešení nefunguje na všechny případy (prozatím pouze na jednotlivé objekty a pole).

I z tohoto porovnání vychází lépe *Unity*. Je pouze potřeba dbát na opatrnost a řešit instancování, destrukci a výkon za běhu. Což by mělo být stejně bráno v po-

taz při vytváření jakéhokoliv skriptu, třídy nebo systému aj. (Unity, 2014)

Renderování a materiály

V oblasti renderování a materiálů nemá systém UE4 konkurenci. Materiálový systém *Unity* je v porovnání s UE4 jednodušší. (Unity, 2014)

Import souborů

UE4 dokáže importovat pouze .fbx soubory. Existují bezplatné konvertory, jako například *Autodesk FBX Converter*, ale většinou se při překladu něco ztratí.

Unity navenek působí, že dokáže pracovat s širší škálou souborových typů. Ve skutečnosti všechny importované soubory konvertuje vnitřně do souborového typu .fbx. (Unity, 2014)

Podpora Oculus Rift

Unity poskytuje vyšší výkon (z hlediska snímkové frekvence), díky čemuž nedochází ke chvění nebo sekání obrazu a následné nevolnosti. UE4 nabízí pro změnu vyšší věrohodnost. Obě tyto vlastnosti jsou ve VR důležité pro výsledný dojem. (Reddit, 2015)

Komunitní podpora/dokumentace

Komunitní podpora/dokumentace je u obou zmíněných enginů na dobré úrovni. UE4 nabízí fórum (monitorované prozatím zaměstnanci *Epic*⁵), online fóra, veřejnou wiki atd. *Unity* je na tom s podporou podobně. (Unity, 2014)

3.3 Osvědčené postupy pro VR

Dokumentace *Oculus Best Practices* (Oculus VR, 2015) uvádí některé, ze základních osvědčených postupů pro poskytování kvalitních výsledků zejména pro práci s virtuální realitou.

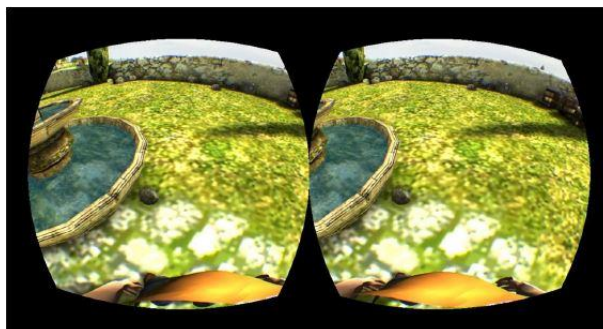
Scéna

- Objekty jsou nejpříjemnější na pohled ve vzdálenosti 0,5 až 2 m (základní jednotkou *Oculus* nástroje je 1 m).
- Uživatel může a měl by být schopen se podívat kdykoliv v kterémkoliv směru.
- Zvuk je rozhodující pro ponoření do děje, stejně jako vize.
- Pohled do scény je obvykle z uživatelského pohledu (*first person*), kdy je uživatel reprezentován tzv. avatarem⁶ nebo jako neviditelná postava – duch (bez těla).

⁵*Epic* (nebo také *Epic Games*) je americká společnost vyvíjející počítačové hry, která vyvinula *Unreal Engine*

⁶Avatar je viditelné znázornění těla uživatele ve VR (tělo bez hlavy).

- Text ve virtuálním světě by měl být dostatečně velký a snadný pro čtení. Doporučuje se také vyhnout tenkým předmětům nebo ozdobným texturám v místech, kde uživatel zaměří svou pozornost.



Obrázek 3: Zobrazení Avatara ve spodní části obrázku (Oculus VR, 2015)

Pohyb

- Pomalejší rychlosti pohybu (zhruba 1,4 m/s u chůze, 3 m/s u běhu).
- Udržení jakékoliv formy zrychlení jako krátké a časté (okamžité zrychlení je pohodlnější než postupné).
- Uživatel a kamerové pohyby by nikdy neměli být odděleny.
- Nepoužívat mírné pohyby nahoru a dolů při simulaci chůze (houpající hlavu) v pohledu z první osoby.
- Nejvíce pohodlná je minimalizace potřeby pohybu dozadu nebo do strany (respektování dynamiky pohybu).
- Pozor je třeba dát také na situace, které mohou vyvolat vizuálně silné pocity pohybu, tzv. simulátor nemoci – strmé schody (svahy), zrychlení nebo opakující se vzory, které se pohybují po velké části obrazovky apod.

Sledování

- *Rift* senzory sbírají informace o vybočení a náklonu uživatele (OR obsahuje gyroskop, akcelerometr a magnetometr). *Rift* v současné době sleduje orientaci uživatele (otáčení hlavy), nikoliv jeho pozici. Objekt v trojrozměrném prostoru má šest stupňů (x, y, a z pozici, stejně tak jako x, y a z rotaci).
- Minimalizace celkové doby mezi pohybem hlavy uživatele a aktualizací obrazu – zpoždění (tzv. latencí). Pokud je zpoždění nevyhnutelné, je lepší zvolit pevné zpoždění oproti proměnlivému.

4 3D Modelování

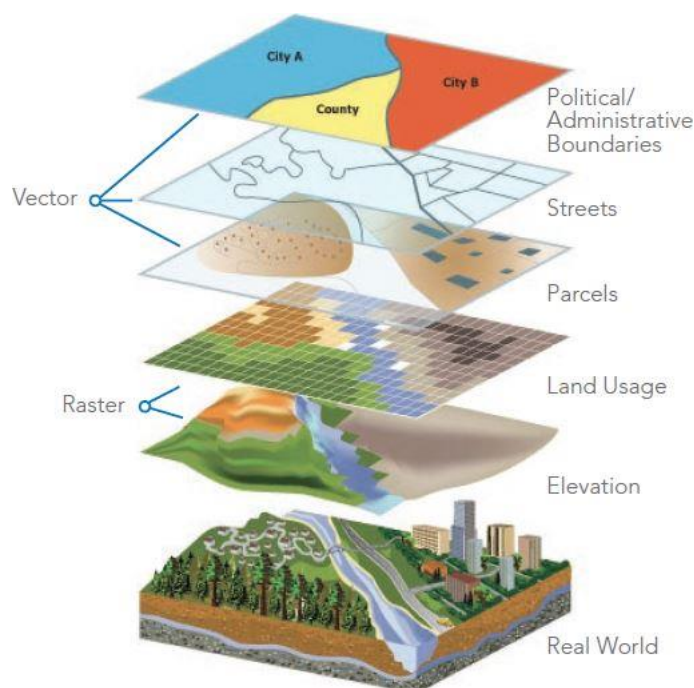
Existuje celá řada modelovacích softwarů. Mezi přední programy patří například *3ds Max*, *Maya*, *LightWave 3D* a jiné, které podporují širokou škálu vlastností a nástrojů. (Vaughan, 2012)

Jednotlivé 3D modely se skládají z trojúhelníků, jelikož současné grafické karty umí pracovat pouze s těmito jednoduchými polygony. Výsledné vykreslení modelu je závislé na počtu polygonů, ze kterých je model sestaven. Offline renderování, které se používá například u filmů, poskytuje vyšší kvalitu vykreslování než renderování v reálném čase, které se uplatňuje v herním průmyslu. Obecně lze říci, že čím je model složitější, tím je méně pravděpodobné, že jej bude možné zpracovávat v reálném čase. (Vaughan, 2012)

Většina herních vývojářů zohledňuje počet polygonů daného modelu, protože jsou omezeni herními enginy. Nikde však není uvedeno, kolik polygonů je pro daný projekt optimálních. Proto je vhodné vytvářet takové modely, které jsou dostatečně jednoduché na renderování a zároveň působí přirozeně na pohled, tzn. vizuálně dostatečně kvalitní. (Vaughan, 2012)

4.1 Prostorová data

Většina dat a měření může být spojována s určitým místem. Takovým datům, kterým je přiřazeno geografické umístění, se říká prostorová. Hlavními modely prostorových dat jsou data vektorová a rastrová. (GIS Geography, 2016)



Obrázek 4: Základní myšlenka skládání vrstev, které obsahují různá data (Esri Press, 2015)

Vektorová data jsou prezentována jako body, přímky nebo polygony, které se skládají z vrcholů a cest. Tato data zobrazují diskrétní entity reálného světa – stavby, studny, cesty a další. Jejich výhodou, mimo jiné, je možnost přiřazení metadat (majitel studny, název ulice u budovy atd.) a estetického výstupu dat. Vektorová data mají složitější datovou strukturu a může dojít k náročnějšímu zpracování. (GIS Geography, 2016)

Rastrová data se skládají z pixelů, ke kterým je přiřazena hodnota nebo třída. Zmíněná data jsou používána pro ukládání dat, která se plynule mění – letecké snímky, nadmořská výška, teplota, únik ropy na moři aj. U rastrových dat je možné použít pro analýzu nástroj, tzv. mapovou algebru. Výhodou rastrových dat je především snadná a rychle proveditelná analýza. U dat takového typu je naopak obtížná reprezentace topologických vlastností a závislost zobrazení na prostorovém rozlišení. Rastrový soubor může být větší v porovnání s vektorovou datovou sadou obsahující stejnou oblast. To z toho důvodu, že hodnota musí být zaznamenána a uložena pro každou buňku v obraze. (GIS Geography, 2016)

Určení vhodného datového typu prostorových dat pro tvorbu není vždy tak přímočaré a záleží na uživateli a jeho konceptu. (GIS Geography, 2016)

Souborové formáty prostorových dat

Prostorová data jsou vytvářena, sdílena a ukládána v mnoha různých souborových formátech. Níže je uveden pouze krátký výčet ze široké škály běžně používaných formátů GIS. Mezi vektorové formáty datového souboru patří:

SHP (*Shapefile*) – formát, se kterým je možné se běžně setkat, označovaný za průmyslový standard geoprostorových dat. Aby byl soubor ve formátu .shp kompletní, musí obsahovat minimálně tři stejnojmenné soubory s rozdílnou příponou (.shp, .shx a .dbf, volitelně poté případně .prj, .sbn a .sbx). (GIS Geography, 2016)

GDB (*Geodatabase*) – je formát vytvořený společností *Esri*, představující kolekci geografických datových souborů různého typu (rastrové, vektorové a tabulkové soubory). Nástupcem .gdb je formát osobní geodatabáze s příponou .mdb, která umožňuje větší kapacitu pro uložení dat (2GB). (GIS Geography, 2016)

KML (*Keyhole Markup Language*) – GIS formát založený na XML, který byl dříve primárně využíván v programu *Google Earth*. Nyní jsou soubory v uvedeném formátu standardem *Open Geospatial Consortium* (zkráceně OGC) a jejich uplatnění je širší. (GIS Lounge, 2011)

Dalšími používanými vektorovými formáty jsou .lyr (*Layer*), .osm (*OpenStreetMap*), .e00 (*E00 ArcInfo Interchange*) a jiné. (GIS Geography, 2016)

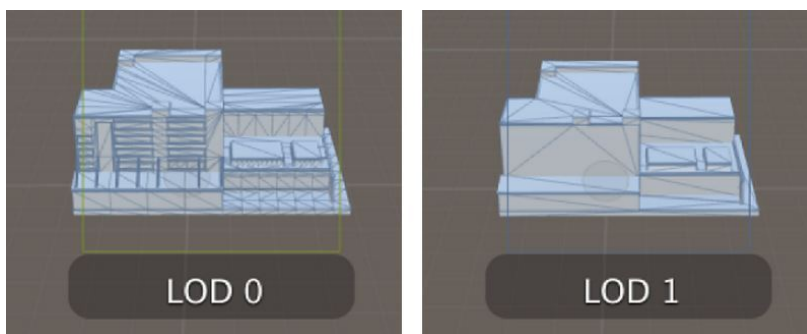
Rastrové formáty jsou zastoupeny průmyslovým obrazovým standardem **GeoTIFF**, který může být doprovázen dalšími soubory (například *XML* apod.). Dále lze využít u prostorových dat komprimovaných formátů .jp2 (**JPEG 2000**), .sid (**MrSID**), .ecw (**ECW**) a další. (GIS Geography, 2016)

5 Optimalizace grafického výkonu

Jedním z klíčových faktorů u hry je výkon. Kromě běžných způsobů (snížení počtu polygonů, odstranění náročných prvků scény – stíny, mlha, dynamické světlo aj.) existují i techniky, které maximalizují rychlost herního vykreslování. (Unity Technologies, 2015)

Level of Detail

U objektů vzdálených od kamery je do značné míry snížena viditelnost detailů daného objektu. Není tedy nutné vykreslovat objekt se stejným počtem polygonů ve větší vzdálenosti od kamery jako při pohledu zblízka. Tento problém umožňuje vyřešit technika *Level of Detail* (zkráceně LOD). Ta optimalizuje vykreslování tím, že umožňuje snížit počet polygonů u modelu s rostoucí vzdáleností kamery. LOD0 se označuje úroveň s největším počtem detailů. (Unity Technologies, 2015)



Obrázek 5: Ukázka objektu v úrovni LOD0 (vlevo) a LOD1 (Unity Technologies, 2015)

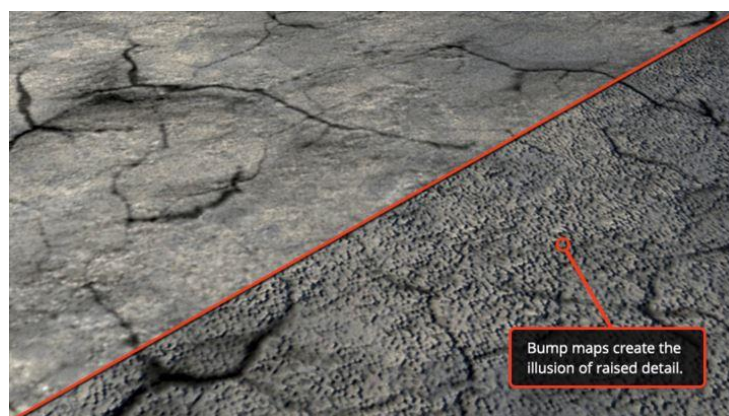
Teslace

Teslace umožňuje modely s malým počtem polygonů předat do grafického procesoru s vysokou kvalitou vykreslování. Tím je vyřešen častý problém mezi grafickým procesorem a centrální procesorovou jednotkou s propustností sběrnice, jelikož *teselace* snižuje množství odeslaných dat. Bez hardwarové *teselace* jsou 3D modely tvořeny často pomocí LOD a vykresleny na základě metriky. Tento způsob je časově náročný (tvorba základního modelu a všech jeho variant) a má větší nároky na paměť. Tyto nedostatky jsou *teselací* vyřešeny. Ta pak umožňuje dynamický LOD pomocí jediného nízkopolygonového modelu. (Varcholik, 2014)

Bump mapping

Zachycení detailů u modelu, jako například šroubů, nýtů nebo škrábanců u trupu letadla, je možné jejich modelováním. Tento způsob ovšem způsobuje nárůst polygonů u modelu a snížení výkonu. (Unity Technologies, 2015)

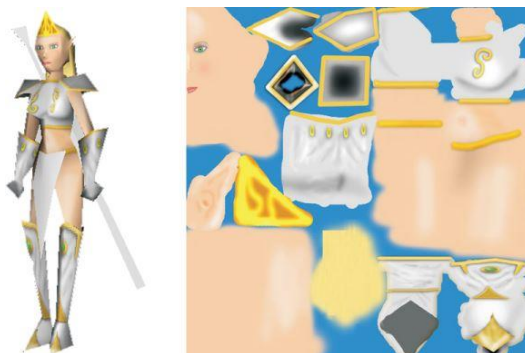
Daleko lepším řešením je použití grafické techniky *bump mapping*, která vytváří iluzi nerovnosti u modelu za použití osvětlení. Při použití této optimalizační metody nejsou přidány do modelu žádné dodatečné polygony nebo jiné prvky. Textury lze navíc poměrně snadno vytvářet a editovat v grafických programech, jako například *Adobe Photoshop*. (Russell, 2015)



Obrázek 6: Použití techniky bump mapping u textury (Russell, 2015)

Atlas textur

Atlas textur se nazývá speciální typ textury, který se skládá z více samostatných textur určených pro jeden objekt. Jednotlivé části atlasu jsou poté mapovány na objekt tak, aby to ve výsledku vypadlo, jako kdyby objekt obsahoval více samostatných textur. Tento speciální typ textury usnadňuje jednodušší správu textury (jeden objekt – jedna textura) a snižuje nároky na paměť grafického akcelerátoru a poskytuje další výhody. Při použití *atlasu textur* není možné využít *MIP-mapping*. (Sedláček, 2006)



Obrázek 7: Aplikovaný atlas textur (vpravo) na postavu (Sedláček, 2006)

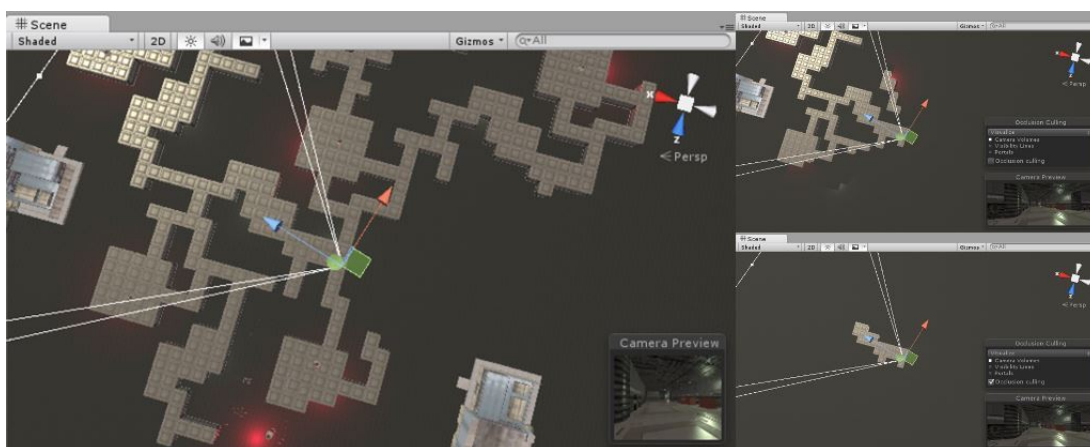
Statické a dynamické dávkování

Herní engine prostřednictvím CPU generuje data, která jsou přenášena přes grafický procesor (GPU) tak, aby mohla být vykreslena na obrazovku. V případě vykreslování různých datových objektů je nejlepší uspořádat tato data do skupin a tím minimalizovat volání z GPU. Skupina obsahující data, která mají být poskytnuta, se nazývá dávka. (Krzeminski, 2014)

Statické dávkování je účinnější než dávkování dynamické (nezatěžuje tolik CPU), ale používá více paměti. Tento typ dávkování má jistá omezení. S objekty nelze manipulovat (otáčet, posouvat, měnit velikost) a všechny objekty musí používat stejný materiál. Dynamické dávkování lze použít z důvodu režie pouze u objektů, které mají méně než 900 vrcholů. (Unity Technologies, 2015)

Culling

Dalším z řešení, jak zvýšit výkon aplikace, může být omezení vykreslování na základě viditelnosti kamery. Funkce *frustum culling* nevykresluje objekty, které nejsou v zorném poli kamery. Oproti tomu *occlusion culling* nevykresluje objekty, které nejsou vidět kamerou, protože jsou zakryty (uzavřeny) jinými objekty. (Shankar, 2015)



Obrázek 8: Znáznornění scény v *Unity* (vlevo) a výsledek při použití *frustum culling* (vpravo nahoře) a *occlusion culling* (vpravo dole) (Unity Technologies, 2015)

Level design

Běžně se celá aplikace vykresluje v jedné scéně. Rozdělení této scény na více částí, které se budou asynchronně nahrávat, zajistí snížení spotřeby paměti a náročnost na vykreslení obrazu. Tento přístup se hodí například u aplikací, kde se uživatel pohybuje v rámci budovy. Pokud se uživatel nachází v jedné místnosti, nemusí být hned při spuštění aplikace načteny okolní místnosti, protože jsou pro uživatele neviditelné. V tomto případě je lepší jednotlivé místnosti umístit do samostatných scén. (Shankar, 2015)

6 Metodika

Zařízením, ve kterém bude zobrazována aplikace, je zvolen *Oculus Rift*. Jedná se o jediné zařízení pro virtuální realitu, které je v době vývoje aplikace dostupné a použitelné (i když pouze ve verzi určené pro vývojáře). Ostatní společnosti, jako *Samsung* nebo *HTC* plánují vydání svých virtuálních helem až v průběhu roku 2016.

Zvoleným herním engineem, ve kterém bude tvořen volný průchod modelem, je *Unity* a to především z důvodu nízké křivky učení a výsledků srovnání, kde *Unity* ve vybraných oblastech (časy rychlé obnovy/kompilace, přidávání kódu/vlastního chování, import souborů a dalších oblastech) převyšovalo *Unreal Engine*. Skripty zajišťující interakci s modelem budou napsány v jazyce *C#*. To především z důvodu sjednocení programovacího jazyka v rámci aplikace. Základní skripty v *Unity* (skripty uvedené u postavy nebo zajišťující inicializaci platformy a další) jsou také napsány v tomto jazyce. Dalším důvodem jsou zveřejněné výsledky ankety na oficiálním fóru *Unity*, kde podle ní je programovací jazyk *C#* používán více než polovinou uživatelů.

Terén bude vytvořen pomocí geografického programu *ArcGIS* a následně převeden do *SketchUp*, kde budou provedeny dodatečné úpravy. Z hlediska uživatelské přívětivosti a volné dostupnosti programu budou veškeré modely vytvořeny ve *SketchUp Make*. Objekty budou vymodelovány na základě plánů poskytnutých Mendelovou univerzitou, kde jsou uvedeny i rozměry budov. U objektů umístěných v areálu (lavičky, koše, lampy a další) budou rozměry získány ručním měřením. Stromy budou staženy z knihovny *Asset Store*. Kvůli zachování důvěryhodnosti budou veškeré modely potaženy texturou. Model areálu bude ohraničen zjednodušenou formou nejbližšího okolí.

Optimalizace grafického výkonu bude u modelů zajištěna pomocí modulu *CleanUp³* ve *SketchUpu* a prostřednictvím použitých technik v *Unity* jako jsou například LOD u budov, *culling*, statické dávkování a další, které sníží náročnost aplikace na technické vybavení počítače.

Splnění osvědčených postupů pro virtuální realitu by mělo zvýšit důvěryhodnost modelu a vtažení uživatele do aplikace. V práci budou uplatněny zejména postupy, které se týkají scény (zvuk, velikost textu, pohled do scény a jiné.) a pohybu (rychlost chůze a běhu, kamera vázaná na uživatele a další).

7 Použité aplikace a technologie

7.1 ArcGIS

Geografický informační systém *ArcGIS*, vytvořený společností *Esri* (2016), umožňuje správu dat, vytvoření profesionálních map, tradiční a pokročilou prostorovou analýzu a řešení reálných problémů. Následně lze tato data vizualizovat v 2D nebo 3D podobě tak, aby v nich byly lépe vidět vztahy, vzory nebo trendy.

Platforma *Esri ArcGIS* nabízí snadnou integraci do stávajících firemních systémů bez úprav, kterou lze využívat na jakémkoliv zařízení, kdykoliv a kdekoliv. (Esri, 2016)

ArcGIS našel uplatnění v mnoha odvětvích jako například obchodu (se zaměřením na pojištění, nemovitosti, bankovníctví a podobně), přírodních zdrojů (zemědělství, lesnictví, vodní zdroje a jiné), vlády (vývoj ekonomiky, městské a regionální plánování a tak dále). (Esri, 2016)

Esri (2016) nabízí svůj produkt v několika verzích – *ArcGIS pro Desktop*, *ArcGIS Online* nebo *ArcGIS pro server*. Blíže bude popsána pouze první zmíněná verze, která je použita v práci.

ArcGIS pro Desktop obsahuje dvě hlavní aplikace *ArcMap* a *ArcGIS Pro*. *ArcMap* slouží pro editaci a zpracování geografických dat, které umožňují vytvářet mapy, upravovat a spravovat prostorová data a provádět celé spektrum analýz potřebných pro změnu surových dat na hodnotné informace. *ArcGIS Pro* je nová 64bitová aplikace umožňující zpracování a vykreslení dat za pomoci nejnovějších 3D vizualizací. (Esri, 2016)

V programu, který podporuje více než 70 datových typů, je možné využít stovky nástrojů pro analýzu. K základním funkcím *ArcGIS* je možné přidat další rozšíření, v podobě pokročilých funkcí, a tak umožnit větší rozsah práce. (Esri, 2016)

Verze *ArcGIS pro Desktop* je dostupná ve třech úrovních – *Basic*, *Standard* a *Advanced*. Licence jsou založeny na rozsahu poskytnutých funkcí a velikosti implementace. (Esri, 2016)

Existuje celá řada volně dostupných alternativ k *ArcGISu* – *QGIS*, *GRASS GIS*, *SAGA* a další, které ovšem mají v porovnání s *ArcGIS* omezenější funkcionalitu. (GIS Geography, 2016)

7.2 SketchUp

SketchUp je volně dostupný, snadno ovladatelný modelovací program, který umožňuje vytvářet 3D modely objektů (budovy, nábytek, krajiny a další), přidání textur, vytvoření animované prohlídky a mnoho dalšího. Modelovací program lze navíc přizpůsobit, a to použitím klávesových zkratk, zobrazením panelu nástrojů, nastavení výchozí šablony a jiné). (Trimble Navigation, 2016)

Kromě výše uvedených výhod *SketchUp* spolupracuje s dalšími programy nebo nástroji, které jsou navrženy tak, aby pracovaly přímo s ním. Pro vytvoření pre-

zentace je možné využít *LayOut*, vytvoření vlastních stylů umožňuje *Style Builder* a sdílení modelů lze zajistit pomocí *SketchUp Mobile Viewer* nebo *3D Warehouse*. Dalším užitečným nástrojem je *Extension Warehouse*, obsahující rozšiřující moduly, kde je možné najít rozšíření pro určitou aplikaci nebo specifické nástroje. Dále *SketchUp* spolupracuje například s aplikací *Google Earth*, kde lze vytvořit náhled modelu v souvislosti s jeho okolím. (Trimble Navigation, 2016)

SketchUp je k dispozici ve dvou verzích – placené *SketchUp Pro* a bezplatné *SketchUp Make*. Verze se od sebe liší, mimo jiné, v počtu podporovaných formátů pro import a export. Zatímco verze *Make* dokáže pracovat pouze se základními formáty jako jsou collada (*.dea), 3Ds (*.3ds), google earth (*.kmz) a dalšími, placená verze navíc podporuje například práci s CAD soubory nebo export do .pdf. (Trimble Navigation, 2016)

7.3 Unity

Unity je vývojová platforma pro hry. Použitím tohoto herního enginu je možné docílit vytvoření kvalitní 3D nebo 2D hry, kterou lze nasadit na různé druhy zařízení jako mobilní telefon, počítač, virtuální/rozšířenou realitu (VR/AR), konzole, web aj. (Unity Technologies, 2016)

Unity umožňuje jednoduché nasazení vytvořené aplikace do zvolené platformy. V současné době podporuje počítačové operační systémy (*Windows*, *OS X*, *Linux*), mobilní systémy (*iOS*, *Android*, *Windows Phone* aj.) a další herní platformy (*Xbox*, *PlayStation*, *Morpheus*, *Oculus Rift*, *Gear VR* a další). (Unity Technologies, 2016)

Každá nová verze *Unity*, má mnoho nových funkcí, vylepšení stávajících funkcí, změn a oprav. (Unity Technologies, 2016)

Momentálně jsou v *Unity 5* dostupné funkce zaměřující se na animace (podpora inverzní kinematiky, integrovaný editor pro animace, stavové automaty atd.), grafiku (statické dávkování, nízko úroňový vykreslovací přístup, fyzikálně založené shadery a další.), skriptování (snadno použitelné www rozhraní, podpora nativního kódu apod.) a další oblasti využitelné v herním světě. (Unity Technologies, 2016)

Součástí *Unity* jsou i nástroje sloužící k optimalizaci vytvořených aplikací (například LOD, *culling*, dynamické dávkování nebo odložený rendering). (Unity Technologies, 2016)

Jak již bylo dříve zmíněno, v *Unity* je prováděna interakce s modelem pomocí jazyků *JavaScript*, *C#* nebo *Boo*. V *Unity* lze využít integrované vývojářské prostředí *MonoDevelop* nebo je možné si zvolit vlastní (například *Visual Studio*). (Unity Technologies, 2016)

Unity disponuje velkou podporou různých formátů (kromě formátů 3D modelů také audio, video nebo textové a další), ale také knihovnou „Asset Store“. Ta obsahuje širokou škálu placených i volně dostupných aktiv jako například textury, modely, skripty nebo animace a mnoho dalšího. Aktiva je možné si stáhnout pomocí jednoduchého rozhraní zabudovaného přímo v *Unity* editoru a importovat ihned do projektu. (Unity Technologies, 2016)

Unity je dostupné, stejně jako *SketchUp*, ve dvou verzích – placené (*Professional*) a bezplatné (*Personal*)⁷. (Unity Technologies, 2016)

7.4 Oculus Rift

Helmu *Oculus Rift* lze označit za průkopníka, který zvedl vlnu zájmu o oblast virtuální reality. Virtuální helma, obsahující velké množství hardwaru (magnetometr, gyroskop, akcelerometr a jiné), je navržena tak, aby byl u uživatele vyvolán pocit ponoření do virtuálního světa. Video je posíláno do *Oculus Rift* přes HDMI nebo DVI pro notebooky s novějšími grafickými kartami (výrobce doporučena grafická karta je *NVIDIA GTX 970* nebo novější). *Rift* je dále vybaven USB portem, prostřednictvím kterého je možné připojit periférie (sluchátka, ovládací zařízení a další). Tím je ale zvýšen odběr energie *Riftu* a ten je tedy vázán na síťový adaptér. (Wareable, 2016)

V poslední vývojářské verzi virtuální helmy je obrazovka tvořena displejem z telefonu *Samsung Note 3*. Jedná se o verzi, na které byla vyvíjena aplikace. Finální verze určená pro zákazníky disponuje již dvěma displeji s vyšším rozlišením a obnovovací frekvencí. (Wareable, 2016)

Princip virtuální helmy je založen na binokulárním vidění, které popisuje způsob, jakými jsou viděny dva pohledy na svět současně. Pohled od každého oka se mírně liší a lidský mozek spojuje tyto pohledy do jednoho trojrozměrného stereoskopického obrazu. (Oculus VR, 2015)

Oculus prošel různými iteracemi u sledovacích technologiích (zásadní pro získání informace o poloze v prostoru). Finální verze bude mít bezdrátové čidlo, malý mikrofon ve tvaru sloupku, který bude snímat řadu zapuštěných infračervených LED diod v helmě. (Wareable, 2016)

Další změnou u nového modelu oproti předchozím jsou odnímatelné sluchátka. *Oculus Rift* dále plánuje rozšířit příslušenství k helmě o ovladače *Oculus Touch*, které simulují činnosti rukou (zvedání a manipulování s věcmi) ve virtuálním světě. Jejich prodej by měl začít na konci roku 2016. (Wareable, 2016)

Společnost *Oculus* uzavřela partnerství se společností *Microsoft*, což znamená, kromě případné podpory *Riftu* v platformě *Xbox*, také kompatibilitu s operačním systémem *Windows 10*. (Wareable, 2016)

Aby došlo k sjednocení rozhraní, vývojáři vytvořili zcela nové rozhraní – *Oculus Home*. To umožňuje zobrazení dostupných her, propojení s přáteli, nastavení zařízení a další. (Wareable, 2016)

⁷Licenční ujednání a možnost použití dané verze je možné dohledat na oficiálních stránkách.

8 Vlastní práce

8.1 Vytvoření terénu

Při vytváření modelu terénu byly vyzkoušeny dva přístupy k tvorbě. Na základě jejich nedostatků nebo přínosů (popsaných v podkapitole Srovnání způsobů tvorby terénu) je vybrán pouze jeden z nich.

ArcGIS

Základní podklad pro terén je vytvořen v geografickém programu *ArcGIS 10.2*, respektive v aplikaci *ArcMap* a *ArcScene*, která umožňuje prostorové zobrazení modelů.

U projektu v *ArcMap* je zvolená relativní cesta⁸ k podkladům a vrstvám, aby projekt bylo možné snadno přenášet mezi zařízeními. Další možností pro přenositelnost dat je jejich uložení do vytvořené osobní geodatabáze.

V České republice je používán souřadný systém *S-JTSK* (v *ArcMap* zvolený *S-JTSK Krovak EastNorth*), který je také potřeba nastavit u projektu a pracovat s podklady a vrstvami ve zmíněném souřadném systému. Uvedený souřadný systém (*S-JTSK*) je také přesnější než další používaná varianta souřadného systému – *WGS84*. Nicméně v rámci vytvořeného projektu v *ArcMap*, který je zaměřen na malé území, jsou tyto rozdíly zanedbatelné.

Terén je vytvořen několika nástroji, které umožňuje *ArcMap* – georeferencováním, tvorbou polygonové vrstvy, tvorbou *TIN* sítě, osamostatněním vrcholů *TIN* sítě a interpolační metodou. Nakonec je terén zobrazen v *ArcScene* a exportován jako 3D model.

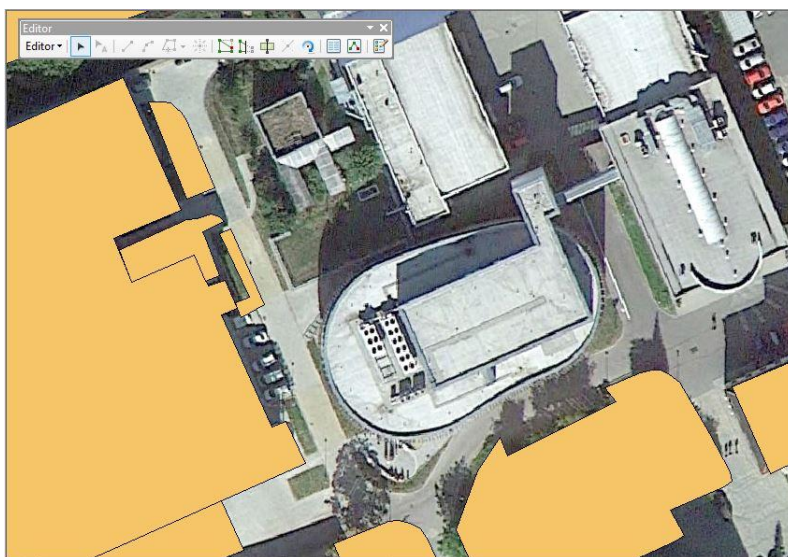
V rámci tvorby terénu jsou stažena vektorová data obsahující celé území České republiky. Stažená data obsahují neaktuální *shapefile*⁹, které jsou ale dostačující pro potřebu georeferencování rastrového obrázku. Rastrový obrázek, v podobě leteckého snímku, je získán z programu *Google Earth Pro*. Program umožňuje uživateli stažení aktuálních dat ve velkém rozlišení oproti dalším možnostem získání takového obrázku (*mapy.cz*, *geoportal.gov.cz* a podobně).

Georeferencování rastrového obrázku je prováděno podle budov uložených ve vektorových datech, protože se jedná o nejjednodušší metodu určení místa na obrázku a ve vektorových datech. Práce v *ArcMap* je zaměřena pouze na areál Mendelovy univerzity a nejbližšího okolí. Proto je vhodné před začátkem georeferencování ořezat vektorová data budov z důvodu snížení celkové velikosti projektu a rychlejší reakci *ArcMapu*. Ořezání je prováděno pomocí nástroje *Select Features* (nalézá se pod hlavní nabídkou programu) a následného exportu vybrané části do samostatného *shapefileu*. Nástroj pro georeferencování je umístěn v hlavní nabídce *Customize* → *Toolbars* → *Georeferencing*.

⁸Relativní cesta se nastavuje pomocí *checkboxu* v nabídce programu *File* → *Map Document Properties* → *Store relative pathnames to data sources*.

⁹*Shapefile* je označení pro průmyslový standardní formát geoprostorových dat.

Po úspěšném umístění rastrového obrázku je nutné vytvořit vlastní polygonový *shapefile*, ve kterém jsou zachyceny půdorysy budov a travnaté plochy areálu. Zmíněné části jsou uloženy společně v jednom *shapefile* z důvodu snadnější editace. Ty jsou vytvořeny v *shapefile* polygonem pomocí nástroje *Editor* (*Customize* → *Toolbars* → *Editor*) podle rastrového obrázku. Nástroj umožňuje přidávání dalších geometrických prvků jako jsou obdélníky, kruhy a jiné. Po dokončení úprav polygonové vrstvy je nutné tyto změny uložit (*Save Edits*) a ukončit úpravy (*Stop Editing*) proto, aby se změny promítly do výsledného *shapefile*.



Obrázek 9: Tvorba polygonového *shapefile* pomocí nástroje *Editor*

Pro prezentaci reálného terénu je použit nástroj *TIN*¹⁰. Výsledkem nástroje jsou vektorová data, která rozdělují geografický prostor do souvislých nepřekrývajících se ploch (trojúhelníků). Vstupním souborem *TINu* je *shapefile* s vrstevnicemi.

K dosažení spojitě prezentace povrchu terénu je použita interpolační metoda *IDW*¹¹, která zároveň s metodou *Spline* poskytuje dobré výsledky ve výchozím nastavení. Metoda *IDW* provádí výpočet bodu podle váženého průměru ostatních bodů v okolí (jejich počet je jedním ze vstupních parametrů metody). Jelikož metoda pracuje s body, tak není možné použít *shapefile* vytvořený pomocí *TINu*, který obsahuje plochy. Z tohoto důvodu je nutné převést posledně zmíněný *shapefile* pomocí nástroje *TIN Node*¹² na body. Vstupním souborem pro nástroj *TIN Node* je *shape-*

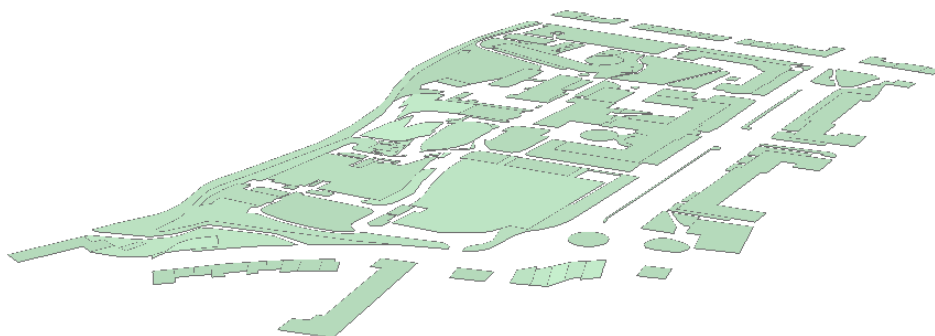
¹⁰ *TIN* je spuštěn prostřednictvím *Catalog* → *Toolboxes* → *System Toolboxes* → *3D Analyst Tools.btx* → *Data Management* → *TIN* → *Create TIN*. V případě nemožnosti spuštění *TIN*, je nutné tento nástroj povolit v nastavení programu *Customize* → *Extensions* → *3D Analyst*.

¹¹ Interpolační metodu *IDW* lze spustit *Catalog* → *Toolboxes* → *System Toolboxes* → *Spatial Analyst Tools.tbx* → *Interpolation* → *IDW*.

¹² Nástroj *TIN Node* je možné spustit v *Catalog* → *Toolboxes* → *System Toolboxes* → *3D Analyst Tools.btx* → *Conversion* → *From TIN* → *TIN Node*.

file z *TIN*. Nyní lze již provést interpolaci vstupních dat vybranou metodou *IDW*¹³. S výstupním souborem metody se dále pracuje ve 3D prohlížeči *ArcScene*.

V *ArcScene* se pracuje s polygonovým *shapefilem* (obsahujícím půdorysy budov a travnatý povrch) a výstupním souborem z *IDW*. Aby polygonový *shapefile* odpovídal skutečnému terénu je zapotřebí kopírovat povrch výstupního souboru z *IDW*. Toho je docíleno změnou nastavení ve vlastnostech *shapefile*¹⁴. Následně lze scénu exportovat do 3D modelu ve formátu *.wrl* přes hlavní nabídku programu (*File* → *Export Scene* → *3D*).



Obrázek 10: Výsledná polygonová vrstva budov zobrazená v *ArcScene*

SketchUp

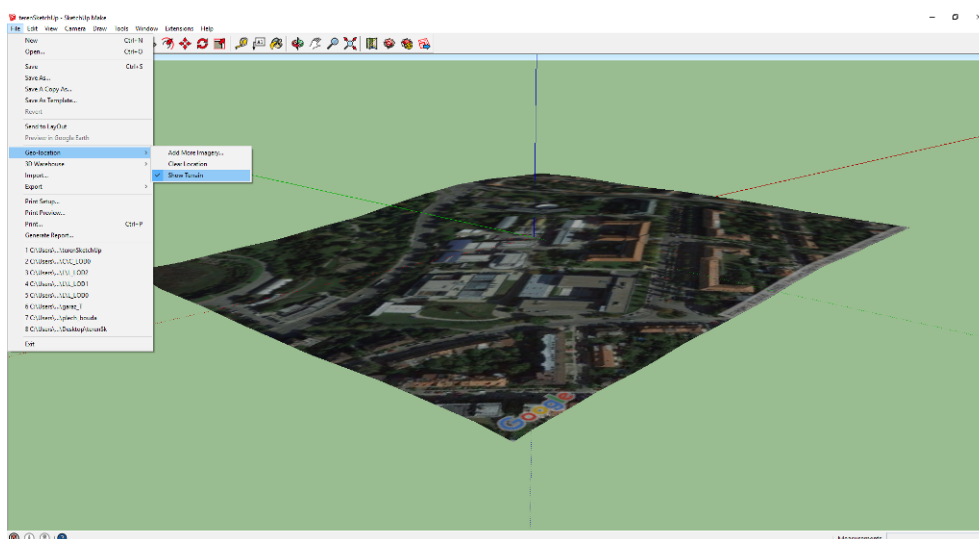
Základní podklad pro tvorbu terénu je získán pomocí nástroje *Geo-location* v programu *SketchUp*. Nástroj se nachází v hlavní nabídce *File* → *Geo-location* → *Add Location*. Po spuštění nástroje se zobrazí nové okno se satelitním snímkem a vyhledávacím polem pro zadání požadovaného místa. Do vyhledávacího pole může být zadána oblast nebo přímo konkrétní adresa. Nalezené místo lze vybrat tlačítkem *Select Region*. Poté může být vybraná oblast ještě před importováním do modelu upravena pomocí špendlíků umístěných v rozích ořezového okna. Import terénu je dokončen stisknutím tlačítka *Grab*. Výsledný terén získaný z nástroje je znázorněn komponentou s texturou leteckého snímku.

Posledním krokem je zobrazení terénu v 3D podobě. To se zapíná opět v hlavní nabídce *File* → *Geo-location* → *Show Terrain*. U takto vytvořeného terénu lze říci, že obsahuje nízký počet polygonů, s ohledem na velikost plochy, kterou pokrývá

¹³Interpolační metodu *IDW* lze spustit *Catalog* → *Toolboxes* → *System Toolboxes* → *Spatial Analyst Tools.tbx* → *Interpolation* → *IDW*. Hlavními vstupními parametry pro spuštění jsou vstupní soubor (*shapefile* s *TIN* body), pole hodnot (*shape.Z*), způsob hledání bodu (*Variable*) a počet hledaných bodů v okolí, na základě kterých bude vytvořen nový bod (*Number of points*).

¹⁴Přízpůsobení terénu je prováděno ve vlastnostech *shapefile* v záložce *Base Heights*, výběrem možnosti *Floating on a custom surface*.

(konkrétně v případě této práce má 351 hran a 208 ploch). Z důvodu nepřírozenosti a špatné editace v případě budoucích změn není vhodné tento terén použít v herním enginu. Proto je terén vytvořený ve *SketchUpu* (dále jen podkladový terén) importován do *Unity*, kde slouží jako podklad pro generování *Unity* terénu a jsou podle něj umístěny budovy a další prvky areálu.



Obrázek 11: Znárodnění 3D modelu terénu ve SketchUpu

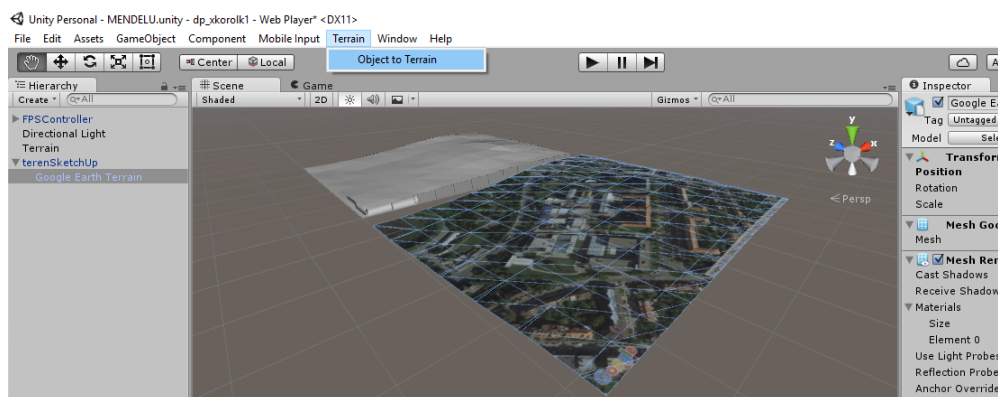
Pro zjednodušení je použit skript *object2terrain*, který převede *mesh* podkladového terénu na výškovou mapu, na jejímž základě je generován *Unity* terén. V případě použití musí být skript *object2terrain* umístěn ve složce */Assets/Editor/* projektu, kde má být použit. Pokud je skript správně umístěn, objeví se v hlavní nabídce *Unity* položka *Terrain*.

Terén v *Unity* je vytvořen jako 3D objekt v hlavní nabídce *GameObject* → *3D Object* → *Terrain*. Vytvořený terén má základní rozměry (500 × 500 × 600 m), které je potřeba upravit podle podkladového terénu¹⁵.

Po upravení rozměrů u terénu stačí označit *mesh* u podkladu v hierarchii *Unity* a spustit generování terénu v hlavní nabídce *Terrain* → *Object To Terrain* (viz obrázek 12). Výsledný terén nezobrazuje terénní úpravy, které je zapotřebí provést pomocí nástrojů pro úpravu terénu v záložce *Inspector* (zvýšení, snížení výšky terénu a jiné). Tento terén je již optimalizován pro vykreslování, snadno upravitelný a působí přirozeným dojmem.

U terénu je možné provádět vykreslování pouze viditelných částí za předpokladu, že terén není tvořen jako jeden celek a jednotlivé části jsou označeny jako statické (viz strana 46). Pro rozdělení terénu může být použit balíček *Dvornik-Split-Terrain*, který obsahuje skript (napsaný v *C#*) rozdělující terén na 4 části se všemi

¹⁵Rozměry objektu ve *SketchUpu* mohou být získány pomocí nástroje svinovací metr nebo modulu, například *jj_get_dimensions.rb*



Obrázek 12: Výsledný terén generovaný skriptem *object2terrain* (v pozadí obrázku) a terén vytvořený ve SketchUpu (podkladový terén)

jeho detaily (stromy, texturou a dalšími). Jednotlivé části mohou být dále rozdělovány.

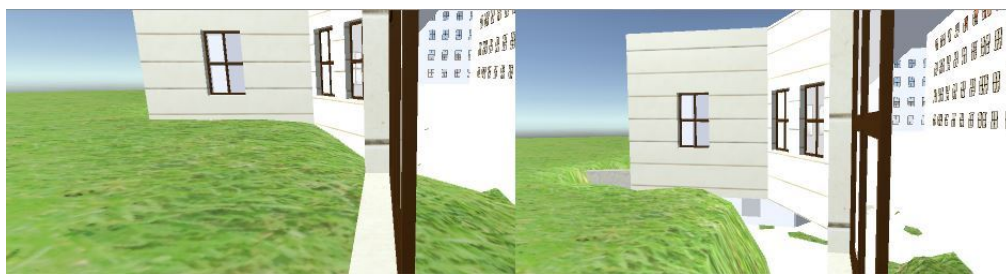
Po importu balíčku do *Unity* se objeví v hlavní nabídce položka *Dvorník*. V případě rozdělení terénu stačí označit terén, který má být rozdělen, a zvolit v nabídce *Unity Dvorník* → *Terrain* → *Split Terrain*. Nově vytvořené terény jsou automaticky přidány do projektu a zároveň je zachován i stávající terén, který může být odstraněn.

Srovnání způsobů tvorby terénu

Oba zmíněné způsoby tvorby terénu mají své výhody, ale také určitá omezení. Zároveň je zapotřebí u obou způsobů provést dodatečné terénní úpravy, protože ve výsledném modelu terénu nejsou zohledněny stavební práce.

V porovnání terénu vytvořeného pomocí *SketchUpu* je terén vytvořený prostřednictvím *ArcGIS* přesnější. Naopak jeho nevýhodou je poměrně náročná tvorba a nedostupnost programu ve volně šiřitelné licenci (např. *freeware*). Import výstupního souboru z *ArcScene* (v datovém formátu *.wrl*) pro dodatečné úpravy může být proveden pouze do programu, který jeho formát podporuje (například *Blender*). *SketchUp* tento formát podporuje pouze ve verzi *Pro*. Výsledný terén působí nepřirozeným dojmem, hůře se upravuje a není nijak optimalizován pro *Unity*.

Druhý způsob tvorby terénu, pomocí programů *SketchUp* a *Unity*, většinu zmíněných nedostatků odstraňuje. Terén působí přirozeným dojmem, je snadno upravitelný a optimalizovaný. Navíc odpadá potřeba použití aplikace *ArcGIS* nebo jiné, která není volně přístupná. Nevýhodou uvedeného způsobu je omezenost nástrojů pro úpravu terénu v rámci *Unity*. Není zde možné navázání terénu na budovu (vytvoření kolmému povrchu terénu, viz obrázek 13). Uvedené omezení je z hlediska této práce podstatné, jelikož má uživatel možnost nahlédnout i do interiéru budov. Bez této úpravy by terén zasahoval dovnitř budovy nebo v případě snížení terénu by terén nenavazoval přímo na budovu (vytvořil by se sráz u budovy).



Obrázek 13: Znárodnění neupraveného terénu zasahujícího do budovy (vlevo) a jeho úprava

Z tohoto důvodu byl vybrán pro tvorbu terénu program *ArcGIS*.

8.2 Modelování

Modely použité v aplikaci jsou vytvořeny ve *SketchUp Make 2016* a rozděleny do jednotlivých skupin. To umožňuje jejich jednodušší editaci a přiřazení barvy nebo textury.

Vytvořené modely musí být uloženy ve starším typu programu, například *SketchUp Make 2015* a níže, jelikož nejnovější verze není podporována v *Unity*. U modelů uložených v nejnovějším formátu nedochází k rychlé obnově v editoru herního engine a nefunguje jejich import. Vhodnějším řešením importu modelů do *Unity*, než v .skp, je v souborovém formátu .fbx. Tohle není momentálně možné z důvodu omezeného exportu programu *SketchUp Make 20xx* a neexistujícího bezplatného modulu pro export do výše uvedeného souborového typu.

Modely lze také importovat ze *SketchUp Make* do *Unity* v souborovém formátu .dea. Ty mají následně v porovnání s modely uloženými v souborovém formátu .skp menší velikost souboru. Nevýhodou u formátu .dea je nutná změna měřítka modelu v jednom z uvedených programů. Další nevýhodou je import modelů do herního engine bez textur.

Budovy

Modely budov jsou vytvořeny na základě digitálních plánů poskytnutých Mendelovou univerzitou. Plány, v souborovém formátu .dwg, obsahují u většiny budov okótovaný jeden řez dané budovy a půdorysy jednotlivých podlaží. U budov, které nemají plány obsahující řez, bylo nutné provést ruční měření a chybějící výška u budov byla zjištěna pomocí digitálního metru.

Pro jednodušší zpracování a získání rozměrů byly dále plány zpracovány v bezplatném CAD programu *Draft IT* (může být použit jakýkoliv program pracující s daným souborovým formátem), ve kterém je možné rychle změřit potřebné chybějící rozměry v plánech.

Každá budova je modelována ve třech úrovních detailu, kvůli optimalizaci výkonu (viz kapitola Optimalizace grafického výkonu).

Pozice v *Unity* je budovám již přiřazena v programu *SketchUp* přímo podle terénu, což není vhodným řešením. *Unity* dokáže pozicování objektů pomocí skriptu nebo nástrojů (rotace, změna měřítko nebo translace). K lepšímu umístění může sloužit i zobrazená mřížka ve scéně editoru. Nebyl však nalezen způsob (kromě použitého, kdy jsou budovy pozicovány již ve *SketchUpu*), kde je možné zcela přesně umístit objekty na určité místo ve scéně.

Přesné umístění budov vyžaduje terén, u kterého je vynechán povrch v místě budovy. Ten je vynechán z důvodu, aby nezasahoval do budovy, která je částečně umístěna i pod úroveň terénu (například budovy A, O, T). Tento nedostatek by mohl být viděn průhlednými plochami u budov při volném průchodu.

Při zobrazení v *Unity*, především u budov, docházelo ke vzniku prolínání, tzv. *Z-fighting*. To bylo znázorněno u jednotlivých částí budov rychle blikajícími texturami. K jevu dochází u částí modelů, které leží ve stejné rovině, tzn. mají stejnou hodnotu v *z-bufferu*. Jedním z možných řešení problému zobrazení je upravení atributů u kamery ve scéně, které slouží k určení vzdálenosti pro začátek a konec vykreslování obrazu (pokud možno co nejvyšší hodnota u *near* a nejnižší u *far*).

V příloze A jsou výsledné modely porovnány se skutečnou podobou budov.

Objekty v areálu

U objektů nacházejících se v areálu (lavičky, koše, cedule a jiné) jsou jejich rozměry získány ručním měřením. Tyto objekty jsou vytvořeny pouze v jedné úrovni detailu.

Stromy jsou získány z knihovny 3D modelů *Asset Store* kvůli nenáročnosti na hardware počítače vedoucí k optimalizaci aplikace. Jejich výška je odhadnuta na základě blízko umístěného objektu (nejčastěji budovy) se známou výškou.

Textury

U všech modelů jsou použity bezešvé textury proto, aby nedocházelo k negativnímu narušení vizuálního dojmu při pohledu na objekt. Textury, které nejsou získány pomocí fotoaparátu, jsou staženy z webových stránek¹⁶. Jedná se především o textury, u kterých by bylo jejich získání obtížné (například střecha budovy) nebo náročné provádění dodatečných úprav (dřevo, kovový plech a další), jelikož pořízené fotografie obsahovaly vinětaci¹⁷. K editaci fotografií jsou použity nástroje v programu *Adobe Photoshop* (klonovací razítko, nástroj výběru a jiné). Textury vytvořené z fotografií jsou zaměřeny především na fasády jednotlivých budov proto, aby se modely co nejvíce blížily k reálnému stavu a byla zachována důvěryhodnost modelu.

¹⁶Textury jsou staženy z webových stránek <http://www.textures.com/>

¹⁷Vinětace je optická vada objektivů, projevující se nižším jasnem na okraji fotografie.

8.3 Scény

Scény obsahují jednotlivé objekty aplikace. Mohou být použity k vytvoření hlavního menu, jednotlivých úrovní (kde lze využít *level design*) a dalších. V každé scéně mohou být individuálně nastaveny její prvky (například osvětlení), aniž by ovlivnily ostatní scény.

Ve vytvořené aplikaci se nachází dvě scény - úvodní obrazovka aplikace a prostředí areálu Mendelovy univerzity, ve kterém je uživateli umožněn volný průchod.

Úvodní obrazovka

Hlavní menu aplikace je vytvořeno v *Unity* pomocí komponent uživatelského rozhraní, tzv. UI objektů. Hlavní komponentou pro vykreslování všech dalších UI objektů je plátno (tzv. *canvas*). Do něj jsou vloženy v hierarchii editoru další prvky jako panel nebo text. Do panelu je vložen obrázek (zobrazující areál z určitého pohledu), který tvoří pozadí celé úvodní obrazovky. Do některých textových objektů je přidána komponenta Tlačítko (*Button*). To z toho důvodu, aby mohlo být u takového textu změněno chování (vzhledové i funkční) oproti běžnému textovému objektu znázorňujícího název aplikace. Přidání komponenty k textovému objektu umožní použití funkce *OnClick()*, kde je chování definováno skriptem.



Obrázek 14: Úvodní obrazovka aplikace

Skript definuje chování pro tlačítko Prohlídka (metoda *StartScene(string scene)*) a Ukončit (*QuitApp()*).

```
using UnityEngine;
using System.Collections;

public class mainMenu : MonoBehaviour {

    public void QuitApp() {
```

```
        Application.Quit();
    }

    public void StartScene(string scene) {
        Application.LoadLevel(scene);
    }
}
```

Než je možné spustit skriptem danou scénu (metoda *StartScene*), musí být přidána do seznamu úrovní požadovaných ve hře (*File → Build Settings...*). Zde zobrazené scény jsou označeny indexem, který může být využit jako parametr metody *LoadLevel*, která načítá scénu. V práci je parametrem řetězec (název scény), což je druhou možností při načítání scény.

Úvodní obrazovka s hlavním menu je vytvořena pomocí UI objektů z důvodu jednodušší a rychlejší editace hlavní obrazovky. Druhou možností tvorby menu je pomocí skriptu.

Areál MENDELU

Druhou a zároveň poslední scénou v aplikaci je prostředí areálu. V této scéně jsou uloženy všechny vytvořené modely, interakce s modelem a další. Součástí této scény je objekt znázorňující přímé osvětlení a *FPSController* simulující reálnou postavu místo statické kamery.

8.4 Interakce s modelem

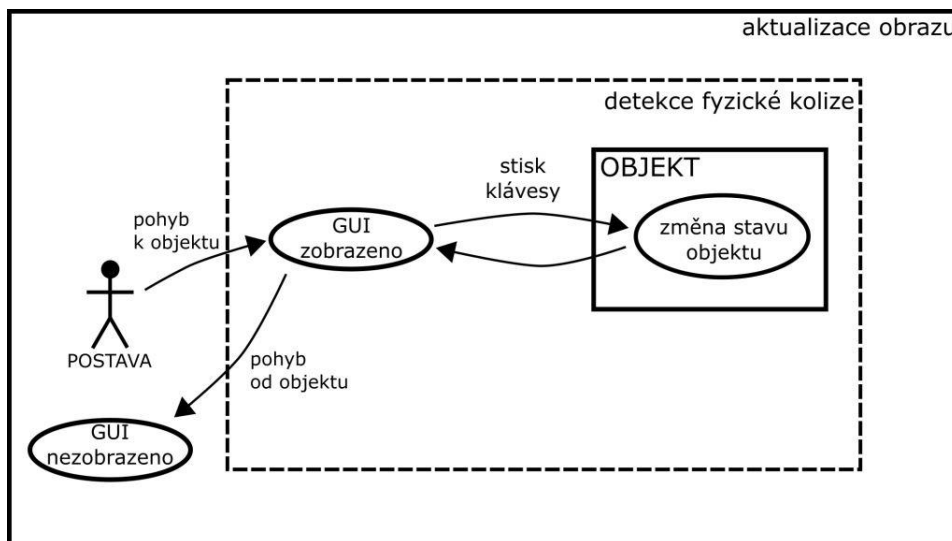
Interakce s modelem může být prováděna v *Unity* minimálně dvěma způsoby – animací nebo skriptem.

Animace, která může být spouštěna i skriptem, je použita u objektů s malým výskytem ve scéně (jedná se především o posuvné brány u vchodů do areálu).

Pohyb objektu spuštěný a vykonaný samotným skriptem je použit u dveří a branek. Zde vzniká riziko, že by byla animace nevhodně použita z důvodu velkého počtu různých kombinací pohybů u dveří, které mohou nastat (otevírání/zavírání směrem do/z budovy o různý úhel).

Výše uvedené schéma popisuje interakci postavy s modelem (zejména otevírání/zavírání dveří u budov). V případě vstupu postavy, označené jako *Player*, do prostoru, který pokrývá *collider* objektu je postavě zobrazena zpráva s možností interakce s objektem. Prováděná činnost s objektem je závislá na hodnotách proměnných, které uchovávají informace o stavu objektu. Při změně snímku a s tím spojené aktualizaci obrazovky dochází ke kontrole, zda-li nemá být změněn stav objektu (otevření nebo zavření dveří). Pokud postava z prostoru *collideru* vystoupí, dojde ke schování zprávy.

V rámci skriptu jsou některé proměnné veřejné. Jedná se o proměnné uchovávající daný objekt a úhel rotace objektu. Tím je zaručeno, že skript není vázaný na



Obrázek 15: Zjednodušené schéma popisující fungování skriptu na otevírání/zavírání dveří

konkrétní objekt a může být použit vícekrát. Hodnotu veřejné proměnné lze také upravit přímo v editoru *Unity*. Tohoto je využito u nastavení úhlu rotace objektu, jelikož každý objekt reprezentující dveře může být otevřen/zavřen o různý úhel.

Animace

Animace je v práci prováděna pomocí staršího systému – komponenty *Animation* (nově se animace provádí pomocí komponenty *Animator*). *Animator*, jehož použití usnadňuje humanoidní animací postav, obsahuje velkou řadu nástrojů, které ovšem způsobují vyšší křivku učení. Jelikož se v práci provádí pouze základní animace, tak je použita komponenta *Animation*, která je v *Unity* zachována pouze z důvodu zpětné kompatibility se staršími verzemi tohoto programu.

Z důvodu použití této komponenty je nutné jednotlivé klipy, z kterých je animace složena, označit jako *Legacy*. V komponentě *Animation* je uložen například seznam klipů, které jsou zpřístupněny u objektu, nebo výchozí klip a jiné.

Animace, jak již bylo dříve uvedeno, je spouštěna na základě skriptu. Každý animovaný objekt může mít svůj vlastní skript pro spuštění klipu (ve scéně se nachází pouze dva animované objekty). Tohle řešení není vhodné v případě nárůstu animovaných objektů ve scéně a s tím spojeného vytvoření velkého počtu skriptů s minimálními rozdíly ve zdrojovém kódu.

Lepším řešením, uplatněném v práci, je vytvoření pouze jediného skriptu, který spouští odpovídající klipy na základě jedinečného jména objektu ve scéně. Níže je uvedena ukázka procedury, která je součástí skriptu *gatewayAction*, pracující zmíněným řešením.

```
void changeDoorState () {
    string msgError = "Objekt se jménem nenalezen.";
```

```

if (closedGateway == true) {
    switch (gateway.name) {
        case "branaAQ": anim.CrossFade("gateway1Open");
            break;
        case "branaQ": anim.CrossFade("gateway2Open");
            break;
        default: Debug.Log(msgError);
            break;
    }
    closedGateway = false;
} else {
    switch (gateway.name) {
        case "branaAQ": anim.CrossFade("gateway1Close");
            break;
        case "branaQ": anim.CrossFade("gateway2Close");
            break;
        default: Debug.Log(msgError);
            break;
    }
    closedGateway = true;
}
}

```

Procedura *changeDoorState()* spouští příslušné klipy u objektů a mění hodnoty proměnných, kterými je určen stav objektů (v tomto případě vchodových bran do areálu). Načtení komponenty *Animation* do proměnné *anim* je provedeno v metodě *Start()*. Ta je jednou z inicializačních metod *Unity*, která je volána ještě před první aktualizací obrazu.

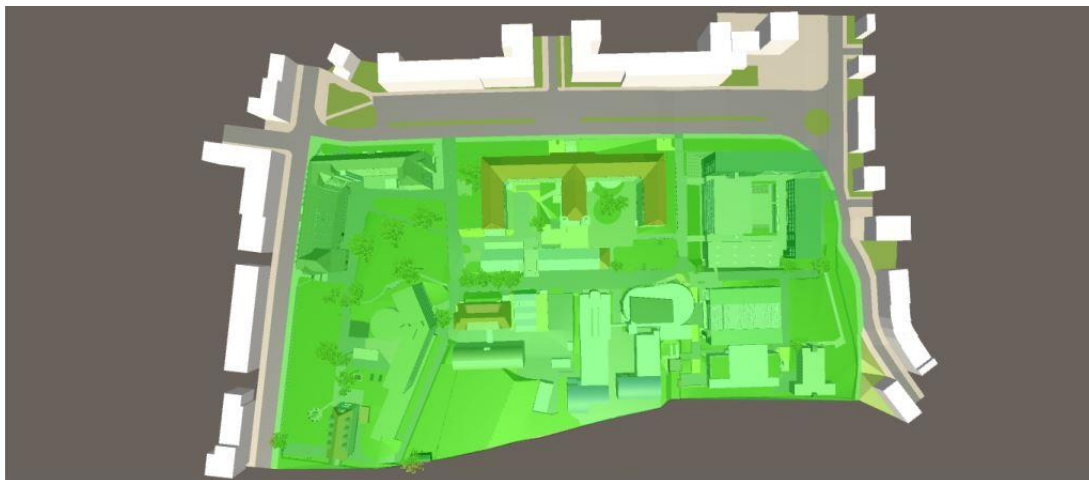
Na základě jména objektu (získané pomocí *gateway.name*) je přepínačem spuštěn odpovídající klip u objektu. V případě neexistujícího jména objektu je do konzole editoru *Unity* uloženo chybové hlášení. Jména jednotlivých objektů jsou vytvořena podle jejich polohy a nejbližše umístěných budov v areálu.

Nevýhodou tohoto řešení oproti prvně uvedenému, kdy každý animovaný objekt má svůj vlastní skript, je udržování aktuálnosti skriptu. V případě změny jména objektu v editoru musí být změna zaznamenána i do skriptu tak, aby skript odpovídal současnému stavu aplikace.

Omezení volnosti pohybu

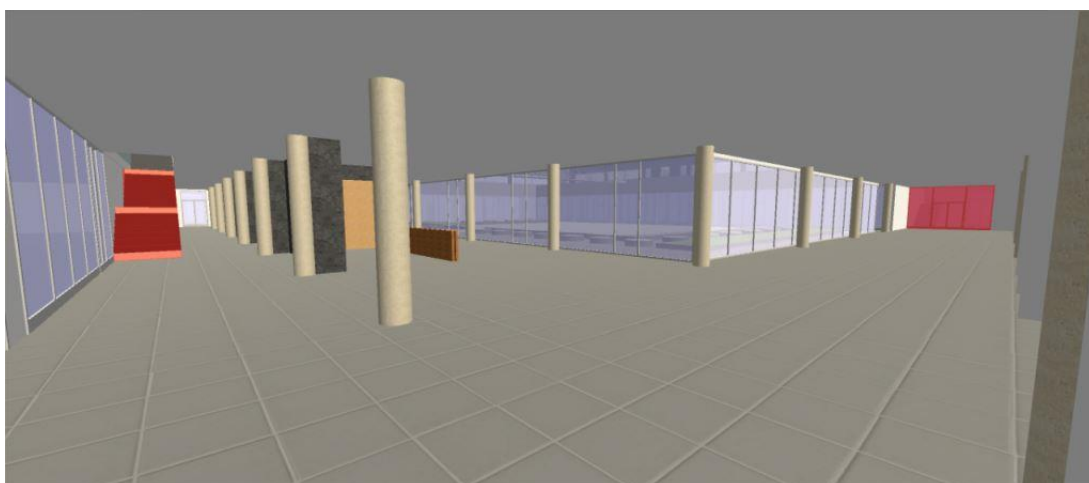
Pohyb uživatele v rámci aplikace je omezený na oblast celého areálu a přidruženého nejbližšího okolí (viz. obrázek 16).

V aplikaci je možné vstoupit i do budovy Q. Pohyb v budově je omezen na první nadložní patro, kde se nachází atrium, chodbu vedoucí ke studijnímu oddě-



Obrázek 16: Zobrazení areálu a možností volného pohybu (zeleně zaznačená oblast)

lení a východu z budovy, schody vedoucí do prvního podlažního patra směrem ke „skriptárně“ (viz obrázek 18). Do ostatních budov v areálu není umožněn přístup.

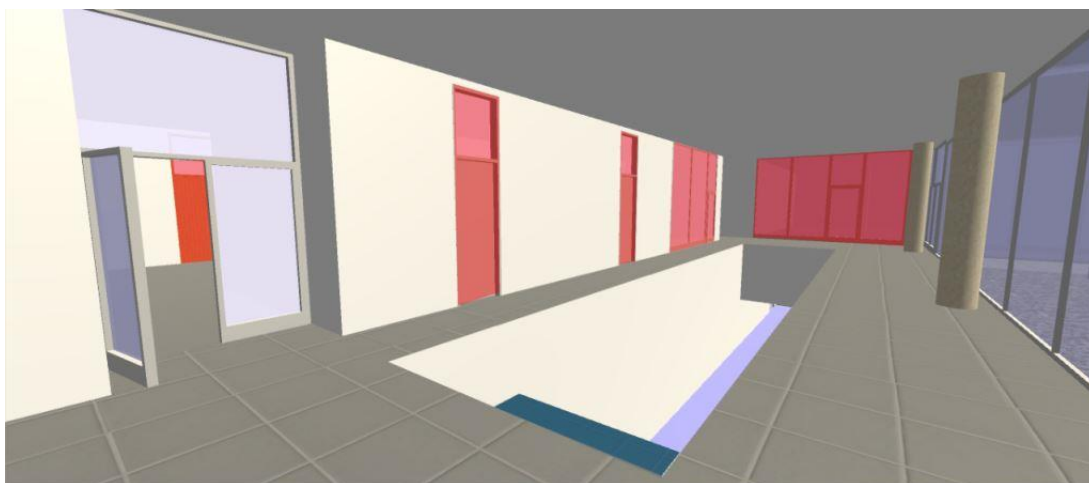


Obrázek 17: Interiér budovy Q po vstupu do budovy (červeně jsou zaznačena místa, kam není umožněn vstup)

V případě pohybu v budově nemůže být použit *Level design*, jelikož všechny přístupné prostory (chodby a místnosti) jsou většinou za prosklenými dveřmi. Dalším důvodem je malá rozsáhlost interiéru oproti skutečnosti. V případě rozšíření interiéru by jej bylo vhodné umístit již do samostatné scény (s ohledem na možnost pohledu zevnitř budovy ven).

8.5 Optimalizace výkonu aplikace

Z optimalizačních technik popsaných v kapitole Optimalizace grafického výkonu byly použity techniky *Level of Detail*, *bump mapping* (v *Unity* podporované normálové



Obrázek 18: Interiér budovy Q a schodiště vedoucí ke „skriptárně“ (červeně jsou zaznačena místa, kam není umožněn vstup)

mapování), statické a dynamické dávkování, *culling*. Dále v práci je použita například úprava rozlišení textur, vhodné použití komponenty *collider*, vytvoření nové úrovně kvality a jiné.

V práci bylo nutné „pročistit“ vytvořené modely, aby neobsahovaly přebytečné nebo duplicitní prvky (hrany, plochy, materiály). Pokaždé když jsou ve *SketchUp* použity textury nebo komponenty, jsou tyto informace uloženy v souboru (zvětšuje se velikost souboru). *SketchUp* umožňuje v základu odstranit nepoužité textury nebo komponenty (*Window* → *Model info* → *Statistics* → *Purge Unused*). K odstranění duplicitních ploch, hran a jiných prvků je již zapotřebí použít moduly z *Extension Warehouse*. V rámci programu *SketchUp* je možné stažení¹⁸ a instalace¹⁹ samotného modulu. Další možností k získání modulu je úložiště *Sketchucation*. Zde může být instalace a stažení²⁰ modulu, která není již v rámci programu *SketchUp*, odlišná oproti *Extension Warehouse*.

Optimalizace v rámci *Unity* spočívá převážně ve snížení počtu polygonů ve scéně (LOD, normálové mapování a jiné) a nároků na paměť zařízení (rozlišení textur). Plynulost obrazu na zobrazovacím zařízení do jisté míry ovlivňuje také nastavení kvality projektu (*QualitySetting*), proto je vytvořena nová úroveň určená pro *Oculus Rift*.

¹⁸Moduly je možné stáhnout přes hlavní nabídku programu *Window* → *Extension Warehouse*.

¹⁹Instalace je prováděna přes hlavní nabídku programu *Window* → *Preferences* → *Extension* → *Install Extension...*

²⁰Modul je stažen ze stránek <http://sketchucation.com/> a v případě souborového formátu .zip se změní tato přípona na příponu .rb. Následná instalace modulu je již stejná jako u modulu z *Extension Warehouse*. V případě stažení souboru .rb je nutné tento modul nakopírovat do příslušné složky (v operačním systému *Windows* do *C:/Users/uživatelskéJméno/AppData/Roaming/SketchUp/SketchUp 20xx/SketchUp/Plugins*) a restartovat program.

Odstranění duplicit v modelu

Modul *CleanUp*³, získaný z uložště *Extension Warehouse*, slouží k odstranění přebytečných prvků (hran, ploch, materiálů) a opravě modelů. Tyto činnosti vedou k jejich optimalizaci pro herní engine a snížení velikosti modelu. Modul pro svoji funkčnost vyžaduje knihovnu sdílených funkcí *TT_Lib*².

*CleanUp*³ umožňuje:

- opravu duplicitních názvů komponent v rámci modelu,
- odstranění skryté geometrie,
- odstranění duplicitních ploch,
- odstranění duplicitních hran,
- odstranění samostatných hran (s výjimkou hran na rovině řezu),
- opravení rozdělených hrany,
- sloučení identických materiálů,
- a další činnosti.

Modul se spouští přes hlavní nabídku *Extensions* → *CleanUp*³ → *Clean...*, kde je možné určit rozsah optimalizace. Druhou možností je spuštění modulu pomocí posledního spuštěného nastavení *Clean with Last Settings*.

Po spuštění modulu *CleanUp*³ je ve stavovém řádku okna programu *SketchUp* zobrazen průběh činnosti (v procentech), doba uplynulá od spuštění modulu a čas zbývající do dokončení. Doba kontroly modelu závisí na nastavení kontrolovaných částí a obsáhlosti modelu. Po skončení jsou uživateli zobrazeny výsledné statistické údaje o provedených činnostech.

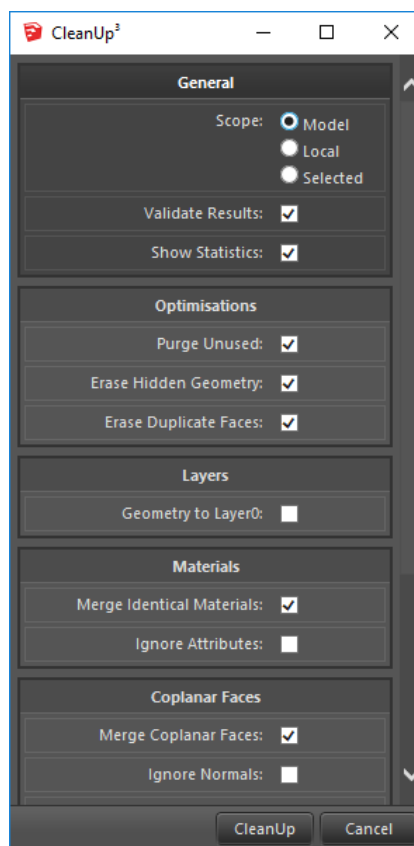
Změna orientace ploch

Modely jsou ve *SketchUpu* tvořeny plochami, které mají určitou orientaci. Zadní strana plochy je zobrazena šedo-modrou barvou, zatímco výchozí barva čelní plochy je bílá. Pokud jsou modely tvořeny pouze v rámci programu *SketchUp*, orientace plochy není příliš důležitá. V případě exportu do jiného programu (například v práci použitým *Unity*) může špatná orientace plochy způsobit chybné zobrazení (překrývající se dvě čelní plochy různých modelů aj.).

Unity, ale i jiné programy, zahrnují do zobrazení pouze plochy, které jsou orientovány čelně. Opačně orientované plochy jsou ignorovány – tím je výrazně snížen čas na výpočet z hlediska zobrazení modelu.

Modul *FrontFace*²¹ umožňuje změnu orientace plochy pomocí myši (události *mouseover*). Jedná se o rychlejší a jednodušší způsob než běžně používaný ve *Sket-*

²¹Modul je stažen z uložště *Sketchucation*.



Obrázek 19: Modul CleanUp³ a část jeho nastavení

chUpu, který spočívá v označení plochy (nebo více ploch) a následné změně orientace (*Reverse Faces*).

Rozlišení textur

Velikost textury nemá příliš velký vliv na výkon, ale především na paměť zařízení. Rozdíl mezi jednotlivými rozlišeními textur nemusí být na první pohled viditelný (viz obrázek 20), ale z hlediska velikosti souboru je velmi podstatný (viz tabulka 1). Po označení daného souboru s texturou v editoru *Unity* je možné provést změnu rozlišení v kartě *Inspector*.



Obrázek 20: Textura TER_stena v rozlišení 256 × 128, 512 × 256, 2048 × 1024

Tabulka 1: Závislost velikosti souboru na rozměru textury

Rozměr textury	Velikost souboru
2048 × 1024	1,3MB
1024 × 512	341,4 KB
512 × 256	85,5 KB
256 × 128	21,4 KB
128 × 64	5,4 KB
64 × 32	1,4 KB
32 × 16	360 B

Normálové mapování

Normálové mapování je novější a lepší typ techniky *bump mapping*. V *bump mapách* se uplatňují při tvorbě iluze hodnoty ve stupni šedi. Oproti tomu normálové mapy používají RGB informace, které přímo korespondují s osou ve 3D prostoru.

Normálové mapy lze vytvořit jednoduchým způsobem z textur přímo v *Unity* bez použití dalšího programu. Prvním krokem je vytvoření kopie textury, u které má být provedeno normálové mapování. Následně je u této kopie zvolen v kartě *Inspector* typ textury jako *Normal map*, místo původního typu *Texture*. Důležitým atributem je *Bumpiness*, který určuje úroveň výsledné nerovnosti povrchu. Tento atribut je zpřístupněn až po označení atributu *Create from Grayscale* v téže kartě.

Posledním krokem je vyhledání daného *shaderu*²² v materiálech a uložení vytvořené normálové mapy do položky *Normal Maps* v sekci *Main Maps*.

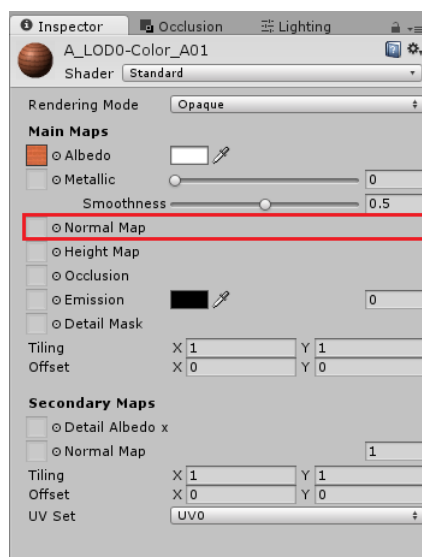
Vhodným řešením u normálových map je použití standardního typu *shaderu*. Další možností je použití typu *shaderu Legacy Shaders/Bumped Diffuse*. Tento typ je již ovšem zastaralý a je obsažen v *Unity* pouze kvůli zpětné kompatibilitě se staršími projekty.

Normálové mapy je vhodné použít pouze u důležitých objektů, i za předpokladu, že jejich uplatnění v aplikaci je méně náročné než tvorba detailů u modelů.

Prefabs

Prefab je označení pro kopii herního objektu. Použití *prefabů* u objektů (modře znázorněné objekty v hierarchii editoru *Unity*), které se v scéně vyskytují vícekrát, umožňuje jejich hromadnou editaci nebo přidání vlastnosti (komponenty) všem najednou. Při použití *prefabů* dochází ke snižování počtu dávek (*batches*) a tím se zvyšuje počet zobrazených snímků za vteřinu, které je třeba co nejvyšší kvůli plynulosti.

²²*Shader* je skript obsahující matematické výpočty a materiály, který se používá v *Unity* skrz materiály. *Shadery* v podstatě kombinují kód s dalšími parametry (například texturou). (Unity Technologies, 2015)



Obrázek 21: Umístění textury normálové mapy do standardního *shaderu* v kartě *Inspector*



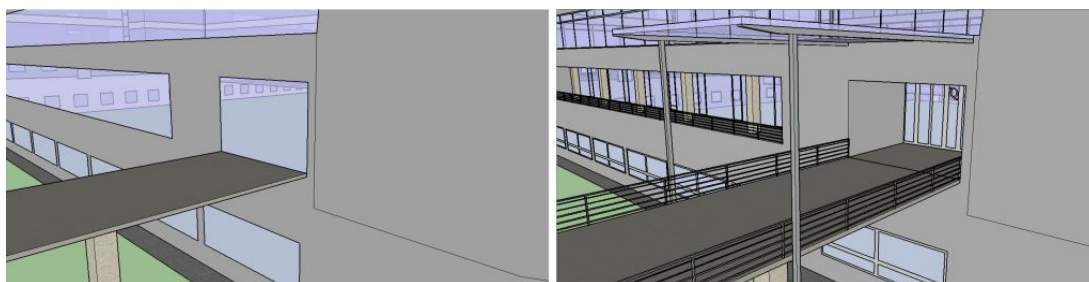
Obrázek 22: Zobrazení rozdílů mezi běžnou texturou a texturou s normálovou mapou (vpravo) u vybraného objektu

LOD budov

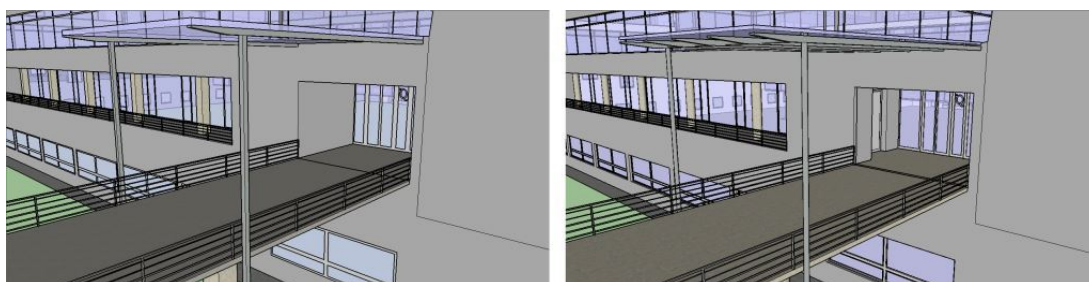
Základní model budov je označen jako LOD2 (obsahuje nejmenší počet polygonů), ze kterého vychází ostatní modely. Jedná se o základní schránku budovy takovou, aby ji bylo možné jasně identifikovat i z větší vzdálenosti.

LOD1 oproti LOD2 obsahuje ve většině modelů navíc zábradlí a sloupy, které mají obdélníkový tvar. Dále jsou již zobrazeny rámy oken a dveří namísto jednobarvných ploch, zjednodušené přístřešky a konkrétnější tvary budovy. U některých budov dochází i k nárůstu počtu textur.

U LOD0, tedy nejvyššího stupně detailu, se v některých případech mění tvar zábradlí (z hranatého na kulaté). Tvar rámy oken a dveří je již vyjádřen plasticky, přístřešky jsou zachyceny detailněji, schodiště dostávají reálný tvar (složený z jednotlivých schodů). Všechny budovy v této úrovni detailu obsahují i vnitřní stěny hlavní schránky budovy.



Obrázek 23: Budova Q zobrazená v LOD2 (vlevo) a LOD1



Obrázek 24: Budova Q zobrazená v LOD1 (vlevo) a LOD0

Jednotlivé úrovně detailu budovy jsou vloženy v *Unity* do komponenty *LOD Group*. Uvedená komponenta pracuje se zobrazením, a proto jsou *collidery* vytvořené u budovy aktivní ve všech jejich úrovních detailu. Skript je jediným řešením, jak zakázat *collidery* u méně početných modelů budov a tím snížit výkon na CPU.

Tabulka 2 znázorňuje rozdíl v počtu obsažených hran a ploch jednotlivých úrovní modelu vybrané budovy. Statistické údaje o počtech jsou zjištěny z programu *SketchUp Make*.

Tabulka 2: Závislost počtu hran na úrovni detailu

Úroveň detailu	Počet hran (ploch)
LOD2	2943 (679)
LOD1	18291 (5791)
LOD0	63898 (27382)

Hranice mezi přechodem z jedné úrovně do druhé je u budov individuální. Nevýhodou tohoto řešení je v případě změny nutná editace všech souvisejících modelů. V případě použití v aplikaci (pouze u budov) se předpokládá, že nebude docházet k častým změnám.

Collidery

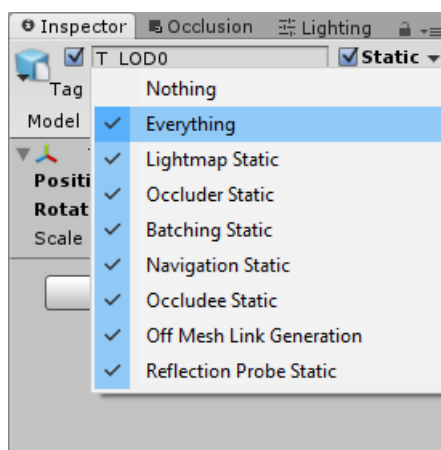
Collidery jsou jedny z nejdůležitějších vestavěných komponent *Unity*. Tato komponenta, která není viditelná, se používá pro definici tvaru pro fyzickou kolizi (nemusí

přesně odpovídat tvaru objektu). *Collidery* vychází z os modelu (případně jejich komponent) a lze u nich měnit pouze velikost, nikoliv provádět rotace.

V práci jsou využity především primitivní *collidery* (*box*, *sphere* a *capsule*), které jsou nenáročné na výkon CPU. Pouze v případě terénu je využit *mesh collider*, který je nejnáročnější, ale také nejpřesnější. Především z důvodu jeho náročnosti je vhodné se vyhnout jeho použití všude tam, kde je možné použít primitivní *collidery*.

Statické objekty

Vlastnost *Static* je vázána na herní objekty. V případě označení objektu jako statický je sděleno systémům, které jsou součástí *Unity* a řeší světlo scény, navigaci, dávkování a jiné, že se daný objekt nepohybuje. Tím pádem není potřeba provádět výpočty nad objektem a ušetřený čas může herní engine využít jiným způsobem. Statičnost objektu může být zohledněna v jednotlivých systémech nebo ve všech.



Obrázek 25: Nastavení vlastnosti *Static* u herního objektu

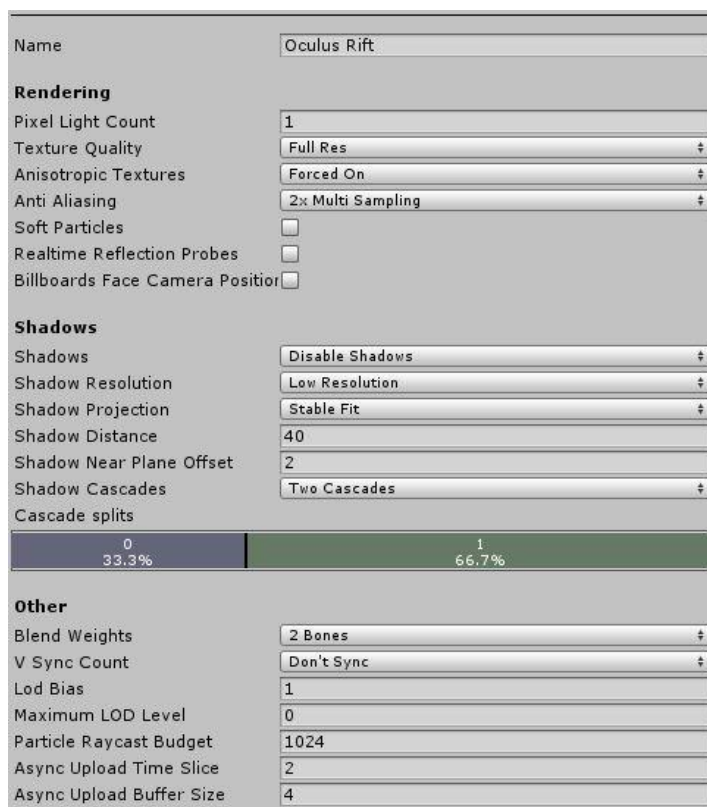
Z dříve uvedených optimalizačních technik je zde možné nastavit statické dávkování označením systému *Batching Static*. Tento systém je nastaven téměř u všech objektů, až na vchodové brány do areálu z důvodu použití animace a dveří, u kterých je prováděna rotace pomocí skriptu. Pokud objekt takto není označen, je u něj prováděno automaticky dynamické dávkování.

Další systémy, které je možné zohlednit, jsou *Occludee* a *Occluder Static*. Zmíněné systémy zahrnuje optimalizační technika *occlusion culling*. Ta způsobuje, že zakryté nebo uzavřené objekty jinými nejsou vykresleny a tím je snížen celkový počet polygonů scény. Nevýhoda této techniky již byla zmíněna dříve (nelze manipulovat s objekty). *Occluder Static* se nenastavuje u transparentních ploch modelů. *Frustum culling* je v *Unity* zapnut již ve výchozím nastavení. Tento typ techniky lze vypnout skriptem.

Nastavení úrovně kvality

Unity umožňuje nastavit různé úrovně grafické kvality, kterou se pokusí vykreslit. Obecně platí, že kvalita je na úrok snímkové frekvence obrazu. Nastavení kvality, které je možné upravit v hlavní nabídce (*Edit* → *Project Settings* → *Quality*), je rozděleno na dvě části. Horní část je tvořena maticí umožňující přiřadit zvolené platformě výchozí úrovně. Druhá část slouží pro úpravy konkrétní úrovně v oblasti vykreslování, stínů a dalšího nastavení. V *Unity* je šest předpřipravených úrovní, další je možné si přidat. (Unity Technologies, 2015)

Předpřipravené úrovně jsou zanechány pro možnost spuštění aplikace na běžném počítači. V herním engine *Unity* je vytvořena nová úroveň, nazvaná *Oculus Rift*, která má níže uvedené nastavení.



Obrázek 26: Nastavení úrovně kvality pro *Oculus Rift* v *Unity*

Pixel Light Count: *Pixel Light Count* udává maximální počet světel při použití *Forward* renderování. (Zhi Eng, 2015)

Se zvyšováním počtu světel ve scéně (hodnoty u *Pixel Light Count*) dochází ke snižování výkonu. Ve scéně se nachází pouze jediné světlo a proto je hodnota nastavena na jedničku.

Texture Quality: Toto nastavení vyjadřuje, zda-li se mají textury zobrazovat v maximálním nebo nižším rozlišení. Nižší rozlišení má menší režii. (Unity Technologies, 2015)

Rozměry textur jsou omezeny u jednotlivých textur na maximální velikost 512×512 (viz podkapitola Rozlišení textur). Proto je možné zanechat nejvyšší možné nastavení.

Antisotropic Textures: Nastavení umožňuje možnost použití (u všech textur nebo pouze u vybraných) nebo vypnutí anizotropní textury. (Unity Technologies, 2015)

Zapnutí anizotropních textur v projektu ovlivní výkon, ale rozdíl v kvalitě při použití nebo vypnutí je zanedbatelný. Proto jsou anizotropní textury v povoleny.

Anti Aliasing: *Anti Aliasing* zjemňuje okraje 3D prvků, aby výsledný obraz vypadal lépe. (Goldstone, 2009)

Čím vyšší úroveň je zvolena, tím vyšší jsou nároky na výkon grafické karty. Proto je zde vybrána nejúspornější možnost poskytovaná *Unity*, tzn. *2 Multi Sampling*.

Soft Particles: Uvedené nastavení pracuje s *Deferred Lighting*, které nepodporuje některé vykreslovací funkce (např. standardní *shader*, reflexy sondy aj.). Dále nemá žádnou skutečnou podporu pro anti aliasing, poloprůhledné objekty a má určité požadavky na grafickou kartu. (Unity Technologies, 2015)

Z výše uvedených důvodů je toto nastavení vypnuto, jelikož není jisté na jakém zařízení bude aplikace spouštěna.

Realtime Reflection Probes: Nastavení odrazu sondy v reálném čase. (Zhi Eng, 2015)

Ve scéně se nenachází žádný objekt, který využívá tuto vlastnost v reálném čase.

Billboards Face Camera Position: Žádný objekt, typu *billboard*²³, se ve scéně nevyskytuje, proto není potřeba nastavení této vlastnosti.

Shadows: Zobrazení stínů je vypnuté.

Jednotlivá nastavení v oblasti stínů není nutné popisovat, jelikož jsou stíny vypnuty. Jedná se o náročný prvek scény, který vedl ke značnému snížení výkonu.

Blend Weights: Uvedené nastavení se týká animace. *Unity technologies* doporučuje zvolené nastavení *2 Bones* jako vhodné z hlediska výkonu a vzhledu.

V Sync Count: Nastavení umožňující synchronizaci vykreslování s obnovovací frekvencí zobrazovacího zařízení. (Zhi Eng, 2015)

Aby se zabránilo případnému vizuálnímu artefaktu, označovaného jako „roztržení“²⁴, tak by měl být *V Sync Count* zapnut. Toto nastavení má však negativní vliv na FPS, proto je *V Sync* vypnut.

²³Billboard je obdélník s texturou, které jsou transformovány tak, že se vždy objeví paralelně k rovině pohledu. Hlavní využití je při nahrazení složitých 3D objektů (tráva, stromy aj.) dvourozměrnými.

²⁴V pravidelných intervalech, ve kterých je obraz na zobrazovacím zařízení aktualizován, nemusí být synchronizován displejem. Takže může nastat vizuální artefakt „roztržení“, kdy je vydán nový obraz, ale displej zobrazuje částečně obraz předchozí. (Unity Technologies, 2015)

Lod Bias: Tato vlastnost se určuje na základě situace, kdy se kamera nachází na hranici dvou úrovní, jaké detaily budou vykresleny. Hodnota mezi 0–1 zohledňuje méně detailů, větší hodnota než 1 naopak více detailů. (Unity Technologies, 2015)

V případě určeného zařízení je zapotřebí co nejméně detailů (polygonů), proto je hodnota nastavena na hodnotu jedna. Tato změna se projeví u již nastavené komponenty *LOD Group* u modelů, kterou je případně potřeba upravit.

Maximum LOD Level: Nejvyšší LOD (úroveň detailu), která bude v aplikaci použita. (Zhi Eng, 2015)

Hodnota je nastavena na nulu kvůli budovám, které mají nejvyšší LOD.

Particle Raycast Budget: Maximální počet vržených paprsků určených pro přiblížení částicového systému²⁵ kolizí. (Zhi Eng, 2015)

Žádné takové objekty nejsou ve scéně použity, proto je převzato nastavení z úrovně *Fastest* (hodnota čtyři).

Async Upload Time Slice: Množství času procesoru (uváděno v milisekundách pro rámeček) stráveného nad nahráváním textur na GPU. (Unity Technologies, 2015)

Z důvodu velkého počtu textur u objektů, zejména z důvodu nepoužití *atlasu textur*, je zachována úroveň nastavení *Fantastic*.

Async Upload Buffer Size: Velikost v MB pro vyrovnávací paměť asynchronního nahrávání. (Unity Technologies, 2015)

²⁵Částicový systém *Particle system* napodobuje subjekty jako jsou kapaliny, mraky nebo plameny generováním velkého počtu malých 2D obrázků. (Unity Technologies, 2015)

9 Diskuze

Na začátku vlastní práce byly modely tvořeny v programu *Rhinoceros*. Případný další rozvoj práce by byl komplikován kvůli dostupnosti programu. *Rhinoceros* není licencovaný jako *freeware* a pro užívání je nutné zakoupit licenci. Kvůli volné dostupnosti programu pro tvorbu 3D modelů byl výběr omezen na dva nejznámější – *SketchUp* nebo *Blender*. Jak již bylo uvedeno dříve, *Blender* je všestrannější, avšak méně uživatelsky přívětivý v porovnání se prvně jmenovaným. Ve scéně aplikace se nachází pouze poměrně jednoduché objekty složené z primitivních tvarů (budovy, lavičky, koše apod.) a proto byl vybrán *SketchUp*, kterým se modelování zjednodušilo.

Tvorbu modelů by výrazně urychlilo použití placené verze *SketchUp Pro*, která oproti bezplatné verzi, podporuje více formátů. V případě použití placené verze by mohly být plány (uložené v datovém formátu *dwg*) importovány přímo do modelovacího programu. Tím by odpadla nutnost použití programu *Draft IT*, kterým byly měřeny chybějící rozměry. Dalším přínosem by byla podpora výstupního formátu *fbx*, který je preferován u modelů v herním enginu.

U textur, které nebyly staženy jako bezešvé, bylo nutné přistoupit k úpravě kvůli vinětaci. Pro tuto úpravu byly vyzkoušeny dva grafické programy – *Gimp* a *Adobe Photoshop*. Bezplatný program *Gimp* nabízí automatický nástroj pro takový druh úpravy. Výstupní textura však nedosahovala takových výsledků, jako při ručních úpravách fotografií fasád budov v *Adobe Photoshopu*.

Uložiště modelů *3D Warehouse* od *SketchUp* obsahuje daleko více druhů stromů oproti *Asset Store* spadajícího pod *Unity*. Nevýhodou modelů získaných z prvně jmenované knihovny je jejich náročnost na technické prostředky počítače. U těchto modelů nepřineslo žádné zlepšení ani použití techniky LOD v herním enginu, kdy jedna úroveň obsahovala obrázek stromu a druhá samotný 3D model. Proto byly použity stromy stažené z *Asset Store*, bez ohledu na reálný druh stromu.

9.1 Zhodnocení obtížnosti vývoje

Na začátku práce byl použit herní engin *Unity 4.x* pro vývoj aplikace. S příchodem nové verze (*Unity 5*), kde došlo snadnější podpoře virtuální helmy, bylo s vývojem aplikace pokračováno v této verzi herního enginu. U starší verze (4.x) byla integrace podpory VR uskutečněna pomocí importu balíčku. Což je oproti stávajícímu řešení, kdy je podpora VR²⁶ přímo v samotném enginu, komplikovanější.

Volba modelovacího programu *SketchUp* sice urychlila vývoj z hlediska tvorby modelů, ale zároveň byl také vývoj aplikace ztížen z hlediska optimalizace. Jak již bylo dříve zmíněno, vytvořené modely obsahují duplicitní prvky.

Samotný vývoj aplikace poté ztěžuje poměrně dlouhý postup při tvorbě terénu v *ArcGIS*.

²⁶Podporu VR lze povolit v hlavní nabídce *Edit -> Project Settings -> Player*. V *Other Settings*, zaškrtnutím *checkboxu Virtual Reality Supported*.

Nejvíce časově náročnou část práce lze označit tvorbu modelů a optimalizaci aplikace pro plynulé zobrazení v *Oculus Rift*.

V práci by měly být zachovány programy *Unity* a *SketchUp*, které tvoří základ pro vývoj aplikace. *Unity 5* z důvodu snadného propojení s virtuální helmou a *SketchUp* pro jeho snadné ovládání a možnosti rozšíření o užitečné moduly.

I když byl v práci pro tvorbu terénu použit *ArcGIS*, vhodnějším řešením je tvorba pomocí *SketchUp* ve spolupráci s *Unity*. To z toho důvodu, že jsou volně dostupné a navíc jsou v prvně jmenovaném tvořeny i modely. Možnosti *ArcGISu* jsou využity minimálně vzhledem k tomu, co v něm může být vytvořeno. U druhého řešení dochází jednoduše k rozdělení terénu na více částí, které by mohlo vést za určitých okolností ke zvýšení výkonu (momentálně je terén tvořen jako jeden celek). Navíc by tím došlo ke snížení používaných programů pro vývoj aplikace pouze na *SketchUp* a *Unity*.

9.2 Doporučené rozšíření

Modely je nutné dále optimalizovat tak, aby v případném dalším rozvoji aplikace (přidání interiéru do dalších budov, rozšíření okolí a jiné) nedošlo k poklesu FPS u zobrazovacího zařízení a následnému trhání či chvění obrazu. Dalším optimalizačním nástrojem, v nějakém bezplatném nebo již používaném programu, by mohl být *atlas textur* a s tím spojené „zapečení“ světla v *Unity*.

Návrhem pro další rozšíření práce je zvýšení počtu prvků určených pro interakci s modelem (například výťah, kiosky, automaty na kávu a jiné). V současné době lze pohybovat pouze s několika málo objekty (především s dveřmi a vstupními bránami). Možnost interakce s více objekty by zvýšila zážitek ze samotné aplikace. Další možností, pro vylepšení stávajícího stavu aplikace, je vytvoření hry do jiné scény aplikace tak, aby měl uživatel možnost volby mezi prohlídkou a hrou. Hra by mohla oslovit i jiné osoby, než ty které mají určitý vztah k univerzitě. Na oficiálních stránkách *Unity* nebo i jiných je k nalezení velká řada návodů na různé druhy jednoduchých her.

V případném budoucím vývoji je také nutné se zaměřit na tvorbu hlavního menu a navigace v aplikaci. Momentálně je spuštěna hlavní obrazovka s menu ve zjednodušené formě. Po spuštění virtuální prohlídky již není možnost návratu zpět na hlavní obrazovku.

9.3 Využití řešení v podnikové praxi

S jistotou nelze říci, jaký bude další vývoj VR, jelikož již v devadesátých letech proběhl neúspěšný pokus o její využití. Ten se nezdařil především z důvodu ceny a provedení zobrazujícího zařízení (větší hmotnost helmy, množství kabelů a další). Navíc tehdejší výkonnost hardwaru nedosahovala takové úrovně jako nyní a to způsobovalo nevolnost uživatelů při jejím používání.

Nyní má virtuální realita daleko větší předpoklad k uplatnění se a průlomu do *mainstreamu*, jelikož dřívější nedostatky byly odstraněny. Průlom bude pravděpodobně přes herní průmysl, na který se momentálně VR zaměřuje. Pokud se v této oblasti prosadí, tak lze očekávat vývoj dalších generací zařízení pro VR a k nim rozšíření příslušného příslušenství, které je současně poměrně omezené. To by mohlo mít za následek i zvýšení potenciálu VR pro využití v podnikatelské sféře.

Důkazem zájmu o VR i z jiných oblastí než jen zábavního průmyslu (počítačové hry, filmy aj.) může být majitel truhlářství, který nás již kontaktoval. Ten by rád zobrazoval vytvořené modely svým zákazníkům v *Oculus Rift*. Pro tyto architektonické, urbanistické a další tvůrčí projekty je to velice nadějné, proto je potřeba zkoumat velmi pečlivě možnosti této technologie. Tato práce ukazuje pouze část potenciálu VR. Je zjevné, že pokud by byla připravena dostatečně uživatelsky přívětivá zobrazovací aplikace pro modely, už nyní by bylo možné nalézt v této oblasti zákazníky.

Z pohledu těchto zákazníků bude nejspíše nejvyšší investicí, mimo samotného hardwaru, tvorba modelů. Proto uvedené řešení bude nejvhodnější pro takové zákazníky, kteří tvorbu modelů mají jako součást své práce. Například zmíněný majitel truhlářství, který své modely pro zákazníky vytváří ve *SketchUpu*.

10 Závěr

Vzhledem k výše uvedeným skutečnostem lze konstatovat, že cíl práce, vytvoření virtuální prohlídky kampusu Mendelovy univerzity v Brně, byl splněn. Vytvořenou aplikaci je možné spustit jak na běžném počítači, tak ve virtuální helmě *Oculus Rift*. Na počítači lze aplikaci spustit bez problémů v nejnáročnější úrovni grafické kvality – *Fantastic*. U virtuální helmy dochází ke zkreslení výsledného obrazu kvůli nízkému rozlišení zobrazovacího zařízení. *Oculus Rift* má svoji vlastní úroveň kvality (nazvanou stejnojmenně), u které je upraveno nastavení tak, aby zobrazení obrazu bylo v zařízení pokud možno co nejplynulejší. Ale ani vlastní vytvořená úroveň a uplatněné optimalizační techniky nezajistily plynulé zobrazení. V určitých částech areálu při volném průchodu dochází k poklesu FPS pod minimální hranici (okolo 60 FPS), což má za důsledek chvění obrazu. Pravděpodobně k tomu dochází kvůli transparentním materiálům používaných u budov (okna, prosklené dveře) a plotu, jejichž použití je drahé z hlediska výkonu. Dalším problémem mohou být samotné modely budov, především budovy A a B, které obsahují velké množství oken a jsou vykresleny jako jeden celek. V případě doplnění základního interiéru do budov, by bylo možné budovy rozdělit na více částí (minimálně zadní a přední část budovy) a tím by se daleko více využila technika *occlusion culling*. Simulace aplikace byla prováděna na grafické stanici, která nebyla optimalizována na *Oculus Rift*. Pro práci s virtuální helmou je doporučována sestava s výkonnější grafickou kartou, která v době řešení práce nebyla k dispozici. Taková stanice by s největší pravděpodobností eliminovala výše zmíněné problémy.

Významným podílem na optimalizaci modelů z hlediska odstranění duplicitních a přebytečných částí, které mělo za důsledek snížení počtu polygonů a zvýšení FPS v *Unity*, byl modul programu *SketchUp – CleanUp*³. Za klíčové optimalizační techniky v *Unity* lze označit *frustum* a *occlusion culling*, statičnost modelů. *SketchUp* a *Unity* tvoří základ pro *workflow* na tvorbu aplikace pro *Oculus Rift*. Výhodou je jejich bezplatnost a nízká křivka učení. Ostatní použité programy, které jsou zmíněny v práci a nejsou volně dostupné, mají bezplatnou variantu. *ArcGis*, který byl použit pro tvorbu terénu, lze nahradit programem *SketchUp Make*. Tohle řešení by bylo vhodnější nebýt nalezeného omezení. V práci bylo totiž využito pouze malé procento potenciálu, kterým *ArcGIS* ve skutečnosti disponuje. *Gimp* může být alternativou za *Adobe Photoshop*, kterým byly upraveny textury.

Navržený postup (spojení programů *SketchUp* a *Unity*) je naprosto dostačující pro menší aplikace (scény s malým počtem jednodušších objektů). V případě komplexnější aplikace by bylo nutné modely upravit (snížit počet polygonů) nebo použít další optimalizační techniky. Tou by mohl být například *atlas textur* a následné „zapčení“ světla v *Unity*. Nejvhodnější řešením by bylo řešit optimalizaci v rámci již použitých programů – *SketchUp* a *Unity*, aby nemusel být použit další program.

11 Reference

- BURDEA, GRIGORE C A PHILIPPE COIFFET. *Virtual reality technology*. 2nd ed. Hoboken, N.J.: Wiley-interscience, 2003, xvi, 444 s. ISBN 0-471-36089-9.
- CARR, AUSTIN. *FORD TAPS OCULUS RIFT FOR FUTURE AUTOMOBILE DESIGNS: VIRTUAL REALITY WAS NEVER JUST FOR VIDEO GAME NERDS*. In: Fast Company: Business + Innovation [online]. 2014 [cit. 2016-01-24]. Dostupné z: <http://www.fastcompany.com/3024328/innovation-agents/ford-taps-oculus-rift-for-future-automobile-designs>.
- ESRI. *Esri: GIS Mapping Software, Solutions, Services, Map Apps, and Data* [online]. Esri, 2016 [cit. 2016-04-13]. Dostupné z: <http://www.esri.com/>.
- ESRI PRESS. *The ArcGis Book: Big Ideas About Applying Geography to Your World*. 1. California: Esri Press, 2015, 152 s. ISBN 1-589-48449-5.
- GILBERT, BEN. *6 things you should know before buying a virtual reality headset*. In: Tech Insider [online]. 2016 [cit. 2016-01-31]. Dostupné z: <http://www.techinsider.io/virtual-reality-headsets-2016-1>.
- GIS LOUNGE. *GIS File Formats*. GIS Lounge: Maps and GIS [online]. GIS Lounge, 2011 [cit. 2016-05-04]. Dostupné z: <https://www.gislounge.com/how-to-tell-gis-file-formats/>.
- GIS GEOGRAPHY. *27 Differences Between ArcGIS and QGIS: The Most Epic GIS Software Battle in GIS History* [online]. GIS Geography, 2016 [cit. 2016-04-27]. Dostupné z: <http://gisgeography.com/qgis-arcgis-differences/>.
- GIS GEOGRAPHY. *GIS Spatial Data Types: Vector vs Raster*. GIS Geography [online]. GIS Geography, 2016 [cit. 2016-04-28]. Dostupné z: <http://gisgeography.com/spatial-data-types-vector-raster/>.
- GIS GEOGRAPHY. *The Ultimate List of GIS Formats – Geospatial File Extensions* [online]. GIS Geography, 2016 [cit. 2016-04-28]. Dostupné z: <http://gisgeography.com/gis-formats/>.
- GOLDSTONE, WILL. *Unity game development essentials*. Birmingham, U.K.: Packt Pub., 2009. From technologies to solutions.
- GREGORY, JASON. *Game engine architecture*. Second edition. Boca Raton: CRC Press, 2014. ISBN 9781466560017.
- JASANI, TEJAS. *The top 10 engines that can help you make your game*. In: VentureBeat: Tech News That Matters [online]. 2014 [cit. 2016-01-24]. Dostupné z: <http://venturebeat.com/2014/08/20/the-top-10-engines-that-can-help-you-make-your-game/>.

- KRZEMINSKI, MAREK. *OpenGL Batch Rendering*. In: GameDev.net Game Development Community [online]. 2014 [cit. 2016-02-24]. Dostupné z: http://www.gamedev.net/page/resources/_/technical/opengl/opengl-batch-rendering-r3900.
- MASTERS, MARK. *Unity, Source 2, Unreal Engine 4, or CryENGINE: Which Game Engine Should I Choose?* In: Digital-Tutors is now Pluralsight: Tutorials training for creative pros > Pluralsight [online]. 2015 [cit. 2016-02-16]. Dostupné z: <http://blog.digitaltutors.com/unity-udk-cryengine-game-engine-choose/>.
- OCULUS VR. *Oculus Best Practices*. Oculus VR [online]. 2015, 1-40 [cit. 2016-01-24]. Dostupné z: <http://static.oculus.com/documentation/pdfs/intro-vr/latest/bp.pdf>.
- OCULUS VR. *Official Oculus Rift VR Forums: Oculus VR Forums* [online]. Oculus VR, 2015 [cit. 2016-01-13]. Dostupné z: <https://forums.oculus.com/>.
- O'REILLY MEDIA. *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*. 1. California: O'Reilly Media, 2015, 166 s. ISBN 1491922834.
- REDDIT. *Unity vs. Unreal: Which has the best VR performance?* In: Reddit: the front page of the internet [online]. 2015 [cit. 2016-02-01]. Dostupné z: https://www.reddit.com/r/oculus/comments/2qphsy/unity_vs_unreal_which_has_the_best_vr_performance/.
- RUSSELL, EDDIE. *Eliminate Texture Confusion: Bump, Normal and Displacement Maps*. In: Digital-Tutors is now Pluralsight: Tutorials training for creative pros > Pluralsight [online]. 2014 [cit. 2016-02-16]. Dostupné z: <http://blog.digitaltutors.com/bump-normal-and-displacement-maps/>.
- SEDLÁČEK, DAVID. *Optimalizace textur*. Praha, 2006. Diplomová. ČVUT. Vedoucí práce Doc. Ing. Jiří Žára, CSc..
- SHANKAR, DARSHAN. *12 performance tricks for optimizing VR apps in Unity 5*. In: Darshan Shankar [online]. 2015 [cit. 2016-02-27]. Dostupné z: <http://dshankar.svbtile.com/performance-optimization-for-vr-apps>.
- TRIMBLE NAVIGATION. *SketchUp: 3D for Everyone* [online]. Trimble Navigation, 2016 [cit. 2016-04-11]. Dostupné z: <http://www.sketchup.com/>.
- VAN OOSTEN, JEREMIAH. *Scripting in Unity*. In: 3D Game Engine Programming: Helping you build your dream game engine. [online]. 2012 [cit. 2016-01-30]. Dostupné z: <http://www.3dgep.com/scripting-in-unity-3-5/>.
- UNITY. *Why Unity 5.0 is STILL a good deal*. In: Unity: Game Engine [online]. 2014 [cit. 2016-01-24]. Dostupné z: <http://forum.unity3d.com/threads/why-unity-5-0-is-still-a-good-deal.235219/page-2#post-1562646>.

- UNITY TECHNOLOGIES. *Unity - Manual: Unity Manual* [online]. Unity Technologies, 2015 [cit. 2016-02-16]. Dostupné z: <http://docs.unity3d.com/>.
- UNITY TECHNOLOGIES. *Unity: Game Engine* [online]. Unity Technologies, 2016 [cit. 2016-04-11]. Dostupné z: <https://unity3d.com/>.
- UNREAL ENGINE TECHNOLOGY. *Unreal Engine 4 Documentation: Unreal Engine*. Unreal Engine Technology [online]. Epic Games, ©2004-2015 [cit. 2016-01-30]. Dostupné z: <https://docs.unrealengine.com/latest/INT/>.
- VARCHOLIK, PAUL. *Real-time 3D rendering with DirectX and HLSL: a practical guide to graphics programming*. Upper Saddle River, NJ: Addison-Wesley, 2014, xiv, 569 pages. ISBN 0321962729.
- VAUGHAN, WILLIAM. *Digital modeling*. Berkeley, CA: New Riders, 2012, xxi, 410 p. ISBN 0321700899.
- WAREABLE. *Wareable: Wearable Technology News and Reviews* [online]. Wareable, 2016 [cit. 2016-04-15]. Dostupné z: <http://www.wareable.com/>.
- ZHI ENG, LEE. *Building a Game with Unity and Blender*. 1. UK: Packt Publishing, 2015. ISBN 978-1-78528-214-0.

Přílohy

A Modely

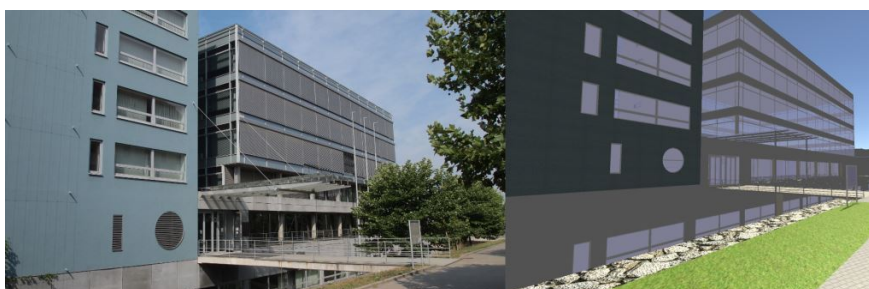
V příloze jsou zobrazeny vybrané modely v aplikaci a jejich srovnání s realitou.



Obrázek 27: Model budovy X v porování se skutečností



Obrázek 28: Model budovy M v porování s realitou



Obrázek 29: Model budovy Q v porování se skutečností



Obrázek 30: Model lavičky s košem a její skutečná podoba



Obrázek 31: Výsledný model areálu Mendelovy univerzity

B Scéna

Níže jsou zobrazeny ukázky z výsledné aplikace z různých pohledů.



Obrázek 32: Ukázka aplikace – pohled z budovy M



Obrázek 33: Ukázka aplikace – pohled od budovy X



Obrázek 34: Ukázka aplikace – pohled z ulice Lesnická



Obrázek 35: Ukázka aplikace – pohled od hlavního vchodu do areálu

C Obsah DVD

Součástí práce je přiložené DVD (na zadním obalu práce) jsou:

- ArcGIS
 - projekt v ArcMap
 - vektorová data ČR v *S-JTSK*
- Unity
 - projekt v Unity - obsahuje spouštěcí soubor aplikace, modely vytvořené ve SketchUpu, skripty v *C#*, textury aj.
 - balíček Dvornik-Split-Terrain - balíček pro import do *Unity* pro rozdělení terénu
- CleanUp³ - modul programu SketchUp pro „pročištění“ modelů
- TT_Lib² - požadavek pro používání modulu *CleanUp*³
- FrontFace - model programu SketchUp pro změnu orientace plochy