

Projekt NCK CAMEB

Metodika přípravy a optimalizace 3D dat s výstupem pro VR a AR

Metodika byla vytvořena s finanční podporou Technologické agentury České republiky, programu Národní centra kompetence 1, v rámci projektu Centrum pokročilých materiálů a efektivních budov CAMEB, registrační číslo projektu: TNo1000056.

Zhotovitel: Technická Univerzita v Liberci
Ústav pro nanomateriály, pokročilé
technologie a inovace

Kočí, Jan

Krčmařík, David

Zdvyhal, Milan



Obsah

Úvod	4
1. Požadavky.....	5
1.1. Virtuální realita (VR)	5
1.2. Augmentovaná realita (AR)	6
1.3. Real-time rendering.....	7
1.3.1. Hardwarové požadavky.....	7
1.3.2. Softwarové požadavky.....	8
2. Optimalizace	9
2.1. Tvorba modelu	9
2.2. Polycount	10
2.3. LODs	11
2.4. Draw calls	12
2.5. Culling	13
2.5.1. Frustum culling	13
2.5.2. Occlusion culling	13
2.5.3. Backface culling	13
2.5.4. Distance culling.....	14
2.6. Textury/Materiály	14
2.6.1. PBR.....	14
2.6.2. Optimalizace textur	14
2.6.3. Optimalizace materiálů.....	15
2.7. Light baking.....	15
3. Proces	16
3.1. Software.....	17
3.1.1. Unreal Engine	17
3.1.2. Twinmotion	17
3.2. Příprava dat	18
3.3. Převod dat.....	19
3.4. Performance.....	21
3.4.1. Nastavení kvality.....	22
3.4.2. Reflexe.....	22
3.5. Nasvícení.....	23
3.6. Materiály	23
3.6.1. Knihovna materiálů.....	23





3.6.2.	Vlastní materiály	24
3.6.3.	UV mapování	25
3.7.	Objekty.....	26
3.7.1.	Knihovna objektů.....	26
3.7.2.	Vlastní objekty	26
3.8.	Prostředí.....	27
3.8.1.	360° fotografie.....	27
3.8.2.	Terén	27
3.8.3.	OpenStreetMap	28
3.9.	Export.....	28
3.9.1.	Statické snímky.....	28
3.9.2.	Animace.....	29
3.9.3.	BIMmotion.....	29
4.	Metodika SKANSKA.....	30
5.	Možnosti rozšíření.....	32
5.1.	Konfigurátor	32
5.2.	Animování	32
5.3.	Export do UE4	33
5.4.	Cloudové řešení	33
	Použité zdroje	33





Úvod

V dnešní době moderních technologií se stále více setkáváme s novými možnostmi, které nám tyto technologie přinášejí. V oblasti vizualizací a tzv. "zážitkových" prezentací je to zejména v souvislosti s technologií virtuální, rozšířené a mixované reality, které umožňují posunout možnosti prezentace mnohem blíže koncovému uživateli. Tyto technologie navíc výrazně rozšiřují možnosti interakce, imersivních zážitků, individualizace, ale také přinášejí nové nástroje v procesech tvorby, designu, validace v různých fázích vývoje a konstrukce budov a objektů. Technologie, které v kombinaci s komunikačními kanály, sociálními sítěmi a globálními platformami, umožňují mimo jiné interakci a týmovou kolaboraci v reálném čase. V souvislosti s cloudovými službami jsou tyto technologie dalším krokem v evoluci digitalizace pracovních a výrobních procesů.

Ve chvíli, kdy chceme tyto moderní technologie aplikovat v oborech, jako je stavebnictví a architektura, musíme vyřešit problém diametrálního rozdílu v přístupu tvorby, definování požadavků a realizace modelů. Ve stavebnictví se využívá tzv. BIM standardu (Building Information Modelling), což je proces tvorby tzv. "digitálního dvojčete" objektu, který v sobě spojuje vlastní 3D data objektu s dalšími metadaty o reálném objektu (např. charakteristiky, materiálové informace, provozní a výrobní údaje, zabudovaná zařízení, a další). 3D data jsou modelována specifickým způsobem, vycházejícím zejména z požadavků na konstrukční a výrobní procesy. Tato data, přestože jsou dobře strukturovaná, jsou příliš komplexní a fragmentovaná a není jednoduché je přímým způsobem vizualizovat metodami renderování v reálném čase. Požadavky na vizualizace ve fotorealistické kvalitě jsou stále limitovány dostupnými hardwarovými prostředky současných technologií.

Tato metodika se zabývá problematikou optimalizace a konverze komplexních BIM dat a procesy přípravy, transformace a vizualizace těchto dat pokročilými metodami zobrazování. Cílem projektu je zejména popsání a zdokumentování procesu optimalizace a vysvětlení problematických bodů v přípravě dat pro renderování v reálném čase. Přestože jsou moderní softwary navrhovány tak, aby technologické procesy co nejvíce zautomatizovaly, klíčové manuální kroky a strategická rozhodnutí jsou stále na straně uživatele. Primárním zdrojem prostředků, který je vždy na začátku konstantní, je dostupný nebo cílový hardware, který hodláme použít. Od hardwarových požadavků se následně odvíjí softwarové požadavky a vhodná míra optimalizace, za účelem dosažení konkrétního výsledku (fotorealistický výstup, gamifikace, imersivní zážitek apod.). Optimalizačních metod existuje poměrně velké množství a každá řeší jinou problematiku procesu renderování obrazu, což je samo o sobě složitý výpočetní mechanismus. Je tedy na uživateli, které metody optimalizace zvolí, aby byl proces efektivní a výsledek se co nejvíce blížil očekávání. Klíčová je i míra jednotlivých typů optimalizace a celkové vyvážení prostředků mezi dostupným hardwarem a výslednou vizuální kvalitou.

Navrhované optimalizační metody vychází především z obecně platných metod a postupů při modelování prvků a objektů pro herní průmysl, který princip real-time renderování využívá po desítky let. Proto je v tomto projektu zdůvodněno použití herního engine a jeho dlouhodobé a perspektivní výhody. Vývojáři těchto engine si potenciál dlouhodobě uvědomují a adaptují tyto nástroje pro použití v konstrukčních průmyslech, kde je nutno pracovat s odlišným typem dat. Přesto ale umožňují uživateli využít možností a nástrojů, které jsou typické právě pro herní průmysl, a na které jsou nové generace uživatelů, vývojářů a zákazníků zvyklí

Součástí metodiky je zároveň demonstrace zpracování konkrétního multifunkčního projektu PORT 7 od developerské společnosti Skanska a.s. Ve spolupráci s touto společností byla dílčí část projektu PORT 7 zpracována pro potřeby prezentace ve virtuální realitě a je na něm demonstrován celý proces





zpracování a optimalizace dat, tvorba materiálů, scény, prostředí i export do různých výstupních formátů.



Klíčovou cílovou skupinou této metodiky je zejména uživatel s pokročilými znalostmi konstrukčních a modelovacích programů, nicméně se základními, či minimálními znalostmi v oblasti real-time vizualizací, herních enginů a virtuální a rozšířené reality.

1. Požadavky

Současné technologie umožňující zobrazení virtuální a rozšířené reality sebou nesou specifické požadavky na vstupní data. Tyto požadavky vycházejí především ze softwarových a hardwarových omezení současných technologií, které se navíc výrazně liší od typu použitého zobrazovacího zařízení. Jinou úroveň optimalizace virtuální scény vyžaduje aplikace renderovaná ve výkonné PC stanici a jinou aplikace renderovaná na mobilním zařízení. Zároveň je ale nutno zohlednit také fyziologický aspekt uživatele, kterého mají tyto zařízení přesvědčit o realističnosti zobrazované scény a objektů. Všeobecně známý jev v souvislosti s virtuální realitou je např. tzv. "motion sickness", který vzniká rozparem mezi tím, co uživatel vidí a co vnímá ostatními smysly a u části uživatelů tak vyvolává nevolnost. Správnou optimalizací jde tento jev minimalizovat např. tím, že budeme simulovat přirozené prostředí okolo uživatele, či snížíme možnost volného pohybu scénou (např. vyklánění se z oken, možnost teleportace).

1.1. Virtuální realita (VR)

Virtuální realita je možnost, jak vizualizovat a simulovat skutečný nebo fiktivní svět. Imerzivního vjemu je dosaženo pomocí stimulace dvou základních lidských vjemů a tím je zrak a sluch. Je ideálním prostředkem k tomu, aby možné přenést uživatele do budovy již v raném stádiu návrhu projektu a dát mu např. možnost verifikace designu z pohledu koncového uživatele. Mimo jiné přináší možnost interakce pomocí virtuálních ovladačů, případně technologie rozpoznávání gest rukou.

Z pohledu technologie je virtuální realita stereoskopicky renderovaný obraz virtuálního prostoru, ve kterém se uživatel nachází. Pomocí speciálních headsetů je tento obraz zobrazován uživateli pro každé oko zvlášť, což z pohledu hardwaru znamená, že scéna je vykreslována dvakrát, tedy počet drawcallů stoupne na dvojnásobnou hodnotu. Dalším





faktorem, který klade zvýšené požadavky na hardware zařízení, je rozlišení displeje, případně displejů headsetu. Běžně dostupná zařízení mají rozlišení displeje okolo 2K pro každé oko (přibližně 2 miliony pixelů), nicméně na trhu existují i varianty s rozlišením 8K (přibližně 4 miliony pixelů pro jedno oko), kde použití těchto headsetů nutně vyžaduje použití vysokého výpočetního výkonu.

Všeobecně se dají headsets pro virtuální realitu rozdělit do dvou kategorií, tedy headsets drátové a headsets bezdrátové. Drátové headsets nemají svůj vlastní výpočetní hardware, nýbrž jsou připojeny k výkonné pracovní stanici, která zprostředkovává výpočet obrazu dedikovaným hardwarem. Tento způsob nám umožní renderovat komplexnější a vizuálně náročnější scény. Naproti tomu bezdrátové headsets mají i svou vlastní výpočetní jednotku, nicméně jedná se ve své podstatě o mobilní zařízení. Jsme tak limitováni velikostí scény a je kladen vysoký důraz na úroveň optimalizace scény. Existuje ale i kombinace těchto dvou přístupů která využívá takzvaného streamování obrazu. Bezdrátový headset je propojen s výkonnou pracovní stanicí, které odesílá informace ze zabudovaných senzorů (např. pohyb uživatele v prostoru, interakce ovladačů atd.). Pracovní stanice tyto vstupy zpracovává renderuje obraz scény pro každé oko a odesílá tento streamovaný obraz zpět do bezdrátového headsetu. Je tak umožněno zobrazovat komplexní a vizuálně náročné scény i na bezdrátovém headsetu s malým výpočetním výkonem. Jsou zde však jistá omezení v podobě zvýšeného datového toku v síti a zpoždění přenosu dat, zejména v kombinaci náročných scén a vysokého rozlišení obrazu.

1.2. Augmentovaná realita (AR)

Rozšířená realita (neboli augmentovaná realita) je technologie, která umožní zobrazení vrstvy s digitálním obsahem nad reálným světem. Umožňuje tak interakci jak s reálnými objekty, tak s objekty digitálními. Uživatel tak má možnost v reálném prostředí simulovat digitální modely, vybavení prostoru, zobrazit virtuální informace o reálném objektu, či umožnit vzdálenou komunikaci, podporu nebo servisní úkon.

Z pohledu technologie a požadavků na zpracování se často jedná především o zobrazení dílčích částí scény, případně konkrétních objektů, nikoli celé scény. Prostřednictvím kamery nebo průhledných displejů jsou pak zasazeny do reálného prostředí.

Mobilní platformy jsou spojeny s vývojářskými nástroji typu Vuforia, ARCore (Android platforma) nebo ARKit (iOS platforma), které umožňují zásadní rozpoznávání prostředí, zejména pak ploch. To umožňuje vývojářům interagovat s prostředím a umístit digitální model v kontextu reálných ploch a objektů, což dává rozšířené realitě novou úroveň realističnosti. Podporu rozšířené reality skrze některý z těchto nástrojů má dnes již spousta moderních chytrých telefonů a tabletů, které jsou využitelné pro různé druhy informativních aplikací a zobrazování digitálních modelů.

Dalším typem zařízení, schopných zobrazování augmentované reality jsou holografické headsets, např. headset HoloLens od firmy Microsoft. Ten využívá systém holografických polopropustných displejů, na které je promítán digitální obsah. Toto zařízení má vlastní výpočetní jednotku s operačním systémem a řadí se také do kategorie mobilních zařízení, které navíc, díky hloubkovému senzoru, dokáže skenovat reálné prostředí, ve kterém se nachází.

Určitou evolucí rozšířené i virtuální reality je tzv. mixovaná realita (MX), která prolíná virtuální a skutečnou realitu dohromady a umožňuje skutečnou interakci virtuálních a reálných objektů.





Ve virtuální realitě si tak například můžete sáhnout na skutečný povrch, v rozšířené realitě zase může digitální model spouštět akci toho reálného.

Data pro augmentovanou realitu je, s ohledem na dostupný výpočetní výkon, nutno opět řešit s důrazem na vysokou úroveň optimalizace. Mobilní platformy navíc využívají jinou renderovací technologii a mají tak např. specifické požadavky na textury a shadery. Nutností je například použití textur ve formátu "power of two", což znamená v rozlišení násobku dvou (např. 128x128 px, 512x512 px, 1024x1024 px, ...).

V případě vývoje aplikace pro širokou veřejnost je nutno stanovit minimální podporovaný hardware, pro který je nutno obsah optimalizovat, aby byla umožněna distribuce na co nejvíce možných zařízení. Tím často ale přicházíme o možnost využití moderních technologií typu mobilní HDR, Vulkan API, vyšší úrovně vyhlazování (antialiasingu), apod.

1.3. Real-time rendering

Renderování v reálném čase neboli real-time rendering je metoda vykreslování obrazu s dostatečnou rychlostí tak, aby na něj mohl uživatel reagovat v reálném čase. Často se s tímto pojmem setkáme zejména v herním průmyslu, kde je plynulost zobrazovaného obsahu a pohybu klíčovým parametrem. Real-time renderování je zároveň klíčovým předpokladem pro vizualizace ve virtuální a rozšířené realitě.

Nicméně v dnešní době se tento požadavek a rychlé renderování vyžaduje i v dalších odvětvích průmyslu, které chtějí využít benefitů nových možností a ekonomických úspor, jenž sebou přináší. V reálném čase tak můžete měnit design a konstrukci budovy, konfigurovat interiéry, měnit materiály, animovat objekty a kamery a to vše, bez nutnosti dlouhého čekání na proces renderování a ve fotorealistické kvalitě.

1.3.1. Hardwarové požadavky

Oproti standardním metodám vizualizací, kdy je obraz počítán procesorem (CPU), je pro renderování využita grafická jednotka (GPU). Grafická jednotka zpracovává požadavky na renderování jednotlivých objektů a jejich materiálů a stará se o plynulost samotného zobrazení. Obsahuje zároveň vlastní paměťové moduly, jejichž kapacita výrazně ovlivňuje výkonnost GPU a její schopnost zobrazovat velké a komplexní scény s detailními texturami, za smysluplné snímkovací frekvence. Integrovaná grafická jádra většinou nemají dostatečný výpočetní výkon, a naopak pro virtuální realitu je nutnost mít dostatečně výkonnou grafickou jednotku. Vhodný typ grafické karty je většinou výrobcem označován jako "VR ready" produkt.

Ve srovnání s GPU, aplikace pro virtuální realitu nejsou tolik náročné na výpočetní výkon procesoru. Ten zajišťuje zejména přípravu dat pro vykreslení jednotlivých objektů. Vyšší výkon procesoru je potřeba např. při složitějších simulacích fyziky ve virtuální scéně.

Operační paměť (RAM) a její kapacita nám pak ovlivňuje velikost aktuálně zobrazené virtuální scény. U objemných a náročných projektů může být vyžadována, stejně jako u modelovacích softwarů, kapacita operační paměti v řádech desítek gigabajtů.





Projekt PORT 7, uvedený v dokumentaci, byl zpracován a testován na pracovní stanici s následující hardwarovou konfigurací:

CPU: Intel Core i9-7940X
GPU: NVidia GeForce GTX 1080Ti
RAM: 64 GB DDR4
OS: Windows 10

1.3.2. Softwarové požadavky

Softwarové požadavky úzce souvisí s těmi hardwarovými. Aby aplikace s virtuální nebo rozšířenou realitou poskytovala přesvědčivý prožitek, musí být její obsah náležitě připraven a zpracován.

Snímkovací frekvence (FPS – frames per second) nám udává počet snímků za vteřinu, kterou je grafická karta schopna v danou chvíli zpracovávat. Ta je ovlivněna různými faktory, jako např. složitost scény, počet objektů, materiálů, výkonnost grafické jednotky, běžící aplikace na pozadí apod. Empirickým zjištěním se došlo v minulosti k závěru, že hraniční hodnota pro lidské oko je přibližně 24 snímků za vteřinu. Tato hodnota je udávána jako konstanta, do které je lidské oko schopno rozeznat jednotlivé snímky. Jakákoliv vyšší hodnota už je vnímána jako plynulý obraz. Nicméně požadavky na aplikace pro virtuální realitu jsou minimálně 90 snímků za vteřinu. K tomuto požadavku přispěl zejména faktor pohybu uživatele v prostoru. Pokud uživatel otáčí hlavou a snímkovací frekvence je příliš nízká, obraz se nestíhá vykreslovat, dochází k trhání obrazu, což může vést k již dříve zmiňovanému motion sickness (nevolnost z pohybu). Navíc dochází také k efektu tzv. "ghosting", kdy obraz je dopočítáván automaticky do obnovovací frekvence displeje a vytváří nepříjemné artefakty. Právě hodnota 90 FPS zajišťuje plynulý vjem obrazu. Hodnota FPS nám slouží jako vhodný ukazatel výkonu celého systému, který je ale v čase proměnný. Je nutné jej ladit pro nejnáročnější část scény, která může být renderována.

Dalším důležitým předpokladem pro rychlé renderování je hodnota tzv. "drawcalls", která je výrazně závislá na grafické jednotce a určuje nám počet objektů, které je grafická karta schopna odbavit, pro dosažení požadované hodnoty FPS.

V neposlední řadě je důležitým faktorem také hodnota počtu polygonů ve scéně a zejména pak počet aktuálně renderovaných polygonů. Vstupní data jsou ve většině případů ve formátu CAD, tedy data modelovaná na principu matematických vzorců a operací, která jsou sice absolutně přesná, nicméně příliš složitá k výpočtu simulací světla. Proto je nutné tato data za pomoci tessellace transformovat z povrchových modelů na objemové (polygonální) modely, které jsou za pomoci interpolace vertexů a polygonů dostatečně zjednodušené. Úroveň interpolace, neboli kvalita tessellace, nám pak udává výsledný počet polygonů (polycount) jednotlivých modelů.





Orientační požadavky pro dílčí zařízení tak mohou být následující:

	FPS	max. polycount	max. drawcall
AR zařízení	30	kolem 100 tis.	cca 100
VR (mobilní)	90	nižší stovky tis.	cca 100
VR (PC)	90	nižší jednotky mil.	cca 1000

Mezi softwarové požadavky můžeme rovněž zahrnout předpoklad fotorealistické kvality obrazu, které nám vedle plynulého běhu aplikace dodává dojem reálného prostředí a objektu a umocňuje nám vizuální vjem. Fotorealistická kvalita je dosažitelná kombinací kvalitních materiálů, detailních objektů a fyzikálně korektního nasvícení scény.

2. Optimalizace

Dosažení nejlepšího možného výsledku je vždy o správném a efektivní balancování jednotlivých optimalizačních metod. Na jedné straně rovnice máme dostupný hardware a vstupní data nutná k vizualizaci, na straně druhé chceme dosáhnout co nejrealističtějšího vizuálního vjemu. Optimalizačních metod existuje několik a každá řeší pouze část komplexního problému. Všechny ale mají za úkol snížit zátěž hardwarových prostředků, aby mohla být zvýšena kvalita výstupu.

Navrhované optimalizační metody vycházejí z analýzy zpracovaného projektu a jsou všeobecně platné při optimalizaci 3D dat pro virtuální a rozšířenou realitu.

2.1. Tvorba modelu

Část optimalizačních metod je uplatnitelná již při tvorbě modelu, případně na konci konstrukčního procesu projektu. Model je nutno připravit dle obecně platných pravidel pro 3D modelování, které zajistí zejména snadnější transfer a zpracování modelu v cílovém softwaru.

Jedním ze základních problémů při tvorbě modelu je špatné měřítko objektu. Vždy je nutné volit reálné měřítko a zároveň je nutné věnovat pozornost při přivedu jednotek z metrické na imperiální soustavu a naopak. Zároveň má spousta softwarů svou vlastní výchozí jednotku rozměru. V případě převodu mezi CAD a polygonálními softwary se často stává, že modely jsou deseti až sto násobkem původního modelu. Je vhodné si vždy nastavit jednotnou hodnotu co celou dobu procesu optimalizace, a to i v případě modelování nových objektů pro potřeby vizualizace. Úprava měřítka v cílové aplikaci může vést k problémům např. při sdružování a sjednocování objektů dohromady.

Dalším prvkem, který se řadí k problematickým je výchozí bod a orientace os. Zejména u velkých konstrukčních objektů, které jsou vytrženy z větších celků se stává, že počátek souřadnicového systému je výrazně mimo oblast objektu. Problém nastává zejména tehdy, přidáváme-li do projektu i objekty, které pocházejí z jiných projektů, či knihoven a které mají

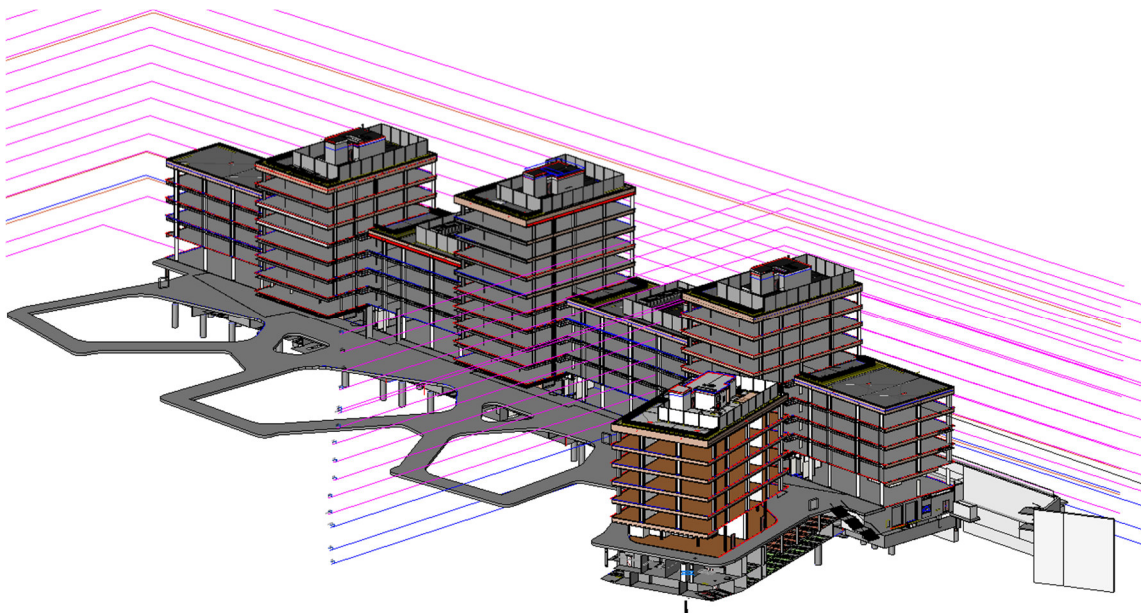




svůj výchozí bod jinde. Snadno můžeme tyto objekty přehlédnout, zapomenout na ně, ale přesto mohou být ve výsledku renderovány a ukrajovat tak z výkonu grafické karty.

V rámci převodu mezi 3D softwary nastává i problém s výchozí rotací souřadnicového systému. Některé softwary mají osu Z orientovanou směrem vzhůru, jiné využívají naopak osu Y, jako vertikální. Tento problém může způsobit rotaci importovaných objektů o 90° ve smyslu jedné z os a v krajních případech může způsobit i deformaci normál jednotlivých polygonů a tím i vyhlazování povrchu objektu.

Zejména pro real-time rendering jsou důležité normálové vektory polygonů. Jedná se o vektory kolmé na plochu polygonu, které nám určují, mimo jiné, viditelnost polygonu, tím pádem informaci o tom, zda jej zahrnout do procesu renderování, či nikoliv. Jedná se o jednu z automatických optimalizačních metod. Z pohledu modelování a transformace objektu je ale nutné si hlídat správnou orientaci normálových vektorů jednotlivých ploch. Vektory, které směřují ve směru kamery, jsou pro kameru viditelné a naopak. Může se tedy stát, že část objektu pro nás bude neviditelná, případně uvidíme skrz něj jeho vzdálenější plochu. V takové případě je nutné vhodným nástrojem obrátit normálové vektory problémových polygonů na opačnou stranu.



Dalším kritickým faktorem je zbytečná granule objektů ve scéně. V situacích, kdy jsou jednotlivé objekty zbytečně rozděleny na dílčí pod objekty, navyšujeme operace na grafické kartě potřebné k vyrenderování všech objektů. Objekty, které spolu souvisejí (např. rámy oken), tvoří nějaký celek (např. nohy stolu a deska), či mají stejný materiál (např. listy zeleně) mohou být spojeny do jednoho objektu.

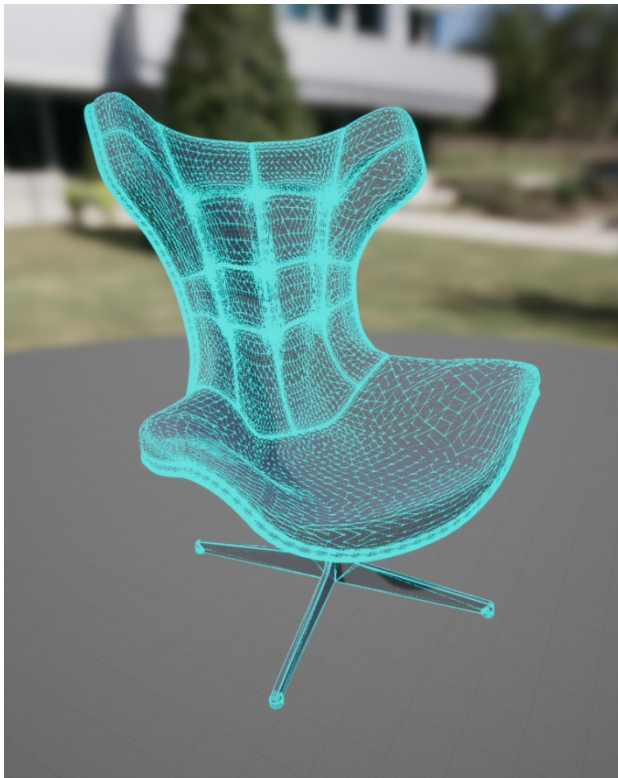
2.2. Polycount

Jak již bylo zmíněno, polycount určuje počet polygonů ve scéně, případně při procesu renderování. Chceme-li dosáhnout kvalitně teselovaného objektu (zejména u organických tvarovaných objektů), musíme vynaložit dostatečný počet interpolovaných bodů. Kritická místa tak vznikají zejména v oblasti konstrukčních rádiusů, zaoblených hran, organicky tvarovaných objektů, složitých Boolean operací (matematické sčítání a odečítání





geometrických objektů), či zeleně. Nicméně počet polygonů má zásadní vliv na výpočetní výkon, neboť pro každý kamerou viditelný polygon ve scéně je počítáno chování světla.



Z pohledu tvorby polygonálních modelů rozlišujeme dva různé přístupy, tedy tzv. "highpoly" modeling a "lowpoly" modeling. V první případě se jedná o modely o vyšším počtu polygonů, vhodný zejména pro statické vizualizace či real-time rendering na výkonných pracovních stanicích. U lowpoly modelů se bavíme o výrazně nižších hodnotách polygonů a tyto modely jsou vhodné především pro virtuální a rozšířenou realitu a mobilní, či webové platformy. Nicméně neexistuje žádná určující hodnota, či standardizovaný proces, který by striktně rozlišoval hraniční hodnotu mezi lowpoly a highpoly modelem. Navíc s rostoucím výpočetním výkonem je tento mezník těžko definovatelný v čase.

V případě, že chcete využít již hotové, či zakoupené objekty do vaší scény pro

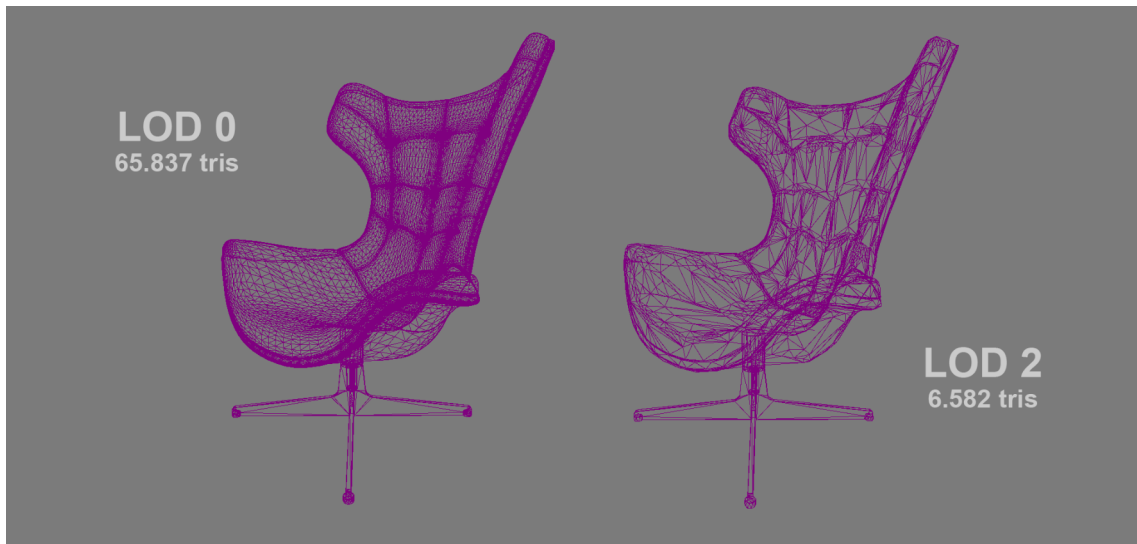
virtuální realitu, vyvarujte se objektům a knihovnám, které jsou určeny pro fotorealistické highres vizualizace. Tyto modely jsou vytvořeny pro statické renderování klasickým procesem vizualizací, často obsahují miliony polygonů, nevhodné textury a jsou fragmentovány na desítky a stovky dílčích objektů. Několik takových objektů může snadno snížit hodnotu FPS o desítky jednotek.

Některé 3D softwary umožňují zobrazení hodnoty vertex count, tedy počet vertexů ve scéně, který, v případě že je výrazně vyšší než polycount, může indikovat fragmentovaný povrch objektu, špatně spojené plochy, příliš členěné UV mapování, či špatné vyhlazování. Zároveň je třeba říci, že hodnota Polycount a Vertex count se může lišit ve výchozím a koncovém softwaru, kdy z pohledu 3D modelovacího procesu se za polygon považuje i n-úhelník (4 a více vrcholů), nicméně z pohledu renderingu se pracuje čistě s trojúhelníkovými polygony.

2.3. LODs

Metoda tzv. "LODů" (level of detail) je hojně využívána jako optimalizační nástroj v herním průmyslu. Jedná se o princip, kdy pro daný model máte k dispozici vytvořeno několik úrovní kvality meshe (polygonální plochy) a renderovací software tyto úrovně přepíná v závislosti na vzdálenosti kamery od daného modelu. Cílem této metody je maximálně snížit počet polygonů u objektů, které jsou kameře vzdáleny natolik, že je zbytečné je vykreslovat vysokým počtem polygonů. Originální model s nejvyšším počtem polygonů, který je renderován nejbližše kameře, se označuje jako LOD₀. Následuje LOD₁, LOD₂, atd., dle požadované úrovně optimalizace. Rozdíl v počtu polygonů mezi prvním a posledním LODem může být až 99 % polygonů.





Smyslem celé redukce polygonů je držet vysokou hustotu polygonů v blízkosti kamery a s postupnou vzdáleností tuto hodnotu snižovat tak, aby detaily byly koncentrovány ve smysluplné oblasti.

Pro jednotlivé LODy se pouze snižuje počet polygonů. Materiál, UV mapování, kolize a další parametry zůstávají zachovány pro všechny LODy jednotné.

Tato metoda je mimo jiné ideální pro velké open space interiéry, kdy u vzdálených, přesto viditelných, objektů můžeme snadno snížit polycount. Některé moderní renderovací enginy dokážou LODy generovat automatickým procesem.

2.4. Draw calls

Draw call je jedním ze základních orientačních hodnot, podle kterých bychom měli optimalizovat virtuální scénu. V procesu renderování se jedná o soubor informací, popisující objekt, textury, shadery, stav atd., který je formulován procesorem a předán k výpočtu grafické jednotce. Úroveň paralelních procesů, které dokáže grafická karta spočítat pro zpracování jednoho obrazu, nám udává právě hodnota Draw calls. Jednoduše řečeno, jedná se o příkaz, který CPU posílá na GPU k vykreslení objektu s konkrétním materiálem.

Draw call je vyžadován pro každý objekt s materiálem a v případě, že daný objekt obsahuje více materiálů (multi-material object), je počet Draw calls multiplikován.

Vliv na počet Draw calls má tedy nejen počet objektů ve scéně, ale především počet materiálů na jednotlivých objektech. Zjednodušeně tedy platí následující pravidlo:

- 1 objekt + 1 materiál = 1 Draw call
- 1 objekt + 5 materiálů = 5 Draw callů
- 5 objektů + 1 materiál = 5 Draw callů

Z výše uvedeného pravidla tedy vyplývá, že je vhodné pro optimalizaci Draw callů spojit objekty se stejným materiálem dohromady. To může vést k výrazné úspoře renderovacích prostředků, nicméně je nutné opět balancovat poměr mezi vykreslovaným počtem polygonů a Draw calls.





Maximální počet Draw calls je ovlivňován výpočetním výkonem GPU. Nicméně obecně platné hodnoty mohou být následující:

	Draw calls per frame
VR (mobilní)	40-160
VR (PC)	500-1200
PC	500-5000

2.5. Culling

Jedná se o soubor automatických optimalizačních metod většiny renderovacích enginů, u kterých je ovšem vhodné pochopit, jak fungují. Nevhodně připravená data může tyto metody výrazně ovlivnit.

Jedná se souhrnně o metody, které na základě konkrétních pravidel, vyřazují objekty nebo jejich části, z procesu renderování a tím snižují počet operací o ty, které by musela grafická karta počítat zbytečně.

2.5.1. Frustum culling

Jde se o metodu, která vyřazuje objekty, nacházející se mimo zorné pole kamery, pro kterou je daný obraz zpracováván. Tato metoda funguje automaticky, nicméně každý objekt je obalem do primitivu, jehož viditelnost kamerou je následně počítána. V případě velkých objektů s nestandardním tvarem to znamená, že daný objekt může být počítán, přestože není vidět ani jeden jeho polygon, ale jeho primitiv stále viditelný je.

2.5.2. Occlusion culling

Tato metoda vyřazuje z procesu renderování objekty, které jsou v zákrytu za jinými objekty a nejsou tak kamerou viditelné. Typickým příkladem může být např. vybavení vedlejší místnosti, kterou přes zeď nevidíte. Tuto metodu je nutné u některých renderovacích enginů nastavit pro jednotlivé objekty zvlášť a celou scénu před počítat (v závislosti na použité metodě Occlusion culling).

2.5.3. Backface culling

Pomocí této metody jsou z procesu renderování vyřazeny pouze polygony, které nejsou pro danou kameru viditelné, tzn. polygony, jejichž směrový vektor směřuje směrem od kamery. Typickým příkladem může být např. jednoduchá obdélníková zeď, u níž vždy vidíte maximálně 2 strany a u ostatních stran je tak možné odstranit polygony. Tato metoda je opět plně automatická, nicméně v případě použití oboustranných materiálů (two-sided material) je u těchto polygonů tato možnost vypnuta.





2.5.4. Distance culling

Jedná se o metodu, která vyřazuje objekty na základě vzdálenosti od kamery. Tuto hodnotu minimální a maximální vzdálenosti od kamery je možné nastavit, ale opět záleží na velikosti objektů. U malých objektů je možné nastavit tuto hodnotu na relativně nízké úrovni (např. tužka na psacím stole již z deseti metrů nemusí být viditelná), u velkých objektů jsou smysluplnější vyšší hodnoty. Zároveň je vhodné Distance culling kombinovat s metodou LODů, aby se nestalo, že objekt bude vyřazen z procesu renderování přesto, že má k dispozici ještě další úroveň LODů k zobrazení.

2.6. Textury/Materiály

Zásadním problémem materiálů a materiálových knihoven je významná absence průmyslových standardů napříč všemi softwary. V praxi to znamená, že nelze v procesu konverze modelových dat transformovat i materiálovou informaci se všemi jejími parametry. Často se stává, že při exportu jste schopni přenést pouze základní informaci o barvě a odrazivosti materiálu, případně jeho průhlednosti. To klade vysoké nároky na tvorbu nových duplicitních materiálů, či materiálových knihoven v softwaru, ve kterém se následně dělají vizualizace nebo scény pro virtuální a rozšířenou realitu.

Pokud je to možné, doporučuje se popisovat materiály zejména pomocí textur, které jsou schopny přenést informace o jednotlivých kanálech materiálů napříč různými modelovacími a renderovacími softwary.

2.6.1. PBR

Pro real-time rendering a fyzikálně založené renderovací enginy je stále častější použití tzv. PBR materiálu (Physically based rendering materials), což je metoda, kdy se materiál nesnaží napodobit svou reálnou referenci, nýbrž se jí snaží fyzikálně korektně nasimulovat. Materiál je tedy popsán hodnotami jako je barva, odrazivost, drsnost povrchu apod. Právě adaptace fyzikálně korektního renderování výrazně zvyšuje vizuální kvalitu renderovaného obrazu.



2.6.2. Optimalizace textur

Vstupní data textur by měly být vždy ve formátu "power of two" (POT textures), tedy rozlišení násobku dvou (256x256, 1024x2048, 128x512 px,...), protože grafická jednotka s nimi dokáže efektivně pracovat při výpočtech. Renderovací enginy těmto texturám zároveň umožňují další optimalizační metody, které pro NPOT textury ("non power of





two textures“) nejsou z výpočetního hlediska možné. Mezi tyto metody patří například “Mipmaps”, což je jednoduše řečeno alternativa LODů pro textury. Software si opět vytvoří zmenšené alternativy pro vstupní texturu a poté je v materiálu přepíná dle vzdálenosti objektu od kamery. Drobný objekt, vzdálený několik metrů od kamery tak nemusí mít 2K texturu, ale stejný efekt se dosáhne použitím textury např. 128x128 px. Tím dojde k uvolnění paměti grafické karty snížení výpočetní náročnosti. Navíc použitím velkých textur s periodickým vzorem na vzdálených objektech vzniká vlivem antialiasingu tzv. “Moiré” efekt. Je to rušivý optický efekt, který způsobuje vzorkování rastru a právě snížení rozlišení textury s patřičnou úrovní komprese můžete tento efekt částečně eliminovat.

Dále platí všeobecné pravidlo nepoužívání příliš velkých textur, které nikdy nebudou zobrazeny v nativním rozlišení. Např. zbytečné používání 4K textur na objektech, které stejně vlivem rozlišení používaného zařízení uvidíme jen ve fullHD. Velké textury by se měly použít jen tam, kde je potřeba viditelná velká hustota pixelů textury na objektu.

Optimalizace velikosti textur je na místě také u dílčích kanálů materiálu. Je vždy na zvážení, zda je opravdu potřeba použití velkých textur i u kanálů typu odrazivost (Specular), či drsnost (Roughness).

2.6.3. Optimalizace materiálů

Primární optimalizace ohledně materiálů se děje právě s texturami, nicméně i zde platí několik zásad. Materiály by se neměly vytvářet příliš komplexní. Složité materiály s násobnými operacemi pro výpočet mohou příliš snížit výkon.

Efektívni materiály a transparentní kanály s IOR (index refraxe) větším než 1 se naopak hůře vykreslují ve virtuální realitě z důvodu stereoskopického renderování pro každé oko zvlášť.

Výraznou částí optimalizace materiálů je používání materiálových instancí. Ty sdílejí parametry s materiálem, pro který byly vytvořeny a který jim dává možnost měnit některé klíčové proměnné (např. textury, barvu, hodnoty, atd.). Výhodou instancí materiálů je fakt, že je není potřeba znovu kompilovat při změně parametru, a to ani v runtime.

2.7. Light baking

U scén pro real-time rendering rozlišujeme světlo na statické a dynamické. Dynamická světla je nutná použít ve scénách, ve kterých jsou animované objekty, světla, případně je-li počítáno se zásadními změnami scény za běhu aplikace. Dynamická světla jsou velice náročná na výpočet, proto se nedoporučuje mít ve scéně mnoho světel. Pro každý viditelný polygon je počítána simulace světla a dopadá-li na daný polygon více paprsků různých světel, tyto operace se multiplikuji. Výhodou je naopak relativně jednoduché nastavení a příprava nasvícení scény. Možné je také měnit osvětlení za běhu, případně měnit světelné scénáře (denní a noční vizualizace).

Statické nasvícení přináší výraznou úsporu výpočetních prostředků tím, že umožňuje simulaci světla před počítat v přípravné fázi projektu. Veškeré informace o šíření světla scénou (globální iluminace) se zapisují do speciálních textur každého objektu (tzv. lightmap) s patřičným





rozlišením a tyto textury jsou pak následně aplikovány na samotné objekty. Vzhledem k tomu, že čas pro výpočet světla ve scéně je libovolně dlouhý, je dostatek prostoru pro algoritmy simulující fyzikálně přesné šíření světla scénou. Je tak možné dosáhnout kvality klasických statických vizualizací.



Tento proces je ovšem vykoupen poměrně náročnou přípravou scény i jednotlivých objektů. Nejen, že do procesu vstupuje čas, potřebný pro renderování světla ve scéně, který může v komplexních scénách dosahovat jednotek hodin, ale časově náročné je především samotné mapování. Statické objekty, pro které má být světlo počítáno, musejí mít specificky připraven kanál UV mapování pro lightmapu. První kanál UV mapy je vždy použit pro mapování textur materiálu, další kanál je možné využít pro uložení informace o světle. Důležité je, aby každý polygon na UV mapě měl svůj vlastní prostor (pixely) pro zápis informace. Ve chvíli, kdy se jednotlivé polygony překrývají, dochází k přepisu informací při výpočtu a vznikají fragmenty stínů a světla. Fragmenty také způsobují nízké rozlišení lightmap, čili nízká hustota pixelů na ploše polygonu.

Vzhledem k tomu, že světlo ve scéně je již před počítáno, není možné s objekty pohybovat nebo měnit jejich transformace, protože stín, který objekt ve scéně vrhá na své okolí je zapsán do textur právě tohoto okolí. Mezi další nevýhody patří také vyšší nároky na paměť GPU, neboť software pracuje s větším počtem textur (lightmap) navíc.

3. Proces

Existují různé přístupy a řešení, jak zpracovat 3D modelová data pro vizualizaci ve virtuální realitě. Mezi ně můžeme řadit i moderní herní enginey, které získávají na stále rostoucí popularitě. Nejedná se o software, které jsou primárně orientovány na vizualizace v architektuře a stavitelství, přesto se do tohoto odvětví úspěšně posouvají. Jejich podstatná výhoda je dlouholetý vývoj v oblasti real-time renderování, průkopnictví v oblasti virtuální reality a možnosti gamifikace, které přinášejí další prvky interakce a využití. Z prostředí herního průmyslu také přejímáme většinu optimalizačních metod a nástrojů, původně vyvinutých pro potřeby her a zábavných aplikací, které mají tyto enginey již integrovány. Mezi nejpobulárnější herní enginey se dnes řadí především Unity a Unreal Engine.





3.1. Software

3.1.1. Unreal Engine

Unreal engine je herní engine vyvíjený americkou softwarovou společností Epic Games (www.unrealengine.com). Aktuálně se jedná o 4 generaci tohoto herního enginu, která se nachází ve své finální vývojové fázi. Poslední verzi této generace by měla být verze 4.26 a v roce 2021 by měla být nahrazena novou generací s označením Unreal Engine 5. Tato společnost již není orientována pouze na herní průmysl, ale její nově vytvořená divize Enterprise se orientuje právě na využití technologií herního prostředí i v ostatních odvětvích průmyslu jako je např. automobilový průmysl, letectví, strojný průmysl, stavebnictví a architektura, filmový a zábavní průmysl, a další. Unreal Engine obsahuje spoustu nástrojů pro tvorbu nejrůznějších aplikací, virtuálních prohlídek, vizualizací, či video prezentací.

Dokáže pracovat s různými druhy vstupních dat a pomocí integrovaného pluginu Datasmith dokáže zpracovávat i konstrukční data v křivkové podobě. Umožňuje tak jednoduše a rychle transformovat a přenášet celé scény, včetně materiálu, světel, kamer a animací, se kterými pak můžeme dále pracovat a vytvářet například interaktivní prohlídku scény ve virtuální realitě. Zároveň se jedná o multiplatformní engine, který dokáže exportovat výstupy pro různá zařízení a různé platformy, což umožní například kolaborativní komunikaci mezi uživatelem na PC zařízení a zároveň uživatelem na mobilní telefonu.

Jedná se o poměrně robustní nástroj, jehož možnosti fotorealistického výstupu jsou vykoupeny relativně vysokými nároky na znalosti a zkušenosti práce s těmito nástroji a postupy.

3.1.2. Twinmotion

Twinmotion je jeden z produktů Epic Games, postavený právě na jádře Unreal Enginu (www.unrealengine.com/twinmotion). Tento software je naopak přímo cílený na profesionály z oblasti stavebnictví a architektury, kterým dává nástroj pro vizualizaci projektu v reálném čase, včetně možnosti výstupu pro virtuální realitu. Jedná se tedy o 3D vizualizační software, který je díky možnosti real-time technologie renderování schopen produkovat vysoce kvalitní obrázky, panoramata, videa, či interaktivní procházky. Twinmotion má možnost přímého propojení s konstrukčními programy, jako jsou Archicad, Autodesk Revit, Rhinoceros nebo SketchUp a pomocí integrovaného pluginu dokáže přímo z konstrukčního projektu vytvořit a připojit nový projekt ve Twinmotionu. Data mezi projekty jsou pak synchronizovaná a uživatel má tak stále možnost editovat konstrukční data a změny promítnout okamžitě do vizualizace ve Twinmotionu.

Dokáže zpracovávat i BIM metadata a využít například zeměpisné souřadnice k tomu, aby umístil geometrii projektu do kontextu reálného prostředí, využitím projektu OpenStreetMap.

Uživatelské rozhraní je velice jednoduché a intuitivní a ovládání lze přizpůsobit nastavení, na které je uživatel zvyklý např. Revitu. Práce v tomto programu je tak velice jednoduchá na naučení a oproti Unreal Enginu jsou požadavky na vstupní znalost a zkušenosti





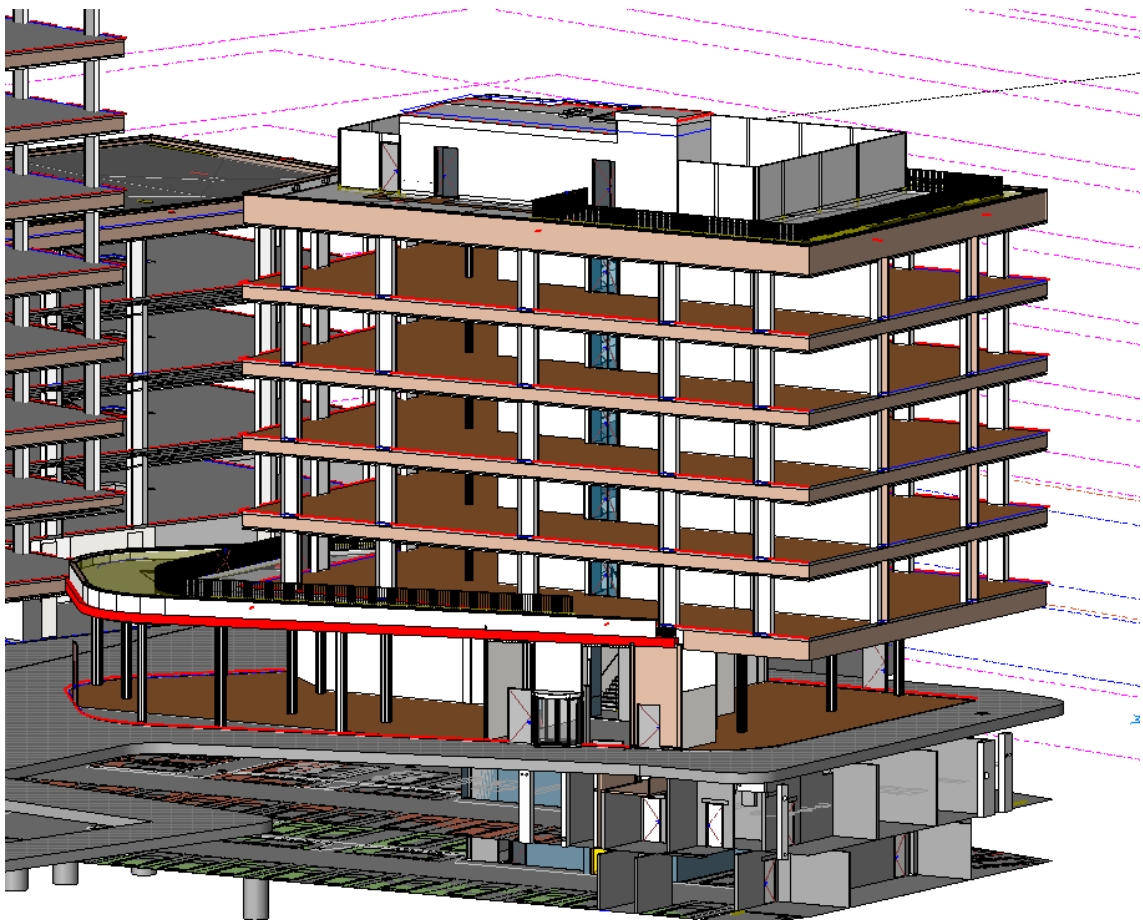
uživatele minimální. Zároveň si ale zachovává možnost vysoké vizuální kvality výstupních formátů.

Právě z důvodu efektivity, rychlého procesu a nižších nároků na trénink a edukaci uživatelů je software doporučen jako vhodný pro daný typ úloh a je v něm zpracováván ukázkový projekt PORT 7.

3.2. Příprava dat

Vstupní data projektu PORT 7 jsou vypracována v konstrukčním softwaru Revit. Jedná se o projekt multifunkčního komplexu několika budov. Cílem je příprava dat a vytvoření digitální scény pro virtuální realitu. Pro demonstrativní účely bylo zvoleno jedno administrativní patro ve věži budovy E, o celkové rozloze přibližně 490 m².

Po vydefinování prostoru, který bude ve virtuální scéně pro uživatele přístupný, bylo z geometrie odstraněno vše, co není vizuálně relevantní, abychom dle navržených optimalizačních metod snížili zatížení výpočetních prostředků a zvýšili snímkovací frekvenci. Objekty jako vnitřní sloupky, šachty a dveře v ostatních patrech a budovách, vybavení střeš a podzemních garáží, vzduchotechnika, generátory, různá zábradlí apod. Cílem je eliminovat zbytečnou geometrii a drawcally ve scéně. Tyto operace je vhodnější dělat ještě v konstrukčním softwaru, v duplicitním projektu určeném pro virtuální realitu. Virtuální realita je poměrně hodně náročná na výpočetní výkon a bylo by těžko realizovatelné použít veškerou geometrii a hierarchii v originálním stavu za rozumného stavu FPS.





Projekt byl rozdělen na dvě hlavní části, tedy na část aktivního interiéru (celé patro) a část viditelného exteriéru. Vzhledem k tomu, že Revit umožňuje automaticky slučovat objekty při importu na základě konkrétních parametrů, bylo možné sloučit objekty exteriéru do jednoho, zatímco interiéru byl slučován po skupinách objektů. Existují tři možnosti, jak tuto optimalizaci připravit.

- Můžeme projekt v Revitu rozdělit na 2 podprojekty, připravit interiéru a exteriéru odděleně a poté oba projekty sloučit v Revit, každý se specifickým nastavením importu.
- Můžeme data ponechat v jednom projektu a využít slučování objektů dle skupin. V takové případě ale musíme mít na paměti to, že exteriéry a interiéru musí mít rozdílné skupiny i v případě, že se jedná o charakterově stejné objekty (okna, stěny, podlahy,...) se stejnými materiály.
- Můžeme k editaci využít jiný polygonální 3D modelovací software a využít např. formát FBX k převodu dat.

Původní scéna bez optimalizace obsahovala několik desítek tisíc objektů, po přípravě dat pro import je to pouze několik desítek objektů. Tato optimalizace nám umožní využít drawcall na vybavení interiéru scény.

Model bylo ještě poté potřeba zbavit duplicitních polygonů, které způsobují tzv. "mesh overlapping". Jedná se o efekt, kdy 2 a více polygonů se kompletně překrývají nebo jsou od sebe pod hodnotou tolerance vzdálenosti při vykreslování. V takové případě dochází k probíhání ploch při renderování, patrně zejména ve chvíli, kdy plochy mají různé materiály. Vlivem modelování k tomu dochází např. ve chvíli, kdy plochy stěn na sebe těsně dosedají a tato styčná plocha je pro uživatele viditelná.

3.3. Převod dat

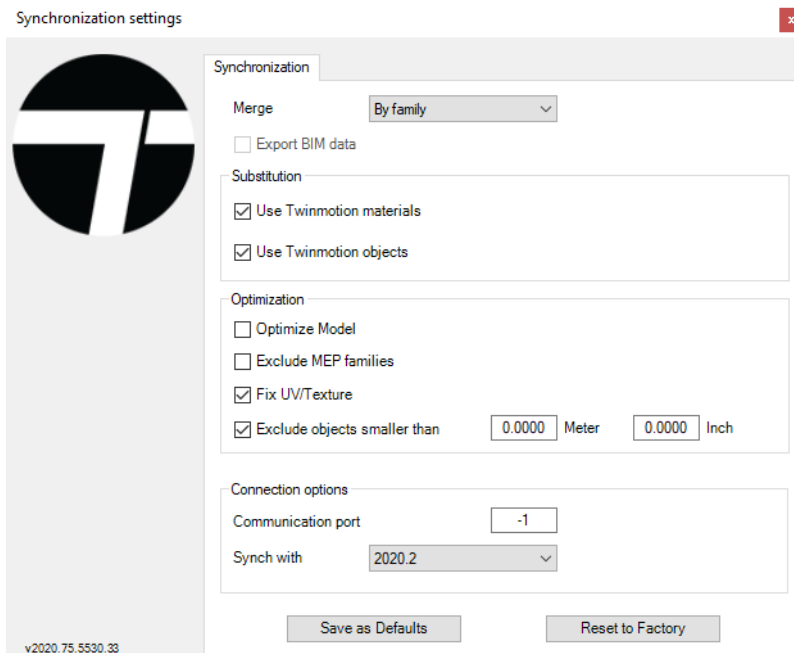
Twinmotion podporuje několik různých typů formátů geometrie k importu. Preferovány jsou formáty FBX, OBJ, SKP a C4D, které jsou schopny přenést kromě geometrie také hierarchii a informaci o materiálech. Dalšími podporovanými formáty jsou např. 3DS, STL, VRML, XYZ, DAE a další. Ty už jsou schopny přenést pouze jeden objekt s jedním materiálem.

Pro Revit nicméně existuje Direct Link plugin, který je dostupný online (www.unrealengine.com/twinmotion/plugins). Po instalaci pluginu se v programu Revit objeví nová záložka Twinmotion, která umožňuje dva způsoby transferu dat. Prvním je klasický export, který vytvoří dle parametrů nastavení exportní soubor, jež je poté možné otevřít v Twinmotionu, případně distribuovat dalším uživatelům.

Druhou možností je přímé propojení a synchronizace projektu Revitu s Twinmotionem.

Revit v tomto režimu funguje pouze jako vizualizační nástroj a některé jeho funkce, týkající se geometrie, jsou omezené. Např. nelze objekty transformovat, či přesouvat, aby nedošlo k asynchronnímu editování scény. Tyto operace je tedy nutné provádět přímo v Revitu.





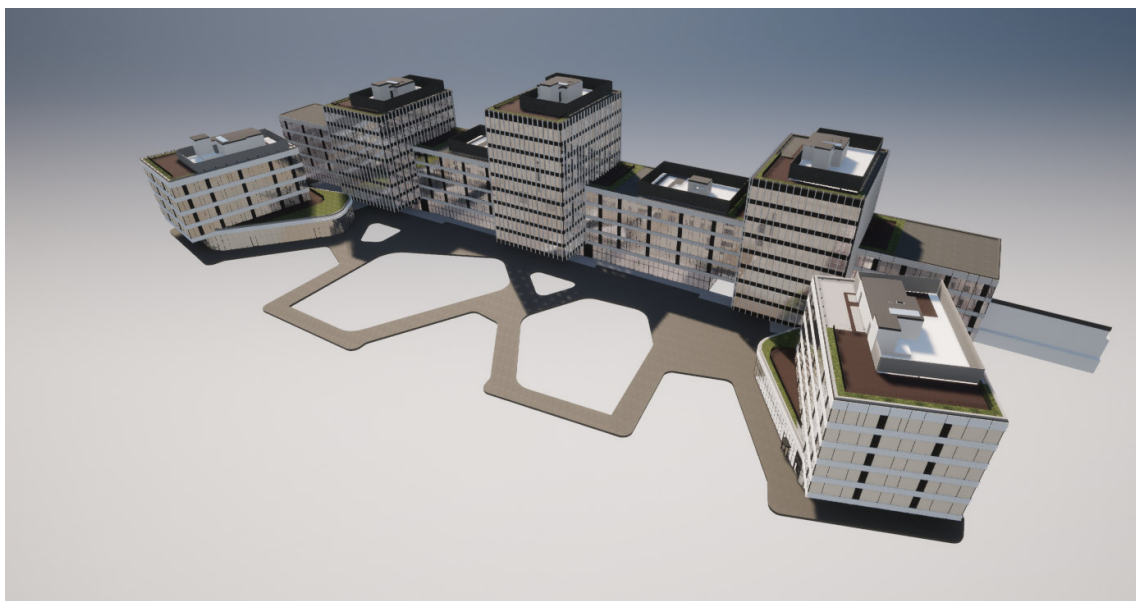
Při exportu dat je na výběr několik možností nastavení. Twinmotion nabízí výběr importovaných objektů dle viditelnosti, čímž můžeme omezit export jen na nutné objekty, aniž bychom museli ze scény cokoli odstranit. Dále máme možnost exportování textur, které byly v projektu Revitu použity. Ty mohou být exportovány do složky nebo integrovány do exportního formátu, který je podporuje. Zde ale platí informace o optimalizaci textur a jejich rozlišení v násobku dvou. Často je tedy vhodné přenést pouze materiálovou informaci bez textur a v Twinmotionu použít optimalizované materiály z dostupné knihovny.

Další možností je automatické slučování objektů při exportu. Na výběr je slučování podle nadřazených skupin (např. skupina židle sloučí všechny objekty židli do jednoho objektu), nebo podle materiálů (např. všechny skleněné výplně oken se stejným materiálem). Na exteriér bylo použito slučování podle skupin (byla vytvořeny skupiny exteriér pro jednotlivé budovy), což nám sloučilo exteriéry budov do tří objektů. Na interiér bylo naopak použito slučování podle materiálů. Po tomto sloučení již ovšem není možné ve scéně Twinmotionu editovat jednotlivé prvky.

Další volbou při exportu je možnost výběru automatické optimalizace geometrie, export vegetace, vyjmutí MEP objektů (mechanické, elektrické a rozvodné instalace), zahrnutí propojených souborů a vyjmutí objektů menších, než je definovaný minimální rozměr, což může být vhodné např. pro eliminování zbytečně malých konstrukčních prvků. Rozšiřující možnosti nám pak dává ještě samotný Twinmotion při importu, kde máme na výběr navíc automatickou opravu UV mapování, definování rotace souřadnicového systému, konverzi jednotek a opět spojování objektů dle definovaného parametru.

Zajímavou možností, kterou Twinmotion disponuje při synchronizaci projektu, je export BIM informací o jednotlivých objektech. Je tak umožněno prohlížet veškerá dostupná data o konkrétních objektech i v Twinmotionu. Nicméně je vyžadováno, aby při exportu nebyla žádná data slučována, což není u velkých projektů použitelné. Chceme-li tedy např. zachovat BIM data pro konkrétní část scény, je vhodné opět rozdělit projekt na dílčí část a tu exportovat zvlášť bez optimalizace.





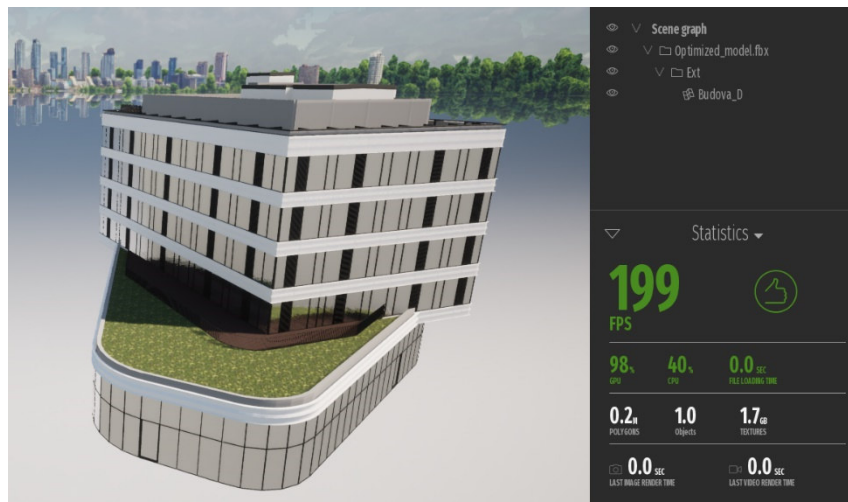
3.4. Performance

Twinmotion nabízí přehledné statistiky o aktuálních hodnotách renderované scény. Lze tak přehledně analyzovat efektivitu, se kterou je scéna renderována. K dispozici jsou v záložce Statistics hodnoty FPS, Polycount, počet objektů, velikost renderovaných textur v paměti, či celkové zatížení CPU a GPU. Pohybem ve scéně a sledováním hodnot FPS tak lze najít problematické části z pohledu drawcallů.

Pro ilustraci jsou zde uvedeny hodnoty scény projektu PORT 7:

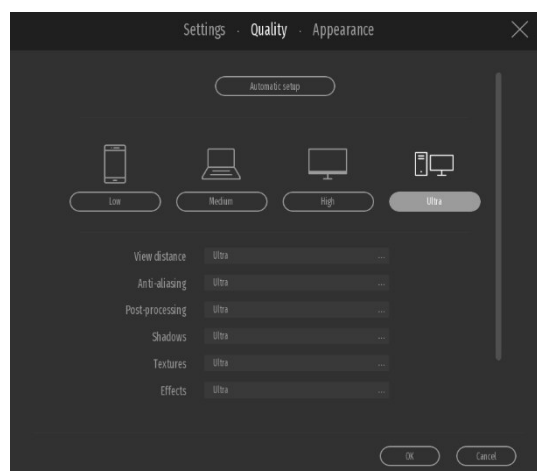
	průměr FPS	polycount	počet objektů
Před optimalizací	0-1	3,4 mil.	41500
Po optimalizaci	100-120	1,3 mil.	17

Ukázka statistik navýšení výkonu na jedné z menších budov (budova D) v projektu:



3.4.1. Nastavení kvality

Další možnosti optimalizace výkonu nám nabízí Twinmotion v záložce Preferences/Quality. Zde je možné nastavit kvalitu renderovaného výstupu v závislosti na dostupném výpočetním výkonu. Opět je zde maximální jednoduchost a přehlednost v nastavovaných parametrech, kde uživatel má možnost výběru šablon nebo úpravy konkrétních hodnot. Pro maximální kvalitu výstupu statických renderů a animací můžeme zvolit nastavení Ultra, nicméně pro výstup virtuální reality je doporučeno nastavení Medium nebo High. Důležitými hodnotami pro VR v projektu PORT 7 jsou parametry Anti-aliasing (vyhlazování hran), Post-processing (tonální úpravy obrazu) a Shadows (kaskádové stíny).



3.4.2. Reflexe

Jedním z důležitých parametrů ovlivňujícím výkon renderování scény jsou i reflexe. Počítání skutečných reflexí tak, jak se světlo odráží ve scéně, je extrémně náročné na výpočet a je umožněno až moderními grafickými kartami a technologií Ray-tracingu, který je ale v současné době stále nevhodný pro zobrazování virtuální reality.

Twinmotion využívá metod tzv. "Screen space reflections" a "Reflection probes".

V prvním případě se jedná o reflektování pouze viditelných objektů kamerou. Tzn. objekty a jejich části, které nejsou aktuálně viditelné kamerou, nejsou reflektovány. Dochází tak často k nepřesným a částečným reflexím obrazu. Kde jsou potřeba přesnější reflexe, je nutno použít sférickou nebo kubickou reflexi. Ta funguje na principu primitivu, na který je v daném místě texturou zachycen obraz okolí. Primitiv disponuje parametrem nastavení velikosti svého působení a na reflektivní objekty uvnitř tohoto objemu následně aplikuje reflexi v podobě zachycené obrazu v textuře. Tímto způsobem lze řešit





lokální reflexe s poměrně použitelným výsledkem. Pro rovné plochy je vhodné využít spíše kubickou reflexi, pro zakřivené plochy sférickou reflexi.

3.5. Nasvícení

Twinmotion využívá čistě dynamického nasvícení. Primárním zdrojem světla ve scéně je obloha (skylight), která je editovatelná v záložce Settings/Lightning. Zde je možné měnit parametry intenzity a teplotu osvětlení, expozici, globální iluminaci (odraz světelných paprsků) a vzdálenost kaskádových stínů, kterou definujeme maximální vzdálenost od kamery, pro kterou jsou stíny ve scéně počítány. Pro potřeby PORT 7 byla tato hodnota nastavena na maximální vzdálenost 300 metrů.

Další možnosti nasvícení už spadají do kategorie lokálních světel. V menu záložce light je několik typů světel, včetně typových světel s IES profilem (Illuminating Engineering Society), což jsou průmyslové standardy k popisu reálných parametrů světel, zahrnující svítivost a světelný úbytek. U těchto světel je rovněž možnost měnit intenzitu, teplota a parametry mající vliv na výpočetní zátěž, jako je zóna vlivu (na kolik polygonů světlo dopadá), vrhání stínů, případně efekt světelných paprsků.

Vzhledem k dynamickému charakteru osvětlení scény je nutné držet množství světel na rozumné úrovni. Polygony, na které svítí více překrývajících se světel, jsou náročné na výpočet.

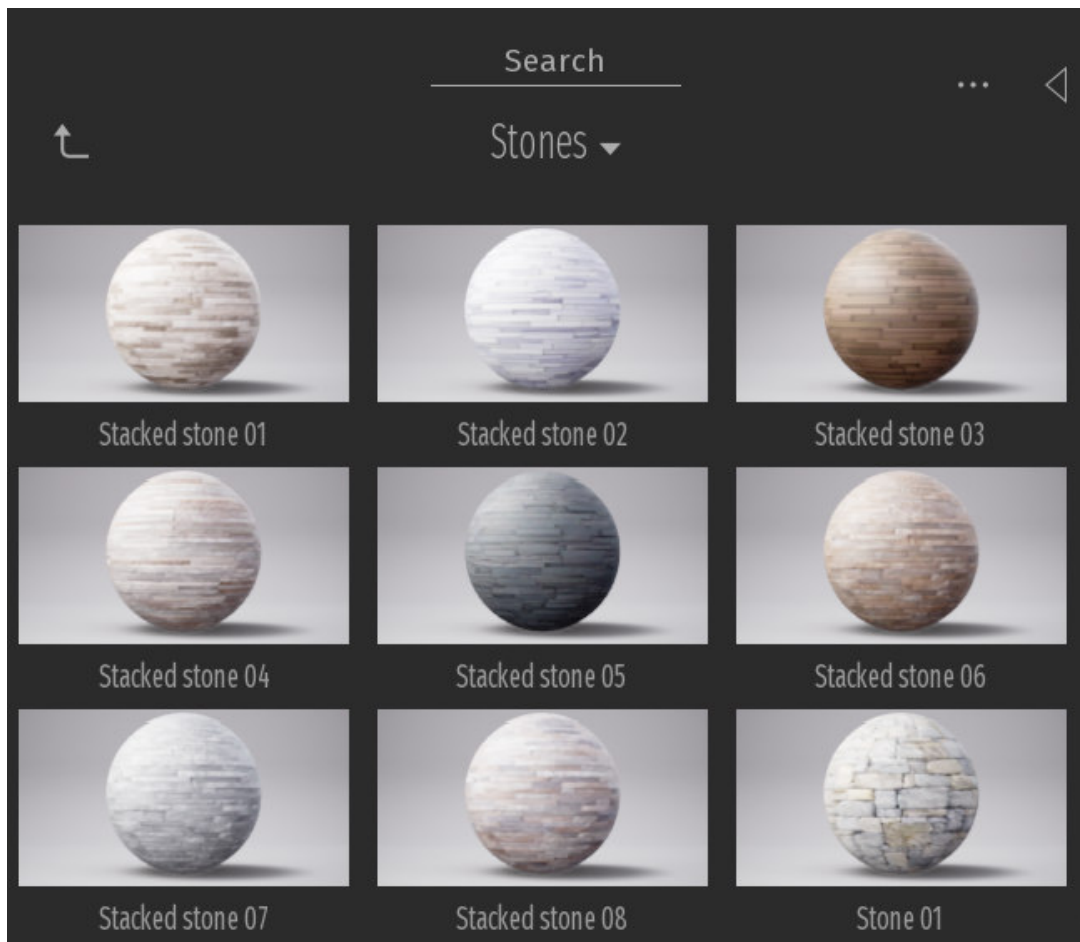
3.6. Materiály

Důležitým prvkem vizualizací jsou materiály. Jak již bylo zmíněno v dřívější kapitole, materiály nepodléhají průmyslovým standardům a zejména materiály tvořené v konstrukčních softwarech nejsou často vhodné pro renderování pomocí PBR formátu. V případě konstrukčních kanceláří, architektonických studií, či firem zabývajících se vizualizacemi, se doporučuje tvorba vlastních materiálových knihoven, reflektujících aktuální potřeby projektů.

3.6.1. Knihovna materiálů

V případě Twinmotionu mají uživatelé k dispozici poměrně rozsáhlou knihovnu základních materiálů. Tato knihovna obsahuje přes 600 zjednodušených PBR materiálů, kterou jsou dále plně editovatelné. Pro běžné potřeby vizualizací ve stavitelství a architektuře tato knihovna pokrývá materiály jako je kov, dřevo, sklo, plasty, kůže, podlahy, stropy, zeleň a mnoho dalších.





Uživatel má zároveň možnost tvořit si vlastní knihovny materiálů, které mohou vycházet ze základní knihovny Twinmotionu, nebo si může uživatel vytvořit své vlastní materiály a tuto knihovnu poté sdílet napříč projekty, či s ostatními uživateli v týmu.

3.6.2. Vlastní materiály

Základní materiály, které byly do projektu přeneseny z Revitu, případně z jiného zdrojového formátu, v sobě nesou většinou pouze RGB informaci o barvě, případně základní Albedo texturu. Už při přípravě dat je vhodné přistupovat k materiálům ve zdrojovém softwaru spíše jako k materiálovým ID. Tzn., že u daných objektů pouze definujete oblasti, kterým budete skutečný materiál přiřazovat až ve vizualizačním softwaru. V konstrukčním softwaru tak můžete použít libovolné (primitivní) materiály. Pro příklad externí části budov ve zpracovávaném projektu byly tvořeny jako jeden objekt s několika materiálovými ID a ve Twinmotionu byly tyto materiály hromadně nahrazeny jejich reálnými PBR verzemi. Twinmotion dává možnost měnit materiály hromadně nebo jen pro konkrétní objekty. Např. hnědou barvu, reprezentující dřevo, lze vyměnit na každém objektu za jinou texturu dřeva nebo je naopak sjednotit pro celý projekt.

Většina materiálů v projektu PORT 7 byla tvořena úpravou základních materiálů z knihovny softwaru. Díky poměrně velké variabilitě a snadné editaci lze rychle osadit objekty ve scéně reálnějšími materiály. Materiály z knihovny přesouváte do scény, kde je následně můžete editovat, či duplikovat. Mezi materiálové kanály, kterým zde můžete přiřadit texturu, patří Diffuse (barva), Metalness (kovový povrch), Normal (optická hrbolatost), Roughness (drsnost nebo odrazivost) a Glow (svítivost).





Jedním z materiálů, který byl ve scéně vytvořen vlastní, je materiál pro strop kanceláře s prvky svítidel. Opět se jedná o jednu z metod optimalizace, kdy místo dynamických světel pro každé svítidlo byl vytvořen částečně emisivní materiál, který svítí do scény za pomoci textury. Pro tento materiál byla vytvořeno několik textur, které definují barvu a odrazivost, dodávají povrchové nerovnosti (normálovou texturou) a černobílou texturou definují oblasti, které na stropě vyzařují do scény světlo. Velikost textur a jejich poměr lze následně ve Twinmotionu upravit, stejně jako jejich offset v osách X a Y.



3.6.3. UV mapování

Aby bylo možné na objekt namapovat texturu, musí se využít některého z dostupných UV mapování. UV mapování je proces promítání 2D obrazu (textury) na plochu 3D modelu. Jednou z možností, typickou pro konstrukční softwary a jednoduché materiály, je mapování pomocí promítání na primitivní objekt. Textura materiálu je nejprve namapována na kouli, válec, jehlan nebo krychli, jejíž plochy jsou následně promítány kolmo na objekt. Tímto způsobem je jednoduše textura promítána na objekt, což je dostatečně vyhovující řešení pro jednodušší modely, zejména s rovnými konstrukčními plochami. Vhodnou volbou primitivního promítaného objektu lze docílit přirozenějšího mapování i u zaoblených tvarů.

Dalším typem mapování je tzv. UVW (UV unwrapping) nebo Object UV. Jedná se roztažení a promítnutí polygonální sítě do 2D plochy. To již často vyžaduje externí modelovací software a patřičný algoritmus, případně manuální zpracování UV mapování. Výhodou je ale maximální kontrola nad promítanou texturou, možnost tvorby vlastních přizpůsobených textur a mapování extrémně složitých objektů, u kterých jsou použitím standardních promítacích postupů viditelné švy textur a deformace promítané textury. UV map může mít jeden objekt několik a např. v herních enginech lze využít různé mapování objektu pro různé materiály.

UVW jsou také základním předpokladem pro zapékání statického světla do textur (Light baking). Pro tento proces jsou nutné tzv. light mapy. UV kanál o je vždy použit výhradně pro mapování textur materiálu. Další kanál pak lze využít pro specifické požadavky light map. Polygony se zde nesmějí překrývat, jinak při procesu zapisování světla dochází k přepisu informací. Dále je nutná určitá mezera mezi jednotlivými plochami, aby nedocházelo k tzv. „light bleedingu“. Důležité je také efektivní využití plochy pro UV



koordináty. Nevyužitá plocha často znamená zbytečně velké textury a zvyšuje nároky na paměť grafické karty.

3.7. Objekty

Nedílnou součástí vizualizace scény je i její osazení digitálními modely. Opět zde ale platí stejné principy optimalizace scény i jednotlivých objektů. Je nutno brát v potaz, které objekty do projektu importujeme, jaký je jejich polycount, počet drawcallů i velikost importovaných textur v materiálech. Snadno se může stát, že projekt, který má po celou dobu 90 FPS má v určitých místech scény radikálně nižší počet FPS. Přestože budeme do projektu vkládat objekty, které jsou maximálně optimalizované, je nutno brát v potaz zejména počet drawcallů, který roste s navyšujícím se počtem objektů a materiálů.

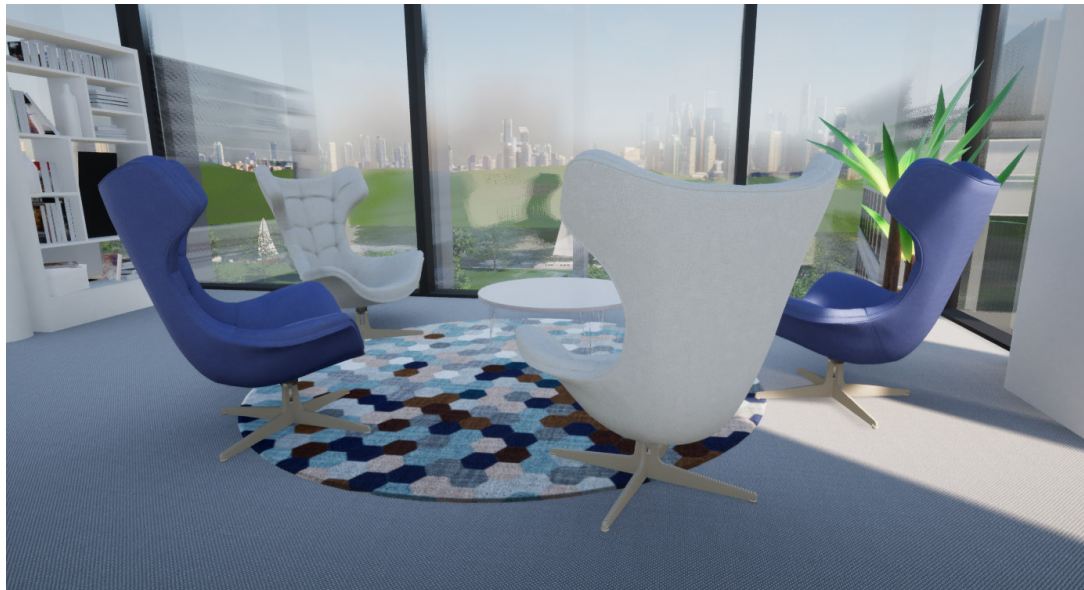
3.7.1. Knihovna objektů

Stejné možnosti, které jsou platné pro materiály, jsou platné i objekty a digitální assety. Twinmotion opět disponuje knihovnou objektů se základními prvky pro osazení exteriérových i interiérových scény. Mezi objekty, které naleznete v knihovně jsou objekty, spadající např. do kategorií kancelář, obývací pokoj, mobiliář, dopravní značení, dopravní prostředky a mnoho dalších. Nechybí zde ani knihovna digitálních charakterů, včetně jejich animací v různých pózách, či modely stromů, keřů a zeleně.

3.7.2. Vlastní objekty

Zejména ve fázi, kdy je např. vyřešen design interiéru kanceláří, nemůže uživatel využít připravených objektů z interní knihovny Twinmotionu a je nucen importovat objekty vlastní. Stejným způsobem, jakým jsou do vizualizačního softwaru importována konstrukční data projektu, lze importovat i dílčí objekty pro jeho vybavení. Děje se tak pod záložkou import, kdy je posléze objekt vložen přímo do počátku souřadnicového systému scény. Poté je možno objekt editovat, vytvářet instance, měnit materiály, či animovat. Tyto objekty lze opět sdružovat v uživatelských knihovnách a sdílet napříč projekty a týmy. Je tedy opět vhodné budovat vlastní knihovny assetů.

V projektu PORT 7 byl importován např. model koženého křesla. V případě importování větších celků vybavení, je opět vhodné spojovat tyto celky do větších objektů a snižovat tak počet drawcallů.



3.8. Prostředí

Pro dosažení přesvědčivější reálnosti můžeme objekt zasadit do kontextu se skutečným prostředím. Máme opět několik možností, které můžeme libovolně kombinovat.

3.8.1. 360° fotografie

Twinmotion nabízí několik typizovaných obrázků pro změnu pozadí, které nalezneme v kategorii Settings/Localization/Background. Můžeme vybrat z několika fotografií měst nebo přírodních prostředí. Nicméně pro vizualizaci konkrétního projektu do konkrétní zástavby je vhodnější mít k dispozici sférickou fotografii z konkrétního místa realizace projektu. V případě vizualizace pro PORT 7 by byla ideální fotografie z přibližného místa reálného patra, které je zpracováno.

Pro aplikování vlastní fotografie pozadí je potřeba vytvořit v některém polygonálním modeláři vlastní sféru a u této sféry obrátit normálové vektory směrem do středu objektu. Vytvoříme pro ni materiál s emisivní texturou sférické fotografie a takto připravený objekt importujeme do Twinmotionu, kde rotujeme objekt s texturou do správné polohy vůči geometrii budovy.

3.8.2. Terén

Přímo v softwaru Twinmotion máme možnost tvorby terénu. Pokud konstrukční projekt již neobsahuje nějakou formu terénu, máme možnost, za pomoci jednoduchých modelovacích nástrojů vytvořit přibližný terén prostředí. Z dokumentace je patrné umístění multiplexu v lokalitě Praha-Holešovice, pro kterou byl vymodelován základní terén s řekou a kopci, aby lépe ilustroval okolí při pohledu z okna administrativního patra.

Po vložení rovinatého terénu bylo využito základních nástrojů pro výškové modelování terénu, bylo vytvořeno koryto řeky a efektem oceánu (Settings/Weather/Effects/Ocean), který byl snížen pod hladinu terénu. Poté byl využit nástroj Paint pro vytvoření materiálové variace povrchu terénu.



3.8.3. OpenStreetMap

Další možností, jak obohatit exteriér, je přidání základních obrysů okolních budov a zasadit tak vizualizaci do kontextu reálného prostředí. Pod záložkou Settings/Urban se nachází funkcionalita, která umožní z knihovny OpenStreetMap generovat prostředí konkrétních lokací. Po vyhledání lokality máte na výběr výběrový box, který ohraničí oblast, jejíž data budou importována do projektu. Importované objekty jsou editovatelné, můžeme měnit jejich výšku, materiál apod. Protože se ale jedná o separátní objekty, není vhodné importovat příliš velká území s mnoha objekty, což by opět zvýšilo počet drawcallů ve scéně. Importované prostředí je potřeba správně rotovat a umístit, případně upravit terén, aby se některé budovy nevznášely.



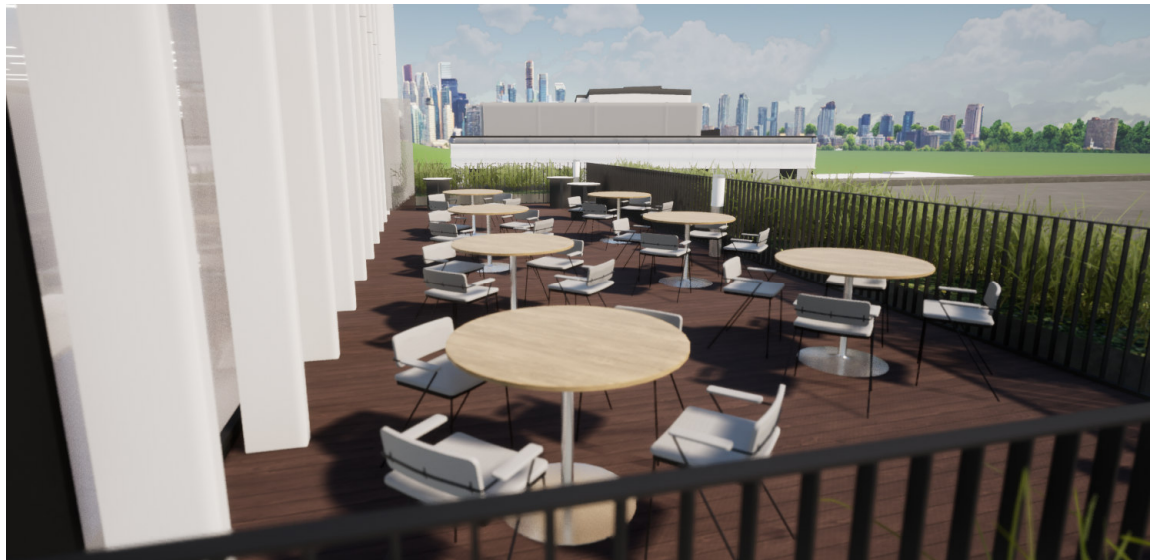
3.9. Export

Twinmotion podporuje renderování statických obrázků, panoramat, videí a interaktivních prohlídek, včetně virtuální reality. Všechny jsou dostupné pod záložkou Media a jejich ovládání je opět jednoduché a intuitivní.

3.9.1. Statické snímky

Export obrázků probíhá do formátu PNG v libovolném rozlišení, až do maximálního rozměru 9000x9000 pixelů. Každý snímek lze před exportem měnit nasvícení scény, měnit počasí, parametry kamery, pozadí, případně přidávat postprodukční efekty nebo zapnout stereoskopický výstup.

Stejným způsobem jsou řešeny i panoramatické snímky.



3.9.2. Animace

Twinmotion umožňuje animovaný pohyb kamery scénou, která se nastavuje pomocí klíčových snímků v jednotlivých polohách. Volným pohybem kamery zaznamenáte klíčové pozice, mezi kterými je poté extrapolována poloha do plynulého pohybu. Do animace lze zahrnout změnu počasí, ročních období, různé fáze procesu budování objektu, či život ve scéně (pohyb lidí, dopravních prostředků, zeleně apod.).

Výstup animace je do formátu MPEG nebo sekvence snímku PNG. Video výstup lze rovněž exportovat do stereoskopického výstupu.

3.9.3. BIMmotion

BIMmotion je výstup interaktivní prohlídky scény, renderované v reálném čase. Výstupem je EXE aplikace, která je spustitelná na PC bez nutnosti instalace aplikace Twinmotion. S ohledem na náročnost scény je opět vyžadován patřičně výkonný hardware, v případě VR aplikace pak headset pro virtuální realitu a VR ready grafická karta.

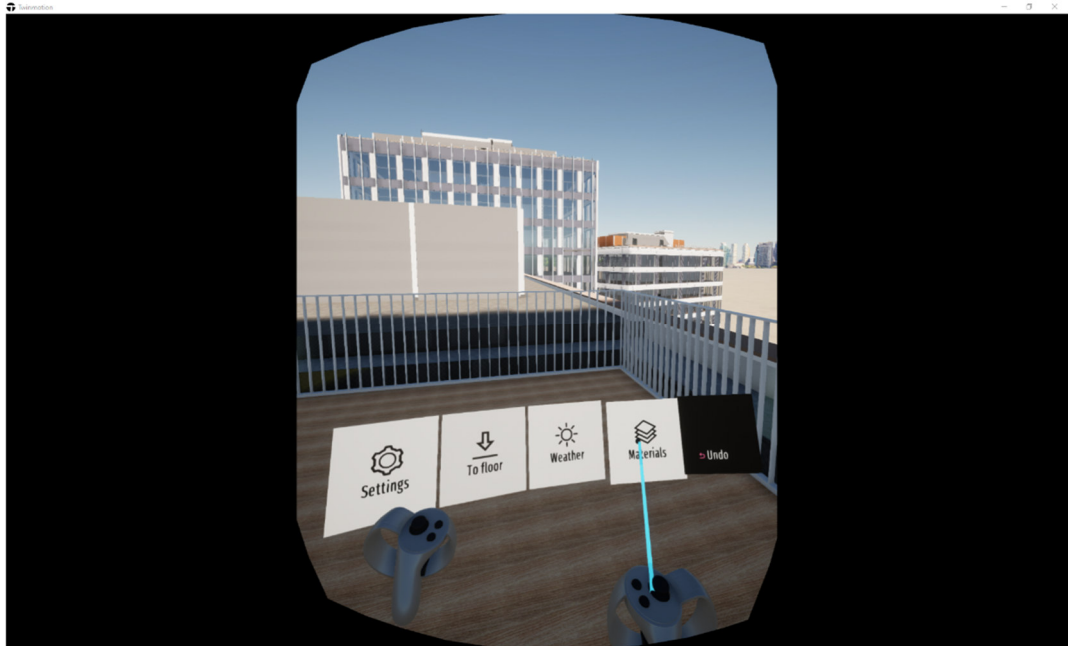




Tvorba této aplikace je extrémně jednoduchá. Stačí pouze pod záložkou Settings/Media/Presenter zvolit výchozí bod interaktivní prohlídky a podobně jako u statických snímků nebo animací editovat vizuální výstup scény.

BIMmotion navíc umožňuje připravit prezentaci projektu a kombinovat statické snímky, videa a jejich libovolné fázi tuto prezentaci přerušit a umožnit uživateli volný pohyb scénou nebo přepnutí do virtuální reality.

Ve virtuální realitě má následně uživatel pomocí ovladačů návod k ovládání scény, možnost upravovat počasí, upravit materiály vybraných objektů, případně volit úroveň nastavení vizuálního výstupu a snížením efektů zvýšit výkon renderované scény.



4. Metodika SKANSKA

Během zpracování projektu PORT 7 a konfrontace s konstrukčními BIM daty společnosti SKANSKA bylo vytipováno několik metodických kroků, na které se byla potřeba během optimalizace a zpracování zaměřit. Pro přehled je uvedeno chronologické zpracování projektu a návrh řešení.

- **Snížení počtu objektů** – největším problémem z pohledu optimalizace je velký počet drawcallů způsobený velkým počtem objektů. Před samotným importem nebo v jeho průběhu je nutné sloučit objekty dohromady, např. podle materiálových skupin. Pozor ale na duplicitní materiály. V projektu REVITU bylo použito celkem 114 materiálů, což by znamenalo minimálně 114 objektů ve výsledné scéně. Např. objekty s materiálem Kov-Ocel a Kov-Ocel-zábradlí lze v těchto případech sloučit dohromady.
- **Rozdělení exteriéru a interiéru** – vzhledem k tomu, že se jedná o velký komplex budov, byla vytipována pouze část projektu, která se bude zpracovávat pro virtuální realitu a ta bude vybavena digitálními assety. Bylo tedy vhodně rozdělit budovy na exteriérovou část (vnější schránku) a vnitřní přístupnou část interiéru. Z pohledu optimalizace je doporučeno rozdělit projekt REVITU na dílčí podprojekty tak, aby bylo možné snadněji optimalizovat jednotlivé





budovy a při importu spojit celý exteriér budovy do jednoho objektu. Zároveň lze snadno exteriéry budov použít v dalším projektu, např. pro jinou lokaci ve virtuální realitě.



- **Relevantní viditelnost** – objekty, které nebudou viditelné je nutno ještě před exportem z Revitu, případně před samotným slučováním objektů, schovat. Exportní plugin Twinmotionu dovoluje exportovat pouze viditelné objekty. Tímto způsobem byl snížen celkový počet polygonů přibližně o 60 %.
- **Materiál skla** - na okolních budovách a vzdálených částech hlavní budovy byl zvolen neprůhledný reflexní materiál. To umožní eliminovat velký počet objektů ve zbytečných a nepřístupných částech projektu. Naopak v oblasti, kde se uživatel může pohybovat, je potřeba zvolit jiný transparentní materiál. Pro 6. NP východní části tak byl použit materiál transparentního skla.



- **Rotace objektu** – Je vhodné při importu dodržovat nulové rotace důležitých objektů ve scéně. Hlavní budova E projektu PORT 7 byla o několik desetin stupňů rotovaná a bylo potřeba ji vyrovnat. Docházelo k tomu, že při osazování scény instancemi objektů docházelo





k nerovnostem na delších vzdálenostech.

- **Export dat** – z důvodu různých postupů sjednocování objektů byl projekt rozdělen na následující 4 podprojekty – Budova A, Budova D, Exteriér budovy E, 5.-7. NP budovy E, kde každý projekt byl importován zvlášť s jinými parametry sloučení objektů. (Pozn. 3 patra budovy E byla vybrána z důvodu částečné viditelnosti horního a spodního patra schodišťovým prostorem. Z patra 5. a 7. patra byla použita pouze viditelná část).
- **Nasvícení scény** – Twinmotion disponuje dynamickým nasvícením, které je navíc editovatelné i z prostředí VR. Pro nasvícení tedy platí obecně platná pravidla pro vizualizace (vyvarovat se tmavých míst, minimalizovat automatickou expozici apod.).
- **Materiály, objekty** – většina materiálů a objektů byla čerpána z dostupné knihovny Twinmotionu, nicméně využití vlastní materiálové knihovny je doporučeno.
- **Prostředí** – okolí projektu bylo řešeno s ohledem na požadavky optimalizace pro VR. Například využití velkého počtu zeleně by scénu příliš zatížilo velkým počtem polygonů. Importování velké oblasti z knihovny OpenStreetMap by naopak výrazně navýšilo počet drawcallů.
- **Performance** – po celou dobu zpracování je nutné hlídat statistiky FPS projektu. Dochází-li v některé části scény k poklesu FPS, bude tento pokles ve virtuální realitě ještě větší (stereoskopické renderování). Zejména při osazování scény assety je nutno tuto hodnotu hlídat a vkládat objekty v rozumné míře, případně využít slučování objektů do větších celků.

5. Možnosti rozšíření

Stakto připraveným projektem je možno dále pracovat. Nemusí zůstat pouze u základní funkcionality. Twinmotion nabízí další možnosti rozšíření interakce, oživení scény, či pokročilejší metody zpracování dat a vizualizace scény.

5.1. Konfigurátor

Twinmotion nabízí při prohlídce ve virtuální realitě i možnost změny materiálu na jednotlivých objektech. Za pomoci přednastavené funkcionality umožní uživateli přístup do knihovny materiálů, které lze posléze snadno aplikovat na konkrétní objekty, skupiny objektů, případně jejich dílčí části. K dispozici je více než 600 dostupných PBR materiálů.

5.2. Animování

K realističnosti scény může přispět také vložení animovaných prvků scény. Twinmotion umožňuje základní animační metody v podobě translace a rotace objektů. Lze tak například otevírat dveře, pokud se k nim uživatel přiblíží.





Další výrazným prvkem je možnost vkládání animovaných cest. Pomocí nich lze například definovat cestu pohybu konkrétnímu objektu. Tyto cesty lze ale zároveň použít k okupování prostoru pohybujícími se lidmi, či dopravními prostředky.

5.3. Export do UE4

Epic Games přináší možnost přímého exportu dat z Twinmotionu do Unreal Engine. Uživatel tak má v kterékoliv části optimalizace, zpracování a vizualizace využít možnost robustnějšího nástroje a s patřičnou znalostí softwaru a zkušeností v jednotlivých procesech dosáhnout vyšší kvality výstupní vizualizace, vyšší míry interaktivity, využít multi-platformního výstupu, či dalších, složitějších optimalizačních metod. Lze například využít statického nasvícení scény a zabezpečení světla do lightmap, systém LODů a MIPů textur, naprogramovat konektivitu na externí systémy, využít moderních renderovacích technologií (Raytracing), apod.

5.4. Cloudové řešení

Pro tvorbu a implementaci řešení lze využít i cloudové architektury. Exportní plugin mezi Revitem a Twinmotion umožňuje komunikaci přes TCP/IP, tím pádem není nutné zpracovávat projekt na stejné pracovní stanici a lze sdílet stanici s výkonnou grafickou jednotkou napříč organizací. Případně lze využít i dedikovaného serveru v cloudu a potřebný grafický výkon si pouze pronajmout na požadované období.

Využití Unreal Engine naopak dává možnosti cloudového streamování, či funkci remote computing, pro zpracování obrazu pro zařízení s nízkým výpočetním výkonem.

Použité zdroje

docs.unrealengine.com

twinmotionhelp.epicgames.com

