

Univerzita Hradec Králové

Pedagogická fakulta

Katedra výtvarné kultury a textilní tvorby

Volný soubor

Vizualizace galerie Vozovna

Bakalářská práce

Autor:	Štěpán Brýdl
Studijní program:	B7507 Specializace v pedagogice
Studijní obor:	Grafická tvorba multimedie
Vedoucí práce:	MgA. Petr Hůza
Oponent práce:	Mgr. et MgA. Pavel Trnka Ph.D.



Zadání bakalářské práce

Autor:	Štěpán Brýdl
Studium:	P16P0216
Studijní program:	B7507 Specializace v pedagogice
Studijní obor:	Grafická tvorba - multimédia
Název bakalářské práce:	Volný soubor - Vizualizace galerie Vozovna
Název bakalářské práce AJ:	Free file - Visualisation of gallery Vozovna

Cíl, metody, literatura, předpoklady:

Bakalářská práce se zabývá tématem tvorby virtuálních prohlídek a aplikuje ho na budovu staré vozovny dopravního podniku v Hradci Králové. Práce bude rozdělena na teoretickou a praktickou část. Praktickou částí bude vytvoření spustitelné aplikace pro prohlídku a případnou instalaci možného galerijního prostoru. Teoretická část popisuje tvorbu od počátku projektu až po jeho konec. Rozebírá jednotlivé postupy při tvorbě projektu a řeší jak je nejlépe provést. Dále pak zmiňuje historický vývoj v oboru 3d vizualizace a jeho použití. Cílem je vytvoření realistické interaktivní vizualizace jak by prostory vozovny DP mohly vypadat a jak by se případně daly využívat.

BRIAN O'DOHERTY, Uvnitř bílé krychle. Nakladatel: tranzit.cz (1. vydání), 2014, 108 s. ISBN: 978-80-87259-30-6 J. ŽÁRA, B. BENEŠ, J. SOCHOR, P. FELKA, Moderní počítačová grafika. Nakladatel: Computer Press (2. vydání), 2008, 609 s. ISBN: 80-251-0454-0 JAN KRÍŽ, Mistrovství v 3ds Max. Nakladatel: Computer Press (1. vydání), 2010, 1152 s. ISBN: 978-80-251-2464-2 RICHARD POULIN, Jazyk grafického designu. Nakladatel: Slovart, 2012, 288 s. ISBN: 978-80-7391-552-0 Daniele, Todd, Poly-Modeling with 3ds Max. Nakladatel: Taylor & Francis Ltd (1. vydání), 2008, 288 s. ISBN: 978-0240810928

Garantující pracoviště: Katedra výtvarné kultury a textilní tvorby,
Pedagogická fakulta

Vedoucí práce: MgA. Petr Hůza

Oponent: Mgr. et MgA. Pavel Trnka, Ph.D.

Datum zadání závěrečné práce: 31.10.2018

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval pod vedením vedoucího bakalářské práce samostatně a uvedl jsem veškeré použité prameny a literaturu.

V Hradci Králové

.....

Štěpán Brýdl

Anotace:

BRÝDL, Štěpán. *Volný soubor - Vizualizace galerie Vozovna*. Hradec Králové: Pedagogická fakulta University Hradec Králové, 2019. 44 s. Bakalářská práce.

Bakalářská práce se zabývá tématem tvorby virtuálních prohlídek a aplikuje ho na budovu staré vozovny dopravního podniku v Hradci Králové. Práce bude rozdělena na teoretickou a praktickou část. Praktickou částí bude vytvoření spustitelné aplikace pro prohlídku a případnou instalaci možného galerijního prostoru. Teoretická část popisuje tvorbu od počátku projektu až po jeho konec. Rozebírá jednotlivé postupy při tvorbě projektu a řeší jak je nejlépe provést. Dále pak zmiňuje historický vývoj v oboru 3D vizualizace a jeho použití. Cílem je vytvoření realistické interaktivní vizualizace jak by prostory vozovny dopravního podniku mohly vypadat a jak by se případně daly využívat.

Klíčová slova: 3D, vizualizace, galerie, design

Annotation:

BRÝDL, Štěpán. *Free file - Visualisation of gallery Vozovna*. Hradec Králové: Faculty of Education, University of Hradec Králové, 2019. 44 pp. Bachelor Degree Theses.

The topic of this thesis is aimed at creation of virtual tours in 3D and uses modern tools to create a interactive 3D visualisation of an old depot building in Hradec Králové. Practical part of this work is to create a executable application that will illustrate how this old building would look like if it was renovated and turned into a gallery. Theoretical part describes individual steps of creation and how they should be done correctly while also briefly mentioning the history and usage of 3D graphics and visualisations.

Keywords: 3D, visualization, gallery, design

Poděkování:

Především děkuji vedoucímu mé bakalářské práce MgA. Petru Hůzovi, za ochotu a čas strávený při konzultacích. Dále, bych chtěl poděkovat Martinu Tomášovi za poskytnutí fotografií a dokumentace budovy Dopravního Podniku v Hradci Králové.

Obsah

Úvod.....	8
I TEORETICKÁ ČÁST	9
1 Vizualizace 3D.....	10
2 Definice 3D.....	11
2.1 Obecný vývoj počítačové 3D grafiky	11
2.2 3D ve filmu.....	14
2.3 3D v herním průmyslu	16
2.4 3D v architektuře	17
3 Základní znalosti pro tvorbu 3D modelů	18
3.1 Modelování 3D modelů.....	18
3.2 Tvorba textur	19
4 Nástroje.....	20
4.1 3ds Max	20
4.2 Substance Painter.....	21
4.3 Unreal Engine	22
II PRAKTICKÁ ŘÁST	25
5 Tvorba Vizualizace.....	26
5.1 Inspirační zdroje	26
5.2 Workflow	27
5.3 Tvorba modelů.....	30
5.4 Příprava prostředí Unreal Engine.....	31
5.5 Sazba modelů do prostředí Unreal Engine	32
5.6 Tvorba UVW map.....	34
5.7 Tvorba textur	36
5.8 Tvorba materiálů v Unreal Engine	37
5.9 Finalizace	38
Závěr.....	40
Seznam použité literatury.....	41
Seznam internetových zdrojů	42
Seznam obrázků v textu	43
Seznam zkratk a pojmů	44

Úvod

Vytvoření náhledu produktu je v dnešní době již téměř samozřejmostí ve většině grafických oborů. Pomáhají v předstihu porozumět formě navrhovaného objektu a bez větších nákladů tak návrh upravovat na základě viditelných dat. Architekti tak mohou například lépe posoudit, jak bude stavba zapadat do zvoleného prostředí, a podle toho tak dělat na projektu změny. Dávají také možnost zadávajícímu klientovi vidět, jak jeho požadovaný produkt bude přibližně vypadat, dříve než je fyzicky vytvořen a tím umožňují větší kontrolu nad výsledkem.

Jako téma své bakalářské práce jsem si zvolil tvorbu interaktivní 3D vizualizace starší budovy Dopravního Podniku v Hradci Králové. Práce má za cíl ukázat, jak by interiér budovy mohl vypadat po případné rekonstrukci a předělání na galerijní prostory. Zároveň ukazuje možnosti tvorby vizualizací v dnešní době.

V teoretické části krátce popisují historii 3D grafiky, hlavní osobnosti, které se zasloužily o vývoj a její použití v některých pro ni významných oborech. Dále zmiňují některé populární programy, které se dnes pro tvorbu 3D grafiky mohou používat.

Praktická část mé práce se zaměřuje na proces při tvorbě samotné interaktivní 3D vizualizace. Zmiňuje zvolený pracovní postup, neboli workflow a popisuje jednotlivé části tvorby projektu.

I TEORETICKÁ ČÁST

1 Vizualizace 3D

Pod názvem vizualizace se skrývá mnoho různých technik a způsobů tvorby. Od výkresů na papír, přes fotomontáže až po nově se vyvíjející interaktivní 3D vizualizace. Všechny mají však stejný cíl. Vizualně, tedy obrazově nebo animací, představit určitá data, která dosud divákovi nejsou známá. S rychlým vývojem počítačové techniky se rozšiřují i možnosti, jak taková data znázorňovat.

Grafické vizualizace 3D objektů nabývají v poslední době stále většího významu. Nacházejí uplatnění v mnoha oblastech produktového a obalového designu, ale především interiérového designu a architektury. Na rozdíl od fotomontáží, které mohou být občas nepřesné, 3D vizualizace mají možnost být tvořeny na základě reálných dat. Díky tomu se mohou více přiblížit realistické podobě, a tak být věrohodnější. Další obrovskou výhodou je, že objekty jsou tvořeny trojrozměrně a tudíž je možné se na ně dívat a renderovat je z více úhlů, což by u ostatních technik vyžadovalo začít tvořit náhled zcela znova. Není tedy divu, že se 3D grafika stává nejčastější a nejreálnější formou vizualizace.¹

Ani zde ale inovace nekončí a stále se hledají nové možnosti, jak toto řemeslo posunout dále. Za posledních 10 let stoupla kvalita vzhledových možností počítačových her neuvěřitelně vysoko a díky uvolnění některých tzv. enginů (základní část aplikace, na které je postavena počítačová hra) pro širokou veřejnost mohou grafická studia poprvé začít tvořit a experimentovat s tzv. interaktivními prohlídkami a vizualizacemi. Jedná se o aplikace, které běží na počítači a umožňují pozorovateli pohybovat se volně ve virtuálním prostoru dané vizualizace v reálném čase. Navíc za pomoci nově se rozšiřujících náhlavních zařízení tvořených právě pro virtuální realitu, jako je například Oculus Rift² nebo HTC Vive³, se vizualizace dají spustit i v tzv. VR prostředí pro maximální vtažení do tohoto nehmotného světa.^[1]

¹ VAUGHAN, William. Digital modeling. Berkeley, CA: New Riders, c2012. ISBN 03-217-0089-9. s. 356-357

² Oculus. Oculus Rift [online]. [cit. 2019-04-22]. Dostupné z: www.oculus.com/rift/

³ VIVE. VIVE [online]. [cit. 2019-04-22]. Dostupné z: www.vive.com/eu/product/

2 Definice 3D

3D počítačová grafika, na rozdíl od 2D, používá pro reprezentaci tří-dimenzionální geometrická data, která jsou uložena v počítači za účelem tvorby výpočtů a renderování dvojrozměrných obrazů. Tyto obrazy jsou pak ukládány pro pozdější zobrazení či zobrazování v reálném čase. V počítačových grafických softwarech jsou odlišnosti mezi 2D a 3D často mlhavé, 2D grafika využívá 3D pro tvorbu efektů jako je například osvětlení a 3D zas naopak používá 2D techniky renderování. Krom 3D tisku jsou matematické trojrozměrné modely zobrazovány jako 2D obraz pomocí techniky zvané rendering. Dále mohou být také používány bez vizuální reprezentace za účelem tvorby simulací a výpočtů.^[2]

2.1 Obecný vývoj počítačové 3D grafiky

Ivan Edward Sutherland je americký počítačový vědec, narozený roku 1938, který je označován jako zakladatel počítačové grafiky. Jeho raná práce a výuka počítačové grafiky, společně s Davidem Cannonem Evansem, na Univerzitě v Utahu v sedmdesátých letech je označována jako průkopnická té doby.

Sutherlandovou závěrečnou prací v roce 1963 při studiu pro titul Ph. D. na univerzitě MIT (Massachusetts Institute of Technology), byl program zvaný *SketchPad*, s jehož pomocí bylo možné vytvořit první 3D modely. Inovativní aplikace, která ukazovala alternativní formu práce a ovládání počítače. Dokázala přijímat definované křivky, vykreslovat horizontální i vertikální přímky a kombinovat je do obrazců a tvarů, ty pak mohly být posouvány, rotovány a mohla jim být měněna velikost. SketchPad byl také první program schopný kresby a díky algoritmu pro ořezávání, dokázal přibližovat a oddalovat obraz.⁴

Od roku 1965 do 1968 byl Sutherland docentem elektrotechniky na Harvardské universitě. Společně s Bobem Sproullem, Quintinem Fosterem, Dannym Cohenem a dalšími studenty, vytvořil roku 1966 první náhlavní zařízení pro virtuální realitu, které dokázalo zobrazovat drátěné modely v prostoru. Zařízení obsahovalo dvě samostatné obrazovky, které vykreslovaly obraz podle divákova pohybu.

V letech 1968 až 1974 byl Sutherland profesorem na Universitě v Utahu, kde mezi mnoha jeho studenty byli například Frank Krow, pozdější vynálezce algoritmů pro vyhlazování hran v digitálním prostředí, zvané antialiasing, Edwin Catmull, spoluzakladatel studia Pixar a později také president společnosti Walt Disney a John Warnok, počítačový vědec a podnikatel, který se stal spoluzakladatelem grafické firmy Adobe Systems Inc.

Sutherland v roce 1968 založil, společně se svým kolegou Davidem C. Evansem, firmu *Evans and Sutherland*. Společnost byla průkopníkem v oblastech počítačového hardwaru pracujícím v reálném čase, akcelerované 3D grafice a programovacích jazycích pro tiskárny.

Další významnou osobností je E. E. Zajac, počítačový vědec z Bell Labs, který v roce 1963 vytvořil krátký film, nazvaný *Simulation of a two-gyro gravity attitude control system*. V tomto snímku generovaném počítačem ukazuje, jak by chování satelitu mohlo být ovlivněno putováním po oběžné dráze země. Později ho následoval Frank Sindon, další vědec z Bell Labs, který vytvořil animovaný film zvaný *Force, Mass and Motion*, který ilustroval Newtonovy pohybové zákony.

Netrvalo dlouho a velké korporace začaly mít zájem o nově se vyvíjející počítačovou grafiku. TRW, Lockheed-Georgia, Sperry Rand a mnoho dalších firem začaly v polovině šedesátých let investovat do počítačové grafiky. IBM, firma vytvářející počítače, byla rychlá s odpovědí na nově rostoucí poptávku a v roce 1964 vydala IBM 2250 grafický terminál, první komerčně dostupný počítač.

V roce 1966 Ralph Baer, vedoucí inženýr v Sanders Associates, vytvořil domácí video hru, která byla později nazvána *Odyssey*. Byla velice jednoduchá a vyžadovala celkem levné elektronické součástky. Hra se stala prvním spotřebitelským grafickým produktem.

Roku 1970 Edwin Catmull začal studovat na Univerzitě v Utahu, kde se přihlásil na Sutherlandovu výuku počítačové grafiky. Catmull, vyrůstající na Disney, miloval animaci, ale již v mládí zjistil, že pro takovou práci neměl dostatek talentu. Když však na universitě viděl vývoj počítačové grafiky, s mnoha ostatními pochopil, že se její rozmach stane revolučním v oblasti animace. V rámci výuky vytvořil jeho první animaci zobrazující otevírající a zavírající se ruku. Jeho cílem se poté stalo vytvoření celovečerního filmu generovaného pomocí počítačové grafiky.

Po dokončení studia začal Catmull pracovat jako ředitel ve společnosti Computer Graphic Lab v LA, kde shromáždil talentovanou skupinu lidí a začal pracovat na 2D animačních nástrojích, které by ulehčily práci animátorů. Jedním z vynálezů byl kreslicí program zvaný Paint. Catmull a jeho tým časem nechávají 2D animace a začínají se věnovat 3D počítačové grafice.

Koncem sedmdesátých let práce Catmulla zaujala George Lucase z Lucasfilm a roku 1979 ho proto oslovil, aby vedl tým, který by přivedl počítačovou grafiku, úpravu videa a digitální zvuk do zábavního průmyslu. Jako vicepresident společnosti *Industrial Light & Magic*, pomohl vytvořit technologii pro kompozici digitálního obrazu, která dokázala přesvědčivě kombinovat více snímků do jednoho celku.

V roce 1986 Steve Jobs, spoluzakladatel firmy Apple, koupil digitální divizi Lucasfilmu a založil společnost Pixar, kde se Catmull stal technickým ředitelem.

Neposledním významným absolventem university v Utahu byl John Warnock. V jeho doktorské závěrečné práci roku 1969 vynalezl tak zvaný Warnok algorithm, pomocí kterého je možné dělit renderovanou scénu na menší části, za účelem ulehčení procesu výpočtu pro počítač. V roce 1976, Warnok pracoval v *Evans & Sutherland*, vytvořil základy pro programovací jazyk později nazvaný *PostScript*. V roce 1978 začal pracovat v Xerox PARC, kde však nebyl schopný přesvědčit řízení firmy, aby tehdejší grafický tiskařský jazyk InterPress využili komerčně, proto odešel, aby roku 1982 založil firmu *Adobe*. V jeho nové firmě společně s Geschkem and Putmanem vynalezl *PostScript* a později ho přivedl na trh společně s tiskárnou LaserWriter od společnosti *Apple*. Roku 1991 Warnok vytvořil systém prvně nazvaný Camelot, který se nakonec vyvinul ve formát PDF (Portable Document Format). Účelem Camelotu bylo efektivně zaznamenat dokumenty z jakékoliv aplikace, následně mít možnost je elektronicky zasílat a poté je zobrazit či tisknout na jakémkoli zařízení.^{5 [2]}

⁴ Sketchpad of Ivan Sutherland. History-Computer [online]. [cit. 2019-04-28]. Dostupné z: history-computer.com/ModernComputer/Software/Sketchpad.html

⁵ RYAN, Daniel L. History of computer graphics. Bloomington, IN: AuthorHouse, 2011. ISBN 978-1456751173. s. 28-31

2.2 3D ve filmu

Jedno z hlavních oblastí, kde se trojrozměrná grafika výrazně prosazuje, je filmový průmysl. Filmové triky se dnes používají v téměř každém filmu a často se k nim využívá právě počítačová technika. V některých případech může být časově či finančně náročné modelovat komplexní scény a předměty v reálném světě, a tak se za pomoci moderní 3D technologie nahrazují v postprodukcí tvorbou CGI (Computer Generated Imagery). Ve 3D je možné vytvořit cokoliv, od menších doplňkových předmětů, přes kulisy, až po celé světy. Trojrozměrná grafika může být buď dosazována do reálného prostředí, nebo využita k tvorbě celých filmů.

Prvním filmem, který zasadil 3D grafiku do svého prostředí, byl *Futureworld* (1976), kde se objevuje počítačem generovaná ruka a obličej vytvořené Edwinem Catmulem a Fredem Parkelem. Roku 1982 vznikl film *Tron*, který se z velké části snažil opřít o virtuální svět. Kvůli limitacím tehdejší techniky však nedosáhl velké kvality a stal se jak komerčně tak vizuálně ne příliš úspěšným. Když však roku 1989 vyšel film *The Abyss* obsahující 3D efekty, které se daly označovat za realistické, dostal cenu americké akademie za vizuální efekty. Studio Industrial Light & Magic zde vytvořilo na tehdejší dobu v celku realistické vizuální efekty, mezi které patřilo i zobrazení tvora z mořské vody.



Obr. 1. Štěpán Brýdl, Ukázka 3D modelu ruky ve filmu *Futureworld* (1976)

V roce 1995 se filmová studia Walt Disney Pictures a Pixar zapsaly do filmové historie tím, že vydaly film *Toy Story*. První celovečerní animovaný film zcela vytvořený počítačem, čímž odstartovalo novou éru jak animace, tak CGI. Na filmu tehdy pracovalo 27 animátorů, kteří nejdříve vytvořili animovaný storyboard, podle kterého následně tvořili 3D animace a modely. Renderování jednoho snímku tehdy trvalo v rozmezí od 45 minut až do 3 hodin při rozlišení 1536x922. Film měl ve finále 114 240 snímků a renderování zabralo kolem 800 000 strojních hodin.^{6 [3]}

Americké studio Pixar, založené roku 1986, je jedním z prvních filmových studií zbývajících se vizuálními efekty. Studio spadá pod Walt Disney Studios a sídlí v Kalifornii. Jeho předchůdcem bylo grafické studio, které bylo od roku 1979 součástí Lucasfilmu. V roce 1986 Steve Jobs tuto digitální divizi od Lucasfilmu odkoupil a založil Pixar. Po roce 2006 se stalo součástí Walt Disney Studios, které studio odkoupilo za 7,4 bilionu dolarů. Pixar také vytváří svůj vlastní renderovací systém pro realistické výstupy, zvaný RenderMan. Systém vydává pro nekomerční projekty zdarma, ale také nabízí placenou verzi pro komerčně tvořená díla. Studio již vydalo přes 20 celovečerních animovaných filmů a získalo mnoho ocenění.⁷

Industrial Light & Magic je americkým studiem, založeným v roce 1975, které je součástí společnosti Lucasfilm, založenou Georgem Lucasem při začátku produkce filmů *Star Wars*. Studio od roku 1978 sídlí v Kalifornii a krom filmů *Star Wars* a *The Abyss* pracovalo na téměř tři-sta dalších filmech, mezi které patří například trilogie *Návrat do budoucnosti*, *Jurský Park*, filmy *Star Trek*, *Piráti z Karibiku* a také spolupracovali na filmu *Avatar*. U filmu *Titanic* byla dokonce polovina z dvousetmilionového rozpočtu vyhrazena jen pro speciální efekty, na kterých IL&M pracovalo.⁸

Mezi další hlavní studia, která se dnes zabývají tvorbou VFX (Visual effects), patří v současnosti především Walt Disney Animation Studios, Avid, DreamWorks, Studio Ghibli a mnoho dalších.

⁶ This week in history: Toy Story and CGI. Spiceworks [online]. [cit. 2019-05-13]. Dostupné z: community.spiceworks.com/topic/412181-this-week-in-history-toy-story-and-cgi

⁷ OUR STORY. Pixar [online]. [cit. 2019-05-13]. Dostupné z: www.pixar.com/our-story-1

⁸ Wired. The Untold Story of ILM, a Titan That Forever Changed Film [online]. [cit. 2019-05-5]. Dostupné z: www.wired.com/2015/05/inside-ilm/

2.3 3D v herním průmyslu

Počítačové 3D hry jsou interaktivní zábavní aplikace, které jsou graficky prezentované pomocí třech dimenzí (výšky, hloubky, šířky). Jsou zpravidla jedním z nejnáročnějších softwarů, které se dají na domácím počítači spustit a nejnovější 3D hry také často vyžadují nejnovější produkty počítačového hardwaru.

První 3D hry se skládaly jen z tzv. wire frame (drátěných) modelů, neboli 3D modelů, které byly tvořeny jen čarami, vrcholy, neměli texturu ani plochu a dalo se skrze ně dívat. Hra *Elite*, vydaná v roce 1984 studiem *Telecomsoft*, byla jednou z prvních takových. V roce 1990 vydalo studio *Bethesda* hru *Terminator*, která již dokázala zobrazovat plochy modelů namísto čar i s jednoduchými texturami a šestnácti barvami. I když 3D hry vypadaly oproti tehdejším konkurenčním 2D hrám hůře, představovaly velký technický posun vpřed a tím získávaly určité uznání.

Zlom nastal v roce 1992, kdy malé americké studio *id Software* vydalo *Wolfenstein 3D*. Tato hra získala obrovskou pozornost celého herního světa, nejen díky kvalitě, ale také byla dostupná jako shareware, neboli zdarma a volně šiřitelná, což v této době bylo velmi výjimečné. I když je *Wolfenstein* označen jako 2.5D či pseudo 3D, jelikož stále používá 2D prostorové prvky, jeho vydání udělalo z *id Software* přední tvůrce herního enginu. Ještě větší úspěch zaznamenalo studio o rok později, 10. prosince 1993, kdy na trh vydalo *DOOM*. Bylo zde použito mnoho nových technik a prvků jako vylepšené pohyblivé osvětlení, mapování textur a možnost hrát přes síť s ostatními hráči. Další významnou hrou byl *Duke Nukem 3D* od *3D Realms*. Její prostředí, ve kterém se hra odehrávala, bylo velmi dynamické. Obsahovalo jezdící vozidla, lesklé plochy, zničitelné objekty a další.

Největší průlom však přišel v roce 1996 s vydáním hry *Quake*. První skutečná 3D hra od *id Software*, která měla plně barevné textury modelů, částicové efekty a možnost spustit hru ve vysoké kvalitě na 3D-akcelerovaných grafických kartách. Všechny předchozí hry se sice odehrávaly v 3D prostředí, ale často si se složitějšími prvky vypomáhaly 2D animovanou grafikou, neboli sprity. *Quake* měl naopak plně trojrozměrné objekty a postavy s texturami. Po vydání *Quake* se začalo odvětví 3D v herním průmyslu vyvíjet neskutečně rychle. Vývojáři si začali uvědomovat, čeho je a není možné se současným hardwarem dosáhnout, a tak se mu začali přizpůsobovat.

vat a tvořit hry na hranicích výkonnostních možností. Od té doby se hranice 3D her posouvají stále kupředu a herní studia se předhánějí v kvalitě, možnostech a příběhů. Dnes jediný charakter může obsahovat i několik tisíc polygonů a herní scény i desítky miliónů.⁹

2.4 3D v architektuře

Architektonické vizualizace jsou používány k zobrazení obrazu, animace nebo prohlídky v reálném čase míněného architektonického designu či renovace. Digitální umělci tak mohou architektům umožnit experimentovat s vizuálním designem a aspektem. Tyto obrazy jsou používány pro prezentace a významně pomáhají při představování projektu zadavateli či budoucímu majiteli.^[1]

Architektura a městský design obvykle vyžaduje spolupráci několika lidí s rozdílnými specializacemi. Každý se na daný projekt dívá ze svého profesního pohledu a tak jsou architektonické vizualizace důležitým prostředkem pro komunikování jednotné vize, mezi všemi co na daném projektu pracují. Dnes se 3D učí skoro každý budoucí architekt, jelikož je nepostradatelným nástrojem důležitým nejen pro technické výkresy, ale také pro kvalitní prezentaci díla.

Používáním výkonných nástrojů architekti, designéři a 3D umělci mohou také s tvořeným designem manipulovat v reálném čase. Rychle přecházet mezi konceptem a finálním produktem, tvořit odlišné verze projektu a zkoumat různé možnosti ve velmi krátkém čase.

Mezi hlavní aplikace pro tvorbu architektonických vizualizací patří především SketchUp, Rhino, V-Ray, Maxwell, AutoCAD, 3ds Max a mnoho dalších modelovacích a renderovacích nástrojů.¹⁰

⁹ Historie 3D grafiky. Fakulta informatiky Masarykovy univerzity [online]. [cit. 2019-05-15]. Dostupné z: www.fi.muni.cz/usr/jkucera/pv109/2001/xkolcar_index.html

¹⁰ 3D Visualization. Easy Render [online]. [cit. 2019-05-15]. Dostupné z: www.easyrender.com/architectural-visualization/is-3d-rendering-and-architectural-visualization-the-future-of-architectural-design

3 Základní znalosti pro tvorbu 3D modelů

3.1 Modelování 3D modelů

Digitální modelování je proces tvorby matematické reprezentace trojrozměrného tvaru. Výsledkem tohoto tvoření je něco, co dnešní průmysl označuje jako 3D model nebo 3D mesh. Tyto modely se mohou tvořit jak manuálně, tak automaticky. Nejčastějším zdrojem digitálních modelů je tvorba umělců nebo techniků, a to pomocí 3D počítačových aplikací a také skenováním předmětů v reálném světě za pomoci speciálních zařízení.^[5]

To jak model bude vypadat, ovlivňuje nejen finální vize či reference, podle které se model tvoří, ale také způsob, jakým se model bude dále používat. Pro filmový průmysl se například modeluje s o dost menším omezením vůči počtu polygonů, než pro herní průmysl. Zatím co filmový průmysl si může dovolit renderovat scénu i několik měsíců, herní průmysl naopak potřebuje modely, aby byly relativně jednoduché, pro vykreslování v reálném čase.¹¹

S pokrokem modelovacích softwarů je dnes častým prvním krokem tvorby velmi detailní skulptura v programech jako je ZBrush či Mudbox, kde se umělec nemusí zabývat optimalizací a tvoří za účelem nejvyšší kvality. Poté je nezbytné celý model přepracovat znovu s lepším polygonálním tokem pomocí retopologických nástrojů, kde cílem je přizpůsobit nový model dalším potřebám, jako je texturace, tvorba animace a další, s minimálním vizuálním dopadem na kvalitu modelu.

Vytváření kvalitního modelu vyžaduje dobré plánování kroků dopředu a znalosti v mnoha odvětvích 3D grafiky. Vytvořit model, který vypadá dobře, může být snadné, ale vytvořit model, kterému se dá dobře vytvořit UV unwrap či animační kostra, je několikanásobně těžší a vyžaduje mnoho znalostí a zkušeností.

Mezi nejpopulárnější aplikace pro tvorbu 3D modelů patří ZBrush, 3ds Max, Maya, Blender, Cinema 4D, Mudbox a další.^[1]

¹¹ VAUGHAN, William. Digital modeling. Berkeley, CA: New Riders, c2012. ISBN 03-217-0089-9. str. 4, 42

3.2 Tvorba textur

Textury přetváří nebarevný povrch 3D modelů na barevná detailní tělesa. Součástí texturovacího procesu je tzv. Unwrapping a tvorba UV, tvorba Ambient Occlusion a Normal map, malování textur s různými atributy a následné spojení všech vrstev do jedné, která se může následně umístit na model.

Dříve než se modelu může přiřadit vlastní textura je nezbytné udělat tzv. Unwrap (rozbalení), což je převedení částí 3D modelu na 2D plátno, na které se následně aplikuje samotná textura. Způsob, jakým se 3D prostor převádí na 2D plátno, je závislý na požadavcích detailu finálního modelu. Některé části modelu například pro diváka nebudou vidět a proto nepotřebují na 2D plátně zabírat velký prostor, naopak to, co divák uvidí nejbližší, by zpravidla mělo mít největší podíl z celého plátna. Pokud je potřeba, aby model byl co nejvíce detailní, může být rozdělen i na několik 2D ploch, kdy každá vyžaduje svou vlastní texturu.¹²

Po vytvoření Unwrapu se může model texturovat pomocí shaderů, které se snaží imitovat reálné fyzikální materiály, generováním textur z fotografií a nebo připůsobenou texturou, která je vytvořena kreativní imaginací umělce. K tomuto procesu se používají aplikace jako Substance Painter, Substance Designer či Marmoset Toolbag, ze kterých se následně exportují soubory cílené pro konkrétní aplikace.

To jakým způsobem textury fungují, je také závislé na prostředí, ve kterém se projekt vytváří. Pro herní průmysl a filmový průmysl jsou odlišné způsoby počítání fyzikálních vlastností, a proto textury musí být také tvořeny přesně pro zvolené prostředí.

¹² VAUGHAN, William. Digital modeling. Berkeley, CA: New Riders, c2012. ISBN 03-217-0089-9. str. 48, 49

4 Nástroje

4.1 Autodesk 3ds Max

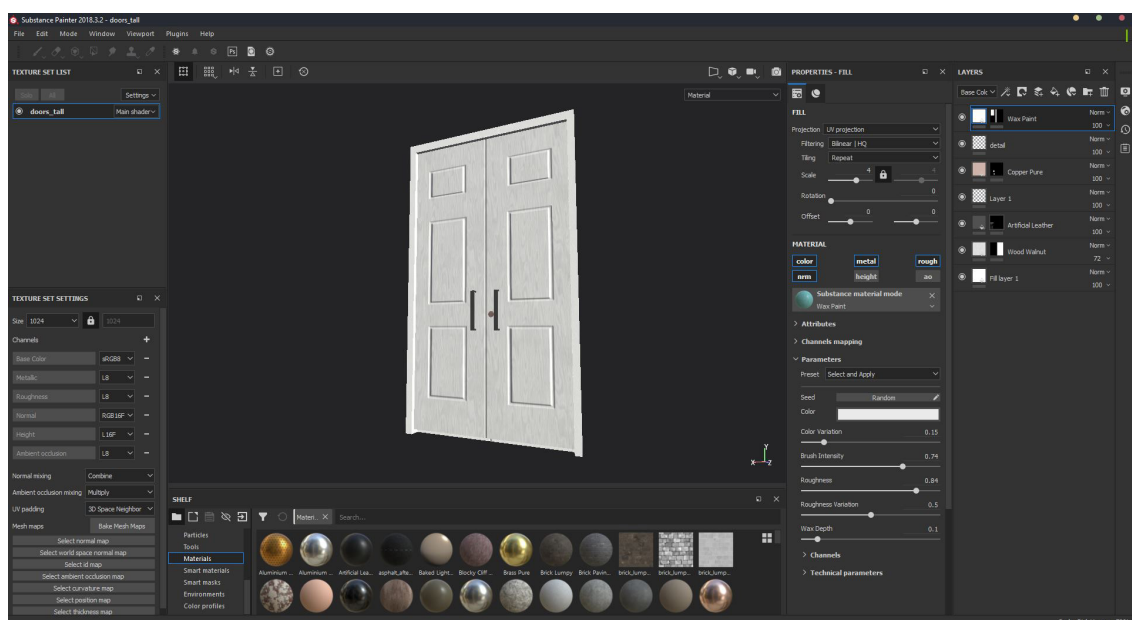
Aplikace Autodesk 3ds Max, dříve 3D Studio a 3D Studio Max, je v dnešní době jednou z nejrozšířenějších v oblasti profesionální tvorby 3D nejen pro architekturu, ale také pro herní či filmový průmysl. Je to program pro tvorbu a práci s modely v trojrozměrném prostoru. Dají se zde tvořit animace, které můžeme dále detailně definovat a upravovat. Můžeme zde jednotlivé objekty nebo části objektů různě barvit a vytvářet jim textury, má bohaté množství nástrojů pro modelování a flexibilní prostředí pro vývoj a integraci pluginů. V dnešní době je aplikace dostupná ve dvou provedeních: 3ds Max a 3ds Max Design. Zatímco první zmíněné je určeno spíše pro umělce, herní studia a filmovou produkci, 3ds Max Design je cílen hlavně na vizualizace a architekturu. V jádru jsou však oba programy velmi podobné a liší se hlavně uspořádáním nástrojů v rozhraní.^[5]

Prvotní aplikaci 3D Studio vydal v roce 1990 pro DOS platformu Gary Yost a Yost Group pod vydavatelem Autodesk. Program měl na tehdejší dobu mnoho funkcí. Možnost kreslit tvary křivkami, vytvářet rychle jednoduchá geometrická primitiva a základní nástroje pro úpravu objektů. Roku 1992 vydává Autodesk 3D Studio R2, ve kterém se poprvé objevuje podpora pro tzv. pluginy. Díky tomu pak Yost Group vydá hned několik samostatných modulů, od částicových systémů až po kamerové efekty. S příchodem nového Microsoft operačního systému na 32-bitové architektuře, začal vývoj nového programu s pracovním názvem Jaguar, později 3D Studio Max, který by byl schopen novou technologii využít.

V roce 1995 Autodesk na LA SIGGRAPH oznámil vydání 3D Studia Max pro Windows NT. Díky interaktivnímu stínování, texturovaným objektům, animaci v reálném čase pomocí hudby, možnosti vytvořit pohyblivou animovací kostru pro modely a mnoha dalších nástrojů, se program stal středem pozornosti celé konference. Během dalšího vývoje přibyl i MAXScript, programovací jazyk přímo pro programování scriptů a pluginů v prostředí aplikace. V roce 2000 je 3D Studio Max přejmenováno na 3ds max a nově přibývá nástroj ActiveShade, který umožnil rendrovací náhled v reálném čase. Nové verze programu vydává Autodesk ročně.¹³

4.2 Substance Painter

Substance Painter je aplikace určená pro texturaci objektů vytvořených v programech pro 3D modelování, jako je například Autodesk 3ds Max, Blender, Autodesk Maya a další. Nabízí rozsáhlé rozhraní nástrojů pro tvorbu detailních materiálů s možnostmi exportu pro mnoho populárních renderovacích enginů, jak v herním tak filmovém průmyslu. Je to velmi flexibilní a rychlý program, kterému roste popularita a byl již využit při tvorbě mnoha CGI (Computer Generated Imagery) pro celovečerní filmy jako například Blade Runner 2049, Logan, The Mill, nebo AAA hrách jako jsou Forza Horizon 3, Uncharted 4, RainbowSix Siege a další.¹⁴



Obr. 2. Štěpán Brýdl, Ukázka prostředí aplikace Substance Painter 2018

¹³ History of Autodesk 3ds Max. Times & Trends Academy [online]. [cit. 2019-05-15]. Dostupné z: timesandtrendsacademy.com/history-autodesk-3ds-max

¹⁴ Substance painter. Substance [online]. [cit. 2019-05-15]. Dostupné z: www.substance3d.com/products/substance-painter

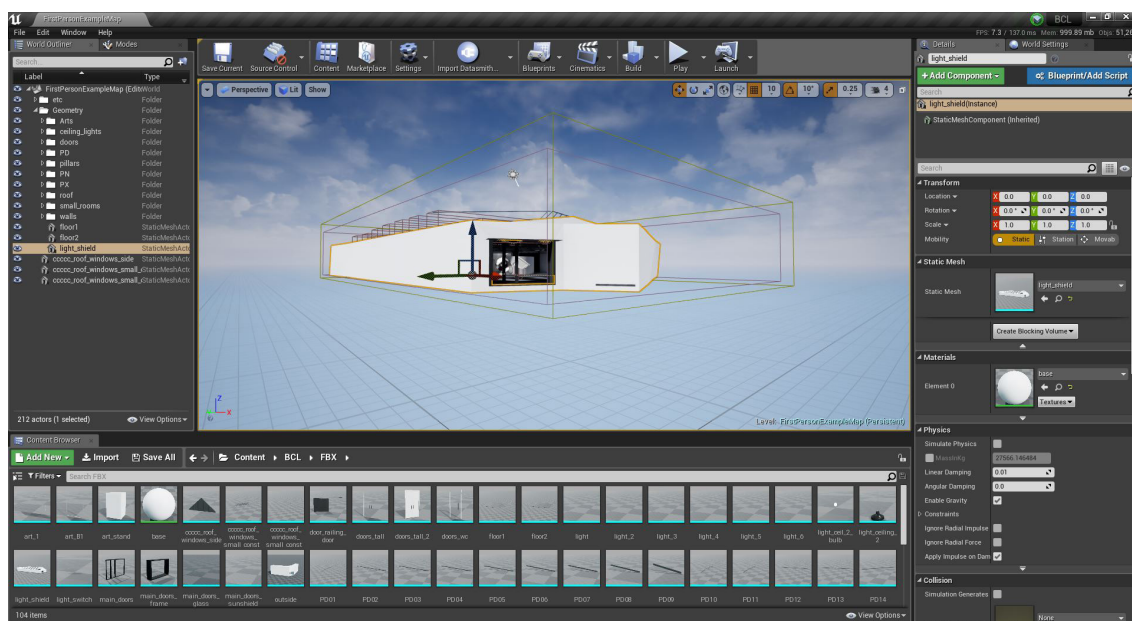
4.3 Unreal Engine

Unreal Engine je herní vývojářský software pro tvorbu počítačových a mobilních her vydávaný firmou Epic Games, která byla založena roku 1991 Timem Sweeneyem a Markem Reinem. Studio sídlí v Cary, Severní Karolíně a vlastní vývojářská studia Chair Entertainment a Cloudgine.

Engine byl poprvé ukázán v roce 1998 ve hře Unreal. I když byl primárně určen pro vývoj her v první osobě, byl úspěšně použit na řadu her jiného žánru. Je postaven na programovacím jazyce C++ a díky tomu je velmi univerzální a výkonou volbou pro mnoho velkých herních studií. Aktuální verzí je Unreal Engine 4 vydaný v roce 2014.¹⁵

První generaci softwaru naprogramoval Tim Sweeney, zakladatel firmy Epic Games. Sweeney začal engine psát v roce 1995 pro hru, která je dnes známá jako Unreal. Engine již tehdy uměl renderovat na úrovni, jak hardware tak software, dokázal řešit kolize objektů a obsahoval editor obsahu, ve kterém se dalo v reálném čase upravovat prostředí her. V roce 2002 byla vydána společně se hrou *America's Army*, kterou vytvořila americká armáda jako rekruční nástroj, druhá verze enginu. Se základy na technologii první verze, některé nástroje byly od základů přepracovány a rozšířeny. Přibyl také plugin pro export a import modelů do aplikací Autodesk Maya a D Studio Max. Engine nyní také podporoval Xbox a Playstation. V roce 2004 byly zveřejněny první náhledy Unreal Engine 3. Prvotní verze UE 3 podporovaly jen platformy Windows, PlayStation 3 a xbox 360, zatímco rozšíření pro IOS a Android bylo přidáno až v roce 2011. Podpora pro OS X přibyla o rok později. Od vydání první verze dostal UE 3 mnoho aktualizací, které přidávaly nové nástroje a možnosti, například deformace objektů, simulace davu, renderování globální iluminace v reálném čase a další. I když byl Unreal Engine 3 příznivý pro uživatelské modifikace, vytvářet vlastní hry a publikovat je pro veřejnost nebylo kvůli licencím enginu zatím možné. Nicméně v listopadu roku 2009, Epic vydalo verzi UE 3 SDK zvanou Unreal Development Kit (UDK), která byla dostupná zdarma pro veřejnost. V srpnu 2005 Mark Rein, viceprezident Epic Games, oznámil že Unreal Engine 4 byl ve vývoji od roku 2003 a prohlásil, že lidé budou v šoku, až UE 4 uvidí. Jedna z hlavních nových funkcí enginu bylo vykreslování stínů globální iluminace v reálném čase pomocí tzv.

voxel cone tracking, což eliminovalo potřebu předprodukčního renderování. Funkce však byla později nahrazena sice podobným, ale méně náročným nástrojem, z důvodů obav vysokých požadavků na výkon. Dále engine podporoval C++ programování v reálném čase, když už engine běžel a díky novému způsobu scriptování pomocí tzv. Blueprintů bylo programování her ještě více zjednodušeno. V březnu 2015 při GDC (Game Developers Conference) Epic oznámilo vydání UE 4 se všemi budoucími aktualizacemi zdarma pro všechny uživatele. Na oplátku požadují 5% podílu z výtěžku produktů, které vydělají více jak 3 000 UDS za čtvrt roku. Od roku 2014 tak používání enginu vzrostlo několikanásobně a používají ho již nejen velká, ale také malá studia. Od otevření Epic Games Store v prosinci 2018, Epic si nenárokuje zmíněný 5% výtěžek na hry, které jsou vydány skrze jejich online obchod.¹⁵



Obr. 3. Štěpán Brýdl, Ukázka prostředí aplikace Unreal Engine

¹⁵ Unreal Engine. Wikipedia [online]. [cit. 2019-05-15]. Dostupné z: en.wikipedia.org/wiki/Unreal_Engine

¹⁶ This is Unreal - The story of Unreal Engine and the games it powers. Hotgates [online]. [cit. 2019-05-15]. Dostupné z: hotgates.eu/this-is-unreal/

II PRAKTICKÁ ČÁST

5 Tvorba vizualizace

5.1 Inspirační zdroje

Tím, jak by prostor galerie mohl vypadat jsem se inspiroval zejména v knize Uvnitř bílé krychle od irského umělce a kritika Briana O'Dohertyho. Autor se zde snaží popsat předpoklady, na kterých jsou postaveny moderní galerie a jak by tyto prostory měly, nebo neměly ovlivňovat vystavovaná díla. Zabývá se komplikovanými vztahy umělců a společnosti vůči uměleckým prostorům a shrnuje všechny zásadní proměny výstavního prostoru v závislosti na tvořeném umění v uplynulém století.

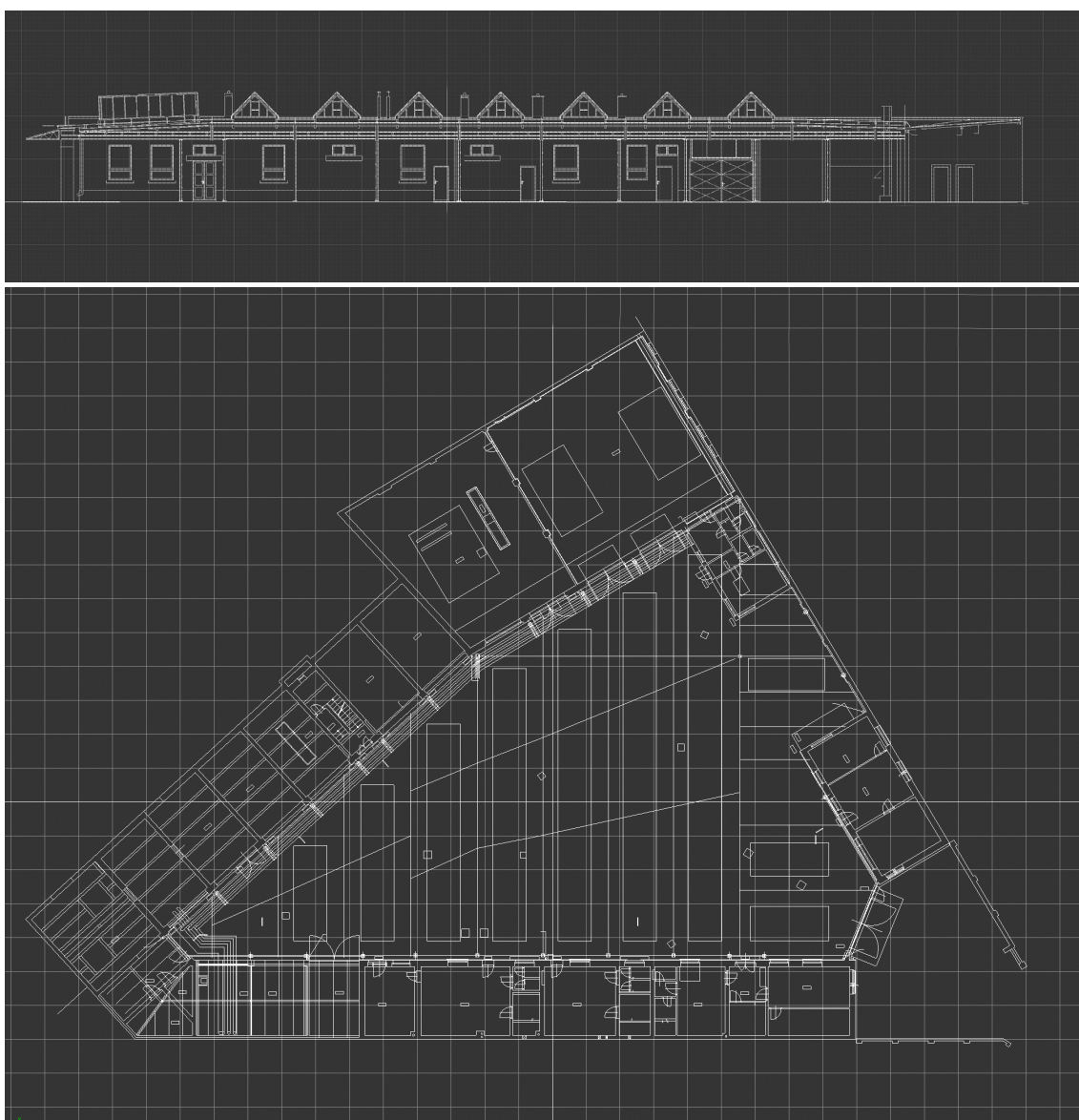
„Ideální galerie zbavuje umělecké dílo všech signálů, které by jakkoliv zpochybňovaly fakt, že jde o „umění“. Umění je potřeba izolovat od všeho, co by omezovalo možnost hodnotit dílo jen jím samotným.“^[6]

„Galerie vznikají podle stejně přísných pravidel jako středověké kostely. Dovnitř nesmí pronikat vnější svět, takže okna jsou obvykle zaslepená. Stěny jsou natřeny na bílo. Strop se stává zdrojem světla.“^[6]

(Brian O'Doherty, 2015)

5.2 Workflow

Díky získaným technickým výkresům a fotografiím prostředí jsem si mohl udělat velmi dobrou představu o tom, jak by finální prostor mohl vypadat. Z nashromážděných referencí jsem si sestavil seznam předmětů, které by v prostorách měly být přidány a případně, podle kterých referencí by se určité části budovy měly upravit. Dále jsem si stanovil náročnost finální aplikace a to přibližně šedesát snímků za sekundu na zařízení s grafickou mobilní kartou Nvidia Geforce 960m, čtyřjádrovým procesorem Intel Core i7-6700HQ a 16 GB operační paměti RAM.



Obr. 4. Štěpán Brýdl, Ukázka technických výkresů, převedených na křivky v prostředí 3ds Max

Po stanovení těchto cílů a s představou, jak by prostor měl vypadat, jsem přešel k modelování. Pro tvorbu modelů jsem si zvolil aplikaci Autodesk 3ds Max, se kterou mám již zkušenost. Díky mnoha nástrojům, které aplikace obsahuje, jsem byl schopný tvořit efektivně a bez větších potíží. Bylo důležité dokončit co nejvíce modelů, dříve než jsem cokoliv mohl vkládat do prostředí herního enginu. Chtěl jsem tak předejít komplikacím, kdy by bylo nutné předělávat více modelů najednou, pokud by na sebe jednotlivé části stavby nenavazovaly či nevycházely rozměry. Bylo tedy potřeba vymodelovat veškerou konstrukci stavby, stěny, zdi a další velké objekty, jako například vstupní dveře.

Mezitím jsem mohl začít pracovat na základním nastavení scény. Pro tvorbu samotné aplikace jsem si zvolil Unreal Engine od studia Epic Games. Tento herní engine je v posledních letech velmi populárním pro tvorbu vizualizací a je také dostupný zdarma pro menší projekty. Zde bylo potřeba naprogramovat charakter, vytvořit tzv. pause menu a také nastavit základní osvětlení vizualizace. Aby byl ve všem pořádek, rozhodl jsem se modely, materiály a textury dávat zvlášť a vytvořil jsem proto několik podsložek projektu.

S dokončenými modely byl čas na testovací import celé scény do Unreal Engine bez jakýkoliv UVW map či materiálů. Bylo potřeba zkontrolovat, jestli velikosti scény a charakteru vytvořeného v Unreal Enginu sedí, jestli se pozice jednotlivých modelů neodchylují od pozic v 3ds Max a také první zjištění, jakou náročnost a kolik výkonu bude přibližně potřeba pro běh finální vizualizační aplikace.

Po menších úpravách scény, většinou kvůli optimalizaci výkonu, jsem mohl přejít k tvorbě UVW map a textur pro jednotlivé modely. Hotové textury pak byly přeneseny do prostředí Unreal Engine a následně se z nich tvořily materiály, které se aplikovaly na modely.

Následovaly finální úpravy osvětlení, barev, kvality stínů a opravy drobných chyb ve scéně. Aplikace se poté mohla exportovat a konečně spustit.



Obr. 5. Martin Tomáš, Fotografie prostoru Dopravního Podniku



Obr. 6. Martin Tomáš, Fotografie prostoru Dopravního Podniku

5.3 Tvorba modelů

K vytváření 3D modelů jsem použil nástroj 3ds Max od firmy Autodesk. Modely musely být tvořeny dostatečně detailní, aby dostaly realistickým výsledkům, ale zároveň ne příliš členité, aby konečná scéna nebyla příliš náročná pro stanovený přibližný cíl šedesáti snímků za sekundu.

Díky zapůjčeným technickým výkresům celého prostoru od města Hradec Králové, bylo možné tvořit podle přesných rozměrů, bez obav porušení poměrů stran a výšek budovy. Výkresy byly pomocí aplikace Adobe Illustrator převedeny do grafického formátu .dwg, který je podporován přímo aplikací 3ds Max.

Po importování technických výkresů bylo možné přesně modelovat podle křivek jednotlivých pohledů. Bylo třeba vymodelovat nejen stěny budovy a střechu, ale také celou ocelovou konstrukci, která je v interiéru budovy z velké části vidět. Postupně tedy vznikly tři hlavní skupiny modelů: stěny, střecha a kovová konstrukce. Z estetických a funkčních důvodů pro galerijní prostor, byla některá okna ve stěnách zazděna, aby vzniklo více prostoru pro umístění výtvarných děl. Díky členitosti ocelové konstrukce v technickém výkresu jsem mohl tvořit jednotlivé části konstrukce zvlášť, a tak si v této skupině modelů ulehčit nutnost optimalizace dělením konstrukce na menší díly. U sloupů konstrukce, které může divák vidět detailněji, jsem ponechal více detailů, jako jsou například šrouby a matice. Naopak u částí, ke kterým se divák tak blízko nedostane, byly detailní prvky upouštěny. U stěn bylo naopak velmi důležité stanovit si, které prvky budou moci být, za účelem optimalizace, tvořeny pomocí textury a které bude naopak lepší vymodelovat. Příkladem zde jsou cihly na stěnách, které by bylo velmi složité vykreslovat pomocí modelu, ale texturou jsou nejen méně náročné, ale také umožňují větší kontrolu při finálních úpravách.

Po vytvoření všech potřebných modelů podle technických výkresů jsem mohl přejít k referencím a vytvořenému seznamu s předměty, které jsem chtěl do prostředí přidat navíc. Mezi tyto objekty spadaly například upravené dveře, stropní osvětlení, doplňující stěny a další.

Jednotlivé hotové modely jsem pak postupně přidával do jediného 3ds Max souboru, kde jsem celou scénu skládal a připravoval na importování do prostředí Unreal Engine.

5.4 Příprava prostředí Unreal Engine

Základní nastavení scény v Unreal Engine vycházelo z předem vytvořeného bluepintu First Person s nastavením kvality na maximum a počítačem jako cílenou platformou. Tento blueprint automaticky vytvořil a naprogramoval charakter s kamerou z první osoby, nastavil základní pokyny pro vstupní zařízení, jako je klávesnice a myš, ale také případné VR náhlavní zařízení a další ovladače. Samotný charakter byl vytvořen společně s animovaným modelem celého těla postavy, ten však pro tento projekt nebyl potřebný, a tak byl za účelem lepší optimalizace odstraněn. Zároveň byly ze scény i obsahového repositáře vymazány veškeré předvytvořené 3D modely, které nebyly potřeba pro finální projekt.

Následovalo vytvoření základní hierarchie repositáře projektu. Snažil jsem se o zachování pořádku v uspořádání složek tak, aby v průběhu tvorby nemohlo dojít ke ztrátě dat či zmatku při budoucích úpravách. Základem je trojice složek: Blueprints, ve které je naprogramovaný charakter a k němu veškeré pomocné soubory, UIX, složka pro uživatelské rozhraní a podklady jako ikony či text a nakonec složka BCL, která má podsložky pro modely, materiály a textury. Při ukládání modelů do souborů .fbx jsem se také snažil co nejvíce pojmenovávat jednotlivé části stavby podle již určených názvů v technických výkresech. Toto rozdělení a pojmenovávání mi umožnilo přehledné a rychlé hledání assetů pro potřebné aktualizace při tvorbě projektu.

Dalším krokem byla úprava některých nastavení a přidání vstupních kláves, jako například mezerník, Escape a další, pro budoucí programování uživatelského rozhraní a rozšíření možnosti ovládání charakteru a scény. V nastavení bylo nutné upravit popis, název a ikonu projektu, dále zdali se má scéna spouštět přes celou obrazovku nebo v okně a jestli může být uživatelem zvětšována a zmenšována.

Posledním nutným krokem bylo importování zkušebního modelu bez textur do prostředí Unreal Engine, aby mohl být nastaven způsob importu pro veškeré budoucí modely. Zde bylo potřeba nastavit, aby Unreal Engine při tvorbě assetů nevytvářel vlastní UVW mapy, ale aby je načítal přímo z importovaných modelů, ke kterým byly manuálně vytvořeny v programu 3ds Max. Když zkušební soubor fungoval tak, jak měl, mohl jsem jej smazat a přejít k sazbě již hotových modelů.

5.5 Sazba modelů do prostředí Unreal Engine

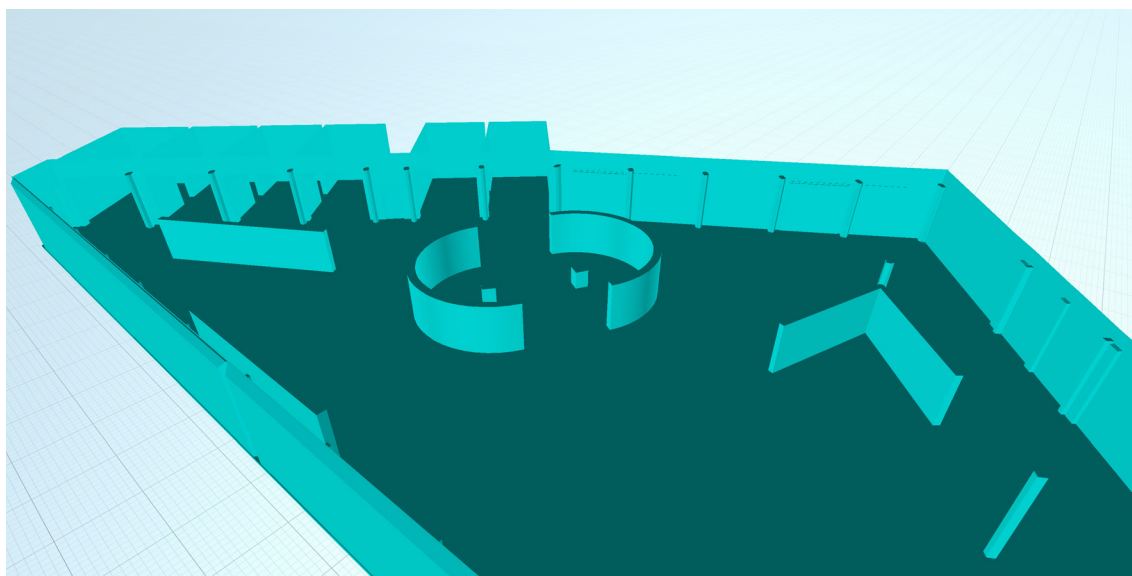
Modely vytvořené v 3ds Max jsem postupně exportoval do samostatných souborů formátu .fbx, který dokáže herní engine číst. Jelikož Unreal Engine po importování modelů resetuje poziční souřadnice pivotů, bylo potřeba myslet dopředu, jak se s tímto problémem vypořádat, aby se celá scéna nemusela po vytvoření v 3ds Max tvořit zcela znova v prostředí Unreal Engine.

Pro ulehčení procesu vkládání assetů na správné místo podle výkresu a s ohledem na optimalizaci scény, jsem si rozdělil modely do dvou skupin. V první byly modely, které se ve scéně objevovaly jen jednou, měly přesně dané místo a byly nepohyblivé. Sem patří většina modelů jako jsou stěny, části střechy, podlaha a valná většina kovové konstrukce. Do druhé skupiny patří modely, které se ve scéně objevují vícekrát, ať už ve stejné či pozměněné velikosti. Patří sem, sloupy konstrukce, plátina a rámy obrazů, některé stěny, stropy, světla a další. Modely patřící do první skupiny mohly mít předem stanovené souřadnice přiřazené v aplikaci 3ds Max, a tak po importování do Unreal Engine stačilo assetu nastavit nulovou pozici a model byl tak na svém přesném místě. Naopak modelům z druhé skupiny bylo nutno v 3ds Max vynulovat souřadnice, aby s nimi po importování do Unreal Engine bylo jednoduché ve scéně pohybovat a následně je duplikovat, případně zvětšovat či zmenšovat.

Důvodem, proč tyto modely musely být vkládány do scény tímto způsobem, byla optimalizace. Příkladem je dvojice identických dveří na levé stěně při spuštění aplikace. Oboje jsou prakticky stejné, však jedny jsou převrácené po ose y, tedy mají kliku na opačné straně. Jednou možností, by bylo v 3ds max vytvořit dva modely dveří, jedny s klikou na straně pravé a druhé na levé. To by však znamenalo, že při spuštění aplikace se musí načíst dva různé modely, jelikož engine neví, že modely jsou až na převrácení identické. Také by to znamenalo, že by se musely vytvořit převrácené textury a tím by herní engine načítal o několik souborů navíc. Kdyby tak bylo zacházeno s více modely, mohlo by to výrazně zvýšit výkonnostní nároky aplikace a snížit tak počet vykreslených snímků za vteřinu. Lepším způsobem je tedy vytvořit jen jeden model, který se převrátí až v prostředí Unreal Engine. Aplikace pak díky tomu může načíst do paměti jen jeden model a při vykreslování ho už jen převrátí.

Po umístění assetů do scény bylo nutné zvolit, se kterými objekty může charakter kolidovat a případně jakým způsobem u jednotlivých modelů kolize počítat. Kdyby například podlaha neměla nastavené kolize, charakter by se ve scéně propadl pod model a nemohl by se hýbat. Kolize mohou být nastaveny buď na jednoduché, komplexní a nebo žádné. Jednoduchá kolize je tvořena pouze kvádrem, který je velikostí a rotací přizpůsoben modelu tak, aby zabral co nejméně místa. Takové kolize mají menší dopad na výkon a jsou dobré pro rovné stěny, sloupky, podlahy a další jednoduché předměty. U složitějších objektů, jako jsou zahnuté stěny či zdi s otvory, skrze které by charakter měl být schopen procházet, je nutné používat komplexní. Ta počítá kolize přesně podle tvaru modelu, je však i náročnější pro výpočet. Žádné kolize je dobré přiřadit veškerým modelům, které nepřijdou do kontaktu s charakterem, tedy stropy, horní části kovové konstrukce a jiné nedosažitelné předměty.

Posledním krokem před tvorbou textur bylo vytvoření orientačních rozlišení lightmap jednotlivých modelů ve scéně. Základní hodnotou pro veškeré assety bylo rozlišení 64x64 pixelů. Menším a méně detailním objektům jsem následně přiřadil velikosti textury od 32 do 256 pixelů, větším a více detailním od 256 do 512. Bylo potřeba počítat s jistou úpravou těchto hodnot při finálních korekturách celého projektu, jelikož zatím nebylo možné vycházet z UVW map, které většina modelů neměla vytvořené.



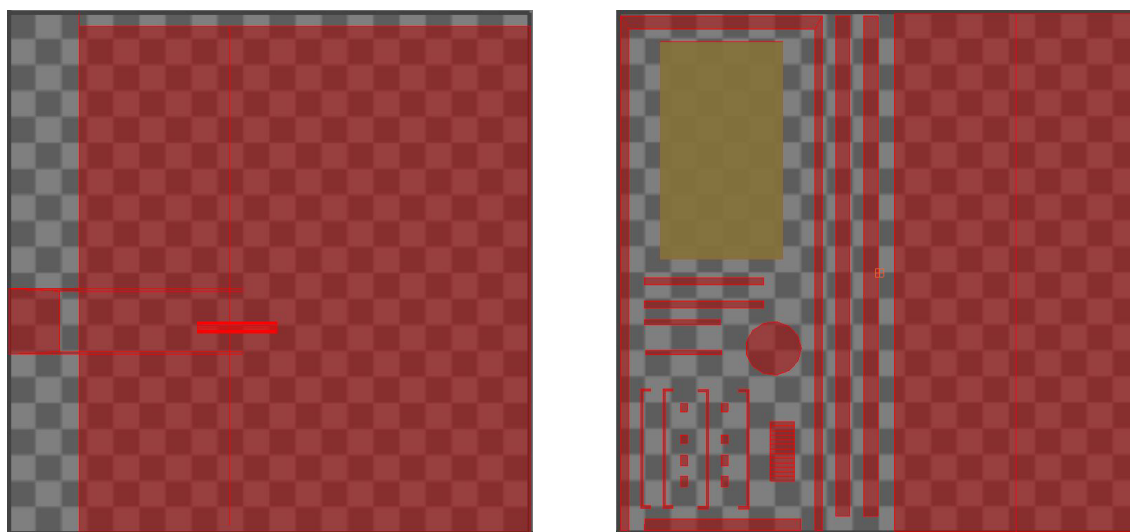
Obr. 7. Štěpán Brýdl, Ukázka objektů, se kterými může charakter kolidovat

5.6 Tvorba UV map

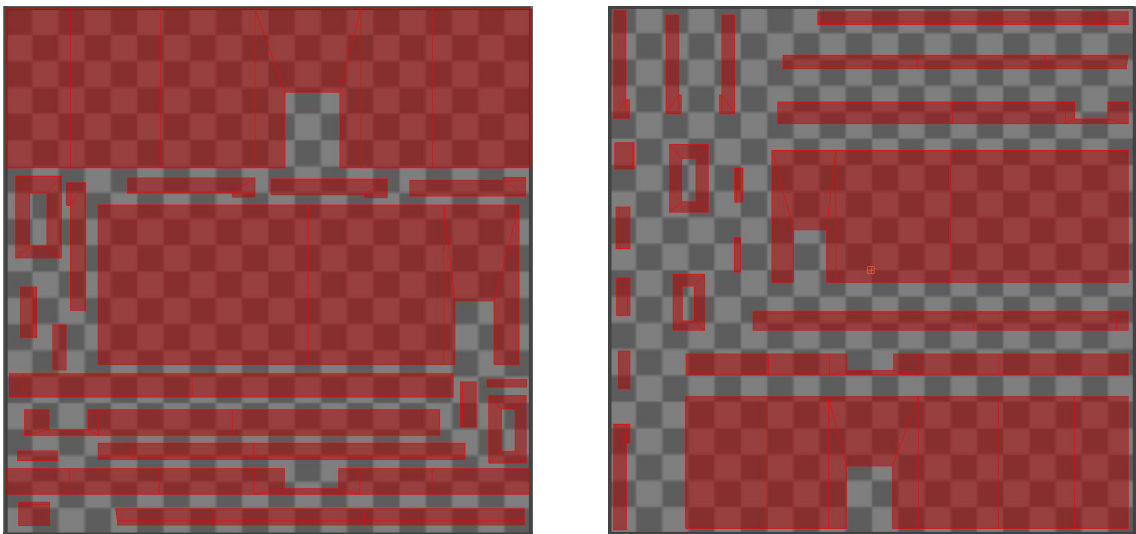
Unreal Engine používá UV souřadnice nejen pro modelové textury, ale také pro tvorbu doplňkových lightmap. Do těchto textur ukládá informace o statickém osvětlení, za účelem snížení výkonnostních požadavků aplikace. Lightmapy však ve většině případů, oproti klasickým modelovým texturám, potřebují jen malé rozlišení. Problémy pak mohou nastat, pokud rozložené části modelu na 2D plátně UV mapy jsou u sebe moc blízko. Pro modelové textury s velkým rozlišením to nemusí být problém, ale lightmapy, které mají svou texturu značně menší, mohou tvořit nežádoucí efekty v podobě přesahů jednotlivých tmavých pixelů v textuře, vytvořených k reprezentaci stínu, na části modelu, kde má být naopak světlo.

Tento problém jsem řešil vytvořením sekundárních UV souřadnic, které sloužily jen pro lightmapping daného modelu. U složitějších objektů jsem navíc zvyšoval rozlišení samotné lightmapy.

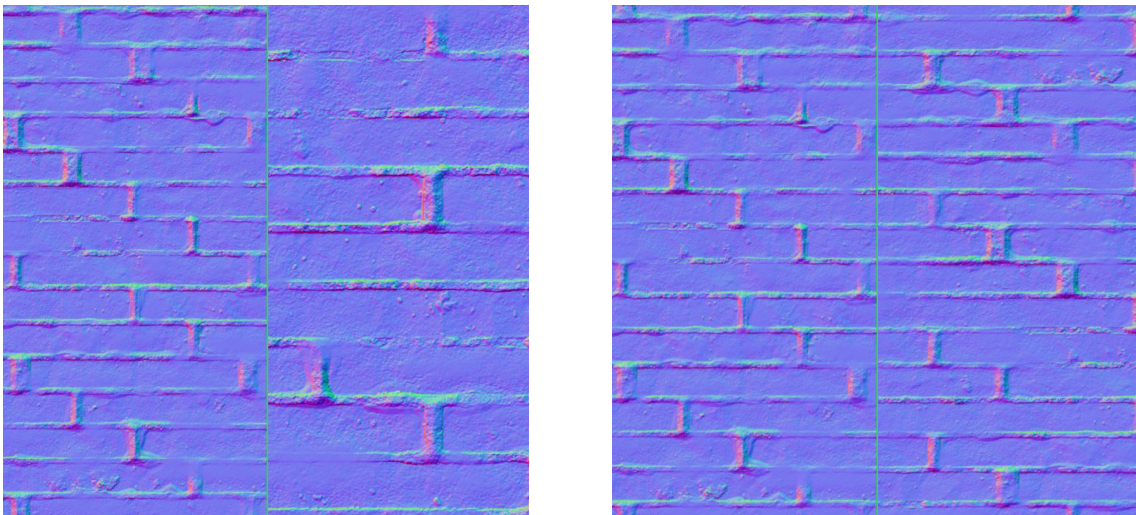
Pro co nejmenší nárůst výkonnostních požadavků aplikace, jsem se snažil co nejvíce modelů ve scéně texturovat pomocí opakujících se textur. Mezi tyto modely patří především zdi, dřevěná střecha a ocelová konstrukce. Zdem například musely být přesně nastaveny UV souřadnice tak, aby cihly v textuře na sebe dokonale navazovaly u modelů stěn, které byly vedle sebe.



Obr. 8, 9. Štěpán Brýdl, Ukázka UV mapy modelu dveří (neupravená vlevo, hotová vpravo)



Obr. 10, 11. Štěpán Brýdl, Ukázka uspořádání UV (normální UV vlevo, lightmap UV vpravo)



Obr. 12, 13. Štěpán Brýdl, Ukázka uspořádání UV map na dvou stěnách (nenačuzující UV vlevo, navazující vpravo)

5.7 Tvorba textur

S dokončenými UV mapami, bylo možné modely postupně importovat do aplikace Substance Painter. Zde byly modelům podle reálných referencí přidávány barvy, odlesky a nerovnosti povrchu. Textury jsem tvořil zejména ve velikostech 2048, 1024 a 512 pixelů, podle toho, jak moc detailní daný model potřeboval být v závislosti na jeho velikosti ve scéně a jak blízko se k němu divák může dostat.

Pro základní detaily modelů jsem používal procedurální masky, jako jsou například opakující se obdélníky, jednoduchý šum, pruhy a další. Pomocí těchto masek jsem tvořil například iluze prostoru mezi jednotlivými prkny dřeva, či základní vrstvu pro cihly na stěnách. Následně jsem na textury pomocí grafického tabletu maloval detailní prvky. Textury dřeva a podlahy jsem vytvořil z již předem vytvořených presetů v aplikaci Substance Painter, které jsem následně podle potřeby upravil.

Krom manuální tvorby bylo také potřeba vytvořit tzv. Ambient Occlusion texturu. Ta, je tvořena pomocí simulace neutrálního světla na model, a znázorňuje stíny v záhybech modelu bez nutnosti vykreslování světla v reálném čase, čímž dále umožňuje optimalizaci scény. Je také více kvalitní oproti stínům tvořeným v reálném čase a tím výrazně zvyšuje realističnost scény.

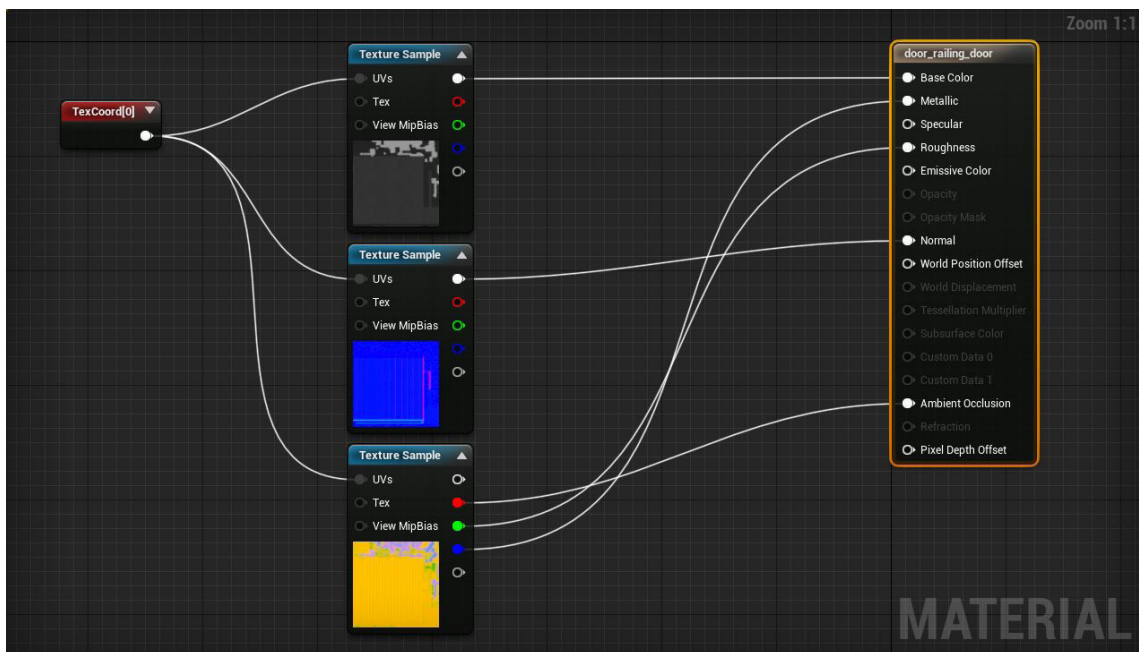
Pro modely, u kterých jsem si stanovil, že budou texturovány pomocí opakující se textury, bylo třeba vytvořit čtvercový shader, který by však při opakování vedle i pod sebe stále navazoval. Horní okraj textury tedy musel navazovat na spodní, a pravý okraj na levý.

Po dokončení tvorby textury pro daný model, jsem ve výběru pro export zvolil preset pro prostředí Unreal Engine a textury exportoval ve formátu .png a se zvolenou velikostí do připravené složky. Zde se následně uložily tři .png soubory, pojmenované: Base Color, Normal a OcclusionRoughnessMetallic. Díky tomuto pojmenování bylo snadné textury rozpoznat a dále s nimi pracovat v prostředí Unreal Engine.

5.8 Tvorba materiálů v Unreal Engine

Po importování textur do prostředí Unreal Engine z aplikace Substance Painter, bylo nutné ze tří importovaných souborů vytvořit jeden materiál. Base Color textura musela být vložena do Base Color vstupu v kombinovaném kanálu RGB, stejně tak Normal textura musela být vložena do Normal vstupu, také v kombinovaném kanálu RGB. Zatímco předchozí textury používaly kanál RGB pro jeden vstup, OcclusionRoughnessMetallic textura rozděluje Ambient Occlusion, Roughness a Metallic vstupy do jednotlivých kanálů RGB za účelem šetření místa na disku. Proto se také jednotlivé kanály musí podle svých vstupů zapojit odlišně od prvních dvou textur. Nakonec se pomocí TexCords modulu nastavil tiling, neboli opakování textur. Tímto modulem se ovládají opakující se textury vedle sebe, nebo se naopak nastavují přesné souřadnice u textur vytvořených právě pro daný model.

Hotové nastavené materiály se pak mohly aplikovat na jednotlivé modely. Pro některé modely bylo potřeba nastavit různé opakování textur pomocí TexCords, proto jsem některé materiály duplikoval, upravil a následně aplikoval na daný model.



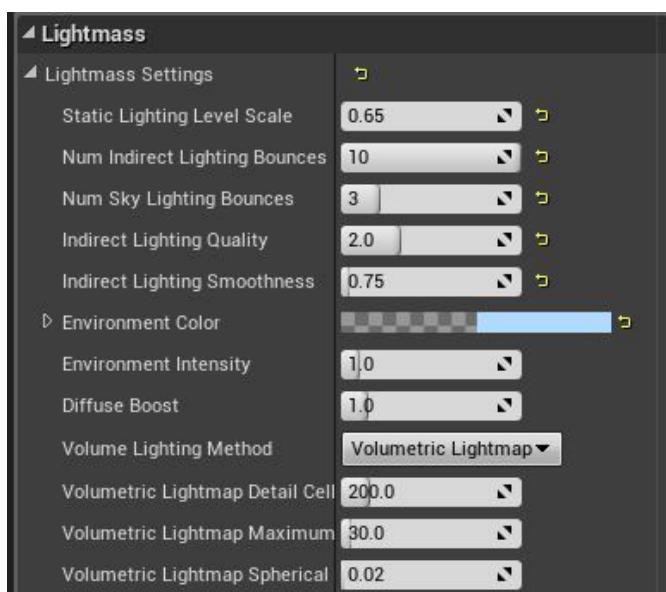
Obr. 14. Štěpán Brydl, Prostředí Unreal Engine Material Editor a ukázka tvorby materiálu

5.9 Finalizace

Po přiřazení materiálů k již umístěným modelům, bylo třeba upravit nastavení osvětlení scény tak, aby vypadalo co nejvíce realisticky. Jako primární zdroj osvětlení jsem nastavil směrové světlo, které funguje jako venkovní sluneční osvětlení a vrhá základní ostré stíny do budovy. Pro vyšší kvalitu stínů jsem také do všech oken umístil tzv. LightmassPortal, což je objekt, který při počítání světla říká hernímu enginu, kam má soustředit nejvíce simulovaných světelných částic a tím tvoří stíny uvnitř budovy detailněji. Zároveň jsem v nastavení zvýšil počet odrazů těchto částic, aby se světlo odráželo v prostoru vícekrát, a tak dopadlo i do tmavých rohů a koutů. V nastavení osvětlení jsem také snížil celkové měřítko statického světla a tím docílil menších a detailnějších stínů v rozích a záhybech předmětů po celé scéně.

S hotovým nastavením scény jsem opakovaně nechával vypočítávat světlo a podle výsledků jsem následně upravoval jak textury, tak rozlišení lightmap, abych se vyvaroval grafickým artefaktům nebo zbytečně velkým nárokům kvůli velkým texturám. Zároveň jsem upravil nastavení expozice a barevné tónování scény, abych scénu oživil.

Když bylo vše hotové, stačilo scénu nechat sestrojít a vyexportovat funkční aplikaci, pro spuštění na platformě Windows.



Obr. 15. Štěpán Brýdl, Nastavení pro výpočet statického světla v prostředí Unreal Engine



Obr. 16, 17, 18. Štěpán Brýdl, Ukázky finální aplikace

Závěr

V bakalářské práci jsem se zabýval tématem tvorby interaktivní vizualizace zobrazující galerijní prostor Dopravního Podniku v Hradci Králové. Myšlenkou bylo vytvořit funkční spustitelnou aplikaci, která by v reálném čase dokázala co nejreálněji zobrazit zvolený prostor.

Teoretická část práce krátce popisuje hlavní historické události, které vedly k vývoji 3D grafiky a jejímu dnešnímu používání v mnoha oborech. Dále zmiňuje hlavní aplikace, které se dnes k tvorbě používají a také pro které hlavní obory v dnešním průmyslu 3D slouží. Zde mi bakalářská práce přinesla mnoho nových znalostí ohledně vývoje a historie oboru, kterému bych se rád i nadále věnoval.

Praktická část práce mi byla velkým přínosem a myslím si, že plní svůj účel, který byl stanoven v zadání. Během tvorby jsem získal mnoho nových zkušeností a znalostí, které bych bez praktického provedení získával jen stěží. Nejen, že celý proces tvorby vizualizací v reálném čase vyžaduje mnoho znalostí, ale také je u větších prostorů nesmírně zdlouhavý. Na reálném projektu by z mého pohledu za ideálních okolností pracovali dva lidé. Jeden by tvořil modely a textury, druhý by pracoval v prostředí herního engine. Tím však nechci utvořit závěr, že člověk pracující jen s modely a texturami, by neměl být seznámen i s tvorbou v herním engine a naopak. Zkušenost projít si celým procesem tvorby sám mě donutilo řešit spoustu problémů, na které bych při tvorbě ve spolupráci s dalším člověkem nemusel narazit. Díky tomu jsem byl donucen dané problémy řešit, jak po modelové stránce, tak přímo v herním engine.

Je také důležité zmínit, že tvorba modelů pro využití v reálném čase je odlišná od tvorby pro statické rendery, kde není potřeba tolik myslet na výkonnostní nároky. Díky tomu jsem se naučil některé modelovací nástroje používat odlišně než doposud, a celkově se na modelování a texturování dívat z trochu jiné perspektivy.

Seznam použité literatury

[1] VAUGHAN, William. Digital modeling. Berkeley, CA: New Riders, c2012. ISBN 03-217-0089-9.

[2] RYAN, Daniel L. History of computer graphics. Bloomington, IN: AuthorHouse, 2011. ISBN 978-1456751173.

[3] GRESS, Jon. Visual Effects and Compositing. San Francisco, California: New Riders, [2015]. ISBN 03-219-8438-2.

[4] ŽÁRA, Jiří. Moderní počítačová grafika. 2., přeprac. a rozš. vyd. Brno: Computer Press, 2004. ISBN 80-251-0454-0.

[5] KŘÍŽ, Jan. Mistrovství v 3ds Max: [kompletní průvodce profesionálního grafika]. Brno: Computer Press, 2010. ISBN 978-80-251-2464-2.

[6] O'DOHERTY, Brian. Uvnitř bílé krychle: ideologie galerijního prostoru. Praha: tranzit.cz, 2014. ISBN 978-80-87259-30-6.

Seznam internetových zdrojů

- ² Oculus. Oculus Rift [online]. [cit. 2019-04-22]. Dostupné z: www.oculus.com/rift/
- ³ VIVE. VIVE [online]. [cit. 2019-04-22]. Dostupné z: www.vive.com/eu/product/
- ⁴ Sketchpad of Ivan Sutherland. History-Computer [online]. [cit. 2019-04-28]. Dostupné z: history-computer.com/ModernComputer/Software/Sketchpad.html
- ⁶ This week in history: Toy Story and CGI. Spiceworks [online]. [cit. 2019-05-13]. Dostupné z: community.spiceworks.com/topic/412181-this-week-in-history-toy-story-and-cgi
- ⁷ OUR STORY. Pixar [online]. [cit. 2019-05-13]. Dostupné z: www.pixar.com/our-story-1
- ⁸ Wired. The Untold Story of ILM, a Titan That Forever Changed Film [online]. [cit. 2019-05-5]. Dostupné z: www.wired.com/2015/05/inside-ilm/
- ⁹ Historie 3D grafiky. Fakulta informatiky Masarykovy univerzity [online]. [cit. 2019-05-15]. Dostupné z: www.fi.muni.cz/usr/jkucera/pv109/2001/xkolcar_index.html
- ¹⁰ 3D Visualization. Easy Render [online]. [cit. 2019-05-15]. Dostupné z: www.easy-render.com/architectural-visualization/is-3d-rendering-and-architectural-visualization-on-the-future-of-architectural-design
- ¹³ History of Autodesk 3ds Max. Times & Trends Academy [online]. [cit. 2019-05-15]. Dostupné z: timesandtrendsacademy.com/history-autodesk-3ds-max
- ¹⁴ Substance painter. Substance [online]. [cit. 2019-05-15]. Dostupné z: www.substance3d.com/products/substance-painter
- ¹⁵ Unreal Engine. Wikipedia [online]. [cit. 2019-05-15]. Dostupné z: en.wikipedia.org/wiki/Unreal_Engine
- ¹⁶ This is Unreal - The story of Unreal Engine and the games it powers. Hotgates [online]. [cit. 2019-05-15]. Dostupné z: hotgates.eu/this-is-unreal/

Seznam obrázků v textu

Obr. 1. Štěpán Brýdl, Ukázka 3D modelu ruky ve filmu Futureworld (1976)

Obr. 2. Štěpán Brýdl, Ukázka prostředí aplikace Substance Painter 2018

Obr. 3. Štěpán Brýdl, Ukázka prostředí aplikace Unreal Engine

Obr. 4. Štěpán Brýdl, Ukázka technických výkresů, převedených na křivky v prostředí 3ds Max

Obr. 5. Martin Tomáš, Fotografie prostoru Dopravního Podniku

Obr. 6. Martin Tomáš, Fotografie prostoru Dopravního Podniku

Obr. 7. Štěpán Brýdl, Ukázka objektů, se kterými může charakter kolidovat

Obr. 8. Štěpán Brýdl, Ukázka neupravené UV mapy modelu dveří

Obr. 9. Štěpán Brýdl, Ukázka hotové UV mapy modelu dveří

Obr. 10. Štěpán Brýdl, Ukázka normálního uspořádání UV mapy

Obr. 11. Štěpán Brýdl, Ukázka uspořádání UV pro lightmapping

Obr. 12. Štěpán Brýdl, Ukázka nenavazující UV mapy na dvou stěnách

Obr. 13. Štěpán Brýdl, Ukázka navazující UV mapy na dvou stěnách

Obr. 14. Štěpán Brýdl, Prostředí Unreal Engine Material Editor a ukázka tvorby materiálu

Obr. 15. Štěpán Brýdl, Nastavení pro výpočet statického světla v prostředí Unreal Engine

Obr. 16. Štěpán Brýdl, Ukázka finální aplikace

Obr. 17. Štěpán Brýdl, Ukázka finální aplikace

Obr. 18. Štěpán Brýdl, Ukázka finální aplikace

Seznam zkratk a pojmů

Workflow	Pracovní postup
UVW map	Způsob rozložení 3D modelu na 2D plátno
Engine	Stěžejní aplikace pro tvorbu počítačových her
Storyboard	Technické zobrazení příběhu v obrazech
VFX	Vizuální Efekty
Mesh	Drátěná část 3D modelu
Plugin	Zásuvný modul pro danou aplikaci
Rendering	Vykreslování obrazových dat
Retopologie	Proces vytváření kvalitnější sítě 3D modelu
Preset	Předem nastavené hodnoty
CGI	Počítačem tvořený obraz (Computer Generated Imagery)
Asset	3D model s texturami a nastavenými vlastnostmi
VR	Virtuální Realita