

**TECHNICKÁ UNIVERZITA V LIBERCI**



Fakulta mechatroniky, informatiky a mezioborových studií

**FOTOREALISTICKÁ SIMULACE AREÁLU HUSOVA V REÁLNÉM  
ČASE**

Bakalářská práce

Jiří Podlipný

Liberec 2013



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# FOTOREALISTICKÁ SIMULACE AREÁLU HUŠOVA V REÁLNÉM ČASE

## BAKALÁŘSKÁ PRÁCE

STUDIJNÍ PROGRAM: B2646 – INFORMAČNÍ TECHNOLOGIE

STUDIJNÍ OBOR:        INFORMAČNÍ TECHNOLOGIE

Autor práce  
Vedoucí práce

Jiří Podlipný  
Ing. Jiří Jeníček, Ph.D.

POČET STRAN TEXTU .....	37
POČET OBRÁZKŮ .....	17
POČET TABULEK .....	03
POČET PŘÍLOH .....	03

# Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

## Poděkování

Touto cestou bych rád poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Jeníčkovi, Ph.D. za metodickou pomoc, cenné rady a připomínky, které byly hodnotným přínosem při zpracování této bakalářské práce.

Dále děkuji vývojářům z týmu Crytek za jejich podporu na oficiálním fóru CryDev. Současně projevuji svůj vděk všem účastníkům diskusního fóra, kteří byli ochotni se podělit o své znalosti a dovést společný problém ke zdárnému řešení. Poděkování si též zaslouží firma Autodesk, jež nepřímo umožnila vznik této bakalářské práce, a to protože poskytuje své produkty pro studenty zcela zdarma.

# **Abstrakt**

Předmětem této bakalářské práce je vytvoření fotorealistické simulace areálu Technické univerzity v Liberci v herním enginu, a to v programu CryEngine ve verzi 3. Pro dosažení cíle práce bylo nutné nejprve provést přípravu, úpravu a vyhotovení podkladů pro tvorbu scény za pomoci nástrojů Autodesk 3ds max a Adobe Photoshop a následně tyto hotové podklady vyexportovat do prostředí CryEngine. Při tvorbě scény byl kladen velký důraz na metody 3D modelingu a texturování, jež jsou pro vytvoření bakalářské práce nepostradatelné. Výsledkem se stala kompletní fotorealistická vizualizace areálu TUL k roku 2011 včetně interiérů a interaktivních prvků, při které byl brán ohled na možné budoucí rozšíření.

## **Klíčová slova**

Fotorealistická vizualizace, CryEngine, modelování, texturování, interakce s okolím

# **Abstract**

The subject of this thesis is to create photorealistic simulations of the areal Technical University of Liberec in the game engine, specifically in the CryEngine 3. To achieve the objective of this thesis was necessary to perform the preparation and editing materials for the creation of a scene using Autodesk 3ds Max and Adobe Photoshop and subsequently export these materials into CryEngine 3. In the creation a scene is a great emphasis on methods of 3D modeling and texturing, which are indispensable to the creation of this thesis. The result of this thesis is complete photorealistic visualization of the areal TUL by the year 2011, including interiors and interactive elements, in which consideration was given to possible future expansion.

## **Klíčová slova**

Photorealistic visualization, CryEngine, modeling, texturing, interaction with the environment

# Obsah

Prohlášení.....	5
Poděkování.....	6
Abstrakt.....	7
Abstract.....	8
Seznam symbolů, zkratk a termínů .....	11
Seznam obrázků.....	13
Seznam tabulek .....	13
1 Úvod.....	14
2 Základní používané softwarové nástroje .....	16
2.1 CryEngine 3 .....	16
2.2 Autodesk 3ds max 2012.....	17
2.3 Google SketchUp .....	18
2.4 Adobe Photoshop .....	18
2.5 PixPlant .....	18
3 Vývojové prostředí CryEngine 3 SDK .....	19
3.1 Systémové nároky a použitá sestava .....	19
3.2 Součásti a adresářová struktura enginu .....	20
3.3 Používané technologie CryEnginu 3 .....	24
3.3.1 Vizuální stránka enginu.....	24
3.3.2 Fyzikální stránka enginu .....	29
3.3.3 Sandbox editor.....	29
3.4 Pluginy pro export do CryEngine 3 .....	30
3.5 Základní stavební prvky v CryEngine 3 .....	31
3.5.1 Solid.....	31
3.5.2 Brush .....	32
3.5.3 Entita .....	32
3.5.4 Group.....	32
3.5.5 Prefab.....	32
3.5.6 Archetype .....	33
3.5.7 Road.....	33
3.5.8 Decal.....	34

4 Vizualizace areálu Technické univerzity .....	35
4.1 Tvorba základních prvků ve scéně.....	35
4.1.1 Analýza a tvorba terénu.....	35
4.1.2 Editace, optimalizace a tvorba modelů.....	38
4.1.3 Tvorba textur a jejich aplikace .....	39
4.2 Vytváření scény v CryEngine 3 .....	42
4.2.1 Export modelů, chyby při exportování a jejich obcházení .....	42
4.2.2 Import do Cryengine 3 a umístění ve scéně .....	44
4.3 Interaktivní prvky.....	45
4.3.1 Simulace oken a dveří .....	46
4.3.2 Základní simulace vozidla.....	46
4.4 Neoficiální problémy a jejich řešení .....	48
5 Závěr .....	51
Literatura.....	<b>Chyba! Záložka není definována.</b>
Příloha A – Obsah přiložených DVD .....	<b>Chyba! Záložka není definována.</b>
Příloha B – První spuštění editoru .....	<b>Chyba! Záložka není definována.</b>
Příloha C – Obrázky hotové aplikace .....	58



## Seznam symbolů, zkratek a termínů

Alfa channel	Alfa kanál
Auto Smooth	Automatické vyhlazování
Blend layer	Mixované vrstvy
Brush	Štětec
Clone stamp tool	Razítko pro klonování
Contrast & Brightness	Kontrast a jas
Direct lighting	Přímé osvětlení
Events	Události
FPS	First person shooter (v překladu střílečí hra z pohledu první osoby)
Game logic	Herní logika
Global Lighting	Globální osvětlení
Group	Skupina
HDR	High Dynamic Range Lighting (v překladu Vysoce dynamický rozsah osvětlení)
Heightmapou	Výšková mapa
Hue & Saturation	Odstín a saturace
In-game	Ve hře)
Luminance & Color Curves	Svítivost a barevné křivky
Noise	Šum
Normal mapping	Normálové mapování
Planes	Plošky nebo také roviny
Polygon mesh	Kolekce bodů (vertices), hran (edges) a plošek (faces), které reprezentují tvar 3D objektu. V tomto textu se budu o polygon mesh zmiňovat jako o mesh
Problem Degenerate faces	Problém zdegenerovaných plošek
Proximity Trigger	Senzor přiblížení
RC	Resource Compiler
Smoothing groups	Vyhlazovací skupiny
Source asset	Zdrojový prvek
Spline	Křivka

SSAO	Screen Space Ambient Occlusion
SSDO	Screen Space Directional Occlusion
Target Asset	Cílový prvek
Tessellation	Teselace
Time of Day System	Systém nastavení denního času
Triggers	Spínače
UDK	Unreal Development Kit
WYSIWYP	What You See Is What You Play (v překladu Co vidíte, to hrajete)

## Seznam obrázků

Obr. 1 Systém Direct lighting a osvětlení Global Lighting .....	25
Obr. 2 Postup teselace .....	27
Obr. 3 Postup při displacement mapping.....	28
Obr. 4 3D objekt terénu areálu TUL.....	36
Obr. 5 Zarovnání v programu Mudbox.....	37
Obr. 6 Importovaný terén v CryEngine .....	38
Obr. 7 Textura bez ošetření tilingu .....	40
Obr. 8 Textura s opraveným tilingem .....	40
Obr. 9 Textura bez Parallax occlusion maping a textura s Parallax occlusion maping ....	41
Obr. 10 Selekce faces.....	42
Obr. 11 Problém černých fragmentů.....	43
Obr. 12 Zobrazení s problémem normálových vektorů a zobrazení s jeho ošetřením ...	44
Obr. 13 Proximity Trigger a jeho funkcionalita v Flow Graph .....	46
Obr. 14 Rozložení komponent na vozidle .....	47
Obr. 15 Seskupení jednotlivých komponent.....	47
Obr. 16 Ztráta informací o obarvení textur.....	48
Obr. 17 Problém červených kol .....	49

## Seznam tabulek

Tab. 1 Systémové nároky pro vývojáře .....	19
Tab. 2 Systémové nároky pro koncové uživatele .....	19
Tab. 3 Použitá sestava.....	20

# 1 Úvod

S rozvojem počítačové grafiky a systémů pro tvorbu herních aplikací se objevují nové možnosti vizualizace, které se stále více přibližují fotorealistickému vjemu. Tyto nástroje lze použít nejen pro vytváření počítačových her, ale také pro tvorbu filmů, simulací reálného světa nebo jako prostředek pro prezentaci v architektuře. Díky dnešním kompletním vývojovým prostředím, jež nám vizualizaci dovolují tvořit, spouštět a zároveň se starají o její chod, odpadá značná starost s návrhem vlastního řešení. Na základě tohoto hlediska se lze zabývat již pouze vizuální stránkou dané aplikace, v důsledku čehož dochází ke zvýšení kvality celého konečného řešení.

Hlavním cílem této bakalářské práce je vytvořit, právě za pomoci zvoleného programu pro tvorbu počítačových her, fotorealistickou vizualizaci areálu Husova Technické univerzity v Liberci (dále jen TUL) v reálném čase. Před samotným realizováním stanoveného úkolu byla provedena analýza herních vizualizačních nástrojů dostupných na celosvětovém trhu s cílem zvolit ten nejvhodnější pro tento účel. Při zjišťování informací byla zaměřena hlavní pozornost na to, aby nástroje byly dostupné zdarma a nabízely přístupnou a srozumitelnou dokumentaci, jež zajistí dosažení relevantních znalostí nezbytných pro vytvoření této bakalářské práce. Mimo těchto parametrů bylo též sledováno, zda obsahují editor a samostatnou aplikaci pro spuštění. Těmto požadavkům vyhovovaly čtyři základní herní enginey – Unity3D, Unreal Development Kit (dále jen UDK), Source Engine a CryEngine 3.

Systém Unity3D byl však automaticky vyřazen, neboť se orientuje zejména na tvorbu aplikací pro mobilní segment, díky čemuž by nemohl dosahovat takových kvalit fotorealistické vizualizace jako ostatní šetřené enginey. Program UDK byl označen na základě zjištěných informací za příhodný engine, ale z hlediska jeho prostředí výrazně založeného na sci-fi prvcích by též nepřinášel nejlepší možné řešení pro fotorealistickou vizualizaci. Další možné varianty představovaly Source Engine a CryEngine 3, jež by potřebám této bakalářské práce zcela vyhovovaly. Na základě komparace zjišťovaných informací o těchto dvou programech bylo zjištěno, že konkurenční systém CryEngine 3 dosahuje na poli fotorealistické vizualizace mnohem lepších výsledků, tudíž byl zvolen jako nejlepší možné řešení pro zpracování bakalářské práce herní engine CryEngine 3 ve verzi 3.4.4. Dalším důvodem pro výběr zmíněného engineu bylo také kompletní hotové řešení scény s již nastavenými

světelnými podmínkami, které se pouze upraví podle potřeby uživatele. Tuto možnost ostatní zkoumané programy nenabízí, což lze považovat za nedostatek, neboť tato funkce zajišťuje vysoce kvalitní osvětlení scény a větší fotorealistický efekt.

Pro dosažení hlavního cíle je bakalářská práce rozdělena do tří kapitol, které na sebe vzájemně navazují. První část se věnuje nástrojům využívaným při tvorbě fotorealistické vizualizace areálu TUL. Druhá kapitola nabízí deskripci programu CryEngine 3, a to konkrétně jeho nároky, využívané technologie v této práci a základní prvky při tvorbě scény v prostředí CryEngine. Hlavní část bakalářské práce je představena ve třetí kapitole, kde je nastíněno modelování budov, jejich úprava a také doplnění o detaily, dále je zde prezentováno texturování zhotovených modelů budov a jejich export do prostředí CryEngine. V této části je též demonstrován ojedinělý postup vytváření terénu založený na reálných datech, jenž využívá výškových dat o nadmořské výšce. Na závěr této kapitoly je stručně popsána samotná tvorba areálu Husova v prostředí CryEngine. Dále zde jsou vysvětleny problémy, které se vyskytovaly při tvorbě bakalářské práce, a zároveň je podán návrh na jejich řešení.

Vytvořená fotorealistická simulace areálu TUL je velmi náročná na výpočetní a grafický výkon, z tohoto důvodu by měla být spuštěna na počítači, který splňuje alespoň doporučené hardwarové nároky, jež jsou patrné z kapitoly 3. Uživatel, který vlastní systém Microsoft Windows 8, tuto práci momentálně nespustí v režimu editor ale pouze v režimu launcher, a to protože Crytek odmítl podporu systému Microsoft Windows 8. Tuto bakalářskou práci se doporučuje spouštět v editoru, neboť pro launcher není kompletně doladěno osvětlení. Jedná se o jednu z chyb CryEnginu 3, která je opravena v placené verzi. Však ve verzi, která je k dostání zdarma, nastává absence multiplikace osvětlení a změna parametrů odrazivosti světla, díky čemuž dochází i k přesvícení nebo přílišnému tmavnutí objektů ve scéně.

Pro vytvoření této bakalářské práce bylo nezbytné pojmout celou řadu informací, které byly získány z knihy 3ds Max Modeling for Game [14], dále z knihy CryENGINE 3 Game Development [25] a také z konzultací na oficiálním diskusním fóru. Velkým přínosem při tvorbě fotorealistické simulace areálu TUL byla e-mailová komunikace přímo s vývojáři CryEngine 3. Jelikož problematika řešená v této práci běžně používá anglickou terminologii, bylo naprosto nezbytnou podmínkou jí využívat i v následujícím textu. České ekvivalenty byly uvedeny v seznamu symbolů, zkratk a termínů.

## 2 Základní používané softwarové nástroje

Na úvod této bakalářské práce je důležité zmínit jaké byly hlavní používané softwarové nástroje při vytváření fotorealistické simulace areálu TUL, zejména se jednalo o CryEngine 3, Autodesk 3ds max 2012, Google SketchUp, Adobe Photoshop a PixPlant.

### 2.1 CryEngine 3

Program CryEngine 3, vytvořený německou společností Crytek, je herní engine, který slouží převážně k tvorbě FPS<sup>1</sup>. Díky jeho rozsáhlým možnostem modifikace a přímému přístupu k jádru enginu<sup>2</sup> může být používán i jako nástroj pro simulaci fyzikálních vlivů na okolí nebo objekty. Typickou ukázkou takového užití je například projekt od studia BeamNG [2], které vytvořilo pokročilou simulaci fyzických vlastností vozidel. Z této simulace se nakonec stal zásuvný modul pro CryEngine 3, jež se uplatňuje jako nástroj pro tvorbu vozidel.

Mezi další možnosti využití patří tvorba multimediálních záznamů pomocí pluginu Cryengine Cinebox, či dokonce vizualizace exteriérů a interiérů v architektuře, kde má zákazník (uživatel) možnost projít si návrh v reálném čase a na základě toho se dále rozhodovat. Lze v něm vytvářet také tzv. Serious Games<sup>3</sup>, jež jsou nepostradatelné například pro americkou armádu, která je využívá nejen pro trénink soubojů a pro nácvik taktiky v konfliktních situacích, ale také jako simulátor ovládání vojenské techniky (letadla a helikoptéry).

První verze CryEngine byla původně vyvinuta Crytekem jako technické demo pro společnost Nvidia, ale poté co byl spatřen potenciál a konkurenceschopnost tohoto enginu, došlo k jeho dalšímu vývoji a následnému použití pro tvorbu herních aplikací. CryEngine 1 byl krátce po jeho uvedení na trh roku 2004 použit při tvorbě hry FarCry, která tento engine proslavila a zajistila jeho výrazné rozšiřování na trhu. S nástupem grafických karet, jež podporovaly technologie pixel a vertex shader ve verzi 3.0, byl vypuštěn CryEngine 1.2, který v té době jako jediný na trhu tyto technologie využíval. Mimo to také pouze CryEngine 1.2 v daném období přišel s podporou technologie polybump, která sloužila pro plastičtější vykreslování textur. Díky schopnosti Cryteku

---

<sup>1</sup> FPS čili „first person shooter“ znamená v překladu střílečí hra z pohledu první osoby.

<sup>2</sup> Dané možnosti jsou nabízeny pouze v placené verzi.

<sup>3</sup> Serious Games znamená v překladu vážné hry.

a CryEnginu podporovat nejnovější technologie se stal velmi oblíbeným mezi herními vývojáři. Například společnost Ubisoft koupila veškerá duševní vlastnictví na herní aplikaci FarCry a s tím i možnost pro budoucí použití CryEnginu pro jiné herní aplikace [23].

V roce 2007 byl představen CryEngine ve verzi 2, jenž byl použit k tvorbě proslulé herní aplikace Crysis a také byl poprvé licencován francouzské společnosti IMAGTP [6], která se specializuje na architektonické a urbanistické plánování komunikací. Důvodem k licencování bylo vytvořit program, jenž by dokázal zákazníkům prezentovat hotovou stavbu ještě před samotným započítáním výstavby.

Nástupcem CryEngine 2 se v roce 2009 stal CryEngine 3, který byl vyvinut pro tvorbu vizualizačních řešení jak pro PC, tak pro Xbox 360, PlayStation 3 a WiiU. Zároveň byla také oznámena podpora DirectX 9, 10 a 11 a bylo vydáno technologické demo Cascaded Light Propagation Volumes for Real Time Indirect Illumination, které prezentovalo novou technologii pro zpracování osvětlení ve scéně. Díky této technologii přišel CryEngine na trh jako první engine, který používal metodu WYSIWYP<sup>4</sup>, která umožňuje rychlé přepínání mezi editorem a navrhovanou vizualizací bez nutnosti zdoluhavého propočítávání parametrů osvětlení ve scéně. Zároveň v roce 2011 australské ozbrojené síly jako první oznámily, že jejich personál bude trénován na virtuálním simulátoru přistávání vrtulníkových strojů na výsadkových lodích, který běží na technologii CryEngine 3 [13].

## 2.2 Autodesk 3ds max 2012

Jako editor pro tvorbu modelů, jejich texturování a exportování byl použit při zpracování této bakalářské práci program Autodesk 3ds max ve verzi 2012. Uplatnění tohoto softwaru bylo nutné z hlediska přímé podpory ze strany společnosti Crytek pro svůj engine CryEngine 3. Firma doporučuje jako hlavní nástroj 3ds max v nejnižší verzi 2009. Mezi dalšími podporovanými nástroji jsou Maya a z neoficiální strany programy Blender a Google SketchUp. V období tvorby fotorealistické simulace areálu Husova TUL docházelo u Blenderu k poruchám v pluginu CryBlend pro export do CryEngine 3, díky čemuž bylo znemožněno jeho využívání při zpracování

---

<sup>4</sup> WYSIWYP - What You See Is What You Play v překladu Co vidíte, to hrajete

bakalářské práce. Jako alternativní editor byl použit Google SketchUp 8 a neoficiální plugin pro exportování do CryEngine 3.

## 2.3 Google SketchUp

Tento editor byl použit jako zdroj některých doplňkových modelů ze sítě Google Warehouse<sup>5</sup> a nástroj pro jejich následný export. Jedná se například o modely okolních budov na dotváření fotorealističtějšího vjemu v exteriérech, ale také o některý nábytek v interiérech budov TUL.

## 2.4 Adobe Photoshop

Pro tvorbu a úpravu textur byl zvolen nástroj Adobe Photoshop, neboť je jediným oficiálně podporovaným nástrojem pro tuto činnost. Další možnou alternativou bylo použití programu Gimp, jehož předností je využití kompletně zdarma. Bohužel pro Gimp není oficiální ani neoficiální exportní nástroj, z toho hlediska dochází k absenci komprese textur přímo pro potřeby CryEnginu.

## 2.5 PixPlant

Jedná se o nástroj, který slouží k tvorbě a generování textur na základě grafického podkladu. V této bakalářské práci slouží tento program nejen pro tvorbu textur ale zejména pro zrychlení práce s texturami a s jejich úpravou. Zejména je využito funkce pro odstranění problému, který je označován jako tiling, více o tilingu v kapitole 4.1.3 této bakalářské práce.

---

<sup>5</sup> Google Warehouse je webová aplikace kde je možná nahrávat, sdílet a stahovat modely vytvořené v programu Google SketchUp.



## 3 Vývojové prostředí CryEngine 3 SDK

### 3.1 Systémové nároky a použitá sestava

Jak je patrné z těchto tabulek, CryEngine 3 SDK je velmi nenáročný herní editor a umožňuje práci i na levnějších a méně výkonných sestavách. Tabulka 1 s nároky pro vývojáře se oproti tabulce 2 s nároky pro koncového uživatele liší pouze v hodnotách velikosti operační paměti, a to protože uživatel spouští pouze Launcher, nikoliv Editor.

Tab. 1 Systémové nároky pro vývojáře [5]

<b>Operační systém</b>	Windows XP SP2, Windows Vista SP1, Windows 7
<b>Procesor</b>	Intel Core 2 Duo 2GHz, AMD Athlon 64 X2 2GHz nebo lepší (doporučeno vícejádrový procesor)
<b>Operační paměť</b>	2 GB RAM (doporučeno 4 GB RAM)
<b>Grafická karta</b>	NVIDIA 8800GT 512MB RAM, AMD 3850HD 512MB RAM nebo lepší

Tab. 2 Systémové nároky pro koncové uživatele [5]

<b>Operační systém</b>	Windows XP SP2, Windows Vista SP1, Windows 7
<b>Procesor</b>	Intel Core 2 Duo 2GHz, AMD Athlon 64 X2 2GHz nebo lepší (doporučeno vícejádrový procesor)
<b>Operační paměť</b>	1 GB RAM (doporučeno 2 GB RAM)
<b>Grafická karta</b>	NVIDIA 8800GT 512MB RAM, AMD 3850HD 512MB RAM nebo lepší

Ve třetí tabulce jsou parametry sestavy, která byla použita pro tvorbu této bakalářské práce. Tato sestava splňuje systémové nároky s velkou rezervou, díky tomu bylo možné ve scéně docílit velkého množství detailů a přiblížit se k fotorealistickému vjemu.

Tab. 3 Použitá sestava

<b>Operační systém</b>	Windows 7
<b>Procesor</b>	Intel Core i5 3570K 3,40 GHz
<b>Operační paměť</b>	16 GB RAM
<b>Grafická karta</b>	NVIDIA Geforce GTX 460 SE 1 GB RAM

## 3.2 Součásti a adresářová struktura enginu

CryEngine 3 se skládá ze tří základních částí – Launcher, Sandbox Editor a samotný engine. Tyto části jsou zařazeny do jednotlivých složek a díky tomu je možné jednoduše a přehledně s CryEnginem manipulovat, například při dokončení projektu lze jednoduchým smazáním a editací konfiguračního souboru vytvořit samostatnou funkční aplikaci. V následujících podkapitolách si představíme, co jednotlivé adresáře obsahují a jaké mají funkce.

### Root

Jedná se o adresář, ve kterém je samotný CryEngine nainstalován, a co obsahuje adresáře v následujících podkapitolách. Jsou zde obsaženy základní logy, jež tvoří engine při výpočtech osvětlení, geometrického modelu, textur atd. Dále obsahuje zprávy o chybových hlášeních a soubor – systém.cfg. Tento soubor slouží pro nastavování parametrů ke spuštění CryEnginu a pro nastavování parametrů samotného enginu (př. nastavení rozlišení nebo běh na celé obrazovce). V této bakalářské práci se užívá například při testování na sestavách, které nepodporují DirectX 10 nebo 11 a slouží k tomu příkaz `r_driver=DX9` [4].

### Bin32 a Bin64

Bin32 a Bin64 obsahují základní spustitelné aplikace. Jedná se o 32bit a 64bit Sandbox editor (**Editor.exe**) a game launcher (**Launcher.exe**). Dále jsou zde obsaženy veškeré DDL knihovny, které tyto aplikace využívají a nutně potřebují ke svému provozu. Bin32 také obsahuje podadresář s názvem rc (**Resource Compiler**). Jedná se o nástroj, který zajišťuje komunikaci sandbox editoru a game launcheru se samotným enginem. Zajišťuje komunikaci mezi externími programy pro export a optimalizování

výstupních formátů a operace se vstupy sandbox editoru, jenž provádí automaticky na pozadí (př. převod TIF textur do formátu DDS, pro který je CryEngine optimalizován) [4].

## Code

V tomto adresáři jsou obsaženy všechny projekty a zdrojové kódy psané v jazyce C++, které nám po zkompilování vytvoří knihovnu GameDDL, která může upravovat a nastavovat vzhled a parametry engine. Hlavní projekt pro CryEngine 3 se nachází v adresáři Code/Solutions/ s názvem CryEngine.sln. Jedná se o projekt pro placenou verzi CryEnginu, který obsahuje veškeré hlavičkové soubory a zdrojové kódy, jež jsou potřeba ke kompletní správě a volitelné změně engine. Jelikož v této bakalářské práci používáme licenci, která je poskytována zdarma, tento soubor nemáme k dispozici. Z tohoto důvodu musíme použít projektový soubor CryEngine\_GameCodeOnly.sln, který je obsažen v adresáři Code\Solutions\. Tento soubor je značně ořezán a neumožňuje nám nijak zásadně zasahovat do kódu engine nebo launcheru. Díky tomu máme pouze možnost měnit vzhled výsledné aplikace spouštěné v launcheru a vytvářet jednotlivé funkce, které běží v prostředí CryEngine. Tyto funkce se nazývají Scaleform a lze je použít například pro vytvoření in-game menu nebo jiných aplikací, které přímo nezasahují do chování a běhu engine [4].

## Editor

Adresář s názvem editor obsahuje základní zdroje dat, které používá Sandbox editor. Jedná se o ikony, schémata vzhledu, UI data, nastavení a skripty vytvořené v samotném editoru. Výhoda osamostatnění těchto dat do jediné složky spočívá při vytvoření samostatně spustitelné aplikace, kdy není těchto dat nadále potřeba, a tím se skýtá možnost jejich smazání a zmenšení výsledné aplikace. Více o Sandbox editoru v kapitole 3.3.3 této bakalářské práce [4].

## Engine

Adresář se samotným enginem zahrnuje balíčky pak, ve kterých jsou obsaženy jednotlivé DDL knihovny, které zajišťují chod engine a poskytují nástroje pro jeho správu a použití. Jedná se o balíčky:

- **Engine.pak** – umožňuje nastavit konfiguraci engine pomocí textových souborů pro různá schémata režimu běhu (například v LOW režimu, kdy se vypne DirectX 11 a zapne se pouze DirectX9, tím se sníží nároky na výsledný grafický výkon),

- **ShaderCache.pak** – obsahuje Shadery<sup>6</sup> uložené z minulého spuštění enginu a tím zajišťuje jeho rychlejší znovunačtení,
- **Shaders.pak** – definuje chování materiálů, které jsou aplikovány na jednotlivé shadery (například určuje, zda se jedná o terrain material, Illum material, Water materiál aj.),
- **ShadersBin.pak** – zde jsou obsaženy všechny binární shadery, ke kterým se vážou nějaké funkce a nastavené parametry. Jedná se například o animace (natáčení vegetace ve větru nebo pohyb mraků) [4].

## Game

V tomto adresáři jsou obsažena všechna data, modely, textury a další soubory využívané k tvorbě této bakalářské práce. Jméno tohoto adresáře je možné změnit nebo vytvořit nový v root adresáři, ale existuje neoficiální pravidlo dodržet jméno tohoto adresáře a umisťovat vše do něj. Pokud bychom se rozhodli pro změnu, lze ji provést pomocí příkazu `sys_game_folder` v konzoli editoru. V následujících řádcích si uvedeme, jaké podadresáře game adresář obsahuje a jaké typy souborů jsou v nich obsaženy.

- **Animations** – reprezentují objekty animací a data k nim (například postava uživatele ve scéně).
- **Entities** – soubory s příponou ent., které představují entity a skripty, jež entity používají. Více o entitách v kapitole 3.5.3 této bakalářské práce.
- **Levels** – všechny levely (scény), které aplikace obsahuje. V této bakalářské práci je použit pouze jeden level, kterým je celý areál TUL, přesněji plocha 1024 na 1024 metrů okolo areálu TUL.
- **Libs** – knihovny a xml skripty, které umožňují nastavení dodatečných parametrů pro funkce v editoru a enginu.
- **Fonts** – písma a TrueType font používaný v editoru a launcheru.
- **Materials** – xml materiály, které lze použít v levelu.
- **Objects** – všechny statické objekty, lze k nim umístit také textury a materiály pro větší přehlednost [4].

---

<sup>6</sup> Shader – „Shadery jsou plně programovatelné součástí grafického čipu, které se společně s několika dalšími jednotkami starají o zpracování obrazu, který poté jako hotový posílají do grafické paměti a vidíme jej na monitoru.“ [22]

- **Prefabs** – prefabs uložené ve formátu xml, více o nich v kapitole 3.5.5 v této bakalářské práci.
- **Scripts** – základní skripty, které nastavují chování jednotlivých objektů (například jsou zde v podadresáři vehicles uloženy skripty, jež definují chování aut v této bakalářské práci).
- **Sounds** – všechny zvuky a jejich nastavení.
- **Textures** – obsahuje všechny textury, které jsou uloženy ve formátu TIF a pomocí nástroje Resource Controler jsou převedeny do formátu DDS, který používá CryEngine.

V hotové aplikaci je možné obsah těchto adresářů zahrnout do samostatných \*.pak souborů, tím je zapouzdřit a zamezit uživateli v jejich editování. Jedinou výjimkou jsou soubory pro level, které reprezentují samotné scény a jsou nutné pro jejich spuštění a zavedení zapouzdřených \*.pak balíčků. K vytvoření se doporučuje nepoužívat externí nástroje, ale použití Resource Compileru, který dokáže vytvoření \*.pak balíčků automatizovat a tím i zamezit možnosti výskytů chyb (například missing dll – v překladu chybějící knihovna) [4].

## TestResults

Obsahuje výsledky testů, které jsou generované v Sandbox editoru využitím příkazové řádky pomocí příkazu **SaveLevelStats**. Z výsledků textu lze zjistit chyby v texturách, modelech, terénu a dalších. Tato funkce bohužel ve verzi CryEngine 3.4.4 není funkční, a to z důvodu špatné implementace v této verzi. Tudíž je nutno využít její značně omezenou alternativní variantu, která je přímo zabudována v Sandbox editoru [4].

## Tools

Obsahuje velké množství samostatně spustitelných nástrojů pro správu CryEnginu 3 (například pluginy psané v jazyce Python pro kontrolu geometrie 3DS modelů a jejich následný export do formátu CGF podporovaným CryEnginem), nástroje pro export textur (**CryTIF**) pro program Adobe PhotoShop a nástroje pro export modelů z programů 3ds max (2010 a novější) a Maya (2009 a novější) [4].

## User

Zde jsou uložena data, která jsou generována průběžně při práci v Sandbox editoru. Tato data zahrnují informace o uživatelském profilu, dále dočasné shadery, jež

byly nejvíce používané při posledním spuštění (pro rychlejší načtení scény v Sadox editoru) a dočasné soubory generované Sandbox editorem, jako jsou například makra, změny nastavení globálního osvětlení a další [4].

## 3.3 Používané technologie CryEnginu 3

### 3.3.1 Vizuální stránka enginu

V této části budou představeny technologie, které jsou použity pro znázornění vizuální stránky této bakalářské práce a slouží k vytvoření fotorealistického vjemu.

#### **Natural Lighting & Dynamic Soft Shadows**

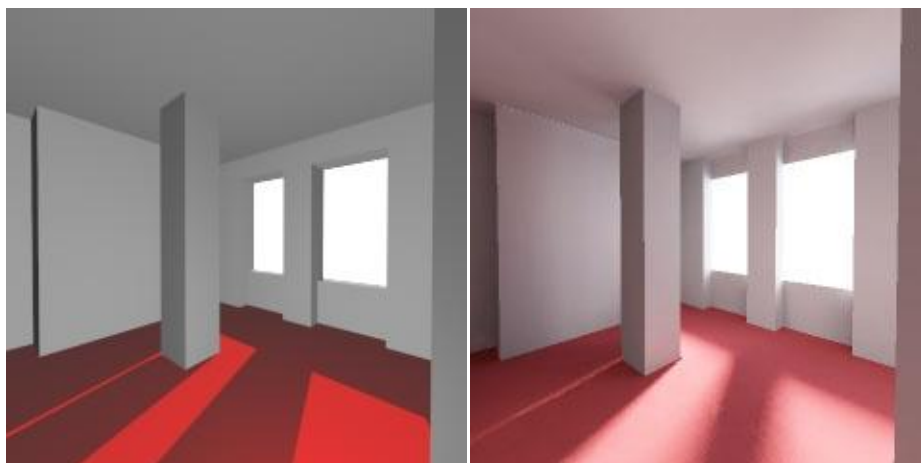
Tato technologie zajišťuje přirozený systém osvětlení, který má velmi nízké hardwarové nároky na použitou sestavu a vytváří dynamické stíny, jež reagují na okolní světlo a interakci uživatele s okolím. V této bakalářské práci se uplatňuje především v osvětlení exteriérových scén a zajišťuje dosažení fotorealistického efektu s prakticky nulovým dopadem na výkon ve scéně. Při vypnutí dochází ke zvýšení výkonu v řádu několika procent, ale zároveň dochází ke ztrátě vykreslování stínů a absenci interakce osvětlení s okolím a celkové ztrátě kvality zobrazení [9].

#### **Irradiance Volume**

Irradiance Volume je technologie, která slouží k vytváření osvětlení scén a má minimální dopad na snížení výkonu ve scéně. Zajišťuje simulaci radiozity<sup>7</sup> a obarvení odraženého světla. V této bakalářské práci se uplatňuje především při osvětlování interiéru a zajišťuje měkčí přechod mezi světlem a stínem. Toto je patrné z obrázku č. 1, kde vlevo je použito pouze systému Direct lighting a vpravo systému radiozity a osvětlení pomocí Global Lighting [9].

---

<sup>7</sup> Radiozita je systém globálního osvětlení scény, který se používá při renderování a vykreslování světla ve scéně [26].



Obr. 1 Systém Direct lighting a osvětlení Global Lighting [26]

## Real-time Dynamic Global Illumination

Podobně jako Irradiance Volume slouží k vytváření osvětlení, ale neváže se na osvětlování scény, pouze na osvětlování objektů ve scéně (jak statických, tak i dynamických). Taktéž je interpretován systémem radiozity, jenž je vypočítáván v reálném čase, což dělá z CryEngine 3 jeden z prvních enginů, který tuto technologii takto interpretoval [9].

## Real-time Local Reflections

Reflexe jsou jednou z nejvíce výpočetně náročných metod při vytváření vizualizací, z toho důvodu se řeší předem vypočítanou nebo vykreslenou reflexí uloženou v rastrovém souborovém formátu jako TIFF (CryEngine) nebo Targa (Unreal Engine) nebo jiném. Takto uložená rastrová reflexe je velmi nenáročná a často dostačující, ale s příchodem DirectX 11 je možné použít technologii Real-time Local Reflections, která funguje na principu metody ray tracing<sup>8</sup> a vypočítává se pouze tehdy, když je v aktivním zorném poli kamery (uživatele). Z tohoto důvodu se nedoporučuje tuto metodu používat ve velkém množství, jelikož by došlo k velkému poklesu výkonu ve scéně [9].

## SSAO – Screen Space Ambient Occlusion

Je technologie, která byla poprvé použita v herním enginu firmou Crytek a postupně se rozšířila i do dalších enginů (Unreal engine, source, a dalších). SSAO

<sup>8</sup> Ray tracing (sledování paprsků) je metoda renderování, která místo odražení paprsků světla od zdroje odráží paprsky světla od kamery (uživatele), a to z toho důvodu, že ze zdroje světla vychází nekonečné množství paprsků a je tedy velmi náročné je vypočítat v rozumném čase [20].

poskytuje simulaci Ambient occlusion<sup>9</sup> v reálném čase a v CryEngine 3 je SSAO využito hlavně k dokreslení fotorealistických stínů u hran a rohů objektů [9].

## **SSDO – Screen Space Directional Occlusion**

SSDO rozšiřuje technologii SSAO, neboť dovoluje vypočítat Ambient occlusion z více zdrojů světla a tím dosáhnout daleko lepších a přesnějších výsledků. Další výhodou je možnost ovlivňovat Ambient occlusion zdroji světla s jinou barvou [9].

## **Bump mapping**

Tato metoda je jednou z nejjednodušších a nejrozšířenějších metod, jak dodat rovnému povrchu detaily, které z něj vystupují, a to bez výrazné ztráty výkonu. Je to dáno úpravou (změnou) normál v každém pixelu na daném objektu. Vzhledem k tomu, že se normály využívají při výpočtu osvětlení, bude výsledný efekt vypadat jako hrbolatý povrch. Výsledek bump mappingu se ukládá do speciální textury, jež se nazývá bump mapa. Ta se využívá pro vytvoření detailů, které z povrchu vystupují pouze nepatrně [9].

## **Normal mapping**

Normal mapping funguje na podobném principu jako bump mapping, ale oproti bump mappingu neudává změnu normál, ale přináší přímo absolutní hodnotu normál pro každý pixel. Výsledek se ukládá do speciální textury, tzv. normal mapy, a vzniká operací rozdílu mezi jednoduchým (low poly) a složitým (high poly) modelem. Normal mapa je následně aplikována na jednoduchý model, čímž dojde k dodání detailů a přiblížení k podobnosti složitěho modelu s minimálním dopadem na výkon ve scéně. V této bakalářské práci se nejvíce používá tato metoda, a to nejen z důvodu výborné kvality, která má přijatelné nároky na výkon, ale také z důvodů přiměřených nároků na časové zpracování [9].

## **Parallax occlusion mapping**

Parallax occlusion mapping je jednou z pokročilejších technik pro dodání detailů do scény, neboť vytváří iluzi nerovného povrchu vytvořenou na základě mapy s daty o výšce v každém pixelu. Tato technika funguje i při pohledu z různých úhlů, díky čemuž nedochází ke ztrátě detailů. Jediným místem, kde nastává ztráta detailů, jsou

---

<sup>9</sup> Ambient occlusion (zastínění okolí) je metoda stínování, která pomáhá modelům vypočítat lokální osvětlení a dodat realističtější dojem, a to díky ztlumení světla v místech kam přímo světlo nedopadá a na základě okolí ve scéně (například rohy místností) [14].

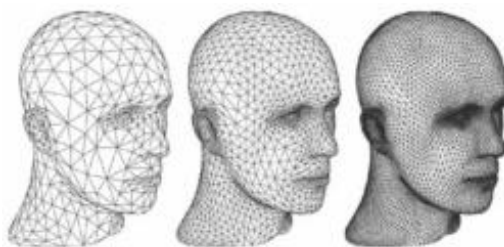


okraje objektu, které jsou stále rovné. Je to dáno tím, že geometrický model objektu zůstává stále rovný.

Další nevýhodou parallax occlusion mapping může být absence stínů, neboť některé techniky stínování nejsou kompatibilní s touto technologií, jednou z nich je například i ambient occlusion. CryEngine 3 však díky vlastní implementaci a rozšíření technologie parallax occlusion mapping tento problém kompenzuje a stíny nahrazuje využitím technologie SSDO. Podporou ze strany CryEngine 3 dochází taktéž k velkému snížení nároků na výkon nebo k možnosti využívání této technologie ve větší míře. Parallax occlusion mapping bohužel nepodporuje žádnou formu anti-aliasingu, neboť není tvořen geometrickou interpretací modelu, který právě anti-aliasing nutně potřebuje. To má za následek, že v některých případech dochází k projevení nepřesností a nerovností na povrchu objektu a ztrátě fotorealističnosti. Tento problém lze řešit pomocí výkonově náročnější varianty, kterou je teselace s použitím techniky displacement mapping [9].

## **Tessellation**

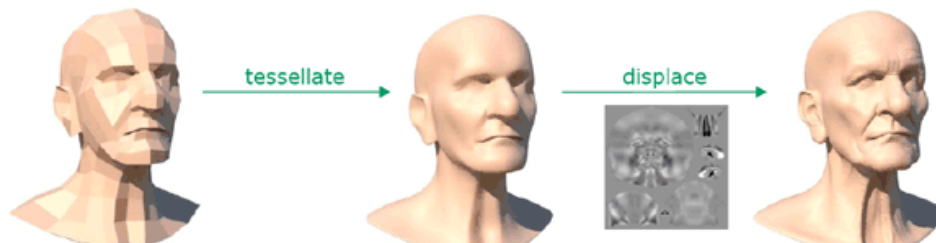
Tessellation je velmi důležitá technologie, která zajišťuje fotorealističtější dojem objektu ve scéně aplikací zaoblení nebo zjemnění geometrického modelu. Jak je vidět na obrázku č. 2, tento proces tedy do objektu nepřidává nové detaily, ale pouze zjemňuje trojúhelníkovou síť daného modelu a tím dochází k jeho vyhlazení a přirozenějšímu vzhledu.



Obr. 2 Postup teselace [21]

Pro přidání dodatečných detailů do objektu ve scéně se používá jeden z druhů teselace, a to displacement mapping, který je možné vidět na obrázku č. 3. Displacement mapping lze označit za plnohodnotnou techniku, kterou lze dodat modelu velké množství detailů při zachování všech vlastností, jako kdyby se jednalo o detaily vytvořené modelováním. Displacement mapping je tvořen heightmapou, jež je uložena v alfa channel rastrové textury. Ta nám udává místa, kde bude možné uplatnit přidání

detailů nebo manipulování s geometrickou interpretací části objektu, kde je textura aplikována. Tato technologie je přímo podporována v CryEngine 3, díky podpoře v DirectX 11 lze využít hardwarové akcelerace a tím umožnit rozšíření této technologie ve scéně za cenu minimálního výkonového zatížení [12] [21].



Obr. 3 Postup při displacement mapping [12]

## Blend Layer

Blend layer je jedna z metod jak zvýšit počet detailů v použitých texturách. Funguje na principu mixování (prolínání) textur na základě černobílé mapy, která je uložena v alfa kanálu textury, kde mají být dodány detaily. Hlavní použití nastává v momentě, kdy dochází u textur k vizuálnímu periodickému opakování a tím snížení fotorealistického vjemu [9].

## Eye adaptation render

Používá se k simulaci lidského oka při přechodu ze světlejší do tmavší scény a naopak. Při použití dochází k rychlé změně jasu a kontrastu a tím k docílení efektu „oslnění“ od okolního osvětlení [9].

## High Dynamic Range Lighting, Color Grading

High Dynamic Range Lighting (dále jen HDR) je technologie, která ve scéně s extrémním jasnem a kontrastem zajišťuje fotorealističtější vykreslování. Jedná se například o scény, kde se vyskytuje velké množství ohně, či o scény v uzavřeném tmném prostoru, který je osvětlován silným zdrojem světla.

Color grading je metoda korekce barev, kterou je možné použít pro závěrečné úpravy scén. Jako je například letní den se sytými barvami nebo zimní den s vybledlými barvami. Zahrnuje funkce jako jsou Hue & Saturation, Contrast & Brightness, Luminance & Color Curves a další [9].

### 3.3.2 Fyzikální stránka enginu

Fyzikální systém je jednou z nedílných částí počítačových vizualizací. Může se jednat o architektonické vizualizace, kde není fyzikální systém nutně potřeba, dále o herní aplikace, které používají fyzikální systém, který se snaží co nejvíce přiblížit fungování reálného světa, a to za použití co nejnižších nároků na výkon nebo se může pojednávat o reálných simulacích, jež mají za úkol simulovat přesné chování reálného světa pro vojenské, vědecké nebo jiné účely.

Fyzikální systém v CryEnginu 3 má široké použití a je možné ho aplikovat téměř na každý objekt ve scéně. Avšak v této bakalářské práci, která je zaměřena striktně na vizuální a funkční stránku, je využití fyzikálního systému velmi omezeno, a to z důvodu snahy o co největší snížení nároků na výpočetní výkon. Fyzikální systém je uplatněn pouze na uživatele (postava ve scéně), ovládání a funkcionalitu vozidel, která jsou umístěna ve scéně (fyzikální systém se uplatňuje pouze při použití daného vozidla vůči okolí a danému vozidlu), a simulaci prověšení lan a drátů, které jsou umístěny ve scéně.

Dalším možným uplatněním je realističtější zobrazení vegetace, a to zejména u stromů, kdy dochází díky simulaci síly a směru větru k ohýbání kmenů. Toto je bohužel v této práci nevyužito, neboť v používaném vydání CryEngine 3 ve verzi 3.4.4 je chyba v hlavičkovém souboru, kde je tato funkce vypnuta. Jelikož je použita neplacená verze, kde nejsou přiloženy editovatelné hlavičkové soubory, ale pouze zkompilovaná DLL knihovna, tak nebylo možné tento problém vyřešit [7].

### 3.3.3 Sandbox editor

Sandbox je editor, jenž slouží ke kontrole a komunikaci s enginem, který obsahuje CryEngine. Jedná se o kompaktní vývojové prostředí, jež je možné použít k tvorbě v reálném čase na všech platformách. Díky technologii Live Create, kterou CryEngine implementuje v editoru, se nabízí možnost přímého testování navrhované aplikace jak na PC, tak i na Xbox 360 a PlayStation 3. Tato technologie se automaticky postará o konverzi, optimalizaci objektů a dat a jejich přenos mezi platformami, což vede k úspoře času a také možnosti vidět změny osvětlení, materiálu a objektů v reálném čase na všech připojených platformách [8].

## **Material Editor**

Jednou z nejdůležitějších částí Sandbox Editoru je Material Editor, který je použit pro interakci s objekty a pro tvorbu a správu materiálů v celé scéně. Material Editor nám dovoluje také nastavovat pokročilé vlastnosti jako jsou teselace, paralax occlusion mapping a enviroment mapping, který slouží k simulaci odrazů světla od daného materiálu [8].

## **Flow Graph**

Jedná se o jednoduchý vizuální editor pro nastavování funkcionality jednotlivých bloků v CryEnginu. Slouží k interakci a nastavování parametrů prvkům jako jsou Events, triggers a game logic (jedná se o body ve scéně, které uživateli podávají informace a pracují s informacemi od uživatele). V této bakalářské práci je Flow Graph použit například na simulaci automaticky otevíraných dveří v budově IC.

## **Procedural Placement Tools**

Obsahuje řadu nástrojů, které slouží k rychlému umístování a generaci objektů ve scéně, například generování realistické vegetace na uživatelem vybranou texturu, jež je aplikována na terén ve scéně. V této bakalářské práci byl nejčastěji používán nástroj snapping tool, který po aktivaci kláves Ctrl, Shift a kliknutí na pozici ve scéně přenese uživatelem označený objekt na zvolenou pozici [8].

## **Time of Day System**

Technologie Time of Day System slouží k nastavování pokročilých vlastností osvětlení scény v enginu. Hlavní předností této technologie je možnost plynule měnit nastavení osvětlení v závislosti na denní době (jiná intenzita osvětlení v deset hodin oproti dvanácti hodinám, dále přechod mezi dnem a nocí), navíc v rámci toho také zajišťuje příslušný pohyb slunce a měsíce na obloze. Díky těmto charakteristikám umožňuje dodat větší fotorealističtější vjem ve scéně.

# **3.4 Pluginy pro export do CryEngine 3**

## **CryTif**

CryTif je nástroj, jenž je po nainstalování integrován v programu Adobe Photoshop. Slouží k optimalizaci a exportu textur do formátu TIF. Takto vyexportovaná textura má označení Source asset [24]. Po exportu textury do formátu TIF dojde ke kontaktování **Resource Compileru**, který se postará o optimalizaci textury

pro CryEngine a její efektivní načtení [10]. K tomuto účelu slouží formát DDS, do kterého se uloží metadata, jež uchovávají informace o vlastnostech textury. Může se jednat o informace, které říkají, zda se jedná o NormalMapu nebo zda textura obsahuje Alpha kanál. Textura ve formátu DDS, tzv. Target Asset, slouží k přímému použití v CryEnginu [24].

## **CryExport**

Vzhledem k tomu, že CryEngine nepodporuje práci s klasickými 3D formáty, musí dojít k jejich optimalizaci a převodu do formátu, který CryEngine dokáže zpracovat a využít. Jedná se o formát CGF pro model a formát MTL pro materiál. K tomu lze využít oficiální nástroj CryExporter, který slouží k exportu modelů, animací a materiálů z 3D modelovacích prostředí 3ds max a Maya. Pokud bychom se rozhodli pro použití jiného modelovacího nástroje, například Blenderu nebo Google SketchUp, lze využít neoficiální nástroje. Při absenci oficiálního i neoficiálního nástroje pro export do CryEnginu lze využít metodu, kdy model vyexportujeme ve formátu collada DAE, následně můžeme tento objekt nahrát do programu Google SketchUp a pomocí neoficiálního nástroje PlayUpTools vyexportovat do CryEnginu.

## **PlayUpTools**

PlayUpTools je plugin pro program Google SकेctUp, který slouží nejen k tvorbě materiálů z textur objektu, ale také k exportu objektů do CryEnginu. Plugin je díky podpoře ze strany vývojářů neustále vylepšován. Za jednu z jeho výhod lze označit volnější pravidla pro export objektů do CryEnginu. Například objekty mohou obsahovat více než 65535 trojúhelníků, což nám dovoluje najednou exportovat kvalitnější a propracovanější objekty, dále odstraňuje negativní natočení normál a tím i problémy, kdy dochází k zobrazování černých fragmentů na objektu v prostředí CryEngine. Plugin kromě CryEnginu 3 podporuje také jeho starší verzi CryEngine 2, Source engine a Unity3D.

## **3.5 Základní stavební prvky v CryEngine 3**

### **3.5.1 Solid**

Solid je zcela nejjednodušší a nejzákladnější prvek v celém editoru. Slouží k tvorbě kvádrů, krychlí, válců a dalších jednoduchých těles. Tato tělesa mají vlastní kolizní model a lze na ně aplikovat materiály, ale sama o sobě nemají žádnou funkci.

Jejich hlavní použití je k dotvoření nějakých větších celků ve scéně, jako jsou například jednoduché schody. Často jsou také využívány jako neviditelná zábrana k určení hranic scény, za které se uživatel nedostane. Toho lze docílit aplikováním proxy materiálu na solid nebo aplikováním materiálu, který má opacity (průsvitnost) nastavenou na 0.

### 3.5.2 Brush

Brush je základní a nejvíce používaný prvek v celém editoru. Brush tvoří polygon mesh (viz. Seznam symbolů, zkratk a termínů, dále jen mesh) a jeho kolizní model. Brush je základním prvkem při budování vizualizace, ale stejně jako solid nemá sám o sobě žádnou funkci, pouze slouží k umístění v prostoru. V tomto textu se bude psát o brush také jako o objektu. Je důležité si ho neplést s mesh, neboť mesh nemá kolizní model, brush tímto modelem disponuje.

### 3.5.3 Entita

Entita je prvek, který dovoluje interakci s uživatelem a okolím. Ovládání a vlastnosti jsou podobné štětcům. Jako příklad entity lze uvést ovládání a animaci dveří, výtahy nebo oheň. Entita sama o sobě není reprezentovaná, pouze se k objektu nebo materiálu přiřazuje.

Jednou z výhod entit je možnost linkování (spojování) s okolními prvky, například s osvětlením. V momentě, kdy se uživatel přiblíží k entitě typu Proximity Trigger, která je propojená s osvětlením, dojde k aktivaci osvětlení. Díky použití linkování dochází k možnosti absence ručního psaní kódu a tím i ke značnému zjednodušení postupu.

### 3.5.4 Group

Group slouží k uspořádávání a uskupování objektů ve scéně. Je to jeden ze způsobů, jak kontrolovat a manipulovat s velkým množstvím podobných objektů nebo jak objekty organizovat.

### 3.5.5 Prefab

Prefab jsou stejně jako group určeny ke kontrole, manipulaci a organizování objektů ve scéně, ale s tím rozdílem, že prefab jsou instance objektů a entit, které lze načítat a ukládat do xml souborů. Výhodou ukládání do xml souboru může být snadná

záloha a přenositelnost prefab do jiných scén. Nejčastěji jsou prefab použity na nějaké funkční objekty, například na model automobilu. Základní části automobilu (karoserie, kola, okna, atd.) jsou realizovány pomocí statických mesh modelů s příslušným kolizním modelem. K těmto částem automobilu jsou připojeny další prvky, jako jsou entity, které simulují jejich chování nebo zvuky. Jako příklad můžeme uvést prefab okno. Prefab se skládá z rámu okna, což je jednoduchý brush s příslušným materiálem, a ze dvou okenic, což jsou entity s nahraným brush objektem. Abychom pokaždé nemuseli vytvářet rám okna a okenice, máme možnost si je uložit jako prefab pro vícenásobné použití.

Výhoda Prefab spočívá především v tom, že se jedná o instanci, což při změně jednoho prefab objektu umožní to, že dojde ke změně všech objektů ve scéně.

### 3.5.6 Archetype

Archetype jsou založeny na entitě, mají pokročilejší možnosti nastavování parametrů a jsou schopny interakce s okolním světem. Stejně jako entita se vážou k objektům. Rozdíl oproti entitě spočívá v tom, že archetype definuje fyzikální vlastnosti a chování objektů vůči uživateli a okolí, zatímco entita definuje pouze chování vůči uživateli. Příkladem může být reakce vody na částečně ponořený objekt jako je loď. Archetype nedovolí vodě, aby objektem prostoupila, neboť nadefinuje její obtékání okolo objektu. Jelikož se archetype chovají jako fyzické modely, lze uvést příklad kolize automobilu s překážkou, kdy dojde k deformaci jednotlivých částí automobilu.

Výhodou archetype je možnost absence skriptování a tudíž jednoduchá implementace ve scéně. Za nevýhodu lze označit značně zvýšenou náročnost na výpočetní výkon. Proto se nedoporučuje používat mnoho těchto objektů ve scéně a raději je nahrazovat naskriptovanými entitami uskupenými v prefab.

### 3.5.7 Road

Road je nástroj, který slouží na tvorbu silnic. V této práci je modifikován a použit také na tvorbu chodníků a obrubníků, a to z důvodu dosažení fotorealističtějšího dojmu. Road je v podstatě křivka, jež se táhne z bodu A do bodu B, a na které je aplikována textura, která se neustále opakuje. Použitím road namísto

modelování silnic, chodníků a obrubníků dojde ke značné úspoře času a výpočetního výkonu na zobrazování scény.

### 3.5.8 Decal

Decal jsou v podstatě jednoduché plošky o stranách stejné délky s velikostí začínající od 256 px až po 8192 px. Na tyto plošky je aplikován materiál s alfa mapou, která zajistí, že bude zobrazena jen určitá část plošky (například rozbitý beton nebo špína na zdi) a následně je materiál projektován na okolní objekty nebo terén. Decal se ve větší míře používá k dosažení velkého množství detailů na terénu nebo na objektech ve scéně. Výhodou decal je jejich velmi nízká výpočetní náročnost, která nám dává možnost doplnit scénu o velké množství detailů, díky čemuž můžeme dosahovat velmi dobrého fotorealistického vjemu.



## 4 Vizualizace areálu Technické univerzity

### 4.1 Tvorba základních prvků ve scéně

#### 4.1.1 Analýza a tvorba terénu

Před samotnou tvorbou areálu TUL muselo dojít nejdříve k vytvoření terénu ve scéně, až poté k následnému osazování budov a okolí areálu. Z tohoto důvodu musela být provedena analýza technik, které by vyhovovaly našim požadavkům a potřebám v takto členité a rozlehlé scéně. Po konzultaci možností na oficiálním fóru CryEnginu a prostudování potřebné literatury došlo ke stanovení tří základních technik na tvorbu terénu.

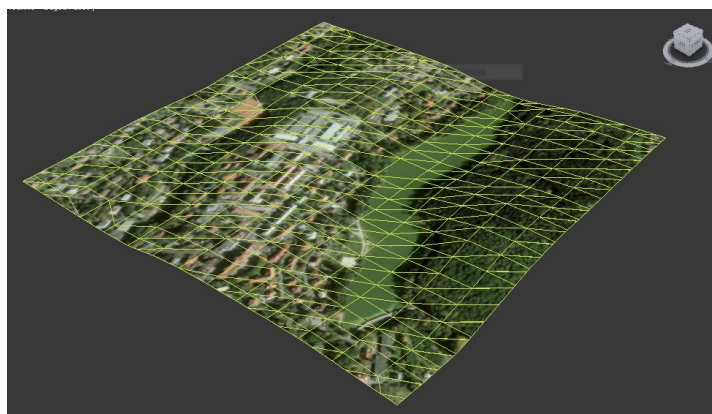
První možnou technikou bylo ruční vytvoření terénu s využitím nástroje CryEngine terrain tool. Tato varianta by se uskutečnila na základě fotografie zachycené pomocí Google maps a fotografií dostupných na internetu. První technika by byla pro tuto bakalářskou práci značně zdoluhavá, složitá a velmi nepřesná, z tohoto důvodu byla vyloučeno její použití. Obecně lze však konstatovat, že pro tvorbu obecného terénu, který si nezakládá na reálných datech, ji lze považovat za dostačující.

Druhá technika tvorby terénu scény je realizovaná také na základě ručního vytvoření dle fotografie zachycené pomocí Google maps, ale již z reálných výškových údajů nadmořské výšky, které lze získat například na internetové adrese daftlogic [11] nebo přímo z aplikace Google Earth. Výšková data jsou pro Českou Republiku měřena v mřížce bodů o rozestupu devíti metrů a zbylý prostor mezi body mřížky je dopočítán. Ačkoliv se jedná o bezplatnou variantu, jež není také zcela přesná, dosahuje již mnohem lepších výsledků než předchozí technika. Pro přesnější vytvoření terénu by bylo nutné do scény ručně zanést velké množství bodů, následně terén upravit a vyhladit do konečné podoby. V takovém případě by se jednalo o největší přiblížení k reálným datům, ale její nevýhodou je značně rozsáhlá časová náročnost, která by mohla zapříčinit nesplnění termínu pro odevzdání bakalářské práce.

Třetí technika byla odvozena od druhé možnosti tvorby terénu, ale byla eliminována časová náročnost za použití metody heightmapy, byly zde uloženy veškeré výškové údaje reprezentované jako odstíny šedé v rozmezí 0 až 255 a následně použity pro vytvoření terénu pomocí nástroje terrain tool. Tato technika se velmi blíží reálným

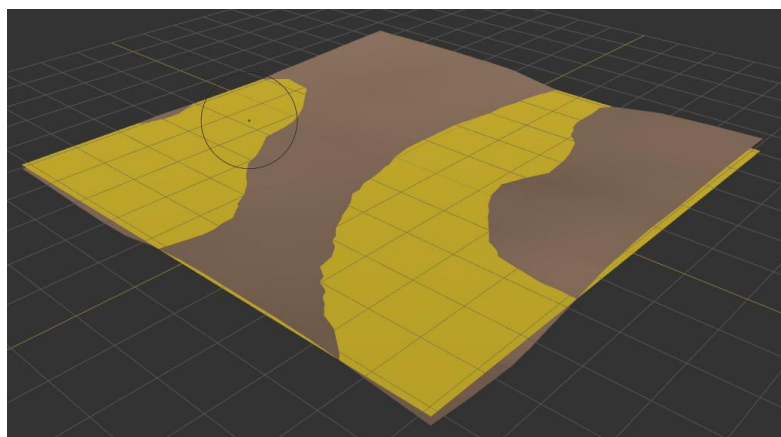
datům popisující terén, navíc její časová náročnost je velmi nízká, z tohoto důvodu byla použita při tvorbě této bakalářské práce.

V této části je také nezbytné vysvětlit, jakým způsobem byla heightmapa terénu získána. Prvním krokem k vytvoření heightmapy bylo získání a seskupení výškových údajů. K tomuto účelu nám posloužil program Google SketchUp, který dokáže z výškových údajů získaných programem Google Earth vytvořit trojrozměrný model a aplikovat na něj texturu zvoleného terénu. Velikost terénu byla volena podle dvou kritérií. Prvním parametrem byla velikost scény, se kterou je CryEngine schopen pracovat. Pohybuje se od 128 pixelů x 128 pixelů až po 8192 pixelů x 8192 pixelů a kde každému pixelu lze nastavit velikost od jednoho metru až po dvaatřicet metrů. Druhým kritériem byla kvalita pokrytí areálu vzorky o nadmořské výšce. Vzhledem k tomu, že je areál TUL velmi členitý bylo nezbytné pokrýt větší oblast, aby nedocházelo ke ztrátám výškové informace a zkreslení terénu. Zvolená oblast o velikosti 1400 metrů x 1400 metrů byla pomocí programu Google Earth převedena a exportována do OBJ modelu. Jelikož CryEngine podporuje pouze určité velikosti scény, byla jako nejbližší velikost zvolena scéna 1024 pixelů x 1024 pixelu s velikostí pixelu odpovídající jednomu metru (plocha o velikosti 106496 metrů čtverečních). Z tohoto důvodu muselo dojít k oříznutí OBJ modelu na velikost 1024 metrů x 1024 metrů. To bylo realizováno importem do programu 3ds max, použitím funkce boolean, která uskutečňuje logické operace nad objekty a následným exportem zpět do formátu OBJ. Jak je vidět na obrázku č. 4, tímto postupem došlo k vytvoření 3D objektu terénu areálu TUL s rozměry 1024 metrů x 1024 metrů, a to s příslušnou texturou, která na něm byla aplikována. Vzhledem k tomu, že program CryEngine však nepodporuje přímou importaci terénu, neboť by bylo příliš náročné propočítat a realizovat jeho kolizní mapu, bylo nutné z 3D objektu terénu vytvořit námi požadovanou heightmapu.



Obr. 4 3D objekt terénu areálu TUL

Druhým krokem k vytvoření heightmapy bylo využití programu Autodesk Mudbox<sup>10</sup>, do kterého byl nahrán OBJ model terénu a následně v něm byl vytvořen objekt plane. Tento objekt byl zarovnán na stejné souřadnice os x a y jako objekt terénu a zhruba do poloviny jeho velikosti souřadnicové osy z, jak je patrné z obrázku č. 5. Program Autodesk Mudbox byl zvolen kvůli jeho možnostem v oblasti rychlého vytvoření heightmapy pomocí nástroje Extract texture map. Tato funkce se postará o promítnutí výškových údajů z objektu terénu na objekt plátna a jeho následné vyrenderování do formátu TIF.



Obr. 5 Zarovnání v programu Mudbox

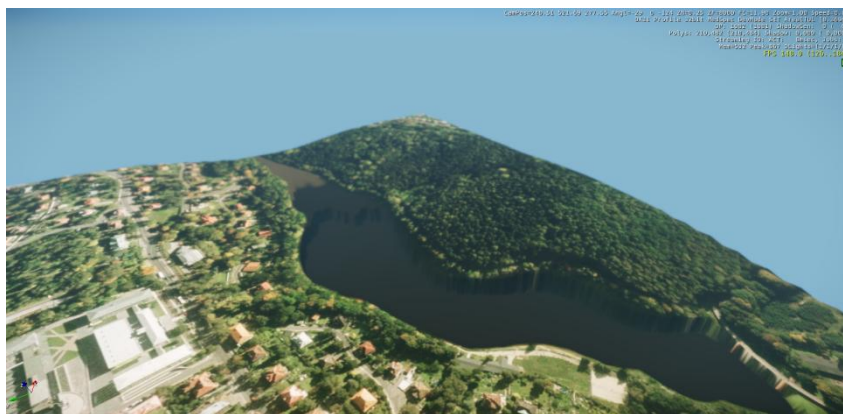
Po provedení exportu z Autodesk Mudbox máme hotovou heightmapu o velikosti 1024 pixelů x 1024 pixelů, kde výškové informace jsou reprezentovány šedými odstíny šedé v rozmezí 0 až 255. V tomto případě však nastává omezení, neboť CryEngine podporuje výšku ve scéně od 0 až po 1023 metrů, a námi vygenerovaná heightmapa by byla v důsledku toho znehodnocena a natažena na výšku. Pro udržení kvality bylo nezbytné provést normalizaci heightmapy na výšku 1024 metrů, k tomuto účelu bylo využito nástroje WorldMachine, který slouží ke generování a vytváření terénů. Normalizace byla provedena importováním heightmapy a znovuvytvořením modelu terénu, nastavením výšky terénu z 255 metrů na 1023 metrů a následným exportem zpět do formátu TIF.

Posledním krokem při tvorbě terénu, který je založen na reálných výškových datech, bylo importování heightmapy vytvořené v programu WorldMachine do CryEngine. Daná akce byla realizována nástrojem Terrain tool, který z heightmapy

---

<sup>10</sup> Autodesk Mudbox je 3D nástroj uzpůsobený k digitálnímu kreslení a modelování technikou, která se nazývá Sculpting (v překladu vyřezávání). Jedná se o metodu, kdy se objekt tesá nebo vyřezává, a tím dochází k tvorbě složitých objektů s velkým množstvím detailů. Autodesk Mudbox také obsahuje velké množství funkcí, jež slouží pro kreslení a práci s texturami na velmi pokročilé úrovni [1].

vyextrahuje výškové informace a vytvoří tzv. terrain mesh. Na tento terrain mesh bylo nutné dodatečně aplikovat třikrát funkci smooth, a to nejen kvůli vyhlazení nepřesností vzniklých normalizací heightmapy, ale také v důsledku dotvoření detailů, které byly vypuštěny kvůli nízkému počtu výškových vzorků. Na závěr byla na terrain mesh aplikována fotografie textury terénu, která sloužila jako reference pro umístování objektů a kreslení textur. Ukázku části importovaného terénu s texturou a modifikacemi pro dodání detailů můžeme vidět na obrázku č. 6.



Obr. 6 Importovaný terén v CryEngine

#### 4.1.2 Editace, optimalizace a tvorba modelů

V této části se zaměříme na práci se samotnými modely, které reprezentují areál TUL. Jedná se o budovy A, B, C, E, F, IC a čtyři přilehlé budovy v ulicích Čížkova a Hálkova (těmto budovám byla věnována speciální pozornost z důvodu větší fotorealističnosti, neboť přímo sousedí s budovami Technické univerzity). Jednotlivé budovy byly převzaty, a to následovně:

- budovy A a B z bakalářského projektu **Model areálu technické univerzity v Liberci** [19],
- budova C z bakalářského projektu **Interaktivní 3D model budovy C** [18],
- budova E z bakalářského projektu **Interaktivní 3D model budovy E** [16],
- budova F z bakalářské práce **Interaktivní 3D model budovy F** [15],
- budova IC z bakalářské práce **Interaktivní 3D-model budovy CIP** [3].

Budovy A, B, C, E a F byly modelovány technikou editable poly modeling v programu 3ds max a u budovy C bylo použita metoda boolean pro vytvoření oken. Budova IC byla modelována v programu Maya, poté byla převedena do programu 3ds max, kde byly odstraněny přebytné prvky, např. kamery, osvětlení, vodící prvky aj.

Jelikož většina modelů nebyla přizpůsobena pro použití v herním enginu, muselo nejprve dojít k jejich úpravě a optimalizaci. U budov A a B, které nemají hotový interiér a slouží pouze jako externí kulisa, došlo k dotvoření detailů a odstranění nepotřebných hran. Ostatní budovy areálu musely být rozřezány na jednotlivá patra, protože jako celek obsahovaly větší počet trianglů než je limit pro export do CryEnginu. Tento postup má také výhodu v pozdější práci v CryEnginu, kdy lze zapnout průhlednost jednotlivých pater a mnohem snadněji do nich implementovat nábytek a ostatní interiérové prvky. U ostatních budov též proběhla optimalizace smazáním nepotřebných hran. U budovy C muselo dojít při použití metody boolean k úpravě otvorů pro dveře a okna pomocí funkce Weld<sup>11</sup>, neboť se zde vyskytovalo velké množství osamocených vertexů. Pro okolní budovy v ulicích Čížkova a Hálkova a pro dodatečné modely byla použita také technika editable poly modeling.

Výhodou CryEnginu je možnost generování kolizní mapy na základě normálových vektorů, díky čemuž dochází k absenci složitého nastavování v programu 3ds max. Generovaná kolizní mapa tak kompletně kopíruje objekt, ke kterému se váže. Na druhé straně může být tato přednost někdy i omezením, a to například u objektů, které jsou příliš složité. Tento problém lze obejít ručním vytvořením kolizní mapy. Toho lze docílit jednodušším modelem, na který se aplikuje materiál s pojmenováním proxy.

### 4.1.3 Tvorba textur a jejich aplikace

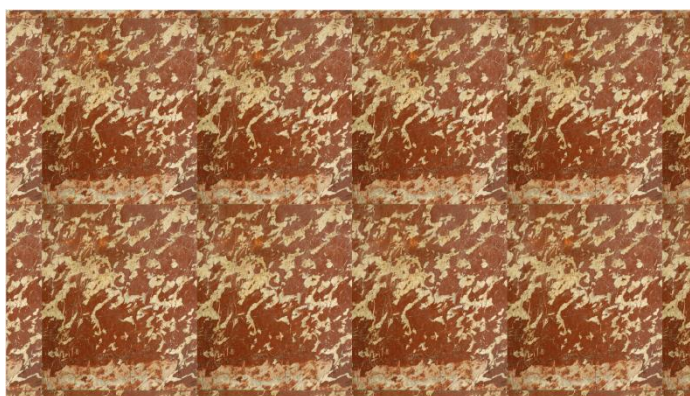
Textury jsou jednou z nejdůležitějších částí určujících vzhled této bakalářské práce, neboť zajišťují požadovaný fotorealistický vzhled scény. Zároveň s množstvím použitých textur stoupá požadavek na grafický výkon ve scéně, proto bylo nutné zvolit kompromis mezi kvalitou a výpočetním výkonem. Bylo použito textur o velikosti 1024 pixelů x 1024 pixelů, které zajistily dostatečnou kvalitu s přijatelnými nároky na výkon.

Textury aplikované v této bakalářské práci byly získávány z různých zdrojů, nejčastěji z internetových katalogů textur, jež jsou volně k dispozici, a textur vytvořených z fotografií areálu. Mezi hojně utvářené textury patřily diffuse textures, které reprezentují vzhled objektů. Na tyto textury jsou aplikovány normal map textury, využívané k dodání detailů na povrch daného objektu. Textury byly zpracovávány

---

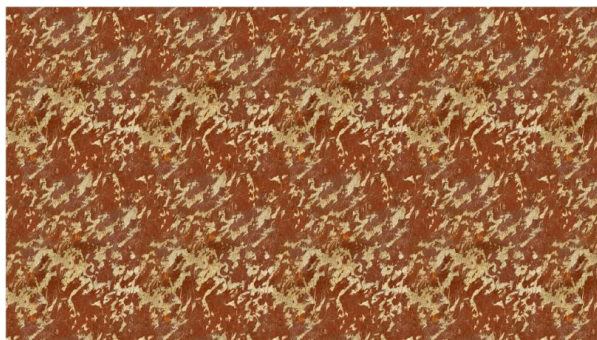
<sup>11</sup> Weld je funkce, která slouží pro sloučení označených vertexů (bodů) v určité vzdálenosti [17].

v programu Adobe Photoshop a pro generování normálových map byl použit program PixPlant. Všechny textury byly uloženy ve formátu TIF, který je podporován CryEngine, a valná většina z nich byla optimalizována pomocí CryTif exportéru. Nejčastějším problémem při tvorbě textur byla jejich horizontální a vertikální nenávaznost, která se označuje termínem Tiling, viz obrázek č. 7. Tento problém působí v konečné scéně velmi rušivě a není shodný s požadavkem na fotorealističnost. Problém tilingu lze snadno řešit pomocí programu PixPlant, který zahrnuje funkci pro generování textur, která tiling odstraní.



Obr. 7 Textura bez ošetření tilingu

Texturu upravenou pomocí tilingu lze vidět na obrázku č. 8. Některé textury jsou však velmi komplexní, v důsledku čehož generování a následné odstranění tilingu nelze provést, neboť by docházelo k velkému množství rozmazání a znehodnocení textury. Takové textury bylo potřeba manuálně upravit v programu Photoshop použitím funkce Offset, která posunula horizontální a vertikální okraje textury směrem ke středu, a tím vznikl uprostřed textury přechod, jenž vytvořil spoj mezi začátkem a koncem hran. Tento spoj bylo potřeba zamaskovat použitím nástroje Clone stamp tool. Uvedený postup byl použit převážně na textury, které se geometricky opakují, například kamenné obložení, dlaždice, kamínky aj.



Obr. 8 Textura s opraveným tilingem

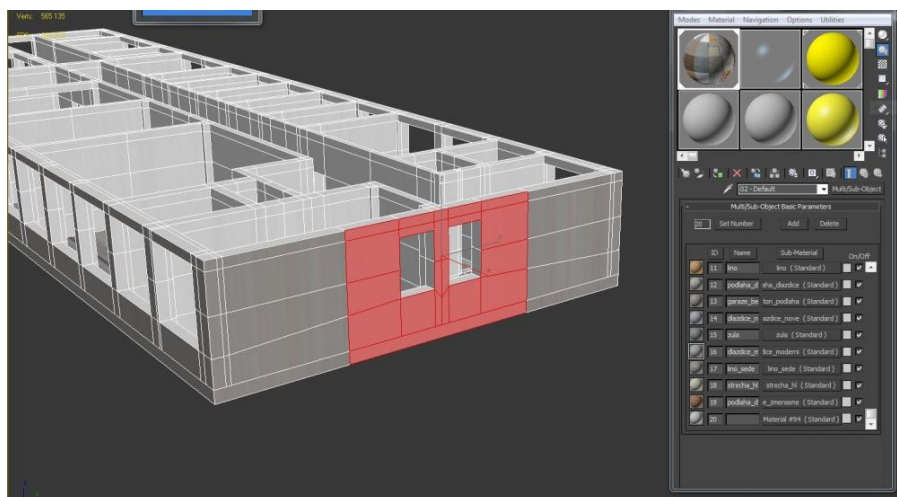


Některé textury byly také vybaveny alpha mapou pro použití technologie Parallax occlusion mapping. Tato technologie byla využita například na dlaždicovém obložení budovy C, kde slouží pro rychlé a jednoduché dotvoření detailů v podobě změny hloubky spáry mezi dlaždicemi. Toto je patrné z obrázku č. 9, kde je možné vidět texturu bez aplikace Parallax occlusion mapping (vlevo) a texturu, kde je již aplikován Parallax occlusion mapping (vpravo). Alfa mapa je tvořena černobílou texturou uloženou v alfa kanálu, kde černá barva reprezentuje okolí, se kterým se bude manipulovat, a bílá barva znázorňuje okolí, které zůstane nezměněno.



Obr. 9 Textura bez Parallax occlusion mapping a textura s Parallax occlusion mapping

Samotné texturování bylo provedeno v programu 3ds max, a to využitím funkce Polygon material ID a materiálu typu Multi/Sub-Object. Tento materiál lze označit za skupinu standardních materiálů (standardní materiály obsahují textury a jejich normálové mapy), které jsou přidělovány objektu na základě ID nastaveného funkcí Polygon material ID. Funkce pracuje na principu selekce faces (plošek), kterým se nastaví unikátní číselné ID, a tím dojde k přidělení požadovaného materiálu, viz obrázek č. 10. Tato názorná ukázka znázorňuje v levé části vybrané plošky a v pravé material editor s nastavenými texturami. Po nastavení materiálů muselo dojít k jejich standardizaci a uzpůsobení velikosti textur požadovanému modelu. Toho bylo docíleno funkcí UVW map v modu box s velikostí 4 metry. Pokud textura velikostně neodpovídala, docházelo k její úpravě přímo v CryEngine, kde se v Material Editoru nastavila multiplikace její šířky a výšky.



Obr. 10 Selektce faces

## 4.2 Vytváření scény v CryEngine 3

### 4.2.1 Export modelů, chyby při exportování a jejich obcházení

Prvním krokem při samotné tvorbě areálu TUL je export modelů a jejich materiálů z programů 3ds max a Google SketchUp do CryEnginu. Jak bylo vysvětleno v předchozích kapitolách, je toho docíleno pomocí exportních nástrojů, které model optimalizují a postarají se o samotný export do formátu CGF a MAT. Tyto vyexportované objekty jsou následně umístěny do příslušných adresářů uvedených v kapitole 3.2, přesněji do adresáře Game. V této bakalářské práci jsou objekty nejčastěji umístěny v adresáři Game\Objects\ArealTUL. Zde jsou rozčleněny do adresářů podle příslušné budovy, ke které patří, nebo podle typu objektu (například nábytek, okolní dekorace aj.). S exportem z programu 3ds max přichází řada komplikací a problémů, které je potřeba vyřešit. V této části si uvedeme nejčastěji vyskytující se problémy při exportu.

#### **Problém Degenerate faces**

Problém Degenerate faces lze označit za ten nejvíce vyskytovaný. Jedná se o problém, kdy při importu, tvorbě nebo úpravě modelu dochází k vytvoření plošek, které nemají žádný obsah. Tento problém lze snadno vyřešit, a to použitím funkce Weld nad všemi vertexy.



## **Problém vertexů ležících na stejné linii**

Tento nedostatek často vzniká jako výsledek odstranění problému Degenerate faces. Jedná se o dva a více vertexů, které leží na společné hraně, ale nejsou součástí žádného trojúhelníku. Řešení je opět velmi jednoduché, je realizováno pomocí funkce 3ds max s názvem Noise. Tato funkce je aplikována nad celým objektem a postará se o deformování objektu a tím i narušení linie, ve které vertexy leží. Aby nedošlo k viditelné deformaci objektu, volí se parametry funkce Noise v řádu centimetrů.

## **Problém černých fragmentů (trianglů)**

I když nejde o nejčastěji vyskytující se problém, lze jej označit za jeden z nejdůležitějších, neboť ovlivňuje především vizuální stránku a narušuje tak požadavek na fotorealističnost této bakalářské práce (viz obrázek č. 11). Jedná se o zobrazení černých fragmentů na objektu ve scéně, jež vznikají díky špatnému postupu při modelování objektů. Nejčastěji dojde k rozhození smoothing groups, které definují zpracování textur. Řešení v tomto případě záleží především na konkrétním objektu, ke kterému se problém vztahuje. Po prostudování několika možností a materiálů se osvědčila metoda vyresetování smoothing groups pomocí volby Clear All a nastavení smoothing groups na Auto Smooth v úhlu 45 stupňů.



Obr. 11 Problém černých fragmentů

## **Problém s normálovými vektory**

Při modelování byly často používány funkce symmetry (vytvoření symetrického modelu podle osy) a funkce Mirror (zrcadlení podle osy), bohužel při použití těchto funkcí nad objekty a jejich následným exportováním docházelo ke špatné interpretaci normálových vektorů v prostředí CryEngine. Díky špatným informacím o normálových vektorech, které jsou potřebné například pro výpočet stínů a osvětlení na objektu, docházelo ke ztrátě vizuální kvality daného objektu, jak si lze všimnout na obrázku

č. 12 (vlevo). Tento problém je relativně složité řešit v prostředí 3ds max, proto byla zvolena jednodušší technika, která exportuje objekt i s texturami do formátu DAE a následně ho importuje do prostředí Google SketchUp. Export do formátu DAE byl vyřešen pomocí pluginu OpenCollada pro 3ds max, neboť integrovaný plugin v 3ds max pro export do formátu DAE není kompatibilní s Google SketchUp. V Google SketchUp bylo použito exportního nástroje PlayUpTools, který opravuje špatnou interpretaci normálových vektorů. Výsledný objekt exportovaný tímto nástrojem lze vidět na obrázku č. 12 (vpravo).



Obr. 12 Zobrazení s problémem normálových vektorů a zobrazení s jeho ošetřením

#### 4.2.2 Import do Cryengine 3 a umístění ve scéně

Po exportu a následném umístění CGF objektů, MAT materiálů a textur do příslušných adresářů bylo vše připraveno k samotné tvorbě areálu v prostředí CryEngine. Import probíhal použitím vestavěného prohlížeče v Sandbox Editoru, který zahrnoval všechny prvky zahrnuté v adresářové struktuře CryEnginu. Objekty CGF zde byly definované jako brush s již aplikovaným materiálem. Díky chybě v CryEngine musely být vždy při prvním použití objektu brush přiřazeny jeho materiálu příslušné textury.

Prvním krokem při tvorbě scény bylo nanesení textur, a to užitím nástroje paint tool na vytvořený terén. Areál obsahuje celkem osm textur nanesených na terén, u kterých je pouze měněno jejich zbarvení a bílé vyvážení. Díky tomu lze dosáhnout velkého množství detailů, navíc vzhledem k malému počtu textur bylo dosahováno též poměrně nízkých nároků na výpočetní výkon. Textury byly nanášeny přes základní texturu terénu, kterou tvořila fotografie z Google Earth. Ta sloužila jako vodící prvek pro umístování a upravování scény. Po dokončení nanášení textur došlo k tvorbě vegetace v okolních částech areálu, a to za pomoci nástroje vegetation, dále k vytvoření

silnic a cest pomocí nástroje road a importu budov exportovaných z Google Warehouse. Vegetace, cesty a jednoduché budovy zajistí dostatečné naplnění okolí areálu TUL důležitými detaily pro fotorealistický vjem této bakalářské práce.

Dalším krokem byla tvorba jednoduchých budov, které neobsahují interiér. Přesněji se jednalo o budovy A, B a budovy v ulicích Čížkova a Hálkova. Po přesném umístění budov podle vodící textury na terénu došlo k přizpůsobení terénu reálným datům (například vytvoření kopečku u budovy B a svažité silnici naproti budově A). Následovalo umístění oken, ale vzhledem k tomu, že budovy neoplývají interiérem, musela být oknům nastavena neprůhledná textura. V rámci zachování fotorealističnosti jim byla nastavena reflexe, která odrážela okolní objekty. Tato reflexe byla z hlediska ušetření výkonu volena jako předem vypočítaná a bylo jí dosaženo metodou CubeMap. Jedná se o metodu, kdy se vytvoří krychle a na základě jejích stěn se vyrenderuje scéna ve formátu TIF, která má být odrážena. Tento render se poté přiřadí materiálu, jež reprezentuje skleněnou plochu okna. Stejná metoda byla použita i pro okna, která jsou u modelů, které jsou vybaveny interiérem, ale s tím rozdílem, že se materiál reprezentující skleněnou plochu okna nastavil na průhlednost v rozsahu od dvaceti do šedesáti procent.

Stejný postup jako u budov bez interiéru byl použit na budovy s interiérem s tím rozdílem, že se budovy stavěly od spodního patra po horní. V každém patře přímo došlo k umístění dveří, oken a interiérového zařízení. Při umisťování interiérového zařízení bylo především dbáno na to, aby se zařízení v místnostech příliš neopakovalo a bylo ke každé budově umístěno přibližně jiné interiérové zařízení. Bohužel nebyl k dispozici žádný automatický generační nástroj a vše se muselo umístit ručně na dané pozice.

Na závěr došlo k doplnění areálu TUL o detaily v podobě kvalitní vegetace a o doplňkové objekty jako jsou ohrubníky, chodníky vytvořené pomocí road tool, detaily vytvořené pomocí decals (například praskliny v betonu, šmouhy na silnici, ale také vodící čáry a značky) a vozidla.

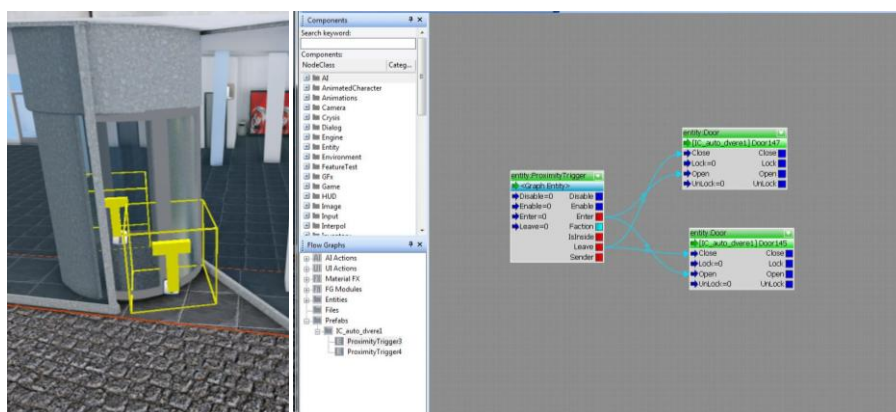
## 4.3 Interaktivní prvky

Jelikož reálný svět není statický, ale jedná se o dynamické prostředí plné interakce, byla také tato bakalářská práce doplněna o interaktivní prvky. Výhodou CryEnginu je úplná absence potřeby animací, vše se vytváří přímo v editoru jako například posuv po ose nebo jako fyzikální model daného objektu.

### 4.3.1 Simulace oken a dveří

Nejjednodušším interaktivním prvkem v celém areálu jsou okna a dveře. Jedná se o jednoduchý Brush s entitami uzavřený do prefab a mnohonásobně umístěný po celé scéně. Například při tvorbě dveří, kde rám reprezentuje brush, bez funkcí pouze s kolizí a samotné dveře reprezentuje entita, která má nastaveny parametry jako jsou úhel otevření dveří, osa (směr) otevření a rychlost otevření. Stejné parametry plátí také pro okna.

Složitějším případem dveří jsou automatizované dveře v budově IC, které se neposouvají do stran, ale zasouvají se v kruhu a jsou aktivovány přiblížením. Zasouvání je realizováno posunutím pivotu objektu dveří přibližně na střed kruhu, po kterém se mají pohybovat. Detekce přiblížení je uskutečněna využitím entity Proximity Trigger, která se stará o zavírání a otevírání dveří. Této entitě je přiřazena funkce využitím nástroje Flow Graph, díky čemuž je i umožněna absence složitého kódování (viz obrázek č. 13).

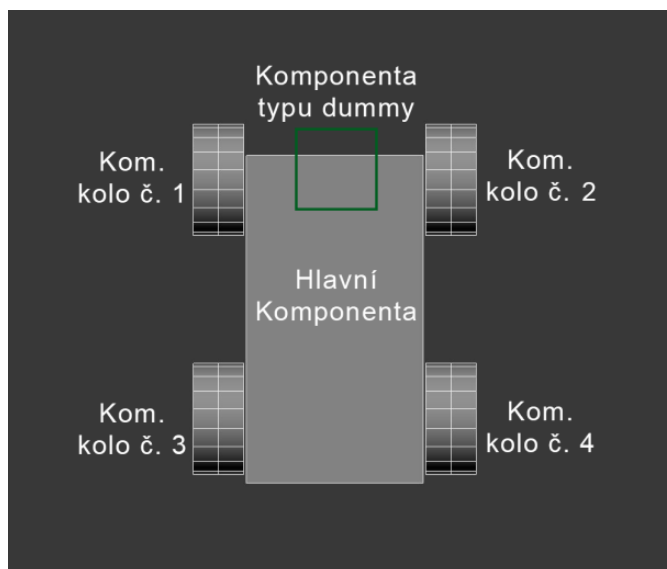


Obr. 13 Proximity Trigger a jeho funkcionalita v Flow Graph

### 4.3.2 Základní simulace vozidla

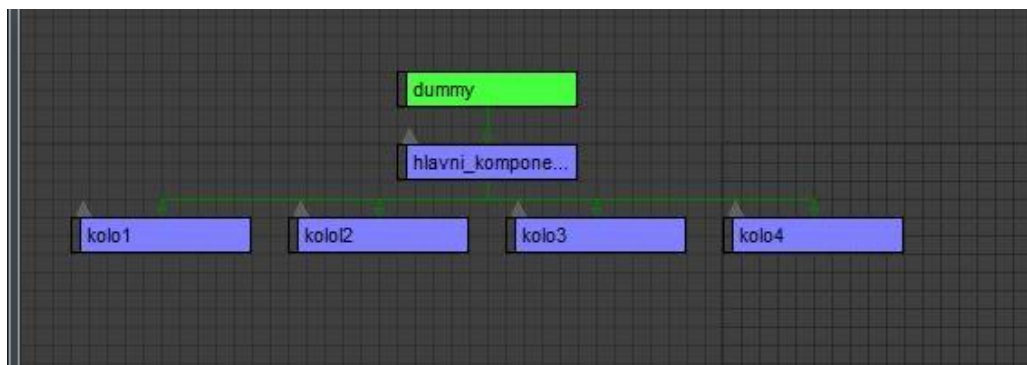
Mezi komplikovanější interaktivní prvky patří simulace vozidel v celém areálu TUL. Vozidla se skládají ze dvou základních prvků, prvním je samotný animovaný model vozidla reprezentovaný jako CGA a druhým je XML soubor, ve kterém jsou nastaveny fyzikální parametry vozidla, jako jsou například maximální rychlost, pozice kamery, váha a další. Animovaný model CGA je velmi podobný klasickému modelu CGF s tím rozdílem, že CGA může obsahovat více než jednu komponentu (například hlavní model vozidla a k tomu příslušnou komponentu, která reprezentuje dveře). Jak lze vidět na obrázku č. 14, samotný CGF model vozidla se skládá z šesti

komponent, první je samotná kostra vozidla, druhou až pátou komponentou jsou kola vozidla s pivotem nastaveným na střed a šestou komponentou je prvek s označením dummy (jedná se o jednoduchý vodící prvek v 3ds maxu, který slouží především pro animace), jehož funkcí je seskupení komponent do jedné skupiny. V této bakalářské práci slouží dummy také jako údaj o pozici přední kamery na vozidle.



Obr. 14 Rozložení komponent na vozidle

Toto seskupení lze provést využitím funkce Graph Editor v 3ds max, kde se komponenty kol přidruží k hlavní komponentě (tělo vozidla). Hlavní komponenta se následně přidruží k dummy, které reprezentuje výslednou komponentu vozidla, viz obrázek č. 15. Takto seskupený objekt se vyexportuje využitím CryExporteru a přiřadí se do příslušného adresáře v CryEngine. Každému vyexportovanému CGA objektu je nutné přiřadit XML skript, kterým bude reprezentován v CryEngine, tak aby byla zajištěna jeho fyzikální interpretace ve scéně. V této bakalářské práci XML skript obsahuje váhu automobilu pro hlavní komponentu, váhu pro jednotlivá kola a také globální vlastnosti jako je rychlost a ovladatelnost vozidla.



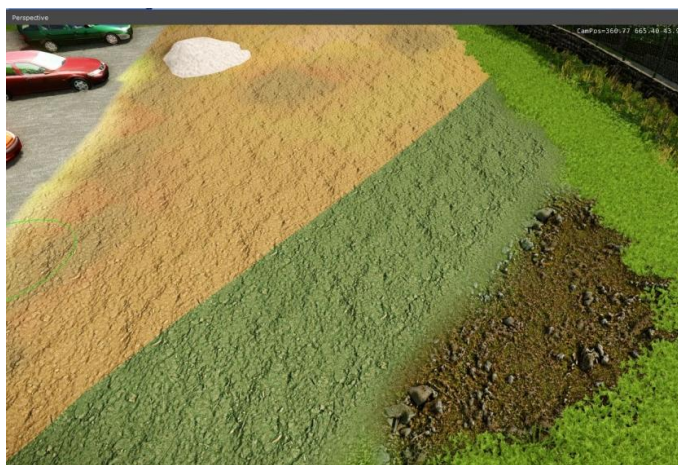
Obr. 15 Seskupení jednotlivých komponent

## 4.4 Neoficiální problémy a jejich řešení

Při tvorbě bakalářské práce docházelo k mnoha problémům, které měly často řešení obsažené přímo v dokumentaci nebo bylo dostupné na oficiálním fóru CryDev. Avšak docházelo také k problémům, u kterých oficiální řešení nebylo nalezeno. V takovém případě bylo nutné tyto problémy vyřešit nebo obejít. Vzhledem k tomu, že neoficiální problémy vyskytující se v této bakalářské práci nebyly příliš časté, ale na druhou stranu byly stěžejní pro tvorbu této bakalářské práce, bylo nezbytné hledat řešení problémů prostřednictvím konzultace na oficiálním fóru, e-mailové komunikace přímo s vývojáři CryEngine nebo v konečném důsledku za pomoci metody pokus-omyl. Jedná se o tyto problémy:

### **Narušení informací o obarvení textur aplikovaných na terénu**

Jedná se o problém, který se vyskytl po pádu aplikace CryEngine v momentě generování textury aplikované na terénu. Následkem tohoto pádu bylo narušení informace o obarvení jednotlivých textur na terénu a vždy při pokusu o namalování nových textur došlo ke ztrátě barev na starých texturách aplikovaných na terénu. Tohoto defektu si lze všimnout na obrázku č. 16.



Obr. 16 Ztráta informací o obarvení textur

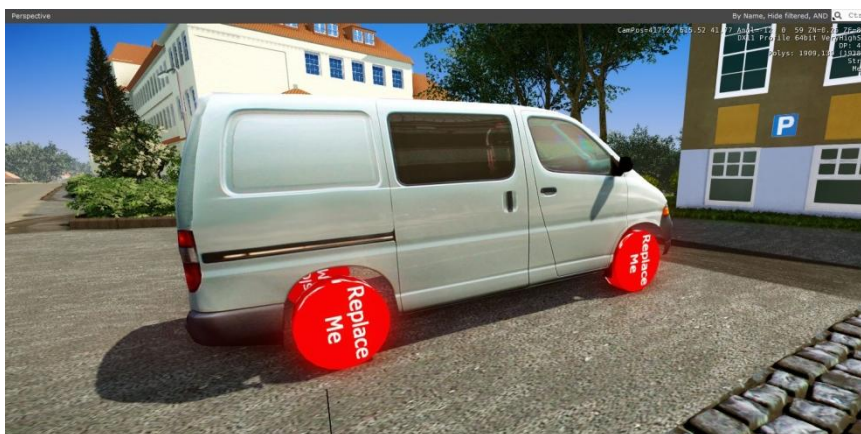
Tento problém byl konzultován přímo se samotnými vývojáři CryEngine pomocí oficiálního fóra a e-mailové komunikace. Ze strany vývojářů byla poskytnuta odezva, že tento problém je v menším množství případů známý, ale řešení momentálně není k dispozici. Z tohoto důvodu došlo k podrobnému nastudování funkce generování textur v dokumentaci a na internetu, a posléze bylo zjištěno, že při pádu aplikace byla



narušena část souboru projektu, která obsahuje informace o obarvení textur aplikovaných na scéně. Díky tomu bylo možné sestavit fungující řešení, jež bylo následně odesláno vývojářům CryEngine, kteří potvrdili jeho funkčnost. Řešení se skládá z exportu textur terénu do formátu BMP, ve kterém je reprezentované zbarvení textur aplikovaných na terén. Následně dojde k vygenerování textur již aplikovaných na terén (textury a jejich zbarvení se exportují do souboru pro přímé použití ve výsledné spustitelné aplikaci). Po vygenerování textur terénu dojde k importování BMP souboru s informacemi o obarvení textur. Díky tomu je možné znovu kreslit nové textury bez poškození těch stávajících, bohužel se tento postup musí opakovat při každém spuštění editoru, vždy když potřeba upravovat textury. Vedlejším efektem tohoto postupu může být blednutí textur v některých částech scény, například na místech, kde se setkává velké množství textur.

### **Problém červených kol u vozidel**

Tento problém se velmi často vyskytuje při špatném postupu vytváření vozidel. Správný postup se ovšem nenachází ani v dokumentaci ani na oficiálním fóru. Problém červených kol se projevuje při importu vozidel do CryEngine, vozidlům zde chybí textura na kolech, která je nahrazena výchozí červenou texturou, viz obrázek č. 17.



Obr. 17 Problém červených kol

Řešení tohoto problému bylo nalezeno po delší konzultaci na oficiálním fóru. Východiskem daného nedostatku je, že se musí nejprve vyexportovat materiál vozidla, který se následně importuje do prostředí CryEngine, kde se materiálu přiřadí příslušné textury. Po přiřazení textur musí dojít v prostředí 3ds max k synchronizaci materiálu daného vozidla z prostředí CryEngine a aplikaci synchronizovaného materiálu na dané

vozidlo. Po uskutečnění synchronizace a aplikování materiálu dochází k odstranění problému červených kol, následně se v tvorbě vozidla pokračuje jako za normálních okolností.



## 5 Závěr

V této bakalářské práci je představena problematika tvorby fotorealistické vizualizace areálu Husova TUL, která je reprezentována užitými nástroji a technologiemi. Zásadní úlohu při vytváření bakalářské práce sehrál vizualizační engine CryEngine 3, dále program Autodesk 3Ds Max, Adobe Photoshop a PixPlant.

Lze konstatovat, že veškeré stanovené cíle a potřebné kroky se podařilo úspěšně splnit. Byly kvalitně otexturovány a upraveny modely poskytnuté TUL, které byly zadány studentům fakulty mechatroniky jako bakalářské projekty v minulých letech. Dále došlo k reálnému vytvoření terénu na základě skutečných výškových dat získaných z programu Google Earth. Tento ojedinělý postup tvorby terénu lze dokonce považovat jako přínos pro praxi, neboť může být užitečnou pomůckou při řešení podobných problémů. Pro dosažení co největšího fotorealistického efektu bylo v bakalářské práci provedeno také dotvoření veškerých externích prvků, od stromů a keřů přes zaparkované automobily, silnice a dlážděné cesty až po interiér jednotlivých budov se snahou o přiblížení se co nejvíce skutečnosti. Výsledkem této bakalářské práce se pak stala kompletní fotorealistická simulace areálu Husova TUL v prostředí CryEngine3.

Při vytváření této práce docházelo k častým chybám a problémům v již existujících řešeních. Jednalo se převážně o omezení, jež byla snadno řešitelná, ale vyskytlo se také mnoho problémů, na které řešení nebylo k dispozici. Tyto otázky mají závažný charakter a výrazně ovlivňující fotorealistický vzhled této bakalářské práce, z tohoto důvodu muselo dojít k analýze problémů a jejich postupnému vyřešení. Jednotlivé výsledky řešení, které jsou v této bakalářské práci prezentovány, lze považovat za přínos pro praxi, neboť tímto mohou pomoci budoucím uživatelům dosahovat při jejich tvorbě daleko lepších výsledků.

Nejen poskytnuté řešení k problémům je přínosem pro praxi, ale rovněž samotná bakalářská práce, která je momentálně jedinou fotorealistickou vizualizací areálu Husova TUL. Při tvorbě byl také brán ohled na budoucí rozšiřitelnost práce. Příkladem rozšíření by mohla být rozsáhlejší optimalizace práce pro slabší sestavy nebo vytvoření umělé inteligence, u které by docházelo k interakci s uživatelem. Jelikož je tato bakalářská práce tvořená ke stavu z roku 2011, mohl by další postup této práce zahrnovat aktuální vzhled areálu Husova a zejména navázat přidáním nových budov.

# Literatura

- [1] AUTODESK. *Autodesk Mudbox* [online]. 2012 [cit. 2013-04-10]. Dostupné z: <http://www.autodesk.com/products/mudbox/overview>.
- [2] BEAMNG. *BeamNG* [online]. 2012 [cit. 2013-04-10]. Dostupné z: <http://www.beamng.com/content/>.
- [3] BURDA, Lukáš. *Interaktivní 3D-model budovy CIT*. Liberec, 2007. Bakalářská práce. Technická univerzita v Liberci.
- [4] CRYTEK. *CryENGINE Directory Structure* [online]. 2010 [cit. 2013-03-15]. Dostupné z: <http://freesdk.crydev.net/display/SDKDOC8/CryENGINE+Directory+Structure>.
- [5] CRYTEK. *Getting Started with the CryENGINE SDK: System Requirements* [online]. 2010 [cit. 2013-03-14]. Dostupné z: <http://freesdk.crydev.net/display/SDKDOC8/Getting+Started>.
- [6] CRYTEK. *ImagTP To Discuss Using CryENGINE 2* [online]. 2008 [cit. 2013-03-12]. Dostupné z: <http://www.crytek.com/news/imagtp-discuss-using-cryengine%C2%AE-2>.
- [7] Mycryengine: physic. CRYTEK. *Technologi: physic* [online]. 2010 [cit. 2013-03-08]. Dostupné z: <http://mycryengine.com/index.php?conid=8>.
- [8] Mycryengine: sandbox. CRYTEK. *Technologi: sandbox* [online]. 2010 [cit. 2013-03-08]. Dostupné z: <http://mycryengine.com/index.php?conid=53>.
- [9] Mycryengine: visual. CRYTEK. *Technologi: visual* [online]. 2010 [cit. 2013-03-08]. Dostupné z: <http://mycryengine.com/index.php?conid=49>.
- [10] CRYTEK. *Texture Compilation* [online]. 2010 [cit. 2013-04-08]. Dostupné z: <http://freesdk.crydev.net/display/SDKDOC3/Using+CryTif>.
- [11] DAFTLOGIC. *Google Maps Find Altitude* [online]. 2011 [cit. 2013-04-20]. Dostupné z: <http://www.daftlogic.com/sandbox-google-maps-find-altitude.html>.
- [12] EMERAN, Riyad. TRUSTEDREVIEWS. *AMD Radeon HD 2900 XT - Real Time Tessellation* [online]. 2007 [cit. 2013-04-04]. Dostupné z: [http://www.trustedreviews.com/AMD-Radeon-HD-2900-XT\\_PC-Component\\_review\\_real-time-tessellation\\_Page-6](http://www.trustedreviews.com/AMD-Radeon-HD-2900-XT_PC-Component_review_real-time-tessellation_Page-6).

- [13] FLANAGAN, Korena. Avatars train on Navy's future ship. AUSTRALIAN GOVERNMENT. *Avatars train on Navy's future ship* [online]. 2011 [cit. 2013-05-04]. Dostupné z: <http://www.defence.gov.au/minister/Claretpl.cfm?CurrentId=11969>.
- [14] GAHAN, Andrew. *3ds Max modeling for games: insider's guide to game character, vehicle, and environment modeling*. Amsterdam: Focal Press. ISBN 02-408-1582-3.
- [15] HOUŽVIČKA, Vojtěch. *Interaktivní 3D model budovy F*. Liberec, 2012. Bakalářský projekt. Technická univerzita v Liberci.
- [16] JINDRÁK, Marek. *Interaktivní 3D model budovy E*. Liberec, 2012. Bakalářský projekt. Technická univerzita v Liberci.
- [17] Kříž, J.: *Mistrovství v 3ds Max*, Computer press, 2010, ISBN 9788025124642.
- [18] MARHOUL, Vojtěch. *Interaktivní 3D model budovy C*. Liberec, 2012. Bakalářský projekt. Technická univerzita v Liberci.
- [19] PODLIPNÝ, Jiří. *Model areálu technické univerzity v Liberci*. Liberec, 2011. Bakalářský projekt. Technická univerzita v Liberci.
- [20] RADEMACHER, Paul. The university of north Carolina. *Ray Tracing: Graphics for the Masses* [online]. 2002 [cit. 2013-03-24]. Dostupné z: <http://www.cs.unc.edu/~rademach/xroads-RT/RTarticle.html>.
- [21] RÁKOS, Daniel. RASTERGRID. *History of hardware tessellation* [online]. 2010 [cit. 2013-04-04]. Dostupné z: <http://rastergrid.com/blog/2010/09/history-of-hardware-tessellation/>.
- [22] ŠULC, Tomáš. PCWORD. *Technologie: Unifikované shadery* [online]. 2008 [cit. 2013-03-20]. Dostupné z: <http://pcworld.cz/hardware/technologie-unifikovane-shadery-co-to-vlastne-je-3865>.
- [23] TEAMXBOX. *Teamxbox* [online]. 2006 [cit. 2013-03-12]. Dostupné z: <http://news.teamxbox.com/xbox/10596/Ubisoft-Acquires-Far-Cry-Intellectual-Property-Perpetual-License-of-CryEngine>.
- [24] TRACY, D., Tracy, S.: *CryEngine 3 cookbook* Packt Publ: Packt Publishing, 2011, ISBN 9781849691062.
- [25] TRACY, Sean a Paul REINDELL. *CryENGINE 3 Game Development: Beginner's Guide*. Anglie: Packt Publishing, 2012. ISBN 1849692009.

- [26] Virgin: freespace. *Radiosity* [online]. 2000 [cit. 2013-03-22]. Dostupné z:  
<http://freespace.virgin.net/hugo.elias/radiosity/radiosity.html>.

## **Příloha A – Obsah přiložených DVD**

Jelikož výsledná aplikace byla příliš velká a nevešla se na jedno DVD, musela být zabalena do dvou archivů \*.rar.

### **Obsah DVD1:**

DVD1 obsahuje (**CryEngine.part1.rar**) první polovinu aplikace, a to větší část archivu.

### **Obsah DVD2:**

DVD2 obsahuje druhou část aplikace, a to zbylou část archivu (**CryEngine.part2.rar**), dále též modely v 3ds max, příslušné textury k nim, fotografie areálu TUL a ovladače, které mohou být potřebné ke spuštění CryEngine 3.

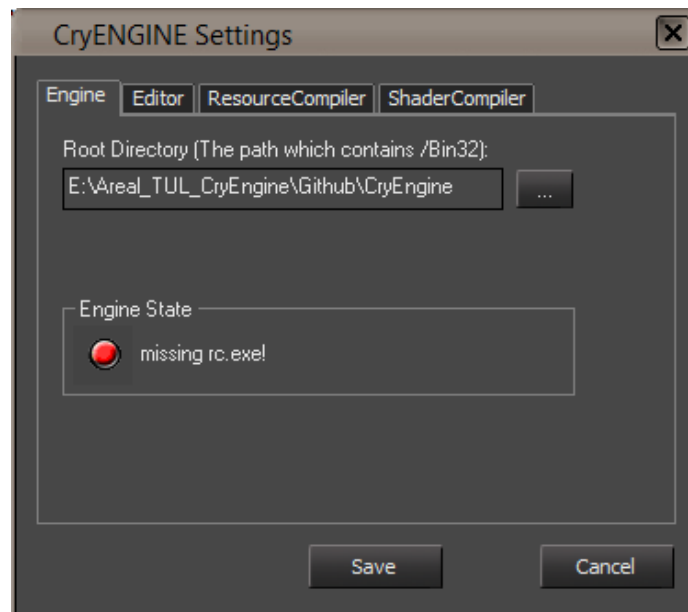
## Příloha B – První spuštění editoru

V této části přílohy se budeme věnovat prvnímu spuštění této bakalářské práce. Jak již bylo zmíněno v úvodu této bakalářské práce, výsledná aplikace by měla být spuštěna alespoň na sestavě splňující minimální hardwarové požadavky, avšak by neměla běžet v prostředí systému Microsoft Windows 8. Dále je vysvětleno, že práce by se měla spouštět v prostředí editoru, neboť vlivem chyby v knihovnách CryEngine dochází ke zkreslení barev mezi editorem a launchrem.

Pro bezproblémové spuštění CryEngine 3 je nutné mít nainstalován nejnovější Microsoft DirectX, dále Microsoft .NET Framework 4 a také se doporučuje aktualizovat ovladače grafické karty na nejnovější verzi. Microsoft DirectX a Microsoft .NET Framework 4 lze nalézt na stránkách Microsoftu nebo na DVD ve složce ovladače v této bakalářské práci.

Jelikož pracujeme s CryEngine 3 ve variantě, která není placená, tak zde CryEngine 3 nepodporuje přímé spuštění, ale požaduje přihlášení k serveru crydev.com. Přihlašovací údaje si můžete vytvořit na serveru crydev.net anebo jako jméno použít ArealTUL a heslo arealtul. Pokud jste se rozhodli použít ke spuštění launcher, naleznete ho ve složce **Bin32** nebo **Bin64**. Pro launcher.exe se doporučuje používat verze Bin32 a tudíž příslušný adresář, ve kterém naleznete soubor launcher.exe. Tento soubor spustíte po zobrazení přihlašovací nabídky, poté vyplníte příslušné přihlašovací údaje a přes položku singleplayer vyberete ArealTUL. Pokud jste se rozhodli pro použití editoru, naleznete ho ve stejném adresáři, ale oproti launcheru se doporučuje použít verzi Bin64 v příslušném adresáři. Před prvním spuštěním editoru je nutné nastavit inicializační nástroj rc.exe, který se stará o komunikaci editoru se samotným enginem nebo o komunikaci mezi externími nástroji a editorem (například exportéry nebo přídatnými plugíny). Tento nástroj naleznete v adresáři **tools** a obsluhuje ho spustitelný soubor **SettingsMgr.exe**. Po spuštění se vám zobrazí okno, které je vidět na následující obrázku č. A.1, kde nastavíte cestu, ve které se nachází adresáře a soubory CryEnginu (například D:\\bakalarska\_prace\\cryengine\\, pozor do cesty nezahrnujte adresář bin32).

Potvrzením této cesty by se měla červená signalizační dioda změnit na zelenou a nápis na „rc.exe found!“. Nyní vše potvrďte tlačítkem save a můžete přejít k samotnému spuštění editoru.



Obr A.1: Resource compiler

Editor se nachází v adresáři Bin32 nebo Bin64 a jmenuje se Editor.exe. Tento soubor tedy spustíme, opět vyplníme přihlašovací údaje a po načtení editoru otevřeme samotný level. To provedeme v nabídce **file** a **open** nebo klávesovou zkratkou Ctrl + O. Vybereme soubor s názvem ArealTUL\_final, pokud by tento soubor nebyl k nalezení je umístěn v adresáři \Game\Levels\ArealTUL\, po jeho vybrání a otevření počkáme na kompletní načtení scény.

Po načtení scény je možné se ve scéně volně pohybovat, a to využitím kláves A, S, D, W a přidržením pravého tlačítka myši pro otáčení kamery. Pro přepnutí do herního módu je nutné se přiblížit k terénu a klávesovou zkratkou Ctrl + G se do něj přepnout. Ovládání je podobné jako u režimu pohybu ve scéně a to tlačítky A, S, D, W a myši pro rozhlížení se ve scéně. K ukončení herního režimu a přechodu do režimu volného pohybu slouží klávesa ESC.

## Příloha C – Obrázky hotové aplikace

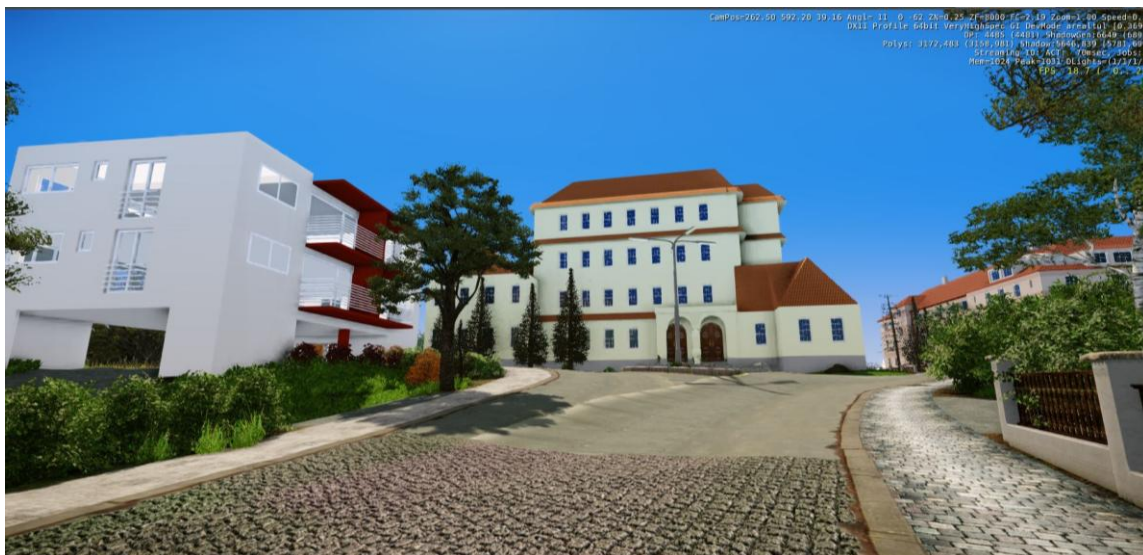


Obr B.2 Ptačí pohled





Obr B.3 Budova B



Obr B.4 Budova A

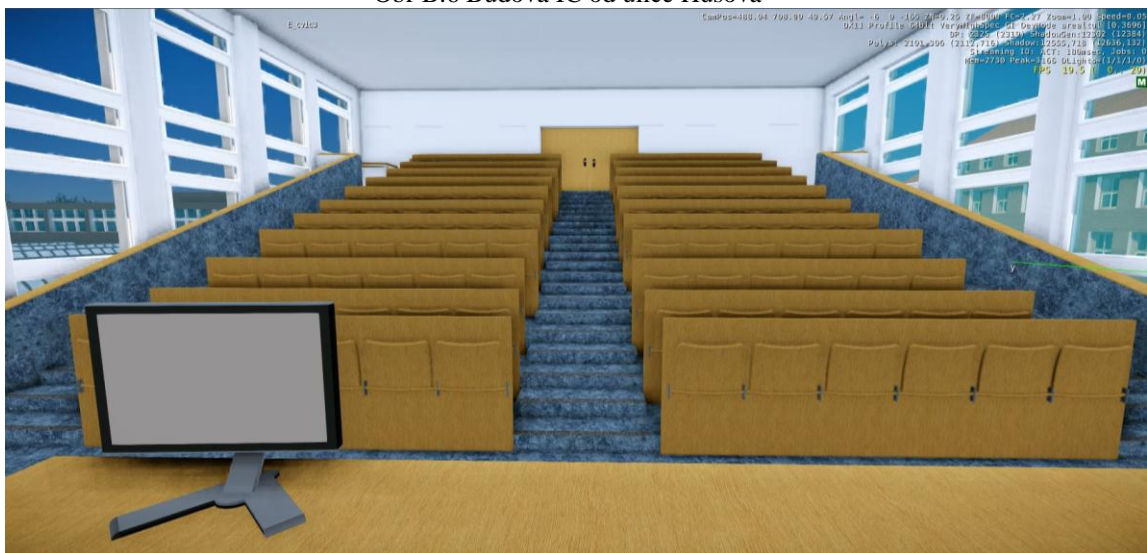


Obr B.5 Budova IC





Obr B.6 Budova IC od ulice Husova



Obr B.7 Místnost E9



Obr B.8 Demontrace rozsahu detailů v okolí areálu TUL