TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

3D MODEL AREÁLU HUSOVA PRO WEBOVÉ APLIKACE

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: Marek Jindrák
Vedoucí práce: Ing, Jiří Jeníček, Ph.D.

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Marek Jindrák**
Osobní číslo:	**M10000158**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**3D model areálu Husova pro webové aplikace**
Zadávající katedra:	**Ústav informačních technologií a elektroniky**

Zásady pro vypracování:

1. Za použití dodaných materiálů vytvořte realistický 3D model areálu Husova Technické univerzity v Liberci.
2. Model musí být plynule zobrazitelný v reálném čase na webu.
3. Pro vývoj použijte prostředí Unity.

Rozsah grafických prací:	**Dle potřeby dokumentace**
Rozsah pracovní zprávy:	**cca 30 stran**
Forma zpracování bakalářské práce:	**tištěná/elektronická**

Seznam odborné literatury:

[1] **Kříž, J.: Mistrovství v 3ds Max, Computer press, 2010, ISBN 9788025124642**

[2] **Goldstone, W.: Unity Game Development Essentials, 2009, ISBN 978-1847198181**

Vedoucí bakalářské práce:	**Ing. Jiří Jeníček, Ph.D.** Ústav informačních technologií a elektroniky
Datum zadání bakalářské práce:	**1. října 2013**
Termín odevzdání bakalářské práce:	**16. května 2014**

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
vedoucí ústavu

V Liberci dne 1. října 2013

Prohlášení

Byl(a) jsem seznámen(a) 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(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Současně čestně slibuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum 15.5.2014

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 bych rád poděkoval společnosti Autodesk, která zpřístupnila studentské licence k softwaru 3d studio max.

Velké díky patří také vývojářům Unity 3D za jejich podporu na oficiálním fóru.

Abstrakt CZ

Cílem bakalářské práce je vytvoření interaktivní prohlídky areálu Technické univerzity v Liberci. Hlavním kritériem této práce je nutnost využití herního enginu „Unity“. Pro splnění cílů závěrečné práce bylo potřeba nejdříve provést úpravu obdržených podkladů, které tvoří 3D modely jednotlivých budov a jejich zastaralé stavební plány. Poté následovalo jejich zpracování pomocí nástrojů „GIMP“, programu na úpravu grafiky, a prostřednictvím programu pro 3D grafiku „Autodesk 3D studio Max“. Cílem této práce není vytvoření fotorealistické simulace, nýbrž vytvoření aplikace, která umožní uživateli interaktivní prohlídku univerzitním areálem ve webovém prohlížeči, která vychází z podoby univerzity v roce 2011.

Abstrakt EN

This bachelor thesis deals with composing of a virtual simulation of the campus at the Technical University of Liberec. The major criteria of this thesis is to present the necessity of using the game engine „Unity“. To satisfy the demands of the dissertation was firstly to modify obtained documentation which were 3D models of particular buildings and outdated building plans. Folllowing procedure was to process the materials using the tools of graphics editing software ”GIMP” and the 3D graphics program „Autodesk 3D studio Max“. The objective of the bachelor thesis is not to create a photorealistic simulation but to make an application, that enables an interactive tour around the university campus in an internet browser. For the purpose of this simulation the form of the university from 2011 was applied.

Obsah

Prohlášení 3

Poděkování 4

Abstrakt CZ 5

Abstrakt EN 6

Obsah 7

Seznam symbolů a zkratek 10

Seznam obrázků 11

Seznam tabulek 12

1 Úvod 13

1.1 Cíl práce 13

1.2 Podklady 13

1.3 Nástroje 13

2 Práce v Autodesk 3D studiu Max 15

2.1 Import modelů a jejich rozmístění 15

2.2 Vytvoření modelu terénu 15

2.3 Texturování 16

2.3.1 Rastrové textury 16

2.3.2 Procedurální textury 16

2.3.3 Vytváření rastrových textur 16

2.3.4 Odstranění efektu dlažebních kostek 17

2.4 Mapování 18

2.4.1 Rozdělení modelu na více menších modelů 18

2.4.2 Využití materiálu Multi/sub-object 18

2.5 Export dat z 3D studia Max 19

3 Unity obecně 20

3.1 Rozhraní aplikace 20

3.1.1 Toolbox 20
3.1.2 Hierarchy 21
3.1.3 Scene 21
3.1.4 Game 21
3.1.5 Project 21
3.1.6 Ispector 22
3.2 Komponenty 22
3.3 GameObjects 23
3.3.1 Světla 23
3.3.2 Geometrické tvary 24
3.3.3 Prefabs 24
4 Práce v Unity 26
4.1 Import modelů 26
4.2 Vytváření materiálů modelů 27
4.2.1 Diffuze material 27
4.2.2 Bumped Diffuze material 27
4.2.3 Specular material 28
4.2.4 Bumped specular material 28
4.2.5 Skleněný materiál 28
4.3 Přiřazení kolizních map 29
4.4 Vytvoření interaktivních oken 29
4.4.1 Chyby interaktivních modelů 31
4.5 Vytvoření interaktivních dveří 32
4.5.1 Chyba samootvíracích dveří 32
4.6 Avatar 33
4.6.1 Kamera 33
4.6.2 First person 34

4.6.3 Third person 34
4.6.4 Vytvoření avatara 35
4.7 Vytváření skriptu pro Unity 36
4.7.1 Popis skriptu 36
4.8 Nastavení osvětlení 39
4.8.1 Stíny ve výsledné aplikaci 39
4.9 Zvýšení realističnosti 40
4.9.1 Využití tranzitivních textur v praxi 40
4.10 Nastavení pozadí 40
4.11 Náročnost aplikace 41
4.12 Vytvoření finální verze 41
5 Závěr 43
Použitá literatura 45
Příloha A – Obrázky hotové aplikace 46

Seznam symbolů a zkratek

TUL	Technická univerzita v Liberci
Frame per second	Počet obrázků za vteřinu
Avatar	Herní postava
Spot light	Směrové světlo
Point light	Bodové světlo
Directional light	Globální světlo
First person	Pohled z první osoby
Thrid person	Pohled třetí osoby

Seznam obrázků

Obrázek 1: Rozložení budov v areálu TUL 15
Obrázek 2: Ukázka texturování krychle 16
Obrázek 3: Efekt dlažebních kostek 17
Obrázek 4: Rozříznutí textury 17
Obrázek 5: Otočení řezu 17
Obrázek 6: Správné mapování 18
Obrázek 7: Chybné mapování 18
Obrázek 8: Model areálu TUL 19
Obrázek 9: Rozhraní Unity 20
Obrázek 10: Podokno Projekct 21
Obrázek 11: Podokno Inspector 22
Obrázek 12: Přehled základních komponent 22
Obrázek 13: Point light 23
Obrázek 14: Directional light 23
Obrázek 15: Spot light 23
Obrázek 16: Soft Shadows 23
Obrázek 17: Hard shadows 23
Obrázek 18: Základní geometrické tvary 24
Obrázek 19: Import 3D modelu 26
Obrázek 20: Bumped diffuse material 27
Obrázek 21: Diffuse material 27
Obrázek 22: Specular bumped material 28
Obrázek 23: Specular material 28
Obrázek 24: Skleněný materiál 29
Obrázek 25: Síťová kolizní mapa 29
Obrázek 26: Nastavení kolizní mapy 30
Obrázek 27: Pohyb kolizní mapy a okna 30
Obrázek 28: Kolizní mapa okna 30
Obrázek 29: Přiřazení skriptu objektu 31
Obrázek 30: Chyba vzniklá změnou velikosti modelu 31
Obrázek 31: Neošetřená zóna mezi kolizními mapami 32
Obrázek 32: Přidělená kolizní mapa dveřím 32
Obrázek 33: Kolizní mapa ve tvaru koule - otevřené dveře 33
Obrázek 34: Kolizní mapa ve tvaru koule - zavřené dveře 33
Obrázek 35: First person pohled 34
Obrázek 36: Third person pohled 34
Obrázek 37: Character Controller 35

Obrázek 38: Základní model avatara *35*
Obrázek 39: Character Controller nastavení *35*
Obrázek 40: Mouse look skript *35*
Obrázek 41: Character motor nastavení *36*
Obrázek 42: Nastavení vlastního skriptu *37*
Obrázek 43: Zvýraznění interaktivní komponenty *38*
Obrázek 44: Stín stromu *39*
Obrázek 45: obrázek s flare *39*
Obrázek 46: Obrázek bez flare *39*
Obrázek 47: Nedokonalý stín stromu *40*
Obrázek 48: Scéna s pozadím *41*
Obrázek 49: Scéna bez pozadí *41*
Obrázek 50: Podporované platformy *42*

Seznam tabulek

Tabulka 1: Podporované formáty *27*
Tabulka 2 Přehled náročnosti sítí *41*

1 Úvod

Rozmach počítačové grafiky za posledních pár let je tak veliký, že to co se zdálo před pár lety jako nemožné je dnes pomocí grafických nástrojů možné vytvořit zcela jednoduše. Rozvoj počítačové grafiky nastartovali nejen zvyšující se nároky hráčů na počítačovou grafiku her ale spíš společnosti zabývající se tvorbou filmů. U filmařských společností byl zájem o kvalitu vizuálních efektu ještě vyšší než zájem hráčů počítačových her. Se zvyšující se kvalitou počítačové grafiky však také rostou nároky na hardware, na kterém vizuální efekty vznikají. V současné době jsou ve společnostech zabývajících tvorbou vizuálních efektů využívány tzv. renderovací farmy. Renderovací farma je vysoce výkonný počítačový systém, který zvládne vytvářet jednotlivé efekty mnohem rychleji než samostatný počítač.

V roce 2013 se vyskytla myšlenka využít současné technologie k tomu, abychom ostatním uživatelům umožnili procházku po univerzitním areálu z pohodlí domova. Jedním z nejvyužívanějších programů v dnešní době jsou webové prohlížeče. Z tohoto důvodu by měla být tato virtuální prohlídka přístupná přes webový prohlížeč.

1.1 Cíl práce

S rozvojem počítačové grafiky do podoby jak ji dnes známe, se vyskytla myšlenka této bakalářské práce. Základní myšlenkou této bakalářské práce je vytvoření interaktivní procházky areálem Technické university v Liberci (dále jen TUL), tak aby byla zobrazitelná přes webový prohlížeč. Uživatel by si tedy mohl z pohodlí svého prohlížeče projít areálem TUL a dokonce i některými vybranými budovami.

1.2 Podklady

Při vytváření interaktivní procházky areálu TUL byly použity některé modely, které vznikly v rámci ročníkových prací. Jednalo se hlavně o budovy C, E, F a IC. Dále byly poskytnuté plány budov A a B. tak, aby bylo možné sestavit celý areál TUL. Ostatní budovy vyskytující se v tom areálu byly vytvářeny z fotografických podkladů.

1.3 Nástroje

Výběr nástrojů pro tvorbu této práce byl relativně úzký. Existovala možnost využít několika různých modelovacích 3D nástrojů pro tvorbu modelů. Mezi nejznámější softwary zabývající se tvorbou 3D modelů patří například Autodesk 3D studio Max, Autodesk Maya či Blender. Každý z těchto softwarů má své výhody.

Velkou výhodou softwaru Blender je jeho přenositelnost mezi operačními systémy Windows a Linux. Při tvorbě modelů byl však využíván Autodesk 3D studio Max.

Výběr herního enginu byl silně omezen zadáním této práce. Na trhu však existuje množství herních enginů, ve kterých jsou v dnešní době vytvářeny hry. Mezi nejznámější herní enginy patří například CryEngine, Unity, UnrealEngine. CryEngine vyniká oproti ostatním vyjmenovaným enginům svým fotorealistickým vzhledem výsledné aplikace. Nevýhodou je však fakt, že aplikace běžící na tom to enginu zabírají velké množství místa na disku. Z tohoto důvodu nebyl vyžit tento herní engine při tvorbě této práce. Pro vytvoření této práce byl vybrán herní engine Unity, který sice nevyniká svým grafickým zpracováním výsledné aplikace, ale oproti předchozím enginům vyniká svým snadným exportem do formátů, které jsou snadno zobrazitelné webovým prohlížečem a výsledná velikost těch to aplikací není až tak náročná na místo na disku jako v případě CryEnginu.

2 Práce v Autodesk 3D studiu Max

Na začátku práce bylo nejprve potřeba sjednotit formáty modelů, se kterými se bude nadále pracovat. Většina modelů byla uložena ve formátu FBX až na model budovy IC, který byl uložen ve formátu studia Maya (.ma). Bylo tedy nutné všechny formáty modelů synchronizovat. Aby byl model ve stejném formátu jako ostatní, bylo ho potřeba otevřít ve studiu, ve kterém vznikl tedy v Maye a následně exportovat do formátu FBX.

2.1 Import modelů a jejich rozmístění

Pro práci s modely byla potřeba provést akce importu všech modelů do jedné scény. Po této činnosti bylo nutné rozmístit jednotlivé modely budov tak, aby odpovídaly reálnému umístění budov v areálu TUL viz.Obrázek 1.

Obrázek 1: Rozložení budov v areálu TUL

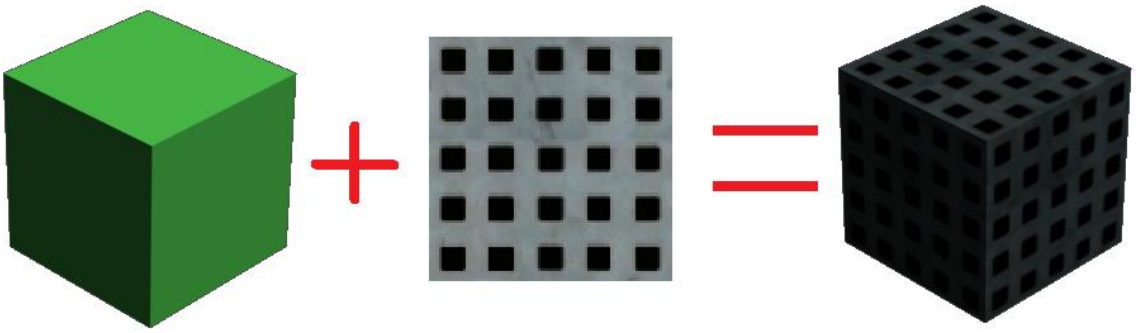
2.2 Vytvoření modelu terénu

Aby model areálu TUL byl kompletní, bylo potřeba vytvořit model terénu. Tento model nebyl doposud vytvořen. Pro vytvoření tohoto modelu byly použity základní modelovací metody. Nejprve byl vytvořen plane v odpovídajících velikostech. Na tento plane byla pak nanesena textura areálu viz. **Chyba! Nenalezen zdroj odkazů.**. ásledně byl tomuto objektu přiřazen modifikátor *Editable Poly*. Díky tomuto modifikátoru lze snadno přizpůsobit polygonální síť reálnému terénu areálu TUL.

2.3 Texturování

Obecně tímto pojmem označujeme metodu obarvení povrchu 3D modelů různými texturami. Důležité je však to, že tato metoda nemění geometrické vlastnosti texturovaných modelů. Jde pouze o změnu zobrazení jejich povrchu viz. Obrázek 2

Pod pojmem textura si můžeme představit jakýkoliv dvoudimenzionální obrázek. Textury můžeme dělit na rastrové nebo procedurální.



Obrázek 2: Ukázka texturování krychle

2.3.1 Rastrové textury

U rastrových textur jde o předem připravený obrázek, který mohl vzniknout kreslením, focením či scanováním obrázku do počítače. U takto vzniklých textur je důležitá detailnost textury. Tato textura má totiž významný dopad na kvalitu vzhledu povrchu modelu. Velmi častými formáty, ve kterých jsou rastrové textury ukládány jsou JPEG,BMP nebo TGA.

2.3.2 Procedurální textury

Na rozdíl od rastrových textur jsou procedurální textury většinou vytvářeny pomocí matematických funkcí. Z tohoto důvodu nezáleží na rozlišení textury, jelikož se textura přizpůsobuje velikosti modelu. Nevýhodou těchto textur je však fakt, že nelze vyjádřit všechny modely pomocí matematických funkcí a tedy použít je obecně na všechny typy objektů

2.3.3 Vytváření rastrových textur

Pro potřebu tohoto projektu bylo zapotřebí vytvořit specifické textury tak, aby se materiálově co nejvíce podobaly reálným materiálům vyskytujících se na budovách v areálu TUL. Pro tvorbu rastrových textur jsou nejčastěji využívány rastrové editory jako jsou například: Adobe Photoshop nebo GIMP. V této práci byl využit rastrový

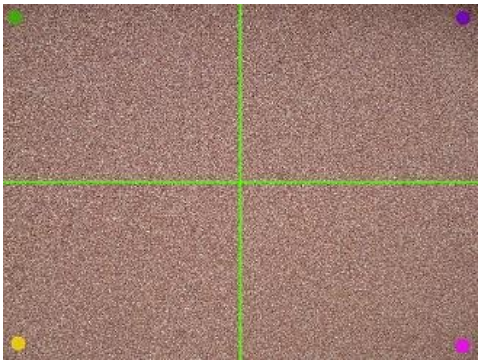
editor GIMP. Jeho výhodou oproti Adobo Photosopu je fakt, že jeho používání je zcela zdarma a pro účely této práce jeho funkčnost postačuje.

Jak již bylo řečeno v kapitole 2.3.1 rastrové textury často vznikají z fotek. Při použití takto vzniklých textur musíme brát v potaz objekt, na který chceme tuto texturu nanést. V případě, že se tato neupravená textura duplikuje na jednom modelu vícekrát může vznikat efekt „dlažebních kostek“ viz. Obrázek 3.

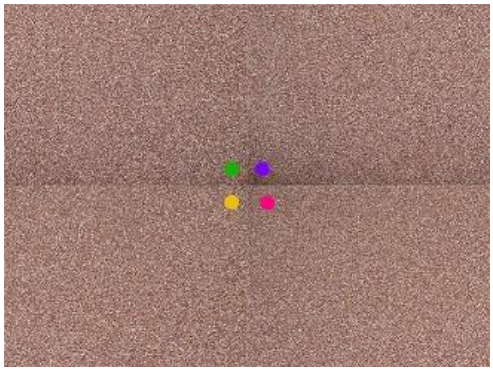
Obrázek 3: Efekt dlažebních kostek

2.3.4 Odstranění efektu dlažebních kostek

Odstraněním tohoto efektu, který nejčastěji vzniká při tvorbě textur z fotografických podkladů získáme texturu, která je opakovatelná. Postup je následující: Nejprve je potřeba zjistit velikost fotografického podkladu. Na základě těchto rozměrů je potřeba obrázek rozříznout na 4 stejné díly viz. Obrázek 4. Tyto díly je pak potřeba



Obrázek 4: Rozříznutí textury

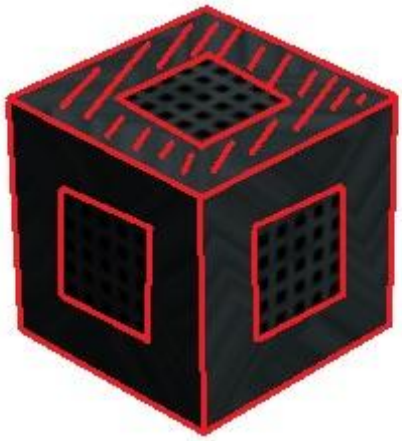


Obrázek 5: Otočení řezu

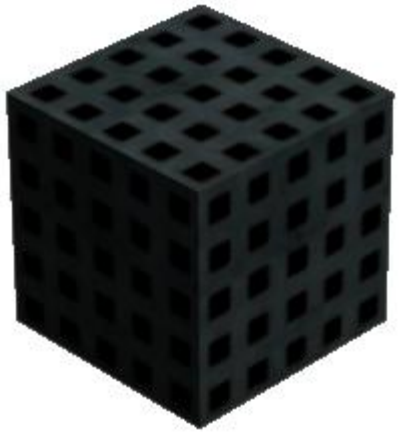
otočit o 180 stupňů viz. Obrázek 5. Nyní jsou na tomto obrázku již viditelná místa, na kterých jsou ostré přechody jasně znatelné. Tato místa je potřeba vyhladit pomocí nástrojů, které rastrový editor nabízí. Po vyhlazení všech míst je potřeba otočit díly zpět do původní polohy.

2.4 Mapování

Mapování je způsob jakým se na model nanáší textura. Jednoduše se dá tedy říct, že mapa určuje místo na modelu, na které se textura promítne. Bez přidání map se na složitějších modelech nezobrazí textura korektně viz. Obrázek 7. Proto je vhodné všem modelům přiřazovat mapy. V rámci 3D studia Max se tato operace provádí tak, že se objektu přiřadí modifikátor UVW map. Po korektním přidělení mapy model vypadá takto viz. Obrázek 6



Obrázek 7: Chybné mapování



Obrázek 6: Správné mapování

2.4.1 Rozdělení modelu na více menších modelů

Jedná se o metodu rozřezání objektu dle vybraných polygonů na menší části. Každá z takto vyříznutých částí se stává novým objektem. Takto vzniklému objetu přiřazujeme následně materiál, který obsahuje informace pouze o jedné textuře vhodné k tomuto objektu. Po přidělení materiálu přidáme modelu také modifikátor *Poly Select* pomocí něhož vybereme všechny polygony, kterým chceme přidat mapu. Mapu přidáme modifikátorem *UVV map*. V menu toho modifikátoru vybereme tvar, který nejvíce odpovídá vybranému modelu v tomto případě box. Tato metoda je snadnější, ale ve velkých projektech takřka nepoužitelná.

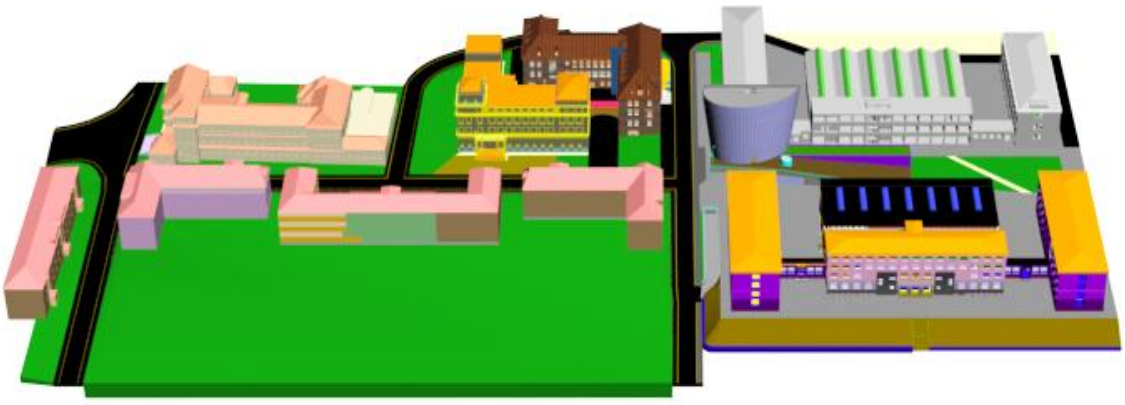
2.4.2 Využití materiálu Multi/sub-object

Vyváření M*ulti/sub-object* materiálu je časově náročnější, ale efektivita této metody je mnohem vyšší. Využitím této metody se nevytvářejí další objekty, což přispívá k přehlednosti celé scény. Jedná se o metodu, která je založena na přiřazování materiálového ID jednotlivým polygonů nad daným modelem. Podle počtu přiřazených materiálových ID se pak vytváří *Multi/sub-object* materiál. Tento matriál je v zásadě schránka, která obsahuje seznam materiálů viz. **Chyba! Nenalezen zdroj odkazů.**.

aždý záznam tohoto seznamu je reference na určitý druh materiálu, který je následně přiřazen dle ID svým cílovým polygonům. Po přiřazení *Multi/sub-object* materiálu modelu je nutné vytvořit UVW mapy. Ze seznamu modifikátorů přístupných danému objektu vybereme modifikátor *Poly Select*. Mezi atributy tohoto modifikátoru patří i možnost vybrání polygonů na základě přiřazeného materiálového ID. Tomuto výběru pak přiřadíme modifikátor UVW map. A takto postupujeme, dokud nepřiřadíme mapu všem materiálům, které jsou na modelu použity. Výhoda této metody je, že tento materiál nemusí být vytvářen pro různé modely znova. Každý materiál má pak tedy na všech modelech stejné vlastnosti.

2.5 Export dat z 3D studia Max

Pro export dat lze vybrat celou řadu formátů, které Autodesk 3D studio nabízí. Nejpoužívanější formát pro export a následný import do jiného vývojového prostředí je obvykle formát FBX. Po výběru formátu je možné dále specifikovat, které prvky z dané scény mají být nutně součástí daného exportu. Tedy zdali má být vyexportována pouze data týkající se polygonové sítě, či tato data mají být obohacena o animační klíče nebo osvětlení. Dále zde lze definovat způsob, jak výsledná síť bude vypadat. Jinak řečeno zdali má síť zachovat polygonovou strukturu nebo se má tato síť převést na síť trojúhelníkovou. Vyexportovaný areál Technické univerzity v Liberci je pak vidět na obrázku. viz. Obrázek 8



Obrázek 8: Model areálu TUL

.

3 Unity obecně

Unity je nástroj sjednocující několik platforem pro tvoru aplikací, které fungují v běžném internetovém prohlížeči. Nutná je pouze instalace zásuvného modulu (pluginu), stejně jako u dnes běžně rozšířeného Adobe Flash. Další nespornou výhodou Unity 3D je funkčnost i mimo webové prohlížeče. Aplikaci lze snadno zprovoznit například na mobilním telefonu iPhone či herní konzoli Nintendo Wii, Xbox360 nebo PlayStation 3.

3.1 Rozhraní aplikace

Rozhraní aplikace Unity se skládá z několika podoken. Každé z těchto podoken má svůj specifický účel. Rozmístění těchto podoken je proměnné. Tedy uživatel si smí vytvořit vlastní rozložení podoken tak, aby mu byla maximálně ulehčena práce v tomto prostředí. Rozhraní je rozděleno do šesti základních podoken. viz. Obrázek 9

Obrázek 9: Rozhraní Unity

3.1.1 Toolbox

Toolbox je jedna z nejvíce využívaných částí této aplikace, jelikož se zde vyskytují všechny potřebné nástroje pro práci se scénou. Nalezneme zde tedy nástroje transformační (pohyb, rotace, zvětšení/zmenšení modelu). Mezi další důležité nástroje tohoto panelu patří toolbar, jehož pomocí lze nastavovat pivot daného modelu. A v neposlední řadě se zde vyskytuje toolbar ovládající podokno „game“ (play, pause).

3.1.2 Hierarchy

V panelu hierarchy jsou uvedeny všechny instance modelů, které se v dané scéně vyskytují. Každá z těch to instancí pak může obsahovat další strukturální seznam přímo dané instance. Jednotlivé instance pak můžeme vkládat od prázdných objektů a tím vytvářet požadovanou hierarchii instancí.

3.1.3 Scene

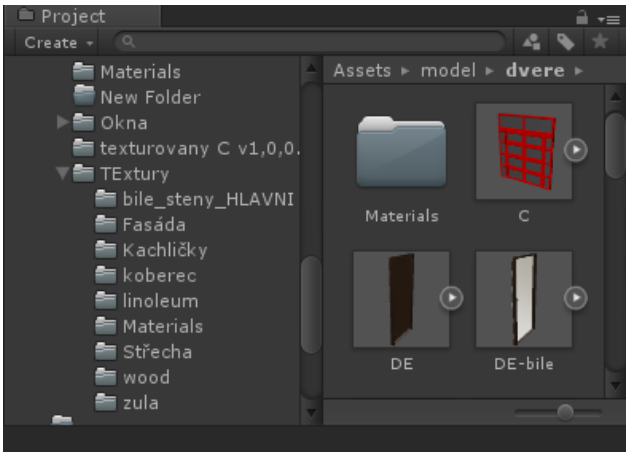
Scéna je dalším velmi důležitým podoknem v této aplikaci. V tomto podokně se vytváří celková kompozice výsledné aplikace. Jinak řečeno v tomto podokně dochází k rozmisťování jednotlivých modelů po scéně tak, aby ve výsledku vytvořily kompletní model požadované aplikace. V tomto podokně pak nejvíce využíváme nástroje dostupné z toolboxu. Při práci v tomto podokně je uživateli usnadněna práce se scénou pomocí zkratek, které umožňují pohyb, rotaci a další prvky důležité při práci se scénou.

3.1.4 Game

Kliknutím na tlačítko play v toolboxu dojde k aktivaci okna „game“. Toto podokno se pak přepne do tzv. play módu. Výhodou toho spuštění je fakt, že nemusí docházet k opětovné kompilaci při testování této aplikace. Další výhodou i nevýhodou tohoto módu je možnost upravovat běžící aplikaci. Tyto změny jsou zachovány pouze při běhu tohoto módu. Při opuštění play módu se dané změny neprojeví na testované scéně.

3.1.5 Project

Všechny projekty vytvořené Uniti obsahují složku asset. Obsah této složky se zobrazuje pak v podokně „Project“ viz. Obrázek 10. Tato složka pak obsahuje všechny soubory, které vytvářejí konečný ráz aplikace. Do této složky jsou tedy ukládány

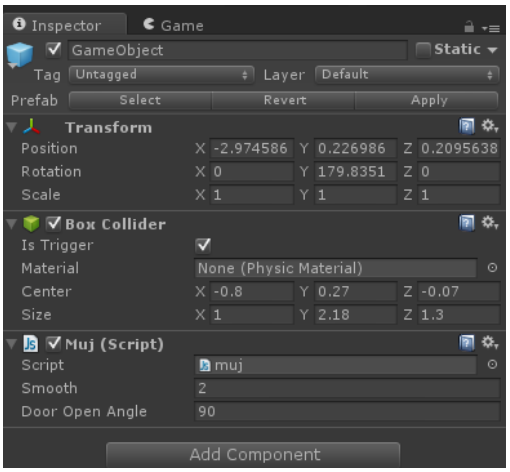


Obrázek 10: Podokno Projekct

jednotlivé 3D modely a jejich textury, různé skripty, zvukové nahrávky a tzv, prefab soubory. U těchto souborů není doporučována manipulace pomocí operačního systému. Často tento přístup může poškodit data, se kterými takto manipulujeme.

3.1.6 Ispector

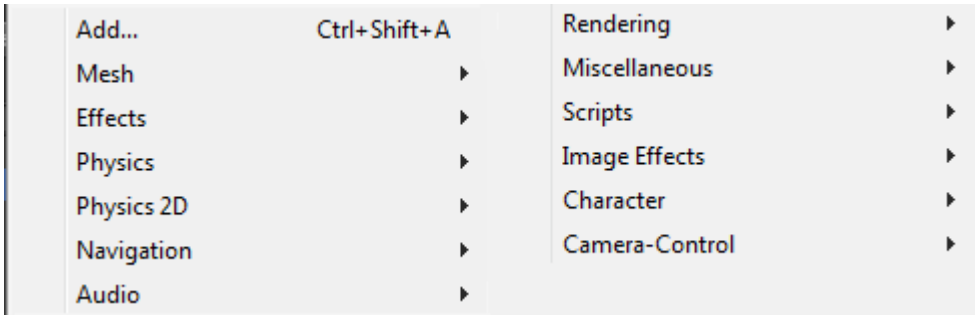
Jak již bylo řečeno aplikace vytvářené v prostředí Unity obsahují vždy několik herních objektů. Každý z těchto objektů může mít své specifické nastavení, jako je například textura, kolizní mapa, způsob jakým tento objekt odráží světlo nebo skripty, které jsou tomuto objektu přiřazeny. viz. Obrázek 11.



Obrázek 11: Podokno Inspector

3.2 Komponenty

Komponenty jsou prvky, které pomáhají při tvorbě výsledné aplikace a přímo ve výsledné hře jsou znatelné. Mezi základní komponenty můžeme zařadit Physisc, Audio,



Obrázek 12: Přehled základních komponent

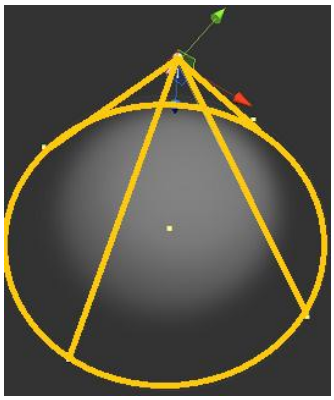
Rendering, Script a další viz. Obrázek 12. Pomocí komponenty Physisc můžeme pak objektu, kterému tuto komponentu přiřadíme nastavovat kolizní mapu. Pomocí komponenty skript pak může ovlivnit, jakým způsobem se daný objekt bude chovat.

3.3 GameObjects

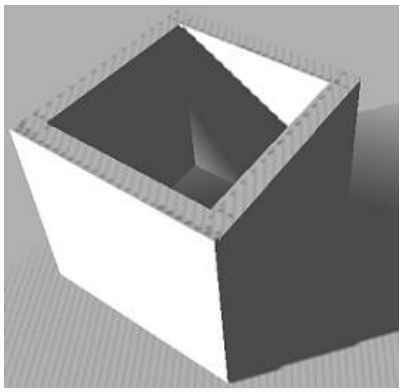
Herní objekty ovlivňují ráz vytvářené aplikace. Jedná se tedy o prvky dodávající této aplikaci různé druhy vizuálních detailů. Mezi tyto objekty patří světla, základní geometrické tvary (box, koule, …), kamery, ale také částicový systém.

3.3.1 Světla

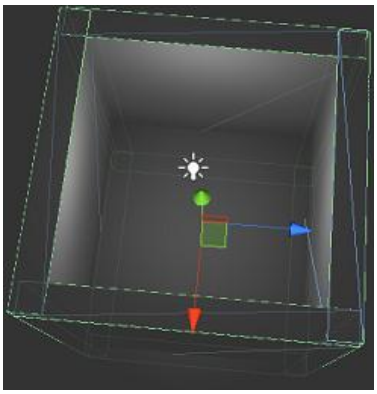
V dnešní době se bez kvalitního osvětlení při tvorbě her nelze obejít. Světla všeobecně přispívají k lepší reálnosti výsledného pocitu z vytvářeného 3D prostředí. Prostředí Unity nabízí k využití tři typy světel. Tato světla lze v rámci scény libovolně kombinovat tak, aby bylo docíleno maximální realističnosti scény. U světel lze obecně



Obrázek 15: Spot light



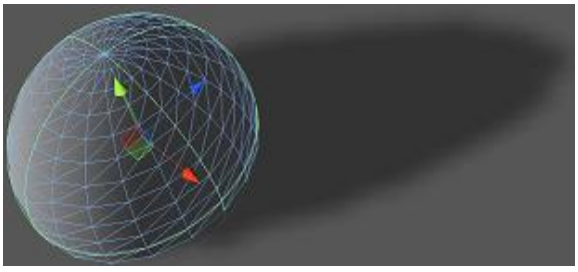
Obrázek 14: Directional light



Obrázek 13: Point light

nastavovat jejich barvu a také stíny, které při kolizi s objektem vytvoří. Lze nastavit dva druhy stínů. Tvrdé a měkké stíny. Při nastavenní tvrdých stínů jsou hranice osvětlené části a stínu od sebe oddělny ostrým přechodem. Při nastavení měkkých stínů je tato hranice rozmazána podle přesně zadané hodnoty rozostření. Viz. obrázky Obrázek 16, Obrázek 17.

Obrázek 17: Hard shadows



Obrázek 16: Soft Shadows

3.3.1.1 Spot light

Jde o směrové světlo, kterému lze nastavit vzdálenost a poloměr plochy, kterou má toto světlo osvěcovat. Důležitým parametrem tohoto světla je též intenzita. Dále

tomuto světlu lze nastavit tzv. „coocies“. Jde o šablonu, která bude promítnuta na dopadovou plochu světla. Toto světlo lze přirovnat ruční k svítilně, nebo automobilovému reflektoru. viz. Obrázek 15..

3.3.1.2 Directional light

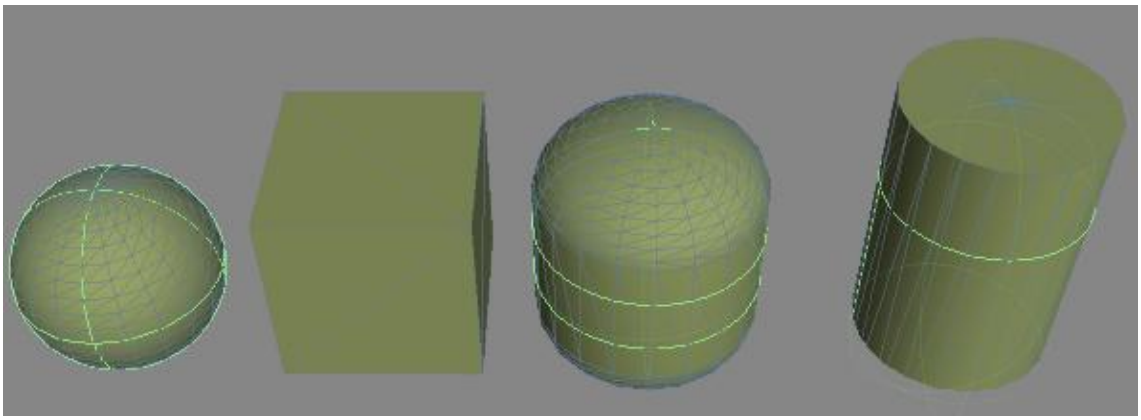
Jde o plošné světlo, které je nekonečně vzdálené od všech objektů ve scéně. Tomuto světlu lze nastavit pouze intenzita. Plochu, kterou toto světlo má osvítit nastavit nelze, jelikož se jedná o plošné světlo, které osvítí celou scénu. Toto světlo bývá v projektech často využíváno jako slunce.

3.3.1.3 Point light

Toto světlo bývá označováno jako bodové. To je z důvodu, že paprsky tohoto světla se šíří na všechny strany a vychází všechny z jednoho stejného bodu. Tomuto světlu lze též nastavit rádius a intenzita světla. Ve hrách se toto světlo objevuje v lampách či v v místech, která jsou osvětlována žárovkou.

3.3.2 Geometrické tvary

Do kategorie herních objektů jsou zahrnuty i všechny základní geometrické tvary, ze kterých jdou odvodit i jiné modely. viz. Obrázek 18



Obrázek 18: Základní geometrické tvary

3.3.3 Prefabs

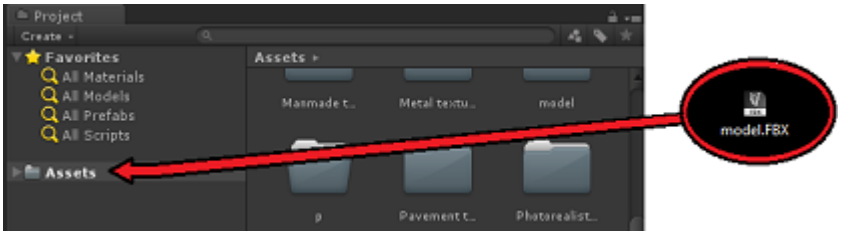
Prefabs může být zvuk, 3D model nebo jakýkoliv jiný herní objekt vyskytující se v prostředí Unity. Prefabs jsou hlavní stavební bloky pro všechny druhy herních objektů, které se vyskytují ve scéně vícekrát a mají mít stejné parametry. Vkládáním prefabů do scény se nevytváří vždy nové objekty, nýbrž instance daného prefabu. To je výhodné z toho hlediska, že pokud se provede změna na prefabu projeví se tato změna i

na všech instancích daného prefabu. Pokud tedy přiřadíme materiál, skript či kolizní mapu hlavnímu prefabu obdrží tedy i všechny instance stejné komponenty.

4 Práce v Unity

4.1 Import modelů

Import souborů do prostředí Unity může probíhat v zásadě dvěma způsoby. Soubor, který chceme importovat do prostředí, nahrajeme do složky assets. Tato varianta importu však není doporučována, jelikož tímto způsobem může dojít k poškození importovaných dat. Lepší variantou importu modelů do prostředí je přetažení modelů do projektového okna. To lze jednoduše provést metodou „drag and drop“ viz. Obrázek 19. Tento způsob importu modelů je oproti jiným nástrojům



Obrázek 19: Import 3D modelu

poměrně nepraktický. Další velikou nevýhodou tohoto importu je fakt, že materiály, které jsou vytvořeny v 3DS Maxu nejsou korektně přeimportovany do prostředí Unity. U všech přeimportovaných materiálů zůstane správně jen přiřazené jméno materiálu a jeho difuzní barva. Ostatní informace jsou tímto importovacím způsobem zapomenuty. Je tedy nutné všechny materiály vytvořit znovu. Seznam podporovaných formátů vhodných pro import viz. Tabulka 1

3D formats	Meshes	Textures	Anims	Bones
Maya (.mb, .ma)	ANO	ANO	ANO	ANO
3Ds Max (.max)	ANO	ANO	ANO	ANO
Cheetah 3D (.jas)	ANO	ANO	ANO	ANO
Cinema 4d (.c4d)	ANO	ANO	ANO	ANO
Blender (.blend)	ANO	ANO	ANO	ANO
Modo (.lxo)	ANO	ANO	ANO	NE
Autodesk (.FBX)	ANO	ANO	ANO	ANO
COLLADA	ANO	ANO	ANO	ANO
Carrara	ANO	ANO	ANO	ANO
Lightwave	ANO	ANO	ANO	ANO
XSI 5.x	ANO	ANO	ANO	ANO
SketchUp Pro	ANO	ANO	NE	NE

Wings	ANO	ANO	NE	NE
3D Studio (.3ds)	ANO	NE	NE	NE
Wavefront (.ob)	ANO	NE	NE	NE

Tabulka 1: Podporované formáty

4.2 Vytváření materiálů modelů

Jak již bylo řečeno v předchozí kapitole prostředí Unity nedisponuje dobrým importačním nástrojem. Při importu dochází ke ztrátě nastavených materiálů. Z toho důvodu je zbytečné vytvářet komplikovanější materiály v prostředí 3D studia Max. V prostředí 3D studia Max pouze přiřadíme modelu materiály obsahující difuzní barvu materiálu. Je též vhodné tyto materiály správně pojmenovat. Správné pojmenování pak usnadní práci v prostředí Unity. V Unity existuje několik základních typů materiálů, které je možno využít bez nutnosti vytváření vlastního pixelshaderu. Pro vytvoření speciálních materiálů je potřebná znalost problematiky programování v jazycích python, javascript nebo C#.

4.2.1 Diffuze material

Jde o základní materiál obsahující specifickou barvu daného modelu. Tomuto materiálu lze přiřadit textura. Na rozdíl od 3D studia Max tato textura nepřekryje difuzní barvu, ale dojde k jejich smísení. Tato skutečnost pak může být využita, když potřebujeme použít stejnou texturu s jiným barevným odstínem. Tento materiál nevytváří odlesky a nelze tomuto mu přidělit detail. viz. Obrázek 21

Obrázek 21: Diffuse material

Obrázek 20: Bumped diffuse material

4.2.2 Bumped Diffuze material

Jedná se o rozšířenou verzi materiálu diffuse. Tento materiál přijímá kromě textury také normálovou mapu. Pomocí této mapy dochází ke zlepšení kvality vzhledu výsledného materiálu. Tento materiál nemění geometrické vlastnosti modelu, přitom tento dojmem vzbuzuje. viz. Obrázek 20

4.2.3 Specular material

Jedná se také o základní materiál. Tomuto materiálu lze přidělit pouze texturu. Na rozdíl od diffuse materiálu tento materiál vytváří odlesky. Proto je vhodný k modelům s lesklými materiály. Nevýhodou tohoto základního materiálu je fakt, že nelze nastavit detail materiálu. viz. Obrázek 23

Obrázek 23: Specular material

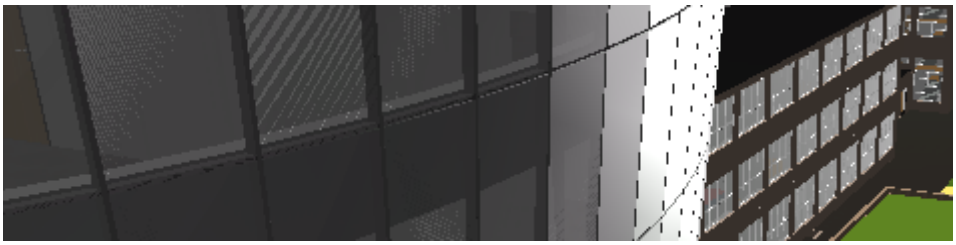
Obrázek 22: Specular bumped material

4.2.4 Bumped specular material

Stejně jako v předchozích materiálech jde o vylepšení svého rodiče. Jde o specular material, který je obohacený o normálovou mapu. Tento materiál stejně jako jeho rodič vytváří odlesky. Doplnění základního materiálu o normálovou mapu vytváří dojem, že tento materiál mění geometrii modelu. Jde ale pouze o dojem. viz. Obrázek 22.

4.2.5 Skleněný materiál

Unity v základní verzi nepodporuje shadery, které dokážou vykreslit skleněný materiál. Pro vytvoření lze postupovat různými způsoby. Prvním a nejlehčím způsobem je zakoupení příslušného pixelshaderu, který toto vykreslení zvládne. V našem případě tato varianta nepřipadá v úvahu. Druhou možností je vytvoření transparentní textury, kterou aplikujeme na materiál a několikrát tuto texturu znásobíme. Výsledkem této metody je průhledný materiál, ktery nebudí dojem reálného skla. Třetí možností je vytvoření pixelshaderu v jednom z podporovaných programovacích jazyků. Tato možnost dopřává uživateli maximální volnost, ovšem tato varianta je časově náročná a bez znalosti tvorby pixelshaderů téměř nemožná. Poslední variantou je použít nějaký materiál z volně dostupných balíčků, které jsou volně stažitelné na oficiálních stránkách. Pro tento účel se náramně hodí balíček „Hard Surface“. Tento balíček je volně dostupný a obsahuje materiály i příslušné pixelshadery, který lze využít pro tvorbu skleněného materiálu. Přesně řečeno, tento balíček obsahuje materiály drahých

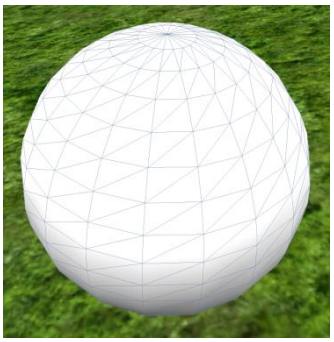


Obrázek 24: Skleněný materiál

kamenů. Na Obrázek 24 si lze prohlédnout tento materiál, který je aplikovaný na modelu budovy Ic v areálu TUL. Tento materiál byl použit i pro vyplnění skleněných vitrín u oken a některých dveří vyskytujících se v této scéně. Na výše uvedeném obrázku si lze též všimnout, že skleněný materiál odráží dopadající paprsky světla správně a také, že tento materiál je průhledný a přirovnatelný kvalitě skla.

4.3 Přiřazení kolizních map

Ve scéně, ve které se chceme pohybovat avatarem nesmíme zapomenout přiřadit kolizní mapy těm objektům, se kterými vytvořený avatar má kolidovat. V zásadě jde o to, že bez přiřazených kolizních map jsou všechny modely průchozí skrze svojí polygonální síť. Mapami můžou být všechny primitivní geometrické obrazce, jako jsou například krychle, kužel nebo koule. Na Obrázek 25 lze vidět přiřazenou polygonální



Obrázek 25: Síťová kolizní mapa

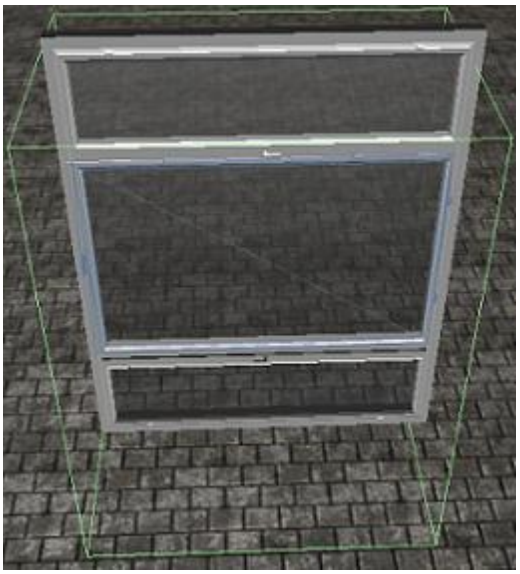
kolizní mapu na kouli. Kolizní mapou může být však i polygonová síť sama o sobě. Kolizní mapy nemusí obecně sloužit jen tomuto účelu, ale může jich být zneužito pro tvorbu triggeru.

4.4 Vytvoření interaktivních oken

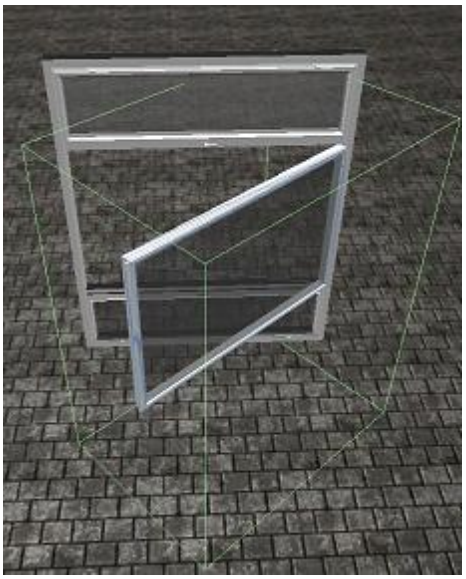
Vytváření samotných modelů oken nejdříve proběhlo ve 3D studiu Max, V tomto studiu byla vytvořena nejen samotná polygonová síť reprezentující model okna, ale byla zde přirazena tomuto modelu i mapa sloužící k následnému texturování v prostředí Unity. Tento model byl následně ve formátu FBX vyexportován z prostředí

3D studia Max. Výsledný model byl následně importován do prostředí Unity. Takto importovanému 3D modelu bylo zapotřebí vytvořit adekvátní materiály tak, aby byly co nejvíce shodné s materiálem reálných oken. Model okna musel být od základu vytvářen tak, aby mohla být část tohoto modelu použita jako interaktivní. Jinak řečeno model musel být rozdělen na minimálně dvě části (rám, okno). S takto připraveným modelem bylo teprve možno začít pracovat.

Připravený model pak přetažením z projektového podokna vložíme do scény. Do scény pak vložíme též prázdný herní objekt, který bude sloužit jako pivot. Okolo toho pivotu se bude daný model otáčet. Tento herní objekt sloučíme dohromady s objektem okna. Ve scéně pak umístíme tento objekt do místa, ve kterém se nacházejí panty daného okna. V podokně toolbar přepneme na možnost rotace vůči pivotu. Nyní

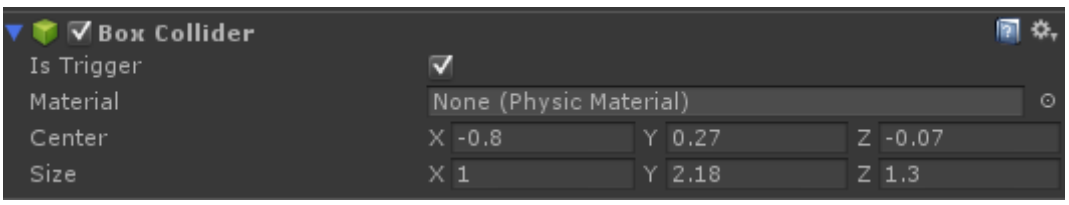


Obrázek 28: Kolizní mapa okna



Obrázek 27: Pohyb kolizní mapy a okna

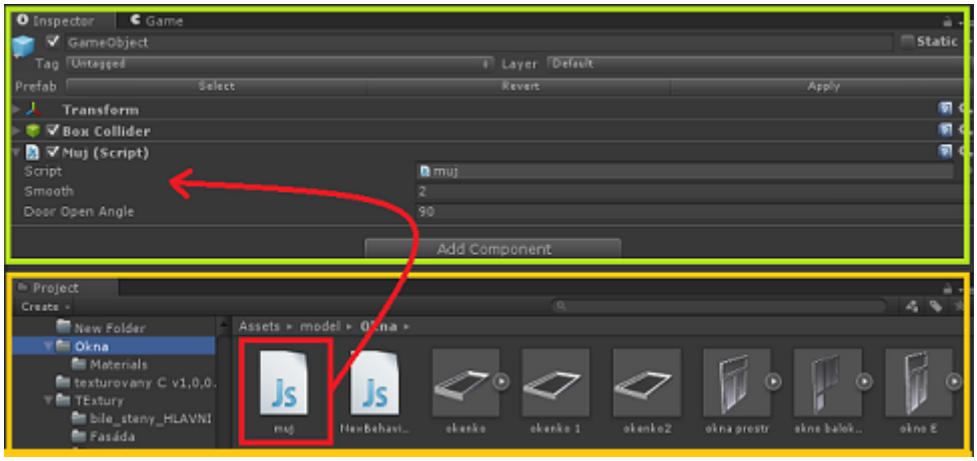
bylo potřeba naplnit prázdný herní objekt modelem časti okna, kterému chceme přidat interaktivu. To provedeme v podokně hierarchie přetažením části modelu do prázdného herního objektu. Tomuto hernímu objektu pak přiřadíme kolizní mapu, která bude sloužit jako spouštěč otevření či zavření okna viz. Obrázek 28. Této kolizní mapě je



Obrázek 26: Nastavení kolizní mapy

nutno nastavit správné umístění a velikost. V nastavení této kolizní mapy pak musí být

zaškrtnuta možnost „Is Trigged viz. Obrázek 28. V tomto kroku je vhodné otestovat, že se daný model i přeřazená kolizní mapa pohybují podle daného pivotu viz. **Chyba! enalezen zdroj odkazů.**. Dalším krokem bylo přiřazení skriptu, který bude s daným herním objektem pohybovat. Přiřazení toho skriptu provedeme přetažením skriptu z projektového podokna do podokna inspektor u daného objektu viz. Obrázek 29.

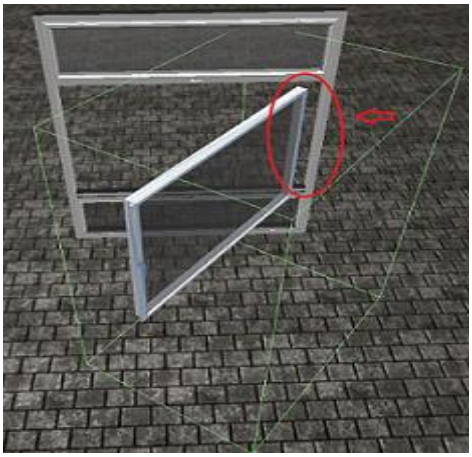


Obrázek 29: Přiřazení skriptu objektu

Pro takto upravený model je výhodné vytvořit vlastní prefab. Kliknutím pravým tlačítkem myši do projektového okna vytvoříme prefab, do kterého nakopírujeme námi modifikovaný model. Při rozmisťování jednotlivých oken po scéně pak vytváříme jen instance tohoto prefabu.

4.4.1 Chyby interaktivních modelů

Tato metoda vytváření oken je vhodná pouze pro modely, u kterých se nebude dále měnit velikost. Chyba nastane při zvětšování nebo zmenšování modelu. Změnou velikosti se totiž nerovnoměrně posune i pivot, kolem kterého se otáčí 3D model. Tento



Obrázek 30: Chyba vzniklá změnou velikosti modelu

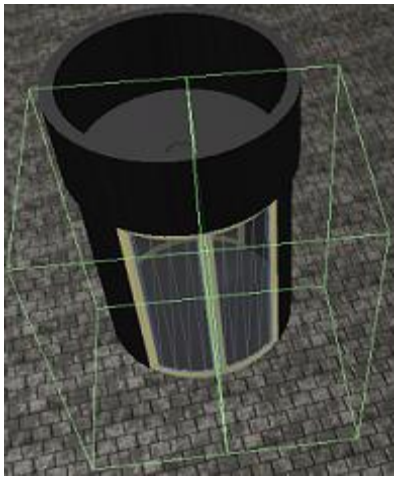
posun pivotu pak zajistí, že se otáčí v jiném místě než by se ve skutečnosti otáčet mělo. Tato chyba je dobře viditelná na Obrázek 30. Tato chyba se v této práci občas objevuje. Výskyt této chyby je dán kvalitou obdržených 3D modelů. Velikosti výřezů pro okna se velice často liší, i když by tyto velikosti měly být podle skutečnosti stejné. Řešením této chyby by bylo pro každý výřez udělat zvláštní model. Toto řešení by bylo nejen časově náročné ale výsledná aplikace by byla znatelně vetší něž v případě, když jsou použity instance jednoho modelu.

4.5 Vytvoření interaktivních dveří

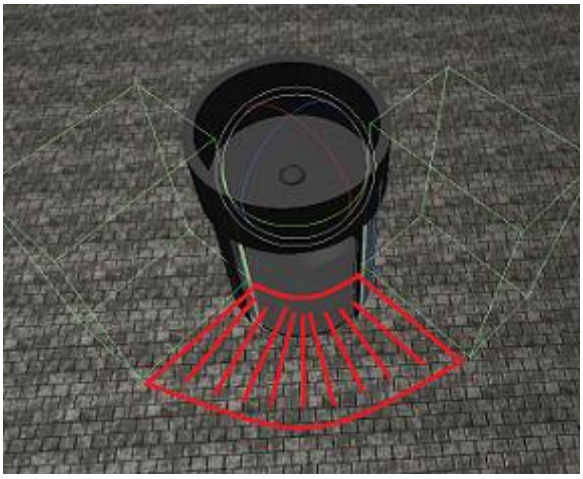
Postup vytváření interaktivních dveří je v zásadě stejný, jak bylo popsáno v kapitole 4.4 zabývající se tvorbou interaktivních oken. Za zmínku by zde však stálo, že pro ovládání dveří muselo být využito více skriptů. Pro běžné dveře vyskytující se v této scéně mohl být použit stejný skript jako pro otevírání oken. Existují však samootvírací dveře vyskytující se v budově Ic, pro které tento skript nemohl být použit. Jelikož první varianta skriptu je spouštěna stlačením tlačítka v blízkosti modelu, který má být otevřen. Což neodpovídá parametrům samootvíracích dveří.

4.5.1 Chyba samootvíracích dveří

Modifikace skriptu pro samootvírací dveře byla poměrně snadná. Skript byl modifikován tak, že když se avatar vykytuje v kolizní zóně dveří, tak se dveře otevřou a zůstanou otevřeny po dobu, kdy avatar pobývá této zóně. Po opuštění kolizní zóny dveří se dveře automaticky zase zavřou. S touto modifikací se však vyskytl problém s kolizní



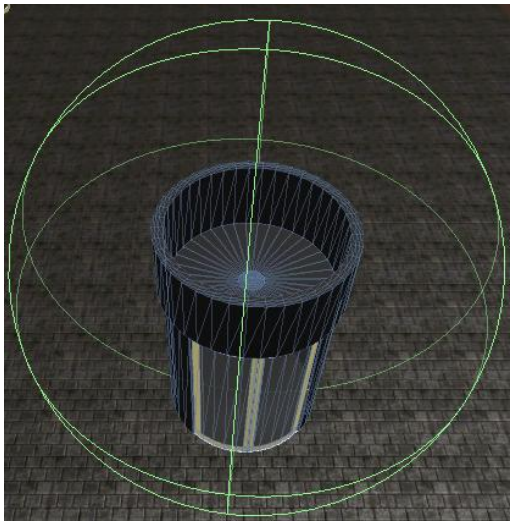
Obrázek 32: Přidělená kolizní mapa dveřím



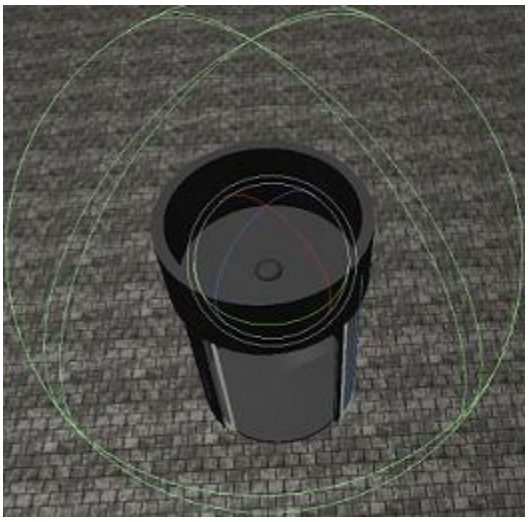
Obrázek 31: Neošetřená zóna mezi kolizními mapami

mapou. Běžně přidávanou kolizní mapou byl totiž box viz. Obrázek 32. Kolizní box se během otevírání dveří vzdálil od avatara. Když se kolizní boxy dveří vzdálily na tolik,

že se nedotýkaly avatara, tak se dveře začaly se automaticky zavírat viz. Obrázek 31. Než došlo k opětovnému kontaktu kolizní mapy a avatara. Výsledkem bylo chaotické otvírání a zavírání dveří. Řešení tohoto problému byla dvě. První varianta počítala s vytvořením animace otevírání dveří bez zakomponování kolizní mapy do modelu. Výhodou této varianty je fakt, že se kolizní mapa nebude hýbat což nebude způsobovat chaotické otevírání a zavíraná těchto dveří. Vytvořením této animace se však zvyšuje i velikost výsledné aplikace. Jelikož tato aplikace je tvořená pro webový prohlížeč, tak by měla být co nejmenší proto byla využita varianta dvě. Varianta dvě spočívá v nahrazení kolizní mapy boxu jiným geometrickým tvarem kolizní mapy. Pro tento účel se tedy nejvíce hodí kolizní mapa ve tvaru koule. Rotováním koule podle svého středu nemůže dojít ke vzdálení avatara od této mapy viz. Obrázek 34.



Obrázek 34: Kolizní mapa ve tvaru koule - zavřené dveře



Obrázek 33: Kolizní mapa ve tvaru koule - otevřené dveře

4.6 Avatar

Slovem avatar bývá v prostředí her označována herní postava, kterou hráč této hry ovládá.

4.6.1 Kamera

Aby bylo zaručeno, že se jakýkoliv pohyb avatara zobrazí uživateli je potřeba přidat této herní postavě přidat kameru. Tato kamera bude snímat virtuální svět a zobrazovat ho uživateli na monitor.

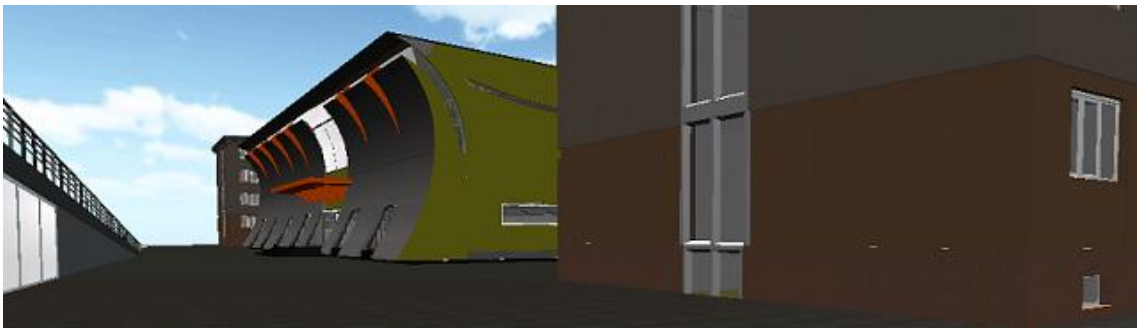
Ve hrách bývají využity dva typy kamer: statické a dynamické. Statickým kamerám je přesně nadefinovaná poloha pohledu a nejde nijak měnit. Těchto kamer

bývá ve hrách obvykle více. Hráč hrající tuto hru se pak smí mezi jednotlivými kamerami v této hře libovolně přepínat.

Dynamická kamera je ovládaná hráčem hrající tuto hru. Tento hráč pak má možnost měnit vzdálenost či úhel natočení pohledu kamery. Tyto změny jsou obvykle prováděny pomocí pohybu myši. Tato kamera pak poskytuje hráči volnost v rozhledu po virtuálním světě. Dynamická kamera bývá nejčastěji využita v akčních hrách nebo v dnes velmi oblíbených „Massively-Multiplayer Online Role Playing Game“ dále jen MMORPG hrách.

4.6.2 First person

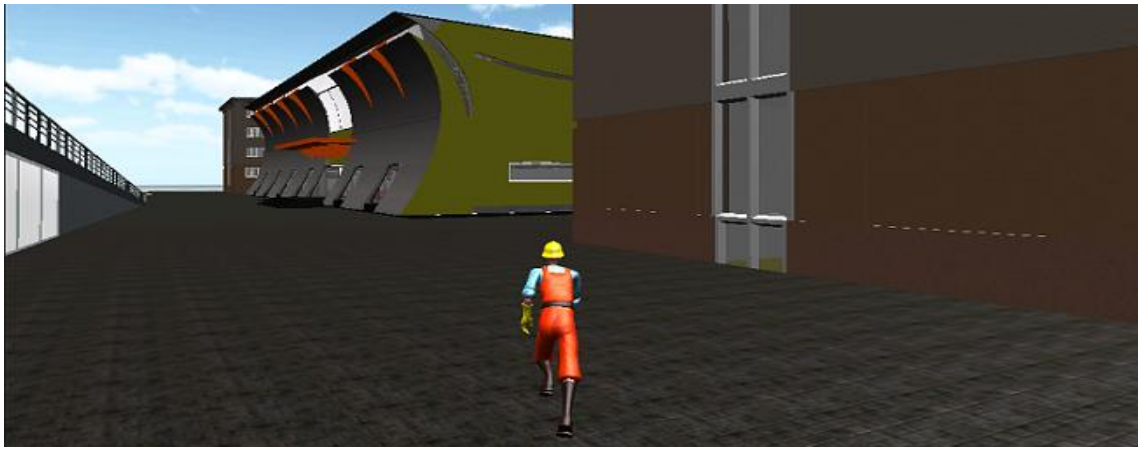
Jedná se o pohled z očí herní postavy viz. Obrázek 35. Tento to pohled vtahuje hráče do hry, jelikož vytváří dojem jako by hráč byl avatarem. S tím to typem zobrazení se můžeme setkat u her jako jsou Half-life, Quake a další.



Obrázek 35: First person pohled

4.6.3 Third person

Jde o pohled třetí osoby viz. Obrázek 36. Postava, kterou hráč ovládá je vidět z určité vzdálenosti či úhlu. Tuto vzdálenost si muže obvykle hráč sám upravit. Tento pohled na herní postavu přináší hráči větší přehled o věcech, které se v dané virtuální



Obrázek 36: Third person pohled

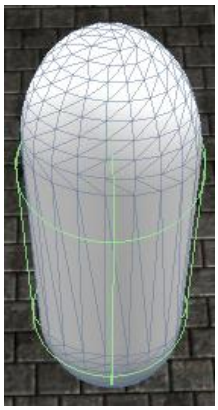
scéně dějí. Pohledu třetí osoby bylo využito např. ve hře Max Pain, Tomb Raider a v dalších.

4.6.4 Vytvoření avatara

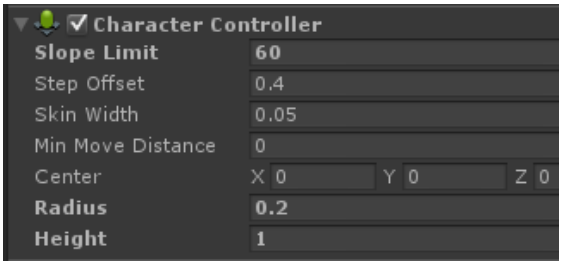
V předchozích podkapitolách byly popsány nejvyužívanější nastavení kamer. Pro tuto aplikaci byl vytvořen avatar, který na scénu shlíží z pohledu první osoby.

Nejprve bylo potřeba vytvořit objekt, který se bude danou virtuální realitou pohybovat. Pro tento prvek byl vybrán model kapsle viz. Obrázek 38. Modelu bylo pak

Obrázek 38: Základní moc avatara



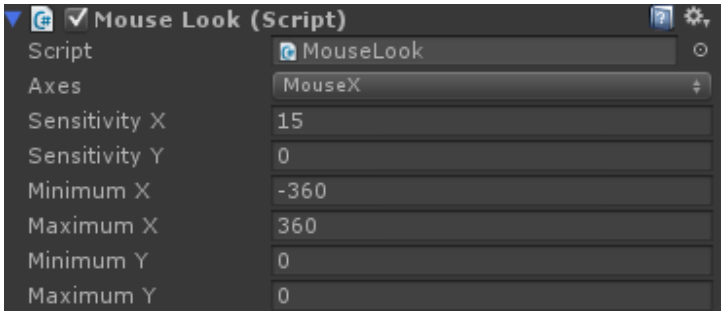
Obrázek 37: Character Controller



Obrázek 39: Character Controller nastavení

nutno přidělit komponentu „Character Controller“. Na Obrázek 37 lze vidět přidělený Character Controller modelu kapsle. Přiřazením této komponenty objektu získáme možnost nastavit tomu to objektu výšku a poloměr kolizní mapy. viz. Obrázek 39. Dále také můžeme nastavit maximální sklon terénu, který bude moci tento model překročit. Lze zde také nastavit jak vysoké překážky dokáže tento model vystoupat. Chybné nastavení této položky se pak muže projevovat například při pohybu na schodech.

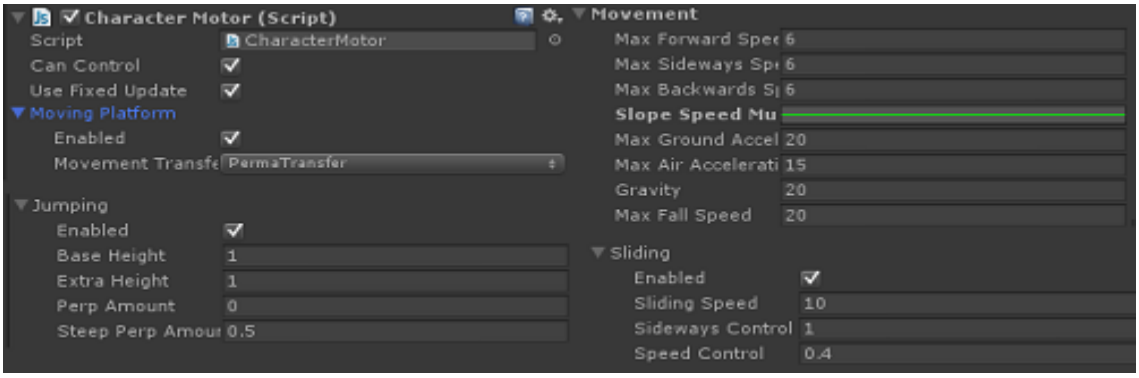
Následně tomuto objektu musí být přiřazeny další tři skripty, které slouží k ovládání avatara. Jde o skripty „Mouse Look, Charecter Motor, FPSInput Controller“. Skript „Mouse Look“ slouží k nastavení otáčení objektu podle pohybu myši. Detaily



Obrázek 40: Mouse look skript

nastavení viz. Obrázek 40. V tomto nastavení lze tedy nastavit senzitivitu myši, maximální a minimální hodnotu, o kterou smí být daný objekt otočen.

Dalším důležitým skriptem je „Character Motor“. Tento skript slouží pro



Obrázek 41: Character motor nastavení

nastavení pohybu objektu, kterému je přidělen. Nastavení tohoto skriptu je možno vidět na Obrázek 41. Zaškrtnutím položky „Can Control“ je zajištěno, že tento objekt lze ovládat. Dále zde jde nastavit jestli avatar muže skákat a bližší nastavení skoku. Důležitým nastavení zde je záložka „Movement“. Zde dochází k nastavení rychlosti pohybu avatara, gravitační zrychlení a maximální rychlost pádu. Dalo by se tedy říct, že se jedná o hlavní skript, co se týče nastavování vlastností avatara.

4.7 Vytváření skriptu pro Unity

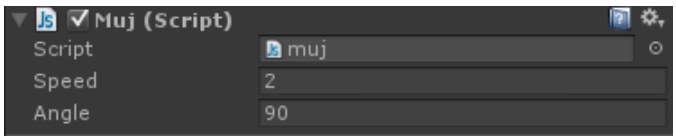
Prostředí Unity podporuje skripty v jazyce javascript, python nebo C#. Záleží tedy na programátorovi, jaký z uvedených jazyků pro tvorbu skriptu využije.

V rámci této bakalářské práce by vytváření skriptů nebylo potřebné. Pro demonstraci funkčnosti byly však vytvořeny skripty ovládající otevírání a zavírán dveří a oken. Skripty byly vytvářeny v jazyce javasript.

4.7.1 Popis skriptu

Pro snadnější nastavování parametru z prostředí Unity do skriptu je nutné vložit public proměnné. Hodnotu v těchto proměnných lze pak měnit v nastavení skriptu viz. Obrázek 42.

```
1.      var speed = 2.0;
2.      var angle = 90.0;
```



Obrázek 42: Nastavení vlastního skriptu

Následně byly vytvořeny privátní proměnné pro potřebu ovládání algoritmu. Do proměnné open, která je definovaná jako boolean je ukládaný stav, ve kterém se okna nacházejí (otevřené, zavřené). Do stejně definované proměnné enter je ukládán stav přítomnosti avatara. Proměnná defaultRot je definovaná jako vektor. Do této proměnné jsou ukládány výchozí souřadnice objektu, kterému je tento skript přiřazen. Proměnná openRot je definovaná též jako vektor a do této proměnné jsou ukládány souřadnice otočených modelů.

```
3.      private var open : boolean;
4.      private var enter : boolean;
5.      private var defaultRot : Vector3;
6.      private var openRot : Vector3;
```

Po prvním spuštění se provede funkce „Start“, které přepočítá souřadnice dle daných parametru a uloží je do proměnné openRot.

```
7.      function Start(){
8.      defaultRot = transform.localEulerAngles;
9.      openRot = new Vector3 (defaultRot.x,
        defaultRot.y - angle, defaultRot.z); }
```

Hlavní metodou tohoto skriptu je metoda „Update“. Tato metoda reaguje na změnu proměnné open a enter. Za předpokladu, že v proměnné enter je hodnota true a dojde ke stlačení tlačítka „F“, tedy ke změně hodnoty v proměnné open. Dojde k otočení modelu z výchozích souřadnic uložených v proměnné defaultRot do souřadnic uložených v openRot nebo opačně.

```
10.     function Update (){
11.     if(open){
12.     transform.localEulerAngles =
        Vector3.Slerp(transform.localEulerAngles,
        openRot, Time.deltaTime * speed);
13.     }else{
14.     transform.localEulerAngles =
        Vector3.Slerp(transform.localEulerAngles,
        defaultRot, Time.deltaTime * speed);}
15.     if(Input.GetKeyDown("f") && enter){
        open = !open;}}
```

Pro zvýraznění interaktivních částí scény byla přidána funkce „OnGUI“. Účelem této funkce je zobrazit uživateli, že se vyskytuje v oblasti interaktivní komponenty ve scéně a navést ho jak tuto komponentu ovládat viz.Obrázek 43.

```
16.     function OnGUI(){
17.     if(enter){
18.     if(!open){
19.     GUI.Label(new Rect(Screen.width/2 - 75,
        Screen.height - 100, 160, 40), "Zmáčkni 'F' pro
        otevření");
20.     }else{GUI.Label(new Rect(Screen.width/2 - 75,
        Screen.height - 100, 200, 40), "Zmáčkni 'F' pro
        zavření");}}}
```

Obrázek 43: Zvýraznění interaktivní komponenty

Dále se ve skriptu vyskytuje funkce „OnTriggerEnter“. Tato funkce detekuje vstup avatara do kolizní zóny.

```
21.     function OnTriggerEnter (other : Collider){
22.     if (other.gameObject.tag == "Player") {
23.     enter = true;}}
```

Pro detekci opuštění kolizní zóny byla přidána funkce „OnTriggerExit“

```
24.     function OnTriggerExit (other : Collider){
25.     if (other.gameObject.tag == "Player") {
26.     enter = false;
```

Tento skript byl použit na všechna okna vyskytující se v celém areálu TUL. Dále byl též použit na většinu dveří. Na zbylých dveřích byl použit skript „samo.js“. Tento skript je modifikována verze výše uvedeného skriptu, u kterého byla upravena metoda „Update“. Tato metoda již neočekává stlačení tlačítka ke spuštění otevírání a zavírání dveří.

Můžeme tedy říct, že využívání skriptů v rámci Unity je poměrně snadná činnost, která dokáže velmi výrazně ovlivnit funkčnost aplikace a též zjednodušit práci při vývoji

4.8 Nastavení osvětlení

Pro docílení lepší reálnosti výsledné aplikace bylo potřeba do scény vložit světla. Jak již bylo řečeno v kapitole 3.3.1 v rámci prostředí existují tři typy světel. Pro vytvoření centrálního světla pro celou scénu bylo využito Directional light . U tohoto světla byla nastavena intenzita odpovídající dennímu světlu a též byly nastaveny měkké stíny, které toto světlo vytváří viz. Obrázek 44.

Obrázek 44: Stín stromu

Aby toto světlo maximálně vzbuzovalo pocit, že se jedná o slunce muselo být tomuto světlu přiřazena záře („Flare“). Na Obrázek 46 je vidět, jak vypadá světlo bez přiřazení záře v porovnání se světlem, kterému záře přidělená byla viz. Obrázek 45.

Obrázek 46: Obrázek bez flare

Obrázek 45: obrázek s flare

4.8.1 Stíny ve výsledné aplikaci

Při nastavování stínů nevznikl žádný vážný problém. V rámci prostředí byly stíny vytvořeny reálně. Toto se však nedá říct o výsledné aplikaci, která při kompilaci do finální podoby tyto stíny ztratí. Při řešení této chyby došlo k zjištění, že tento

problém se vyskytuje u aplikací vytvářených pro webové prohlížeče. A tato chyba v prostředí bude řešena až v nové verzi vývojového prostředí Unity.

4.9 Zvýšení realističnosti

Pro zvýšení realističnosti je vhodné do scény vložit vegetaci viz. Obrázek 44. Více vegetace do této scény nebylo vkládáno záměrně. Jelikož tato aplikace je již dostatečně náročná a přidáním dalších prvků jako jsou stromy, keře a tráva by se stala maximálně nevyužitelnou.

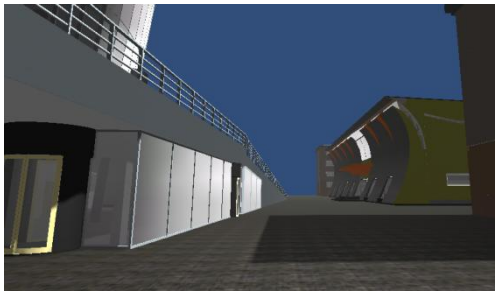
4.9.1 Využití tranzitivních textur v praxi

Jak již bylo zmíněno v kapitole zabývající se texturami v běžné praxi se využívají tranzitivní textury. To z důvodu vytváření snadnějších polygonových sítí,které nejsou tak náročné na zpracování. Nanesením tranzitivní textury na geometrická primitiva lze dosáhnout stejných efektů, jako kdyby model byl řádně vymodelován a následně mu byla přidělena odpovídající textura. Nevýhodou této metody je fakt, že renderovací systém prostředí Unity nedokáže rozeznat, že se jedná o tranzitivní texturu. Osvícený objekt pak vrhá stín přesně podle své polygonové sítě což nevzbuzuje realistický dojem viz. Obrázek 47.

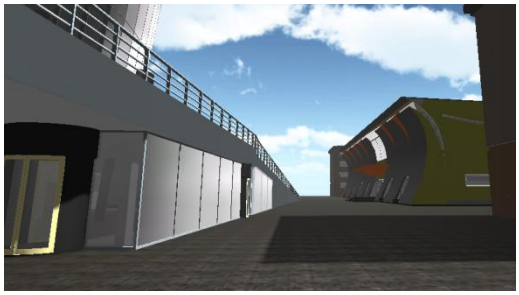
Obrázek 47: Nedokonalý stín stromu

4.10 Nastavení pozadí

V rámci zvyšování grafické kvality výsledné aplikace je vhodné nastavit pozadí. Z předdefinovaných pozadí si můžeme vybrat z několika typů, které vzbuzují lepší vizuální prožitek než obyčejné modré pozadí bez textury.



Obrázek 49: Scéna bez pozadí



Obrázek 48: Scéna s pozadím

4.11 Náročnost aplikace

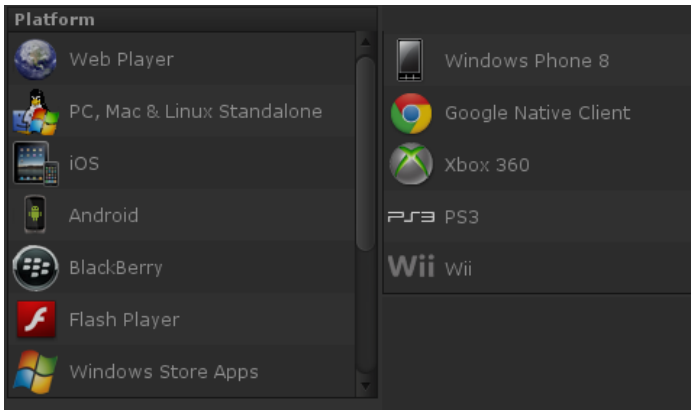
Vytvořená aplikace neběží úplně plynule. Byla testovaná na pěti počítačích různých výkonů. Výsledky FPS (Frames Per Second) testů byly téměř shodné na všech stanicích. Z toho lze usoudit, že běh aplikace, který není plynulý, není způsobený výkonem počítače, nýbrž samotnou aplikací. Při bližším zkoumání se zjistilo, že neplynulý běh způsobují neoptimalizované sítě jednotlivých modelů budov. Informace o polygonových sítích viz. Tabulka 2

Tabulka 2 Přehled náročnosti sítí

Model	Vertices	Faces	Size (Kb)	Počet objektů
Budova A	8176	13372	890	7
Budova B	16553	28014	967	2
Budova C	17254	32067	2519	10
Budova E	214762	427947	18939	100
Budova IC	18224	21505	15847	97
Budova F	179267	341415	1891	53
Model terénu a přilehlých budov	26920	33322	1687	17
CELKEM	481156	897642	42740	286

4.12 Vytvoření finální verze

Jak již bylo zmíněno aplikace vytvářené v Unity jsou multiplatformní. Prostředí Unity nabízí export do formátů spustitelných na herních konzolích (PlayStation 3, Wii, Xbox 360), mobilních zařízeních (iOS, Android, Blackberry, Windows Phone 8) nebo ve webových prohlížečích (web player, flash player, google nativ klient) viz. Obrázek 50.



Obrázek 50: Podporované platformy

Cílovou platformou pro tuto bakalářskou práci jsou však pouze webové prohlížeče. Pro spuštění aplikace pomocí Web Playeru musí být nainstalován zásuvný modul do webového prohlížeče. Při prvním startu aplikace je uživatel vyzván, aby si tento zásuvný modul stáhnul. Další možností jak pustit tuto aplikaci je přes Flash Player. Ten musí být též součástí webového prohlížeče.

Při testování výsledných aplikací se ukázalo, že nezáleží na cílovém formátu pro webové aplikace, jelikož zobrazení výsledných aplikací je stejně i co se týče FPS.

5 Závěr

Cílem této bakalářské práce bylo vytvoření interaktivní procházky areálem Technické univerzity v Liberci. Tato aplikace musela být vytvářena tak, aby byla zobrazitelná přes webový prohlížeč. Bylo tedy nutné se nejdříve seznámit s problematikou tvorby her pro webové prohlížeče. K tvorbě této bakalářské práce byl doporučen herní engine Unity 3D, který umožňuje výslednou aplikaci spustit přes webový prohlížeč. Pro vytvoření této aplikace byly použity 3D modely budov, které vytvořili studenti druhých ročníku bakalářského studia fakulty mechatroniky, informatiky a mezioborových studii Technické univerzity v Liberci. Tyto modely velmi výrazně ovlivnily funkčnost výsledné aplikace. To z důvodu, že tyto vytvořené modely byly příliš náročné pro toto využití. Jejich optimalizace by zabrala více času než samotné vytvoření těchto modelů od samého začátku. Náročnost modelu lze vidět v Tabulka 2. Z důvodu náročnosti těchto modelu nebyla do scény zakomponovaná vegetace, jelikož použití vegetace by zvýšilo náročnost aplikace.

V rámci porovnávání dostupných herních enginu lze Unity zařadit mezi lepší z nich. A to nejen díky výsledným aplikacím, které jsou multiplatformní. Příkladem muže být velikost výsledných aplikací. Pro příklad tedy porovnejme velikosti výsledných souborů z herního enginu Unity a herního enginu CryEngine. V rámci bakalářské práce s tématem „Fotorealisticka simulace areálu Husova v reálném čase“ vznikla aplikace, při které byl využit CryEngine. Výsledná aplikace byla co se týče grafického zpracování byla téměř dokonalá, ale využití této aplikace je téměř nemožné. Jelikož pro spuštění této aplikace je zapotřebí velice výkonný počítač. Tato aplikace zabírala cca 17 Gb místa na disku. Pro porovnání výsledné velikostí aplikace vytvořené v Unity zabírá pouze 50 Mb. Nevýhodou je však fakt, že aplikace vytvořená v Unity není grafickým zpracování srovnatelná s aplikací vytvořenou v CryEnginu. Výhodou však je, že aplikace, které jsou vytvořené v Unity lze snadno prezentovat skrze webové prohlížeče a na počítačích, které nemusejí být zdaleka tak výkonné jako v případě aplikace běžící na CryEnginu.

Při plnění cílu této bakalářské práce byly získány užitečné zkušenosti s nástroji pro tvorbu 3D modelů (Autodesk 3D studio Max), dále s nástroji pro úpravu fotek (GIMP) a neposledně i herním enginem Unity. Při plnění cílů této práce bylo též přihlíženo k rozšiřitelnosti výsledné aplikace. Mezi základní možné rozšíření, které by bylo vhodné do aplikace dodat, jsou nově postavené budovy v areálu TUL. Jelikož tento

model areálu byl aktuální v roce 2011. Nejsou tedy součástí areálu nově postavené budovy. Dalším vhodným rozšířením může být vložení optimalizovaných modelů jednotlivých budov. Toto rozšíření by mělo mít kladný dopad na náročnost samotné aplikace. Velice dobrým rozšířením této aplikace by bylo její předělání do formátů, které jsou zobrazitelné na mobilním zařízení. Následně pak tuto aplikaci rozšířit o navigaci po areálu TUL.

Použitá literatura

[1] KŘÍŽ, Jan. *Mistrovství v 3ds Max: [kompletní průvodce profesionálního grafika]*. Vyd. 1. Brno: Computer Press, 2010, 1151 s. ISBN 978-80-251-2464-

[2] -GOLDSTONE, Will. *Unity game development essentials*. Birmingham, U.K.: Packt Pub., 2009, vii, 298 p. From technologies to solutions. ISBN 978-1847198181.

[3] THORN, Alan. *Unity 4 fundamentals: making games with Unity*. First edition. Burlington, MA: Focal Press, 2013, pages cm. ISBN 9780415823838.

[4] *Learning C# by Developing Games with Unity 3D*. Birmingham: Packt Publishing, 2013. ISBN 978-184-9696-586.

[5] HARPER, Jeffrey Alan. *Mastering autodesk 3ds max 2013*. 1st ed. Indianapolis, IN: WIley Pub. Inc., 2012, p. cm. ISBN 11-181-2971-7.

[6] HOUŽVIČKA, Vojtěch. *Interaktivní 3D model budovy F*. Liberec, 2012.Bakalářský projekt. Technická univerzita v Liberci.

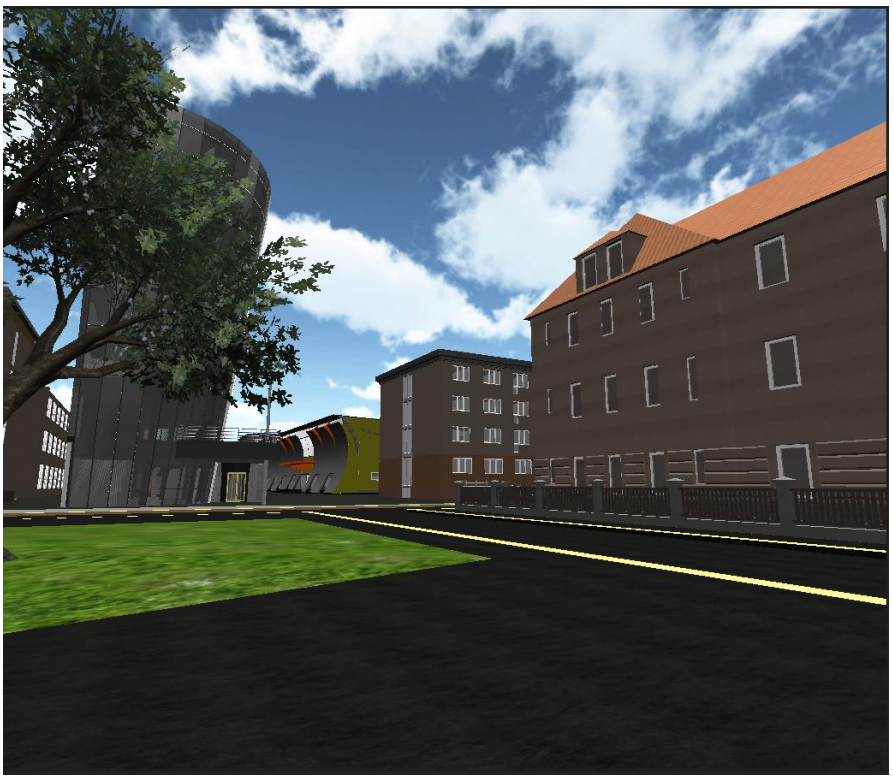
[7] JINDRÁK, Marek. *Interaktivní 3D model budovy E*. Liberec, 2012. Bakalářský projekt. Technická univerzita v Liberci.

[8] MARHOUL, Vojtěch. *Interaktivní 3D model budovy C*. Liberec, 2012.Bakalářský projekt. Technická univerzita v Liberci.

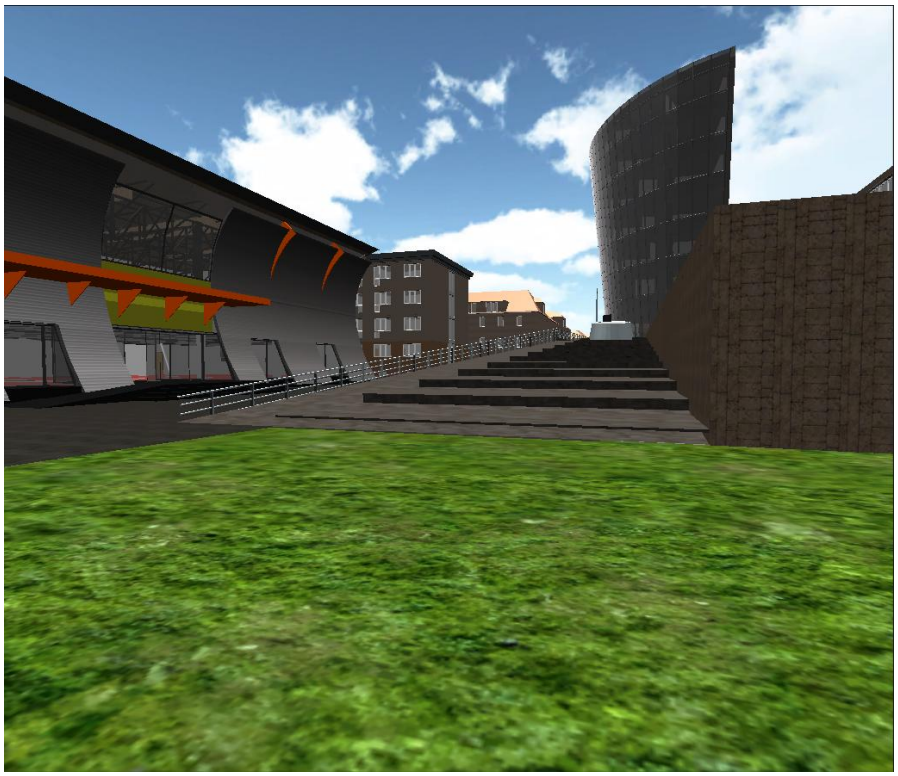
[9] PODLIPNÝ, Jiří. *Model areálu technické univerzity v Liberci*. Liberec, 2011.Bakalářský projekt. Technická univerzita v Liberci.

[10] PODLIPNÝ, Jiří. *Fotorealistická simulace areálu Husova v reálném čase*. Liberec, 2013.Bakalářský práce. Technická univerzita v Liberci.

Příloha A – Obrázky hotové aplikace



Obrázek přílohy 2: Pohled na budovu Ic a F



Obrázek přílohy 1: Pohled na budovu F a Ic

Obrázek přílohy 4: Pohled ze střechy budovy E

Obrázek přílohy 3: Interiér posluchárny E9