

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHOLOGII

## *BAKALÁŘSKÁ PRÁCE*



## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením MgA. Roberta Chudého.

Další cenné informace mi poskytli:

- Milan Špaček
- Jiří Hnídek

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal, stejně jako osoby, od nichž jsem získal potřebné informace.

v Brně dne 21. dubna 2005

---

### **Abstrakt:**

Tato bakalářská práce se zabývá tvorbou interaktivní aplikace pro real-time průchod virtuální budovou. K jejímu vypracování byl zvolen open source 3D modelační program Blender, převážně jeho součást GameEngine pro tvorbu real-time aplikací a her. Součástí práce je též tvorba univerzálního engine pro snadný import modelu, vytvořeného v jiné aplikaci, do Blenderu, nastavení vlastností a parametrů modelu a možnost virtuálního průchodu v něm. Tvorba tohoto engine spočívá v napsání skriptu spouštěného přímo v prostředí Blenderu, který vytvoří potřebné grafické uživatelské rozhraní a provádě ho s potřebnými funkcemi.

### **Klíčová slova:**

počítačová grafika, interaktivní aplikace, real-time zobrazení, virtuální realita, průchod virtuální budovou, Blender, Blender GameEngine, stereo-zobrazení, skript, Python, GUI, grafické uživatelské rozhraní, 3D model

**Abstract:**

Abstract in english :-)

**Key words:**

computer graphics, interactive application, real-time displaying, virtual reality, virtual building walk-through, Blender, Blender GameEngine, stereo-visualisation, script, Python, GUI, graphic user interface, 3D model

# Obsah

<b>1 Úvod</b>	<b>3</b>
1.1 Stručný přehled práce . . . . .	3
<b>2 Požadavky a současný stav</b>	<b>5</b>
2.1 Současný stav oblasti real-time vizualizace . . . . .	5
<b>3 Program Blender</b>	<b>7</b>
3.1 Jak ho získat . . . . .	7
3.1.1 Stabilní verze . . . . .	7
3.1.2 Vývojové a ostatní verze . . . . .	7
3.2 Součásti programu Blender . . . . .	8
<b>4 Tvorba pomocí Blender GameEngine</b>	<b>12</b>
4.1 Logic Bricks . . . . .	12
4.1.1 Vlastnosti a proměnné objektu . . . . .	12
4.1.2 Senzory (Sensors) . . . . .	13
4.1.3 Ovladače (Controllers) . . . . .	13
4.1.4 Aktivní prvky (Actuators) . . . . .	13
4.2 Možnosti skriptování . . . . .	13
4.3 Nástroj GameEngine pro řešení kolizí . . . . .	14
4.4 Nástroje GameEngine pro řešení pohybu . . . . .	14
4.5 Shrnutí tvorby v Blender GameEngine . . . . .	15
<b>5 Tvorba Blender Walkthrough Engine</b>	<b>16</b>
5.1 Příprava a návrh . . . . .	16
5.2 Implementace interaktivní části . . . . .	17
5.2.1 Základ šablony . . . . .	17
5.2.2 Úprava interaktivní šablony . . . . .	18
5.2.3 Možnost použití jiné šablony v tomto projektu . . . . .	18
5.3 Podmínky a doporučení pro importovaný model . . . . .	20
5.4 Tvorba engine . . . . .	21
5.4.1 Spuštění skriptu . . . . .	21
5.4.2 Import modelu a nastavení kalibračního objektu . . . . .	22
5.4.3 Nastavení naimportovaných objektů . . . . .	24
5.4.4 Výpočet radiozity . . . . .	24
5.4.5 Nastavení real-time a uložení projektu . . . . .	24
5.5 Zjištěné problémy a nedostatky . . . . .	24

<b>6</b>	<b>Možnosti prezentace vytvořených dat</b>	<b>28</b>
6.1	Zobrazování pomocí webového prohlížeče . . . . .	28
6.1.1	Nalezené problémy se zobrazováním . . . . .	29
6.2	Přehrávání pomocí BlenderPlayer . . . . .	30
6.3	Export do spustitelné aplikace . . . . .	30
6.4	Potíže při přehrávání real-time obsahu . . . . .	31
6.5	Rozšiřující možnosti . . . . .	31
<b>7</b>	<b>Závěr</b>	<b>32</b>
7.1	Návrhy pro pokračování práce . . . . .	32
<b>8</b>	<b>Slovníček pojmů</b>	<b>34</b>
<b>9</b>	<b>Přílohy</b>	<b>36</b>
9.1	Návod k používání Blender Walkthrough Engine . . . . .	36

# Kapitola 1

## Úvod

Počítačová grafika je již od počátků prvních grafických adaptérů oborem, který se neustále velice rychle zdokonaluje a rozvíjí, a silně tak ovlivňuje technologický pokrok v oblasti počítačů. Posledních pár let se objevuje stále rostoucí zájem o virtuální realitu, real-time zobrazení a 3D stereo zobrazení, potažmo projekci, ať již formou prezentací, zábavy nebo stále oblíbenějších 3D animovaných filmů.

Vytvořit 3D model lze mnoha způsoby, namátkou se dá třeba využít různých 3D modelačních programů, naprogramovat jej pomocí některé z mnoha grafických knihoven, z nichž jsou některé prohlášeny za standard (OpenGL, DirectX, SDL a mnohé další) nebo vytvořit iluzi 3D prostoru různými optickými efekty (blending<sup>1</sup>, stereo-projekce<sup>2</sup> a další).

V současné situaci, kdy je dostupná spousta 3D modelačních programů široké veřejnosti, je vytvoření modelu otázkou pár okamžiků. Ale možností převedení těchto modelů do interaktivního virtuálního zobrazení již zdaleka není tolik.

Za interaktivní aplikaci lze považovat takovou, kde má uživatel možnost volného pohybu na základě svých požadavků. Výborným příkladem interaktivní aplikace jsou např. počítačové hry, převážně 3D akční hry z prvního pohledu (pohledu ovládané postavy). Některé programy sice dokáží vygenerovat animaci průletu modelem podle různých trajektorií, ale takovou aplikaci nelze nazvat interaktivní, protože je uživatel omezen pouze na sledování předdefinované animace a nemá možnost její běh ani děj nijak ovlivnit.

Existuje sice spousta prostředí, grafických engine nebo konverzních nástrojů, které toto dokáží, ale většina z nich neobsahuje buď všechny potřebné funkce, jednoduchost a uživatelskou přítulnost ovládání nebo alespoň přijatelnou cenu pro obvyčejného uživatele. Navíc některé neumožňují přímou tvorbu interaktivní aplikace, ale pouze výše zmíněné předdefinované animace. Jiné jsou pro laiky takřka nedostupné kvůli nutnosti vše naprogramovat.

### 1.1 Stručný přehled práce

První část pojednává o požadavcích a úkolu této práce, všeobecných problémech, současném stavu v několika zvolených projektech věnujících se virtuální nebo real-time grafice a aktuálních možnostech v této oblasti.

V další kapitole se seznámíme s programem Blender jako celkem, nahlédneme do jeho historie, zjistíme jeho vlastnosti, schopnosti a možnosti a povíme si, jak ho získat pro libovolnou platformu.

---

<sup>1</sup>Používá se ve spoustě grafických efektů, převážně však ke zprůhlednění objektů. Při dobrém zvládnutí této metody lze nasimulovat trojrozměrný prostor. Více o blendingu např. na URL <http://www.root.cz/clanky/opengl-28-blending/>

<sup>2</sup>Technologie používaná různými kiny a zábavními komplexy – v české republice se s ní lze setkat např. v kině OskarIMAX

Následující část obsahuje popis tvorby v součásti Blender GameEngine, a to jak prostřednictvím rozhraní samotného Blenderu, tak pomocí skriptovacího jazyka Python, se kterým je prostředí a API Blenderu kompletně provázáno a který představuje mocný prostředek při tvorbě real-time obsahu. Jsou zde též zhodnoceny možnosti GameEngine, prostředky pro řešení základních problémů, např. kolizí nebo interaktivního pohybu, a potenciál Blenderu při real-time zobrazování.

Další část popisuje proces tvorby uživatelského rozhraní univerzálního engine. Jedná se o sadu skriptů, které jsou kompletně vytvořeny pomocí prostředků nabízených python API rozhraním Blenderu. Povíme si o implementaci a funkcích jednotlivých částí rozhraní, jeho možnostech jako celku a některých implementačních detailech. Jsou zde též vyjmenovány různé problémy, které byly během práce objeveny a řešeny.

V poslední kapitole se dozvíme o možnostech prezentace výsledků prostřednictvím webového rozhraní či samostatných spustitelných binárních aplikací a budou zde zmíněna různá rozšíření, jejichž podrobnější popis by již byl mimo rozsah této práce nebo ještě nejsou plně implementována a mohla by se objevit v některé z příštích verzí. Též jsou zde zhodnoceny výsledky práce a návrhy pro její pokračování nebo rozšíření.



## Kapitola 2

# Požadavky a současný stav

Hlavním požadavkem je vytvořit funkční, interaktivní real-time aplikaci pro průchod virtuálním modelem budovy. Tato tvorba by měla probíhat co nejintuitivněji pro obyčejného uživatele bez znalosti jakéhokoliv programovacího jazyka a nejlépe ve volně dosažitelné aplikaci. Úkolem tedy bylo najít freeware, open source nebo jiný snadno dostupný program, mezi jehož portfolio funkcí spadá import modelu budovy, provedení určitých akcí s tímto modelem a umožnění uživateli interaktivního průchodu v reálném čase.

Cílem této práce je seznámení s možnostmi interaktivního virtuálního průchodu modely budov, a jedním ze zástupců této oblasti, 3D grafickým programem Blender, který spojuje modelační software, herní engine pro tvorbu real-time aplikací s podporou zvuku, jednoduchý video editor a spoustu dalších užitečných modulů do jedné aplikace. Úkolem práce je zjistit potenciál Blender GameEngine v real-time oblasti, popsat různá úskalí tvorby v tomto modulu a dále vytvořit demonstrační aplikaci pro virtuální průchod naimportovaným modelem budovy. Dále je potřeba naimplementovat uživatelské rozhraní pro snadný import a nastavení vlastností. Též budou dodána demonstrační data.

Práce si klade za cíl přiblížení virtuálního zobrazení a software Blender širšímu okruhu uživatelů a jejich seznámení s možností tvorby 3D her a real-time prezentací v software, který je zdarma, multiplatformní, aktivně vyvíjen a se silnou uživatelskou a vývojářskou komunitou a jehož instalační balíček nedosahuje ani 10MB velikosti.

Výsledkem této práce bude již zmíněný engine pro import modelu uloženého v některém z podporovaných formátů do Blenderu, nastavení jeho parametrů a interaktivní průchod tímto modelem. Tento engine byl pojmenován jako **Blender Walkthrough Engine**.

Dalším výsledným produktem bude tato technická zpráva, která může posloužit jako malá dokumentace k Blender GameEngine i jako uživatelská příručka k vytvořenému engine.

### 2.1 Současný stav oblasti real-time vizualizace

Možnosti současných počítačů jsou pro real-time grafiku a virtuální realitu více než dostačující. Na trhu je spousta aplikací, které jsou zaměřené pouze na tuto oblast, jiné zase kombinují možnost tvorby ve 2D a 3D prostoru a následnou konverzi těchto výsledků do virtuální reality nebo real-time zobrazení. Nicméně tato oblast se stále bouřlivě vyvíjí, protože ještě nebyla zbavena všech "dětských nemocí".

Vznikají nové technologie více přístupné obyčejným lidem, nové produkty jsou uživatelsky přívětivější a mnohem menší a lehčí. Stavějí se komplexy, které se specializují i na stereo-projekci

(např. multiplexové kino OskarIMAX<sup>1</sup>), kdy se trojrozměrný film na filmové plátno promítá prostřednictvím dvou polarizovaných čoček, které na plátno odděleně promítají filmové pásy pro pravé a levé oko. Diváci mají na sobě polarizované brýle, které jsou vybaveny čočkami stejného polárního zasměrování pro pravé a levé oko jako projektor. Pravé a levé oko tak dostávají pouze obraz pro ně určený a uvnitř našeho mozku se pak skládá iluze trojrozměrné reality.

Existuje mnoho programů, modulů a rozličných grafických engine, v nichž lze dosáhnout interaktivního real-time průchodu v požadovaném modelu. Některé z nich jsou ale určeny pouze pro uživatele se znalostí některého programovacího jazyka, jiné zase finančně nedostupné. Zde je stručně zmíněno několik aplikací zastupujících jednotlivé skupiny.

- **Ogre3D** – zástupce programovací skupiny. Obsahuje širokou škálu funkcí, v nichž nechybí ani částicové efekty, práce s texturami a možnost importu modelu z externího souboru, je multiplatformní, zdarma a stále vyvíjen. Mezi jeho největší "nevýhody" patří již zmíněná nutnost vše naprogramovat, jedná se tedy o 3D herní engine. Jeho domovská stránka je k nalezení na <sup>2</sup> a program lze stáhnout ve formě SDK do nejrozšířenějších vývojových prostředí a různých zásuvných modulů zvyšujících jeho schopnosti a funkčnost.
- **TurboSquid Rtre** – interaktivní renderer, dodávaný jako plug-in do programů 3ds Max a Autodesk VIZ. Umožňuje vytvářet real-time fotorealistické 3D prezentace formou virtuálních procházek včetně zvuku. Modul podporuje real-time rendering statických 3D scén s rozlišením až 65 000 x 65 000 bodů, stejně jako nefotorealistické (umělecké) styly renderingu, napodobující např. malbu nebo kresbu. Jeho cena je ale příliš vysoká i pro většinu cílových firem. V USA lze pořídit za 1295\$. Tento modul je navíc použitelný pouze na platformách podporovaných výše zmíněnými aplikacemi. Více informací o projektu lze nalézt na adrese<sup>3</sup>.
- **FloorPlan 3D** – zástupce komplexnějších řešení. Jedná se o program typu CAD, zaměřený spíše na stavební použití, obsahující zabudovaný modul pro okamžitou interaktivní real-time procházku vytvořenou budovou. Letmé seznámení s tímto programem a jeho umístění v testu podobných aplikací se můžete dočíst v [3, str. 132]. Cena programu je přijatelná, i s českou lokalizací stojí 1999 Kč. Bohužel, ani tento CAD systém nezvládá vše, potíže mu činí např. složité a rozsáhlejší modely, navíc je určen pouze pro platformu Windows. Také virtuální průchod modelem je umožněn pouze při spuštění aplikaci. Domovská stránka programu se nachází na webu výrobce<sup>4</sup>.

---

<sup>1</sup><http://www.oskarimax.cz>

<sup>2</sup><http://www.ogre3d.org>

<sup>3</sup><http://www.turbosquid.com>

<sup>4</sup><http://www.imsisoft.com>

## Kapitola 3

# Program Blender

### 3.1 Jak ho získat

Vzhledem ke své open source politice je program Blender volně dostupný každému uživateli. Možností jeho získání je spousta, dokonce si uživatel může zvolit verzi, kterou chce používat. Může pracovat ve stabilní verzi, v některé ze starších verzí nebo sáhnout po některé z vývojových verzí, které většinou obsahuje nové funkce, ale nemusí být úplně stabilní.

Dále existují různé projekty vznikající ze stabilních či vývojových verzí Blenderu a implementující nové užitečné funkce. Tyto verze však bývají zpětně kompatibilní, takže uživatel nepřijde o možnost přenosu souborů mezi různými verzemi.

Všechny níže zmíněné verze (kromě Instinctive Blender) se kromě dále uvedených oficiálních zdrojů nalézají ke stažení též na adrese <http://blendertestbuilds.kidb.de/>.

#### 3.1.1 Stabilní verze

Stabilní verzi Blenderu lze stáhnout z mnoha serverů po celém internetu, nejlépe však přímo z jeho domovské stránky <http://www.blender3d.org> pomocí odkazu *Download*. Kompletní přehled všech verzí lze nalézt na adrese <http://download.blender.org/release/>. Zde je možné získat aktuální i všechny starší stabilní verze, stejně jako webový modul či externí raytracer yafray.

#### 3.1.2 Vývojové a ostatní verze

Vzhledem k popularitě této aplikace a poměrně dlouhému vývojovému cyklu lze používat i některou z vývojových verzí, popř. verzi, která vznikla ze stabilní, ale je vyvíjena jinou organizací či firmou. Tyto verze obvykle obsahují některé přídavné funkce, které se ale časem synchronizují s hlavní větví Blenderu, takže o ně nebudou ochuzeni ani uživatelé používající stabilní verzi.

- **Vývojové verze**

V současné době existují dvě vývojové větve Blenderu. Tzv. bf-blender, což je větev, z níž později vznikne stabilní verze a Tuhopuu, v současnosti s pořadovým číslem 3. Do ní jsou implementovány nové a neozkoušené funkce. Pokud se projeví jako užitečné a stabilní, jsou přesunuty do bf-větev.

Obě verze lze získat ve vývojářském diskuzním fóru na adrese <http://www.blender.org> a pomocí menu zvolit *Developers forum*. Zde se potom v sekci *Testing builds* nacházejí aktuální buildy obou verzí pro nejpoužívanější platformy (Windows, Linux, Mac OsX).

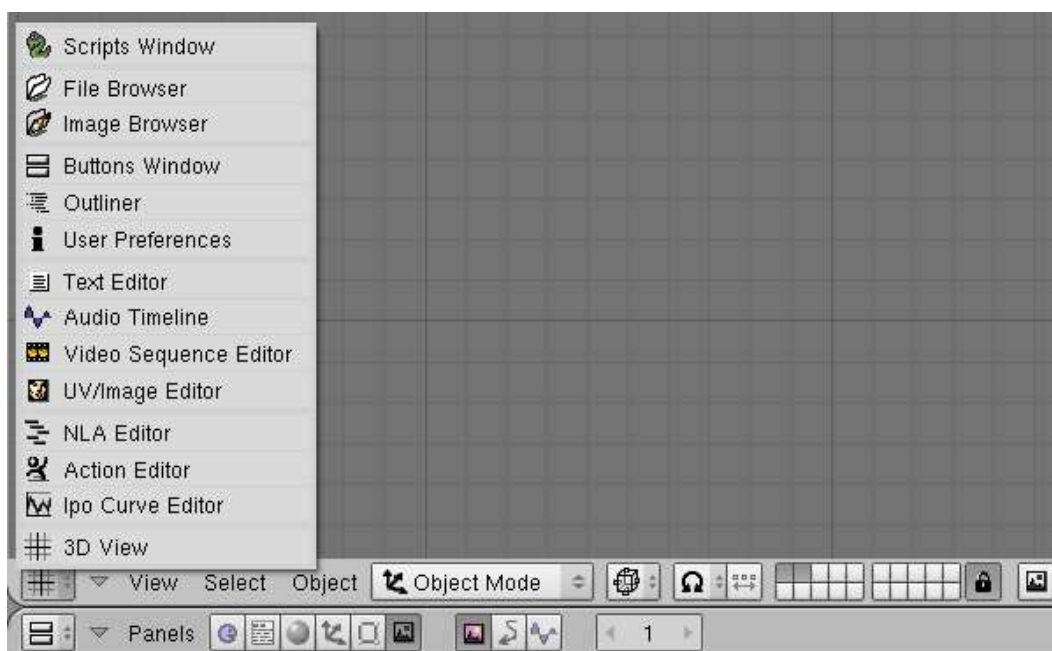
- **Ostatní verze Blenderu**

Asi nejnámější odnoží Blenderu je jeho verze od německé firmy Instinctive <sup>1</sup>, pojmenovaná Instinctive Blender. Jsou v ní implementovány některé funkce navíc (např. cloth simulator, vylepšený LOD – level of detail – a simulace vln) a snaží se skloubit nějnovější funkce a vlastnosti poslední stabilní a vývojové verze s grafickým rozhraním starší verze Blender Publisher, na nějž je spousta uživatelů stále navyklá a nepřejde kvůli němu na novější verzi.

Aktuální i všechny předchozí verze jsou volně ke stažení na této adrese <sup>2</sup>. Zde se také nachází popis všech vylepšení a ukázkové soubory demonstrující největší vylepšení.

## 3.2 Součásti programu Blender

Program Blender je složen z více součástí, které všechny dohromady tvoří kompletní řešení pro tvorbu 3D modelů, animací, real-time obsahu, zahrnují jednoduchou práci se střihem zvuku a videa a mnoho dalších akcí. Menu obsahující všechny moduly, nejdůležitější součásti 3D okna a zbytek modulů, které jsou součástí Buttons Window, jsou zobrazeny na obrázku 3.1. Ty si nyní stručně popíšeme.



Obrázek 3.1: Součásti Blenderu

- **Scripts Window**

Modul pro práci se skripty. Ty jsou rozděleny do jednotlivých sekcí a lze je odtud jednoduše vyvolat. Po přidání nového skriptu je potřeba zavolat *Update Menus* a ten bude poté zahrnut do příslušného submenu.

- **File Browser**

Vyvolá dialog pro otevření souboru. V něm lze soubory volit stiskem prostředního tlačítka myši,

<sup>1</sup><http://www.instinctive.de>

<sup>2</sup><http://blender.instinctive.de/downloads/release/>

popř. výběrem levého tlačítka myši a stiskem klávesy Enter. Tato volba by však měla být volána pouze pomocí skriptu, popř. tlačítka rozhraní Blenderu, jinak není funkční.

- **Image Browser**

Stejně jako předchozí možnost, pouze slouží k prohlížení a nahrávání obrázků.

- **Buttons Window**

Jedna ze stěžejních částí Blenderu. Obsahuje funkční tlačítka, nastavení textur, světel, materiálů, vlastností objektu, výpočet radiozity a mnoho dalších. Pomocí této části lze ovládat většinu modelačních funkcí. Při implicitním sezení se nachází ve spodní části. Jeho podmoduly jsou popsány později.

- **Outliner**

Strom objektů. Lze zobrazit buď starší verze, kdy jsou jednotlivé objekty a jejich součásti zobrazeny jako jednotlivé bloky propojené logickými spoji nebo novější verze, ve které jsou již seřazeny do přehledného stromu. Lze z něj zjistit parentace objektů, jejich materiály, IPO křivky a ostatní vlastnosti.

- **User Preferences**

Hlavní menu pomocí něhož lze ovládat práci s obsahem a aktuálním sezením. Dále je zde obsaženo nastavení lokalizace, fontů, vzhledu a ostatních uživatelských nastavení. Při implicitním sezení se nachází ve formě lišty úplně nahoře.

- **Text Editor**

Textový editor, který je přímo součástí Blenderu. Slouží ke psaní textových poznámek a skriptů, které lze přímo v editoru spouštět pomocí položky menu *File* → *Run Python Script*. Ve vývojové verzi Tuhopuu obsahuje editor i jednoduché zvýrazňování syntaxe. Samozřejmě lze při psaní skriptů využívat funkcí Blender Python API [5].

- **Audio Timeline**

Jednoduchý lineární audio editor. Umožňuje načtení zvuku, jeho jednoduchou editaci, volbu sekvence a její přiřazení určité části animace. Jeho další část se nalézá na liště modulu **Buttons Window**. Ve spojení s video editorem tvoří jednoduché, ale mocné stříhové studio, s jehož pomocí lze vytvářet vysoce kvalitní 3D animace.

- **Video Sequence Editor**

Video editor, sloužící k tvorbě a úpravě animací. Mezi jeho schopnosti patří práce se sledem obrázků, videí, lze do něj vložit i scénu ze sezení Blenderu, použít různé předdefinované animace (zprůhlednění, prolnutí a mnoho dalších).

- **UV/Image Editor**

Editor obrázků, s jehož pomocí lze otexturovat objekty v tzv. Face Mode, kdy se na jednotlivé strany (faces) objektu mapuje požadovaná část textury. Tento postup je znám jako UV Mapping. Lze použít i jednoduchý UV Unwrapper, který se pokusí vyhodnotit jednotlivé strany objektu a přiřadit jim příslušné části textury automaticky.

- **NLA Editor**

Editor pro tvorbu nelineárních animací. Lze v něm pomocí bloků prolínat různé animace a efekty. Používá se například pro tvorbu plynulých akcí, kde se musí zesynchronizovat IPO křivky, pohyb a střih.

- **Action Editor**

Modul používaný pro tvorbu pohybu, kdy se pomocí kosterního systému (armature) vytváří pohyb po framech a zbytek je dopočítáván. Pokud je v pojmenování kostí řád, Action Editor dokáže zrcadlit pohyb pro levou i pravou stranu.

- **IPO Curve Editor**

Editor IPO křivek, pomocí nichž se dá specifikovat pozice, rotace a nastavení jednotlivých barev objektů. Dále se používají k nastavení vlastností materiálů, textur, vertexů prolínání mezi snímky animace atd. IPO křivky jsou jedním z nejkompaktnějších a nejmocnějších nástrojů.

- **3D View**

Hlavní okno pro modelaci. Obsahuje mřížku, 3D kurzor a jsou do něj "vykreslovány" všechny modelované objekty. Lze různě natáčet, měnit pohledy, přibližovat a vzdalovat a měnit zobrazené informace.

Vedle rozvinutého menu se nachází dvě lišty. Horní náleží k panelu 3D View a spodní k Buttons Window. V liště 3D view se postupně nacházejí tato tlačítka:

- **Menu**

V menu se nalézají všechny funkce dostupné v příslušném panelu. Každý panel má své specifické menu.

- **Pracovní mód**

Tímto tlačítkem se přepíná pracovní mód. Lze pracovat v následujících režimech.

- *Object Mode* – práce s objekty.
- *Edit Mode* – práce s jednotlivými vertexy, hranami a stranami.
- *UV Face Select* – texturovací mód, ve kterém se vybírají jednotlivé faces a lze jim přiřadit potřebnou část textury.
- *Vertex Paint* – barvení vertexů. Ty lze obarvit různými barvami, které se poté interpolují na hranách a stranách.
- *Texture Paint* – barvení textur. Stejně jako v předchozím případě, pouze s tím rozdílem, že lze obarvovat textury.
- *Pose Mode* – zakostňovací mód. Objeví se v menu pouze v případě, že pracujeme s Armatures (kostní systém).

- **Mód zobrazení**

Zde lze zvolit způsob, jak se budou objekty ve 3D okně zobrazovat. Na výběr je mezi

- *Bounding Box* – objekty budou zobrazeny pouze jako jejich obálka, která celý objekt zahrnuje.
- *Wireframe* – objekty budou zobrazeny jako vertexy a strany, které spolu tvoří drátový model.
- *Solid* – z objektu jsou zobrazeny stěny, není však zahrnuto obarvení vertexů, stínování ani otexturování.
- *Shaded* – stejné jako *Solid*, pouze jsou zahrnuty i stíny.
- *Textured* – zobrazení stěn, zahrnující i otexturování a stínování.

- **Pivot rotace**

Následujícím tlačítkem je možno přepínat pivota, podle něhož budou aplikovány veškeré polohové a rotační operace a operace změny velikosti. Mezi volbami je střed obálky, střed objektu, 3D kurzor a střed shluku objektů. Tlačítko napravo slouží k přepínání, jestli se budou pohybovat celé objekty nebo pouze jejich středy.

- **Zobrazené vrstvy**

Stejně jako v každém kvalitním modelačním programu i v Blenderu lze pracovat ve vrstvách. Dvacet "čtverečků" představuje vrstvy a zobrazeny jsou ty, které jsou stisknuty.

- **Zamknutí kamery a vrstev**

Dalším tlačítkem lze zamknout kameru a vrstvy k aktuální scéně.

- **Rychlý render**

Poslední tlačítkem se vyrenderuje aktuální pohled. Nastavení renderu je stejné jako poslední použité.

Panel Buttons Window je rozdělen na více podpanelů (modulů). Ty lze přepínat pomocí lišty. Tlačítka vlevo slouží k výběru požadovaného podpanelu, tlačítka uprostřed upřesňují volbu a u každého podpanelu jsou jiná. Poslední tlačítko, obsahující číslici 1, zobrazuje aktuální frame. Buttons Window obsahuje následující podpanely v levé části tlačítek.

- **Logic (GameEngine)**

Modul, v němž je vypracován požadovaný engine, sloužící ke tvorbě real-time obsahu. Jeho podrobnější popis se nalézá v kapitole 4.

- **Scripts**

V tomto podpanelu lze nastavit skripty, které se budou spouštět po provedení požadované akce. Skripty lze specifikovat pro celou scénu, objekty, okolní svět a materiály.

- **Shading**

Podpanel obsahující nastavení materiálů, textur, výpočet radiozity a činnosti s ní spojené a též nastavení okolního světa. Zde se nachází jedna důležitá položka pro tvorbu real-time obsahu – gravitační zrychlení. Také je v něm obsaženo nastavení světel.

- **Object**

Nastavení objektu. Vlastnosti animace, vrstvy ve kterých se nachází, zobrazené informace ve 3D okně a též zde lze specifikovat různé efekty, např. vlny (waves) nebo částicové efekty (particles).

- **Editing**

Volby pro editační mód. Lze zde nastavit jméno objektu a příslušného meshe, přiřazené materiály, vytvářet skupiny vertexů pro tvorbu zakostnění a další potřebná nastavení meshe. Dále se zde po vstupu do editačního módu nacházejí různé nástroje pro tvorbu a deformaci meshe, jako Screw, Spin nebo Spin Dup, pomocí nichž lze vytvářet šroubovitě nebo jinak pravidelné objekty.

- **Scene**

Nastavení scény a renderu. Obsahuje možnosti ovlivnění výstupu, jako jméno výstupního souboru, použitý způsob renderingu a použitý renderovací engine a formát výstupního renderu. Též se zde nachází nastavení animace, její formát, popř. použitý kodek.

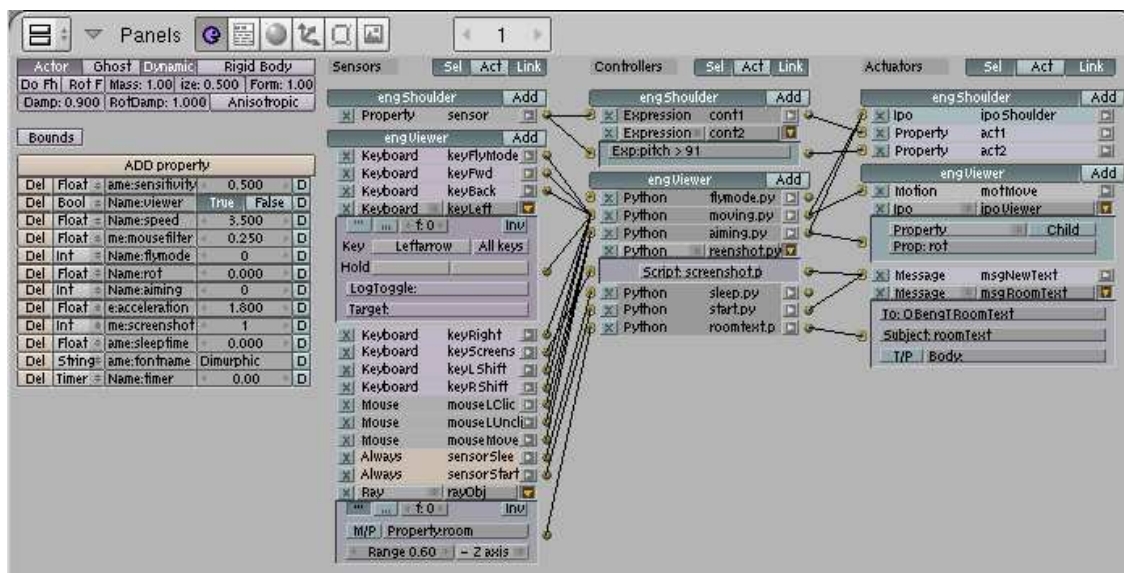
## Kapitola 4

# Tvorba pomocí Blender GameEngine

Blender GameEngine je modul pro tvorbu interaktivního real-time obsahu implementovaný přímo do prostředí Blenderu. Tvorba v něm se skládá ze dvou na sobě nezávislých částí, ty se ale dokonale doplňují.

### 4.1 Logic Bricks

Lze v něm pracovat pomocí uživatelského rozhraní Blenderu, kdy se veškerý real-time obsah, aktivní prvky a reakce na události dá nadefinovat pomocí tzv. Logic Bricks (logických bloků), jak je ukázáno na obrázku 4.1. Tento modul lze rozdělit na 4 hlavní části (sloupce), které si zde postupně popíšeme.



Obrázek 4.1: Logic Bricks

#### 4.1.1 Vlastnosti a proměnné objektu

Úplně vlevo nahoře se nachází blok ovládacích prvků, pomocí nichž lze nadefinovat vlastnosti objektu, jako možnosti interakce s okolím, jestli se pro něj mají zjišťovat kolize, případně zda je nehmotný



(tzv. Ghost mode), dále jeho odrazivost od stěn, podlahy, vektor tření a další nastavení týkající se pohybu a výpočtu kolizí.

Pod tímto blokem se nachází proměnné patřící k objektu. Ty mohou být 5 různých typů – časovač, číslo s plovoucí desetinnou čárkou, celé číslo, řetězec a logická hodnota – a lze je využít jak pro uchovávání dat během práce v Blenderu, tak v rámci real-time zobrazení. Proměnné lze vytvářet staticky přímo v prostředí nebo dynamicky pomocí skriptů, což je ale umožněno pouze skriptům běžícím mimo real-time režim – tam lze pracovat pouze s Python Game Logic API [6]. Proměnnými lze též určit identifikaci objektů a tu poté využívat pomocí senzorů pro řešení kolizí (viz sekce 4.3).

#### 4.1.2 Senzory (Sensors)

Tlačítka ve druhém sloupci se nazývají senzory a používají se k detekci událostí. Lze si vybrat ze spousty typů, od senzoru klávesnice, myši, obecného generátoru impulzů, radaru až např. po detektor vzdálenosti od objektu, kolize s objektem nebo detekčního paprsku vyslaného podél požadované osy.

Pomocí senzorů lze nadefinovat událost, na kterou se má reagovat a podrobně specifikovat její podmínky. Pokud se tato událost během real-time běhu vyskytne a jsou splněny všechny podmínky, senzor vyše pulz zvolené orientace (pozitivní, negativní), který může být vyslán pouze jednou nebo opakovaně v daném intervalu.

Po odchycení události se můžeme přesunout více vpravo ke specifikaci ovladače senzoru.

#### 4.1.3 Ovladače (Controllers)

Ovladače se používají k upřesnění, jakým způsobem mají být senzory aplikovány. Lze zvolit mezi logickým součtem (OR) a součinem (AND), výrazem (Expression) a skriptem v Pythonu (Python). Pomocí logických operátorů lze spojit více senzorů do jednoho celku a ten se potom vyvolá pouze při jejich odpovídající kombinaci. Výrazovým ovladačem lze naopak poslat impuls ze senzoru do aktivního prvku pouze tehdy, kdy daný výraz nabývá platnosti. A posledním ovladačem lze po obdržení impulsu od senzoru zavolat interpret jazyka Python na zvolený skript, který se poté kompletně vykoná.

Jestliže jsou splněny všechny podmínky ovladače, impuls se přesune do poslední části své logické "trasy", samotného aktivního prvku, který provede požadovanou akci. A to buď na základě pulzu patřičné orientace nebo pomocí API volání z ovládacího skriptu.

#### 4.1.4 Aktivní prvky (Actuators)

Aktivní prvek je zakončením logické trasy impulsu. Zde se specifikuje požadovaná akce, která bude při úspěšných předchozích podmínkách provedena. Lze opět zvolit mezi spoustou typů, od prvků pohybových, IPO volání, prvků ovládajících real-time sezení, objekt, scénu, posílání zpráv až po ty, týkající se práce se zvukem, CD-ROM.

Jestliže dojde impuls až do některého z aktivních prvků (nebo byl aktivován pomocí skriptu), dojde k vykonání požadované akce.

### 4.2 Možnosti skriptování

Další možností práce v GameEngine je využití interpretovaného jazyka Python a skriptů, které lze tvořit též jako nedílnou součást obsahu. Skripty lze psát přímo v textovém modulu Blenderu nebo jako externí, které je ale nutno přilinkovat.

Při psaní se dá využívat bohatého Blender GameEngine Python API, jehož dokumentace je volně ke stažení [7]. Lze využít při tvorbě interaktivního pohybu, detekci objektů, práci s objekty a mnoha dalších akcích potřebných k tvorbě kvalitního real-time obsahu.

Blender GameEngine Python API lze rozdělit na tři moduly – `GameLogic`, který obsahuje vše co se týká logických bloků a objektů ve scéně, `Rasterizer` obsahující funkce ohledně zobrazování, jako uložení snímku obrazovky, získání rozměrů okna a nastavení rozteče eye-separation při stereo-zobrazení a modulu `GameKeys` obsahujícího symbolické názvy všech možných kláves a konstant.

Pomocí skriptů lze vykonávat téměř vše, co jde specifikovat logickými cestami, kromě vytváření nových logických bloků. Ty bohužel musí být vytvořeny vždy v procesu návrhu mimo real-time režim. Skripty se dají volat pouze pomocí ovladače Python a jsou vždy provedeny kompletně celé. Během real-time běhu se v nich nesmí vyskytnout příkaz `print`, jinak se celá real-time aplikace zacyklí.

### 4.3 Nástroj GameEngine pro řešení kolizí

Blender GameEngine přímo v sobě obsahuje jednoduchý, ale mocný nástroj pro řešení kolizí. Pro každý objekt lze pomocí ovládacích prvků specifikovat, zda se pro něj mají počítat kolize, jestli má pevnou polohu nebo je pohyblivý, popřípadě jeho průhlednost a nehmotnost. Pro případ kolizí lze určit tvar jeho obálky, která může být např. krychle, koule, jehlan, dokonce i polyhedr.

Při řešení kolizí se též aplikuje nastavení vektoru tření, který lze nastavit ve stejném bloku tlačítek pro každou osu zvlášť. Dále je v real-time módu používána gravitační konstanta. Její nastavení můžeme nalézt ve World části panelu **Buttons Window**. Gravitaci lze též měnit dynamicky pomocí skriptu.

Možnosti, které nebyly aplikovány v tomto projektu, ale jsou v modulu obsaženy, se týkají senzorů. Některé typy senzorů jsou určeny pro kolizní interakci. Jsou to následující senzory:

- **Touch** – vyšle impuls, pokud se avatar dostane do kontaktu s požadovaným materiálem
- **Collision** – indukuje impuls, pokud se avatar dostane do kolize s objektem obsahujícím zadanou proměnnou.
- **Near** – vygeneruje impuls, pokud se avatar přiblíží do požadované vzdálenosti k objektu obsahujícímu zadanou proměnnou.
- **Radar** – vyšle impuls, pokud se v daném poli radaru, specifikovaném pomocí úhlu a hloubky kuželu, nachází objekt s požadovanou proměnnou. Tento senzor je hojně využíván při minimalizaci počtu objektů ve scéně, kdy jsou viditelné pouze ty v dosahu radaru.
- **Ray** – vyšle paprsek požadovaným směrem a do požadované vzdálenosti. Pokud narazí na objekt se specifikovanou proměnnou, generuje impuls. Tento senzor je využíván např. při akčních hrách jako zaměřovač. Jako jediný byl použit i v této práci na rozpoznání místností (viz sekce 5.4.3).

### 4.4 Nástroje GameEngine pro řešení pohybu

Pohyb lze řešit oběma uvedenými způsoby tvorby. Chceme-li vytvořit interaktivní pohyb pomocí Logic Bricks, stačí použít senzor **Keyboard**, respektive **Mouse** a přivést ho do aktivního prvku **Motion** nebo **Constraint**. Tímto způsobem lze snadno vyřešit pohyb, ten ale bude vždy pevné velikosti (rychlosti) a při velkém množství možností pohybu může být obtížné orientovat se ve spoustě logických cest.

Mnohem mocnějším nástrojem pro řešení pohybu je využití skriptů. Stačí propojit se všechny pohybové senzory a příslušný skript a v něm vše kontrolovat a nastavovat. Nakonec z něj lze zavolat jediný připojený aktivní prvek, jehož vlastnosti se nastaví rovnou pomocí skriptu a prvek pouze odešle informaci do real-time prostředí.

Tímto způsobem je řešen pohyb i v tomto projektu, kde se o něj starají dva spolupracující skripty, kdy jeden obsluhuje události myši a druhý události klávesnice.

## 4.5 Shrnutí tvorby v Blender GameEngine

Jak můžeme vidět, tvorba real-time obsahu v GameEngine je na první pohled velice snadná a promyšlená a práce v něm jde rychle od ruky. Na druhou stranu, materiálů, které by popisovaly tento modul, je příliš málo na to, aby se v něm dalo již od začátku pracovat naplno. Nabízí totiž ohromnou škálu možností.

Existují sice nějaké knihy zabývající se touto částí Blenderu, ale u nás jsou nedostupné a navíc se zabývají GameEngine verze 2.25, ve které jistě spousta začátečníků tvořit nebude. Také je volně ke stažení popis Blender GameEngine Python API a spousta tutoriálů a ukázkových aplikací na internetových stránkách [7].

## Kapitola 5

# Tvorba Blender Walkthrough Engine

Jedním z klíčových úkolů práce je vytvoření interaktivní real-time aplikace, která bude umožňovat průchod libovolným modelem budovy (model však musí splňovat určité podmínky, o nichž se dočteme v části 5.3 na straně 20). Práce se tak rozděluje na dvě části – vytvoření interaktivních prvků a práce se samotným modelem. Implementace mohla proběhnout mnoha způsoby, např. naprogramováním pomocí některé z grafických knihoven (OpenGL, DirectX). Tento způsob by se ale nakonec mohl projevit jako příliš náročný, zdoluhavý a navíc se nejedná o způsob, který by byl adaptivní. Interaktivní část by sice mohla zůstat neměnná, ale každý další model by musel být znovu naprogramován.

Další cestou mohlo být vytvoření modelu v některé z modelačních aplikací. To je docela snadné i pro ne příliš zdatné uživatele. Ale v těchto aplikacích lze vytvořit většinou jen statický model bez jediného prvku interakce. Dalším problémem tedy je, jak tento statický model dostat do real-time prezentace. Právě řešením tohoto problému se budu věnovat v této kapitole.

### 5.1 Příprava a návrh

Zvolil jsem možnost využití modulu Blender GameEngine. Jak jsme se dozvěděli v předchozí kapitole, je tvorba v něm snadná i pro ne příliš zdatného uživatele, pokud nevyžaduje pokročilejší funkce, které by se musel naučit. Též umožňuje tvorbu univerzálních skriptů a je provázán s jedním z nejmodernějších interpretovaných jazyků, jazykem Python.

Dalším zjednodušením pro cílového uživatele, které jsem zvolil, je napsání skriptu, po jehož spuštění se zobrazí uživatelské rozhraní (dále jen GUI), které umožňuje každý krok přípravy real-time obsahu jednoduše "naklikat".

Ještě předtím, než se pustíme do samotného procesu tvorby, je nutno si rozvrhnout, jak bude celý engine nakonec vypadat. Měl by být uživatelsky přívětivý a obsahovat všechny potřebné funkce. Mohou se v něm však vyskytovat i některé doplňkové funkce pro pokročilejšího uživatele. Ty ale nesmí působit rušivě pro uživatele neznalého, který je nebude využívat.

Měl jsem na výběr mezi různými způsoby zpracování rozhraní engine. Prvním návrhem bylo jeho vytvoření s využitím jednoho z mnoha grafických toolkitů (lze jmenovat např. GTK+, Qt, FLTK a spoustu dalších). Tento návrh se však projevil jako absolutně nevyhovující, protože by zde byla nulová nebo velmi nízká možnost provázání s funkcemi Blenderu. Ty jsou však potřebné při manipulaci s modelem.

Další z možností bylo vytvořit skript přímo v Blenderu, který načte konfigurační soubory, v nichž se nachází nastavení scény, modelu, materiálů pro výpočet radiozity a vše potřebné, aplikuje ho na importovaný model a výsledný produkt uloží do specifikovaného adresáře.

To by sice umožňovalo jednoduše spustit kompletní proces jedním krokem, což pro uživatele na jednu stranu znamená výhodu, že nemusí do tohoto procesu nijak zasahovat, ale na druhou stranu je to nevýhodou pro ty, kdo chtějí mít nad celým procesem kontrolu. Navíc uživatelé Windows nejsou příliš navyklí na používání konfiguračních souborů, tudíž by pro ně byla tato cesta konfigurace poněkud nezvyklá. Formát konfiguračních souborů by také nebyl příliš jednoduchý, jednalo se o způsob zápisu datového typu slovník (dictionary) jazyka Python.

Proto z této fáze práce vznikl poslední návrh, podle něhož jsem vypracoval finální řešení. Vytvořit GUI pomocí téhož skriptu, do něhož by byla veškerá funkčnost naimplementována pod jednotlivé grafické prvky a uživatel by komunikoval s činností procesu tvorby aplikace pomocí grafického rozhraní. Tím sice odpadá výhoda provedení celé akce naráz, ale uživatel tím získává možnost neustálé kontroly činnosti programu.

Naštěstí Blender Python API obsahuje podporu pro tvorbu grafických ovládacích prvků, takže celá tvorba spočívala v naprogramování tohoto GUI pomocí jazyka Python a Blender API.

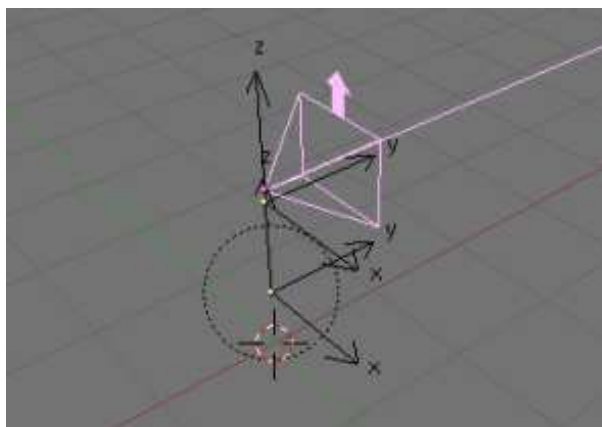
Nejprve je ale třeba vytvořit interaktivní část aplikace, která bude zajišťovat pohyb uživatele, reakce na jeho požadavky a zbývající interaktivní prvky. Až poté se budu věnovat tvorbě samotného GUI.

## 5.2 Implementace interaktivní části

Základem celé práce je vytvoření interaktivní části obsahu. Celá tato část je vytvořena pomocí Blender GameEngine. Na internetu lze nalézt spoustu šablon implementujících virtuálního avatara. Já jsem použil jednu přímo z domovských stránek Blenderu a poupravil ji podle potřeb. Tuto šablonu jsem zvolil z toho důvodu, že je v ní naimplementován mód volného pohybu (fly mode), který může sloužit např. k rychlému přesunu mezi převýšenými místy nebo po vypadnutí z modelu, důsledkem jeho špatného vytvoření, k návratu zpět.

### 5.2.1 Základ šablony

Šablona obsahuje objekt dotýkající se podlahy v odrazové obálce, nejlépe kulové, aby se mohl snadno pohybovat po rovině i do schodů. Dále objekt symbolizující rameno avatara, podél kterého lze pohled směřovat nahoru a dolů a nakonec objekt kamery symbolizující hlavu, potažmo oko avatara. Viz obrázek 5.1.



Obrázek 5.1: Šablona pro průchod

Tuto šablonu je třeba upravit přímo pro potřeby této práce. Všechny objekty jsou spojeny (připarentovány) k sobě, aby se při pohybu spodní styčné části přesunuly i ostatní. Dále jsou použity skripty, které obstarávají pohyb pomocí klávesnice a myši.

### 5.2.2 Úprava interaktivní šablony

Jednou z nutných úprav bylo přejmenování objektů tak, aby se shodná jména vyskytla v importovaných modelech s co nejmenší pravděpodobností. Dále bylo potřeba poupravit vlastnosti styčného objektu, jeho velikost, vektor tření a odrazivost od stěn. To lze provést pomocí tlačítek nastavení objektu (viz 4.1.1).

Největší změny jsem provedl v ovládacích skriptech. Ty byly upraveny pro pohyb pomocí kurzorových kláves a byl do nich přidán mód rychlejšího pohybu pomocí kláves Shift. Další funkce, které byly implementovány navíc jsou uvedeny v následujícím výčtu.

- **Uložení snímku obrazovky**

Klávesou S lze během real-time režimu uložit snímek aktuálního obrazu do adresáře, odkud je aplikace spuštěna. Snímky jsou pojmenovány jako `screenshotXXXX.png`, kde XXXX je pořadové číslo snímku doplněné zleva nulami.

- **Snížení zátěže procesoru**

Pomocí nastavení parametru `Sleep time` ve Walkthrough Engine (viz sekce 5.4.5) lze snížit zátěž procesoru při spuštění real-time režimu pomocí čekacích cyklů. Tato volba ale sníží počet snímků za vteřinu a plynulost celé prezentace.

- **Informativní texty**

Po provedení některých akcí se na obrazovku vypíše informativní text. Je možno je použít i pro vypisování textu příslušejícímu místnosti, ve které se avatar právě nachází. Tento text lze specifikovat pomocí GUI.

- **Ukončení pomocí ESC**

Přidal jsem i možnost okamžitého ukončení přehrávání obsahu pomocí stisku klávesy ESC.

### 5.2.3 Možnost použití jiné šablony v tomto projektu

Protože není příliš obtížné vytvořit vlastní real-time šablonu, která by implementovala některé nestandardní funkce, popř. ty nepoužívané by opomíjela, nebo protože jich lze na internetu nalézt nepřeberné množství, je do projektu zahrnuta možnost použít jinou real-time šablonu, která by uživateli více vyhovovala. Ta ale musí splňovat následující podmínky, aby nebyl chod skriptu přerušován různými chybami.

- **Jméno styčného objektu**

Při použití nové šablony je třeba skript seznámit s novými jmény objektů. Na začátku skriptu v oddělené sekci se nalézají specifikace těchto jmen. V šabloně se musí nalézat minimálně jeden objekt, který bude implementovat styčnou plochu s podlahou. Ten by se měl buď jmenovat `engViewer` nebo musí být jeho jméno přiřazeno do proměnné `avatarName`.

- **Proměnné pro bezchybný chod skriptu**

Engine využívá různých stavových proměnných, které ovlivňují jeho správný chod. Ty mohou být přiřazeny buď samostatnému objektu (nejlépe typu *Empty*) nebo mohou být součástí některého z objektů obsažených v šabloně. Všechny proměnné však musí náležet stejnému objektu.

Jméno tohoto objektu musí být přiřazeno do proměnné `propertyName`, popř. ho stačí pojmenovat jako `engProperties`. Jedná se o následující proměnné (jména i hodnoty jsou case sensitive):

- `mlx1Name` – typ `String`, výchozí hodnota `"=None1x1="`.
- `resizeRatio` – typ `Float`, výchozí hodnota `0.0`.
- `resizeSmaller` – typ `Int`, výchozí hodnota `1`.
- `modelFile` – typ `String`, výchozí hodnota `" "`.
- `MODEL_IMPORTED` – typ `Int`, výchozí hodnota `0`.
- `OBJ_1X1_SELECTED` – typ `Int`, výchozí hodnota `0`.
- `MESHES_COLLECTED` – typ `Int`, výchozí hodnota `0`.
- `RAD_CALCULATED` – typ `Int`, výchozí hodnota `0`.

#### • Proměnné styčného objektu

Objekt zmíněný v prvním bodě musí též obsahovat některé proměnné. Šablona s nimi nemusí nijak pracovat, ale je nutná jejich existence ve správném objektu. Jsou to tyto následující:

- `speed` – typ `Float`, výchozí hodnota `3.0`.
- `acceleration` – typ `Float`, výchozí hodnota `1.8`.
- `sleeptime` – typ `Float`, výchozí hodnota `0.0`.

#### • Použití informativních textů

Pokud budou i v externí šabloně použity fonty, musí v ní být obsaženy, správně namapovány na zobrazovací objekty, konfigurační proměnnou `useFonts` je třeba nastavit na `1` a hlavní objekt musí obsahovat následující proměnnou:

- `fontName` – typ `String`, výchozí hodnota jméno požadovaného fontu.

Do proměnné `strFonts` v konfigurační sekci skriptu musí být též vypsána jména používaných fontů. První položka tohoto seznamu ale musí zůstat prázdná.

Pokud fonty obsaženy nebudou, je potřeba konfigurační proměnnou `useFonts` nastavenit na `0`, jinak bude skript generovat chyby.

#### • Vyplnění seznamu objektů a textur

Dále je potřeba vyplnit seznam všech objektů obsažených v šabloně kvůli možnosti zjištění těch naimportovaných. To je provedeno vyplněním pole `initObjs` v konfigurační sekci.

Je též doporučeno umístit objekty související s fonty a informativními texty umístit do druhé vrstvy, pro jejich případné snadné zneviditelnění.

#### • Zbývající nastavení

Dále je nutno vyplnit další dvě pole – pole obrázků obsažených v šabloně (fonty, textury... ) a materiálů. Jedná se o pole `initImgs` a `initMats`. Pokud nebudou tyto informace vyplněny správně, nelze od skriptu očekávat korektní chování.

#### • Názvy IPO křivek

Protože je ve většině šablon naimplementováno otáčení a naklánění avatara pomocí IPO křivek, je nutné naplnit konfigurační proměnnou `avatarIpo` jménem IPO křivky styčného objektu, pomocí ní lze avatarem natáčet do stran, popř. tuto křivku pojmenovat jako `ipoViewer`.

V konfigurační části se též nachází jméno IPO křivky specifikující průhlednost dveří (proměnná `alphaIpo`), tato volba ale v současné době není funkční.

Poslední konfigurační proměnná se jménem `recalcNormals` určuje, jestli se po naimportování modelu má Blender pokusit přepočítat normály objektů směrem ven. Nicméně, pokud jsou normálové vektory umístěny špatným směrem již v importovaném modelu, nelze od této volby očekávat 100% výsledky.

### 5.3 Podmínky a doporučení pro importovaný model

Vzhledem k tomu, že bude model importován pomocí skriptu a budou nastavovány jeho vlastnosti, navrhnul jsem podmínky, které musí model splnit, aby měl uživatel usnadněnou práci s jeho pozdějším zpracováním.

- **Krychle o rozměrech 1x1**

V modelu by se měla nacházet krychle o velikosti 1x1, tak aby tato velikost odpovídala reálným rozměrům 1x1 metr. Podle této krychle bude také importovaný model zvětšen nebo zmenšen na správnou velikost a může být na jeho pozici snadno posunutý interaktivní avatar (virtuální osoba procházející modelem, obsahující kameru ve výšce asi 1,7 metru). Po importu modelu tato krychle nemusí působit rušivě, lze totiž pomocí engine zneviditelnit nebo ji lze zachovat, pokud má být nedílnou součástí modelu.

- **Výška schodů**

Avatar se dokáže vypořádat s pohybem do a ze schodů. Ty by ale měly dosahovat standardní výšky 20-30 cm (poměrově k jednotkové krychli). Do vyšších schodů se již avatar nemusí dostat, nižší mu samozřejmě nečiní žádné potíže.

- **Šířka dveří**

Pokud se v objektu vyskytují dveře nebo jiná propojovací místa, jejich šířka by měla dosahovat minimálně 1 metru. Jinak totiž hrozí, že jimi avatar nebude moci projít. Také by tyto dveře měly být jako samostatný objekt.

- **Formát výsledného modelu**

Importovaný model by měl být uložen v některém z následujících formátů podporovaných engine. Je zde též popsáno, která data lze v daném formátu naimportovat.

- `.3ds` – formát souborů programu 3D Studio MAX. Spolehlivě importuje objekty, UV mapování textur, jména objektů a materiály. Též je zde podpora pro obarvené vertexy (vertex colours). Tento formát je používán jako primární a byly na něm testovány všechny funkce.
- `.obj` – soubory programu Wavefront Maya. Importuje správně objekty, materiály a mírně upravená jména objektů. Problém mu nečiní ani textury a UV mapování, stejně jako obarvené vertexy.

Dále bych rád zmínil určitá doporučení, která by měl model dodržet pro snadnější a příjemnější import a následné zpracování importovaného modelu. Dodržení těchto doporučení později usnadní práci s modelem a zvýší možnosti nastavení.



- **Vizuální stránka**

Je lepší, pokud je model správně otexturovaný, protože s ním lze v engine provádět více operací. Textury musí být dodány současně s modelem a musí být dodržena velikost písmen v názvech souborů s texturami, obzvláště na systému Linux.

Model může sice pouze mít přiřazené materiály, ze kterých lze v engine spočítat radiozitu, ale toto řešení má spoustu omezení (viz sekce 5.5 na str. 24) a lze jej doporučit maximálně pro rychlou kontrolní procházku. Výsledné modely doporučuji dodávat otexturované.

- **Struktura modelu**

Toto doporučení se týká pouze otexturovaných modelů, protože při výpočtu radiozity jsou všechny zahrnuté objekty nahrazeny jedním výsledným.

Model by měl být členěn pomocí objektů na jednotlivé místnosti, dveře a ostatní objekty. Pro každý objekt lze totiž specifikovat jeho vlastnosti – dveře, místnost nebo skleněný objekt. Pokud je model dobře členěný, jeho nastavení bude mnohem jednodušší a navíc tím uživatel získá větší možnosti nastavení.

- **Cesty k texturám**

Jestliže model obsahuje textury, měly by se nacházet nejlépe ve stejném adresáři nebo maximálně v podadresáři první úrovně. Jejich cesta by též měla být specifikována pomocí relativního adresování. Je to kvůli ukládání výsledného projektu. Jinak by totiž engine nemusel textury najít nebo uložit na správnou pozici.

## 5.4 Tvorba engine

Během tvorby grafického rozhraní jsem rozdělil funkce Blender Walkthrough Engine do 4 skupin podle jejich pole působnosti a předpokládaného pořadí. V první skupině jsou funkce pro práci s externím souborem modelu, jeho naimportování a nastavení scény pomocí zvoleného referenčního objektu.

Do druhé skupiny spadají funkce pro nastavení vlastností jednotlivých naimportovaných objektů. Nalézá se zde specifikace typu objektu a jeho vizuální parametry.

Třetí skupina slouží k výpočtu radiozity. Vzhledem ke komplexnosti celého procesu je tato skupina nejpočetnější a obsahuje nejvíce ovládacích prvků.

Do poslední skupiny jsem zařadil funkce sloužící k nastavení vlastností real-time sezení a uložení celého projektu a všech potřebných souborů do oddělené složky.

V celém skriptu jsou bohatě využívány funkce Blender Python API [5] a je kompletně napsán v jazyce Python. V následujících sekcích bych rád popsal jednotlivé skupiny funkcí rozmístěné do příslušných záložek uživatelského rozhraní skriptu. Zmíním též některé implementační detaily uživatelských prvků. Tento popis neslouží jako uživatelský manuál pro práci s grafickým rozhraním engine, ten je k nalezení v příloze na straně 36.

### 5.4.1 Spuštění skriptu

Po spuštění Blender Walkthrough Engine je ještě před vykreslením samotného GUI provedeno několik akcí.

- Ze všeho nejdříve se nahrají do paměti obrázky používané v GUI – logo, případně ukázky použitých fontů.

- Poté se naplní slovník všech objektů ve scéně, které nejsou přímou součástí real-time šablony – nebyly specifikovány v konfigurační části, viz sekce 5.2.3. Tento slovník je dále používán prvky pro práci s objekty a je při každé změně aktualizován.
- Dále skript zjistí, jestli již byl proveden import modelu, a pokud ano, nastaví ovládací prvky do takové polohy, ve které se nacházely při ukončení předchozího sezení. To se týká všech nastavení objektů, polohy atd. Neobnovuje se pouze seznam objektů, na něž měla být aplikována radiozita. Místo toho je vytvořen nový seznam z vyselektovaných objektů ve scéně.
- Následujícím krokem je nastavení rotace avatara podle jeho IPO křivky.
- Na závěr je spuštěno samotné grafické rozhraní skládající se ze čtyř záložek zastupující jednotlivé tematické skupiny funkcí.

Pro všechny záložky je vytvořena logovací část, do níž jsou vypisovány informační a chybové hlášky skriptu. Každá záložka má svou část pouze se svými zprávami. Jsou navrženy tak, aby se zobrazovaly vždy nejnovější zprávy. Pokud je větší počet zpráv, než je oblast schopna zobrazit, lze mezi nimi rolovat pomocí kláves se šipkou nahoru a dolů.

Tyto logy lze též vyčistit pomocí tlačítek vpravo dole. **Current** vymaže všechny logovací zprávy na aktivní záložce, zatímco tlačítkem **All** lze odstranit logovací zprávy ze všech záložek. Pomocí **Exit** se ukončí činnost skriptu.

### 5.4.2 Import modelu a nastavení kalibračního objektu

První záložka obsahuje ovládací prvky sloužící k naimportování modelu do scény a specifikování jednotkového objektu v něm. Dále zde lze měnit velikost modelu oběma směry, přemísťovat avatara a nátačet jeho pohled v rozmezí  $-180^\circ$  až  $180^\circ$ . Záložka je zobrazena na obrázku 5.2.

- **Browse**

Vyvolá standardní dialog Blenderu pro otevření souboru. Toho je dosaženo pomocí volání fce Python API

```
Window.FileSelector(loadModelFile, 'Select model file...')
```

V tomto případě je `loadModelFile` callback funkce zajišťující akci se zvoleným názvem souboru.

- **Import model**

Vytvoří instanci třídy `Importer` naimplementované ve stejném skriptu, které předá jako parametr název souboru. Ta podle přípony rozezná typ souboru a pokud je podporován, spustí správnou importovací funkci. Je nutné dodávat spolu s tímto skriptem i adresář `import`, v němž se nacházejí potřebné importovací skripty.

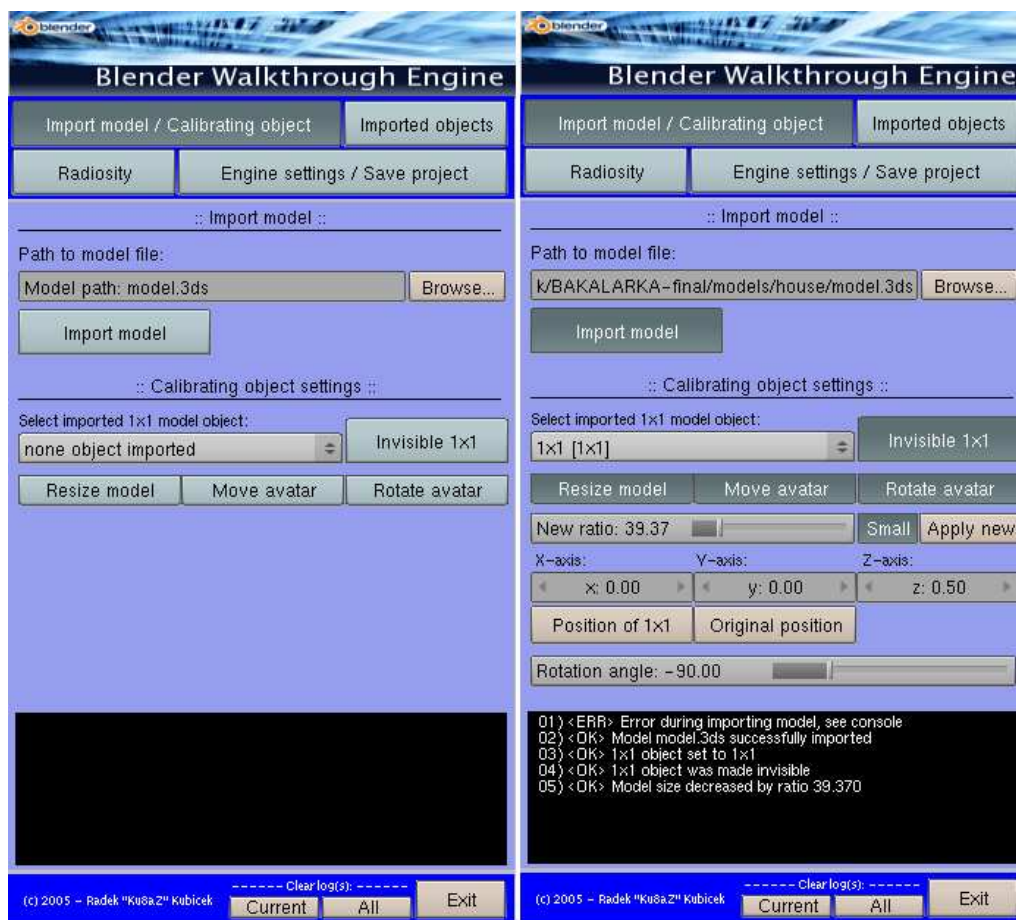
Proběhne-li import modelu v pořádku, tlačítko se stiskne a zaktivují se i všechna ostatní.

- **Select imported 1x1 model object**

V tomto seznamu je nutno zvolit kalibrační objekt. Podle něj se poté aplikují všechna další nastavení.

- **Invisible 1x1**

Podle stavu tlačítka zapne neviditelnost a vypne kolize u zvoleného kalibračního objektu. To je provedeno pomocí příznaků `Blender.NMesh.FaceModes[ 'INVISIBLE' ]` a `Blender.NMesh.FaceModes[ 'DYNAMIC' ]` kalibračního objektu.



Obrázek 5.2: Záložka menu č.1

- **Resize model**

Automaticky změni velikost modelu v poměru ku zvolenému kalibračnímu objektu a zobrazí část GUI pro úpravu velikosti modelu, jejíž součástí jsou tyto ovládací prvky:

- **New ratio:** – nový poměr velikosti původního naimportovaného modelu a modelu s upravenou velikostí. Při změně hodnoty nejsou změny ihned aplikovány.
- **Small** – pokud je tlačítko stisknuto, velikost importovaného modelu se v tomto poměru zmenší, jinak je zvětšena.
- **Apply new** – aplikuje nové nastavení velikosti. Volá funkci

```
resizeScene(sldRatio.val)
```

Ta provádí následující akci – pokud není zvolen kalibrační objekt, ukončí se. Jinak zjistí souřadnice BoundBoxu tohoto objektu. Není-li zadán parametr funkce, automaticky změni velikost modelu podle aktuálního poměru velikostí. Je-li parametr zadán, aplikuje se nové nastavení velikosti. Model se změni zpět na původní velikost a poté je na něj aplikován nový poměr velikosti. Pokud poměr dosáhne hodnoty 1.0, příslušná část GUI se zneviditelná a tlačítko uvede do počáteční polohy.

- **Move avatar**

Zviditelňuje část GUI sloužící k přesunu avatara na novou počáteční pozici. Ovládací prvky dostupné v této části:

- **x, y, z**

Změna pozici na příslušné souřadnici. Bohužel, kliknutí na okraj číselných tlačítek není funkční, je nutné stisknout levé tlačítko myši na požadovaném okraji a tažením měnit pozici. Přesnější hodnoty lze zadat kliknutím na číselnou hodnotu uvnitř tlačítka a zadáním na klávesnici. Změny pozice je dosaženo voláním následující funkce

```
engViewer.setLocation(numX.val, numY.val, numZ.val)
```

- **Position of 1x1**

Přesune objekt avatara na pozici kalibračního objektu. Zjistí BorderBox tohoto objektu, spočítá jeho střed a přesune do něj střed avatara.

- **Original position**

Vrátí avatara na původní pozici. Volá API funkci

```
engViewer.setLocation(0.0, 0.0, 0.5)
```

- **Resize avatar**

Zobrazuje příslušnou část GUI patřící této volbě. Nachází se zde ovládací prvek **Rotation angle**. Tím lze natočit avatara v rozsahu  $-180^\circ$  až  $180^\circ$ . Změna je aplikována okamžitě a je jí dosaženo úpravou IPO křivky pomocí následujícího kódu:

```
ipo = Blender.Ipo.Get(avatarIpo) # (1)
rot = ipo.getCurve('RotZ') # (2)
pts = rot.getPoints() # (3)
pts[0].setPoints([pts[0].pt[0], sldRotate.val/10]) # (4)
pts[1].setPoints([pts[1].pt[0], (sldRotate.val/10)+18.0]) # (5)
rot.Recalc() # (6)
engViewer.RotZ = (sldRotate.val*PI)/180 # (7)
engViewer.getData().update() # (8)
```

V kroku (1) získáme objekt IPO křivky. Krokem (2) v něm nalezneme křivku pro rotaci kolem osy Z a krokem (3) její body (pro rotační křivky se souřadnice na ose Y násobí  $10^\circ$ ). Příkazy (4) a (5) zajistí přeměnu IPO křivky na požadované hodnoty, kde se do spodního bodu uloží požadovaná počáteční hodnota a do bodu 2 hodnota o  $180^\circ$  vyšší. V (6) je přepočítána křivka a pomocí příkazu (7) se provede otočení avatara ve 3D okně. Poslední příkaz (8) aktualizuje objekt avatara.

### 5.4.3 Nastavení naimportovaných objektů

### 5.4.4 Výpočet radiozity

### 5.4.5 Nastavení real-time a uložení projektu

## 5.5 Zjištěné problémy a nedostatky

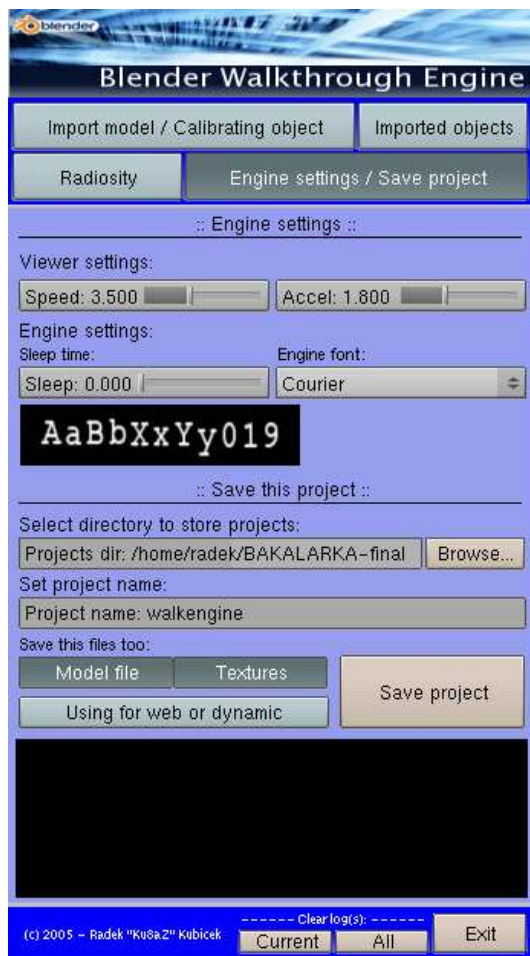
- problémy s informativními texty
- radiozita
- různé verze blenderu



Obrázek 5.3: Záložka menu č.2



Obrázek 5.4: Záložka menu č.3



Obrázek 5.5: Záložka menu č.4

## Kapitola 6

# Možnosti prezentace vytvořených dat

Výstupem Blenderu nemusí být pouze vyrenderovaný obrázek nebo animace, ale též prezentace vytvořeného real-time obsahu. Vždy je možnost všechna vytvořená, importovaná nebo jinou cestou vložená data, nacházející se v aktuálním sezení, uložit do *.blend* souboru. Tento datový soubor obsahuje vše potřebné a je nativním formátem pro výměnu dat aplikace Blender. Navíc neslouží pouze pro uchovávání dat, ale velice snadno z něj můžeme spustit připravenou real-time scénu pomocí přehrávače **BlenderPlayer**, jenž je standardně dodáván jako součást balíku Blender, nebo exportovat jeho obsah do samostatně spustitelné aplikace.

Této možnosti přehrání real-time obsahu je využíváno i v pomocném plug-inu (modulu) do webových prohlížečů, díky němuž snadno zobrazíme real-time obsah nacházející se v *.blend* souboru přímo v okně webového prohlížeče. Je třeba si však uvědomit, že tato možnost prezentace není bez omezení. Právě naopak, v této oblasti budeme omezení více než při ostatních způsobech.

Způsob spuštění pomocí aplikace BlenderPlayer je již častější. Nemá taková omezení, lze nastavit různé parametry, např. velikost okna, použití zvuků a dokonce způsob stereo-zobrazení.

Další možností je exportování obsahu do spustitelné aplikace. Jsou zde též určitá omezení, o kterých se zmíním v sekci 6.4. Tato forma prezentace je však nejlépe využitelná pro ty, kdo nemají ve svém počítači program Blender.

### 6.1 Zobrazování pomocí webového prohlížeče

Real-time obsah lze za pomoci speciálního modulu (Blender Web Plug-in) zobrazit přímo v okně internetového prohlížeče. Tento modul, který je volně ke stažení, bohužel není od doby, co se Blender stal open-source projektem, nadále vyvíjen. A protože GameEngine za ten čas ušel obrovský kus cesty, nefungují v tomto modulu některé aplikace na 100%. Výjimkám a různým problémům se budu věnovat v sekci 6.1.1.

Integrace modulu do prohlížeče je velice jednoduchá. Na jeho domovské stránce <sup>1</sup> stačí kliknout na tlačítko pro zvolený prohlížeč a modul se do něj nainstaluje automaticky. Tato starší verze podporuje MS Internet Explorer (musí být povolena technologie ActiveX) a Netscape Navigator.

Pro prohlížeče rodiny Mozilla lze použít také, nejprve je však nutné změnit příponu staženého souboru s modulem z **.jar** na **.zip**. Též vzniká nová verze, jejíž domovská stránka se nachází na <sup>2</sup>, ale protože je ve velice ranném stádiu vývoje, není výstup vždy perfektní. Podrobnosti jsou popsány v sekci 6.1.1.

---

<sup>1</sup>[http://www.blender3d.org/cms/3D\\_web\\_plug-in.15.0.html](http://www.blender3d.org/cms/3D_web_plug-in.15.0.html)

<sup>2</sup><http://www.neeneenее.de/blender/plugin/>



Po nainstalování modulu se můžeme pustit do tvorby webové stránky, která bude sloužit k vlastní prezentaci. Na domovském webu uživatelů Blenderu lze nalézt šablonu <sup>3</sup>, po jejímž vložení do kódu stránky se bude volat rozhraní modulu. Jsou zde podrobněji popsány i možnosti nastavení, předávání parametrů real-time aplikaci, způsoby interakce HTML kódu s modulem, změna úvodní animace při načítání souboru a další potřebné informace.

### 6.1.1 Nalezené problémy se zobrazováním

Během používání webového modulu se může vyskytnout některý z následujících zjištěných problémů, popř. jiný, na který jsem nenarazil při přípravě této práce. Dále jsou zde přiblíženy i některé nedostatky tohoto modulu.

#### (Ne)multiplatformnost

Původní modul byl sice vyvíjen jako multiplatformní, ale pod Linuxem se mi ho nepodařilo zprovoznit, protože Mozilla/Firefox nedokáže příliš spolupracovat se starým formátem, který byl používán prohlížečem Netscape. Pod Windows však modul v prohlížeči Mozilla Firefox zprovoznit lze. Je tedy použitelný pouze na platformě Windows, naštěstí ve většině prohlížečů.

#### Použití příkazu `print`

Tento příkaz v některém ze zpracovávaných skriptů je přímou zkázou celé aplikace, protože se skript zacyklí a aplikace se začne chovat nepředvídatelně nebo bude neovladatelná. Je to jedna z vůbec nejčastějších chyb, na kterou ale uživatelé nejčastěji zapomínají, a nevyskytuje se jen u webového modulu, ale u všech real-time výstupů, včetně přehrávání pomocí BlenderPlayeru.

#### Verze Blenderu

Dalším problémem je verze Blenderu. Modul je totiž stavěn pro verzi 2.25, která je považována za přelomovou právě v oblasti GameEngine a modul byl dokončen v době, kdy byla aktuální a placená.

Tato verze však neobsahuje téměř žádné GameEngine Python API, používá jinou verzi BGL a chybí jí některé funkce implementované v pozdějších verzích. Dalším problémem může být i její rozdílné uživatelské rozhraní, což sice není problém přímo web modulu, ale uživatelé navyklému na novější verzi 2.3x, může pravděpodobně působit nemalé potíže při tvorbě real-time obsahu.

#### Zprávy

V GameEngine lze používat pro komunikaci mezi objekty zprávy (Messages). Webový modul tyto zprávy nedokáže interpretovat, tudíž nebudou fungovat akce s nimi přidružené. Na to je při tvorbě obsahu dávat pozor a předem se vyvarovat návrhu s použitím zpráv. V real-time obsahu vytvořeném pomocí Blender Walkthrough Engine se to projeví tím způsobem, že nebudou funkční informativní texty.

#### Problémy s novější verzí pro Mozilla/Mozilla Firefox

Novější verze modulu se vyznačuje tím, že lze importovat do prohlížečů rodiny Mozilla a měla by implementovat některé z nových funkcí GameEngine. Ale protože není dostatečně dlouho vyvíjena, obsahuje některé závažné chyby:

---

<sup>3</sup>[http://www.blender3d.org/Education/index\\_old.php?sub=TutorialEmbedplugin](http://www.blender3d.org/Education/index_old.php?sub=TutorialEmbedplugin)

- *chybná interpretace událostí myši* – myš se chová podivně a je velice těžké se pohybovat s její pomocí.
- *pomalejší pohyb* – pokud v real-time aplikaci pohybujeme nějakou postavou či virtuálním avatarem, je pohyb znatelně pomalejší než ve starším modulu běžícím pod Internet Explorerem.
- *zhoršení grafické kvality zobrazení* – real-time aplikace po grafické stránce vykazuje více artefaktů než ve starší verzi modulu. Může to být způsobeno samotným webovým modulem, verzí Blenderu či špatným namapováním textur, ale tato "chyba" se projeví vždy.
- *nestabilita* – občas se stane, že se modul chová nestabilně, většinou po delší době práce. Ale jak již bylo zmíněno, je to způsobeno tím, že se jedná o rannou verzi a chyba jistě bude s příští verzí opravena. Nestabilním chováním je myšleno pozdní a často chybné zareagování na vstupní událost a zamrzání prohlížeče.

## 6.2 Přehrávání pomocí BlenderPlayer

Pokud máte všechna potřebná data uložena v .blend souboru a rádi byste jejich obsah prezentovali bez spuštění Blenderu nebo použití omezených schopností web plug-inu, lze využít aplikaci BlenderPlayer dodávané společně s balíkem Blender.

Práce s přehrávačem je velice snadná. Stačí mu předat jako parametr jméno .blend souboru s požadovaným real-time obsahem, který přehrávač automaticky spustí. Navíc obsahuje spoustu nepovinných parametrů, kterými lze ovládat přehrávání. Jejich podrobnější přehled se dozvíte po zadání `blenderplayer -h`. Nachází se zde parametry pro nastavení velikosti okna (daná pozice a rozlišení nebo přes celou obrazovku), spuštění stereo-zobrazování a jeho typ nebo nastavení chování samotného přehrávače, kde lze zapnout zobrazování ladicích proměnných, počtu snímků za vteřinu (fps) nebo zapnout či vypnout některé funkce, čímž lze dosáhnout vyšší rychlosti zobrazování na úkor kvality.

Po spuštění s parametrem `-c` vypisuje všechna upozornění a varování. Tento parametr je obzvláště důležitý na platformě Windows, pokud se real-time obsah nechová, jak by měl. Zde totiž `blenderplayer` implicitně nevypisuje žádné zprávy, dokud není specifikován tento parametr.

Ani tento způsob prezentace není bez nedostatků. O problémech, které se mohou vyskytnout se můžeme dočíst v 6.4.

## 6.3 Export do spustitelné aplikace

Tento způsob se od možnosti přehrávání liší pouze v tom, že je při něm vytvořena binární spustitelná aplikace, do níž je zakomponován `blenderplayer`, tudíž na cílovém počítači nemusí být vůbec přítomen. Výsledný soubor lze spouštět se stejnými parametry jako pomocí `blenderplayeru`.

Samotný export se dá provést jediným způsobem. Pomocí hlavního menu Blenderu posloupností těchto kroků: *File -> Save Runtime*. Ve verzi pro Windows se v menu *File* nachází i položka *Save Dynamic Runtime...*, která tam ale zůstala z předchozích verzí a v současnosti pouze vygeneruje chybu.

Po exportu se vytvoří požadovaný soubor, do něhož je automaticky přidán `blenderplayer`. Pro jeho spuštění jsou tedy potřebné jen 2 dynamické knihovny nacházející se v adresáři s Blenderem – **Python23.dll** a **SDL.dll**, popř. v linuxové verzi s příponou **.so**. Tyto knihovny je nutno distribuovat současně s vygenerovanou aplikací.

Také je potřeba dodat s aplikací veškeré soubory s texturami, jinak bude model během přehrávání neotexturován. Je zde možnost zahrnout všechny použité textury do .blend souboru, aby nemusely být dodávány s aplikací, ale tato akce musí být provedena ručně (viz sekce 6.5).

## 6.4 Potíže při přehrávání real-time obsahu

Vzhledem k tomu, že je dynamický obsah spustitelného souboru interpretován pomocí integrovaného blenderplayeru, je množina problémů a nedostatků shodná pro obě dvě výše zmíněné možnosti prezentace.

- **Použití informativních textů a radiozity současně**

Pokud je v real-time obsahu použit model, který má vypočítanou radiozitu a ta je na něj aplikována, a současně jsou použity informativní texty, během přehrávání není vidět nic jiného než právě tyto texty a barva okolního světa.

Tato "chyba" se objevuje ve všech stabilních verzích, při použití vývojové verze (viz sekce 3.1) je však již vše přehráno správně.

Dalším možným řešením je přepnutí viditelných vrstev pouze na první vrstvu. Protože jsou tyto texty uloženy ve druhé vrstvě, ve výsledku prezentace se neuplatní a vše je prezentováno správně.

- **Uložení snímku obrazovky při změně velikosti okna**

Jestliže uživatel během přehrávání změní velikost okna přehrávače a rozhodne se poté uložit snímek obrazovky, ten se neuloží správně, ale s rozhozeným barevným spektrem a nepřesnou strukturou. Při uložení snímku v okně o původní velikosti je vše provedeno správně.

Tato chyba se opět objevuje ve všech stabilních verzích, při použití novější, vývojové verze je snímek uložen v pořádku.

- **Použití parametru `-c` při přehrávání obsahu s chybou**

Je-li v prezentovaném obsahu zanesena chyba, většinou se přehrávání nechová správně, ale toto chování nijak neovlivní rychlost chodu aplikace. Pokud je ale při přehrávání takového obsahu použit parametr `-c`, chybové hlášení se neustále vypisují do okna konzole (terminálu) a rychlost běhu aplikace se rapidně sníží.

Toto však není chyba, ale předpokládané a korektní chování a zde ho uvádím pouze pro případ, že by uživatel nedokázal zjistit důvod zpomalení přehrávání.

## 6.5 Rozšiřující možnosti

- zabalení textur
- stereo-zobrazení

# Kapitola 7

## Závěr

Cílem této práce bylo vytvořit univerzální engine, který by umožňoval interaktivní průchod naimportovaným modelem budovy. Ten byl vypracován s použitím programu Blender. Jeho implementace prošla mnoha etapami, mezi nimiž byl i kompletně nový návrh a následné celkové přepracování.

O tvorbě a použití tohoto engine, stejně jako základních informacích o použité aplikaci pojednává tato technická zpráva. Tu lze použít jako referenční příručku k základům práce s aplikací Blender, stejně jako uživatelský manuál k vytvořenému engine.

Projekt zdaleka neobsahuje vše, co by šlo naimplementovat a co umožňuje použítá aplikace, ale je plně funkční a většině uživatelům může sloužit k plné spokojenosti. Jsou zde i jistá vylepšení, která by ještě mohl mít naimplementována a zesílil tím svou funkčnost, ale ty již v rámci vypracování projektu nebyla tak nutná. Lze je chápat i jako návrhy do budoucna pro možné vylepšení této práce.

### 7.1 Návrhy pro pokračování práce

V této části bych rád zmínil některé implementační body, na které jsem v průběhu řešení narazil, ale nepovažoval jsem je za důležité k dokončení projektu. Dále zde jsou uvedena možná vylepšení, která nejsou současnými schopnostmi Blenderu uskutečnitelná, popř. byla mimo rozsah a zadání práce.

- **Zvuky**

Vhledem k tomu, že Blender i jeho GameEngine obsahují komunikační vrstvu pro grafickou a zvukovou knihovnu SDL, lze do real-time obsahu zahrnout i podporu zvuků. V tomto projektu by bylo nejlepší použít zvuky pro chůzi, otevírání dveří (které se ve skutečnosti neotevírají, ale jsou průchozí), popř. některé další zvukové efekty. Jejich přidání není obtížné a zároveň dokáží navodit správnou atmosféru.

- **Podpora pro více formátů**

V současné době jsou podporovány pouze dva formáty souborů, ze kterých lze naimportovat model. Dalším vylepšením by mohlo být naimplementování podpory pro více formátů souborů s modely.

- **Importer pro X3D a VRML2**

Toto vylepšení souvisí i s předchozím bodem. Protože jsou X3D a VRML2 standardizované formáty pro výměnu grafických dat, tudíž jsou velice dobře popsány, bylo by vhodné naimplementovat importéry pro tyto formáty. Většina modelačních programů ve svých nejnovějších verzích dokáže exportovat data do těchto XML formátů, jejich podpora by tedy mohla být vítána.

- **Real-time stínování**

Tato možnost není současnou verzí Blenderu podporována. Zde jsou stíny spočítány pouze staticky a poté jednoduše připojeny k modelu. Do budoucna by ale mohlo jít počítat stíny a radiozitu dynamicky a aplikovat je na objekty v reálném čase.

- **Vylepšené "otevírání" dveří**

Lepší volbou pro rozpoznání a "otevření" dveří v modelu by bylo nejspíše jejich přímé otevření nebo alespoň postupné zneviditelnění při přiblížení se k nim. Tato možnost je již zčásti naimplementována, ale protože GameEngine stále nepodporuje dynamické vytváření logických bloků za běhu, vyžadovala by ruční zásah uživatele.

Klasické otevírání dveří by bylo pravděpodobně velmi obtížně proveditelné, protože se musí specifikovat strana, na níž se nachází panty, přesunout na ni pivot objektu a dále vytvořit logické bloky s požadovaným pohybem dveří, což by se neobešlo bez zásahu uživatele. Navíc existuje spousta typů dveří a mnoho způsobů otevírání, engine by tedy musel obsahovat velké množství různých IPO křivek specifikujících pohyb různých dveří.

## Kapitola 8

# Slovníček pojmů

**BGL** – Blender OpenGL. Rozhraní Blenderu implementované kompletně v OpenGL, což zajišťuje rychlost a multiplatformnost grafického rozhraní. Tento modul obsluhuje volání OpenGL příkazů ve skriptech.

**Blender Python API** – API rozhraní Blenderu pro interpret jazyka Python. Najde široké uplatnění při tvorbě různých skriptů a modulů běžících v prostředí Blenderu. Umožňuje práci s veškerými moduly Blenderu (kromě Game Engine), práci s objekty, vertexy, IPO křivkami, .blend soubory, světly, texturami a dalšími objekty uloženými v .blend souborech.

**Blender GameEngine Python API** – API rozhraní modulu GameEngine pro interpret jazyka Python. Používá se při tvorbě real-time aplikací, převážně her. Umožňuje reagovat na události nebo různé události vytvářet a volat. Je používáno pouze vnitřním rozhraním Blenderu během spuštěné real-time aplikace.

# Literatura

- [1] Findseiss F., Heizer A., McKay R., Oppel J., Roosendaal T., Selleri S., Veldhuizen B., Wartmann C.: Blender Documentation, 2003 Dokument dostupný na URL <http://download.blender.org/documentation/BlenderManual.pdf.zip>
- [2] knížka o Pythonu

## Časopisy:

- [3] CHIP Duben/2005

## Internetové zdroje:

- [4] <http://www.oskarimax.cz>
- [5] <http://www.blender.org/modules/documentation/236PythonDoc/index.html>
- [6] [http://www.blender.org/modules/documentation/pydoc\\_gameengine/PyDoc-Gameengine-2.34/index.html](http://www.blender.org/modules/documentation/pydoc_gameengine/PyDoc-Gameengine-2.34/index.html)
- [7] <http://www.blender.org/modules/documentation/>

Všechny zde uvedené URL zdrojů byly na zadaných adresách dostupné v dubnu 2005.

## **Kapitola 9**

# **Přílohy**

### **9.1 Návod k používání Blender Walkthrough Engine**