

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Animovaný interaktivní 3D model části počítače

Diplomová práce

Vedoucí práce:

Ing. Naděžda Chalupová, Ph.D.

Autor práce:

Bc. Petr Michalík

Brno 2015

Rád bych poděkoval vedoucí mé práce Ing. Naděždě Chalupové, Ph.D. za odborné vedení této diplomové práce, cenné rady a věcné připomínky, které mě vedly správným směrem při vypracovávání této práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Animovaný interaktivní 3D model části počítače** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmetná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 21. května 2015

Abstract

Michalík, P. Animated interactive 3D model parts of computer. Diploma thesis. Brno: Mendel University in Brno, 2015.

This diploma thesis describes possibilities of creation, use and presentation three-dimensional models, selection of a suitable solution for creation and presentation three-dimensional models, creating static three-dimensional models of computer components and their completion of animation and interactive elements to these models.

Keywords

Interactive three-dimensional model, 3D modelling, animation, VRML, X3D, Rhinoceros, Blender, SketchUp, computer components.

Abstrakt

Michalík, P. Animovaný interaktivní 3D model části počítače. Diplomová práce. Brno: Mendelova univerzita v Brně, 2015.

Diplomová práce popisuje možnosti tvorby, využití a prezentace trojrozměrných modelů, výběr vhodného řešení tvorby a prezentace trojrozměrných modelů, vytvoření statických trojrozměrných modelů počítačových komponent a jejich doplnění o animaci a interaktivní prvky.

Klíčová slova

Interaktivní trojrozměrné modely, 3D modelování, animace, VRML, X3D, Rhinoceros, Blender, SketchUp, počítačové komponenty.

Obsah

1	Úvod a cíl práce	15
1.1	Úvod.....	15
1.2	Cíl práce.....	16
2	Možnosti tvorby trojrozměrných modelů	17
2.1	Použití 3D grafických programů.....	17
2.1.1	Program Rhinoceros	17
2.1.2	Program Blender	19
2.1.3	Program SketchUp	20
2.1.4	Ostatní programy	21
2.1.5	Výběr vhodného 3D grafického programu.....	21
2.2	Použití programovacích jazyků	22
2.2.1	Jazyk VRML.....	23
2.2.2	Jazyk X3D.....	23
2.3	Výběr vhodné metody pro tvorbu modelů.....	25
3	Možnosti zobrazení modelů	26
3.1	Zobrazení na internetu.....	26
3.2	Zobrazení pomocí programu SketchUp.....	26
3.3	Zobrazení pomocí 3D grafických programů	27
4	Současný stav	28
4.1	Využití trojrozměrných modelů	28
4.1.1	Prototypování.....	28
4.1.2	3D tiskárny	28
4.1.3	Průmyslová výroba.....	29
4.1.4	Architektura	30
4.1.5	Prezentace výrobků.....	30
4.2	Trojrozměrné modely PC komponent.....	31

5	Příprava materiálů	34
5.1	Využití internetu	34
5.2	Nafocení komponent	35
5.3	Shromáždění komponent	35
6	Vytvoření trojrozměrného modelu	37
6.1	Metody využití při tvorbě modelů	37
6.1.1	VRML a Rhinoceros	38
6.1.2	Příkazy DEF a USE	39
6.1.3	Použití prototypů	42
6.1.4	Nastavení možností procházení modelu	44
6.1.5	Vytvoření a mapování textur	46
6.2	Tvorba jednotlivých modelů	49
6.2.1	Processor	49
6.2.2	Paměť RAM	50
6.2.3	Audio konektor	50
6.2.4	Konektor PS/2	51
6.2.5	USB konektor	51
6.2.6	Síťový konektor	52
6.2.7	Patice procesoru	52
6.2.8	Konektory DVI a D-Sub	53
6.2.9	Sběrnice PCI-Express	54
6.2.10	Ostatní modely	54
6.3	Sestavení trojrozměrného modelu z částí	55
6.3.1	Doplnění animace	55
6.3.2	Návrh aplikace	56
6.3.3	Vytvoření aplikace a doplnění interaktivních prvků	57
6.3.4	Použití a popis zásuvného modulu	59
7	Závěr	61
7.1	Využití modelu	61
7.2	Možnosti dalšího rozšíření práce	61

Obsah	11
8 Literatura	63
A Ukázky ostatních modelů	67
B Seznam souborů na přiloženém CD	70

Seznam obrázků

Obr. 1	Rhinoceros	18
Obr. 2	Blender	19
Obr. 3	SketchUp	21
Obr. 4	Ukázka 3D tisku	29
Obr. 5	Ukázka gravírování	30
Obr. 6	Ukázka modelu základní desky	32
Obr. 7	Detailní pohled na model základní desky	32
Obr. 8	Ukázka propracovaného modelu základní desky	33
Obr. 9	Zděř audio konektoru	40
Obr. 10	Textury procesoru	47
Obr. 11	Textura paměti RAM	47
Obr. 12	Textura základní desky	48
Obr. 13	Textura patice procesoru	48
Obr. 14	Procesor	49
Obr. 15	Paměť RAM	50
Obr. 16	Audio konektor	51
Obr. 17	Konektor PS/2	51
Obr. 18	Konektory USB 2.0 a USB 3.0	52
Obr. 19	Síťový konektor	52
Obr. 20	Patice procesoru	53
Obr. 21	Konektory DVI a D-Sub	54
Obr. 22	Sběrnice PCI-Express	54

Obr. 23	Základní deska	55
Obr. 24	Struktura aplikace	57
Obr. 25	Navržená aplikace	58
Obr. 26	Ukázka prostředí zásuvného modulu Cortona3D	59

Seznam tabulek

Tab. 1	Hodnocení programů dle stanovených kritérií	22
---------------	--	-----------

1 Úvod a cíl práce

1.1 Úvod

V dřívějších dobách byli lidé nuceni provádět některé činnosti, které my v dnešní době díky výpočetní technice považujeme za triviální, velmi složitým a zdoluhavým způsobem. Před tím, než se výpočetní technika začala rozšiřovat, byli lidé nuceni ve všech oblastech své činnosti jako je vzdělávání, plánování, výroba či různé prezentace využívat pouze fyzické předměty. Mnohdy tyto předměty mohli být velmi velké nebo jich mohlo být velké množství a manipulace s nimi mohla být velmi složitá.

Dnes si při mnoha svých činnostech vystačíme pouze s výkonným počítačem nebo dokonce i s obyčejným tabletem. Rozvoj výpočetní techniky usnadnil lidem práci ve všech ohledech. Dříve, když jsme potřebovali někoho kontaktovat, mu bylo nutné poslat dopis. Než byl takový dopis doručen, uplynulo několik dní a navíc jsme museli další dny čekat, než nám dotyčný odešle dopis s odpovědí a než nám tento dopis přijde. Dnes se jednoduše přihlásíme na email a dotyčnému napíšeme zprávu během krátké chvíle a zpráva mu ihned dorazí.

S rozvojem počítačů a internetu bylo usnadněno i získávání informací. Dříve byli lidé odkázáni pouze na tištěné knihy, noviny či časopisy. Potřebné informace museli složitě hledat a byli nuceni přečíst velké množství textu. Dnes si pomocí počítače a internetu najdeme informace takřka o čemkoliv a máme je ihned k dispozici. Nemusíme tak složitě hledat vhodnou knihu a konkrétní článek v ní.

Zmíněné příklady jsou však pouze nejzákladnější činnosti. Potenciál výpočetní techniky je však mnohem větší. Počítače nám umožňují provádění složitých výpočtů, různých simulací od štěpení atomů po výbuch jaderné bomby, také pomocí nich můžeme provádět různé návrhy, plánování a mnoho dalších věcí.

Počítače vše usnadnily. Tam kde bylo dříve potřeba spousta lidí s různými nástroji pro provedení konkrétní činnosti, nám dnes mnohdy postačí jeden či pár lidí s počítačem.

Příkladem může být samotný návrh, výroba a prezentace různých předmětů. Dříve byli lidé nuceni složitě navrhnout konkrétní předmět, zhotovit jeho plány a na základě těchto plánů tento předmět vyrobit. Poté jej bylo nutné nějak prezentovat a představit veřejnosti. K tomu mohly sloužit veletrhy či různé výstavy, kde byl daný předmět vystaven. V dnešní době, kdy se stále více rozvíjí počítačová grafika, je velmi jednoduché udělat různé trojrozměrné modely budoucích výrobků, a z těchto modelů jednoduše vytvořit plány a tyto plány nebo dokonce i samotný trojrozměrný model použít k výrobě. Díky internetu je i prezentace těchto výrobků velmi jednoduchá. Kdejaká firma má v dnešní době internetové stránky, které slouží k prezentaci jednak samotné firmy, ale i k prezentaci výrobků. Na tyto stránky lze snadno umístit trojrozměrný model a lidé si mohou tento model libovolně prohlédnout a udělat si tak představu o samotném předmětu.

Co si však můžeme pod pojmem trojrozměrný (též trojdimenzionální či zkráceně 3D) model představit? Jedná se o model, který je možné popsat třemi rozmě-

ry. Když si vezmeme do ruky nějaký předmět, můžeme si ho prohlédnout ze všech stran, libovolně si ho natočit a případně zkusit jeho funkčnost. Při tvorbě trojrozměrných modelů je pro nás důležité zachytit a poté zobrazit zachycované předměty na dvojrozměrném zařízení tak, jako bychom tyto předměty drželi v ruce.

To je hlavní rozdíl mezi dvojrozměrným a trojrozměrným zobrazením. Při dvojrozměrném zachycení dané věci (např.: vyfocení předmětu) si můžeme danou věc prohlédnout pouze tak, jak jsme ji v danou chvíli zachytili. Nemůžeme si ji prohlédnout z jiné strany, ani si ji nemůžeme libovolně natočit.

Pokud trojrozměrný model doplníme o prvky, které budou reagovat nějakou událostí na činnost uživatele (např. na kliknutí myši či zobrazení určitého prvku), a uživatel bude moci tento model libovolně zobrazovat a procházet v reálném čase, můžeme o daném modelu říct, že se jedná o trojrozměrný interaktivní model nebo o virtuální realitu.

V dnešní době se s interaktivními prvky můžeme setkat téměř kdekoliv. Může se jednat o již zmíněné interaktivní trojrozměrné modely nebo o různé dvojrozměrné modely či informativní tabule nebo o interaktivní výukové materiály. Příkladem trojrozměrného interaktivního modelu může být model části města, místa či nějakého areálu. Zástupcem dvojrozměrného interaktivního modelu může být informativní tabule v nákupním centru nebo u nějaké historické památky s popisem daného objektu. Tyto tabule bývají často doplněné i o zvukový popis. V dnešní době existují i různé interaktivní výukové materiály. Může se jednat o modely nebo různé aplikace doplněné o interaktivní prvky, jako jsou zvukový popis a animace, nebo tyto aplikace mohou zapojit uživatele do samotné výuky například pomocí přesouvání prvků, jejich skládáním či propojováním.

1.2 Cíl práce

Cílem práce bude tvorba statického trojrozměrného modelu vybraných částí počítače, animace tohoto modelu a doplnění vhodné formy interaktivity.

Nejprve bude nutné získat poznatky o tvorbě statických a interaktivních trojrozměrných modelů a navrhnout vhodné metody a postupy pro samotné vytvoření a prezentaci modelů počítačových komponent.

Dále bude potřeba provést analýzu současného stavu využití trojrozměrných modelů k prezentaci počítačových komponent a dostupnost těchto modelů se zaměřením na jejich využití.

Nejdůležitější částí práce bude vytvoření trojrozměrných modelů počítačových komponent podle nashromážděných materiálů. Poté bude nutné jednotlivé modely spojit v sestavu. Pro výsledný model bude zapotřebí navrhnout a vytvořit ovládací rozhraní. Jelikož by měl model sloužit k výukovým účelům, bude ještě nutné zvolit vhodné interaktivní prvky, kterými bude model doplněn, a doplnit animaci.

V samotném závěru práce bude zhodnoceno výsledné řešení a bude navrženo, co by bylo možné do modelu doplnit či jak jinak jej využít.

2 Možnosti tvorby trojrozměrných modelů

Na začátku samotné práce bylo nutné zodpovědět spoustu otázek týkajících se jednotlivých postupů a metod pro tvorbu modelů a také to, který program bude vhodné použít pro modelování. Mimo to bylo potřeba rozhodnout, jakou formou bude výsledný model distribuován a jak bude dostupný.

Jako nejjednodušší řešení tvorby modelů se nabízí použít některý z *3D grafických programů*, kterých v dnešní době existuje velké množství. Další možností, jak vytvořit a pracovat s trojrozměrným modelem, je *využít* některý *programovací jazyk* s případnou kombinací s některou z jeho rozšiřujících knihoven.

2.1 Použití 3D grafických programů

Jednou z možností, jak vytvořit trojrozměrný model, je využít některý z *3D grafických programů*. Může se jednat o profesionální programy se spoustou různých knihoven, které běžný uživatel ani nevyužije, jejichž cena se pohybuje řádově až v desítkách tisíc korun. Případně lze využít zkušební verze těchto programů. Obvykle však tyto verze fungují pouze 30 dnů a mají velmi omezené funkce. Dále existují i programy pro tvorbu trojrozměrných modelů, které jsou zcela zdarma či dokonce open source. Pomocí těchto programů můžeme snadno vytvořit dobře vypadající a propracované modely. Tyto modely však mohou zabírat velké množství místa. Dále se u 3D grafických programů můžeme setkat s problémem, že mnoho z nich využívá své vlastní přípony, strukturu a kódování souborů, které jsou vázány právě na tyto programy. To může být problémem při zobrazování a distribuci hotových modelů.

2.1.1 Program Rhinoceros

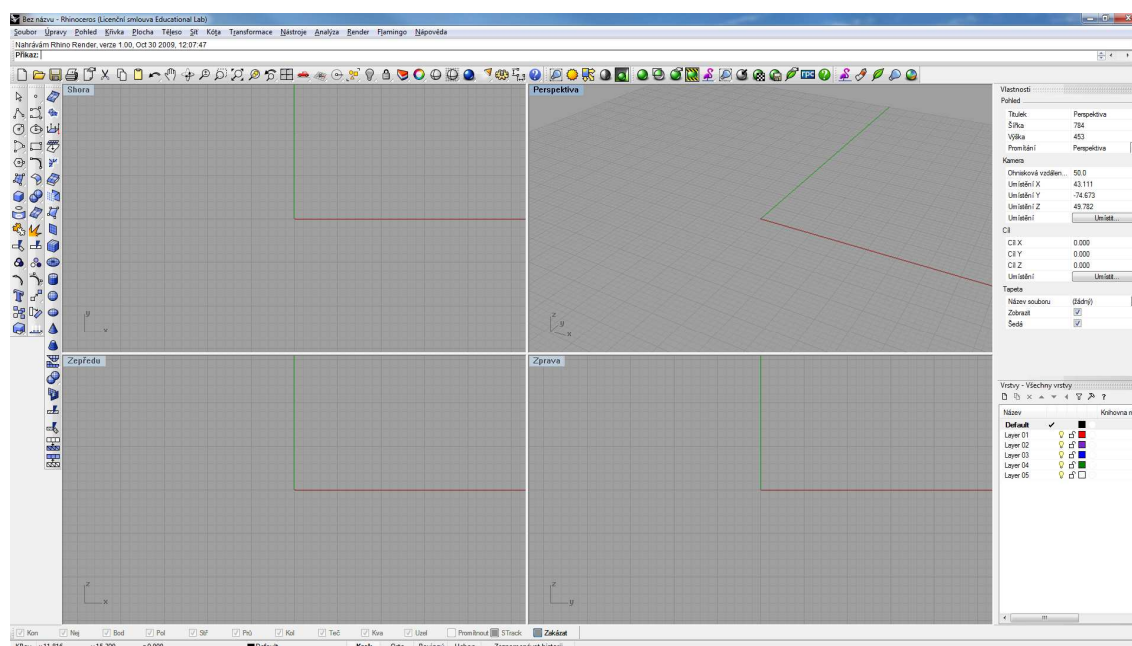
Rhinoceros, často také nazývaný zkráceně *Rhino*, je 3D modelář, který pracuje s křivkami a tělesy. Aktuální verzí tohoto programu je verze 5. Pomocí tohoto programu lze velmi snadno a přesně vymodelovat jakýkoliv tvar. Rhino disponuje přehledným rozhraním, rychlým stínováním a mnoha modelovacími funkcemi, které činí modelování v 3D prostoru intuitivním a přirozeným. Rhino navíc dokáže přečíst a zapisovat soubory většiny programů pro design, 2D kreslení, animaci a ilustraci (McNeel, 1993).

Rhino je založené na *modelování pomocí NURBS* (neuniformní racionální B-spliny), což je matematická reprezentace 3D geometrie, která umožňuje přesně popsat jakýkoliv tvar od jednoduché 2D čáry, oblouku, kružnice nebo křivky až po nejsložitější 3D plochy a tělesa volných tvarů. Díky jejich všestrannosti a přesnosti lze využívat NURBS modely v libovolném procesu od ilustrace po animaci nebo sériovou výrobu (McNeel, 1993).

Jak je uvedeno v uživatelské příručce (McNeel, 1993), pomocí Rhina lze vy-modelovat téměř cokoliv. Na velikosti modelovaných předmětů nezáleží. Rhino je využíván při designu lodí, letadel, automobilů, interiérů, nábytku, obuvi či sportovního vybavení. Rhino disponuje rychlým pracovním prostředím. Modelování v Rhinu také přináší mnoho výhod. Mezi tyto výhody můžeme zařadit:

- Rychlou tvorbu ploch volného tvaru.
- Rychlejší modelovací postupy než u polygonových modelářů.
- Z modelu z Rhina lze vytvořit libovolně husté polygonové sítě.
- Import mnoha typů obrazových souborů.
- Export do většiny animačních produktů.
- Import dat z 2D a 3D CAD software.
- Import vektorových souborů z 2D ilustračního software.

Uživatel si může prostředí v Rhinu upravit dle libosti. Libovolně si může poskládat nástrojové lišty, zobrazit různé typy lišt dle potřeby nebo dokonce vytvořit vlastní lišty pouze s nástroji, které využívá. Také si může zobrazit různé množství pohledů a tyto pohledy v rámci okna různě uspořádat. Pohledům lze také nastavit různé vlastnosti, jako jsou možnosti zobrazení, volba pozadí, hustota mřížky či umístění kamery. Dobře přizpůsobené prostředí může vést ke snadné, rychlé a hlavně efektivní práci s tímto programem. Základní rozložení okna lze vidět na obrázku níže.



Obr. 1 Rhinoceros

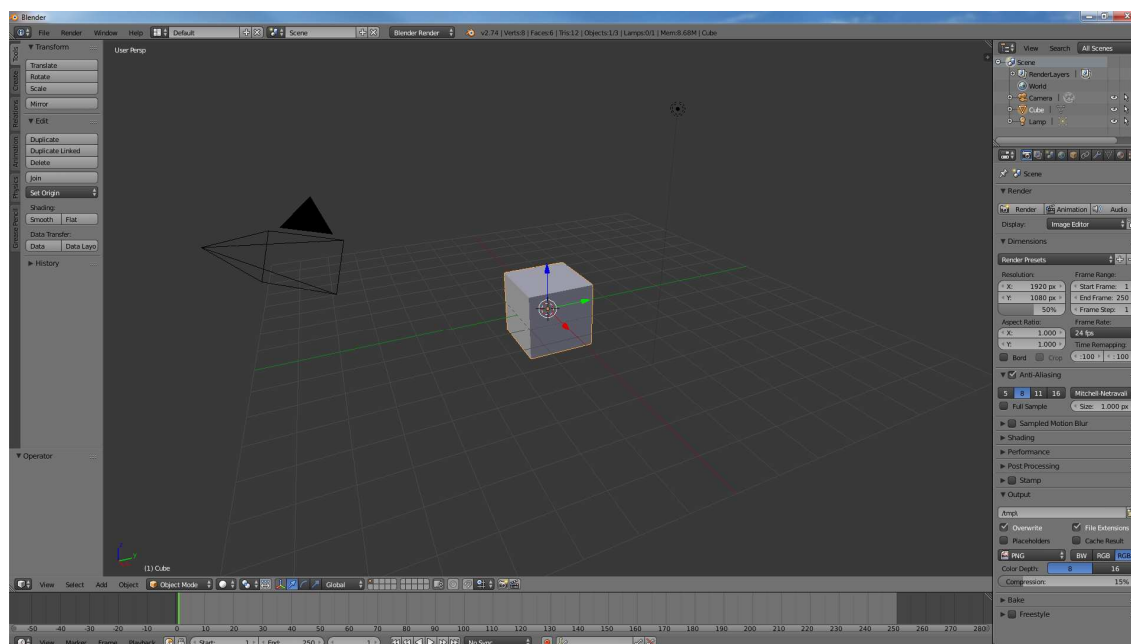
Rhino lze dále rozšířit o různé moduly. Může se jednat o modul s různými knihovnami textur, rostlin, renderovací modul či různé animační moduly.

2.1.2 Program Blender

Blender je software pro 3D modelování, animaci, tvorbu her, rendering a přehrávání. Poslední verze Blenderu, kterou je verze 2.74, byla vydána na konci března roku 2015. Tento program je založen na grafické knihovně OpenGL. Díky tomu je dostupný pro většinu operačních systémů. Další velkou výhodou Blenderu je, že je *k dispozici zcela zdarma* a to včetně zdrojových kódů.

V dnešní době existuje velké množství programů pro 3D modelování a není možné říct, který z nich je nejlepší. Každý z nich má své výhody i nevýhody. Navíc každému uživateli vyhovuje něco jiného. Největší výhodou Blenderu oproti ostatním programům je, jak už bylo zmíněno, že je zcela zdarma. Tím pádem se jeví jako vhodný nástroj pro začínající zájemce o 3D grafiku, aniž by museli do programu investovat finanční prostředky. Další velkou výhodou Blenderu je jeho velikost a malá hardwarová náročnost (Pokorný, 2009). Velikost instalačního souboru aktuální verze je okolo 60 MB a pro svou instalaci vyžaduje okolo 270 MB volného místa na disku.

Blender disponuje plně nastavitelným uživatelským prostředím, které je pro všechny operační systémy stejné. Uživatel si může vytvořit libovolný počet nepřekrývajících se oken. Každé z těchto oken je plně konfigurovatelné v závislosti na jeho typu (Pokorný, 2009). Rozložení uživatelského prostředí je znázorněno na následujícím obrázku.



Obr. 2 Blender

Blender ukládá veškerá data do jediného souboru. Dále umožňuje číst a zapisovat velké množství formátů, jejichž množství lze ještě více rozšířit pomocí Python skriptů. Mimo to umožňuje vytvářet spustitelné soubory s interaktivními 3D apli-

kacemi nebo umožňuje tyto aplikace přehrávat v internetových prohlížečích s nainstalovaným příslušným zásuvným modulem (Pokorný, 2009).

Tak jako každý program má i Blender své nevýhody. Jednou takovou nevýhodou může být pro někoho fakt, že prostředí programu je ryze anglické. Navíc začátečníkům se toto prostředí může jevit mírně chaotické a mohou mít problém se v tomto prostředí zorientovat. Z tohoto důvodu jim může déle trvat, než se s programem naučí pracovat. To se týká především uživatelů Windows. Příkladem může být rozdílné použití funkční klávesy F1. V prostředí Windows slouží k vyvolání nápovědy. To však neplatí v Blenderu. Zde slouží tato klávesa k vyvolání okna pro otevření souboru. To však není jediný značný rozdíl. Tento způsob ovládní je totiž dán tím, že Blender je vyvíjen, jak už bylo zmíněno, pro většinu operačních systémů (Pokorný, 2009).

2.1.3 Program SketchUp

Program *SketchUp* je 3D modelář, který svým jednoduchým a intuitivním ovládním umožňuje neobyčejně rychle graficky vyjádřit představu uživatele. Není orientován na žádný konkrétní obor. Své uplatnění najde téměř kdekoliv. Ovládní programu lze uživatelsky volně doplňovat o další funkce. Určitým omezením programu jsou jeho slabší možnosti vytvořit profesionální výrobní výkresy. Pro vytvoření 3D návrhu modelů a pro následné vyjádření pomocí přesných 2D obrázků, šablon a výkresů v něm však nic nechybí (Mikač, 2008).

První verze programu SketchUp byla uvedena na trh v roce 2000. Tehdy byl vyvinut společností *@Last Software*. V roce 2006 však tuto firmu kupuje *Google* a SketchUp je dále vyvíjen jako zásuvný modul pro *Google Earth*. Postupem času prošel program spoustou úprav a vylepšení. V roce 2012 byl SketchUp prodán společnosti *Trimble Navigation*, která ho vyvíjí do dnešní doby (Sikdar, 2014).

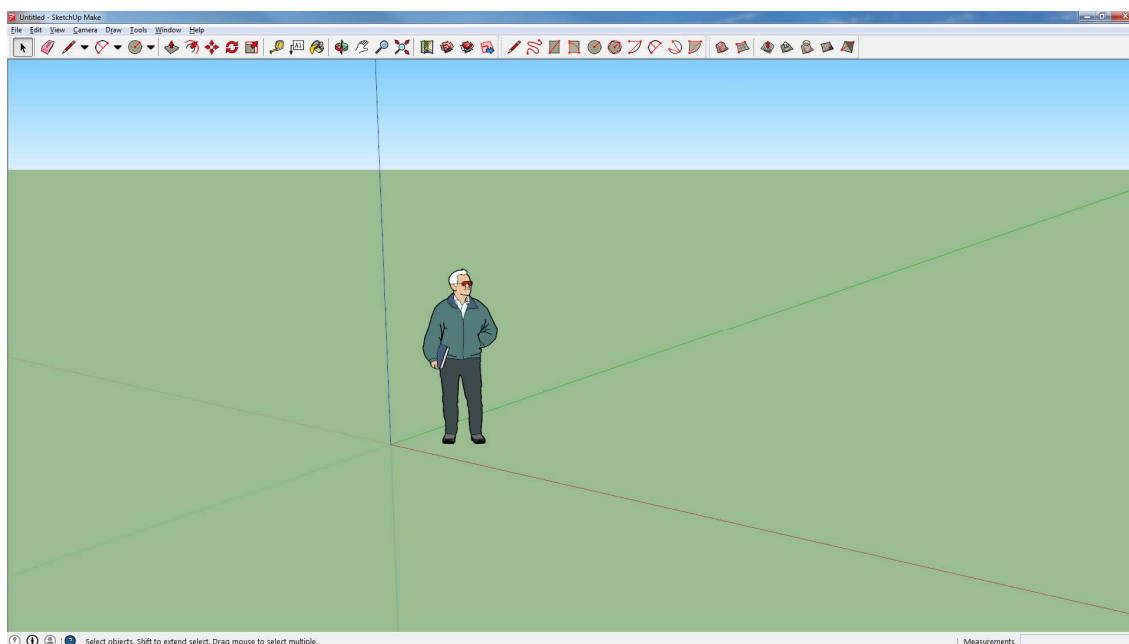
SketchUp byl vyvíjen jako volně šiřitelný software, což platí i v dnešní době. Mimo placených verzí lze stáhnout i základní verzi zcela zdarma.

Velkou výhodou programu je jeho *3D Web Warehouse*, kam uživatel může bezplatně nahrávat své modely. V případě, že uživatel potřebuje rychle nějaký model, zde může najít různé modely ostatních uživatelů, které si může zdarma stáhnout a použít při své tvorbě.

SketchUp umožňuje zobrazovat i zapisovat soubory jiných 3D programů. Při práci v něm lze také použít načtení obrázku jako předlohy pro samotnou tvorbu modelu.

Základní rozložení uživatelského okna je velice jednoduché a pro běžnou práci nedostačující. Uživatel je tak nucen si toto prostředí nastavit sám. Přinejmenším je nutné, aby si uživatel zobrazil další panely s modelovacími nástroji, případně aby si mírně nastavil modelovací okno. Zmíněné základní rozložení okna lze vidět na obrázku 3 níže.

Tento program má také své nedostatky. S 3D grafikou dokáže pracovat omezeně a využívat pouze základní prvky pro modelování. Postrádá mnoho důležitých pokročilých funkcí z řady svých větších konkurentů. V bezplatné verzi má značná omezení co se týká exportu a importu souborů (Softonic, 2015).



Obr. 3 SketchUp

2.1.4 Ostatní programy

Tři výše zmíněné programy byly popsány podrobněji z důvodů jejich dostupnosti, případně jsem s některými z nich měl osobní zkušenost. Mimo tří výše zmíněných a popsaných programů se můžeme setkat i s mnoha dalšími programy, které mohou být stejně kvalitní ba i lepší. Těmito programy mohou být *Autodesk Maya*, *Cinema 4D*, *3D Studio Max*. Náplní této práce však není porovnávat všechny programy pro tvorbu trojrozměrných modelů, proto nebudou tyto programy popsány podrobněji.

2.1.5 Výběr vhodného 3D grafického programu

Výběr vhodného programu bude probíhat ze tří výše popsaných programů. Jsou to programy:

- Rhinoceros
- Blender
- SketchUp.

Mezi hodnocenými programy jsou v mnohých ohledech velké rozdíly. Některé obsahují důležité pokročilé funkce a jiné zase pouze funkce základní. Některé lze získat zdarma a za jiné je nutné zaplatit. Také se velice liší jejich uživatelská přívětivost a přehlednost. Proto je nutné stanovit vhodná kritéria, podle kterých budou tyto programy hodnoceny. Při výběru kritérií a jejich váhy vždy záleží na pohledu daného jedince a na tom, co je pro něj nejdůležitější. Proto není možné vybrat

vhodné řešení pro všechny. Nejdůležitějšími kritérii pro porovnání těchto programů jsou:

- Dostupnost
- Přehlednost prostředí
- Možnosti úpravy prostředí
- Intuitivní a jednoduché ovládání
- Podpora exportu a importu různých formátů
- Vybavenost

V tomto případě jsou všechna kritéria stejně důležitá, proto nebudou při porovnávání žádná kritéria zvýhodňována. Díky tomu je možné k hodnocení těchto kritérií zvolit metodu známkování, kde každé kritérium bude ohodnoceno hodnotou 1 až 5, kde 1 je nejlepší možné hodnocení a 5 je nejhorší hodnocení. Navíc díky tomuto způsobu hodnocení stačí pouze jednotlivé známky u každého programu sečíst. Neří nutně nic složitě přepočítávat. Poté stačí vybrat jen nejvhodnější řešení. V tomto případě bude tím nejvhodnějším řešením to, které bude mít nejnižší součet hodnot. Ohodnocení kritérií pro jednotlivé programy lze vidět v následující tabulce.

Tab. 1 Hodnocení programů dle stanovených kritérií

	Rhinoceros	Blender	SketchUp
Dostupnost	3	1	1
Přehlednost prostředí	1	3	1
Možnosti úpravy prostředí	1	1	4
Intuitivní a jednoduché ovládání	2	3	1
Podpora exportu a importu různých formátů	1	1	3
Vybavenost	1	2	5
Součet	9	11	15

Z tabulky lze zjistit, že jako *nejlepší program*, z výše porovnávaných programů, je stanoven *Rhinoceros*, který měl součet hodnot jednotlivých kritérií 9. Na druhém místě je Blender se součtem 11 a jako nejhorší z hodnocených programů byl stanoven SketchUp s celkovým součtem 15.

2.2 Použití programovacích jazyků

K tvorbě trojrozměrných modelů lze použít i různé *programovací jazyky*. Při výběru tohoto řešení se však můžeme setkat s celou řadou nedostatků. Většinou musíme popsat dané objekty pomocí příkazů daného programovacího jazyka. To může vést k velkému množství programového kódu a k jeho nepřehlednosti. Tento způsob je také velmi náchylný na chyby, které mohou vzniknout i nepozorností. Odha-

lit takovou chybu pak mnohdy může být velmi složitě. Další nedostatek této metody může být i to, že obvykle touto metodou dokážeme pracovat pouze se základními geometrickými tělesy, jako je krychle, koule, jehlan a válec, a kombinací těchto těles vytvářet složitější objekty. Tvar objektu lze také popsat pomocí jeho polygonové sítě. Tato metoda spočívá v reprezentaci povrchu tělesa pomocí trojúhelníkové sítě, což může být mnohdy složitě a vést k chybné reprezentaci objektu.

2.2.1 Jazyk VRML

VRML (Virtual Reality Modeling Language) je jazyk, který dovoluje popsat skutečný svět a jeho objekty téměř stejně věrohodně, jako ho vnímají lidské smysly. O takovém světě pak mluvíme jako o virtuálním světě. V tomto světě se můžeme volně pohybovat, prohlížet věci ze všech stran, poslouchat skutečné zvuky, přemísťovat se z jednoho světa do druhého nebo pomocí tlačítek měnit tvary předmětů (Zrzavý, 1999).

Takový svět pak lze distribuovat pomocí internetu a k jeho prohlížení stačí internetový prohlížeč doplněný o zásuvný modul. Většina těchto modulů je volně přístupná na internetu a některé z nich lze využít i na více operačních systémech. V dnešní době mezi nejpoužívanější zásuvné moduly pro zobrazení jazyka VRML patří Cortona3D Viewer a Cosmo Player (Zrzavý, 1999).

Jazyk VRML definuje způsob zápisu virtuálních světů do souboru v textovém tvaru, což je jednou z výhod tohoto zápisu. Soubor s popisem scény lze otevřít pomocí běžného textového editoru a kdykoliv tuto scénu změnit.

Virtuální svět lze pomocí jazyka VRML rozšířit o různé multimediální prvky, jako je obraz, video a zvuk. Statické světy lze dále jednoduše rozšířit o různé animace či interakci a tím z nich vytvořit světy dynamické. Jazyk VRML také umožňuje spolupráci s dalšími programovacími jazyky, jako je jazyk Java nebo JavaScript.

Jednou z velkých výhod jazyka VRML je možnost vytváření *prototypů*. Prototyp je pojmenovaná část kódu. Obsah prototypu definujeme pouze jednou a poté ho můžeme libovolně používat. Je to obdoba funkcí a procedur z běžných programovacích jazyků. Vhodným použitím prototypů můžeme značně snížit paměťovou náročnost celého světa.

Použití jazyka VRML má také své nevýhody. Nejdůležitější nevýhody byly obecně zmíněny při popisu využití programovacích jazyků.

2.2.2 Jazyk X3D

X3D (Extensible 3D) slouží k popisu plošných i prostorových scén. Tento formát je mnohdy označován jako *nástupce jazyka VRML*, ze kterého také vychází. Doplnuje některé nedostatky jazyka VRML a umožňuje zápis v podobě původní syntaxe, ale i zápis pomocí XML (Extensible Markup Language). Zápis pomocí XML přináší mnoho výhod. Těmito výhodami je možnost využívat knihovny a další objekty pro práci s XML (Tišnovský, 2008).

Jak již bylo zmíněno, pro zápis scény je možné použít původní syntaxi VRML nebo syntaxi jazyka XML. Rozdíl mezi oběma způsoby zápisu je sice na první po-

hled značný, ale z praktického hlediska jsou si obě syntaxe rovnocenné. Jakoukoliv scénu dokážeme popsat pomocí kterékoliv syntaxe. Příklady obou syntaxí jsou znázorněny v krátkém příkladu níže. V příkladu je popsána jednoduchá scéna, ve které je umístěn zelený válec. Nejprve bude tato scéna popsána pomocí VRML syntaxe a poté pomocí XML. Na první pohled je vidět, že klíčová slova jsou stejná. Zápis se liší pouze v provedení (Tišnovský, 2008).

```
# zápis pomocí syntaxe jazyka VRML
Transform {
  children Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0 1 0
      }
    }
  }
  geometry {
    Cylinder {
      height 0.1
      radius 0.5
    }
  }
}

# zápis pomocí syntaxe XML
<Transform>
  <Shape>
    <Appearance>
      <Material diffuseColor="0 1 0"/>
    </Appearance>
    <Cylinder height="0.1" radius="0.5"/>
  </Shape>
</Transform>
```

Z ukázky jednotlivých syntaxí si můžeme všimnout, že zápis pomocí XML je poněkud elegantnější a přehlednější. Na první pohled vidíme, která uzavírací závorka patří k čemu, protože součástí této závorky je i název daného tagu. V případě syntaxe pomocí VRML slouží k uzavření daného uzlu vždy pouze uzavírací složená závorka. V případě, jako je v dané ukázce, že máme těchto složených závorek za sebou více, je mnohdy na první pohled těžké rozeznat, která uzavírá který uzel.

I když je formát X3D v mnohých ohledech lepší a oproti samotnému VRML má mnoho výhod, není však v dnešní době ještě příliš rozšířen a stále se častěji setkáváme pouze s klasickým jazykem VRML. Důvodů může být více, ale hlavním důvodem asi pravděpodobně bude, že k samotnému VRML lze nalézt mnohem více materiálů, ať už se jedná o články na internetu, různé manuály, vzorové příklady, dokumentaci nebo samotné knihy. Materiálů týkajících se samotného X3D je stále strašně málo.

2.3 Výběr vhodné metody pro tvorbu modelů

Hlavní náplní této práce je vytvoření trojrozměrných modelů počítačových komponent. Spojení těchto modelů do jedné sestavy a doplnění této sestavy vhodnými interaktivními prvky a animací. Mimo to je nutné zajistit snadnou distribuci a jednoduché zobrazení této výsledné aplikace. Z těchto důvodů je nutné vybrat vhodný postup pro tvorbu modelů a vhodné metody pro jejich další rozšíření a distribuci.

Jak již bylo zmíněno, pro tvorbu trojrozměrných modelů lze využít některý 3D grafický program a v tomto programu tyto části vymodelovat. Také lze jednotlivé části popsat pomocí zdrojového kódu nějakého jazyka. Obě metody mají své výhody, ale často mají také mnoho nevýhod. Výhody a nevýhody obou metod již byly zmíněny v kapitole možnosti tvorby trojrozměrných modelů.

I přes velké množství zjištěných a popsaných nevýhod však převažovaly výhody jazyka VRML. Mezi tyto důležité výhody jazyka VRML byly zařazeny:

- Velké množství knih, návodů a jiných materiálů.
- Možnost jednoduchého popsání a úpravy objektů bez nutnosti speciálního programu.
- Jednoduchý popis základních objektů.
- Možnost vytváření prototypů.
- Možnost doplnit model o interaktivní prvky.
- Jednoduchá tvorba animací modelu.
- Možnost zobrazení pomocí internetového prohlížeče doplněného o zásuvný modul.

Jazyk VRML má však i některé nedostatky a jistá omezení, co se možností popisu objektů týká, a to hlavně při tvorbě složitých objektů. Proto bylo nutné zvolit vhodnou metodu pro jejich tvorbu. Z tohoto důvodu byl k tvorbě složitějších ploch či alespoň k tvorbě částí těchto ploch použitý program Rhinoceros, který byl ze všech výše hodnocených programů hodnocený jako nejlepší.

Pomocí tohoto programu byly vymodelovány různé složité a zaoblené plochy, které by bylo velmi obtížné a zdlouhavé pomocí jazyka VRML popsat. Tyto vymodelované části byly dále převedeny na polygonovou síť. Při tomto převodu bylo nutné dbát na dostatečné množství těchto polygonů tak, aby byly zachované potřebné detaily a aby model nevypadl moc uměle. Poté byly tyto modely vyexportovány do jazyka VRML a dále tyto části byly začleněny do jednotlivých modelů.

Při vhodné kombinaci jazyka VRML s možnostmi programu Rhinoceros lze vytvořit velmi propracované a dobře vypadající modely. Dále je tyto modely možné díky možnostem jazyka VRML, velmi jednoduše doplnit o animaci a interaktivní prvky. Navíc při vhodném používání prototypů lze popsat i složité objekty tak, aby výsledný model nebyl moc náročný na paměť. Díky tomu lze modely snadno a rychle zobrazovat pomocí internetového prohlížeče a to i při jejich načítání z internetu.

3 Možnosti zobrazení modelů

Jak již bylo zmíněno několikrát, vytvoření kvalitních trojrozměrných modelů je zdlouhavý a mnohdy náročný proces. Vytvořit trojrozměrný model je jedna věc, dále však nastává otázka, jak tento model dále prezentovat a zobrazit tak, aby byl co nejlépe dostupný.

3.1 Zobrazení na internetu

Jako nejvhodnější prostředek ke globální propagaci se nabízí možnost, využít internet. V dnešní době, kdy existuje spousta metod, jak prezentovat cokoli na internetu, se tato metoda přímo nabízí. Lze využít některou z nynějších webových technologií pro tvorbu interaktivních webových aplikací nebo stačí pouze použít zásuvný modul pro webový prohlížeč. Nejdůležitější ovšem je, aby si uživatel nemusel složitě instalovat různé, mnohdy i placené, programy, které umožňují tyto modely zobrazit. To může uživatele mnohdy odradit. Nejvhodnější je si vystačit pouze s možnostmi internetových prohlížečů.

Pro tvorbu těchto modelů je vhodné použít již zmíněný jazyk VRML nebo některý jiný, který umožňuje zobrazení pomocí zásuvného modulu v prohlížeči. Prezentace těchto modelů je poté velmi jednoduchá.

Takový model však musí být pro tuto formu prezentace přizpůsoben. U modelů prezentovaných pomocí internetu není žádoucí, aby jejich velikost byla velká. Musíme brát v potaz, že prohlížeč tento model musí nejprve načíst. Doba načtení může být různá. Vždy záleží na tom, jak rychlé připojení k internetu daný uživatel má. Proto je pro tuto metodu důležité, aby prezentovaný model byl možná co nejmenší. Mnohdy je také důležité, aby byl model co nejjednodušší a aby neobsahoval velké množství překrývajících se ploch. Jednotlivé programy mohou mít se zobrazením takovýchto ploch problémy. Navíc výpočet a zobrazení toho, co uživatel zrovna vidí, může být s rostoucím množstvím ploch zdlouhavé.

3.2 Zobrazení pomocí programu SketchUp

Jak již bylo zmíněno výše, program SketchUp slouží jednak v tvorbě trojrozměrných modelů, ale i k jejich zobrazení a distribuci. Program SketchUp obsahuje 3D Web Warehouse, kam může kterýkoliv uživatel zdarma ukládat své modely a tím je dále šířit. Jiní uživatelé mohou tyto modely zdarma stahovat a využívat přímo v tomto programu. K tomuto využití každému uživateli vystačí základní verze, která je ke stažení zdarma.

V 3D Web Warehouse lze najít modely téměř čehokoliv. Může se jednat o modely různých významných staveb, interiérů budov, modely letadel, lodí, různého průmyslového vybavení, modely rostlin, modely různých malých předmětů a další. Některé tyto modely jsou vytvořeny velmi detailně a hezky. Častěji se však můžeme setkat s modely, které jsou velmi jednoduché a působí dosti uměle.

Nevýhodou této metody ovšem je, že si uživatel tento program musí stáhnout a nainstalovat do počítače a pro vyhledávání modelů musí být samozřejmě připojen k internetu. Samotné vyhledávání je navíc mnohdy pomalé a déle trvá, než se modely v 3D Web Warehouse načtou. Jelikož sem může modely ukládat kdokoli, je jich zde velké množství, ale mnohé z nich jsou velmi jednoduché a nepoužitelné. Při základním zobrazení před stáhnutím vidíme pouze miniatury těchto modelů, ze kterých však často nepoznáme, zda je tento model kvalitní nebo ne. Náповědou pro nás může být jeho velikost či počet použitých polygonů. Není však pravidlem, že čím větší má model velikost a čím více polygonů obsahuje, že je lepší. Proto je uživatel mnohdy nucen stáhnout a prohlédnout velké množství modelů, než nalezne nějaký, který mu vyhovuje.

Jak již bylo uvedeno dříve, modely sem může ukládat kdokoli a také je může kterýkoliv uživatel stáhnout. Proto sem umístit model s tím, aby si jej cíleně stáhl pouze určitý uživatel, je nesmysl. Tento náš model si totiž může zobrazit a stáhnout kdokoli a nijak neovlivníme, aby si jej stáhl pouze jeden konkrétní uživatel.

3.3 Zobrazení pomocí 3D grafických programů

Další možnou cestou, jak vytvořené modely šířit, je sdílením jejich originálních souborů na internetu. Pro zobrazení je pak nutné mít nainstalovaný program, ve kterém byly modely vytvořeny nebo alespoň nějaký dostupný prohlížeč, který umožňuje tyto modely zobrazit.

V případě této metody šíření však často do poslední chvíle nevíme, co přesně stahujeme a do jaké míry je model detailní. Originální soubor těchto modelů totiž pomocí internetového prohlížeče nezobrazíme a jsme pouze odkázáni na obrázky přiložené k danému modelu, pokud však k němu nějaké obrázky přiloženy jsou. Navíc tyto obrázky mnohdy zobrazují model jako celek a proto nedokážeme odhadnout jakou kvalitu tento model má.

Sdílení touto cestou je pohodlné pouze pro vlastníka modelu, který ho dává k dispozici. Zájemce, který si chce model prohlédnout, si však musí stáhnout a nainstalovat některý z programů pro zobrazení daného modelu. Dále je důležité, aby pochopil, jak aplikace funguje a jak může daný model v této aplikaci procházet. Nakonec musí stáhnout i samotný soubor s modelem, jehož velikost může být mnohdy velká.

4 Současný stav

V dnešní době se s trojrozměrnými modely setkáváme pořád častěji. Možnosti využití trojrozměrných modelů jsou různé. Lze je použít při návrhu výrobků, při strojírenské výrobě, k prezentaci nových výrobků, k výuce, v architektuře a v mnoha dalších oblastech lidské činnosti.

4.1 Využití trojrozměrných modelů

Trojrozměrné modely lze využít mnoha způsoby. Nemusí se vždy jednat pouze o prezentaci znázorněného zboží. Níže budou popsány některé možnosti využití těchto modelů.

4.1.1 Prototypování

Za prototyp je považován jakýkoliv vzorový výrobek. Místo tohoto vzorového výrobku může být vytvořen trojrozměrný model, který daný výrobek znázorňuje. Používání takovýchto prototypů má mnoho výhod.

Použitím trojrozměrných modelů můžeme snadno popsat výsledný objekt a to včetně mnoha jeho vlastností a využitých materiálů. Model lze dále snadno a rychle upravovat a vylepšovat. Díky tomu během krátké doby vidíme, jak bude budoucí předmět vypadat a jaké bude mít vlastnosti. Na základě toho můžeme určit, zda je předmět již možné vyrobit nebo zda jej bude potřeba ještě upravovat. Když máme hotový model, tak můžeme na základě tohoto modelu vytvořit podrobné technické výkresy, které lze použít při výrobě.

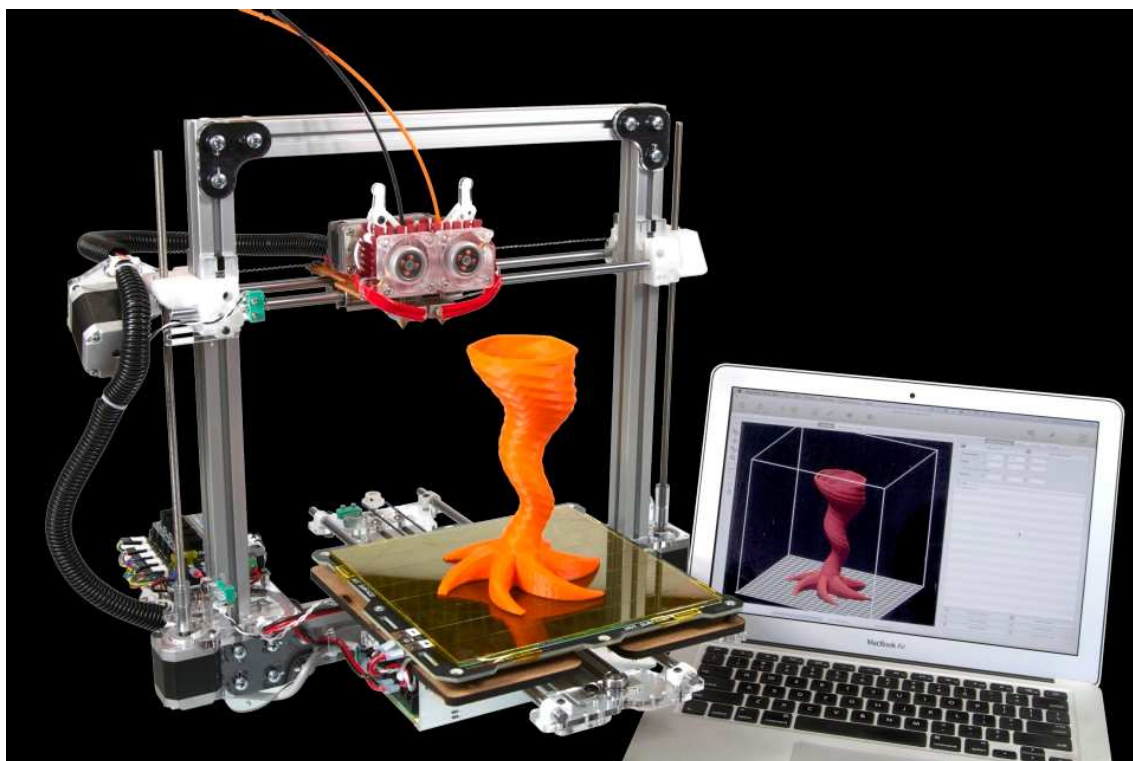
Kdybychom daný předmět nevyamodelovali a začali rovnou s výrobou, tak bychom po zhotovení výrobku mohli zjistit, že to není přesně to, co jsme si představovali a bylo by nutné provést změny a tento předmět vyrobit znovu. V případě některých výrobků může být jejich vytvoření velmi zdoluhavý proces a navíc k této výrobě musíme použít potřebný materiál. Proto by se nám v takovýchto případech výroba, bez vytváření předlohových modelů, mohla prodrazit. V případě vytvoření modelů se tyto nesrovnalosti snadno upraví již při prototypování na modelu a poté se teprve začne s výrobou.

4.1.2 3D tiskárny

V dnešní době se stále častěji můžeme setkat s možnostmi trojrozměrného tisku. Tato technologie se stále vyvíjí a umožňuje vytvářet přesné a detailní trojrozměrné předměty. Proces výroby spočívá v nanášení velmi tenkých vrstev obvykle roztaženého plastu podle určité předlohy.

Předlohou pro tvorbu těchto objektů pomocí 3D tiskáren obvykle bývají přímo trojrozměrné modely, znázorňující daný objekt. Tyto modely lze přímo vytvořit nebo nahrát do speciálního programu, který tuto 3D tiskárnu řídí. Tento program rozloží daný objekt do jednotlivých miniaturních vrstev, které jsou poté při tisku nanášeny postupně na sebe.

Na následujícím obrázku je znázorněna 3D tiskárna Bukobot V2 (Garo3D, 2015) s vyhotoveným objektem podle znázorněné předlohy.



Obr. 4 Ukázka 3D tisku

4.1.3 Průmyslová výroba

V dnešní době se často trojrozměrné modely vytváří právě před výrobou různých předmětů a slouží k určení výrobních postupů a k vytvoření technických výkresů, podle kterých se dále tyto předměty vyrábí.

Ve strojírenské výrobě se můžeme setkat s CNC stroji, které umožňují vyrábět předměty na základě vytvořeného trojrozměrného modelu. Tyto CNC stroje často fungují pod nějakým běžným operačním systémem pro počítače a umožňují přímo vytvoření trojrozměrných modelů pomocí připojených zařízení nebo jejich nahrání z externího úložiště. Dále obsahují různé algoritmy, které umožňují převedení těchto modelů do příkazů programu pro samotnou výrobu.

Se zpracováním podle modelů se můžeme setkat také při gravírování. Gravírování je technologie, která nahrazuje ruční rytí. Tato metoda umožňuje vytváření různých nápisů, log a ornamentů díky odebrání materiálu obvykle tenkou, ale pevnou jehlou. Předlohou pro tento proces často bývá nějaký ať už dvojrozměrný nebo trojrozměrný model, který je pomocí speciálního programu zpracován a výstupem tohoto programu jsou data pro samotnou gravírku.

Pomocí gravírování lze zpracovávat různé materiály. S produkty gravírování se můžeme setkat každý den. Může se jednat o různé informační tabule, pamětní

desky, medaile, upomínkové předměty, šperky a mnoho dalších. Na následujícím obrázku (Gravo, 2015) je zachyceno gravírování.



Obr. 5 Ukázka gravírování

4.1.4 Architektura

Další oblastí lidské činnosti, kde se v poslední době setkáváme s trojrozměrnými modely téměř pořád, je architektura. Zatímco v dřívějších dobách jsme se zde mohli při tvorbě návrhů setkat pouze s kreslícím prknem, rýsovacími pomůckami a tužkou, trend tvorby trojrozměrných modelů je v této oblasti stále populárnější. To je způsobeno obrovským rozvojem v oblasti počítačové grafiky a také stále více dostupnějšími prostředky pro tvorbu těchto modelů.

Zatímco si architekti v dřívějších dobách vystačili pouze s papírem a tužkou při tvorbě plánů a modelů budov, v dnešní době se s takovým postupem setkáme velmi zřídka. Tvorba plánů na papír byla velmi zdoluhavá a bylo zde nutné přesné rýsování. S nástupem trojrozměrných modelů se tato práce velmi zrychlila a o dost zjednodušila. Je k tomu sice potřeba počítač a specializovaný software, ale pokud vezmeme v úvahu jednoduchost tvorby modelů a hlavně mnohdy rychlost oproti rýsování, tak se tato metoda velmi vyplatí.

4.1.5 Prezentace výrobků

Trojrozměrné modely lze využít i k prezentaci výrobků. K tomuto účelu lze využít například modely, které byly dříve pouze předlohou k výrobě. Modely pro prezentaci výrobků by měli být dosti detailní a propracované. V případě, že model použitý při výrobě není pro prezentaci dostačující, lze jej doplnit a upravit, případně vytvořit model nový.

Takový model lze umístit například v internetovém obchodě či na stránkách výrobce, kde je tento výrobek popsán, místo běžných fotografií. Model lze také využít k vytvoření animace daného předmětu či k vytvoření videa, které může tento výrobek představovat. Tyto prvky mohou být použity k prezentaci tohoto výrobku a uživatel se tak může s daným předmětem, který je zastoupen těmito prvky, blíže seznámit a prohlédnout si ho z různých stran.

Ale pro takovou prezentaci často bývá nevýhodou velikost těchto modelů. Tyto modely mohou mít velkou velikost, a proto nemusejí být vždy vhodné k prezentaci na internetu. V případě velkých modelů se může stát, že se bude model jednak dlouho načítat, ale i poté jeho zobrazení nemusí být plynulé.

Hlavním faktem, který hraje velkou roli v rozšíření modelů při prezentacích výrobků, může být to, že model určený k výrobě nemusí být tak propracovaný, a proto musí tento model někdo upravit či udělat znovu. Často je tento proces zdoluhavý a je k němu potřeba využít různé programy, které mohou být placené. Proto se s možností využít modely k prezentaci výrobků často neseznamujeme. Pro firmy je totiž jednodušší daný produkt pouze nafotit a propagovat jej pomocí fotografií. Firmy tak ušetří spoustu času, který by takto musely věnovat tvorbě těchto modelů a také tím ušetří mnohdy nemalé částky.

4.2 Trojrozměrné modely PC komponent

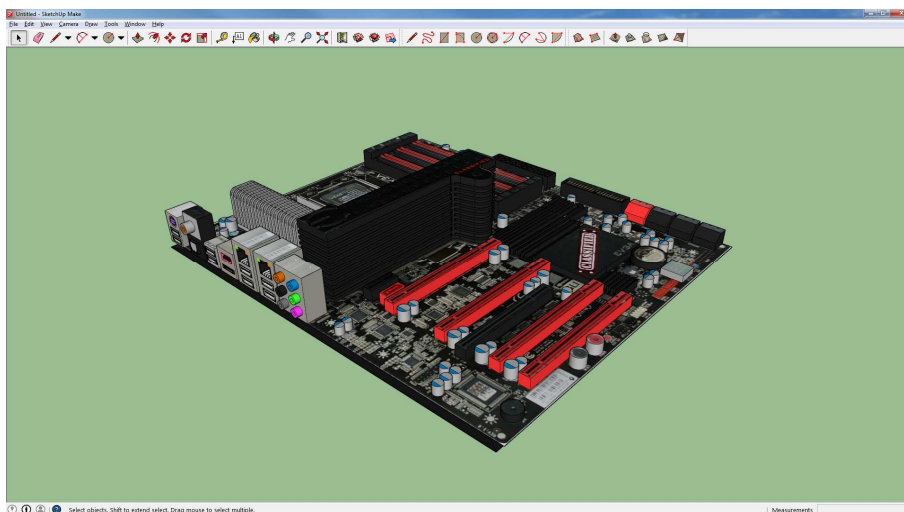
Trojrozměrné modely počítačových komponent nejsou zatím moc rozšířené. Sice se s nimi můžeme setkat na různých internetových stránkách nebo v 3D Web Warehouse přes program SketchUp, ale obvykle se jedná o neoficiální modely, které vytvořili běžní uživatelé a tyto modely nejsou určeny k prezentaci těchto komponent.

Obvykle se jedná pouze o statický model, který si uživatel může natočit a prohlédnout dle libosti, ale neseznamujeme se zde s žádnými interaktivními prvky či s animací.

I když se můžeme setkat s dobře propracovanými modely, realita je obvykle jiná. Většinou se jedná o jednoduché modely, které nemají vymodelované některé důležité detaily a místo těchto detailů je použita textura. Jak už bylo zmíněno, lze se setkat i s kvalitními modely, ovšem tyto modely nejsou volně dostupné a musíme za ně tedy zaplatit. Navíc zde platí pravidlo, že čím je model náročnější a detailnější, tím více místa zabírá. Proto většina těchto modelů byť i základních, není určena k zobrazení pomocí internetu.

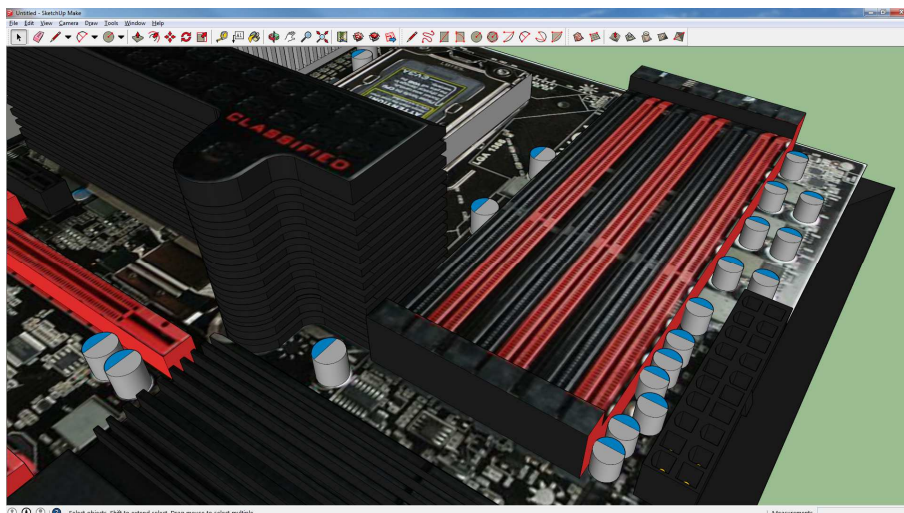
Jako ukázka modelu, který lze volně stáhnout, bude zobrazen statický model základní desky. Tento model základní desky (Cronicash, 2014) lze zdarma stáhnout pomocí programu SketchUp v 3D Web Warehouse. Podobných modelů tam, ale i jinde na internetu, lze najít velké množství. Tyto modely však postrádají mnoho důležitých detailů a při zobrazení z větší dálky se jeví jako propracované díky různým texturám, které pouze nahrazují tyto důležité detaily. Pro zobrazení z dálky toto nahrazení stačí, ovšem pro detailní zobrazení či prezentaci těchto modelů je to nevhodné. Důležité také je, aby daná textura měla určitou kvalitu. Tím se

ale mnohonásobně zvětšuje celková velikost výsledného modelu. Ukázkový model lze vidět na následujícím obrázku.



Obr. 6 Ukázka modelu základní desky

Pokud však takový model přiblížíme a podíváme se na něj z blízka, tak zjistíme, že co se dříve jevílo jako propracované, je zcela obyčejné a navíc to z blízka nevypadá ani moc dobře. Na následujícím obrázku lze mimo jiné vidět detail šesti slotů pro umístění paměti RAM. Autor jednak použil pro zobrazení detailů texturu a navíc se ani neobtěžoval vymodelovat každý slot zvlášť. Z obrázku si lze všimnout, že všech šest slotů je tvořeno pouze jedním objektem. Také textura samotné základní desky je velmi špatná a při detailním zobrazení je spíše rozmazaná.



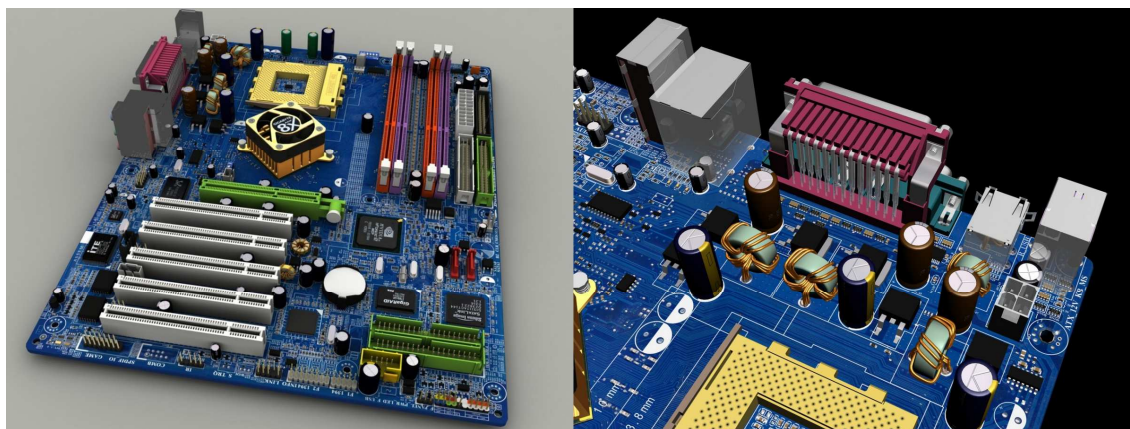
Obr. 7 Detailní pohled na model základní desky

Jak již bylo zmíněno, tak lze na internetu nalézt i velmi kvalitní a dobře vypadající modely zpracované do posledního detailu. Tyto modely však nejsou běžně ke sta-

žení. Tyto modely si daný zájemce musí koupit. V případě takovýchto modelů může být jejich cena velmi vysoká.

Ukázku propracovaného modelu základní desky (GrishaLutsenko, 2006) lze vidět na následujícím obrázku. Jedná se o velmi pěkný a detailně zpracovaný trojrozměrný model základní desky. Tento model je doplněn o vhodné textury. Je zde kvalitní textura znázorňující tištěný spoj samotné desky. Dále si můžeme všimnout, že nastavení vlastností a barev jednotlivých materiálů je velmi dobré a model díky tomu vypadá velmi realisticky. Příkladem může být materiál pro kovové části, který i zrcadlí okolní prvky tak, jako by se jednalo o skutečný kov. Pravá část tohoto obrázku znázorňuje detailní pohled na tuto základní desku. Na tomto obrázku si můžeme všimnout důkladně propracovaných jednotlivých částí desky. Detailně jsou zpracovány i jednotlivé elektronické součástky této základní desky, jako jsou cívky či kondenzátory.

Volně k dispozici jsou v případě tohoto modelu pouze jeho obrázky. Samotný obrázek znázorňující tento model je vytvořen ze dvou vybraných obrázků. Uvedený model stojí v přepočtu okolo 5000 Kč.



Obr. 8 Ukázka propracovaného modelu základní desky

5 Příprava materiálů

Před samotnou tvorbou modelů jednotlivých komponent bylo nutné rozhodnout, které komponenty bude vhodné modelovat. Jelikož je modelování poměrně zdlouhavý proces, nebylo možné vymodelovat celou počítačovou sestavu. Z tohoto důvodu byly pro tvorbu modelů vybrány tři základní komponenty počítače. Těmito komponentami jsou *základní deska* jakožto základ počítače, který propojuje ostatní komponenty ve funkční celek. Dále pak *paměť RAM*, která je určena pro dočasné ukládání zpracovávaných dat a spouštěného programového kódu. Třetí modelovanou komponentou je *procesor*, jakožto samotné jádro počítače, které provádí jednotlivé strojové instrukce (Dembowski, 2009). Pro samotné modelování bylo poté nutné získat nějaké materiály, podle kterých by bylo možné dané části vymodelovat. Možností bylo několik.

5.1 Využití internetu

První možností kde sehnat některé materiály jako předlohu pro modelování bylo použít internet. V dnešní době lze na internetu sehnat velké množství fotografií různých počítačových komponent. Jsou zde k dispozici i různá videa představující jednotlivé komponenty.

Ovšem modelování pouze podle fotografií nebo videa by bylo velmi obtížné. Proto bylo nutné nějakým způsobem určit alespoň některé základní rozměry pro získání představy o celkové velikosti a dále aby bylo alespoň nějak možné určit rozměry ostatních částí.

Proto bylo nutné stanovit konkrétní komponenty a dále určit jejich základní rozměry. Rozměry základní desky byly vyhledány v jejím manuálu, kde tyto rozměry byly uvedeny. Na základě těchto rozměrů a fotografií bylo již jednodušší stanovit velikost ostatních částí.

Obdobně bylo nutné najít rozměry paměti RAM a procesoru. V návodech k těmto komponentám však jejich rozměry uvedené nejsou, proto bylo nutné najít nějaký podrobný popis či pláněk přímo podle zvoleného typu. U paměti RAM bylo toto hledání velmi rychlé. Po krátké chvíli byl nalezen obrázek potřebného typu DDR3 s uvedenými rozměry. V případě procesoru však stále nebylo nic vhodného nalezeno.

I přesto, že byly nalezeny některé vhodné základní rozměry jednotlivých komponent a pomocí těchto rozměrů bylo možné alespoň odhadem určit velikosti jednotlivých částí, bylo modelování stále velmi obtížné. Jednak pořád zůstávaly nějaké nesrovnalosti v některých rozměrech, ale navíc i přes velké množství fotografií různých komponent, které lze na internetu najít, byly informace o těchto komponentách stále nedostačující. Většina fotografií je totiž zaměřená na danou komponentu jako na celek a nalézt detailní fotografie jednotlivých částí konkrétních komponent byl problém. Natož nalézt detailní fotografie, na kterých by byla daná část znázorněna vždy z jiné strany. Z těchto důvodů, bylo toto řešení nevyhovující.

5.2 Nafocení komponent

Další možností bylo pořízení vlastních fotografií jednotlivých komponent. Zde však nastal jeden základní problém, kde vzít jednotlivé komponenty, které by bylo možné nafotit.

Z tohoto důvodu bylo nutné obejít některé obchody zaměřené na prodej výpočetní techniky a poptat se tam. Jednotlivým zástupcům těchto prodejen tedy bylo nutné popsat moji situaci a požádat je zda by mi neumožnily nafocení některých komponent. Když jim bylo vysvětleno, že tyto fotografie budou využity pro tvorbu diplomové práce, tak byli vždy ochotní mi pomoci či alespoň nějak poradit.

Nakonec byla využita jedna možnost u jednoho nejmenovaného prodejce, která byla pro mě nejpříjemnější. Bylo mi řečeno, že k nafocení mi mohou poskytnout pouze zboží, které je buďto rozbalené nebo použité. Že kdyby rozbalili nové zboží, tak by je museli prodat se slevou. Proto mi bylo doporučeno objednání jakéhokoli rozbaleného nebo použitého zboží na jejich internetovém obchodě a zaslání tohoto zboží k osobnímu odběru na tu prodejnu. Prý jakmile bude zboží na prodejně, že mě kontaktují a že mi umožní jednotlivé komponenty libovolně nafotit. Poté že moji objednávku stornují.

Na jejich internetovém obchodě bylo tedy vybráno potřebné zboží a toto zboží bylo dále objednáno. Druhý den mi přišla zpráva o tom, že mnou objednané zboží je připravené k vyzvednutí. Poté tedy již stačilo pouze na danou prodejnu zajít a pořídit potřebné detailní fotografie jednotlivých částí ze všech možných úhlů a stran. Samotné vyfocení ovšem nebylo dostačující. Bylo ještě nutné stanovit rozměr jednotlivých částí. Proto byly potřebné části přeměřeny nebo při tvorbě některých fotografií k nim bylo přiloženo pravítko.

Po pořízení fotografií je bylo nutné roztřídit a promazat. Některé z těchto fotografií totiž byly rozmazané, na jiných byly odlesky světla nebo byly zcela přesvětlené a nebylo na nich nic poznat. Po promazání těchto nevhodných fotografií mi stále zůstalo k dispozici dostatečné množství dobrých fotografií, podle kterých bylo možné vytvořit jednotlivé modely počítačových komponent.

Poté bylo možné dané komponenty modelovat. Toto řešení však také nebylo zcela ideální a mělo spoustu nedostatků. Tvorba modelů zde byla velmi omezená tím, co bylo na jednotlivých fotografiích zachycené a stále bylo nutné přepínat mezi několika fotografiemi tak, aby bylo možné si pořádně prohlédnout modelovanou část z více stran.

5.3 Shromáždění komponent

Z výše zmíněných důvodů bylo zapotřebí využít ještě jiné možnosti. Jako další možnost se nabízelo obstarat si nějaké komponenty. Ovšem kupovat tyto komponenty nepřipadalo v úvahu.

Proto bylo nutné se poptat známých a přátel a zkusit si od nich vypůjčit nepoužívané komponenty, případně sehnat nějaké zastaralé či nefunkční části. Tímto způsobem byly nakonec shromážděny některé potřebné části, jako je nefunkční procesor a nějaké vyřazené základní desky. V případě základních desek se však

jednalo o starší typy a většinou i nefunkční desky. I přes tyto skutečnosti však v jistých ohledech ve spojení s fotografiemi komponent byly tyto části pro modelování základní desky a procesoru postačující.

Modelování těla základní desky probíhalo dle obrázků a stejně tak probíhalo umístování jednotlivých částí na základní desku. Mnohé konektory a patice však byly i na shromážděných zastaralých základních deskách totožné, proto pro modelování těchto částí byly využity tyto konektory starých základních desek. Pro jednoduché prohlížení těchto konektorů byla jedna stará základní deska rozřezána tak, aby pro modelování byly k dispozici pouze samotné konektory.

Ještě však bylo nutné sehnat paměť RAM. Aby tvorba paměti RAM nebyla vázána pouze na fotografie, bylo tedy nutné dočasně vyndat jednu z pamětí z mého počítače. A tuto paměť použít jako předlohu pro modelování. Poté mi už v samotném modelování nic nebránilo.

6 Vytvoření trojrozměrného modelu

Pro účel tvorby trojrozměrných modelů byla použita, jak již bylo zmíněno dříve, kombinace jazyka VRML a 3D grafického programu Rhinoceros. Při tvorbě modelů bylo využito výhod každé z těchto metod tak, aby výsledný model byl detailní, ale prostorově nenáročný.

Před samotnou tvorbou modelů byly stanoveny předpoklady na jejich výslednou funkčnost a možnost distribuce. Tyto předpoklady byly pokryty možnostmi zvolených metod tvorby modelů. Mezi tyto předpoklady výsledného modelu lze zařadit:

- Snadná dostupnost
- Malá velikost modelu
- Dostatečná detailnost
- Zobrazení v reálném čase
- Možnost zobrazení bez nutnosti instalace náročných a placených programů
- Možnost doplnění animace
- Vhodné doplnění interaktivních prvků

Snadná dostupnost, možnost zobrazení bez nutnosti náročných programů a zobrazení v reálném čase zde byla docílena pomocí zobrazení modelu za použití internetového prohlížeče. Samotný prohlížeč však v tomto případě nestačí. Je nutné do něj nainstalovat některý ze zásuvných modulů, které dokážou tyto modely vykreslit a umožní jejich procházení. Tyto zásuvné moduly již byly zmíněny.

Dostatečná detailnost byla zajištěna snadným vymodelování složitých částí pomocí programu Rhinoceros. Tyto části byly dále převedeny na dostatečně detailní polygonovou síť a poté vyexportovány do jazyka VRML. Tyto části byly díky překopírování jejich kódu zahrnuty do modelu.

Možnost doplnění animace a vhodných interaktivních prvků byla docílena přímo možnostmi jazyka VRML. Jelikož se jedná o jazyk pro zápis virtuální reality, nebyl problém tyto prvky do modelu dodělat pomocí doplnění zdrojového kódu k daným modelům.

Vhodným popisem jednotlivých částí modelu v rámci možností jazyka VRML, doplněním složitých ploch v podobě exportovaných částí z programu Rhinoceros a vhodným využitím prototypů bylo možné docílit *malé velikosti modelu*.

6.1 Metody využité při tvorbě modelů

Při tvorbě modelů bylo dbáno na to, aby zápis jazykem VRML byl přehledný. Z tohoto důvodu nebylo vše popsáno pouze v jednom souboru, ale pro popis každé části byl použit jiný soubor. Některé části jsou ale dosti složité, a proto jsou popsány značným množstvím kódu a to může být důvodem, že i při zápisu každé části do jiného souboru může být některý z těchto soubor mírně nepřehledný.

Modelováním byly vytvořeny modely procesoru a paměti RAM v samostatných souborech. Při modelování základní desky však nebylo možné, aby celá základní deska byla popsána v jednom souboru. Jelikož jsou některé konektory a patice velmi složité a jejich popsání by bylo velmi obsáhlé a mnohdy i méně přehledné samo o sobě, bylo nutné model základní desky rozdělit na jednotlivé části a tyto části základní desky popsat v samostatných souborech. Tudíž každý typ konektoru, každá patice a ostatní části základní desky jsou popsány vždy v samostatném souboru.

Při tvorbě některých částí stačilo pouze použít možnosti jazyka VRML, u jiných bylo nutné použít program Rhinoceros pro vytvoření složitějších částí. Použité metody při tvorbě jednotlivých modelů jsou popsány níže.

6.1.1 VRML a Rhinoceros

Pomocí jazyka VRML lze snadno a rychle popsat základní tvary a předměty. Složitější části však bylo nutné modelovat pomocí programu Rhinoceros. Bylo však nutné stanovit hranici mezi tím, co se vyplatí popsat pomocí jazyka VRML a tím, co bude vhodnější vymodelovat pomocí Rhina.

V případě že by se jednalo o časovou náročnost tvorby těchto modelů, tak by ve všech ohledech vyhrálo Rhino. V případě popisu byť i základních těles pomocí jazyka VRML je totiž velice nutné přepočítávat různá posunutí a velikosti jednotlivých částí. Posunutí tělesa se provádí pomocí určení středu tohoto tělesa v dané trojrozměrné soustavě. Proto když chceme umístit i dvě obyčejné tělesa, jako je kvádr a koule, vedle sebe tak, aby se pouze dotýkali, je nutné stanovit střed prvního tělesa a poté na základě jednotlivých rozměrů obou těles spočítat střed druhého tělesa. V případě že se jedná o větší množství takovýchto těles, je tento přepočet velice zdlouhavý. Při této tvorbě modelů se však jednalo o to, aby měl model co nejmenší velikost.

Proto, aby bylo možné stanovit zmíněnou hranici, bylo nutné vyzkoušet pomocí obou metod vytvořit jednoduché cvičné příklady. Nejprve byl daný předmět popsán pomocí jazyka VRML a poté byl ten samý předmět vymodelován pomocí Rhina. Program Rhinoceros, ale i ostatní modelovací programy, mají totiž značné nedostatky v exportu do VRML. Tyto nedostatky spočívají v tom, že tyto programy nedokážou popsat ani základní tvary jako je koule, krychle atd. pomocí reprezentace jazyka VRML, a pro export do VRML je nutné tyto objekty převést na polygonovou síť. Zkušební objekty byly proto vyexportovány na co nejjednodušší polygonovou síť, aby bylo možné porovnat tu nejlepší možnost s jazykem VRML.

Pro porovnání bylo vytvořeno několik takovýchto testovacích dvojic. Poté byly porovnány velikosti jednotlivých dvojic souborů a tím bylo zjištěno, že ve všech případech byla reprezentace těchto poměrně jednoduchých objektů vždy menší pomocí zápisu jazykem VRML. Občas rozdíl mezi velikostmi těchto souborů nebyl příliš velký, ale v mnoha případech byly tyto rozdíly významné. To se však jednalo o popis objektů, které byly popsány několika desítkami případně pomocí stovek řádků. Jak velké by pak byly rozdíly mezi těmito velikostmi, kdyby takovýto model byl reprezentován tisíci řádky kódu.

Po podrobném prozkoumání zdrojového kódu vyexportovaného modelu bylo zjištěno, že pro popis pomocí polygonové sítě jsou uvedeny číselné hodnoty mnohdy až na patnáct desetinných míst. To je způsobeno převodem na tuto síť, kde je nutné přesně popsat polohu vrcholů jednotlivých trojúhelníků, které tímto převodem vznikly. Kdyby tyto vrcholy nebyly popsány dostatečně přesně, mohlo by docházet k chybám při vykreslování těchto objektů a mezi jednotlivými trojúhelníky by mohly být vidět díry.

Závěrem tohoto hodnocení bylo rozhodnutí, *popsat co nejvíce možných částí pomocí jazyka VRML a pouze složité části, které by bylo nutné ručně popsat pomocí polygonové sítě v jazyku VRML, modelovat pomocí programu Rhinoceros.*

6.1.2 Příkazy DEF a USE

Při popisu modelů jazykem VRML se můžeme často setkat s tvorbou stejných případně podobných těles. Tato tělesa se mohou lišit velikostí, barvou nebo pouze polohou. Přesně popsat tato tělesa může být mnohdy zdlouhavé a pracné. Když pak máme jedno určité těleso v modelu použít několikrát, tak nás místo opětovného a složitého popisování tohoto tělesa napadne jednoduše zkopírovat kód popisující toto těleso a pouze změnit třeba polohu.

Jazyk VRML je programovací jazyk a mnoho věcí je v něm již popsáno a připraveno pouze k použití. Proto není nutné v takovýchto případech dané těleso popisovat vícekrát, ale pouze stačí toto těleso pojmenovat pomocí příkazu *DEF* a uvedením jména dané struktury za tento příkaz.

V případě, že potřebujeme tento objekt použít jinde, tak nám stačí použít pouze příkaz *USE* a napsat jméno potřebné struktury za tento příkaz. Pak jen stačí tento objekt umístit na potřebné místo. Takto lze vytvářet libovolné množství kopií jednotlivých těles.

Pomocí těchto příkazů lze vytvářet kopie jednotlivých uzlů jazyka VRML. Těmito uzly mohou být jednotlivé tělesa či materiály. Mimo to lze vytvářet i kopie celých stromových struktur.

Žára ve své knize (Žára, 1999) uvádí níže popsané výhody, ale i omezení používání těchto příkazů. Možnost vytvářet kopie objektů pomocí příkazů *DEF* a *USE* má hlavně výhodu v tom, že se pomocí těchto kopií dá ušetřit mnoho znaků, které by byly pro popis stejných těles potřeba, a tím výrazně snížit velikost souboru. Další výhodou je, že se mnohdy pomocí tohoto zápisu zrychlí zobrazení těchto modelů.

Použití příkazů *DEF* a *USE* má však i určité omezení. Vytvářet takovéto kopie určitého objektu je možné pouze v souboru, ve kterém máme tento objekt definovaný pomocí příkazu *DEF*. Vytvářet kopie objektů z jiného souboru není pomocí této struktury možné.

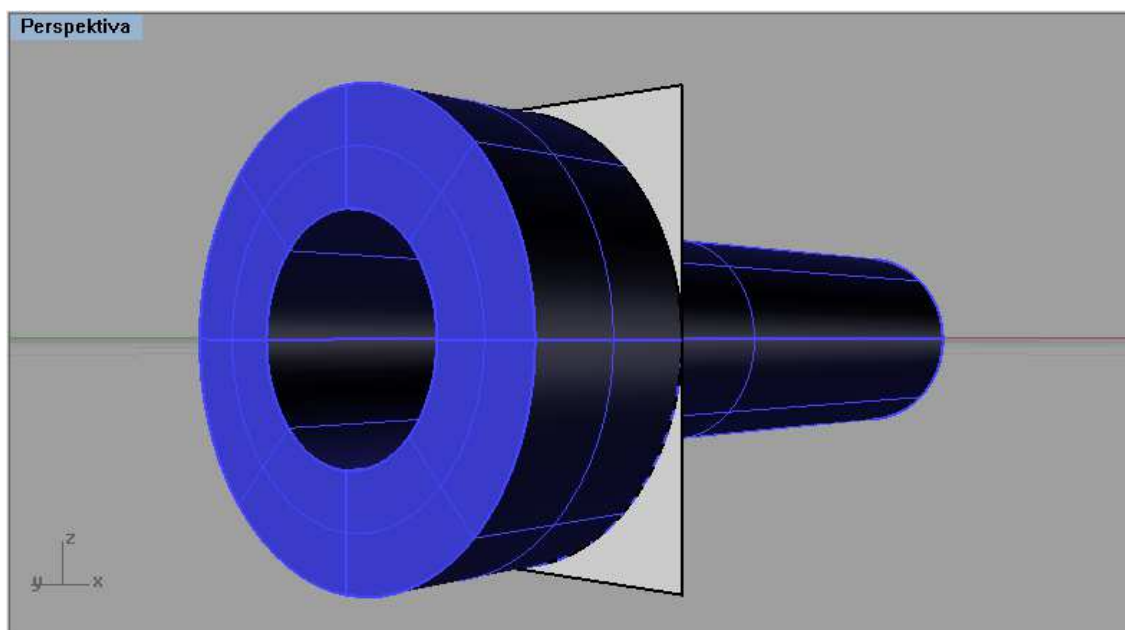
Pro pojmenovávání objektů se doporučuje používat různá jména. Není to sice pravidlo a je možno použít jedno jméno pro více objektů. V případě vytvoření kopie takového objektu se nám vytvoří kopie toho objektu, který byl definován nejbližší před touto tvorbou kopie.

Při používání těchto příkazů musíme dbát na to, aby příkaz *USE* nebyl použitý ve stromu, který je volán. V takovém případě by mohlo dojít k situaci, že se tento

příkaz volá pořád dokola a vytváří nekonečné množství potomků. Jedná se o zacyklení. Některé prohlížeče jsou proti tomu ošetřeny a v případě výskytu takové struktury rovnou hlásí chybu. Mohlo by se však stát, že prohlížeč ošetřen proti takovému zápisu není a bude tento příkaz vykonávat, dokud jej uživatel neukončí.

Při tvorbě všech modelů byly tyto příkazy používány dost často. Každý model, který byl v této práci vytvořen, obsahoval těchto příkazů několik. Pomocí vytváření kopií jednotlivých částí bylo ušetřeno velké množství místa.

Příkladem použití těchto dvou příkazů může být vytvoření zděře u audio konektoru. Jelikož je tato zděř kulatá a její tvar by bylo velmi složitě popsat jazykem VRML, byla vymodelována v programu Rhinoceros. Ukázkou vymodelované zděře lze vidět na následujícím obrázku. Modře je znázorněna samotná zděř a šedé okolní plošky jsou pouze část samotného těla audio konektoru. Tyto plošky bylo nutné vyexportovat společně se zděří, protože při samostatném exportu by nemusela přesně sedět daná polygonová síť a mohly by být mezi těmito objekty škvíry.



Obr. 9 Zděř audio konektoru

Nyní již stačilo pouze tuto zděř převést na dostatečně podrobnou polygonovou síť a vyexportovat ji. Po provedení exportu bylo zjištěno, že při popisu pomocí dostatečně podrobné polygonové sítě je velikost této zděře okolo 30 kB. Řekneme si, že 30 kB v dnešní době nic není, ale jedná se pouze o jednu z částí jednoho konektoru. Navíc tyto zděře jsou na tomto konektoru 3, a kdybychom je měli mít popsané všechny takto, tak by to už bylo 90 kB. A to máme pouze částečné zděře. Ještě v nich chybí kontaktní plošky a také chybí zbytek celého konektoru. Audio konektor je tvarem celkem jednoduchý. Na základní desce nalezneme spoustu jiných konektorů, které jsou mnohdy složitější. A pokud má být určen takový model k prezentaci na internetu, je nutné, aby jeho velikost byla co nejmenší. Proto bylo

nutné tento vymodelovaný konektor ještě upravit. Zděř byla rozdělena na čtvrtiny a vyexportovaná byla pouze jedna čtvrtina. Tím bylo ušetřeno poměrně značné množství celkového místa.

Nyní již pouze stačilo upravit tuto vyexportovanou část pomocí jazyka VRML. Vyexportovaný soubor obsahoval dvě tělesa. Jedním byla část této zděře a druhým zobrazená okolní ploška. Zde byl použit příkaz DEF k pojmenování obou částí. Poté pouze stačilo třikrát použít příkaz USE ke zkopírování těchto částí a upravit jejich rotaci kolem středové osy konektoru. Tuto osu nebylo nutné určovat, protože při modelování zděře v programu Rhinoceros byl střed zděře umístěn na střed souřadné soustavy. Z důvodu aby nebylo nutné určovat každý úhel rotace, byly jednotlivé kopie také pojmenovány a k vytvoření další kopie byla použita vždy kopie předchozí. Z tohoto důvodu mohl být úhel ve všech případech stejný. Hodnota úhlu, o který se těleso otočilo, byla 90°. V kódu však bylo nutné tuto hodnotu zapsat v radiánech. Strukturu použití tohoto kopírování lze vidět v kódu níže.

```
# čtvrtina zděře pojmenovaná jako diraCast1
DEF diraCast1 Shape {
  appearance Appearance {
    material Material {
      # barva zděře
      diffuseColor 0.4 0.4 0.8
    }
  }
  # zde byl umístěn popis polygonové sítě
  # pro ukázkou využití příkazů DEF a USE není tento popis potřebný
}
# okolí této zděře pojmenované jako okoliDiry1
DEF okoliDiry1 Shape {
  appearance Appearance {
    material Material {
      # barva této okolní plošky
      diffuseColor 0.7 0.7 0.7
    }
  }
  # zde byl umístěn popis polygonové sítě
  # pro ukázkou využití příkazů DEF a USE není tento popis potřebný
}
# vytvoření 1. kopie a její pojmenování jako diraCast2
DEF diraCast2 Transform {
  # rotace o zmíněných 90° okolo osy X
  rotation 1 0 0 1.5707963267
  children [
    # použití původních objektů diraCast1 a okoliDiry1
    USE diraCast1
    USE okoliDiry1
  ]
}
# pro vytvoření zbylých dvou kopií jsou použity kopie předešlé
DEF diraCast3 Transform {
  rotation 1 0 0 1.5707963267
```

```

children
  USE diraCast2
}
DEF diraCast4 Transform {
  rotation 1 0 0 1.5707963267
  children
    USE diraCast3
}

```

6.1.3 Použití prototypů

Mimo výše zmíněných příkazů DEF a USE lze vytvořit pojmenovanou strukturu i pomocí prototypů. Prototyp slouží k definování vlastních uzlů a k popisu jejich vlastností. Využitím prototypů lze tak snadno definovat celé stromové struktury nebo vytvářet své vlastní knihovny materiálů nebo speciálních objektů. Prototyp je možné přirovnat k funkcím a procedurám z běžných programovacích jazyků.

Použití prototypů nám přináší oproti příkazům DEF a USE hned několik výhod. Pokud definujeme nějakou strukturu jako prototyp, tak se nám tato struktura nezobrazí. Pro zobrazení ji musíme zavolat pomocí definovaného jména. V případě pojmenování pomocí příkazu DEF se nám zobrazovala i tato původní pojmenovaná posloupnost příkazů. Další výhodou je to, že prototyp lze snadno načíst i do jiného souboru a zde jej použít. Není vázán pouze na soubor, ve kterém byl vytvořen, tak jako tomu bylo v případě příkazu DEF. Hlavní rozdílem je zde i možnost měnit jednotlivé vlastnosti této struktury pomocí nadefinovaných parametrů. V případě příkazů DEF a USE bylo možné změnit pouze vlastnosti, které byly spojené s transformací, ve které byl příkaz USE použitý. Těmito vlastnostmi byla velikost, otočení a posunutí. V případě prototypů můžeme měnit i barvy, rozměry jednotlivých částí a další věci (Žára, 1999).

Vytvoření prototypů je velmi jednoduché. Prototyp se uvádí slovem *PROTO*. Za tímto slovem je uveden *název prototypu*. Za názvem jsou v hranatých závorkách *definované parametry* včetně počátečních hodnot. Nakonec se mezi složené závorky zapíše samotné *tělo prototypu* vytvořené z příkazů jazyka VRML. Pro lepší pochopení je níže uvedena struktura pomocí příkazů.

```

PROTO jméno [
  # seznam deklarácí parametrů včetně výchozích hodnot
  field <typ dat> <jméno parametru> <výchozí hodnota>
  # například
  field SFVec3f posunutí 0 0 0
  field SFCOLOR barva 1 0 0
] {
  # tělo prototypu
}

```

V předchozí ukázce je znázorněna struktura vytvoření prototypů. Tato struktura byla popsána v předchozím odstavci. Nyní se zaměříme na deklaraci parametrů. V ukázce je parametr uveden slovem *field*, za tímto slovem je uveden *datový typ*. Za datovým typem je nutné napsat samotné *jméno parametru* a dále pak *výchozí hod-*

notu tohoto parametru. Pro každý prototyp je možné mít libovolné množství parametrů. Také je však možné deklaraci parametrů vynechat.

Takto deklarované parametry lze pak snadno využít v samotném těle prototypu pro definování jeho různých vlastností. Pro připojení parametru k dané vlastnosti použijeme *přiřazovací příkaz*. V jazyku VRML je přiřazovací příkaz realizován slovem *IS*. Pomocí přiřazovacího příkazu tak snadno můžeme přiřadit nový parametr *posunutí* skutečnému parametru *translation* a nově definovaný parametr *barva* skutečnému parametru *diffuseColor* (Žára, 1999). Pro vhodnou práci s prototypem je dobré uvádět samotný popis tohoto prototypu jako potomka uzlu *Transform*. Parametry tohoto uzlu nám totiž umožní snadno definovat vlastnosti, jako je *posunutí*, *otočení* nebo *měřítko* celého prototypu. Pro lepší názornost je tato možnost popsána včetně přiřazovacího příkazu v následující ukázce.

```
PROTO prototyp1 [
  field SFVec3f posunuti 0 0 0
  field SFRotation natoceni 0 0 0 0
  field SFVec3f meritko 1 1 1
]{
  Transform {
    translation IS posunuti
    rotation IS natoceni
    scale IS meritko
    children [
      # telo prototypu jako potomek uzlu Transform
    ]
  }
}
```

Vytvořený prototyp můžeme snadno použít kdekoliv v souboru, ve kterém je tento prototyp vytvořen. To provedeme uvedením *jména prototypu* a případným *doplňným parametřů*, u kterých chceme změnit předdefinovanou hodnotu. Tentokrát však tyto parametry zapisujeme do závorek složených a pouze ve tvaru *název parametru* a *nová hodnota*. Použití prototypu z minulé ukázky je znázorněno níže.

```
prototyp1 { posunuti 10 5 2 meritko 1.5 1.5 1.5 }
```

Pokud však budeme chtít použít prototyp v jiném souboru, než ve kterém byl vytvořen, musíme ho do tohoto souboru nejprve načíst. Načtení prototypu provedeme příkazem *EXTERNPROTO*, který oznamuje, že takto popsáný prototyp je definovaný v jiném souboru a že v tomto souboru nalezneme pouze parametry tohoto prototypu. Za klíčovým příkazem pak následuje *jméno prototypu* přiřazené pro tento soubor. Toto jméno může, ale také nemusí, být totožné jako skutečné jméno prototypu. Dále je pak uveden *seznam parametřů* v hranatých závorkách. Jedná se o opsání skutečných parametrů prototypu bez výchozích hodnot. A nakonec je nutné do uvozovek napsat *adresu souboru*, ve kterém se prototyp nachází a dále pak pomocí symbolu *#* uvést *skutečné jméno prototypu* a to stále uvnitř zmíněných uvozovek. Níže je uveden příklad obecného zápisu načítání prototypů a pod tímto příkladem je konkrétní ukázka.

```
# obecný zápis
EXTERNPROTO nazev_prototypu [
  # seznam skutečných parametrů
] "soubor_s_prototypem#skutečný_název_prototypu"
# konkrétní příklad
EXTERNPROTO prototyp1 [
  field SFVec3f posunuti
  field SFRotation natoceni
  field SFVec3f meritko
] "nazev_souboru.wrl#prototyp1"
```

Po načtení prototypu do daného souboru, lze dále s tímto prototypem pracovat stejně, jako kdyby byl v tomto souboru definován.

Možnosti prototypů jsou stejně jako možnosti DEF a USE v této práci hojně využívány. Jedním z důvodů, proč byly prototypy tak často používány, bylo rozdělení modelů do více souborů. Příkladem může být základní scéna. Tato scéna se skládá z 3 objektů, kterými jsou základní deska, procesor a paměť RAM. Každý z těchto objektů je přitom popsán v jiném souboru. Navíc samotná základní deska je složena z velkého množství konektorů a jiných částí. Každý prvek základní desky je také popsána ve svém souboru. Z tohoto důvodu bylo nutné nějakým způsobem docílit načtení těchto částí v jednom souboru a tím tyto části spojit.

Dalším důvodem, proč byly použity při tvorbě modelů prototypy, byla možnost vícenásobného zobrazení jednotlivých částí. K tomu lze sice využít i příkazy DEF a USE, ale v případě těchto příkazů, jak už bylo zmíněno výše, nelze definovat některé vlastnosti těchto objektů. Takovou vlastností může být například barva, kterou bylo zapotřebí v případě některých objektů změnit. Využití prototypů bylo vhodné i v případě již zmíněného audio konektoru a jeho zděře. Jak bylo popsáno výše, tak k vytvoření zděře konektoru byly využity příkazy DEF a USE. Dále byla zděř doplněna o kontaktní plošky. Tím byla zděř hotová. Poté ji však bylo nutné zobrazit na třech různých místech a bylo nutné zajistit, aby měla jinou barvu. Toho pomocí příkazů DEF a USE nešlo docílit. Z tohoto důvodu byla tato vytvořená struktura zděře zabalena do těla prototypu a k jednotlivým vlastnostem materiálů pak byly přiřazeny parametry tohoto prototypu. Díky tomu bylo možné snadno vytvořit libovolné množství kopií této zděře a pokaždé s jinou barvou.

6.1.4 Nastavení možností procházení modelu

Při tvorbě modelů bylo nutné nadefinovat možnosti zobrazení modelu a možnosti procházení scény. Pro toto nastavení je určen uzel *NavigationInfo* (ovládání avatara). Zde je možné nastavit vlastnosti Avatara a to jak bude danou scénu procházet.

V souboru může být těchto navigačních uzlů více, ale pouze jeden může být v danou chvíli aktivní. Při načtení scény se vždy nastaví ten, který je definován jako první.

Jak uvádí Žára ve své knize (Žára, 1999), pomocí uzlu *NavigationInfo* lze nastavit následující vlastnosti:

- avatarSize (velikost avatara)

- headlight (svítilna)
- visibilityLimit (omezení viditelnosti)
- speed (rychlost)
- type (typ průchodu)

Pomocí parametru *avatarSize* se nastavují vlastnosti samotného avatara. Přesněji řečeno jeho rozměry. Tento parametr je polem a nastavují se zde 3 hodnoty. První hodnota definuje maximální povolenou vzdálenost, na kterou se smí avatar přiblížit k překážce. Druhá hodnota určuje výšku očí nad terénem. Pomocí třetí hodnoty se definuje maximální výška, kterou dokáže avatar překročit.

Dalším parametrem je *headlight*. Tento parametr slouží k zapnutí a vypnutí avatarova čelního světla. Toto světlo sleduje směr pohledu avatara a vždy ozařuje tu část světa, která je právě v zorném poli.

Třetím parametrem, kterým je *visibilityLimit*, lze určit jak daleko avatar vidí. Výchozí hodnota tohoto parametru je 0, což znamená, že avatar vidí do nekonečna.

Pomocí parametru *speed* lze nastavit rychlost pohybu avatara. Tato rychlost se uvádí v m/s.

Posledním parametrem je parametr *type*. Díky tomuto parametru můžeme nastavit, jaké metody pohybu bude možné využít při průchodu scénou. Je možné nastavit více metod současně a uživatel může mezi těmito metodami přepínat. Tyto metody jsou tři a jsou jimi:

- WALK (chůze)
- FLY (let)
- EXAMINE (zkoumání)

Metoda *WALK* slouží k běžnému procházení scénou. Tato metoda simuluje běžnou chůzi. Avatar se během této metody pohybuje po zemi či podložce a zároveň kopíruje tvar této podložky. Působí na něj přitažlivost ve směru záporné části osy y.

Pokud je zvolena metoda *FLY*, tak avatar může procházet scénou stejně jako při předchozí metodě. Zde však na něj nepůsobí přitažlivost.

Poslední metodou jak se ve scéně pohybovat je metoda *EXAMINE*. Tato metoda je určena ke zkoumání objektů. Avatar může při této metodě libovolně kroužit kolem objektů ze všech stran. Při této metodě zobrazení je ignorována rychlost a také jsou vypnuty kolize, tudíž je možný průchod objekty. Avatar se zde otáčí kolem samotného středu scény, který je automaticky určen.

V některých případech je však možné tento střed určit zadáním jeho polohy. Některé prohlížeče těchto virtuálních světů, jako je například Cortona3D, totiž umožňují v popisu samotného avatara, tedy v *avatarSize*, zadat více hodnot. První 3 hodnoty běžně definují avatara a další 3 hodnoty určují střed otáčení právě pro metodu *EXAMINE* (ParallelGraphics, 2015). Na to však nelze spoléhat. Vždy záleží na samotném prohlížeči a na tom, jak má tuto možnost implementovanou.

Pro prohlížení vytvořených modelů byla zvolena možnost *EXAMINE*. Scéna je zde tvořena jednotlivými modely a z tohoto důvodu prolétat či procházet libovolně

tuto scénu není v tomto případě vhodné. Uživatel by se tak mohl dostat sice kamkoliv, ale v tomto případě se jedná pouze o model umístěný na středu a nikde jinde nic není. Navíc tyto modely jsou určeny k prezentaci jednotlivých počítačových komponent, a proto je vhodné, aby si tyto komponenty mohl uživatel libovolně prohlédnout ze všech stran. K tomuto účelu je právě určena tato zvolená metoda.

Díky zvolené metodě procházení nebylo nutné řešit některé další věci. Jak již bylo zmíněno, při procházení scénou pomocí metody EXAMINE se ignoruje nastavená rychlost a avatar může procházet tělesy. Proto nebylo zapotřebí řešit rozměry avatara a jeho rychlost. Proto byla jednotlivým rozměrům avatara nastavena hodnota 0.

6.1.5 Vytvoření a mapování textur

Různé objekty ve scéně mají různé vlastnosti. Každý objekt vyžaduje různé vlastnosti materiálů a barev. Při vytvoření některých objektů však nebylo možné vhodně popsat jeho povrch pouze pomocí barev a vlastností této barvy. Na povrch takových objektů bylo nutné nanést texturu.

Při tvorbě modelů bylo potřeba nanést texturu na povrch samotného těla základní desky, paměti RAM, na tělo procesoru a na horní stranu patice pro připojení procesoru.

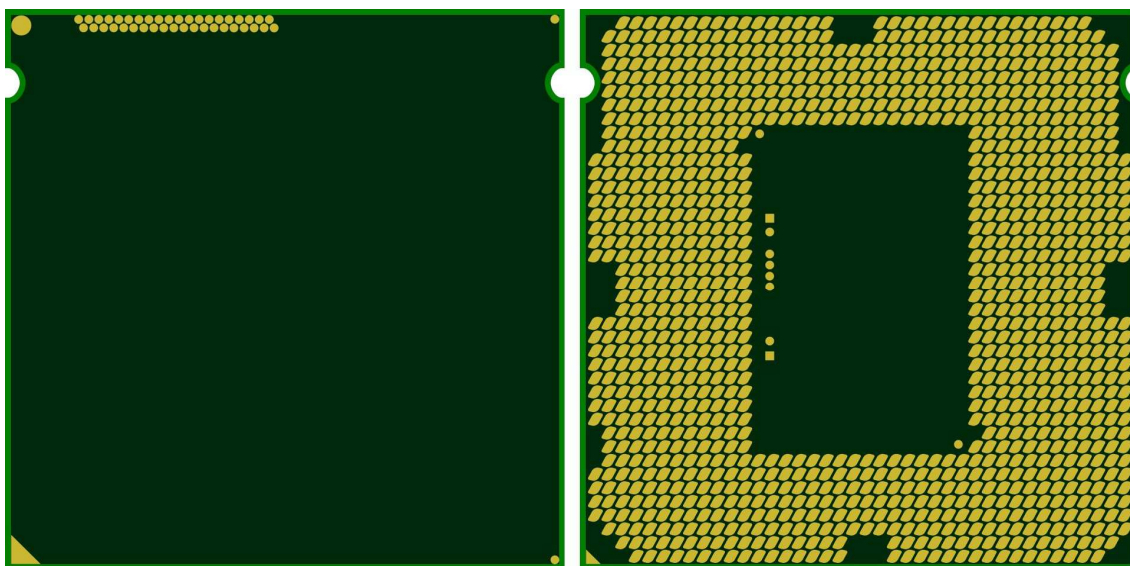
Tyto objekty jsou vždy něčím specifické a tyto specifické vlastnosti se liší typ od typu. Proto využít nějaký obrázek či vytvořenou texturu, kterou by bylo možné stáhnout na internetu, nebo kdekoli jinde sehnat, nebylo možné. V této oblasti člověk totiž moc textur nenalezne a použití obrázku není vhodné, jelikož obrázky znázorňují tento předmět se všemi ostatními částmi a bylo potřeba tuto texturu nanést pouze na samotné tělo těchto komponent.

Z těchto důvodů bylo nutné tyto textury vytvořit. Textury byly vytvořeny v programu Corel Draw, který je určen pro práci s vektorovou grafikou.

Textury bylo nutné vytvořit dostatečně velké a detailní, aby bylo zaručeno jejich dobré rozlišení i při pohledu z blízka. Z těchto důvodů bylo zapotřebí tyto textury vytvořit mnohem větší, než byla samotná plocha, na kterou byly tyto textury později naneseny. Také však bylo nutné docílit co nejmenší velikosti, aby načítání textur netrvalo moc dlouho. Části některých textur bylo nutné zprůhlednit, proto byl jako formát těchto textur zvolen formát gif, který průhlednost podporuje. Dále bylo nutné vhodně zvolit hranici mezi kvalitou a velikostí, kterou obrázek zabíral.

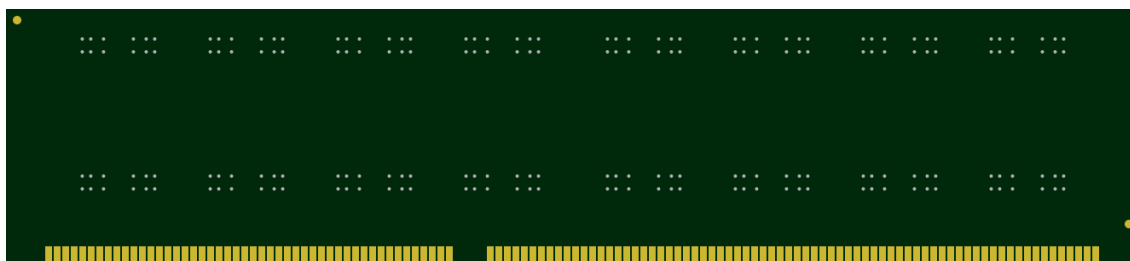
V případě procesoru bylo potřeba vytvořit 2 textury. Jedna byla nanesena na horní stranu a druhá na stranu spodní. Textura horní strany byla jednoduchá. Jednalo se o tmavě zelený čtverec, který měl dva zářezy a okraj tohoto čtverce byl zvýrazněn světlejší zelenou barvou. Dále zde bylo několik žlutých teček a v jednom rohu žlutý trojúhelník. Spodní textura byla mnohem složitější. Základ byl stejný zelený čtverec se světlejším okrajem. Dále však bylo nutné znázornit všechny dotykové plošky procesoru, které pokrývaly značnou část z jeho celkové plochy. Tyto plošky mají tvar obdélníku s různě zaoblenými rohy. Tyto plošky jsou rozloženy v jednotlivých řadách a bylo je potřeba zobrazit podle toho, jak jsou umístěny na procesoru. V některých místech, jako jsou rohy, tyto plošky nejsou. Textury proce-

soru jsou zobrazeny na následujícím obrázku. Vlevo je textura určená pro horní stranu procesoru a vpravo je textura po spodní stranu.



Obr. 10 Textury procesoru

Další texturou, kterou bylo potřeba vytvořit, byla textura paměti RAM. V tomto případě stačilo vytvořit pouze jednu texture, která byla použita na obě strany této paměti. Zde bylo nejprve nutné nastavit barvu této paměti, která byla opět tmavě zelená. Dále bylo potřeba znázornit jednotlivé kontakty této paměti. Tyto kontakty byly tvořeny žlutými obdélníkovými ploškami podél jedné strany. Tato řada byla rozdělena mezerou na dvě části. Nakonec již stačilo vytvořit několik skupin teček a textura byla v dostatečné míře hotová. Tuto texture lze vidět na následujícím obrázku.



Obr. 11 Textura paměti RAM

Poté byly vytvořeny texture základní desky. Pro základní desku bylo potřeba vytvořit texture pro horní i spodní stranu. Přesněji řečeno, pro vytvoření texture horní strany, byla použita texture pro stranu spodní, která byla pouze doplněna o některé prvky. Základem texture byl hnědý obdélník, na kterém byly umístěny skupiny teček. Takto vypadala texture pro spodní stranu. Ta byla pouze doplněna

arance příslušného objektu. K připojení textury stačí tomuto uzlu přidat jako potomka uzel *ImageTexture* a jako parametr tohoto uzlu uvést adresu této textury. V ukázce níže je popsáno připojení horní textury procesoru.

```
# ukázka připojení horní textury procesoru
appearance Appearance {
  material Material { diffuseColor 0 0.5 0 }
  texture ImageTexture {
    # umístění textury
    url "textura_processor.gif"
  }
}
```

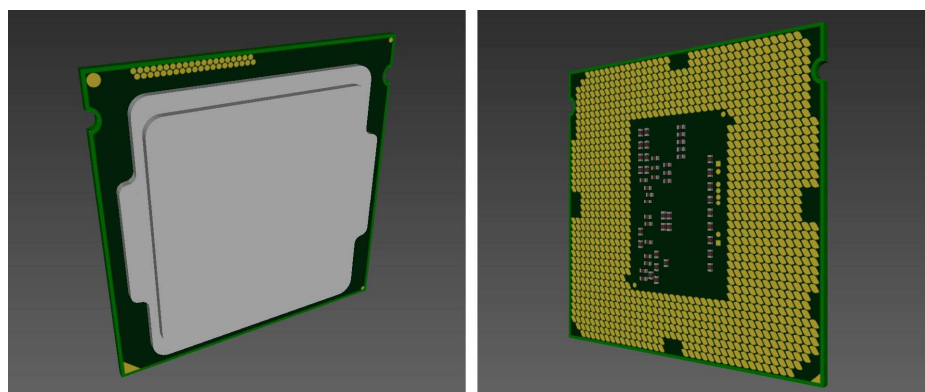
6.2 Tvorba jednotlivých modelů

Jednotlivé části modelů byly vytvořeny v samostatných souborech. Při tvorbě modelů bylo dbáno na to, aby byly využity co nejvíce možnosti samotného jazyka VRML. Jednotlivé dílčí části byly vytvářeny jako prototypy, aby je poté bylo možné načíst do společného modelu. Níže bude popsáno vytváření vybraných částí.

6.2.1 Procesor

Při tvorbě procesoru bylo nutné jeho jednotlivé části vymodelovat pomocí programu Rhinoceros. Procesor sice vypadá jednoduše, ale je na něm spousta zaoblených ploch a hran. Tyto zaoblené části nebylo možné jednoduše popsat pomocí jazyka VRML.

Jelikož bylo zapotřebí docílit dostatečných detailů, zabíraly jednotlivé zaoblené části po převedení do VRML značné množství místa. Z těchto důvodů byly ve značné míře použity příkazy DEF a USE, pomocí kterých byly vytvořeny kopie těchto částí. Těmito kopírovanými částmi byly zaoblené části. Jednalo se hlavně o rohy. Svou velikostí tento model patří mezi největší. Výsledný model procesoru lze vidět na následujícím obrázku.



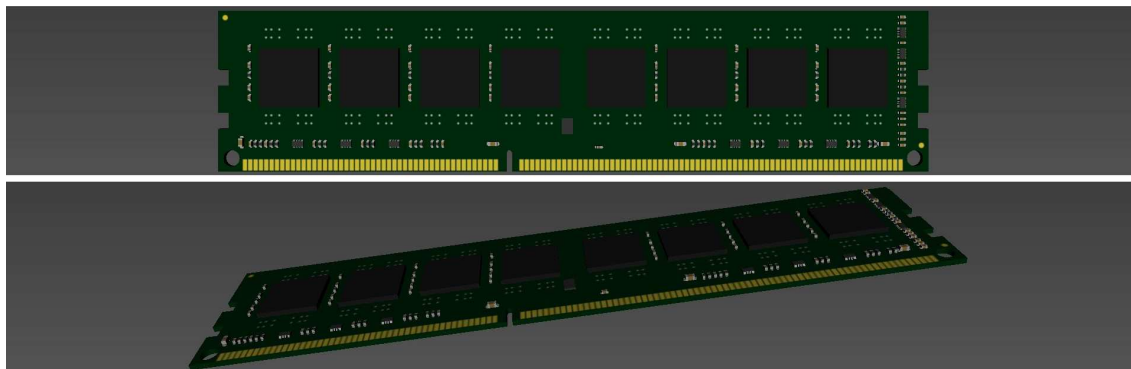
Obr. 14 Procesor

6.2.2 Paměť RAM

Tělo paměti RAM obsahuje několik zářezů a dvě kulaté díry. V případě že by tam nebyly tyto dvě díry, by bylo možné toto tělo jednoduše popsat pomocí jazyka VRML. Ovšem kvůli těmto dírám bylo nutné toto tělo paměti vymodelovat v programu Rhinoceros a do VRML je exportovat.

Jelikož jsou zářezy po stranách paměti stejné, byly vytvořeny kopírováním jedné předlohy. Obdobně byly vytvořeny jednotlivé díry a obě boční strany.

Paměť RAM na svém těle také obsahuje velké množství drobných součástek. Nejprve byly vytvořeny prototypy jednotlivých typů součástek. Poté bylo nutné tyto prototypy umístit na povrch paměti. Zde bylo nutné stanovit souřadnice jednotlivých součástek. Střed modelu byl umístěn v samotném středu paměti. Díky vhodným fotografiím a informacím o středu modelu bylo toto umístění součástek poměrně jednoduché. Fotografie byla nejprve otevřena v grafickém editoru tak, aby střed fotografie byl umístěn na počátek souřadné soustavy v tomto editoru a poté již pouze stačilo umístit kurzor v tomto editoru nad jednotlivé součástky a takto získané souřadnice jednotlivých součástek použít pro umístění součástek v modelu. Model paměti je zobrazen na následujícím obrázku.

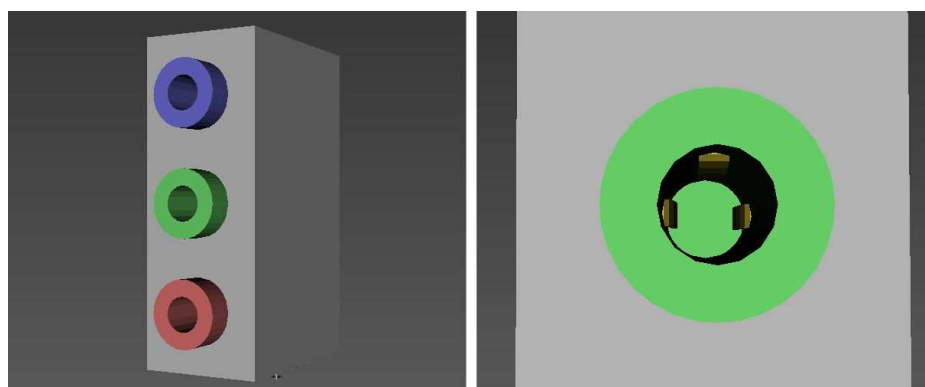


Obr. 15 Paměť RAM

6.2.3 Audio konektor

Tvorba audio konektoru byla z části popsána při popisu možností využití prototypů a příkazů DEF a USE. Pro modelování tohoto konektoru byl použit program Rhinoceros. Nejprve byla vymodelovaná zděř, která byla rozdělena na čtvrtiny a tato čtvrtina byla vyexportovaná do VRML. Ve VRML za využití příkazů DEF a USE byly vytvořeny další kopie této části, které byly pootočeny kolem středu samotné zděře. Tím vznikla celá zděř. Zděř pak byla doplněna o vnitřní kontakty. Opět byl vytvořen pouze jeden a nakopírován na další místa.

Takto vytvořená zděř byla vložena do těla prototypu a doplněna o potřebné parametry. Parametry zděře umožnily změnu její barvy a libovolně ji umístit na potřebné místo. Pomocí tohoto prototypu byla tato zděř použita třikrát, pokaždé však s jinou barvou. Nakonec stačilo doplnit samotné tělo audio konektoru a model byl hotový. Tento model lze vidět na obrázku níže.

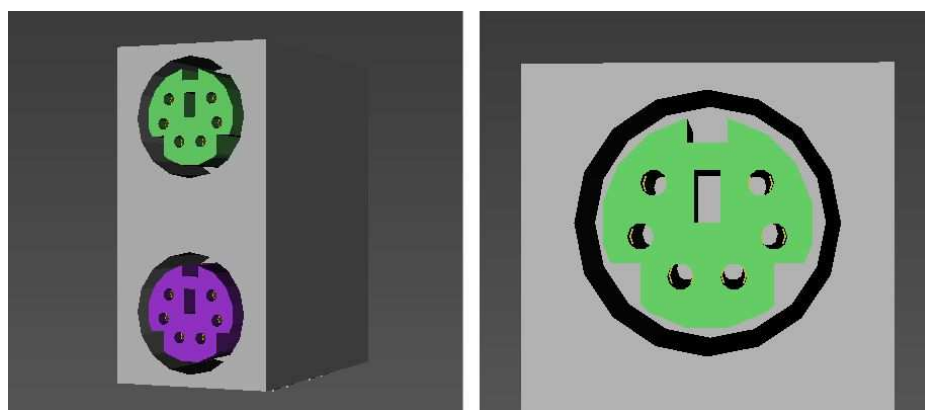


Obr. 16 Audio konektor

6.2.4 Konektor PS/2

K vytvoření tohoto konektoru byl použit podobný postup jako u audio konektoru. Nejprve byl vymodelován samotný konektor, který byl rozdělen na poloviny. Vyexportovaná byla polovina tohoto konektoru, ale pouze přední a boční plochy, a opět byla vytvořena kopie těchto ploch. Dále bylo nutné doplnit jednotlivé díry s kontaktními ploškami. K tomuto účelu byla vyexportována opět pouze jedna z těchto děr a pomocí příkazů DEF a USE byly vytvořeny kopie této díry.

Z této definice konektoru byl opět vytvořen prototyp, který byl použit dvakrát. Nakonec bylo doplněno tělo tohoto konektoru a konektor byl hotový. Ukázka tohoto modelu je vidět na následujícím obrázku.



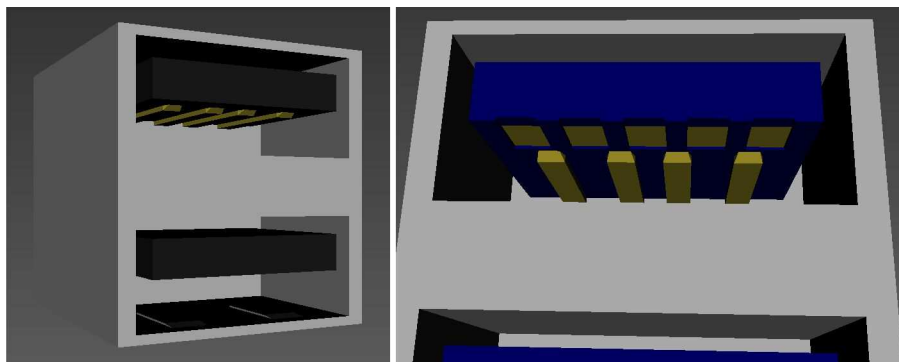
Obr. 17 Konektor PS/2

6.2.5 USB konektor

Na vybrané základní desce se nachází USB 2.0 a USB 3.0 konektor. Tělo těchto konektorů je stejné a tyto konektory se liší pouze v množství kontaktních plošek a v barvě. Tělo těchto konektorů bylo popsáno pomocí jednotlivých ploch přímo v jazyku VRML.

Dále bylo nutné popsat jednotlivé konektory. Ty byly popsány částečně jazykem VRML a část byla vymodelována v programu Rhinoceros. Základem konektoru

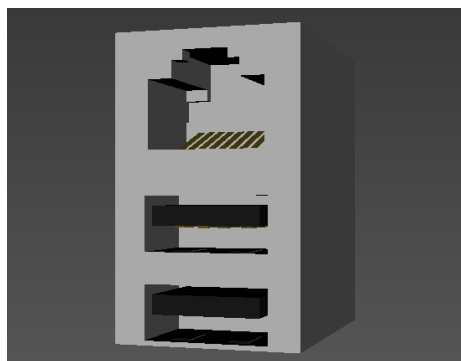
USB 2.0 je plocha ve tvaru kvádru, která je doplněná o jednotlivé kontaktní plošky. Konektor USB 3.0 má mimo čtyř kontaktních plošek, které jsou i u USB 2.0, ještě dalších 5 kontaktních ploch. Tyto plochy jsou zapuštěné do samotného konektoru. Z těchto důvodů bylo nutné tuto plochu, kterou v případě USB 2.0 tvořil pouze jeden kvádr, rozdělit na více částí a poté doplnit zmíněné plošky. Výsledné modely těchto konektorů lze vidět na obrázku níže. Vlevo je umístěn konektor USB 2.0 a vpravo konektor USB 3.0.



Obr. 18 Konektory USB 2.0 a USB 3.0

6.2.6 Síťový konektor

Základem tohoto konektoru je konektor USB 2.0, nad kterým je ještě umístěn samotný síťový konektor. Tvar tohoto konektoru byl vymodelován pomocí programu rhinoceros. Dále do tohoto konektoru byly doplněny jednotlivé kontaktní plošky pomocí vytváření kopií z jednoho vzoru. Tento konektor lze vidět na obrázku níže.



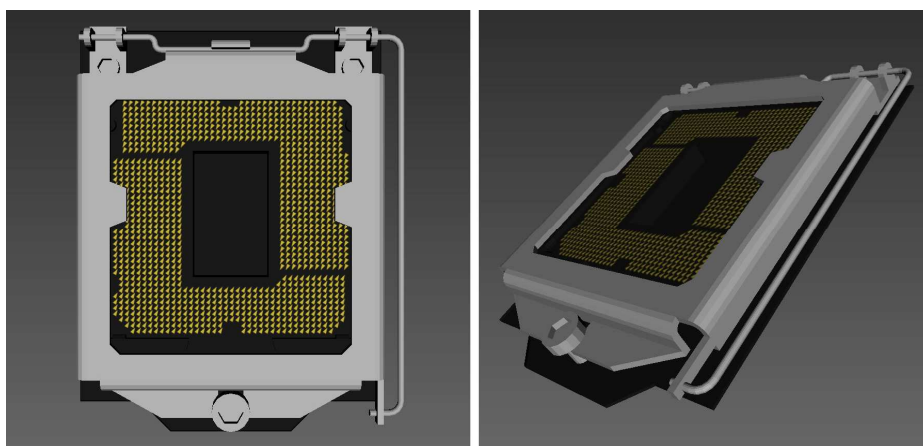
Obr. 19 Síťový konektor

6.2.7 Patice procesoru

Tvorba této patice byla provedena v programu Rhinoceros. Model této patice je velmi složitý a obsáhlý. I přes značné použití prototypů a příkazů DEF a USE se jedná o největší ze všech modelů, jehož velikost je mnohem větší než velikosti ostatních modelů. Tato velikost je způsobena velkým množstvím zaoblených hran a kulatých ploch.

Samotné tělo bylo velmi jednoduché. Jednalo se o spojení několika kvádrů, které byly pouze doplněné o zkosené plochy. Na horní stranu těla této patice byla nanесena textura.

Složitým objektem na tomto modelu byl upínací mechanismus, který má mnoho zaoblených hran a pro upínání je použit kulatý drátek. Pro tvorbu celého upínacího mechanismu byl použit program Rhinoceros, ve kterém byly vymodelovány jednotlivé části. Největší problém nastal při modelování a exportu drátku sloužícího pro zajištění upínacího mechanismu. Jednotlivé jeho ohyby bylo nutné popsat dostatečně podrobnou polygonovou sítí. Tato síť zabírala velké množství místa a během převodu byla vytvořena chybně. Proto bylo nutné tyto ohyby vymodelovat znovu ale tentokrát jinou metodou. Sice i potom tato část zabírala pořád velké množství místa, což bylo způsobeno velkým množstvím polygonů, pomocí kterých byla popsána, ale byla již zobrazena dobře. Ukázkou celé patice lze vidět na následujícím obrázku.

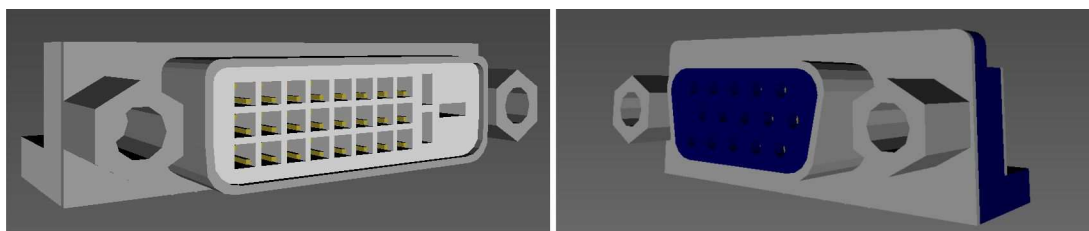


Obr. 20 Patice procesoru

6.2.8 Konektory DVI a D-Sub

Tělo obou dvou konektorů je pro popis velmi jednoduché. Popis těchto těl je možný pomocí několika kvádrů v jazyku VRML. Těžší částí na popis jsou však samotné kontaktní plochy. Tyto plochy jsou jednak zaoblené a obsahují velké množství děr.

V případě tvorby těchto částí bylo nutné použít program Rhinoceros. Po jejich vymodelování byly vyexportovány jejich venkovní plochy. Jelikož v případě konektoru D-Sub jsou všechny díry stejné, stačilo vyexportovat pouze jednu z nich a dále vytvořit pouze kopie této díry. U konektoru DVI jsou tři typy děr. Vždy bylo nutné vytvořit jednu z těchto děr a tu dále použít pro případnou tvorbu kopií. Také bylo důležité vhodně rozdělit a vyexportovat společně s dírami okolní plochu těchto děr tak, aby nemusela být exportována zvlášť, což by zabralo mnoho místa. Tyto zmíněné konektory lze vidět na obrázku níže. Vlevo je umístěn konektor DVI a vpravo konektor D-Sub.



Obr. 21 Konektory DVI a D-Sub

6.2.9 Sběrnice PCI-Express

Tato sběrnice má jednoduchý tvar, ale obsahuje velké množství malých kontaktních plošek. Celá tato sběrnice byla vytvořena pouze pomocí popisu jazykem VRML. Samotný popis byl však velmi složitý. Bylo zde nutné složitě přepočítávat vzdálenosti posunutí a dobře rozvrhnout rozložení této sběrnice na jednotlivé části.

Nejprve byla vymodelovaná vnější část těla z několika kvádrů. Poté však bylo nutné dobře rozvrhnout tvorbu jednotlivých malých kontaktních plošek tak, aby jich byl správný počet v jednotlivých částech, a aby jejich velikost byla stejná.

Po správném navržení těchto plošek byla vytvořena jedna taková ploška i s částí okolních příček, které tvoří mezeru mezi těmito kontakty. Tato ploška byla umístěna do těla prototypu a byla využita k vytváření kopií. Aby však nebylo nutné popsat toto velké množství plošek jednotlivě, byly vytvořeny jednotlivé prototypy se skupinami těchto plošek tak, aby bylo možné pomocí těchto skupin snadno popsat potřebné množství těchto kontaktů. Nejprve byla vytvořena skupina pěti kontaktů tak, že v ní byl pětkrát zavolán prototyp jednoho kontaktu. Tato skupina byla dále použita k vytvoření skupiny deseti kontaktů. Poté byly použity tyto skupiny k vytvoření potřebného množství kontaktů této sběrnice. Výsledný model sběrnice PCI-Express je možné vidět na následujícím obrázku.



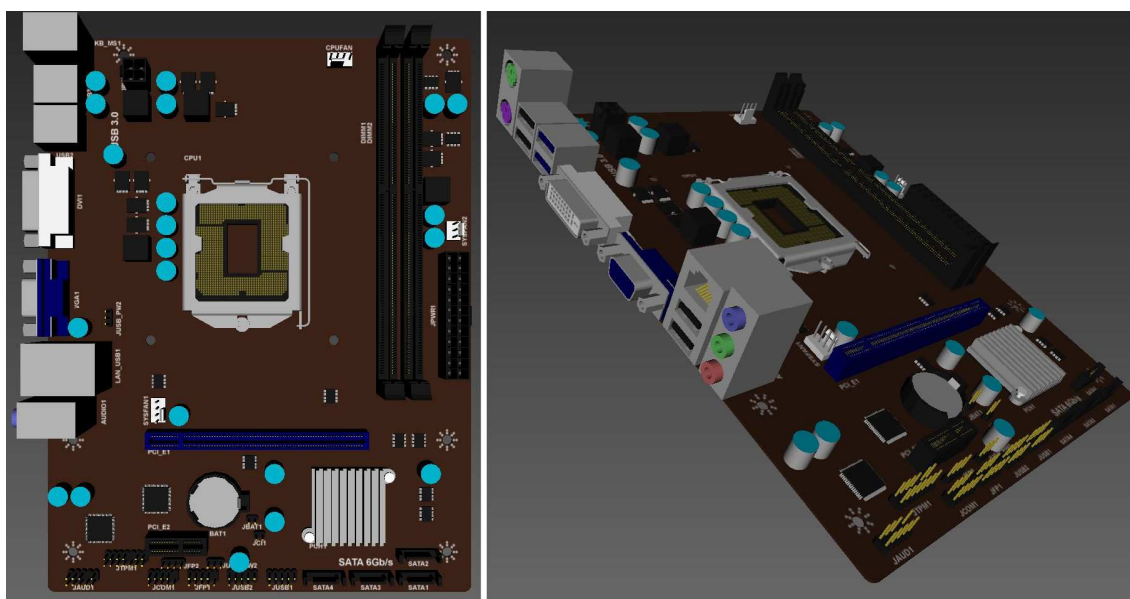
Obr. 22 Sběrnice PCI-Express

6.2.10 Ostatní modely

Pro tvorbu ostatních konektorů, patič a těla samotné základní desky byly využity podobné postupy, které byly popsány při tvorbě výše zmíněných modelů. Některé z nich bylo možné popsat pouze jazykem VRML a jiné byly ze značné části vymodelovány v programu Rhinoceros. Ukázkové obrázky ostatních modelů lze vidět v příloze.

6.3 Sestavení trojrozměrného modelu z částí

Po vytvoření jednotlivých konektorů a všech potřebných částí bylo možné jednotlivé části spojit do výsledného modelu. Nejprve bylo potřeba zkompletovat samotnou základní desku. Využitím prototypů byly jednotlivé její části načteny v jednom souboru a zde také pomocí změny své polohy a rotace umístěny na potřebné místo. Poté byla na samotné tělo základní desky nanesena textura. Dále byla základní deska doplněna o základní modely některých elektronických součástek, aby její tělo nebylo zcela jednoduché. Takto sestavená základní deska byla použita při tvorbě výsledného modelu. Samotnou základní desku lze vidět na následujícím obrázku.



Obr. 23 Základní deska

Jednotlivé modely komponent, kterými jsou základní deska, paměť RAM a procesor, byly použity pro tvorbu výsledného modelu. Do nového souboru byly tyto modely načteny pomocí prototypů a sestaveny tak, jako by se jednalo o skutečnou počítačovou sestavu.

6.3.1 Doplnění animace

Tento model byl sestaven jako výchozí, který uvidí uživatel jako první a ze kterého by bylo možné zobrazit jednotlivé části. Z tohoto důvodu byl model doplněn o úvodní animaci, která znázorňovala sestavení tří základních komponent do celku.

Po načtení souboru s modelem uživatel vidí pouze vzdálený model základní desky. Základní deska se nejprve k uživateli přiblíží. Poté se otevře patice procesoru a směrem od uživatele je do této patice vsunut samotný procesor. Poté je patice procesoru opět uzavřena. Po tomto vložení procesoru a uzavření patice se roze-

vřou zajišťovací části slotu pro umístění paměti RAM a samotná paměť RAM se také vsune do tohoto slotu. Na závěr animace dojde k zajištění paměti RAM.

Samotná animace je popsána pomocí jazyka VRML. Jazyk VRML umožňuje vytvářet animace metodou klíčování (Zrzavý, 1999) a tím plynule měnit vlastnosti objektů. Klíčování spočívá ve vytvoření tzv. klíčových snímků. Tyto klíčové snímky obsahují polohu, rotaci, barvu a jiné vlastnosti objektů, které jsou pro samotnou animaci důležité. Podle takto stanovených klíčových snímků a na základě zadané časové funkce počítač vypočítá jednotlivé interpolované snímky této animace. Způsob této interpolace určuje tvář celé animace. Nemusí se jednat pouze o animaci pohybu, ale změnit lze jakýkoliv parametr. Může se jednat i o plynulou změnu barvy tělesa (Žára, 1998).

6.3.2 Návrh aplikace

Důležitým krokem byl samotný návrh aplikace a vhodné propojení jednotlivých modelů. Také bylo nutné vhodným způsobem doplnit jednotlivé části modelu o popis.

Jako první metodou doplnění popisu se nabízela možnost vložit popis přímo do samotného modelu pomocí příkazů jazyka VRML a tím rozdělit okno s modelem na dvě části. V jedné části by byl samotný model a v druhé části popis. Toto řešení se ukázalo jako nevhodné. I přes nutnost doplnění různých událostí reagujících na otočení modelu a propojení těchto událostí s prvkem popisu tak, aby se text nenatáčel se samotným modelem a byl stále natočen k uživateli, měla tato metoda mnoho dalších nedostatků, díky kterým bylo od této realizace upuštěno. Mezi tyto nedostatky patřila hlavně práce s textem a nedostatky textového popisu. Nebylo možné tyto popisy doplnit o české znaky a bylo nutné formátování textu a zalamování řádků řešit ručně. Další nevýhodou bylo, že tento text je určen k popisu větší velikosti. Když bylo písmo zmenšené na standardní velikost, byl text špatně čitelný.

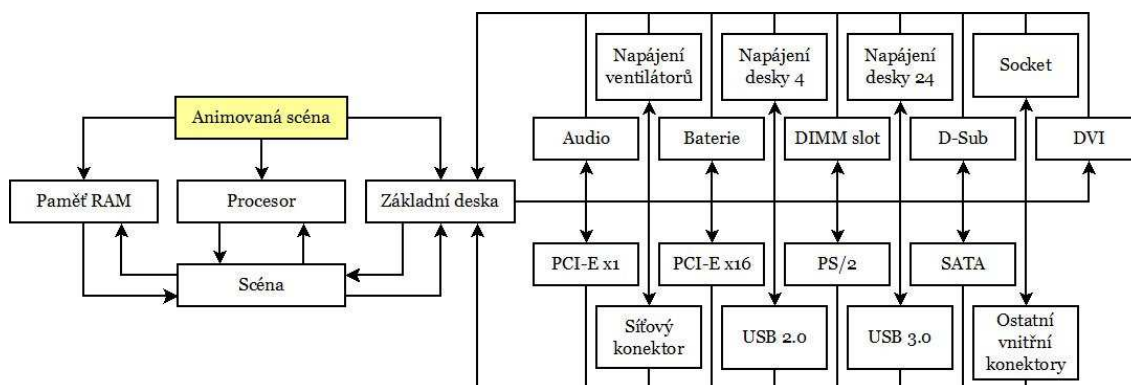
Druhou možností a zároveň možností, která byla ve výsledné aplikaci použita, bylo rozdělení již samotné internetové stránky této aplikace na dvě části. V jedné bylo umístěno okno s modelem a v druhé byl popis pomocí internetové stránky. Bylo tedy nutné zajistit, aby se při vyvolání konkrétní události, jako bylo kliknutí myši na určitou komponentu, vyslala zpráva o tom, že má být zobrazen daný popis. To bylo vyřešeno pomocí odkazů v samotném VRML. Tyto odkazy slouží k načtení jiných modelů nebo k přechodu do jiné části stejného modelu, ale také umožňují zobrazení či přesměrování stránky.

Výše zmíněným řešením bylo tedy stanoveno, že aplikaci bude nutné rozdělit na dvě části. Na část s modelem a část určenou k zobrazování popisů. Nyní bylo zapotřebí stanovit samotnou strukturu aplikace a určit co a kdy bude možné zobrazit a kam bude z jednotlivých modelů možné přejít.

Struktura aplikace byla tedy navržena následovně. Nejprve se po samotném načtení webové stránky s těmito modely uživateli zobrazí úvodní animovaná scéna. Z této scény může uživatel dále přejít na samostatné zobrazení kterékoli ze tří modelovaných komponent. Z tohoto zobrazení může uživatel přejít zpět na scé-

nu, ve které jsou zobrazeny všechny komponenty. Tato scéna se však od původní liší tím, že již není animovaná. Z této scény může uživatel opět přejít na kteroukoliv z těchto komponent a zpět na tuto scénu.

Dalším předpokladem bylo, aby uživatel mohl z modelu základní desky zobrazit samostatně jednotlivé konektory a patice, které jsou na základní desce zobrazeny. Poté aby se z těchto jednotlivých modelů mohl vrátit zpět na zobrazení základní desky. Samotná struktura této aplikace je vidět na následujícím diagramu. Žlutě podbarvené pole tohoto diagramu znázorňuje onu základní scénu.



Obr. 24 Struktura aplikace

6.3.3 Vytvoření aplikace a doplnění interaktivních prvků

Dle navržené struktury aplikace byla postupně celá aplikace sestavena. Aplikace je dostupná formou internetové stránky. Jak již bylo zmíněno, tato stránka je rozdělena na dvě části. Levé okno slouží k zobrazení modelů a pravá část k zobrazení popisů k jednotlivým částem. Každý model je reprezentován jedním zdrojovým souborem a jednotlivé popisy jsou uloženy také samostatně.

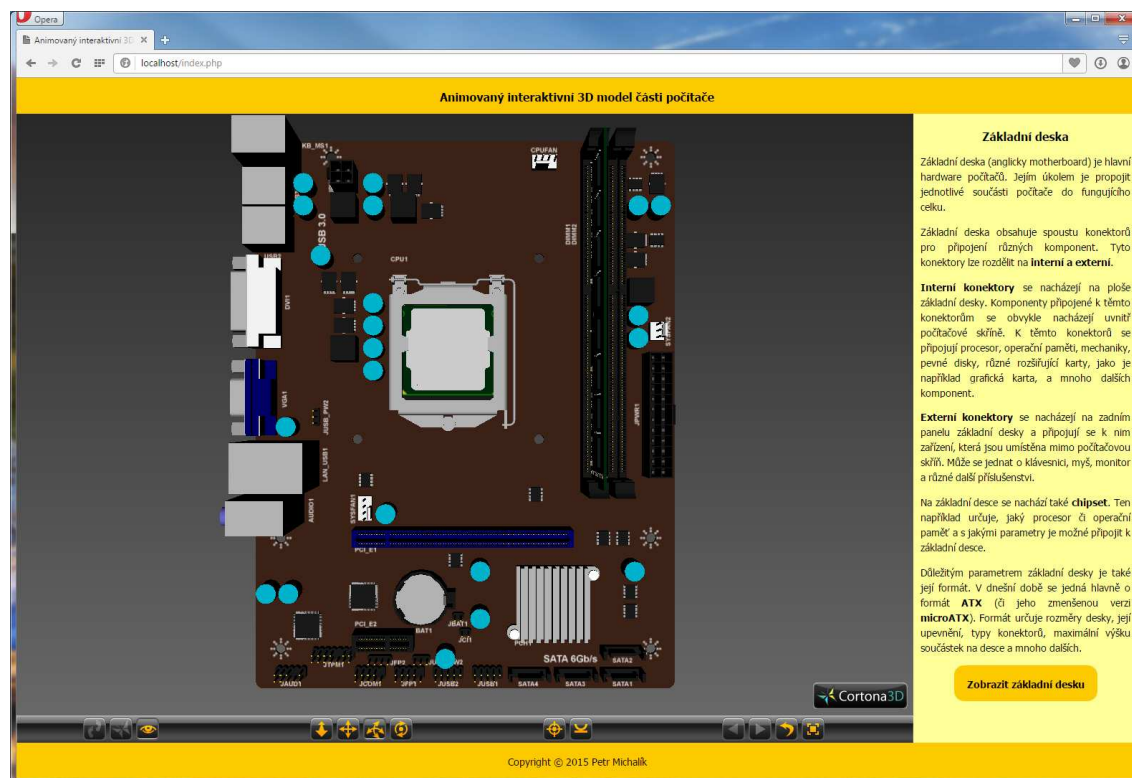
Internetová stránka byla vytvořena pomocí jazyka HTML, PHP a CSS. Základem stránky je zdrojový kód HTML doplněný o dynamické prvky a některé funkce PHP. Pomocí CSS jsou definovány styly pro jednotlivé části a prvky stránky.

Aby bylo docíleno zobrazení jednotlivých popisů, které byly umístěny v jednotlivých souborech, a modelů v potřebné části okna, bylo zajištěno předávání těchto hodnot, o tom co se má v které části zobrazit, pomocí parametrů v adrese odkazu. Z tohoto důvodu bylo také nutné zajistit kontrolu, zda soubor zadaný jako parametr vůbec existuje a v případě chybného zadání stanovit který soubor bude otevřen.

Po vytvoření ovládacího rozhraní a zajištění zobrazení základního modelu, bylo nutné propojit jednotlivé modely mezi sebou a se samotnými popisy. Z tohoto důvodu bylo potřeba v jednotlivých zdrojových souborech modelů definovat odkazy na potřebný popis, který se poté zobrazil v části pro popisy. Zobrazení těchto popisů bylo vyvoláno jako reakce na uživatelův klik myši na danou část komponenty. Níže v ukázce je uveden příklad zobrazení popisu procesoru, který se vyvolá po kliknutí na procesor v úvodní scéně.

```
# příkaz Anchor slouží k vytvoření odkazu
Anchor{
  # v části parameter se udává kam má být odkaz načten
  parameter ["Target=popis"]
  # url obsahuje adresu načítaného souboru nebo stránky
  url "../popisy/popis_procesor.php"
  # v části children se uvádějí prvky,
  # které slouží k přechodu na uvedený odkaz
  children[
    procesor{}
  ]
}
```

Dále bylo nutné v jednotlivých souborech s popisy vytvořit tlačítka pro možnost zobrazení konkrétního modelu. Tato tlačítka obsahovala formou HTML odkazu adresu dané stránky, doplněnou o parametr s názvem konkrétního souboru s modelem, který se má zobrazit. Pomocí proměnné v PHP pak bylo zajištěno, aby se tento model zobrazil v konkrétní části stránky. Tím byla zajištěna interaktivita celého modelu, která se řídí dříve navrženou strukturou. Ukázka navrženého rozhraní je zobrazena na obrázku níže.



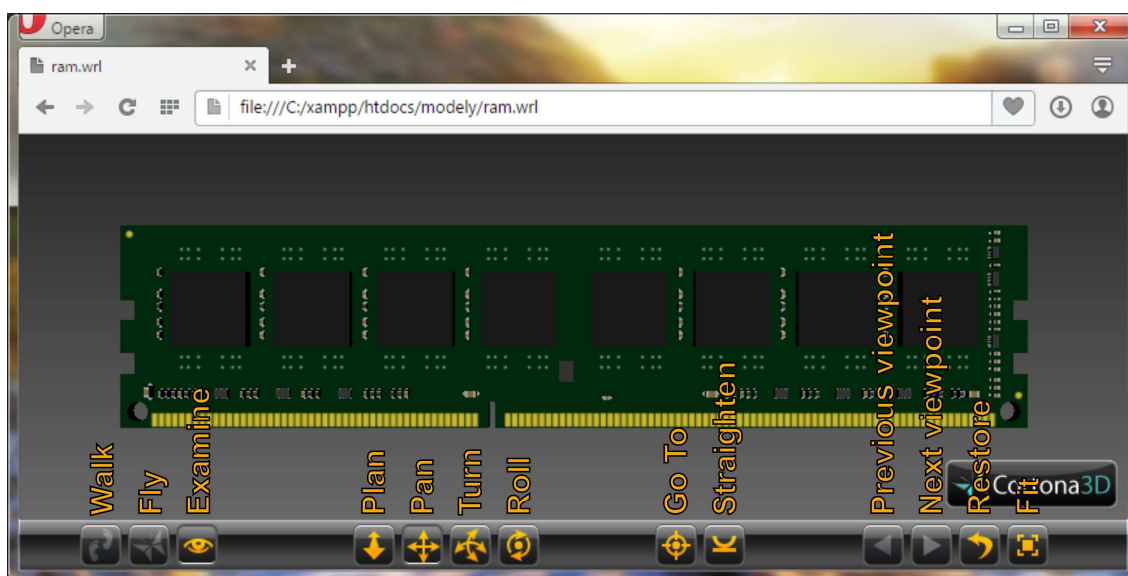
Obr. 25 Navržená aplikace

Na předchozím obrázku je zobrazena úvodní scéna. Když zde uživatel klikne na paměť RAM, procesor nebo na tělo základní desky, zobrazí se popis k dané části. Pod tímto popisem je tlačítko, které umožní zobrazit tu danou část samostatně.

Pokud by uživatel takto zobrazil například procesor, zobrazil by se mu pouze model procesoru. Při kliknutí na model procesoru by se mu opět zobrazil popis k této komponentě a pod tímto popisem by se zobrazilo tlačítko určené pro přechod na scénu, která opět obsahuje všechny komponenty.

6.3.4 Použití a popis zásuvného modulu

Pro zobrazení modelů je nutné mít ve webovém prohlížeči nainstalovaný potřebný zásuvný modul. Těchto modulů je velké množství. Mnoho z nich lze stáhnout a nainstalovat zdarma. Při tvorbě této práce byl využit modul *Cortona3D*. Ukázka prostředí tohoto modulu je uvedena na obrázku níže.



Obr. 26 Ukázka prostředí zásuvného modulu Cortona3D

Tento modul se skládá ze dvou částí. Hlavní částí je samotné okno pro zobrazení modelů. Druhou částí je lišta s tlačítky, umožňující volby prohlížení modelu. Na této liště je umístěno 13 tlačítek, které budou dále postupně popsány.

První skupina tlačítek slouží k volbě metody procházení scény. Pomocí nich lze vybrat jednu z metod *WALK*, *FLY*, *EXAMINE*. První dvě z těchto metod, jak již bylo zmíněno dříve, však v této aplikaci nelze zvolit.

Druhá skupina tlačítek slouží k výběru pohybu Avatara. Tlačítko *Plan* slouží k pohybu vpřed a vzad. Tlačítko *Pan* je určené k posunu nahoru a dolů či doleva a doprava. Tlačítko *Turn* umožňuje libovolné otáčení okolo středu scény. Poslední tlačítko této skupiny, kterým je tlačítko *Roll*, slouží k náklonu do stran.

Následujícími tlačítky jsou tlačítka *GoTo* a *Straighten*. První z těchto tlačítek slouží k přesunu Avatara k vybranému objektu. Druhé tlačítko slouží k navrácení do základní polohy.

Poslední skupina tlačítek obsahuje čtyři tlačítka. První dvě z těchto tlačítek, kterými jsou tlačítka *Previous viewpoint* a *Next viewpoint*, slouží k přepínání pře-

dem nadefinovaných stanovišť ve virtuálním světě. Další tlačítkem je tlačítko *Restore*. Toto tlačítko slouží k přechodu na počáteční stanoviště. Posledním tlačítkem je tlačítko *Fit*. Pomocí tohoto tlačítka lze umístit Avatara tak, aby v jeho zorném poli byly všechny objekty virtuálního světa.

7 Závěr

V této práci byly popsány možnosti tvorby trojrozměrných modelů. Na základě hodnocení jednotlivých metod tvorby trojrozměrných modelů bylo vybráno vhodné řešení, které bylo dále použito k vytvoření modelů počítačových komponent. V práci byly dále popsány možnosti prezentace těchto modelů a některé možnosti využití trojrozměrných modelů v dnešní době.

Cílem práce bylo vytvoření trojrozměrných modelů počítačových komponent a doplnění těchto modelů o animaci a vhodné interaktivní prvky. Před samotnou tvorbou těchto modelů byly stanoveny požadavky na funkčnost a možnosti distribuce těchto modelů. Při tvorbě modelů proto bylo dbáno na dodržení těchto požadavků.

Po vytvoření jednotlivých komponent byly tyto komponenty spojeny ve výsledný model a tento model byl začleněn do internetové stránky, která umožnila snadnou prezentaci modelu na internetu.

Díky vhodnému vytvoření textur, optimalizaci jednotlivých modelů a využívání možností prototypů je výsledný model velmi malý. Díky tomu by se měl model snadno a rychle zobrazovat i lidem s pomalým internetovým připojením a to i na starších a méně výkonných počítačích.

7.1 Využití modelu

Výsledný model byl vytvořen pro výuku počítačových komponent a jejich částí. Proto při jeho tvorbě bylo hlavně dbáno na detailní provedení důležitých a funkčních částí těchto komponent. Model by měl být přístupný vyučujícímu jako názorná pomůcka při výuce. Měl by však být dostupný i jednotlivým studentům, kteří se budou chtít blíže seznámit s jednotlivými počítačovými komponentami, ale nemají možnost si tyto komponenty pro tyto účely pořídit.

7.2 Možnosti dalšího rozšíření práce

Veškeré části modelů jsou popsány pomocí jazyka VRML. Díky tomu je možné tyto modely snadno upravit a doplnit pomocí obyčejného textového editoru. Komponenty a jednotlivé části základní desky jsou popsány v samostatných souborech. Jakožto hotové části lze tyto modely snadno načíst a využít při tvorbě nových komponent.

Výsledná stránka umožňuje zobrazení popisů k jednotlivým komponentám. S rozvojem jednotlivých částí a jejich možností by bylo vhodné tyto popisy upravit či doplnit. Jelikož jsou tyto popisy ve formě jednotlivých internetových stránek, které se pouze načítají v dané části okna, lze tyto popisy snadno upravit a libovolně doplnit.

Dále by bylo možné doplnit modely konektorů a patič, které se na modelované desce nevyskytují, případně různé alternativy těchto konektorů, a ze všech těchto

konektorů vytvořit knihovnu, která by umožňovala snadné a rychlé vytváření různých typů základních desek pouhým využitím jednotlivých částí z této knihovny.

Také by bylo možné doplnit modely dalších počítačových komponent, jako je grafická karta, napájecí zdroj, pevné disky nebo další rozšiřující desky. Jednotlivé části by bylo možné navzájem propojit a tím vytvořit model celé počítačové sestavy, kde by bylo možné vytvořit výukovou animaci znázorňující sestavení počítače, přidělování paměti a další.

8 Literatura

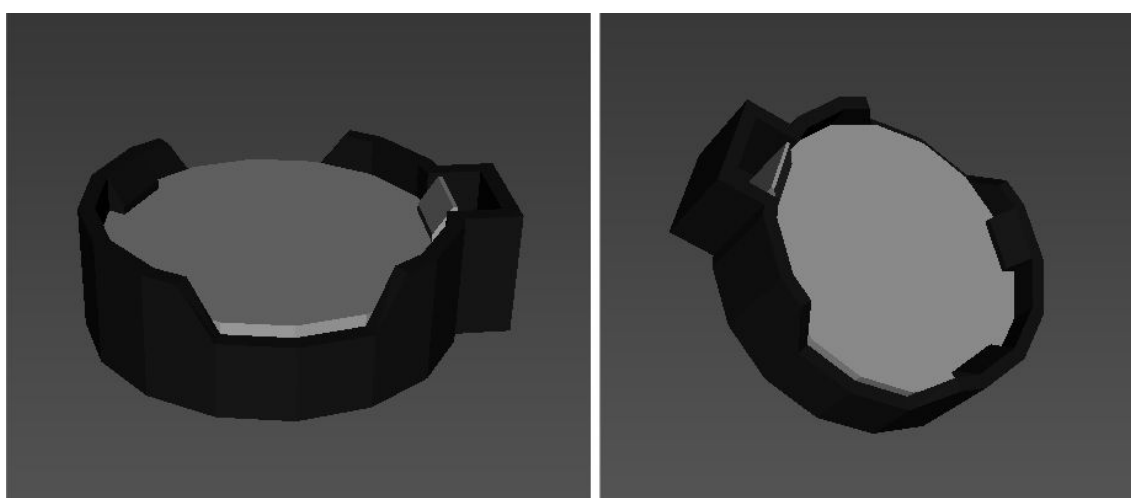
- CRONICASH, C. 2014, *Cronicash-Evga Classified* [on-line]. [cit. 2015-4-18], Dostupné na <https://3dwarehouse.sketchup.com/model.html?id=728d75f5d77fa782d8fc2ff09b558226>
- DEMBOWSKI, K. 2009, *Mistrovství v hardware*. 1. vyd. Brno: Computer Press, ISBN 978-80-251-2310-2.
- GARO3D, 2015, *Bukobot V2* [on-line]. [cit. 2015-4-10], Dostupné na <http://garo3d.com/bukobot-v2>
- GRAVO, 2015, *Co je to gravírování* [on-line]. [cit. 2015-4-13], Dostupné na <http://www.gravo.cz/index.php?s=055076&uuzz=400322e7c07a44c3fcfec7a3f5914247>
- GRISHALUTSENKO, 2006, *Motherboard* [on-line]. [cit. 2015-4-8], Dostupné na <http://www.turbosquid.com/3d-models/3d-model-of-mainboard-gigabyte/326560>
- MCNEEL, R. 1993, *Rhinoceros: NURBS modeling for Windows, uživatelská příručka* [on-line]. [cit. 2015-1-20], Dostupné na http://home.zcu.cz/~lavicka/subjects/ITG/texty/Prirucka_Rhino3.pdf
- MIKAČ, I. 2008, *Učebnice SketchUp pro modeláře* [on-line]. [cit. 2015-3-11], Dostupné na <http://www.mikac.cz/3D-free-cad-sketchup-model-ucebnice.html>
- PARALLELGRAPHICS, 2015, *NavigationInfo* [on-line]. [cit. 2015-4-3], Dostupné na www.parallelgraphics.com/developer/products/cortona/extensions/navinfo
- POKORNÝ, P. 2009, *Blender: naučte se 3D grafiku*. 2. vyd. Praha: BEN – technická literatura, ISBN 978-80-7300-244-2.
- SIKDAR, N. 2014, *Evolution of Sketchup* [on-line]. [cit. 2015-3-15], Dostupné na <http://sketchup-ur-space.com/2014/jan/evolution-of-sketchup.html>
- SOFTONIC, 2015, *SketchUP* [on-line]. [cit. 2015-3-20], Dostupné na <http://google-sketchup.cs.softonic.com>
- TIŠNOVSKÝ, P. 2008, *XML + 3D = X3D* [on-line]. [cit. 2015-3-30], Dostupné na <http://www.root.cz/clanky/xml-3d-x3d>
- ZRZAVÝ, J. 1999, *VRML: Tvorba dokonalých www stránek*. 1. vyd. Praha: Grada, ISBN 80-7169-643-9.
- ŽÁRA, J. 1998, *Moderní počítačová grafika*. 1. vyd. Praha: Computer Press, ISBN 80-7226-049-9.
- ŽÁRA, J. 1999, *VRML 97: Laskavý průvodce virtuálními světy*. 1. vyd. Praha: Computer Press, ISBN 80-7226-143-6.

Přílohy

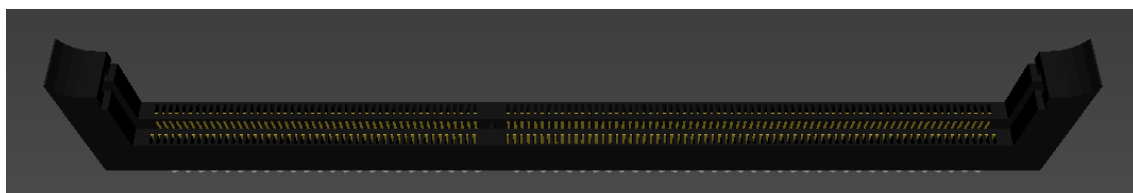
A Ukázky ostatních modelů



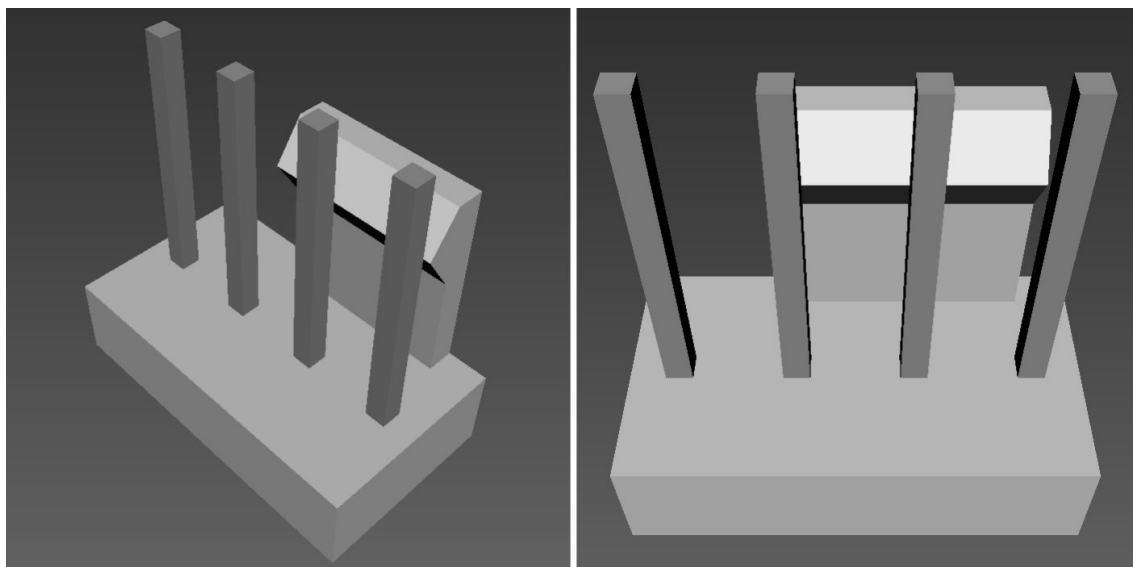
Základní deska



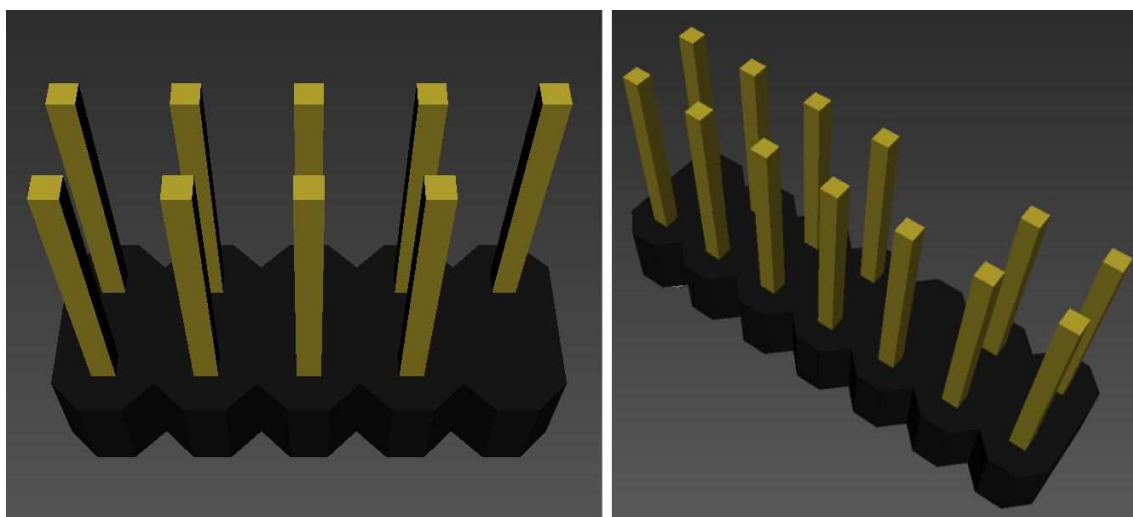
Baterie



Slot pro paměti RAM



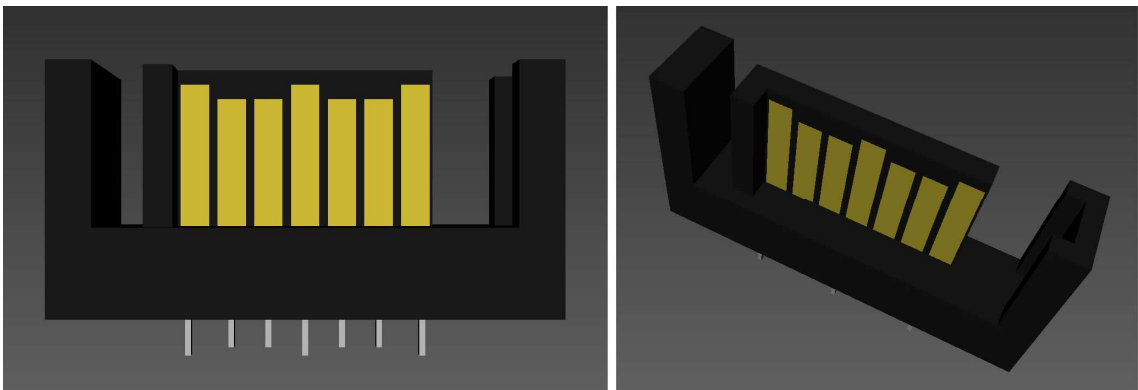
Konektor pro napájení ventilátorů



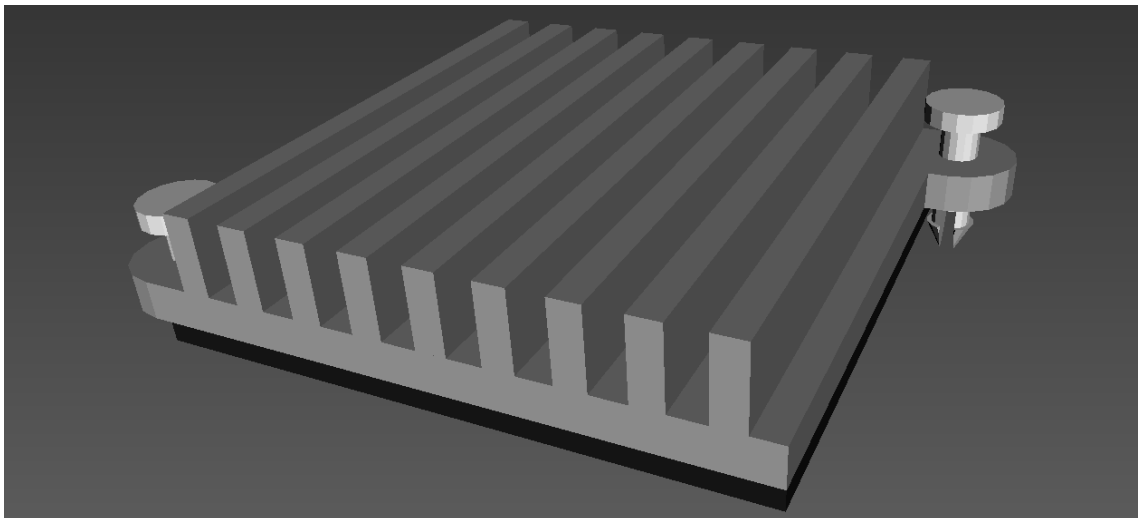
Konektory pro připojení prvků předního panelu počítače



Konektory pro napájení základní desky



SATA konektor



Chipset

B Seznam souborů na přiloženém CD

Přiložené CD obsahuje jednotlivé soubory výsledné aplikace. Aplikace je rozdělena na podadresáře. V kořenovém adresáři se nacházejí 3 soubory popisující samotnou úvodní stránku a její části. Dále se zde nachází 3 adresáře. V jednom adresáři je umístěný soubor se styly. V Dalším adresáři jsou umístěny jednotlivé modely a textury k těmto modelům a v posledním adresáři se nacházejí popisy k modelům. Stromová struktura aplikace je zobrazena níže.

- design
 - styly.css
 - tlacitko.jpg
- modely
 - audio.wrl
 - baterie.wrl
 - dimm.wrl
 - dsub.wrl
 - dvi.wrl
 - fan.wrl
 - front_panel.wrl
 - chipset.wrl
 - kontakty.wrl
 - napajeni.wrl
 - napajeni24.wrl
 - pcie.wrl
 - pciex16.wrl
 - procesor.wrl
 - ps2.wrl
 - ram.wrl
 - sata.wrl
 - scena.wrl
 - scena_start.wrl
 - socket.wrl
 - soucastky.wrl
 - textura_procesor.gif
 - textura_procesor_spodni.gif

- textura_ram.gif
- textura_socket.gif
- textura_zakladni_deska.gif
- textura_zakladni_deska_spodni.gif
- usb_sit.wrl
- usb2.wrl
- usb3.wrl
- zakladni_deska.wrl
- popisy
 - popis.php
 - popis_audio.php
 - popis_audio_zpet.php
 - popis_baterie.php
 - popis_baterie_zpet.php
 - popis_dimm.php
 - popis_dimm_zpet.php
 - popis_dsub.php
 - popis_dsub_zpet.php
 - popis_dvi.php
 - popis_dvi_zpet.php
 - popis_fan.php
 - popis_fan_zpet.php
 - popis_fp.php
 - popis_fp_zpet.php
 - popis_napajeni.php
 - popis_napajeni_zpet.php
 - popis_napajeni24.php
 - popis_napajeni24_zpet.php
 - popis_pciex1.php
 - popis_pciex1_zpet.php
 - popis_pciex16.php
 - popis_pciex16_zpet.php
 - popis_procesor.php
 - popis_procesor_zpet.php
 - popis_ps2.php

- popis_ps2_zpet.php
- popis_ram.php
- popis_ram_zpet.php
- popis_sata.php
- popis_sata_zpet.php
- popis_sit.php
- popis_sit_zpet.php
- popis_socket.php
- popis_socket_zpet.php
- popis_usb2.php
- popis_usb2_zpet.php
- popis_usb3.php
- popis_usb3_zpet.php
- popis_uvod.php
- popis_zakladni_deska.php
- popis_zakladni_deska_zpet.php
- hlavicka.php
- index.php
- pata.php