



UNIVERZITA KOMENSKÉHO V BRATISLAVE

**FAKULTA MATEMATIKY, FYZIKY
A INFORMATIKY**

KATEDRA INFORMATIKY

Virtuálne múzeum, Považské múzeum

BAKALÁRSKA PRÁCA

Michal Hudek

2007

Vedúci/školiťel' záverečnej práce

Doc. RNDr. Andrej Ferko, PhD.

Prehlasujem, že som prácu vypracoval samostatne
s použitím uvedených prameňov v zozname

.....
Michal Hudek

Obsah

1.	Prehľad	5
1.1.	Virtuálne prostredia	6
2.	Technológie 3D prostredí	9
2.1.	Reprezentácia objektov	9
2.2.	N-Patch	10
2.3.	Mapovanie textúr	11
2.4.	Filtrovanie textúr	12
2.5.	Antialiasing	14
2.6.	Osvetlenie	15
3.	Virtuálne múzeum	18
3.1.	Motivácia a ciele projektu	18
3.2.	Funkcionalita a riešenie	18
3.3.	Popis použitých formátov	20
3.4.	Vstupné dáta	23
3.5.	Použité aplikácie	24
3.6.	Tvorba modelu Považského múzea	26
4.	Záver	27
4.1.	Príloha CD	27
	Pramene	28

Abstrakt



Táto práca si dáva za cieľ popísať virtuálne prostredia, ich tvorbu a spôsob použitia na internete. Popisuje technológie používané pri tvorbe 3D prostredí. Reprézntácia modelov polygónmi, textúrovanie a techniky pri simulácii osvetlenia s dôrazom na ich využitie pri tvorbe reálnych modelov. Práca popisuje formáty využiteľné na prezentáciu virtuálnych svetov na internete a ich výhody. Poskytuje aj jednoduchý prehľad aplikácií použitých pri tvorbe s dôrazom na tie voľne dostupné.



The goal of this project is to describe virtual worlds, their creation and areas where they can be used. We describe as well technology used for creating 3D environments and representation of objects by polygons, texturing and lighting techniques to create more realistic results. Work also defines formats which are used for internet presentation of virtual worlds and their advantages. It also makes an overview of applications used in this project, free software is preferred.

1. Prehľad

Už dlhší čas sa ľudstvo snaží o uchovanie a rozširovanie informácií. Z počiatku to bolo len posunkami a gestami, neskôr prišla reč, písmo. Od klinového po latinku a postupom času prešlo obrovským vývojom aj médium na, ktoré sa kládli čoraz vyššie nároky. Je nemysliteľné, aby pri dnešných potrebách bežného človeka po informáciách používali kamenné doštičky či papyrusové zvitky. Potreba nových technológií je teda prirodzená a je už časom preukázané, že prináša zväčša pozitívny efekt.

Nové technológie sa rozširujú veľmi rýchlo, zrejme aj vďaka relatívne mierovým časom sa záujem vedcov a technikov upriamuje týmto smerom. Firmy sa pretekajú a vynakladajú nemalé sumy na výskum v tejto oblasti. Už z roku 1965 je známy výrok, jedného so spoluzakladateľov spoločnosti Intel, Gordona Moora : „*Počet polovodičových prvkov v integrovanom obvode bude možné každý rok zdvojnásobiť.*“ [Moore65], ten síce musel o desať rokov neskôr upraviť na každé dva roky, ale napriek tomu je tento priemysel najrýchlejšie sa rozvíjajúcim na svete. Neostáva však len pri tranzistoroch a procesoroch, počítač už neslúži zďaleka len na matematické výpočty ako to bolo pri obrovských sálových počítačoch. Počítač nás prenasleduje na každom kroku a zdomácnel. Stáva sa z neho multimediálne, informačné, výučbové a pracovné zariadenie. Rozvoj počítačov sa prejavuje skutočne už v každom odvetví od zábavy, vzdelávania, obchodu až po zdravotníctvo a umenie. Vždy ma fascinovali prepočty kapacity pevného disku na počet listov papiera alebo rôzne prirovnania typu: *keby sa naskladali za seba dosiahli by zo zeme na mesiac, dokonca dvakrát*, to len dokazuje že výkonnostne už počítače pokročili a je na čase, aby sa pohli od 1 (text) a 2D (obrázky) do tretieho rozmeru.

Spolu s počítačmi sa vytvoril ďalší fenomén dnešných dní, Internet. Tu sa často hovorí o jeho vzniku v armáde a neskorším použití na amerických univerzitách, ale skutočný internet pre masové použitie poznáme až od roku 1991. Tim Berners-Lee publikoval svoj projekt World Wide Web, dva roky neskôr značkový jazyk HTML a protokol HTTP. Marc Andreessen v tom čase vytvoril ako študent prvý prehliadač a roku 1994 ako spoluzakladateľ Netscape C.C. vydali prehliadač na stiahnutie zadarmo. Zrejme aj toto viedlo k záujmu verejnosti o dovedy akademicko-technický internet.

Úmera medzi počtom používateľov (miera obľúbenosti) a jednoduchosťou interakcie s počítačom resp. s ostatnými počítačmi je tu jasná. Dierne štítky sú dnes smiešne populárne len tým načo kedysi slúžili. Rozhrania počítačov sa rozšírili o textové zadávanie, WIMP (window, icon, menu, pointing device), WYSIWYG editačné systémy. S príchodom Web 2.0 (tzv. Semantic Web) by sa mala interakcia ešte viac zjednodušiť, mohla by k tomu prispieť i modelová reprezentácia objektov a avatari.

Často sa hovorí, že na internete je všetko, dôležité je povedať že to „všetko“ je zdigitalizované. Digitálny text a obrázky sú najmä zásluhou internetu a digitálnej fotografie zvládnuté počítačmi na relatívne vysokej úrovni, často možno na vyššej ako ich analógový ekvivalent. Už niekoľko rokov sme toho svedkami pri hudobných nosičoch a v neposlednej dobe aj video médiách. Pri knihách v elektronickej podobe nie sme zatiaľ svedkami takého rozšírenia, čo je zrejme zapríčinené nedostupnosťou lacného a dostatočne komfortného čítacieho zariadenia veľkosti knihy, ale aj tu sme svedkami migrovania predovšetkým staršej literatúry do digitálnej podoby. Nesporne digitalizácia prináša nové možnosti, ktoré neboli predtým ani mysliteľné a pri pojme virtuálne prostredia sa možností vynára celý rad.

1.1. Virtuálne prostredia

Čo si mám predstaviť pod pojmom virtuálny priestor ? „Nemal by znamenať ani paralelný svet s vlastnými pravidlami ani byť len realistickou reprezentáciou skutočného sveta. Virtuálny priestor by mal byť reprezentácia ľudskej skúsenosti a to ako má zažitý pojem skutočného sveta“ [Qvor02]. Tu sa ukazuje užitočné poznať prostriedky a techniky, ktorými reagujeme a poznávame skutočný svet. Pozorovanie dvoma očami a ušami tzv. pasívne stereo, pohyb v priestore – aktívne stereo paralaxy (zdanlivé posunutie pozorovaného predmetu, spôsobené zmenou miesta pozorovateľa). Spôsob interakcie, pohybu s reálnymi objektmi a možnosť ich zaznamenania. Pri tvorbe virtuálneho priestoru začíname vyslovene od nuly (vo výnimočných prípadoch od jednotky), preto je každá informácia o vnímaní reálneho sveta užitočná a jej spresnenie vedie k reálnejším výsledkom vo svete virtuálnom.

Ďalej sa môžeme pýtať, čo všetko existuje a umožňuje virtuálne prostredie ? No správnou odpoveďou bude zrejme všetko, čo si v ňom vytvoríme. Vo virtuálnom prostredí nie sme nijako obmedzený fyzikou, zákonmi, rozpočtom je len na nás koľko úsilia a času sme ochotní venovať tvorbe niečoho čo v konečnom dôsledku bude len virtuálne. Jediným obmedzením, ktoré môžeme pociťovať je výkon počítača, čo pri realizácii niektorých objektov, scén a prostredí môže byť až neprekonateľné ale ako som už spomínal výkon počítačov ide stále napred a mi by sme mali v rozumnej miere napredovať s ním.

Počítač nám umožňuje vytvárať virtuálne prostredia. Virtuálne prostredie je simuláciou skutočného sveta. Používateľ tam vstupuje s jedinečnou identitou (avatar), ku ktorej je priradená jeho reprezentácia v tomto priestore, ktorá mu umožňuje interakciu s inými používateľmi a objektmi. Takýto svet je väčšinou riadený rovnakými pravidlami a fyzikálnymi zákonmi ako svet reálny. To je z časti zväzujúce, ale náročnosť používania je veľmi jednoduchá. To čo sme sa naučili počas svojich prvých dvoch - troch rokov

života tu môžeme plne využiť, teda používateľ je hneď schopný pohybovať sa, rozpoznávať objekty a rozumieť vlastnostiam a správaniu sa v tomto priestore. Interakciu zjednodušuje aj priamy hlasové prepojenie. Veľa takýchto svetov [1] ide ešte ďalej a neostáva len pri pozorovaní. Ako populárne riešenie by som spomenul Second Life [2]. Názov už veľa naznačil a naozaj postava si môže zájsť ku kaderníkovi, kúpiť si oblečenie podľa poslednej módy, postaviť si vlastný dom a vybaviť ho podľa vlastných želaní, to všetko za virtuálne alebo v prípade podobného projektu PS3 Home od SONY za skutočné doláre, ale to je už len otázka implementácie. Nechýba kultúrne vyžitie v podobe hier, hudby ktorú si zakúpite priamo online či návštevou kina kde premietajú to načo si zmyslíte. Samozrejme môžete ostať doma zavolať priateľov a pozerať vo vašej plazmovej televízii priamy prenos z ligy majstrov. Samozrejme je tu aj priestor pre vlastnú prezentáciu či reklamu veľkých firiem čo vytvorí väčší záujem o tieto svety. Možnosť nastavovať pravidlá sa tu ukazuje ako veľmi výhodná a dá sa zaručiť nulová kriminalita, bezporuchovosť zakúpených spotrebičov a čo ma najviac prekvapuje biela ostáva bielou aj po častom praní.

Počítač nám umožňuje vytvárať virtuálne prostredia. Virtuálne prostredie je vymyslené. Tu sa môže ktokoľvek trošku techniky zdatný autorsky realizovať a vytvárať svety, ktoré by v reálnom prostredí boli príliš komplikované alebo by vôbec neboli realizovateľné. Existuje celá rada CAD (Computer-aided design) systémov, čiže návrhových systémov za pomoci počítača. Rozhodujúca je presnosť, možnosť testovať a simulovať vlastnosti ešte predtým ako sa objekt vyrobí. Ako reprezentácia objektov postačuje sieť základných čiar. Používa sa najmä v automobilovom, stavebnom a strojárskom priemysle. Pri simulácii prúdenia vzduchu nie je dôležitá priesvitnosť, odraz svetla, funkcionalita. Každá sa sústreďuje len na konkrétne vlastnosti tuhosť, odpor vzduchu, spotreba. V súčasnosti nie je systém, ktorý by úplne simuloval vonkajší svet, ale určite nie je nereálne tvrdiť, že sa takýchto systémov dočkáme. Ako doplnok k svetom kde jed dôležitá presnosť, interakcia a fyzikálne vlastnosti sú tu prostredia zamerané na realistický zvähľad. Majú využitie najmä vo filmovom priemysle ako špeciálne efekty alebo dokonca celovečerné animované filmy. Známa je najmä japonský film na základe hry Final Fantasy, ktorá podáva výsledky veľmi realistické na rozdiel od firmy PIXAR, ktorá je zameraná na rozprávkové príbehy, kde postavy ani nemajú tendenciu podobať sa ľuďom. Treba povedať, že zatiaľ čo Pixar a Disney zarobili na svojej prvotine Toy Story 358 miliónov dolárov, Final Fantasy zostala v strate 120 miliónov. Zatiaľ z toho odvodzujem len to, že skutočným hercom v 3D svete ešte nenarástli plnohodnotní konkurenti. Kreslené príbehy to však už nemajú také jednoduché, no veru už je tomu nijaký rôčik keď kraľoval Lion King. Úplne iný typ zážitku prináša IMAX 3D (Image Maximum), ide o premietanie trojdimenzionálneho filmu, resp. jeho reprezentácie tak ako ho vníma človek. Obraz je snímaný dvoma kamerami vzdialených od seba 64mm (priemerná vzdialenosť medzi očami), zaznamenáva sa ľavé a pravé oko zvlášť. Pri premietaní sa tieto obrazy rýchlo striedajú a divák má nasadené okuliare s LCD stmievacími sklami, ktoré sú

synchronizované s projektorom. Inou možnosťou je premietiť polarizované a divákovi sa odfiltruje zložka pre každé oko polarizačnými okuliarmi. Pred divákmi sa potom vytvárajú trojrozmerné virtuálne objekty, ktoré sú v skutočnosti len na plátne.

2. Technológia 3D prostredí

V oblasti 3D prostredí sú ťahúňmi technológie najmä hry. Trh je v tejto oblasti obrovský a žiadny z vývojárov nemôže ostať pozadu. Nové funkcie grafických čipov pribúdajú najmä pre potreby herného priemyslu. Nové nápady v tejto oblasti neprichádzajú každý deň preto je grafika a hudba hry často dôležitejšia a najmä viac diskutovaná ako hrateľnosť či náročnosť. Obvykle až tretie, štvrté pokračovanie hry zaznamenáva úspech starou zabehnutou myšlienkou a vylepšenou grafikou a ostatnými efektmi. Pre tvorbu 3D aplikácii vo všeobecnosti sa používajú dve API (Application programming interface), ktoré výrazne zjednodušujú prácu a prispievajú ku vzájomnej kompatibilite už tak zložitých zariadení a aplikácii.

Ako prvé tu bolo OpenGL (Open Graphics Library) [3] roku 1992. Je to štandard špecifikujúci množinu funkcií pre 3D ale aj 2D grafiky. Používa open-sourcovú licenciu podobnú licenciám BSD alebo Mozilla. Autorita stojaca za ním je OpenGL Architecture Review Board v ktorej majú zastúpenie popredné firmy ako 3Dlabs, Apple Computer, ATI, Dell, IBM, Intel, NVIDIA, SGI a Sun Microsystems. Do roku 2003 bol členom aj Microsoft. Výhodou OpenGL je že nezávisí od programovacieho jazyka ani platformy. Používa jednoduché primitíva na vytvorenie komplexných trojrozmerných scén. Našiel si široké uplatnenie v CAD systémoch, vedeckých, leteckých simuláciách a samozrejme pri tvorbe hier. OpenGL je naozaj univerzálne, dôkazom je široké uplatnenie na platformách, ktoré zrejme už ani nemajú nič iné spoločné Mac OS, Windows, Linux(knižnica funkcií mesa, ktoré interpretujú OpenGL), Unix, BSD, Playstation 3, Wii.

Od roku 1995, s príchodom vtedajšej verzie operačného systému je tu DirectX, ktoré vytvára API nielen pre grafiku ale pre všetky multimediálne zariadenia od zvukovej karty, myši po volant a hru po sieť. DirectX síce nie je program s otvoreným zdrojovým kódom ale je poskytovaný zadarmo tak používateľom ako aj vývojárom (SDK). Microsoft používa toto rozhranie hlavne vo svojich produktoch Windows, Xbox a Xbox 360. Ťažko porovnávať, ktoré z API je lepšie, treba povedať, že DirectX je používanéjšie, najmä vďaka integrácii v operačnom systéme Windows. Už dlhší čas má navrch nad OpenGL a s príchodom Windows Vista a DirectX verzie 10 sa stáva načas víťazom, keďže rozhranie OpenGL 3.0 s podobnými funkciami sa ešte len chystá. Vo verzii 3.0 by sa mala spojiť pôvodná verzia, verzia pre mobilné zariadenia (OpenGL ES), Collada a gLFX (API efektov pre OpenGL), takže DirectX môže očakávať silného súpera.

2.1. Reprezentácia objektov

Pre reprezentáciu objektov vo virtuálnom svete sa používa Euklidova geometria, ktorá zrejme najlepšie a najjednoduchšie reprezentuje svet na aký sme zvyknutí. Základné objekty v 3D priestore sú body s troma súradnicami. Spojením dvoch bodov vzniká čiara

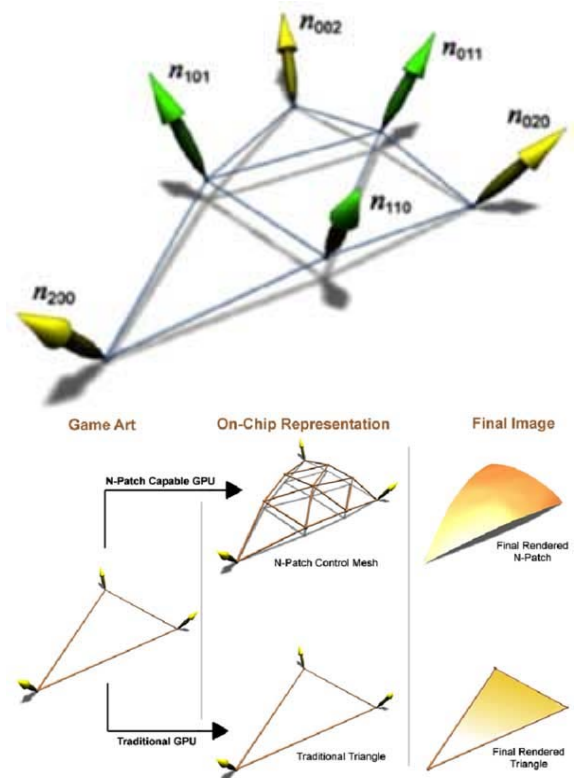
(tieto objekty sú stále neviditeľné keďže nemajú rozmer). Spojením troch bodov vzniká konečne plocha, trojuholník. Spájaním viacerých trojuholníkov vzniká väčšia plocha nie nutne v jednej rovine. Pre plochu môžeme nastaviť konkrétnu farbu alebo textúru. Textúra je bitová mapa, ktorá sa zobrazí na povrchu plochy. Keď si predstavíme hraciu kocku, textúry tejto kocky budú fotky všetkých jej strán. Použitím dostatočného počtu polygónov, pre jednoduchosť stačí uvažovať trojuholníky, sme schopní vytvoriť aproximáciu ľubovoľného reálneho modelu. Koľko je to dostatočný počet? Keby sme išli až na atomárnu úroveň, náš počítač by sa veľmi nepotešil, preto aj tu platí že všetko s rozumnou mierou a za dostatočný počet môžeme považovať model, ktorý na pohľad nerozlíšime od reálu.



Obr. 1: Guľa tvorená 15 segmentmi, v strede 60 segmentov, napravo to isté otextúrované.

2.2. N-Patch

Motiváciou pre použitie technológie N-Patch bolo, že moderné grafické karty už bežne zvládajú renderovanie pri 60+ FPS (obrázkoch za sekundu) vo vysokom rozlíšení, výrobcovia sa chceli zamerať na zvýšenie kvality pri zachovaní pôvodnej štruktúry a kvality vstupných modelov. Aby boli objekty viac podobné reálnym snažíme sa zvýšiť množstvo polygónov. Veľké množstvo polygónov znamená pamätať si všetky jeho vrcholy a textúru viažucu sa k nemu. Technológia N-Patch slúži práve na úsporu počtu polygónov pri dosiahnutí porovnateľnej kvality ako s použitím ich niekoľko-násobného počtu. Dizajnér navrhne jednoduchý model, ktorý v pamäti aj pri presúvaní po zberniciach zaberá malo miesta respektíve času spracovania a až v grafickom procesore sa model zjemní. Využívajú sa normálové vektory (na obrázku žltou farbou) z vrcholov polygónu, tieto slúžia na výpočet odrazu



Obr. 2: Zjemňovanie pomocou N-Patch

svetla od objektu takže nevzniká výpočet navyše. Ako interpolácia pôvodných vektorov a bodov vznikajú nové, samozrejme pôvodný polygón zanikne a vznikajú namiesto neho nové, tie v tomto prípade vystupujú navrch, čo vieme určiť z uhlu aký v danej rovine zvierajú pôvodné vektory, to tiež určuje konkávnosť alebo konvexnosť oblúku medzi nimi. Uhol pod, ktorým stúpa hrana nového polygónu smerom od starého k novému vektoru je určená tým, že medzi oboma vektormi a hranou bude rovnaký uhol. Takto interpolovať môžeme samozrejme aj novovzniknuté vektory a tým dosiahnuť ešte lepšie výsledky vzhľadom na hĺbku takejto rekurzie, väčšinou sa však volí hardwarovo menej náročná nerekurzívna cesta a interpoluje sa rovno viac bodov naraz. Tieto výpočty sú také jednoduché na výpočet, že sa dokonca ani neukladajú do grafickej pamäte, čo je ďalšia úspora, a počítajú sa v reálnom čase. Táto technológia sa už je zapracovaná pod názvom TruForm v OpenGL a aj DirectX 8 rozšíreniach pre grafické čipy od firmy ATI. TruForm vytvára z jedného trojuholníku deväť, najskôr na každej hrane vytvorí dva nove vrcholy a potom z týchto šesť vytvorí siedmi centrálny. S ohľadom na to aké vylepšenie prinášajú, hrany sú hladšie, objekty sú detailnejšie a realistickejšie, je zníženie výkonnosti skutočne zanedbateľné.

2.3. Mapovanie textúr

Ďalšou súčasťou realistického modelu je textúra, a práve od jej správneho zvolenia závisí výsledok. Objekt bez textúr je na vstupe reprezentovaný len polygónmi, kde každý vrchol má určenú farbu, plocha je potom vyfarbená interpoláciami farieb vrcholov medzi nimi. Priradením textúry na polygón určujeme farbu každého bodu plochy polygónu. Zatiaľ čo na trávu, cestu si vystačíme s textúrou, ktorá bude zoskupením náhodných farieb blízkyh zelenej respektíve sivej na realistické modely potrebujeme textúry vytvorené z fotografií. V minulosti sa používali textúry na simuláciu realistického osvetlenia a tieňovania, odlesk alebo tieň boli priamo nakreslené na textúre. Toto sa môže sa samozrejme využívať aj v súčasnosti, pri statickom osvetlení si toto môžeme vypočítať dopredu a uložiť priamo ako textúru, pri dynamickom osvetlení však tento trik bude veľmi rýchlo odhalený. Lepšia je technológia Light map, ktorá umožňuje ukladanie informácie o osvetlení do akoby paralelnej vrstvy textúry, čo nám umožňuje ďalšie prepočty svetla nezávisle na farbe textúry. Ukladanie viacerých vrstiev na polygón sa nazýva multitexturing.

Asi najzaujímavejšou v súčasnosti používanou z týchto vrstiev je Bump mapping. Nie vždy je reprezentácia polygónmi vyhovujúca, napríklad pri stene alebo špongii, ktorá má nespočetne pórov a praskliniek je bump mapping akoby simulácia povrchu objektu textúrou. Bumpmapová mapa je čiernobiela, kde tmavšia farba znamená väčšiu hĺbku. Samozrejme toto sa zohľadňuje v umiestnení textúry a taktiež osvetlenia, takže objekt vyzerá akoby bol vytvorený niekoľko rádovo viac polygónmi ako v skutočnosti je. Z Kocky

by sme guľu síce takto vytvoriť mohli, ale vo väčšine prípadov nie je hĺbka bump mappingu nijako výrazná a používa sa skôr na kozmetické úpravy povrchu.

Ďalšou zaujímavou vrstvou je vrstva alfa. V textúre môžu byť niektoré miesta textúry (okno, voda) priesvitné. Alfa nám určuje aj stupeň priesvitnosti podobným systémom ako sa pri bump mappingu určuje hĺbka.



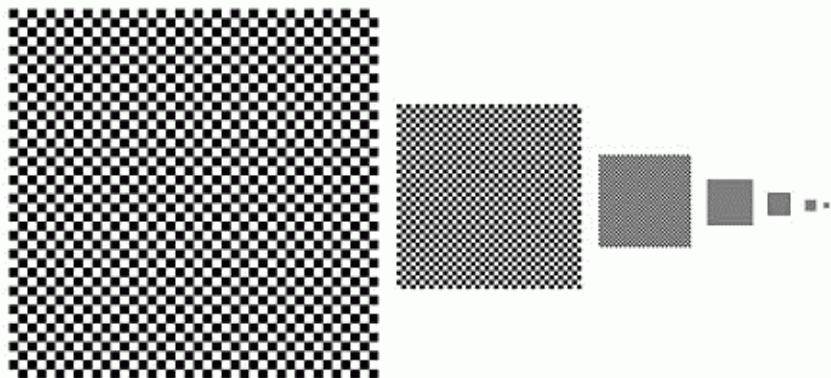
Obr. 3: Vpravo použitie Bump mappingu

Zaujímavosťou pri navrhovaní pohyblivých scén alebo animácii je použitie dynamickej textúry, tá nám umožňuje bez zmeny objektu zmeniť jeho farbu, svetelnosť či iné informácie priradené k polygónu. Pri takomto množstve informácii v textúre bola snaha na ich optimalizáciu. Firma S3 prišla so síce stratovou kompresiou textúry, ktorá sa však od bežných kompresii používaných na obrázkoch odlišuje tým, že potrebuje len jeden prístup do pamäte napriek tomu môže dosahovať až kompresný pomer 8:1. V blokoch 4x4 sa uchováajú 32-bitové RGBA dáta (červená, zelená, modrá, alfa). Formát tejto DXT textúry s S3TC algoritmom (S3 texture compression) sa v jemných obmenách používa stále a napriek tomu, že S3 už nie je lídrom vo vývoji grafických kariet, licenčné poplatky za DXT stále prinášajú zisky.

2.4. Filtrovanie textúr

Hovoríme mu tiež vyhladzovanie textúr, ide vlastne o napasovanie textúr na objekty tak ako ich reprezentuje grafická karta. Komplexná textúra sa napasuje na konkrétny počet pixlov objektu v scéne. Textúram sa musí upraviť veľkosť, rotácia a prispôbiť aktuálnej perspektíve. Čo v diaľke vyzerá dobre z blízka mapované na viac pixlov už nemusí byť tak presvedčivé, naopak mapovanie komplexnej textúry na objekt v diaľke nemusí byť dobre čitateľné. Filtrovanie určuje farbu z textúry použitéj na objekt, keďže počet pixlov a texlov (pixel textúry) nemusí byť rovnaký.

Čiastočným riešením je mipmapping. Tento dokáže, k textúre vytvoríť jej menšiu verziu, tieto sa používajú potom v závislosti na vzdialenosti od kamery. K textúre o veľkosti 256x256 teda máme vytvorené ďalšie verzie 128x128, 64x64, 32x32, to sa však nedá používať vždy. Preto sa najbližšia verzia posúva na ďalšie spracovanie inými filtrami.



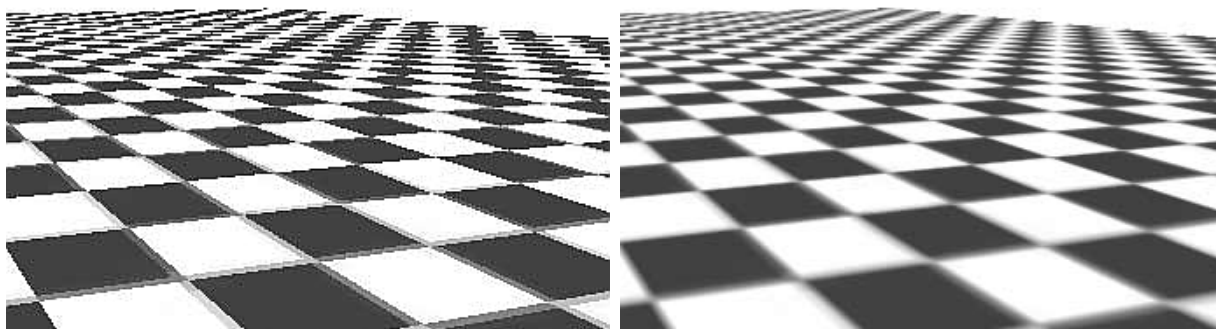
Obr. 4: mipmapping

Najjednoduchším je použiť interpoláciu na najbližšieho suseda. Jednoducho použije ten texel, ktorý je najbližšie. Obrovskou nevýhodou je, že nezobrazuje celé textúry, ale len nejaké ich časti, ktoré sa radikálne menia spolu so zmenou perspektívy. Takto vznikajú tzv. artefakty, chyby zobrazenia, rovná čiara môže byť prerušovaná, nebude vôbec viditeľná alebo viditeľná len z určitého uhla, čo nie je samozrejme žiadané.

Lepším sa ukazuje Bilineárne filtrovanie, ktoré už uvažuje štyri najbližšie texely a z tých potom robí vážený priemer podľa vzdialenosti. Aj tu sa však môžu vyskytovať chyby čo sa Trilineárne filtrovanie snaží riešiť použitím dvoch bilineárnych a až ich lineárnou interpoláciou stanoviť výslednú farbu.

Bez filtrovania

Anizotropické

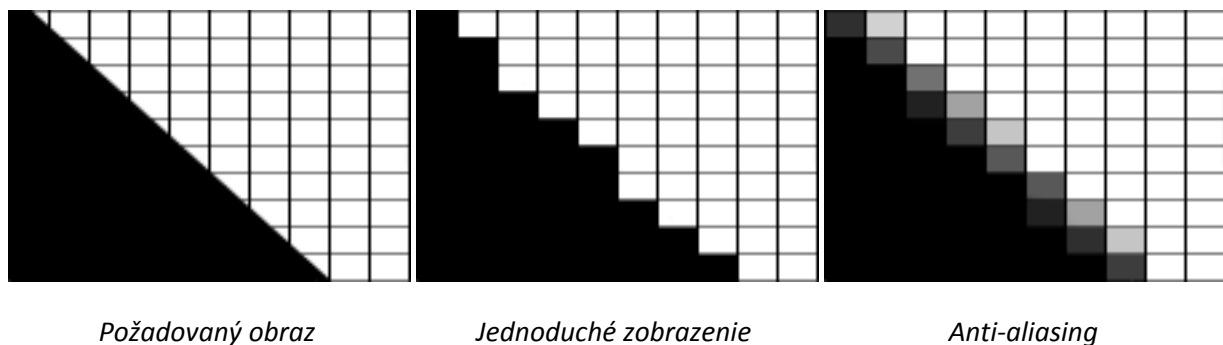


Obr. 5: Filtrovanie textúr

Obe tieto filtrovania však zabúdajú na perspektívu a tým, že pracujú na štvorcovej oblasti ich výsledok bude výborný čelom k pozorovateľovi, ale skreslený a rozmazaný z nepriameho uhla. Tento problém rieši Anisotropické filtrovanie, ktoré používa trapezoid zodpovedajúci pozorovaciemu uhlu [4].

2.5. Antialiasing

Potom čo máme textúry mapované na objektoch, objekty sú umiestnené v 3D priestore, chceli by sme takúto scénu konečne zobraziť. Robíme projekciu do 2D plochy konečných rozmerov, jej veľkosť označujeme v pixloch, zvyčajne 1 až 2 Miliónov Pixlov a 100 bodov na palec (dpi). Musíme riešiť situáciu, podrobná textúra v scéne sa má v projekcii zobraziť na plochu oveľa menšiu a perspektívne upravenú. Zjednodušene by sa dalo povedať, že hľadáme farbu pre bod plochy, na ktorý sa zobrazuje viac bodov zo scény. Podobne ako pri aplikovaní textúry aj tu kvôli neinjektívnosti takéhoto zobrazenia vznikajú rôzne artefakty a skreslenia obrazu. Základnou myšlienkou ako odstrániť tieto chyby je supersampling. Tento vyberá niekoľko bodov z pôvodného obrazu, ktoré sa majú zobraziť na rovnaký bod a z nich potom vypočíta farebný priemer. Počet bodov závisí od zvolenej kvality.



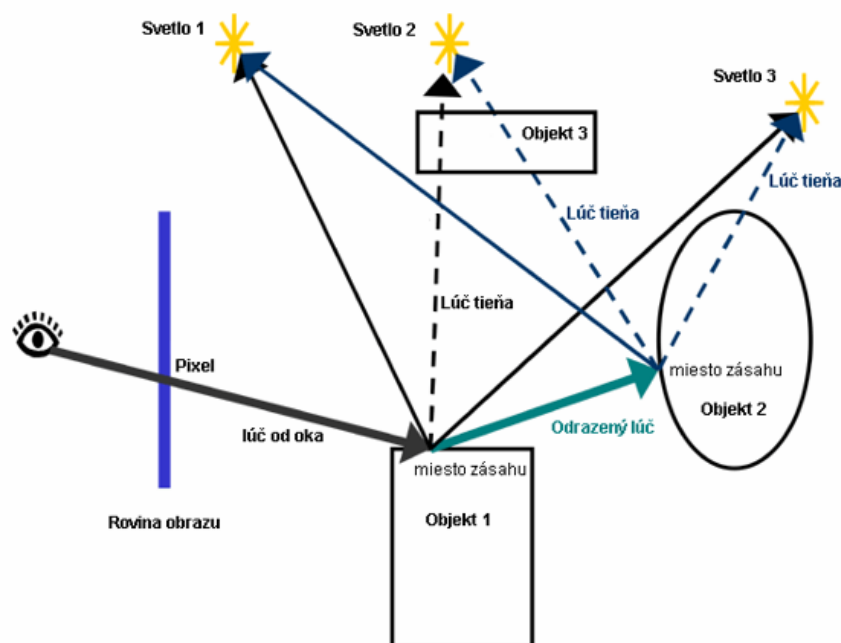
Obr. 6: Požitie anti-aliasingu

Dôležitým faktorom je tak isto rozmiestnenie bodov v pôvodnom obraze, s rovnomerným rozdelením totiž nedosiahneme výsledok veľmi odlišný od neupraveného zobrazenia. Na rad prichádza náhodné rozmiestnenie a jeho vylepšená verzia tzv. poissonovo rozdelenie, ktoré zabezpečuje aby žiadne dva body neboli príliš blízko pri sebe. Najdokonalejšie rozmiestnenie je rozklepané (jittered), ktoré rozmiestňuje poissonovo v mriežke. Tento výpočet je veľmi náročný pretože musí zbehnúť pre každý bod obrazu, preto sa najmä v dynamických scénach používa adaptívny supersampling, ktorý využíva že na hranách je v skutočnosti len málo bodov a stačí počítať len tieto. V praxi sa používa technika nazývaná anti-aliasing, ktorá nemá taký komplikovaný výpočet, keďže ako základ nepotrebuje scénu vo vysokom rozlíšení, ktorú potom prepočíta na menšie ako to robí supersampling, ale pracuje s rovnakým

rozlíšením ako bude mať výstup a interpoluje len body, ktoré potrebuje. Zaťaženie systému bude rovnaké ako keby anti-aliasing zapnutý nebol, svoje si odpracuje len grafický akcelerátor, napriek tomu výsledok bude porovnateľný s použitím vysokého rozlíšenia. Anti-aliasing má tak isto rôzne stupne kvality, 2xFSAA znamená že sa použije celo obrazový aliasing, ktorý použije dva body na interpoláciu výsledného jedného, dostávame teda výsledok podobný dvojnásobnému rozlíšeniu. Súčasnú grafickú kartu zvládajú úroveň 16 v reálnom čase, môžeme však použiť aj vyššie hodnoty rôznymi softwarovými implementáciami používanými najmä na bitmapovú grafiku. Úspešná je taktiež technológia Microsoft ClearType, ktorá používa anti-aliasing pre lepšiu viditeľnosť písma najmä na ostrých hranách aké má LCD monitor. Tu sa berie do úvahy aj rozmiestnenie subpixelov červenej, zelenej a modrej farby, čo robí projekciu trochu viac komplikovanejšou.

2.6. Osvetlenie

Posledným kúskom v skladačke je osvetlenie. Princíp je jednoduchý, máme vytvorené zdroje priameho alebo rozptýleného svetla a podľa toho vzniká podľa umiestnenia objektov v scéne svetlo alebo tieň, zosvetlením alebo stmavením textúr. Toto je v skutočnosti v porovnaní s realitou veľmi zjednodušený model, napriek tomu aj tento bol dočasu náročný na výpočet. A bude ešte náročnejší ak budeme chcieť simulovať skutočné svetlo, konkrétne jeho cestu od svetelného zdroja, odraz a prechod rôznymi materiálmi ako je sklo, zrkadlo, voda ako aj zníženie intenzity po odraze od difúzných materiálov ako drevo, kameň, tehla až ku pozorovateľovi.



Obr. 7: Ray tracing [5]

Prvým algoritmom čo riešil tento problémom bol Ray casting, ktorý vymyslel Arthur Appel už roku 1968. Vychádza z myšlienky, že rátať osvetlenie sa oplatí len pre objekty, ktoré sú viditeľné. Vyrátaval lúč od pozorovateľa cez konkrétny bod na zobrazovacej ploche až ku prvému objektu, na ktorý narazil, potom spätne priradil tomu ktorému bodu na obrazovke farbu podľa vzdialenosti a farby objektu, ktorý sa preťal. Ako jeho nasledovník vznikla technológia Ray Tracing, kde sa pridali nové typy lúčov: odrazený, rozptýlený a tieňový. Lúče teda pokračujú odrážajú sa, menia farbu, intenzitu a vytvárajú v konečnom dôsledku oveľa realistickejší obraz. Na výstupe je taktiež výborné tieňovanie, ktoré by bolo použitím iného algoritmu náročné na výpočet. V súčasnosti sa ray tracing používa len na modelovanie statických scén pretože výpočet dráhy desiatok tisíc lúčov je nesmierne náročný. V spoločnosti Intel však už vyvinuli zatiaľ len softwarový systém na realizáciu ray tracingu v reálnom čase s použitím niekoľkých procesorov. Na renderovanie zaujímavejších scén ray tracingom v reálnom čase sa odporúča výkon zhruba 450 miliónov raysegs/s (segmentov svetla za sekundu), zatiaľ čo súčasne pentium dosahuje asi 100 miliónov, čo naznačuje, že budúcnosť plného využitia nie je veľmi vzdialená. Ray tracing je často používaný aj vďaka voľne dostupným programom na jeho realizáciu ako je Pov-Ray, YafRay alebo Kerkythea.

Ďalším zaujímavým algoritmom na výpočet osvetlenia je Radiosity [6]. Používa sa na vyhodnotenie intenzity svetla dopadajúceho na výlučne difúzne povrchy. Svetelný lúč sa sleduje od svetelného zdroja k objektom, kde sa časť pohltí a časť odrazí na ďalšie objekty. Výsledné osvetlenie objektu je potom uložené ako súčet všetkých dopadnutých lúčov. Kvalita takéhoto renderovania závidí samozrejme od počtu vypočítaných odrazov, teda do akej hĺbky necháme rekúziu ísť. Pre uspokojivý výsledok by to malo byť aspoň tri, ale často to býva aj dvadsať. Pri odraze sa lúče vyžarujú vo všetkých smeroch podľa gausovho rozdelenia, kde vrchol je práve uhol ako pri obyčajnom zrkadlovom odraze. Teda náročnosť výpočtu každého ďalšieho odrazu rastie kvadraticky, keďže počet lúčov stále rastie.

V ostatnom čase sa do popredia dostala ďalšia technológia vylepšujúca realističnosť osvetlenia HDR Rendering (High dynamic range) najmä vďaka jej podpore v nových grafických kartách. Už pred desiatimi rokmi s ňou prišiel Paul Debevec [7] na konferencii SIGGRAPH(Special Interest Group for computer GRAPHics) [6], jeho spracovaním syntetických objektov na reálne scény sa inšpirovali aj výrobcovia. HDR sa vychádza z vnímania svetla ľudským okom a schopnosti rozširovať a zužovať zreničky podľa potreby. Má schopnosť scénu úplne oživiť, vyfarbiť, oblohu presvetliť a pridáva realnosť tretiemu rozmeru. Dosahuje sa to tým, že svetlo nemá intenzitu v rozsahu od 0 po 1 ale túto hranicu prekračuje a napríklad slnko môže dosahovať 60000, to zaručí, že svetlo v scéne akoby nebude chýbať. Pri fotografiách sa vyšší svetelný rozsah vytvára inak. Objekt sa odfotoграфuje v niekoľkých verziách, čo sa mení bude expozícia. Takto budeme mať napríklad jednu fotografiu tmavú, na ktorej však vidno detaily objektov, ktoré sú dobre osvetlené. Jednu fotografiu normálnu a jednu presvetlenú, kde zas budú dobre

viditeľné objekty v tme a tieni. Spojením týchto fotografií programom vzniká obrázok s oveľa väčšou farebnou škálou ako pri normálnom fotografovaní.

Zaujímavým efektom so svetlom je nesporne Motion blur, rozmazanie obrazu naznačujúce rýchly pohyb. Je to zasa priblíženie zobrazenia vnímaniu ľudského oka, kde pri vysokých rýchlostiach sa zužuje oblasť ostrého videnia. Využitie nachádza najmä v rôznych simuláciách závodných áut a lietadiel.

3. Virtuálne múzeum

3.1. Motivácia a ciele projektu

V posledných desaťročiach sa počítače stali súčasťou našej spoločnosti. Počítače našli uplatnenie v mnohých smeroch, aj v takých ku ktorým sa možno determinizmus a matematika, ktoré tvoria základ počítačov, nehodia. Mám namysli najmä umenie, či už hudobný alebo filmový priemysel. Na internete vznikol nový literárny útvar, blog. Ľudia si užívajú demokraciu a môžu sa deliť a konzultovať svoje názory na konkrétne každodenné problémy. Veľa ľudí sa nedelí len o svoje názory, ale aj o svoje vedomosti. Internet je plný rôznych informácií a je tu snaha o ich skompletizovanie a ľahšie vyhľadávanie. Vznikajú internetové stránky wiki typu, ktoré vychádzajú zo spoločného autorstva viacerých ľudí. Ich úspešnosť svedčí o ochote ľudí rozdeliť sa a prispieť, stačí vytvoriť vhodný nástroj alebo spôsob. Týmto novým spôsobom prezentácie a sprostredkovania informácií môže, byť aj virtuálne múzeum [8][9]. Cieľom môjho projektu je popísať technológie, ktoré stoja za virtuálnymi múzeami a ich tvorbou, odhaliť problematiku časti a v neposlednom rade vytvoriť prototyp virtuálneho múzea na, ktorom by sa dali demonštrovať použité technológie. Jedným z dôležitých podcieľov je vytvoriť múzeum a celý projekt tak, aby sa dal prezentovať na internete a splnil tak svoje informatívne poslanie. Toto môže byť zaujímavé nielen pre konzumentov, návštevníkov virtuálnych múzeí, aby sa oboznámili s novým typom internetového obsahu, ale najmä pre vývojárov a tých, ktorí sa chcú prezentovať aj inou formou.

3.2. Funkcionalita a riešenie

Spôsoby prezentácie virtuálneho múzea môžeme rozdeliť do troch základných kategórií. Pomocou statických obrázkov, vnika niečo podobné ako fotogaléria sprevádzaná textovým popisom. Výborne sa môžu uplatniť pri galériách obrazov, dobrým príkladom je galéria Vincenta Van Gogha [10].

Druhým spôsobom prezentácie sú panorámy. Ide o poskladanie viacerých obrazov snímaných z jedného miesta na guľovú alebo valcovú projekciu. Pozorovateľ má potom možnosť meniť uhol pohľadu a priblíženie, ale nie miesto z ktorého pozoruje (v rámci jednej panorámy). Napriek tomu panoráma poskytuje dobrú predstavu o prezentovanom priestore s vynaložením relatívne malého úsilia.

Najzaujímavejším spôsobom je použitie skutočných 3D modelov. Ten nám poskytuje plnú voľnosť pohybu a interakciu z objektmi. Reprezentácie objektov sú matematicky totožné alebo aspoň pripodobené tým reálnym, návštevník tak nie je o nič ukrátený. Ba naopak, objekt si môže prezrieť zo všetkých uhlov, aj takých v ktorých by za normálnych

okolností použil žeriav alebo vrtuľník. 3D objekty nemusí pozerať len prostredníctvom monitora, s použitím nových zariadení ako sú 3D okuliare alebo dotykové rukavice získa reálny zážitok. Ja som sa rozhodol práve pre tento spôsob, lebo poskytuje vizuálne najpresvedčivejší výsledok a možnosti interakcie.

Pri projekte vhodnom na internet s tým treba počítať už od začiatku a tomu prispôbiť zopár faktorov.

- Univerzálnosť. Internet je univerzálne médium prístupné rôznym platformám. Dokonca na rovnakej platforme je dostupných niekoľko internetových prehliadačov čo sa v prvom momente môže javiť ako výhoda pre klienta, ktorý si tak môže vyberať to čo mu najviac vyhovuje. Na opačnej strane spôsobuje nemalé problémy vývojárovi, ktorý tak musí počítať s odlišnými interpretáciami jeho stránok rôznymi prehliadačmi. V našich končinách to nie je až také zlé, stačí uvažovať Microsoft Windows a platformy založené na jadre Linux. V oboch sa používajú najmä prehliadače Mozilla Firefox a Opera. Ďalej treba uvažovať natívny prehliadač vo Windows Microsoft Internet Explorer. Tieto tri prehliadače majú prakticky rozdelený trh. Natívny linuxový prehliadač Konqueror podobne ako Safari v MacOS majú len minoritné postavenie. Problémy si môže ušetriť aj autor sám, keď si zvolí namiesto HTML4 radšej XHTML, ktoré je trochu poučené z problémov svojho predchodcu a nevzniká toľko nedorozumení pri interpretácii. S rozvojom mobilných zariadení sa internet dostáva aj tam. Pri voľbe funkcionality, ktorú HTML priamo nepodporuje namiesto rôznych doplnkov a rozšírení pre prehliadače je lepšie použiť javové applety, keďže väčšina mobilných zariadení má pre javu priamu podporu.
- Optimalizovať veľkosť. Http protokol je vo svojej podstate služba na požiadanie. Klient si vyžiada nejakú informáciu a v konečnom čase na ňu čaká aj odpoveď. Všetko sa deje prostredníctvom siete, ktorá má taktiež konečnú kapacitu prenesených informácií za určitý čas. Keď teda chceme aby komunikácia, interakcia bola plynulá musíme optimalizovať veľkosť dát, ktoré sa budú prenášať. Pomáhajú nám v tom najmä úsporné formáty pre obrázky, video, hudbu alebo iného obsahu. Súčasný 3D hry dosahujú niekoľko gigabajtov, čo je pre internetové použitie absolútne nemysliteľné. Treba teda uvažovať v rozumných mierach a neodradiť ľudí zbytočným čakaním.
- Voľba správnej technológie, vhodnej pre internet. Hlavne kvôli používaniu rôznych platforiem nie každú technológiu dokážeme implementovať univerzálne. A ani rámci rovnakej platformy užívatelia nie sú ochotní si sťahovať a inštalovať ďalšie aplikácie. Obsah by teda mal vychádzať zväčša z niečoho čo už väčšina ľudí nainštalované má. Za takéto môžeme považovať javu, flash a nejaký prehliadač vrml, ktorý síce veľa ľudí nemá, ale je štandardom pre zobrazovanie 3D obsahu.
- Prehľadnosť, jednoduchosť ovládania. Napriek tomu, že by sme chceli do projektu začleniť veľkú funkcionality, musíme vytvoriť menu tak aby návštevník

intuitívne našiel to čo potrebuje. Menu preto nemôže byť priveľmi rozsiahle a radšej bude obsahovať podpoložky, nie ale z veľkou hĺbkou tak aby bolo možné pristupovať ku všetkým položkám na niekoľko kliknutí.

V mojom projekte budem tieto problémy riešiť najmä správnym výberom formátu vstupných a výstupných dát a použitej technológie. Nebudem používať nič až tak prevratne nové a teda viem vopred čo sa dá od použitia toho alebo onoho očakávať. Jazykom na popis štruktúry a vzhľadu stránok použitých pre internet, bude pre mňa XHTML verzie 1.0 v Transitional formáte s použitím CSS verzie 2.1. Pre popis 3D objektov som sa rozhodol použiť VRML97, ktoré je zaužívaným štandardom pre internetové prezentácie virtuálnych svetov.

3.3. Popis použitých formátov

XHTML (Extensible Hypertext Markup Language) [11] je nasledovníkom známeho HTML a je odporúčaný ako formát na prezentáciu na internete už spomínaným W3C, ktoré stálo za vznikom moderného internetu v deväťdesiatych rokoch a v súčasnosti určuje štandardy. Zatiaľ čo HTML vychádza zo starého SGML (standard generalized markup language) vyvinutého ešte v šesťdesiatych rokoch, XHTML vychádza z relatívne nového štandardu na štruktúrovanie dát XML [12]. To prináša významový rozmer zapísaným dátam. Význam je určený ďalším súborom, ktorý voláme DTD (document type definition), teda definícia typu dokumentu. To umožňuje správnu interpretáciu XML súboru aplikáciou, v našom prípade XHTML súboru prehliadačom. Ďalšou výhodou je, že takzvané „parsovanie“, teda rozbor dokumentu je jednoduchší a rýchlejší vďaka univerzálnosti XML. A v neposlednom rade vzniká menej chýb, keďže v DTD je určená presná štruktúra dokumentu a aj samotný autor si vie odkontrolovať chyby.

Významným rozdielom oproti starému zápisu HTML je tiež, že mnohé prvky boli zjednodušené a ich atribúty sa dajú nastavovať len pomocou CSS (cascading style sheet) [13] kaskádových štýlov. CSS tu bolo už dávnejšie pred príchodom XHTML, ale až spolu s ním sa docielilo, že dokumenty vyzerajú rovnako aj na rôznych platformách a prehliadačoch. Formáty ako napríklad XHTML Strict umožňujú len definíciu štruktúry a samotný vzhľad musí byť definovaný v oddelenom súbore s použitím CSS. Pri editácii meníme štruktúru a vzhľad oddelene. Táto vlastnosť umožňuje rýchle a efektívnu zmenu vzhľadu, keďže tá sa viaže na definované štruktúry a nie konkrétne elementy. CSS napríklad definuje dvojité orámovanie pre všetky obrázky, nemusíme to definovať pri každom výskyte obrázku. Kaskádovosť tohto štýlu spočíva v tom, že vlastnosti elementov sa prenášajú aj na elementy rovnakého typu definovaných nižšie v štruktúre, napríklad obrázky vložené v tabuľke. Aj tie sa však dajú predefinovať a potom sa prenášajú nižšie už tieto. Novšie verzie priniesli do CSS aj triedy, elementom rovnakého typu sa môže nadefinovať trieda a vzhľad je potom definovaný pre túto triedu, napríklad rozdelením

na dve triedy dosiahnem orámovanie všetkých obrázkov dievčat červenou a chlapcov modrou. Oddelením štruktúry od vzhľadu získame jednoduchý a prehľadný kód a zmenší sa tým aj objem dát, najmä znovu použitím rovnakých definícií a vzhľadu v rôznych dokumentoch. Ako som už povedal budem používať modifikáciu Transitional, ktorá nie je až taká prísna na oddelenie vzhľadu od dokumentu, keďže nie vždy to je potrebné. Transitional taktiež umožňuje vkladanie appletov priamo do kódu, čo Strict neumožňuje. Kódovanie pre znakovú sadu zvolím UTF-8 (Unicode transformation format). Je to bezstratové kódovanie s premenlivou dĺžkou pre všetky unicode znaky. Zabezpečím tak, že aj na inej platforme môžem prezerať a editovať bez obavy, že sa mi pokazia dĺžne a mäčkene. Taktiež prepojenia z inými systémami ako napríklad databázou sú jednoduchšie.

VRML (Virtual Reality Modeling Language) [14], vznikol tiež na podnet podobného konzorcia ako je W3C a to Web3D, ako vývoj viacerých firiem a odborníkov, takže bol dobre prijatý ako univerzálny štandard. Štandardom sa stal roku 1997 ako VRML97 (ISO/IEC 14772-1), okolo jeho označenia sú síce nejaké nejasnosti. V súčasnosti používaný VRML 2, prípadne 1.2 sú len programátorské označenia štandardu VRML97 [Kuzev01], a teda formát sa používa už 10 rokov. Ako už názov naznačuje je to jazyk na modelovanie virtuálnych svetov. Formát určuje aj spôsob zápisu do textového súboru v SGML tvare. Formát popisuje scénu pomocou zoskupovania mnohých grafických primitív ako sú guľa, valec, čajník. Keď však poviem, že podporuje definíciu bodov a ich spájanie do polygónov, je nám jasné že VRML dokáže popísať ľubovoľný 3D objekt. Všetkým objektom viem priradiť farbu alebo obrázkovú textúru, lesk, priesvitnosť a ďalšie parametre. VRML je tak plnohodnotným jazykom na opis virtuálnych prostredí. Napriek tomu, že nejde o príliš zložitý zápis, pri viacerých objektoch sa stáva ťažkopádny a na jeho tvorbu a editáciu sa používajú modelovacie programy s importom a exportom do VRML.

Vlastnosti jazyka VRML:

- VRML obsah je možné umiestniť priamo na stránku
- Umožňuje vkladanie multimédií ako sú zvuky a video
- Virtuálny svet má zabudované prvky na interakciu ako sú udalosti po kliknutí, prejdení ponad určitým objektom alebo časovače, ktoré spúšťajú vnútorné alebo aj externé udalosti
- Špeciálny skriptovací element umožňuje pridávať ďalší kód (najmä Java a JavaScript)
- Pre popis statického aj dynamického prostredia sa používajú rovnaké prostriedky, nie je preto problém zo statickej scény spraviť dynamickú a naopak.
- Pohyb vo virtuálnom prostredí je súčasťou definície jazyka, prechádzanie medzi rôznymi svetmi je tiež možné

- V súbore je možné používať objekty definované v ďalších súboroch uložených lokálne alebo niekde na internete
- Formát súboru je textový, ale dá sa komprimovať pomocou programu gzip (musí to byť gzip ako ho poznáme z unixových systémov, existuje aj verzia pre Windows, obyčajný zip bohužiaľ nefunguje), prípona sa zmení z wrl na wrz, prehliadač takýto súbor dokáže priamo otvoriť [15]

Kedže VRML nie je priamou súčasťou XHTML, na jeho prezeranie potrebujeme špecializovaný prehliadač, ten sa často inštaluje ako rozšírenie pre obyčajný internetový prehliadač. Na výber je celá rada voľne dostupných riešení, uvediem aspoň dva s ktorými mám dobré skúsenosti: Octaga [16], Cortona [17]. Treba spomenúť, že VRML tu má aj od roku 2005 nasledovníka v podobe X3D, ktoré využíva XML. Prichádza samozrejme aj s novou funkcionalitou ako napríklad ľudskými animáciami a podporou pre moderné tieňovanie (shaders) grafických kariet. Pre naše účely postačí dobre známe VRML97, aj vďaka tomu, že v prípade potreby je konverzia do X3D reaktívne jednoduchá (napríklad programom VrmlMerge [18]).

JPEG [19] je kompresia na ukladanie obrázkov do súborov s rovnomenným formátom, vyvinutá rovnomenným združením Joint Photographic Experts Group. Štandardom je od roku 1994. Používa sa najmä na uloženie fotografií, kde sa väčšinou používa stratová kompresia, s ktorou sa dosahujú veľmi dobré výsledky pomeru kvality ku veľkosti súboru. Práve preto práve preto našiel široké uplatnenie najmä na prenos a publikáciu fotografií na internete. Tu ostatné formáty ako PNG (Portable Network Graphics) alebo GIF (Graphics Interchange Format) ostávajú pozadu, keďže PNG nemá ani rádovo takú dobrú kompresiu na bežné obrázky a GIF má nedostatočné farebné podanie, keďže podporuje maximálne 256 farieb. JPEG nie je vhodný na zobrazenie presných čiar, plánov, textu alebo ikon, kde jeho stratová kompresia vytvára mnohé viditeľné artefakty. Na tieto sú naopak vhodnejšie spomínané bezstratové formáty, ktoré navyše podporujú tzv. priesvitnú farbu. Existujú však aj varianty JPEG, ktoré používajú bezstratovú kompresiu.

Kompresia JPEG nepozostáva len z jednej operácie, pozostáva naopak z niekoľkých na seba nadväzujúcich optimalizácií. Vernosť farieb sa zachováva zmenou rozloženia farieb. Počítač implementuje farby ako Zložky RGB (červená, zelená, modrá), pri JPEG je to 24 bitov na každý bod, 8 bitov na každú zložku, to pokýva viac ako 16,7 milióna rôznych farieb. Ľudské oko však nie je citlivé na každú farbu rovnako, preto sa prevedú farby do reprezentácie YCbCr, kde Y reprezentuje svetlosť (luma), Cb a Cr je farebný rozdiel medzi modrou resp. červenou a lumou. Takto upravená farebná reprezentácia si pri rovnakej kompresii zachová vyššiu kvalitu. Viac detailov vidí človek práve na Y komponente ako na Cb a Cr, preto sa ďalej robí farebné zníženie zastúpenie týchto zložiek. Z pôvodného rovnomerného pomeru 4:4:4 sa znižuje na 4:2:2 alebo dokonca 4:2:0 pri prechode nielen horizontálne ale aj vertikálne. Týmto procesom priamo ušetríme 33% alebo 50% miesta

vo výslednom súbore. Ďalej sa obraz rozdelí na bloky po 8x8 pixlov, keď má obraz veľkosť nedeliteľnú ôsmimi doplní sa blok priemernými hodnotami farieb bloku, preto aby sme neovplyvnili ďalšie spracovanie Diskrétnou kosínusovou transformáciou. Táto ráta entropiu, tie neporiadne pixle čo vyčnievajú z davu, na blokoch pre jednotlivé zložky farieb zvlášť. Každá zložka používa 8 bitov, teda hodnoty do 256, tie sa teda upravia na rozsah od -127 po 128, aby sa mohla aplikovať kosínusová funkcia. Hodnoty sa zaokrúhlia na celé čísla a tie blízke sa zahodia, keďže nie sú až také významné. Nasleduje Quantizácia, zoskupenie farieb podobnej frekvencie s použitím výsledku kosínusovej transformácie. Výsledné dáta sa ešte komprimujú bezstratovou kompresiou. Najskôr sa použije „zigzag“, ktorý usporiada dáta tak, aby rovnaké čísla boli viac pri sebe a mohlo sa použiť run-length kódovanie (xxxxyyy -> 4x3y), potom sa používa huffmanovo alebo aritmetické kódovanie, ktoré svoje váhy nastavujú podľa počtu výskytov jednotlivých slov. Dekódovanie je samozrejme rovnaké až na niektoré stratové časti, ktoré sa domyslieť nedajú.

JPEG má široké uplatnenie, zaujímavou funkcionalitou pri použití spolu s digitálnymi fotoaparátmi je pridanie metadát. Exif (Exchangeable image file format) obsahuje prídavné informácie ako napríklad typ fotoaparátu, clonu, ohniskovej vzdialenosti, dátum a čas poprípade GPS súradnice. Ukladá sa tu aj náhľad fotografie pre veľmi rýchle zobrazenie na LCD obrazovke fotoaparátu, alebo aj programov na spracovanie fotografií, ktoré najskôr zobrazia Exif a až potom dekódujú JPEG obsah. Exif sa používa aj pri tvorbe panorám, keďže je v ňom uložený uhol záberu, čo eliminuje skreslenia pri spájaní.

Existuje aj progresívny JPEG formát, ktorý má upravenú práve fázu preusporiadania dát pred kompresiou. Umožňuje už pri prenose pomerne malej časti dát získať hrubý obraz, ktorý sa s pribúdajúcimi dátami postupne zjemňuje. Toto je výhodné najmä pri veľkých obrázkoch a pomalejšej prenosovej rýchlosti.

Nasledovníkom JPEG je JPEG2000, ktorý však nemá takú popularitu ani širokú podporu v aplikáciách. Prichádza však s mnohými vylepšeniami. Nedelí už obrázkov na striktné rovnaké bloky, čo odstraňuje veľké množstvo artefaktov vznikajúcich v obyčajnom JPEG. Namiesto diskretnej kosínusovej transformácie používa diskretnú vlnkovú transformáciu, ktorá dokáže reprezentovať obraz efektívnejšie a dosahuje vyšší kompresný pomer.

3.4. Vstupné dáta

Ako vstupy poslúžia najmä fotky vo formáte JPEG. Mnoho z nich som dostal od tvorcov projektu „Považské múzeum 3D on-line“ [20] z FMFI UK pod vedením Andreja Ferka. Textové dokumenty využiteľné najmä pri tvorbe informatívnej časti. Pri modelovaní mi poslúži pamäť a priestorová orientácia. Mám síce fotografiu pôdorysu zámku, ktorý je umiestnený pri vchode do budovy, to však zďaleka nie je dostačujúci údaj na tvorbu

reálne vyzerajúceho modelu. Napriek tomu, že musím pri modelovaní spoliehať na odhad, lebo presné rozmery nemám, aj zjednodušený model bude dostačovať pre zámery práce.

3.5. Použité aplikácie

Samozrejme v dnešnej dobe už autor nie je odkázaný len na seba a pri svojej práci používa aplikácie z iných zdrojov. Nie všetky vytvárajú veľké firmy za účelom zisku, ale sú tu i aplikácie distribuované voľne, pre prospech celého ľudstva. Vývoj aplikácie už dávno nie je záležitosť jedného človeka, počet takýchto známejších programov by sa dala spočítať na prstoch (len na ukážku dvaja veľikáni: Total Commander – Christian Ghisler, IrfanView – Irfan Skiljan). Mnoho vývojárov pracuje spolu na open-source aplikáciách (s otvoreným zdrojovým kódom), sú to aplikácie vydávané s licenciou, ktorá zabezpečuje ďalšie voľné šírenie a zabráni tak zneužití nezištnú prácu týchto autorov. Najznámejšie takéto licencie sú GPL (GNU General Public License), Mozilla Public License alebo BSD License. Použitie voľne dostupných aplikácií nemá len ekonomický rozmer, ich propagácia nie taká ako u komerčných produktov, musia teda za ne hovoriť ich činy. Ja taktiež budem pri svojej práci používať programy od iných autorov, budem však preferovať voľne šíriteľné alternatívy.

Vi [21] je obrazovo orientovaný textový editor, jeho autor Bill Joy ho vytvoril pre BSD systém už v roku 1976. Tento open-sourcový program je jednoduchý a zložitý zároveň. Problém je v tom, že jeho používanie nie je ani náznakom intuitívne a tak jeho kvality môžeme doceniť až potom čo sa ho naučíme používať. Keďže už čosi pamätá a myš bola v tom čase nedostatkovým tovarom, nie všetky klávesnice mali šípky a funkcionálne klávesy, vi si musel vystačiť bez nich. To sa môže javiť ako nevýhoda, ale v skutočnosti to urýchľuje prácu, keďže všetko sa dá spraviť bez prekladania rúk rýchlo a efektívne. Pri používaní prepíname dva základné módy: editačný, kde sa píše to čo stláčame a ovládací mód, kde kombinácie kláves vykonávajú funkcie editora. Vi je v súčasnosti rozšírený na všetkých unixových systémoch od komerčných HP-UX, IBM AIX, Solaris, cez BSD po ľubovoľný Linux, preto aspoň jeho základné znalosti sú devízou. Často sa používa jeho zľudštená verzia vim (Vi IMproved), ktorá zachováva funkcionálnosť vi a pridáva farby, rozpoznávanie syntaxe, používanie šípok a ďalšie. S príchodom okienkových systémov vznikol gvim, ten už poskytuje aj pohodlie menu, dá sa však pracovať aj bez neho.

PSPad [22] je voľne šíriteľný univerzálny editor, určený len pre operačný systém Microsoft Windows. Je určený na editáciu HTML stránok, a tomu má prispôsobenú aj funkcionálnosť ako napríklad dopĺňovanie príkazov, kontrola a zvýraznenie syntaxe, konverzie kódovania vrátane UTF-8, podpora pre CSS i XHTML.

IrfanView [23] sa sám definuje ako voľne šíriteľný prehliadač, pravda editor to nie je ale funkcionalitou zahanbuje ne jeden komerčný produkt. Autor Irfan Skiljan pochádza z Bosny a Hercegoviny, ale študoval tuhľa vo Viedni. V čom je tento program výnimočný? V prvom rade je neuveriteľne rýchly, podporuje veľa súborových formátov a to nielen obrazových ale aj video a zvuk, flash, ikony v exe a dll súboroch. Umožňuje dávkové spúšťanie konverzie spolu s možnou úpravou obrázku, prezentácia, skenovanie a tlač, úprava farieb, svetla, veľkosti, farebnej hĺbky, zobrazenie Exif informácie. Má bezstratovú rotáciu JPEG súborov vychádzajúcu priamo zo štandardu. Rôzne efekty a ďalšie prídavné moduly (PlugIns) podporuje aj filtre používané v Adobe Photoshope. To všetko v príjemnom intuitívnom GUI, o to príjemnejšom, že je zadarmo.

Picasa [24] je ďalším voľným programom na prezeranie fotografií z dielne Google. Na rozdiel od IrfanView zobrazuje primárne náhľady v viacerých fotografiách, tie sa triedia do skupín, predvolene podľa dátumu a umiestnenia na disku, hľadanie fotografie je tak oveľa jednoduchšie. Jeho editačné vlastnosti sú len základné, ale jednoduchosť a prehľadnosť sú jeho devízy. Tento program je výbornou voľnou na správu fotogalérie a obrazových súborov všeobecne. Zaujímavou funkcionalitou je publikovanie na internete a posielanie e-mailom jedným kliknutím vďaka kooperácii s ďalšími Google produktmi gmail, picasaweb a Google Earth.

GIMP [25] je plnohodnotným editorom bitmapovej (rastrovej) grafiky. Je distribuovaný pod GPL licenciou, spolu s GTK+ [26] resp. GTK+ 2 čo je balík na zobrazovanie v X window systémoch. Využitie GTK robí GIMP byť multiplatformovým programom. Obsahuje naozaj širokú paletu funkcií od štetcov, farieb, prechodov, selekcie, masiek, vrstiev po efekty, filtre a skriptovanie. Spolu so širokým záberom podporovaných formátov je silným konkurentom komerčných produktov.

Blender [27] sa používa ako 3D animačný program s GPL licenciou. Zaujímavosťou, že pôvodne to bol shareware a od autorov ho odkúpili za 100 000 €. Výhodou je, že je dostupný na viacero platforiem Windows, Mac Os, Linux, Solaris, BSD a mnoho ďalších. Podporuje relatívne dosť grafických primitív, ich úpravy, textúrovanie, a renderovanie obrázkov s použitím ray tracingu. Vytvára tiež animácie z nadefinovaných snímkou. Podpora skriptovacieho jazyka Python [28] dodáva programu možnosť veľkej rozšíriteľnosti. Scéna sa dá vyexportovať do VRML súboru vo VRML97 štandarde práve vďaka pythonu. Jediné čo môže prekážať je svojské užívateľské rozhranie, ktoré hlavne pre používateľov MS Windows môže pôsobiť zmätočne.

3.6. Tvorba modelu Považského múzea

Základom pre model som si zvolil jedno podlažie Budatínskeho zámku. Budú ho tvoriť steny reprezentované základnými grafickými primitívami, najmä kvádrami. Práca je o to komplikovanejšia, že steny majú rôzne šírky a spájajú sa pod nepravidelnými uhlami. Výraznou pomocou je mi pri tom už spomínaný pôdorys [CD]. V Gimpe som mu zvýšil kontrast, odstránil okolie a upravil veľkosť, tak aby 1 pixel na pôdoryse reprezentoval 10 jednotiek v 3D zobrazení. Meraním v obrázku som tak ľahšie odhadoval veľkosť a rozmiestnenie stien. Podobne som umiestnil schody, okná a priechody (dvere). Nasledovala fáza textúrovania, prilepenie obrázkov na objekty. Pri úprave obrázkov, orezanie, zosvetlenie, postačil zvyčajne IrfanView. Keď bolo treba aj upraviť skosenie a perspektívu zapojil som aj Gimp. Pri textúre na oknách som použil polovičnú priehľadnosť. Vložením viacerých kamier sa umožnilo prepínanie pohľadov. Naprogramovaním pohybu jednej z nich som vytvoril krátku animáciu [CD] vo formáte avi. Pri frekvencii 25 snímok za sekundu (norma Pal) som na 80 sekúnd musel vyrenderovať 2000 záberov. S použitím vyššej kvality zahrnujúcej anti-aliasing a anizotropické filtrovanie textúr to na mojom počítači (CPU 2GHz) trvalo 2 hodiny, s využitím ray tracingu 12 hodín. Druhým výstupom je už spomínaný formát VRML97, ten môžem umiestniť aj na internet. Využitím programu Picasa vyexportujem fotogalériu priamo do HTML. Nasleduje len práca v PSPade a vytvorenie úvodnej stránky, dizajnu a menu, ktoré spojí všetky časti dokopy.



Obr. 8: Model Považského múzea

4. Záver

V tejto práci som sa pokúsil popísať virtuálne svety, ich definovanie a základné spôsoby použitia. Napriek tomu, že sa počítač v tejto sfére nevyužíva tak často ako v iných sférach popísal som nejaké tie prvé zrnká využitia virtuálnych svetov, napríklad ako aj virtuálne múzea. Boli tu popísané technológie používané na tvorbu 3D prostredí. Od základných mriežkových modelov cez textúrovanie až po využitie osvetlenia. Na každom stupni tvorby 3D modelov som sa snažil nájsť vylepšenia, ktoré vedú k lepším a reálnejším výsledkom. Uviedol som definície formátov využívaných za tvorbu internetových stránok, spracovanie obrazu a reprezentáciu 3D modelov. Výber formátov zohľadňuje použitie na internete a tým univerzálnosť riešenia aby bolo dostupné širokému spektru užívateľov. Prinášam prehľad použitých aplikácií, ktoré využijeme pri tvorbe virtuálneho prostredia a internetových stránok. Snažil som sa použiť voľne dostupné riešenia, ktoré ocenia najmä študenti a autori virtuálnych svetov, ktorí nemajú finančnú podporu sponzorov a rôznych inštitúcií.

V práci som taktiež definoval určité typické problémy pri návrhu virtuálnych svetov, a naznačil aj určité riešenia, či už vylepšenia kvality zobrazenia, optimalizácie veľkosti pre internet alebo kompatibility. Na záver by sa patrilo uviesť, prečo je dôležité sa podobnou problematikou zaoberať. Problematika je pomerne nová a vďaka informatizácii a pokroku výkonu počítačov a stáva zaujímavá pre viacero ľudí, nielen vo vedeckých kruhoch. Možno sa ani nenazdáme a virtuálne prostredia na internete budú onedlho jeho bežnou súčasťou.

4.1. Príloha CD

Keďže v práci ide o znázornenie virtuálneho priestoru, ako prílohu pridávam CD s týmto 3D obsahom. VRML súbor je tu, na rozdiel od internetu, dostupný aj v nekomprimovanom tvare spolu s použitými textúrami. Animácia vo formáte avi s použitím XViD kompresie. V prílohe sa nachádzajú aj ďalšie obrázky a videá viažuce sa k spomínaným pojmom a technológiám. Keďže sa zaoberám hlavne zobrazovaním, obrázky sú správnym doplnkom k textovým popisom.

Pramene

- [Moor65] Cramming More Components Onto Integrated Circuits
<http://download.intel.com/research/silicon/moorespaper.pdf>
- [Qvor02] QVORTRUP, L. ed. 2002. *Virtual Space: Spatiality in Virtual Inhabited 3D Worlds*. Springer-Verlag London Berlin Heidelberg. ISBN 1-85233-516-5.
- [1] Virtual Worlds Review <http://www.virtualworldsreview.com>
- [2] Second Life <http://www.secondlife.com/>
- [3] OpenGL <http://www.opengl.org/>
- [4] Wikipedia http://en.wikipedia.org/wiki/Texture_filtering
- [5] IT News – ray tracing
http://www.itnews.sk/buxus_dev/generate_page.php?page_id=48467&usr_forum_page_size=30&usr_forum_page_id=0&usr_forum_msg_list%5B766890%5D=ON
- [6] ACM SIGGRAPH <http://www.siggraph.org/>,
http://www.siggraph.org/education/materials/HyperGraph/radiosity/overview_1.htm
- [7] Paul Debevec <http://www.debevec.org/>
- [8] Hutnícke múzeum železiarní podbrezová <http://www.zelpo.sk/hmuzeum/>
- [9] Olga Filippova, Museum of the USSR <http://www.olgafilippova.com/>
- [10] Vincent Van Gogh Gallery <http://www.vangoghgallery.com>
- [11] XHTML 1.0, World Wide Web consortium <http://www.w3.org/TR/xhtml1/>
- [12] XML, World Wide Web consortium <http://www.w3.org/XML/>
- [13] CSS, World Wide Web consortium <http://www.w3.org/Style/CSS/>
- [14] VRML, World Wide Web consortium <http://www.w3.org/MarkUp/VRML/>
- [15] Floppy's Web3D guide
<http://web3d.vapourtech.com/tutorials/vrml97/en/tut11.html>
- [Kuzev01] Ing. Žofia Kuzevičová a Ing. Štefan Kuzevič, Vizualizácia mestskej zástavby a jej analýza prostredníctvom VRML, Acta Montanistica Slovaca – mimoriadne číslo, 2001
- [16] Octaga <http://www.octaga.com/>
- [17] Cortona <http://www.parallelgraphics.com/products/cortona/>
- [18] VrmlMerge <http://vrml.deem7.com/vrmlmerge/>
- [19] Photographic Experts Group <http://jpeg.org/>
- [20] Považské múzeum 3D online <http://www.sccg.sk/~projects/pav-pm3d/>
- [21] Textový editor Vi <http://en.wikipedia.org/wiki/Vi>
- [22] PSpad www.pspad.com/
- [23] IrfanView <http://www.irfanview.com/>
- [24] Picasa <http://picasa.google.com/>
- [25] GIMP <http://www.gimp.org/>
- [26] GTK+ <http://www.gtk.org/>
- [27] Blender <http://www.blender.org/>
- [28] Python <http://www.python.org/>
- [CD] CD príloha