

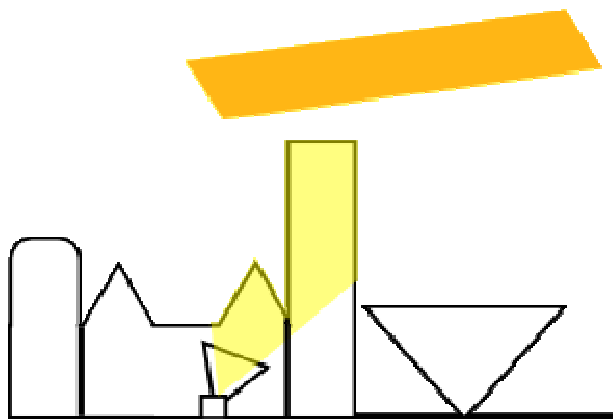
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Katedra aplikovanej informatiky

9.2.9 aplikovaná informatika



OBLOHA AKO INFORMAČNÝ BILBOARD VO VRML
BAKALÁRSKA PRÁCA



Autor: **Jana Dadová**

Vedúci bakalárskej práce: **Doc. RNDr. Andrej Ferko PhD.**

Bratislava, 2008

Čestné prehlásenie

Čestne prehlasujem, že som túto prácu robila samostatne a uviedla som všetku použitú literatúru.

.....
Jana Dadová

Pod'akovanie

Touto cestou by som sa rada poďakovala Doc. RNDr. Andrejovi Ferkovi, PhD. za venovaný čas, odborný dohľad a pripomienky.

Ďakujem aj Petrovi Kánovi za poskytnutie modelu kostola, ktorý v práci využívam.

Ďakujem Davidovi Běhalovi za pomoc pri vymýšľaní názvu aplikácie.

Abstrakt

V práci navrhujem spôsob zobrazenia informácií do priestoru oblohy vo virtuálnom prostredí, najmä na internete a poskytujem ich viditeľnú chronologizáciu.

V rámci práce sa objasňujem pojmy ako virtuálna realita a virtuálne prostredie. Zaoberám sa jazykmi VRML, ako aj novším formátom X3D, ich využitím pri transformácii skyboxom a následným prepojením na skriptovací jazyk PHP na strane servera a databázu MySQL.

Interakciu s používateľom riešim pomocou JavaScriptu, ktorý používam v technológii Ajaxu, konkrétne využitím iniciatívy s názvom Ajax3D. Tieto pojmy v práci bližšie vysvetľujem a následne aj implementujem v praktickej časti práce.

Abstract

In this thesis I propose new way of information mapping to virtual skybox. They are in visible time correnspondence.

I explain terms as virtual reality and virtual enviroment. This thesis deal with VRML and newer X3D format, their use in skybox transformation and connection to script language PHP and MySQL database.

Moreover I use Javascript in Ajax texhnology for user interaction. Newer Ajax initiative is called Ajax3D and I explain all this terms and implement this technologies in practical part of thesis.

Kľúčové slová:

virtuálna realita, VRML, X3D, Ajax3D

Obsah

Úvod.....	7
1. Analýza problematiky	8
1.1 Základné pojmy.....	8
1.1.1 Virtuálna realita.....	8
1.1.2 Sky box	9
1.1.3 VRML	10
1.1.4 X3D.....	11
1.1.5 ECMAScript.....	12
1.1.6 Java.....	12
1.1.7 PHP	13
1.1.8 AJAX	14
1.1.9 AJAX3D.....	15
1.2 Podobné projekty	16
1.2.1 Snow Crash	16
1.2.2 Alpha World.....	17
1.2.3 Second Life	18
1.2.4 Google Earth	19
1.2.5 Virtual Reality Search Machine [han99].....	19
2 Špecifikácia a návrh riešenia.....	20
2.1 Cieľ práce	20
2.2 Požiadavky	20
2.2.1 Klient – Server technológia.....	20
2.2.2 Všeobecné požiadavky	21
2.2.3 Konkrétne požiadavky	23
2.3 Platforma.....	25
2.4 Technológia.....	25
2.4.1 Spracovanie	25
2.4.2 Dáta, formáty typy súborov.....	27
2.4.3 Použité technológie softvér	27
2.4.4 Použitý softvér	28
2.5 Používateľské rozhranie (user interface).....	28
2.6 Predpríprava	29
2.7 Návrh riešenia	30
3 Implementácia.....	31

3.1 Práca so scénou prostredníctvom Ajax engine.....	32
3.2 Štruktúra databázy.....	33
3.2.1 Štruktúra databázy entít.....	33
3.2.2 Štruktúra databázy SkyPortov.....	33
3.3 Obsah jednotlivých súborov.....	34
3.3.1 index.html	34
3.3.2 sensors.js	34
3.3.3 ajax3d.js	36
3.3.4 HttpRequest.js	37
3.3.5 php_response.php.....	37
3.4 Vzájomné prepojenie jednotlivých funkcií.....	38
3.4.1 Vytvorenie stránky	38
3.4.2 Otočenie pohľadu.....	39
3.4.3 Posunutie entít na oblohe	40
3.4.4 Vrátenie pohľadu.....	41
3.4.5 Zmena výberu entít	42
3.4.6 Zobrazenie podrobnejších informácií o entitách	42
3.5 Plány do budúcnosti	43
Záver	44
Zoznam obrázkov.....	45
Zoznam použitých skratiek	46
Prílohy.....	47
Štruktúra .x3d súboru	47
Štruktúra .wrl súboru.....	47
Bibliografia	48

Úvod

„It's our mission to connect us all to an online world that advances the human condition.“

„Naším cieľom je spájať všetkých do virtuálneho sveta, ktorý obohacuje ľudské možnosti.“

vízia Linden Labs [tao08]

Žijeme v dobe, v ktorej informácie tvoria veľkú časť nášho sveta. Z každej strany na nás prúdia. Od billboardov¹, televízie, nápisov na autách, až po informácie na čistiacom prípravku alebo potravinách. Na to, že sme informovaní o všetkom, všade a niekedy bez vlastného pričinenia sme si tak zvykli, že pokiaľ niekde chýbajú informácie v podobe textu, dožadujeme sa vysvetlenia. Napríklad, keď vidíme reklamu bez nápisu o akú firmu ide, hľadáme ďalšie pokračovanie ako vysvetlenie o čo vlastne ide.

Vychádzajúc z tohto poznatku prišla myšlienka spracovania témy “Obloha ako informačný billboard” prirodzená. Vo virtuálnom prostredí je obloha často pokladaná za nevyužitý priestor. V mnohých spracovania virtuálnych svetov nájdeme na tomto priestore len rôzne animácie dennej, alebo nočnej oblohy. Prečo však nevyužiť tento priestor vo forme ponúkaniu ďalších informácií v prípade záujmu používateľa?

Internet ponúka ideálne prostredie pre realizáciu tohto projektu, keďže počet používateľov sa neustále zvyšuje a ich záujem o informácie stúpa a kapacita viditeľných plôch má ohraničenia. Rovnako vývoj predpokladá záujem o virtuálne svety, ktoré sú pohodlnou a lacnou voľbou spoznávanie sveta.

Práca je doplnená praktickým využitím teoretických riešení na priloženom CD.

¹ billboard – reklamný pútač

1. Analýza problematiky

Virtuálne svety, ktoré sa nachádzajú na internete, sa väčšinou snažia priblížiť používateľom svet reálny, preto ponúkajú najmä možnosti, ktoré si vieme ľahko predstaviť a sme na ne zvyknutí. Príkladom týchto možností je chôdza, ktorá väčšinou umožňuje pohyb v horizontálnej rovine a pohľadmi hore a dole, teda presne tak, akoby sme niekde kráčali v reálnom svete. Časť projektov ide smerom k rozšíreniu možností z reálneho sveta, ako projekt Second Life, ktorý umožňuje lietanie avatarov². Tento prístup môže byť neprirodzeným pre niektorých používateľov, avšak mladšia generácia zvyknutá na virtuálne svety z počítačových hier, ktoré často krát poskytujú neprirodzené ovládanie, alebo nadprirodzené schopnosti takéto rozšírenia privíta. Náš prístup bude zlepšovať informačnú stránku urbánnych modelov, pretože bude pridaná možnosť zobrazovania relevantných chronologických entít, ktoré si bude môcť prehliadať na oblohe. V reálnom svete môžeme oblohu popísať pomocou počítačovej grafiky ako animáciu obrázkov mrakov, či hviezd, preto sa táto animačná technika používa aj vo väčšine projektov virtuálnych svetov.

Problém, ktorý riešim v práci je prepojenie entít uložených v databáze s virtuálnym svetom. Zároveň prinesiem interakciu s používateľom, aby mohol aplikáciu ovládať a určiť si, ktoré entity chce zobraziť.

Tento problém nie je jednoduchý a na nasledujúcich stranách vysvetlím základe pojmy a ukážem riešenie problému v podobných projektoch.

1.1 Základné pojmy

1.1.1 Virtuálna realita

„Po počiatočnom prekvapení z nového pojmu *virtuálna realita* sa väčšina z nás už ani nepozastavuje nad nezmyselným spojením dvoch slov s opačným významom (virtuálny – neskutočný, reálny - skutočný) a správne si vysvetľujeme virtuálnu realitu ako prostredie, ktoré umožňuje prácu v trojrozmernom priestore, vymodelovanom v pamäti počítača“ [zar97].

² avatar – používateľova reprezentácia, pomocou ktorej môže interagovať s virtuálnym prostredím

Virtuálna realita sa v posledných rokoch dostáva z vedecko-fantastických filmov, či hier aj na obrazovky počítačov bežných používateľov. Veľkú zásluhu na tomto rozvoji má internet ako aj snaha priblíženia sveta pomocou projektov virtuálnych miest.

Aplikácie z oblasti virtuálnej reality by mali spĺňať nasledovné parametre:

- všetky deje sú vykonávané a zobrazované v reálnom čase, čo znamená s okamžitou odozvou a s frekvenciou 25 snímok za sekundu
- scéna (virtuálny svet) a objekty sú trojrozmerné, alebo sa javia ako trojrozmerné
- používateľ interaktívne vstupuje do scény a vplýva na ňu, objekty reagujú na používateľa
- virtuálne prostredie kombinuje multimediálne prvky – zvuk, video, 2D obrazy [zar00]

Z týchto vlastností je najdôležitejšia prvá. Pokiaľ by zobrazovanie alebo interakcia boli pomalšie ako používateľ predpokladá, resp. ako je zvyknutý z reálneho sveta, ten virtuálny by bol neprirodený a používateľ by stratil záujem pokračovať. Aplikácia by preňho nebola atraktívna.

1.1.2 Sky box

Sky box [sky07] je pomenovanie pre otexturovaný kváder, ktorý sa používa na zobrazovanie oblohy, prípadne okolitého sveta. Celý svet je týmto veľkým kvádom obalený a používateľ nadobúda dojem pozorovania vzdialeného okolia (teda napr. obloha, vzdialené pohoria) bez nutnosti ich modelovania a renderovania. Na Skybox je možné okrem statickej textúry namapovať aj dynamické obrazy a tým získať realistickejší obraz. Táto technika je často používaná vo virtuálnych svetoch zobrazovaných v reálnom čase ako sú napríklad počítačové hry. V mojej práci je cieľom dynamicky meniť obsah na oblohe a nie len ho animovať preto som sa rozhodla tento efektívny spôsob nevyužiť. Príklad textúry použitej pre Sky box je zobrazený na nasledujúcom obrázku:



Obr.1 [sky01]: Textúra pre Sky box.

1.1.3 VRML

Virtual Reality Modeling Language [VRML] je medzinárodne uznávaný formát slúžiaci na popis trojrozmernej virtuálnej reality ako aj formát textového súboru, do ktorého sa tento popis ukladá. Keďže je medzinárodne uznávaný a podporovaný prehliadačmi, konkrétne Internet Explorer (Microsoft) a Navigator (Netscape) dodávajú prehliadač VRML svetov spolu s prehliadačom WWW stránok, do ostatných je nutné si doinštalovať plugin³, napr. Cosmo Player (Silicon Graphics, Inc.), World View (Intervista Software, Inc.) alebo Cortona (Parallel Graphics).

Historické začiatky jazyka VRML nachádzame vo firme Silicon Graphics, Inc., ktorej programátori začiatkom 80. tých rokov navrhli knižnicu pre prácu s priestorovými objektmi. Neskôr postupným vývojom v roku 1995 bola navrhnutá 1. verzia VRML 1.0, táto bola doplnená o jednoduché zvuky a animáciu a vytvorila verziu VRML 1.1. Na medzinárodnej konferencii SIGGRAPH v New Orleans v Marci 1996 zo šiestich návrhov nakoniec vyhral Moving Worlds a stal sa základom pre verziu VRML 2.0,

³ plugin - zásuvný modul, ktorý ponúka ďalšiu funkčnosť

ktorá sa po doplnení stala základom ISO štandardu VRML 97 [VRML97]. V ďalšom texte budeme pod názvom VRML vždy myslieť vyššie uvedený štandard VRML 97.

Samotný jazyk VRML je teoreticky využiteľný pre akúkoľvek počítačovú aplikáciu, pre zložitejšie aplikácie je však nutné použiť taktiež ďalší programovací jazyk. Najčastejšie sú kombinácie s jazykmi: ECMAScript, Java, SQL, C++.

Medzi základné vlastnosti VRML patria [zar00]:

- podpora hraničnej reprezentácie objektov⁴
- scéna organizovaná do stromovej štruktúry⁵
- jazyk umožňuje spoluprácu s ďalšími programovacími jazykmi (Java, Javascript)
- virtuálne svety VRML môžeme vkladať do stránok a rámkov HTML⁶
- obsahuje prostriedky pre popis animácie a interakcie
- je textovým formátom, nezávislým na platforme
- orientácia na internet

1.1.4 X3D

EXtensible 3D Graphics, čoho skratka je X3D [X3D] je ISO štandard, ktorý podobne ako VRML umožňuje vytvárať robustné, plne funkčné internetové aplikácie podporované rôznym voľne šíriteľným softvérom. V porovnaní s VRML, X3D je novší formát poskytujúci viacero funkcií, avšak prístup vytvárania virtuálnych svetov je podobný. Formát X3D poskytuje navyše animáciu avatara ako aj modelovanie pomocou NURBS kriviek. Podobne ako VRML aj X3D súbor môže mať koncovku *.wrl*. Tento zápis je starší a syntax v tomto súbore je rovnaká ako vo VRML. Formát X3D však umožňuje zápis virtuálneho sveta aj do súboru s koncovkou *.x3d*, ktorého syntax sa viac podobá na XML dokumenty a prispôsobuje sa tým trendu vo vyvíjaní internetových aplikácií.

Viditeľným rozdielom z používateľského hľadiska je potreba inštalovania iného 3D prehliadača pre VRML a iného pre X3D. Väčšina týchto prehliadačov v podobe

⁴ hraničná reprezentácia – teleso je zadané pomocou vzájomných vzťahov medzi vrcholmi, hranami, stenami

⁵ stromová štruktúra – objekty sú organizované do vetviacej sa logickej štruktúry, ktorú nazývame strom

⁶ HTML – hypertext markup language – značkovací jazyk používaný na popis web stránok

zásuvných modulov však dokáže zobrazovať oba tieto formáty, keďže výrobcovia sa snažia prispôbiť požiadavkám používateľa. Príklady X3D prehliadačov: Flux Player (Media Machines, Inc.), Xj3D (web3D Consortium), Octaga.

1.1.5 ECMAScript

ECMAScript [ECMA], alebo tiež známy pod názvom JavaScript⁷ je skriptovací jazyk⁸, ktorý bol vytvorený prevažne pre účely použitia v internetových prehliadačoch. Pôvodný názov JavaScript bol nakoniec zmenený na návrh firmy SUN, ktorá poukázala na podobnosť názvu s jej programovacím jazykom Java, ktorý je odlišný, aj keď má mnoho podobných čŕt. V práci používam názov JavaScript, pretože tento je známejší pre programátorov web stránok a nachádza sa v názve Ajax3D (vysvetlené nižšie).

Medzi ďalšie vlastnosti som zaradila neschopnosť komunikácie s diskovými súbormi a operačným systémom. Tento jazyk je veľmi obľúbený pre tvorcov web stránok, pretože má relatívne jednoduchú implementáciu a jeho syntax je podobná jazyku C++⁹.

V kombinácii s VRML je ľahko použiteľný za pomoci uzlu Script a toto rozšírenie umožňuje lepšiu interakciu s používateľom. Niektoré objekty v JavaScripte boli vyvinuté špeciálne pre VRML. Sú to napríklad: SFCOLOR, SFImage, SFRotation... Pomocou tejto kombinácie je možné zdefinovať rôzne funkcie, ktoré sa vykonajú po reakcii s avатарom. Ďalším objektom je objekt VRMLMatrix, ktorý ponúka efektívnu prácu s rôznymi transformáciami.

JavaScript je rovnako možné použiť v samostatnom súbore s príponou .js a X3D uzly vytvárať a rušiť priamo z tohto skriptu, ktorý je vložený do HTML¹⁰ stránky.

1.1.6 Java

Java [JAVA], ktorej tvorcom je tím Sun Microsystems, patrí medzi mladšie programovacie jazyky, napriek tomu sa pre ňu nachádza široké využitie. Ide

⁷ Javascript - jazyk ECMAScript s dialektom, ktorý v decembri 1995 vytvorili Sun Microsystems a Netscape

⁸ skriptovací jazyk – programovací jazyk, ktorého kód je spúšťaný za behu iným procesom, nemusíme ho najprv skompilovať

⁹ C++ - rozšírený objektovo orientovaný jazyk

¹⁰ HTML – HyperText Markup Language – používaný na vytváranie statických web stránok, má koncovku .html

o objektovo orientovaný jazyk, ktorý sa dá využiť aj vo forme appletov¹¹, a tak dopĺňať obsah a funkčnosť stránky. Pre spustenie appletu v okne prehliadača treba mať nainštalovaný voľne dostupný plugin – JRE (Java Runtime Environment).

Jazyk Java sa dá taktiež kombinovať s VRML, sú na to špeciálne určené knižnice a funkcie, ktoré sa volajú priamo z appletu. Pomocou týchto funkcií sa dá pristupovať k jednotlivým uzlom, ktoré tvoria virtuálny svet a aj ich meniť podobne ako v JavaScripte.

Ďalšou možnosťou je použitie priamo vo VRML uzle Script. JDBC (Java Database Connectivity) rozhranie ponúka aj prácu s databázami a jednoduché použitie sieťových tried pre komunikáciu po sieti, umožňuje vytvorenie kooperatívneho (multi-user)¹² sveta. Ide o nástroj, ktorý sa taktiež dá použiť na doplnenie interakcie a prepojenie na databázu vo prostredí VRML. Na prehliadanie takto generovaného sveta nie je potrebný prehliadač VRML, alebo X3D ale je potrebný prehliadač appletov, čiže Java Runtime Environment, ktorý je voľne stiahnuteľný a v dnešnej dobe existuje mnoho internetových aplikácií, ktoré vyžadujú mať tento zásuvný modul nainštalovaný, čo znamená, že mnohí používatelia nemusia si nič dodatočne inštalovať.

1.1.7 PHP

PHP (**P**ersonal **H**ome **P**age) [PHP] je voľne dostupný skriptovací jazyk, založený na myšlienke klient-server¹³. Skript je vykonávaný na strane servera a umožňuje vytváranie moderných dynamických a interaktívnych webových aplikácií. Obsahuje funkcie, ktoré umožňujú jednoduché prepojenie na databázový server, preto sa často používa v kombinácii s databázovým systémom, napríklad MySQL, alebo PostgreSQL. Rovnako podporuje Objektovo orientované programovanie. Momentálne najnovšou a aj najpoužívanejšou je verzia PHP5 [PHP], ktorá od predchádzajúcich verzií prešla rôznymi zmenami a vyšla v roku 2004.

Pre spustenie kódu PHP treba nainštalovať HTTP server, akým je napríklad Apache, ktorý je tiež voľne dostupný, dajú sa stiahnuť aj balíky aplikácií, ktoré podporujú

¹¹ applet – softwarový komponent, ktorý sa často vkladá do web stránok na doplnenie obsahu

¹² multi-user – operačný systém, alebo aplikačný software umožňujúci pripojenie viacerých používateľov

¹³ klient-server - softvérová architektúra, bližšie vysvetlená v kapitole špecifikácia

priamo kombináciu PHP a nejakého databázového servera (napríklad PHP Home Edition).

1.1.8 AJAX

Skratka AJAX znamená **A**synchronous **J**avaScript and **X**ML [AJAX] a prvý krát ju použil J. J. Garret v roku 2005 [JJG]. Nejde však o nový programovací jazyk, ide skôr o nový prístup v použití existujúcich technológií. Popularita týchto postupov stúpila, keď firma Google použila AJAX v aplikácii Google Suggest (Google Maps, Gmail ...).

AJAX je založený na JavaScripte a príkaze XMLHttpRequest(). Tento príkaz je volaný priamo zo skriptu a umožňuje posielanie informácií medzi serverom a klientom bez nutnosti natiiahnutia stránky. Umožňuje použitie a prepojenie JavaScriptu na strane klienta (používateľa, resp. jeho internetového prehliadača) a PHP alebo ASP na strane servera.

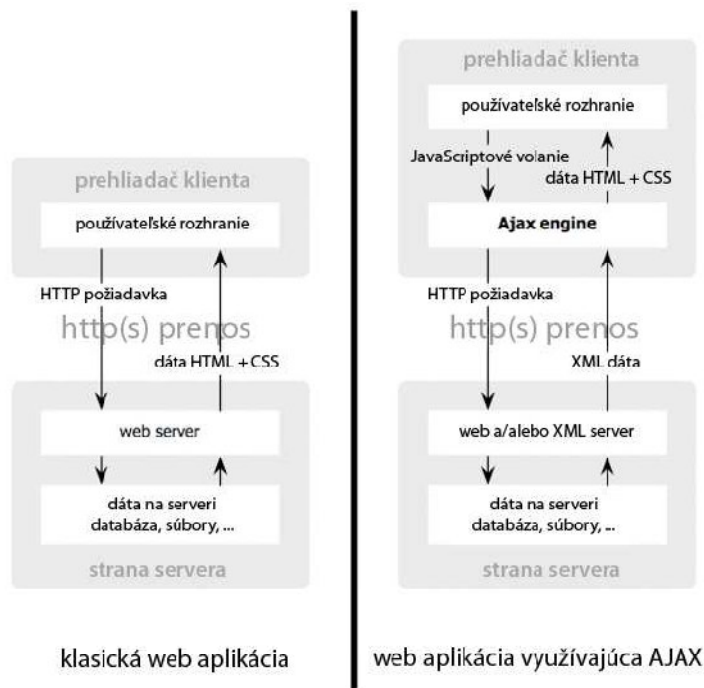
AJAX umožňuje vytváranie internetových aplikácií, ktoré sú z hľadiska odozvy oveľa vhodnejšie a rýchlejšie ako staršie postupy vytvárania dynamických aplikácií, pretože umožňujú zmenu obsahu časti stránky bez nutnosti znovu natiiahnutia celého obsahu.

Všeobecne môžeme povedať, že Ajax zahŕňa:

- internetovú prezentáciu využívajúcu XHTML a CSS,
- dynamické zobrazovanie a interakciu použitím Document Object Modelu¹⁴,
- výmenu dát pomocou XML a XSLT,
- asynchrónne získavanie dát pomocou metódy XMLHttpRequest,
- JavaScript, ktorý spája všetky tieto technológie pod názvom Ajax.

Namiesto stiahnutia celej stránky zo servera na začiatku práce, internetový prehliadač stiahne Ajaxový engine (napísaný v JavaScripte). Tento engine je zodpovedný za renderovanie používateľského rozhrania, ako aj za komunikáciu medzi používateľom a serverom. Komunikácia medzi serverom a internetovým prehliadačom je asynchrónna – nezávislá na komunikácii so serverom [JJG].

¹⁴ Document Object Model - Štandard objektovo orientovaného prístupu k reprezentácii HTML (a iných SGML podobných) dokumentov – dokument je reprezentovaný stromom dátových objektov a metód, ktoré slúžia na manipuláciu s obsahom objektov.



Obr. 2 [AJAX]: Porovnanie práce štandardnej web aplikácie a aplikácie využívajúcej AJAX.

1.1.9 AJAX3D

Skratka Ajax3D znamená Ajax + X3D a základom je spojenie týchto dvoch technológií. Táto iniciatíva vznikla v auguste 2006, keď jeden zo spolutvorcov formátu X3D, Tony Parisi, navrhol tento nový prístup k vytváraní interaktívnych dynamických internetových aplikácií, čo podložil aj publikovaním whitepaper¹⁵ ako aj príkladmi použitia tejto technológie [AJAX3D].

Ajax3D kombinuje silu X3D ako štandardu, ktorý dokáže v reálnom čase zobrazovať 3D virtuálny svet na internete a jednoduchý prístup v Ajaxe. Ajax3D rozvíja X3D SAI (Scene Acces Interface), čo je vlastne X3D ekvivalent k DOM (Document Oriented Model) na kontrolovanie scény pomocou JavaScriptu [AJAX3D]. Dynamickosť scény je potom dotváraná pomocou Ajaxu, ktorý tiež umožňuje prácu s databázou, resp. so súborami na strane servera a následným odosielaním na vstup funkciám JavaScriptu.

Ajax3D je aplikácia v internetom prehliadači, ktorý:

- používa SAI na prístup k 3D scéne,
- používa DOM na manipuláciu s obsahom web stránky, zmeny v 3D scéne

¹⁵ whitepaper – prvá publikácia o začínajúcom projekte

- používa tzv. server-request metódy¹⁶ na ukladanie a získavanie dát.

Na dôkaz funkčnosti Ajax3D pre konferenciu SIGGRAPH 2006 vytvoril Larry Rosenthal hru The Strikefighter, ktorá využíva X3D na zobrazovanie sveta, Javascript na interakciu a dynamické zobrazovanie, Ajax na ukladanie rebríčku skóre do databázy. Bez existencie Ajax3D by nebolo možné vytvoriť podobnú hru, ktorá by bežala v 3D prostredí v internetovom prehliadači [AJAX3D].

1.2 Podobné projekty

Existuje veľké množstvo projektov, ktorých úlohou je vytvárať kooperatívne virtuálne svety pre virtuálne populácie. V nasledujúcich odsekoch sa spomínajú len niektoré, ale významné z nich.

1.2.1 Snow Crash

Román Snow Crash [ste92], v českom preklade Sníh, predstavuje hypermodernú víziu do budúcnosti Ameriky. V deji sa prelína reálny svet s virtuálnym, ktorý je zobrazený pomocou špeciálnych okuliarov a počítača. Reálny svet je zničený a väčšina ľudí je donútená žiť v malých neútulných príbytkoch a žiť sa nevhodnou prácou, preto sa väčšina z nich pripája na Sieť do virtuálneho sveta s názvom Metaverzum, v ktorom je možné žiť svoj vysnívaný svet pomocou naprogramovaných avatarov, miest aj činností. Lepší programátori si vedia fungovanie tohto sveta prispôsobiť doprogramovaním vhodných aplikácií. Metaverzum umožňuje stretávanie a komunikáciu medzi ľuďmi, ktorý túto možnosť využívajú, pretože prostredníctvom avatarov majú možnosť vidieť emócie prostredníctvom avatara s ktorým komunikujú (čo je výhoda oproti telefónu, či mailu) a zároveň nemusia celú precestovať celý svet, iba sa pripojiť (výhoda oproti realite).

Toto vykreslenie virtuálnej reality ako vízia budúcnosti je jednou z prvých, napriek tomu veľmi presne opisuje cieľ mnohých projektov snažiacich sa o vytvorenie kooperatívnych virtuálnych svetov. O obsahu oblohy sa v knihe nepíše, napriek tomu interakcia používateľa je vykreslená veľmi vhodným spôsobom a preto v mojej práci sa budem snažiť priblížiť sa k týmto ideám.

¹⁶ server-request metódy – v prostredí Ajax3D sú XMLHttpRequest a createX3DFromURL

1.2.2 Alpha World

Alpha World [alp95] je najstarší kooperatívny 3D virtuálny svet na internete, ktorý dovoľuje spoluprácu a komunikáciu používateľov. Vznikol ako potreba 3D nástroja, v ktorom by mohli avatari spolupracovať. Keďže tento svet bol prvý svojho druhu, museli sa tu vyriešiť niektoré problémy, ako napríklad:

- pohyb po scéne – aby sa avatari mohli stretávať, bol vymyslený štvorsmerný systém pohybu, neskôr dokonca verejná doprava, ktorá umožňovala lepšie stretávanie a nájdenie hľadaného avatara.
- zrýchlený pohyb (warping) a teleportovanie – bol implementovaný pre potreby používateľov, čo však malo za následok, že niektorí prepadli za okraj virtuálneho sveta. Prvým takýmto avatarom bola „Laura“.
- detekcia kolízií¹⁷, aby sa prostredie priblížilo čo najviac reálnemu svetu, kde sa všeobecne prechádzať cez stenu nedá, ale naopak napríklad, cez blato áno
- rôzne gestá a viditeľný pohyb avatarov – pre toto najali tanečnicu Pamelu Forth v roku 1996, ktorá im pomohla spraviť tento pohyb prirodzenejší.

V ďalších verziách vznikla možnosť vytvárania viacerých svetov, ktoré sú prepojené pomocou teleportovania¹⁸, a preto si ktokoľvek mohol zakúpiť svet, ktorý bude pripojený k ostatným a vybudovať si ho podľa svojich predstáv. Takto vznikol komplex Active Worlds [act96], ktorý funguje dodnes.

Pre vstup do sveta je potreba nainštalovania špeciálneho prehliadača, ktorý umožní zobrazovať obsah. Do tohto sveta sa dá vstúpiť aj zadarmo ako návštevník, ktorý na rozdiel od ostatných nemá právo na zmenu obsahu, ale môže komunikovať s ďalšími avatarmi. Všeobecne platí, že obsah vytvárajú používatelia.

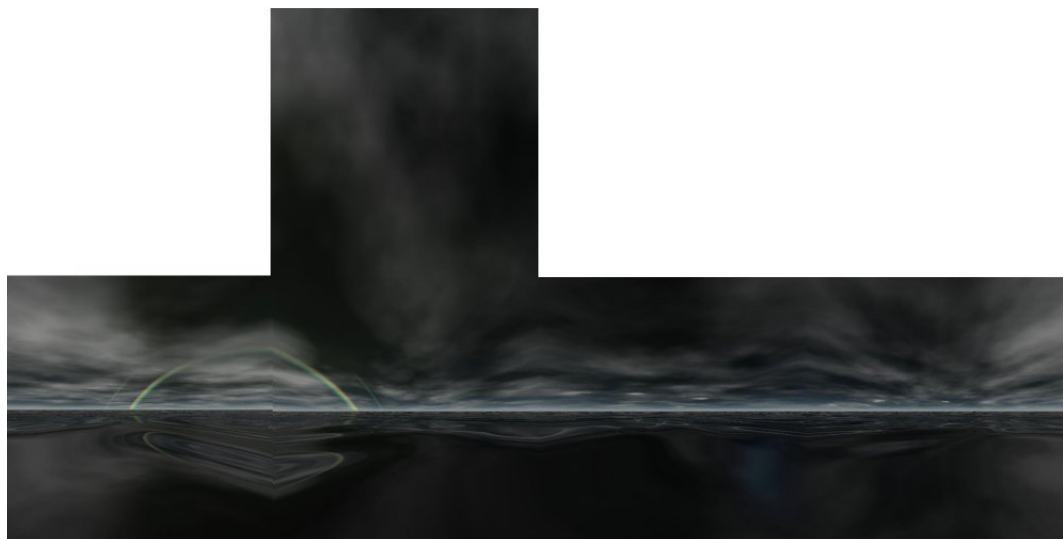
Pokiaľ ide o oblohu v tomto svete platí, že sa dá nastaviť tzv. „backdrop colour“, ktorá nastaví farbu oblohy, pod ktorou je zvyčajne na pozadí obrázok. Táto farba pozadia sa dá meniť podľa toho, či je deň, alebo noc. V Patagónii¹⁹ je to zvyčajne obrázok pohoria a tmavomodrá farba, bez ohľadu na to, či je deň, alebo noc [qvo02].

¹⁷ detekcia kolízií – programátorské riešenie, ako zakázať avatarom prechádzať cez steny

¹⁸ teleportovanie – presúvanie objektov z miesta na miesto priamo, bez viditeľného presúvania priestorom

¹⁹ Patagónia – ďalší zo svetov Active Worlds, ktorý je veľmi populárny

Rovnako ako pre ostatné virtuálne svety aj pre Active Worlds bolo vytvoreným mnoho textúr pre Sky boxy. Jedna z týchto textúr je zobrazená na nasledujúcom obrázku:



Obr.3 [SAW] : Textúra pre Skybox v Active Worlds virtuálnych svetoch

1.2.3 Second Life

Projekt Second Life [sec] je podobný už vyššie spomínanému Alpha World. Second Life vytvorila skupina z Linden Labs., ktorá vznikla v roku 1999, špeciálne pre vývoj revolučne novej formy prostredia, v ktorom môžu používatelia navzájom interagovať, komunikovať a kooperovať. Second life, ako 3D svet bol predstavený verejnosti v roku 2003 [lin95].

Podobne ako Alpha World aj tento svet vytváranú používatelia, čo znamená, že používatelia majú možnosť stavať budovy a na ich podnety sa môže tento svet zlepšovať. Tento projekt sa odlišuje od Alpha World v tom, že je novší a využíva modernejšie technológie, čo môžeme pozorovať pri rôznych animáciách, ktoré sú spravené bez rušivých sekaní a spomalení. Inak sa tieto dva svety veľmi podobajú aj tým, že každý má svoj vlastný prehliadač, ktorý zobrazuje daný svet.

Ďalšou zaujímavosťou sú rôzne veľké firmy ako IBM, ktoré sa zapojili do projektu vytvorením avatarov, keďže tento svet umožňuje virtuálnu propagáciu firiem, ako aj vytvorenie virtuálnych sídiel. Nachádzame tu taktiež prvky z reálneho sveta, ako sú kópie veľvyslanectiev, prípadne Brazílske TAM airlines, ktoré ako prvé ponúkajú

virtuálne lety [tam07]. Silnou stránkou Second Life je veľká popularita a počet obyvateľov, ktorý presiahol už 1,3 mil.

Pokiaľ ide o oblohu aj tento virtuálny svet ponúka rôzne animácie oblakov, slnka, ktoré sa menia podľa dňa a noci. Ide o veľmi graficky a dizajnovy zaujímavé obrázky. Na oblohe nachádzame aj rôzne reklamy, ktoré plávajú nad budovami.

1.2.4 Google Earth

Google Earth [ear07] je projekt firmy Google, ktorý začal podrobnými leteckými snímkami, ktoré si mohli používatelia ľubovoľne pozrieť pomocou web stránky, alebo stiahnutím aplikácie. V ďalších verziách boli k mapám pridané aj 3D modely a informácie z Wikipédie²⁰. Používateľ sa môže interaktívne pohybovať pomocou štyroch šípok znázorňujúcich sever, juh, východ a západ ako aj voľným pohybom myši.

Nová verzia 4.2 umožňuje detailné pozorovanie oblohy pomocou skutočných fotografií vesmíru. Ide o spoluprácu Google, NASA a mnoho ďalších spoločností. Pri významných oblastiach je možnosť obohatiť si vedomosti prostredníctvom textových doplnkov. Rovnako sa ponúka možnosť rôzneho prepínania pohľadov, napríklad Hubblov teleskop [kas07]. Vlastne ide o prehliadku fotografií vesmíru s výkladom. To znamená, že nejde o trojrozmerný pohľad, ale ide o virtuálny náhľad do reálneho vesmíru.

Podobným projektom je aj Virtuálne observatórium, ktoré umožňuje podrobné zobrazenie astronomických entít, ktoré sa budú využívať najmä pre študentov astronómie a príbuzných odborov [sko08].

1.2.5 Virtual Reality Search Machine [han99]

Virtual Reality Search Machine je aplikácia, ktorá prehľadáva databázu a data v 3D formulároch. Používa VRML a JAVA prostredia ako aj databázy. Riešenie prepojenia medzi jednotlivými prostrediami je pre nás zaujímavé, pretože môže viesť aj k riešeniu nášho problému. Výhodou tohto prístupu je nezávislosť na prehliadači, databáze, ako aj nízke hardvérové nároky.

²⁰ www.wikipedia.org – najväčšia online encyklopédia fungujúca na princípe kolaboračného prispievania

2 Špecifikácia a návrh riešenia

2.1 Cieľ práce

Cieľom práce je vytvoriť softvérové dielo, ktoré umožní používateľom pomocou internetového prehliadača pozorovať chronologické entity na oblohe vo virtuálnom prostredí.

2.2 Požiadavky

Pod požiadavkami na softvérové dielo rozumiem všetku funkčnosť, ktorú môžu budúci používatelia očakávať, resp. ktorú bude toto dielo ponúkať. Požiadavky tohto typu som zaradila do kategórie konkrétne požiadavky, keďže sú závislé od konkrétneho diela. Okrem požiadaviek, ktoré kladú budúci používatelia treba vziať do úvahy aj požiadavky z hľadiska vytvorenia aplikácie, ako čas, či cenu, ako aj požiadavky na výkonnosť hardvéru²¹.

V prvej časti opíšem architektúru klient-server z môjho pohľadu vzťahujúcemu sa k tejto aplikácii, ako základný princíp, ktorý sa v práci používa. Následne vysvetlím požiadavky na dielo a to všeobecné, ktoré sa vzťahujú na všetky softvérové diela, konkrétne, platformy, technológie ako aj návrh riešenia.

2.2.1 Klient – Server technológia

Základom tejto technológie je vytvorenie dvoch častí, z ktorých každá tvorí celok, ale zároveň spolu navzájom komunikujú a vymieňajú si informácie. Používa sa aj v prostredí WWW, kde server poskytuje súbory s obsahom stránok a prípadne databázu a klient je vlastne prehliadač, ktorý dokáže tieto informácie správne zobrazit'. Tento prístup využijem na zobrazovanie statickej časti stránky, ktorá bude obsahovať HTML.

- ❖ **Server** - Za časť server pokladám priestor, kde budú uložené súbory, ako aj celá databáza s entitami, ktoré budú zobrazené na oblohu. Z hľadiska funkčnosti by strana serveru mala poskytovať najmä výpis relevantných entít ako prvkov z virtuálneho sveta. To, ktoré entity sú relevantné a teda ktoré má z databázy strana

²¹ hardvér – spoločný názov pre pevné časti elektrických zariadení

servera vybrať, bude závisieť na vstupe poskytovanom klientom. Všetky funkcie, ktoré táto strana bude poskytovať, budú spracovávané pomocou technológie PHP [PHP].

- ❖ **Klient** – Za časť klient z hľadiska zobrazenia virtuálneho sveta považujem prehliadač, ktorý umožňuje zobrazovať tento virtuálny svet v podobe zásuvného modulu pre internetový prehliadač. Ďalej sú to funkcie, ktoré budú umožňovať interakciu s používateľom, ako napríklad odchyťávanie kliknutí na objekty a následná akcia. Funkčnosť poskytovanú klientom bude uložená v súboroch so skriptami, ktoré sa vykonávajú na strane servera v mojom prípade ide o spracovanie pomocou JavaScriptu [ECMA].

Z toho vyplýva, že hlavnou myšlienkou je presúvanie dát cez internet, čo musím zohľadniť pri výbere poskytovanej funkčnosti ako aj pri zohľadnení všeobecných požiadaviek, pretože obe tieto časti musia pracovať v reálnom čase aby mohli poskytovať viditeľnú funkčnosť.

2.2.2 Všeobecné požiadavky

- ❖ **Efektívnosť** - Z hľadiska efektívnosti je zaujímavá najmä časová a pamäťová zložitosť. Obe závisia na rýchlosti pripojenia, používaného prehliadača (ako web tak aj X3D, resp. VRML prehliadača), keďže dáta budú presúvané cez internet. Tieto vlastnosti neviem priamo ovplyvniť, pretože závisia od konkrétneho používateľa a jeho hardvérového a softvérového vybavenia, ktorému však viem odporučiť optimálne nastavenie.

Vlastnosti, ktoré viem ovplyvniť sú Level of Detail²², ktorého správne nastavenia dokážu zrýchliť zobrazovanie veľkých svetov, ako aj možnosť vybrať si množstvo zobrazenej informácie pomocou prepnutia medzi detailným a menej detailným množstvom informácií. Z hľadiska veľkosti presúvaných dát VRML priamo umožňuje ukladanie informácie v komprimovanej verzii, ale keďže ja v tejto práci vytváram svet dynamicky a nepoužívam celý vopred vytvorený, rozhodla som sa túto možnosť nepoužiť.

²² Level of Detail – technika, pri ktorej sa objektom priradí tzv. LOD vlastnosť a podľa toho sú presne zobrazované najmä tie objekty, ktoré sú bližšie

Úroveň zobrazenia každej informácie bude uložená v databáze a používateľ to nebude môcť ovplyvniť, bude si však môcť vybrať o ktorú úroveň má záujem, ako aj oblasť záujmu (kliknutím na objekt v scéne, ktorý umožňuje zobrazovanie entít na oblohe).

- ❖ **Ekonomickosť** - Jedným z cieľov práce je, aby bola aplikácia prístupná veľkému množstvu používateľov pripojených k sieti internet. Preto toto softvérové dielo bude voľne stiahnuteľné a spustiteľné po inštalácii potrebného zásuvného modulu. Toto všetko znamená, že na prezeranie naplneného obsahu nebudú potrebné žiadne dodatočné finančné prostriedky.

Z hľadiska naplňania informácií do databázy bude potrebné zabezpečiť naplnenie entít pomocou web stránky, ako aj server, ktorý umožňuje prehliadanie štandardných web. stránok.

- ❖ **Bezpečnosť** - Celý prístup je založený na internetovom pripojení, preto musím počítať s možnými hrozbami počas prenosu informácie. Ide najmä o nesprávne zadané, resp. ohrozujúce informácie. Najohrozenejšiu časťou je server a v ňom údaje v databáze. Do databázy sa však informácie pri spojení nevkladajú, iba prenášajú, takže bez bližšieho poznatku nie je možné do nej vložiť nekorektné dáta. Napriek tomu, budú všetky vstupy používateľa ošetrené po prenose. Rovnako moje softvérové dielo neobsahuje časti, do ktorých by sa dalo priamo písať a ktoré by boli priamo posielané na server, interakcia je zabezpečená prostredníctvom objektov citlivých na myš (kliknutie, posúvanie myšou nad objektom...) ako sú tlačítka, alebo objekty v 3D scéne, preto si myslím, že takáto hrozba podľa môjho názoru nie je relevantná. Pre poskytovanie lepšej bezpečnosti budem kontrolovať všetky vstupy a výstupy generované pomocou príkazov `HttpRequest` ako aj výbery z databázy.

Lepšia bezpečnosť sa dá zaručiť odchyťávaním chybových hlásení a tým poskytnúť menšiu šancu na poškodenie, či modifikáciu posielaných dát. Zároveň potrebné prehliadače ani môj softvér nijako nepoškodzujú stranu klienta a ani nezasahujú do dát, ktoré má používateľ uložené.

- ❖ **Presnosť** - Presnosť zobrazovania dát závisí najmä od pluginu, ktorý prehliada virtuálny svet, pretože tieto sa vo svojej ponúkanej funkčnosti môžu líšiť, ako aj od

použitej platformy, keďže táto kombinácia prehliadača a platformy nemusí byť vhodná a spolupracujúca. Tento problém sa snažím riešiť najmä poskytnutím informácie o najvhodnejšom výbere ako aj otestovaním pre rôzne platformy, prehliadače a zásuvné moduly. Presnosť z hľadiska funkčnosti riešim pomocou spolupráce s nezávislou vzorkou ľudí.

2.2.3 Konkrétne požiadavky

Ide o požiadavky, ktoré sa vzťahujú na konkrétne softvérové dielo a sú kladené rôznymi cieľovými skupinami na funkčnosť, ako aj požiadavky, ktoré chcem ja ako autor poskytovať. Z môjho pohľadu sem patria:

- ❖ **Cieľové skupiny** - Rôzne cieľové skupiny môžu mať rôzne požiadavky, preto som aj ja rozdelila svojich cieľových používateľov na skupiny a popísala ich predpokladané očakávania. Keďže chronologické údaje sú väčšinou historické entity, aj tieto skupiny som tomuto poznatku prispôbila.
 - žiaci, študenti – ide o skupinu, ktorá sa chce jednoduchou, intuitívnou cestou naučiť čo najviac údajom, ktoré ich zaujímajú, resp. ktoré potrebujú vedieť kvôli naučeniu učebnej látky. Títo budú mať možnosť vybrať si zobrazovanie všetkých poskytovaných údajov relevantných k danému miestu.
 - historickí nadšenci – ide o skupinu, ktorá sa zaujíma o rôzne entity, pretože ich to, na rozdiel od 1. skupiny zaujíma, chcú mať entity prehľadne usporiadané a chcú ich mať čo najviac. Pre zohľadnenie požiadaviek tejto skupiny budú rôzne kategórie entít zobrazené rôznou farbou, v rámci kategórie rovnakou farbou. Taktiež si budú môcť vybrať zobrazenie všetkých entít.
 - bežní okoloidúci – zaujímajú sa najmä o bežný prehľad miesta, v ktorom sa nachádzajú, nepotrebujú všetky poskytované informácie. Pre tento typ používateľov bude pripravená možnosť voľby menšieho množstva ponúkanej informácie, čiže iba najvýznamnejšie entity.
 - ostatní – používatelia, ktorí sa náhodou objavili vo virtuálnom svete, a konkrétne entity ich vôbec nezaujímajú. Používateľom tohto typu som poskytla možnosť takú,

že nemusia vôbec kliknúť na definovaný objekt a teda nebude im zbytočne zobrazovaný priestor oblohy v podobe entít.

To, ktoré miesto je zaujímavé, si budú môcť používatelia vybrať pomocou kliknutia na preddefinovaný objektu v scéne, ktorý bude jasne naznačovať, že sa naň dá kliknúť. Rovnako si budú môcť vybrať, ktorá kategória je pre nich zaujímavá.

- ❖ **Prehľadnosť** - Na zabezpečenie prehľadnosti pre používateľa použijem logické rozmiestnenie na obrazovke a zároveň všetko ovládanie bude intuitívne a vhodne popísané. To znamená, že použijem štandardné tlačidlá, s ktorými sa môžeme stretnúť na internetových stránkach. Na tieto tlačidlá bude možné kliknúť a spustiť akciu.

Z hľadiska prehľadnosti zdrojového kódu, jednotlivé funkcie budú delené do súborov podľa toho, s ktorými dátami pracujú. Pôjde najmä o JavaScriptové [ECMA] funkcie a súbory, pretože tieto musia byť schopné pracovať s 3D objektmi, serverom pomocou technológie Ajax [Ajax], ako aj zabezpečiť prepojenie týchto technológií. Zároveň lepšiu prehľadnosť funkcií zabezpečím logickým pomenovaním aj pridaním komentáru do zdrojového kódu.

- ❖ **Interakcia** - Moderná tvorba internetových aplikácií vyžaduje, aby umožňovali interakciu s používateľom. To znamená, že potom, čo používateľ spraví nejakú aktivitu (väčšinou klikne, či pohne myšou, alebo napíše niečo na klávesnici), tak softvér automaticky vykoná nejakú akciu.

Interakciu v mojej aplikácii zabezpečujú najmä na dotyk citlivé 3D objekty v podobe žltej gule, ktorým som dala názov **SkyPort**. Tieto pri pohybe myšou nad nimi pulzujú (zväčšujú a zmenšujú sa) a tým vyjadrujú citlivosť na kliknutie. Po kliknutí na ne sa automaticky otočí pohľad smerom na oblohu a zobrazí sa text na nej. Ďalším interaktívnym prvkom budú ovládacie prvky umiestnené mimo priestoru 3D sveta, aby som nenarušila tento pohľad. Jedným z nich je tlačidlo „**Back**“, ktoré umožňuje vrátenie pohľadu späť v prípade, že sa pozeráme na oblohu. Na samotné prehliadanie entít slúžia škálovacie úsečky v podobe šípok v šiestich smeroch (v každom smere súradnicových osí sú 2), ktoré posúvajú oblohu, podľa záujmu používateľa. Niektoré entity budú mať možnosť podrobnejšej

informácie, ktorá sa im zobrazí v špeciálnej časti pod oknom po kliknutí na daný popis entity.

- ❖ **Jazyk** - Cieľom mojej aplikácie je aby bola navštevovaná a používaná pre rôzne skupiny ľudí, aj medzinárodné, keďže cieľom je zvyšovať kultúrne povedomie, preto som sa rozhodla všetky texty a nápisy písať v anglickom jazyku. Pôjde však o jednoduché hlásenia chýb ako aj o odporúčania a nebude problém ich lokalizovať do iného jazyka. Samotné entity budú skôr názvy a roky, ktoré reprezentujú čísla, teda nebude ich treba ani prekladať. Čo sa týka zdrojového kódu a popisu funkcií v ňom, aj tieto budú písané v anglickom jazyku, aby tu vznikla možnosť na neskoršie rozširovanie.

2.3 Platforma

Časť programu, ktorá je platformovo závislá je samotný 3D prehliadač, resp. zásuvný modul, ktorú však nájdeme pre rôzne platformy ako je Microsoft Windows, Linux, Unix, MacOS. Z hľadiska veľkého nároku na pracovnú pamäť tento softvér nie je určený pre mobilné telefóny a operačné systémy v nich. Vo všeobecnosti je moja aplikácia prispôbená a testovaná pre operačný systém Windows Vista, prehliadač Firefox 2.0 a plugin Flux Player určený a podporovaný najmä operačným systémom Windows. Možno však použiť aj iné prehliadače a operačné systémy, ktoré podporujú prehliadanie súborov typu VRML, X3D. Pretože chcem zachovať multiplatformovosť²³ z hľadiska rôznych operačných systémov určených pre PC, aj nižšie popísaná technológia bude na to prispôbená.

2.4 Technológia

2.4.1 Spracovanie

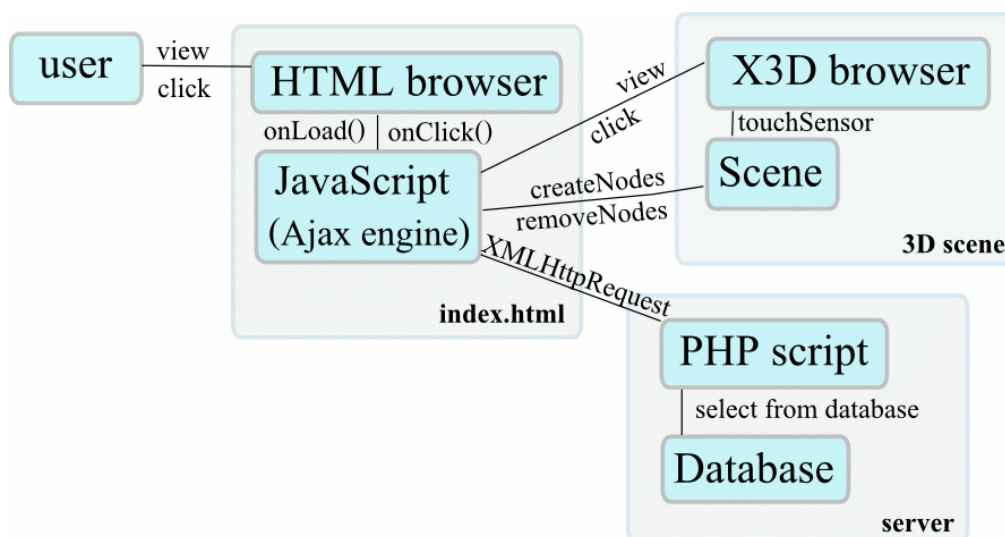
V spracovaní 3D dát na internete a špeciálne pre VRML dáta sú najpoužívanější 2 prístupy kombinácie s programovacím jazykom:

- ❖ Java [JAVA] - Čo to je programovací jazyk Java je uvedené v prvej kapitole. Java umožňuje všetky potrebné funkcie pre moju aplikáciu dokonca podporuje aj

²³ multiplatformosť – použiteľnosť na rôznych platformách

spoluprácu s databázou. Ja som sa tento prístup rozhodla nepoužiť, pretože vyžaduje väčšie systémové nároky na spustenie samotného appletu a prepojenie na databázu je intuitívnejšie v prístupe s využitím technológie Ajaxu.

- ❖ JavaScript [ECMA] - Tento skript je široko používaný v internetových aplikáciách, zabezpečuje interakciu a pomocou Ajaxu dokáže komunikovať s PHP, ktoré má intuitívnu prácu s databázou. Priama podpora VRML pomocou AJAX3D [AJAX3D] ma presvedčila k použitiu práve tohto prístupu. Umožňuje pridanie senzoru (tieto sú používané vo VRML, X3D [X3D] na zachytenie udalosti vyvolanej používateľom), prácu s pohľadmi, ako aj nahrávanie údajom z databázy. Uvediem schému vzájomného prepojenia technológií a prehliadačov. Čiary znázorňujú vzájomnú komunikáciu.



Obr.4 Schéma spolupráce jednotlivých technológií

Jediná časť, ktorú si používateľ otvorí pomocou internetového prehliadača (v mojom prípade je to FireFox), je vlastne HTML stránka. Táto stránka musí obsahovať tag

```
<embed TYPE="model/x3d">
```

Tento tag²⁴ určuje aký typ informácie obsahuje (v našom prípade ide o X3D formát). Pomocou funkcií:

- `browser.createX3DFromString()`
- `browser.loadURLS ()`

²⁴ tag - je kód popisujúci inštrukciu, ktorú internetový prehliadač interpretuje a zobrazí

- `context.getRootNodes()`
- `context.removeRootNode()`
- `context.addRootNode()`
- `addFieldEventListener()`

sa dá pristupovať k rôznym objektom na 3D scéne, meniť ich, vytvárať aj rušiť. Podobne sa dajú priradiť rôzne spracovanie udalostí vzhľadom na objekty. Taktiež sa dá natiahnuť na nový virtuálny svet (v mojom prípade to bude zaujímavé miesto, na ktoré chcem mapovať oblohu). Pri každom natiahnutí sa predošlý svet vymaže, to znamená, že takto sa dá natiahnuť práve jeden svet aj s jeho uzlami sa dá dynamicky pracovať.

2.4.2 Dáta, formáty typy súborov

Z hľadiska typov súborov, do ktorých sa dá zapísať VRML a X3D poznáme *.wrl*, *.x3d*, *.x3dv*. Pre formát VRML je určený *.wrl*, pre formát X3D je určený *.x3d*, ale tento formát sa dá zapísať aj tzv. klasickým spôsobom, čo je vlastne syntax VRML a súbor má koncovku *.x3dv*.

V mojej práci budem používať technológiu Ajax3D, ktorá je síce primárne určená pre formát X3D, avšak dokáže dynamicky pridávať a rušiť uzly písané syntaxou VRML, keďže X3D vie priamo pracovať s touto syntaxou. Vychádzajúc z existujúcich virtuálnych svetov, ktoré by som chcela do svojej práce vložiť a ktoré sú uložené v súboroch *.wrl*, rozhodla som sa pre tento typ súboru, ale na porovnanie rozdielov prikladám ten istý objekt s rovnakou geometriou a farbou v každom z nich, ktoré sú priložené na záver práce.

Ak v súbore *.wrl* vymeníme riadok

```
#VRML V2.0 utf8 za riadok #X3D V3.0 utf8
```

a dostaneme súbor, ktorý sa správne označuje koncovkou *.x3dv* a je to súbor formátu X3D zapísaný tzv. starou syntaxou, teda pomocou VRML formátu, bude však čitateľný akýmkoľvek prehliadačom X3D.

2.4.3 Použité technológie softvér

Z hľadiska použitia technológií som sa rozhodla pre vyššie spomínaný prístup Ajax3D so skriptovacím jazykom PHP na serveri Apache a databázu MySQL, keďže toto

spojenie je štandardné, rozšírené a dostupné pri tvorbe internetových stránok. Ďalšou výhodou tohto spojenia je bežná dostupnosť prostredníctvom bežných webhostingov.²⁵ Jednou z najlepších vlastností MySQL je robustný výkon a jednoduchosť v spravovaní a prepojení na iné systémy. Služba Flickr [fli08] napríklad využíva tento databázový systém na správu miliárd fotiek svojich používateľov.

2.4.4 Použitý softvér

Z hľadiska použitia vhodného softvéru na tvorbu tejto internetovej aplikácie som sa rozhodla použiť voľne dostupný editor PSPad [pspad], ktorý ponúka dobrú prácu s použitými súbormi, jednoduché používateľské rozhranie a zvýraznenie syntaxe pre jazyk PHP aj JavaScript, čo mi umožňuje ľahšiu prácu. Na rozdiel od HTMLKit (ďalší softvér používaný na rovnaký účel) ponúka priamy náhľad statických stránok a ľahké vyhľadávanie v nedávno prezeraných súboroch.

Pre spracovanie VRML, X3D kódu som si vybrala Flux Studio [studio], ktoré je voľne stiahnuteľné a vytvorené tvorcami Ajax3D, umožňuje jednoduché modelovanie, interakciu a animáciu vo VRML, X3D, export a import týchto typov súborov.

2.5 Používateľské rozhranie (user interface)

Táto časť je podľa mňa veľmi dôležitá, pretože dokáže používateľovi pomôcť v lepšej orientácii, alebo ho naopak môže dezorientovať. Keďže cieľová skupina nie je špecificky definovaná so špeciálnymi požiadavkami na toto rozhranie, tak sa ovládanie nachádza v štandardnej dolnej časti pod oknom, ktoré zobrazuje virtuálny svet. Nachádza sa tam tlačidlo s nápisom „BACK“, ktoré po kliknutí vráti pohľad z oblohy späť. Ďalej sa v tejto dolnej časti nachádzajú vyššie spomínané škálovacie úsečky, pomocou ktorých si používateľ môže vybrať, o ktoré roky má záujem. V hornej časti nad oknom s virtuálnym svetom sa bude nachádzať Status riadok, do ktorého vkladám pomocou JavaScriptu špecifický text vzťahujúci sa ku konkrétnej situácii, ktorý by mal pomôcť používateľovi v lepšej orientácii. Ide najmä o chybové hlásenia a stav (pohľad hore, ...).

²⁵ webhosting - servis, ktorý umožňuje sprístupniť stránky do siete internetu

3D scéna bude obsahovať objekty citlivé na dotyk používateľa (SkyPorty), budú to žlté gule, ktoré sa budú zväčšovať a zmenšovať, keď sa nad nimi bude nachádzať myš používateľa. Tieto gule sa budú nachádzať v polovici výšky avatara, čiže približne vo výške vypínačov na svetlo v reálnom svete, čo bude evokovať, že sa na nich dá kliknúť. Z hľadiska rozlohy tieto objekty sa budú nachádzať pri zaujímavých objektoch (väčšinou historických pamiatkach), aby informovali používateľa v prípade záujmu o entity.

Poslednou časťou používateľského rozhrania je zobrazenie podrobnejších informácií o vybranej entite. Toto okno bude pod oknom s 3D svetom.



Obr. 5: Náhľad používateľského rozhrania

2.6 Predpríprava

Pod pojmom predpríprava rozumiem všetky operácie (funkcie), ktoré treba vykonať pred spustením samotnej aplikácie na strane klienta. Ide najmä o naplnenie entít do databázy, vytvorenie virtuálneho sveta v podobe objektov, teda 3D scény, vloženie SkyPortov doplnením do kódu.

- ❖ **Entity v databáze** - pri každej entite je potrebné zadať textový reťazec (krátky popis, prípadne názov), začiatok udalosti, koniec udalosti, úroveň podrobnosti, kategóriu, ako aj súradnice, pre ktoré je objekt relevantný. Pod súradnicami sa myslí pravý dolný a ľavý horný roh obdĺžnika, v ktorom je objekt zobrazovaný.

- ❖ **Databáza pohľadov** – v tejto databáze sa nachádzajú súradnice a orientácia jednotlivých pohľadov, do ktorých sa používateľ vráti po pohľade na oblohu a ktoré dynamicky vytváram pri vytváraní oblohy. Tieto údaje je potrebné zadať aby som mohla používateľovi poskytnúť prirodzený návrat pohľadu späť.
- ❖ **3D scéna** – vytváranie objektov tejto scény nie je súčasťou mojej práce, na jej realizáciu však bude potrebné mať tento súbor. Súčasťou mojej práce bude pomocou PHP skriptu k už existujúcim objektom pridať SkyPorty do jedného súboru *.wrl*.

2.7 Návrh riešenia

Táto časť obsahuje načrtnuté riešenie problému podľa požiadaviek uvedených v prvej časti. Aplikáciu som nazvala **SkyBoard**.

Pri prvom natiiahnutí stránky sa pomocou JavaScriptu a funkcie `browser.loadURLS()` zobrazí v okne 3D prehrávača (toto okno bude menšie ako štandardná web stránka) virtuálny svet s objektmi budov, ako aj SkyPortmi. Pod, vedľa a nad oknom sa zobrazia prvky používateľského rozhrania ako je popísané v časti „2.5 Používateľské rozhranie“. Po kliknutí na ktorýkoľvek SkyPort sa spustí funkcia v JavaScripte, ktorá bude mať za úlohu spracovať túto akciu pomocou technológie Ajaxu. Zavolá PHP skript na strane servera s údajmi o polohe (ktoré zistím pomocou polohy zobrazovača). Tento skript pošle naspäť na stranu klienta časť VRML kódu (resp. starého štýlu zápisu X3D), ktorá bude obsahovať text ako aj 3D objekty v podobe valcov, ktoré budú mať za úlohu znázorňovať vzájomnú súvislosť, chronologizáciu entít.

Tieto objekty budú v jednom zoskupené do jedného objektu (Group) a zobrazia sa pomocou `browser.createX3DFromString()`. Celý tento objekt bude možné posúvať pomocou úsečiek pri 3D okne. Používateľ si bude môcť posúvaním vybrať časť entít, o ktoré má záujem. Taktiež sa pri zobrazení natočí pohľad smerom k oblohe a vytvorí sa znovu pomocou Ajax technológie nový pohľad, na ktorý sa bude možné vrátiť pomocou kliknutia na tlačidlo „BACK“. Pri vrátení pohľadu sa zruší celý objekt, obsahujúci entity na oblohe, aby potom bolo možné znovu vytvorenie ďalších častí oblohy.

3 Implementácia

V poslednej kapitole sa budem venovať samotnej implementácii internetovej aplikácie **SkyBoard**, ktorá vychádza z návrhu v predchádzajúcej kapitole. V jednotlivých podkapitolách popíšem rôzne funkcie podľa toho, v ktorom type súborov sú zapísané a ako za sebou nasledujú. Rozhodla som sa pre programovanie pomocou funkcií namiesto objektovo orientovaného programovania, pretože toto je viac používané pri využití JavaScriptu. Tieto funkcie sú potrebné na poskytnutie požadovanej funkčnosti.

Súbory z hľadiska obsahu, ktoré sa nachádzajú v tejto práci sú:

- **.html** (HTML)
Tento súbor používam na vytvorenie statickej stránky, ktorá rozdelí používateľské okno, vloží objekty do stránky a tieto budem môcť dynamicky meniť.
- **.css** (CSS)
Tento súbor, ktorý je jediný tohto typu, používam na vytvorenie kaskádových štýlov pre .html stránku.
- **.js** (JavaScript)
Tento typ súborov používam na vytvorenie engine pre prácu pomocou Ajaxu v podobe niekoľkých súborov, ktoré obsahujú funkcie potrebné na interakciu a dynamickú prácu so scénou, tieto priamo komunikujú s prehliadačom používateľa.
- **.php** (PHP)
Súbory typu PHP používam na posielanie častí VRML kódu Ajaxovému enginu podľa výstupov z databázy. Tento súbor budem používať aby som mohla zobrazit' oblohu. V prístupe, ktorý používa Ajax, PHP súbory nekomunikujú priamo s prehliadačom používateľa, ale s Ajax engine, čo v mojej aplikácii predstavuje súbor JavaScriptových funkcií.
- **.sql** (MySQL)
Tento súbor obsahuje MySQL príkazy volané pomocou jazyku PHP. Sú to funkcie connect() a disconnect(), ktoré umožňujú pripojenie na databázu.

- **.wrl (VRML)**

Tento súbor obsahuje celý virtuálny svet vrátane objektov, materiálov, senzorov, pohľadov, ako aj SkyPortov.

- **.jpg (JPEG)**

Obrázky typu JPEG používam dvoch typov a to textúry budov, resp. iných objektov vo virtuálnom svete, ktoré budú v samostatnom priečinku, aby boli oddelené od ostatných súborov. Možnosť navigácie entít na oblohe vkladám do stránky tiež pomocou obrázka, ktorý obsahuje šesť šípok.

3.1 Práca so scénou prostredníctvom Ajax engine

Na vytvorenie dynamickej a interaktívnej scény je potrebná komunikácia medzi Ajax engine a zásuvným modulom. Pre účely mojej práce používam Flux Player. Táto komunikácia je zabezpečená najmä funkciou `getContext()`. Pomocou tejto funkcie môžem meniť existujúce objekty, pretože výstupom z tejto funkcie je hierarchická stromová štruktúra scény, kde objekty sú zoskupované do skupín a uzlov. K týmto uzlom viem pristupovať pomocou ich poradia v scéne, a to funkciou `getRootNodes()`. Keď mám v premennej daný uzol tento viem meniť tak, že zmením jeho parametre podobne, ako by som ich menila pomocou VRML syntaxe. Jediný rozdiel je prístup k nim, na ktorý používam bodku vloženú medzi názov uzlu a parameter.

Odchytávanie akcií na objekty viem spracovať pomocou pridania pozorovateľov na objektu, ktoré obsahujú senzory. Následná akcia je potom spracovaná príslušnou funkciou.

Pre pridávanie objektov do scény používam metódu `createX3DFromString()`, ktorej na vstup viem poslať akúkoľvek syntakticky správnu premennú typu `String`. Syntakticky správna znamená ukončený, ucelený VRML kód, ktorý je samostatne spustiteľný.

Pomocou týchto niekoľko funkcií viem spraviť dynamickú a interaktívnu scénu, keďže podpora JavaScriptových funkcií a DOM je dostatočná, je možné kombinovať objekty v

HTML stránke a udalosti v nich a scénu, prípadne informácie, ponúkajúce sa zásuvným modulom.

3.2 Štruktúra databázy

3.2.1 Štruktúra databázy entít

Pre každú entitu je nutné zistiť a pamätať si tieto údaje:

- **ID** - Poradové identifikačné číslo entity, na jednoduchšiu identifikáciu pri hľadaní podrobnejších informácií.
- **názov** - Teda krátka textová informácia, ktorá bude zobrazená na oblohu.
- **kategória, detail** - Aby som vedela, či daná entita má byť zobrazovaná pre konkrétnu vybranú kategóriu, resp. detail.
- **dátum začiatku, konca** - Tieto budú zobrazované na začiatku a konci jednotlivých línií, zároveň sa podľa nich vypočíta dĺžka jednotlivých línií ako aj ich posunutie, aby podávali požadovanú vizualizáciu chronologizácie, pomocou priradenia dĺžkovej konštanty na roku a ostatné údaje sa pomerne vypočítajú. Pri niektorej nezadanej informácii na začiatku sa zoberie na vypočítanie dĺžky číslo 0, pri nezadaných informáciách na konci sa pre dni počíta s číslom 31, pre mesiace číslo 12 a pre rok sa počíta s rokom začiatku. To znamená, že pokiaľ máme jednoduchú udalosť, je potrebné zadať aj dátum začiatku aj dátum konca, aj keď budú rovnaké.
- **obdĺžnik pôdorysu** - Keď nastane kliknutie na SkyPort v tomto obdĺžniku, tak sa má táto entita zobrazovať. Obdĺžnik je zadaný pravým horným rohom a ľavým dolným rohom aktívneho pôdorysu.
- **podrobnejšie informácie** - Pri zobrazení podrobnejšej informácie zobrazím túto textovú informáciu.

3.2.2 Štruktúra databázy SkyPortov

Pre každý SkyPort je nutné si pamätať informácie o pohľade, do ktorého sa má používateľ vrátiť. Informácie, ktoré si potrebujem pamätať pre pohľad "Back" sú tieto:

- **ID** – Na identifikáciu daného pohľadu.

- **súradnice SkyPort** – Súradnice prislúchajúceho SkyPortu, aby sme vedeli pridať prislúchajúci pohľad Späť po kliknutí na SkyPort.
- **súradnice** - Kam má byť konkrétny pohľad zobraziť, resp. posunúť.
- **orientácia** - Kam má byť konkrétny pohľad orientovaný.

3.3 Obsah jednotlivých súborov

V tejto kapitole popíšem obsah jednotlivých súborov z hľadiska funkcií, ktoré obsahujú. Budem sa venovať hlavne popisu najdôležitejších funkcií.

3.3.1 index.html

Tento súbor obsahuje hlavičku a päť štandardných HTML stránok, sú to tagy `<html>` a `<head>`, `<title>SkyBoard v. 0.1</title>` - obsahuje názov internetovej stránky, `<body onLoad="onLoad()">` - parametrom tohto tagu je JavaScriptová funkcia `onLoad()`. Do hornej časti som umiestnila popis, čo treba mať nainštalované pre korektné zobrazenie obsahu stránky. Pod týmto textom, v oblasti **StatusArea** zobrazujem textovú informáciu pre ľahšiu orientáciu používateľa.

Virtuálny svet, ako aj ovládacie prvky sa nachádzajú v tabuľke, aby sa dali správne zobraziť. Po pravej strane sú ovládacie prvky a oblasť, kde zobrazujem bližšie informácie o entitách. Okno zásuvného modulu vkladám týmto riadkom, ktorý oznamuje X3D obsah:

```
<embed          WIDTH="600"          HEIGHT="400"          NAME="FLUX"
TYPE="model/x3d" DASHBOARD="0">
```

3.3.2 sensors.js

Interakcia ponúkaná mojou aplikáciou je vykonávaná pomocou funkcií uložených v tomto súbore. Sú to tieto:

- `setupListeners()` - táto funkcia nastavuje pozorovateľov 3D scény na všetky objekty, ktoré majú vo VRML senzory citlivé na dotyk (`TouchSensor`). Ide najmä o SkyPorty, ktoré keďže chcem používať dynamicky prostredníctvom JavaScriptu, preto ich musím takto pridať.

- `sensorTouched(f, t)` - táto funkcia sa spúšťa po kliknutí na SkyPort. Spúšťa funkcie na vytvorenie pohľadu na oblohu, vytvorenie entít na oblohe, pridanie pozorovateľov na entity, ktoré ponúkajú bližšie informácie, odstraňuje pohľad s názvom "**Back**", ktorý je potrebný na návrat z pohľadu na oblohu (tento ak je, tak je ešte zo starého návratu, preto nie je potrebný).

VSTUP: `f` - názov objektu, ktorý spustil túto funkciu, `t` - čas spustenia

- `addObservers()` - pomocou komunikácie so serverom pridá pozorovateľov na práve zobrazené entity, ktoré ponúkajú bližšie informácie.
- `addToSky(nodeName)` - zmení polohu už existujúceho pohľadu s názvom "**To Sky**" tak, aby sme smerovali priamo na oblohu s rotáciou v smere normály na rovinu zobrazených entít.

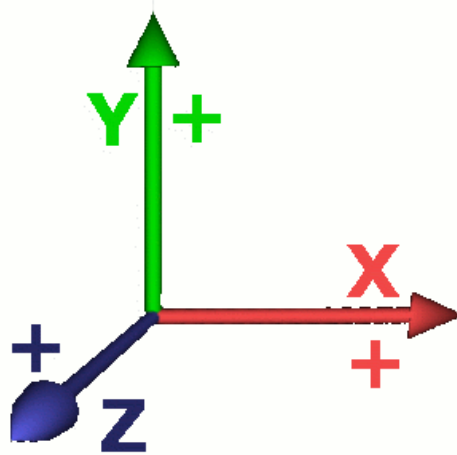
VSTUP: `nodeName` - názov objektu, ktorý spustil interakciu.

- `removeBack()` - odstráni pohľad späť z oblohy
- `addFacts(nodeName)` - pridáva celý obsah oblohy pomocou komunikácie s `php_response.php`, ako aj nový pohľad "**Back**", aby bol možný návrat späť. Informácie o týchto údajoch sú uložené v databáze, s ktorou konkrétny PHP kód pracuje.

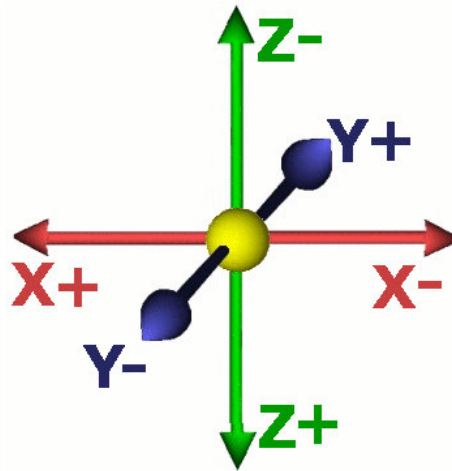
VSTUP: `nodeName` - názov objektu, ktorý spustil interakciu.

- `changeView()` - zmení pohľad používateľa na oblohu plynulou transformáciou, o ktorú sa stará zásuvný modul.
- `viewBack()` - vráti pohľad späť po kliknutí na tlačítko a zruší všetky entity z oblohy, je tu možnosť vrátenia sa z oblohy aj pomocou zmeny pohľadu prostredníctvom ponuky zásuvného modulu, ale tento postup nezruší entity z oblohy.
- `changeFacts()` - pokiaľ bolo kliknuté na zmenu oblohy, znovu sa natiahne tento obsah podľa požadovaných kategórií a detailu zobrazenia.
- `moveBox(move_dir)` - posunie oblohy oproti požadovanému smeru, pretože tento posun je prirodzenejší. Používateľ nadobúda dojem posunu v smere šípky, v

skutočnosti však posúvam oblohu v opačnom smere. Ako sa zmenia súradnicové osi oblohy vzhľadom na súradnice celého priestoru vidno na nasledujúcich obrázkoch:



Obr. 6a: Svetové súradnice



Obr. 6b: Súradnice pri posúvaní oblohy

- `factTouched(f, t)` – pre spustenie tejto funkcie je potrebná interakcia používateľa s konkrétnou zobrazenou entitou, napr. kliknutie myšou. Táto funkcia zavolaním ďalších zabezpečí zobrazenie podrobnejšej informácie o danej entite.

VSTUP: `f` - názov objektu, ktorý spustil túto funkciu, `t` - čas spustenia

- `displayInfo(ID)` – v časti HTML stránky zobrazím podrobnejšie informácie o konkrétnej entite prostredníctvom vyžiadania informácií zo servera.

VSTUP: `f` - číslo entity, ktorej bližšie informácie chcem zobrazit'

3.3.3 ajax3d.js

Tento súbor obsahuje funkcie, ktoré prebiehajú pri inicializácii stránky a rôzne pomocné funkcie. Sú to tieto:

- `onLoad()` - táto funkcia je spustená pri načítaní stránky a spúšťa funkcie na zobrazenie virtuálneho sveta a pridanie senzorov
- `createX3DFromString(str)` - pridá celý virtuálny svet zo súboru do virtuálnej scény zásuvného modulu

3.3.4 XMLHttpRequest.js

Do tohto súboru som umiestnila najmä funkcie, ktoré zabezpečujú komunikáciu s `php_response.php`. Sú to tieto:

- `createXMLHttpRequest()` - skontroluje, či vytváranie `HttpRequest` je podporované prehliadačom, ak nie, vypíše hlásenie, že túto funkciu prehliadač nepodporuje. Spúšťa sa pri každom spustení funkcie `sendRequest()`.
- `sendRequest(url)` - pošle požiadavku na stranu servera s informáciami pomocou GET, ide hlavne o x, z súradnice senzora aby sme vedeli, kam pridať pohľad a entity. V url posielam aj kategórie a detail, o ktorý má používateľ záujem (teda je zaškrtnutý).

3.3.5 php_response.php

Štruktúra môjho .php súboru je iná ako štruktúra súboru s koncovkou .js, pretože k funkčnosti ponúkanej jazykom PHP nepristupujem pomocou funkcií, je k celému súboru naraz. Súbory formátu PHP sa dajú načítať na celú stránku podobne ako súbory formátu HTML. K danému súboru pristupujem pomocou URL ²⁶, a funkcie `sendRequest()` volanej z Ajaxového enginu. Do URL ukladám za adresou samotnej stránky aj parametre, podľa toho, o ktoré informácie zo servera mám záujem. K týmto informáciám pristupujem pomocou metódy **GET**, ktorú ponúka samotný jazyk PHP. Rôzne udalosti, ktoré potrebujem spracovať pomocou tohto súboru, sú:

- **vybranie entít z databázy**

Podľa polohy SkyPortu vyberiem tie entity z databázy, ktoré sú aktívne pre oblasť, v ktorej sa nachádza. Zároveň zohľadním aj požadované kategórie a detail tým, že vyberiem len tie potrebné riadky z databázy.

VÝSTUP: Názvy jednotlivých entít, ako tvar Text, ktoré už majú správne nastavený materiál a posunutie, ako aj jednotlivé dátumy a línie v podobe tvaru valca. Všetky tieto objekty sú obalené do skupiny s názvom Facts, aby sme ich mohli posúvať naraz a spĺňajú syntax jazyka VRML.

²⁶ URL – Uniform Resource Locator - slúži na nájdenie zdroja súboru

KEDY: Po kliknutí na SkyPort, pri zmene požadovaných kategórii, prípadne pri zmene detailu.

- **pridanie pohľadu „Back“**

Z databázy pohľadov vyberiem orientáciu a súradnice pohľadu pre daný SkyPort, ktorého súradnice dostanem pomocou vyššie spomínaného príkazu GET. Všetky ostatné nastavenie ako uhol záberu a povolenie pohybu po scéne dodám pomocou jazyku PHP a budú rovnaké pre všetky pohľady „Back“.

VÝSTUP: výstupom je text obsahujúci všetky potrebné nastavenia parametrov pre správne zadanie pohľadu prostredníctvom syntaxe jazyka VRML.

KEDY: Po kliknutí na SkyPort.

- **bližšie informácie o entite**

Z databázy entít vyberiem príslušnú entitu, o ktorú mám záujem pomocou ID čísla, ktoré má v databáze.

VÝSTUP: Čistá textová informácia, ktorou sú podrobnejšie informácie, ktoré dokáže JavaScript zobrazit'.

KEDY: Po interakcii s vybranou entitou ponúkajúcou túto možnosť.

Výstupom zo súboru PHP je textový reťazec, ktorý obsahuje všetky potrebné informácie a tento posielam pomocou metódy `echo ""`; späť klientovi (Ajax engine), ktorý ich následne spracuje.

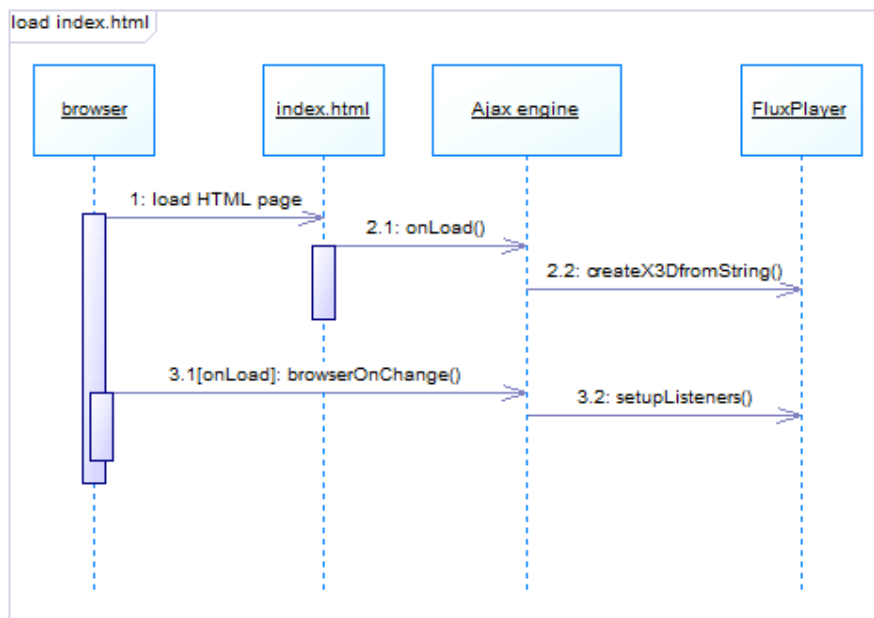
3.4 Vzájomné prepojenie jednotlivých funkcií

V tejto kapitole popíšem v akej následnosti jednotlivé funkcie prebiehajú ako aj vzájomnú komunikáciu jednotlivých častí. Tento prístup pomôže k bližšiemu pochopeniu samotnej aplikácie. Nebudem však vysvetľovať jednotlivé funkcie, pretože tie sú vysvetlené vyššie.

3.4.1 Vytvorenie stránky

Pri otvorení stránky v prehliadači na strane klienta sa táto pomocou HTML načíta, zároveň je spustená funkcia `onLoad()`, ktorá nahrá X3D scénu do prehliadača

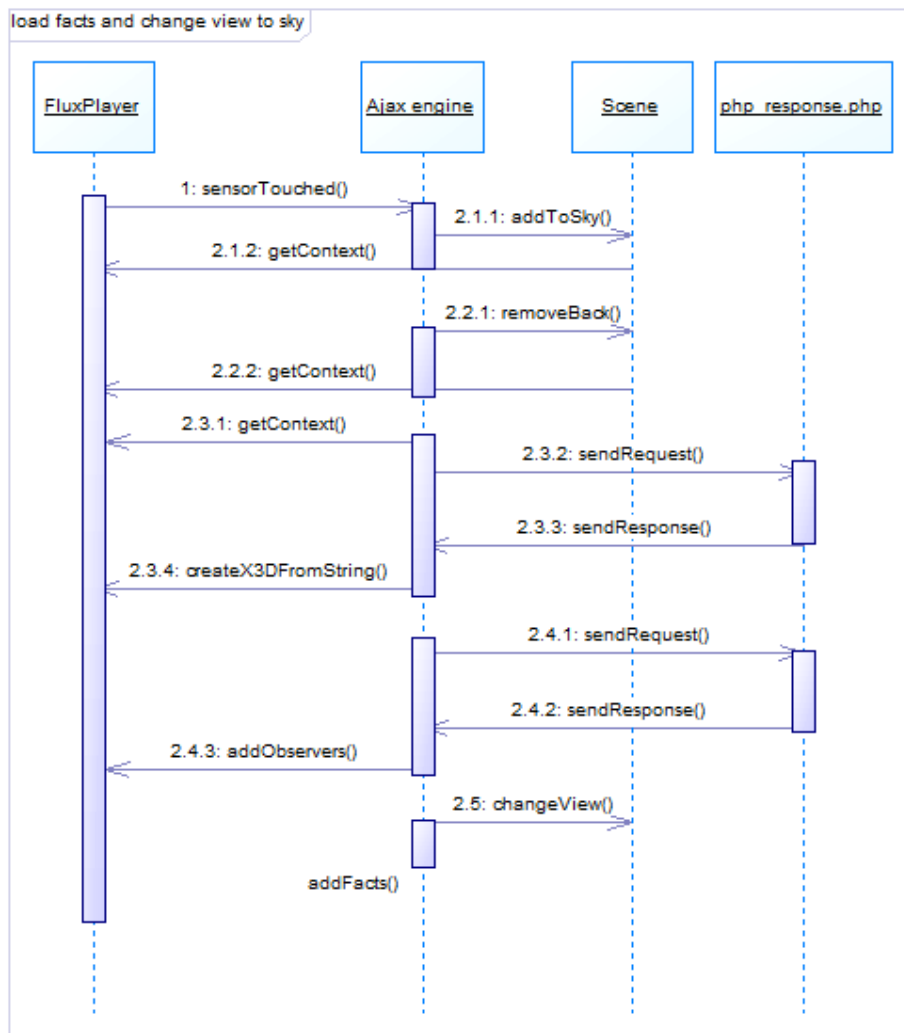
pomocou `createX3DfromString()`. Pri načítaní prehliadača je spustená aj funkcia `browserChanged()` - keďže aj načítanie stránky je zmena prehliadača a nastaví pozorovateľov na SkyPorty prostredníctvom funkcie `SetupListeners()`. Túto synchronizáciu je lepšie vidieť na obrázku na nasledujúcej strane.



Obr. 7: Následnosť funkcií pri načítaní stránky.

3.4.2 Otočenie pohľadu

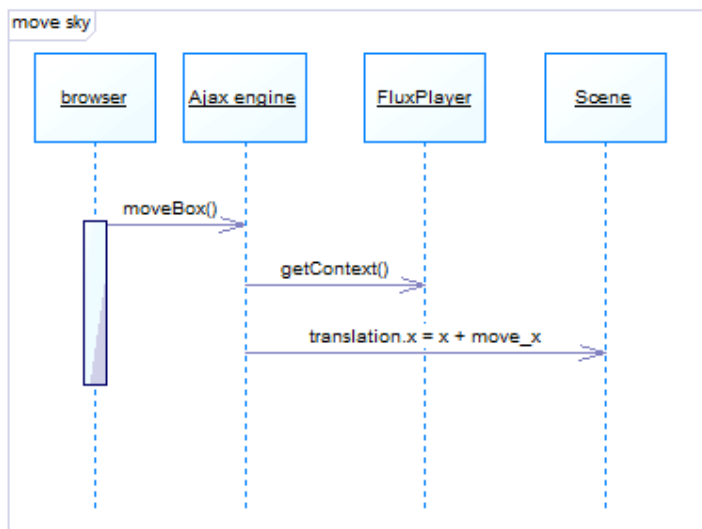
Akcia otáčania pohľadu nastáva pri kliknutí na **SkyPort**. Pri otáčaní treba okrem posunutia avatara smerom k oblohe a následnej rotácii pohľadu aj zrušiť už zobrazené entity, ak také sú, zobraziť požadované entity (zároveň ich pomocou Ajaxu musím vybrať z databázy) a pridať možnosť návratu v podobe pridania pohľadu späť. Na to, aby som mohla pridať pohľad späť musím predchádzajúci zrušiť (z predchádzajúceho pohľadu späť). Zrušenie pohľadu znamená odstránenie uzlu, ktorý ho predstavuje. Pod pohľadom späť rozumiem pohľad nastavený v blízkosti SkyPort tak, aby nebol nasmerovaný neprirodzeným pohľadom (tesne vedľa budovy prípadne priamo v stene budovy...). Súradnice ako aj nasmerovanie spätných pohľadov mám uložené v samostatnej databáze, keďže tieto súradnice nie je možné všeobecne vytvoriť. Samotná komunikácia medzi jednotlivými funkciami je zobrazená na nasledujúcom obrázku:



Obr. 8: Následnosť funkcií pri zobrazení entít a zmene pohľadu na oblohu.

3.4.3 Posunutie entít na oblohe

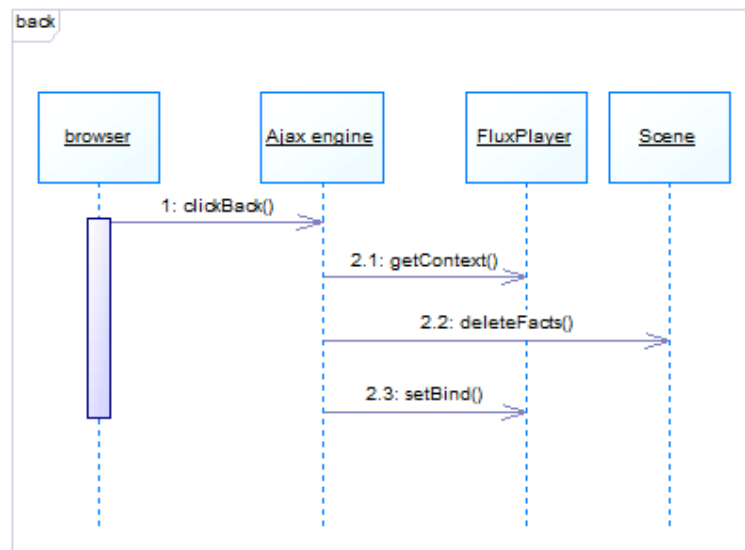
Po kliknutí na ktorúkoľvek os v obrázku, ktorý znázorňuje posunutie oblohy, vykoná môj program funkcie, ktoré zabezpečia posunutie súradníc oblohy, ktorú mám uloženú v jednej skupine, preto viem celú oblohu posúvať naraz. Samotné posunutie vykonám zmenou parametrov `translation.x`, `translation.y`, `translation.z` celej skupiny prostredníctvom práce s Ajax engine, ktorého podrobný popis je v kapitole (3.1). Týmto posunutím zároveň zmením aj lokálny súradnicový systém vzhľadom na svetový, keďže tieto navzájom súvisia a všetky objekty v skupine majú súradnice vzhľadom na objekt v hierarchii vyššie. Komunikácia jednotlivých funkcií je znázornená na nasledujúcom obrázku:



Obr. 9: Následnosť funkcií pri posúvaní oblohy.

3.4.4 Vrátenie pohľadu

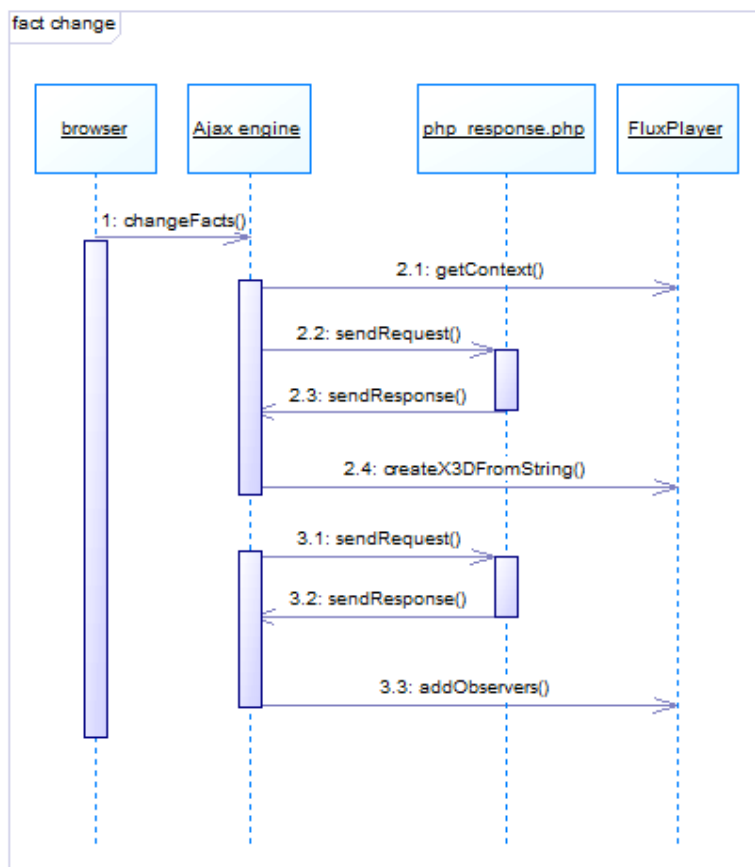
Používateľ vo virtuálnom prostredí pri pohľade na oblohu niekedy má potrebu sa z tohto pohľadu aj vrátiť, preto mu túto možnosť ponúkam aj tým, že klikne na tlačidlo, ktoré ho vráti späť, tento postup zároveň zruší zobrazenie entít na oblohe. Samozrejme môže využiť aj možnosť na teleportovanie, teda prepnutia medzi pohľadmi, ale tento postup nenastaví správne všetky údaje. Následnosť použitých funkcií zobrazuje nasledujúci obrázok:



Obr. 10: Následnosť funkcií pri vrátení pohľadu späť z oblohy.

3.4.5 Zmena výberu entít

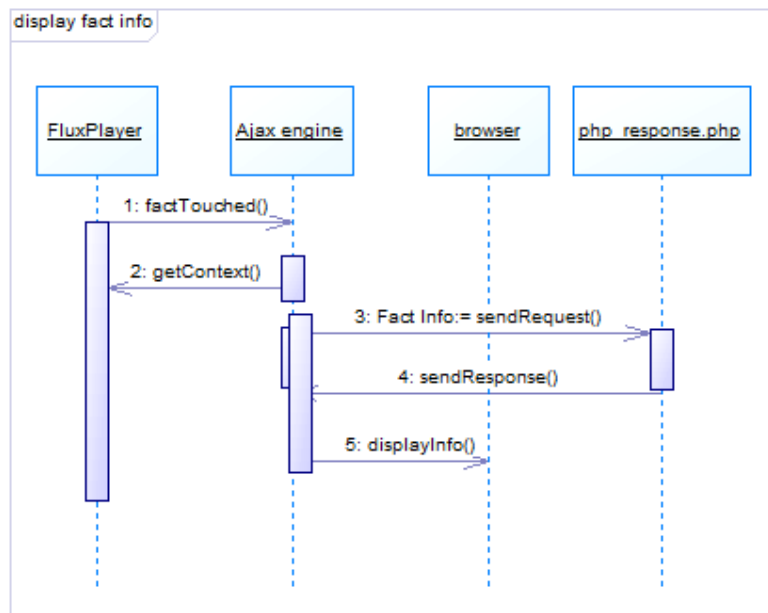
Po kliknutí na tlačidlo zmeny výberu entít vykoná môj program funkcie, ktoré zrušia zobrazené entity a zobrazia nové podľa aktuálnych požiadaviek. Použité funkcie sú veľmi podobné ako pri kliknutí na SkyPort, tentokrát však nepridávam a nemením žiadne pohľady. Jednotlivé synchronizácie sú zobrazené na nasledujúcom obrázku:



Obr. 11: Následnosť funkcií pri zmene výberu entít.

3.4.6 Zobrazenie podrobnejších informácií o entitách

Po kliknutí na entitu poskytujúcu podrobnejšie informácie musím tieto prostredníctvom prístupu Ajax vybrať z databázy a následne zobraziť do okna prehliadača HTML dokumentov na vyhradené miesto. Následnosť jednotlivých funkcií je zobrazená na nasledujúcom obrázku:



Obr. 12: Následnosť funkcií pri zobrazení podrobnejších informácií.

3.5 Plány do budúcnosti

Za hlavné plány do budúcnosti považujem najmä naplnenie samotného obsahu oboch databáz, ako aj doplnenie SkyPortov, keďže tieto sa v bežnom virtuálnom svete nenachádzajú. Na toto dopĺňanie by bolo vhodné vytvoriť jednoduché používateľské rozhranie.

Z používateľského hľadiska by som odporúčala vylepšenie pohybu po oblohe, ktorý by podporoval rýchlejší pohyb na miesto podľa voľby používateľa.

Medzi ďalšie možné zlepšenia by som zaradila vylepšenie efektívnosti a bezpečnosti tejto aplikácie najmä spoluprácou s väčšou a reprezentatívnejšou vzorkou používateľov.

Záver

Ako som už v úvode spomenula, virtuálne prostredia sú čím ďalej tým viac žiadanými, ako v počítačových hrách, tak aj na internete. Oproti tradičným aplikáciám tretí rozmer poskytuje lepšiu vizuálnu predstavu ako aj atraktivitu, preto predpokladám, že vývoj softvérových aplikácií pôjde práve týmto smerom

V tejto práci som si vybrala ako kľúčovú technológiu Ajax, pretože ponúka možnosť jednoduchej, interaktívnej a dynamickej práce s obsahom stránky. Jedným zo základných programovacích jazykov používaných v Ajaxe je JavaScript, ktorý ponúka jednoduchú spoluprácu s VRML scénou a je často používaný pri tvorbe internetových stránok, preto som sa pustila cestou technológie Ajax3D. Táto technológia je dostatočne nová, preto bolo ťažšie nájsť pomoc v podobe tutoriálov, prípadné fór, napriek tomu sa táto voľba ukázala vhodnou a podpora dostačujúcou.

Svojou aplikáciou by som chcela podporiť najmä mladých ľudí, ktorí sú zvyknutí a hlavne motivovaní sa pohybovať vo virtuálnom svete ako aj na internete. Preto sa môžu jednoduchou, nenútenou a intuitívnou formou naučiť niektoré historické fakty. Využitie mojej aplikácie (po správnom doplnení všetkých údajov potrebných na predprípravu) nachádzam napríklad aj v školstve pri výučbe dejepisu, kde by sa ku konkrétnemu miestu pridali relevantné informácie a študenti by sa mohli učiť pohybom po scéne a pozorovaním oblohy. Jednoduchším použitím môže byť zobrazovanie oblohy v existujúcich virtuálnych svetoch, či už založených na realite, alebo nie.

Zoznam obrázkov

1. Textúra pre Sky box	10
2. Porovnanie práce štandardnej web aplikácie a aplikácie využívajúcej Ajax	15
3. Textúra pre Skybox v Active Worlds virtuálnych svetoch	18
4. Schéma spolupráce jednotlivých technológií	26
5. Návrh používateľského rozhrania	29
6a. Svetové súradnice	36
6b. Súradnice pri posúvaní oblohy	36
7. Následnosť funkcií pri načítaní stránky.	39
8. Následnosť funkcií pri zobrazení entít a zmene pohľadu na oblohu.	40
9. Následnosť funkcií pri posúvaní oblohy.	41
10. Následnosť funkcií pri vrátení pohľadu späť z oblohy.	41
11. Následnosť funkcií pri zmene výberu entít.	42
12. Následnosť funkcií pri zobrazení podrobnejších informácií.	41

Zoznam použitých skratiek

Ajax	Asynchronous JavaScript and XML
Ajax3D	Ajax + X3D
ASP	Active Server pages - skriptovací jazyk na strane servera
C++	
COLLADA	COLLABorative Design Activity
DOM	Document Object Model
ECMAScript	
HTML	HyperText Markup Language
ISO	International Standard Organization
JAVA	
JavaScript	
JDBC	Java Database Connectivity
JRE	Java Runtime Enviroment - prostredie potrebné pre beh Java appletov
MySQL	
PHP	Personal Home Page - skriptovací jazyk na strane servera
SAI	Scene Access Interface
SIGGRAPH	Special Interest Group for Computer GRAPHics
VRML	Virtual Reality Modeling Language
WWW	World Wide Web
X3D	eXtensible 3D Graphics
XML	eXtensible Markup Language

Prílohy

Štruktúra .x3d súboru

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN">
<X3D profile='Immersive' >
<head>
</head>
<Scene>
<WorldInfo title='Příklad c.1' info='Jana Dadova'/>
<Transform translation='-1.25835 0 1.18971'>
  <Shape containerField='children'>
    <Appearance containerField='appearance'>
      <Material DEF='Red' containerField='material'
        ambientIntensity='0.200'
        shininess='0.200'
        diffuseColor='1 0 0'/>
    </Appearance>
    <Sphere containerField='geometry'
      radius='1.000'/>
  </Shape>
</Transform>
</Scene>
</X3D>
```

Štruktúra .wrl súboru

```
#VRML V2.0 utf8
WorldInfo {
  title "Příklad c.1"
  info ["Jana Dadova"]
}
Transform {
  translation -1.25835 0 1.18971
  children [
    Shape {
      appearance Appearance {
        material DEF Red Material {
          ambientIntensity 0.200
          shininess 0.200
          diffuseColor 1 0 0
        }
      }
      geometry Sphere {
        radius 1.000
      }
    }
  ]
}
```

Bibliografia

[act96] - Active Worlds. [Online] Activeworlds Inc. [Datum: 21. 02 2008.]
<http://www.activeworlds.com/>.

[AJAX] - AJAX Tutorial. *w3schools*. [Online] [Datum: 13. 02 2008.]
<http://www.w3schools.com/ajax/default.asp>

[AJAX3D] - Parisi, Tony. 2006. AJAX3D Whitepaper. *AJAX3D.org*. [Online] August 2006.
[Cited: April 05, 2008.] <http://www.ajax3d.org/whitepaper/>.

[alp95] - Alpha World. *Active Worlds*. [Online] Active Worlds, Inc. [Datum: 14. 02 2008.]
<http://www.activeworlds.com/worlds/alphaworld/>.

[ASP] - Slaninka, Peter. 2004. ASP. [Online] 2004. [Datum: 10. 04 2008.]
<http://ics.upjs.sk/~jirasek/sps/ASP>.

[ear07] - Google. 2007. Google Earth. [Online] 2007. [Datum: 14. 02 2008.]
<http://earth.google.com/>.

[ECMA] - 16262, ISO/IEC. 1998. *Information technology - ECMAScript language specification, 1998*. 1998.

[fli08] - Flickr. [Online] [Cited: 04 16, 2008.] <http://www.flickr.com/>.

[han99] - Hanuška, Dušan. 1999. Information Systems and VRML Worlds. [Online] 1999.
[Datum: 10. 02 2008.] <http://www.cescg.org/CESCG99/DHanuska/index.html>.

[JAVA] - Sun Microsystems. *Java for developers*. [Online] [Datum: 13. 02 2008.]
<http://java.sun.com/>.

[JJG] - Garrett, Jesse James. 2005. Ajax: A New Approach to Web Applications. *adaptive path*. [Online] 02 18, 2005. [Cited: 04 10, 2008.] Ajax: A New Approach to Web Applications.

[kas07] - Kasík, Pavel. 2007. Google Earth nabízí pohled na hvězdy jako nikdy předtím. *technet.idnes.cz*. [Online] 23. 08 2007. [Datum: 14. 02 2008.] http://technet.idnes.cz/google-earth-nabizi-pohled-na-hvezdy-jako-nikdy-predtim-pi1-/sw_internet.asp?c=A070822_151404_tec_denik_pka.

[lin95] - Linden Lab. [Online] [Datum: 14. 02 2008.] <http://lindenlab.com/>.

[PHP] - PHP.sk. [Online] [Cited: 03 20, 2008.] <http://www.php.sk/>.

[pspad] - PSPad. [Online] [Cited: 04 16, 2008.] <http://www.pspad.com/>.

[qvo02] - Qvortrup, L., Jensen, J.F., Kjems, E., Lehmann, N., Madsen, C. (Eds.),. 2002. *Virtual Space · Spatiality in Virtual Inhabited 3D Worlds*. London; Berlin; Heidelberg : Springer-Verlag, 2002.

[SAW] - Free Active Worlds Sky Boxes. *GreyLight*. [Online] [Cited: 05 26, 2008.]
<http://www.greylight.com/awskybox.html>.

- [sec] - Second Life. [Online] Linden Lab. [Datum: 14. 02 2008.] <http://secondlife.com/>.
- [sko08] - RNDr. Petr Škoda, CSc. Pod společnou oblohou. [Online] [Datum: 21. 02 2008.] http://projekty.astrovm.cz/ps0/vystupy.php?obsah=vystupy_clanek5.php.
- [sky01] - Tiger. 2001. The Sky Box. *Design Tutorials*. [Online] 03 31, 2001. [Cited: 05 26, 2008.] <http://www.blendernation.com/2007/10/03/making-a-skybox-with-blender/>.
- [sky07] - PlantPerson. 2007. Making a Skybox with Blender. *BlenderNation*. [Online] 10 03, 2007. [Cited: 05 26, 2008.] <http://www.blendernation.com/2007/10/03/making-a-skybox-with-blender/>.
- [ste92] - Stephenson, Neal. 1992. *Snow Crash*. New York : Penguin Books, Ltd., 1992. ISBN-10: 0553562614.
- [studio] - *Media Machines*. [Online] [Cited: 04 16, 2008.] <http://www.mediamachines.com/downloads.php>.
- [tam07] - Reuers. 2007. Arabian Business. *TAM takes to virtual skies with Second Life*. [Online] 12. 04 2007. [Datum: 21. 02 2008.] http://www.arabianbusiness.com/index.php?option=com_content&view=article&id=11071.
- [tao08] - The Tao of Linden. [Online] Linden Research, Inc. [Datum: 21. 02 2008.] <http://lindenlab.com/about/tao>.
- [VRML] - 14772-1, ISO/IEC. 1997. *Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language (VRML) - Part 1: Functional specification*, 1997.1997.
- [VRML97] - *VRML97 and Related Specifications*. [Online] [Datum: 10. 02 2008.] <http://www.web3d.org/x3d/specifications/vrml/>.
- [X3D] - 19776, ISO/IEC. 2005. *X3D encodings: XML and Classic VRML*. 2005.
- [zar00] - Doc. Ing. Jiří Žára, CSc. 2000. *Jazyky pro popis virtuální reality*. Praha : Vydavatelstvo ČVUT, 2000.
- [zar99] - Doc. Ing. Jiří Žára, CSc. 1999. *VRML 97 - Laskavý průvodce virtuálními světy*. Brno: Computer Press, 1999.