



SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ
Odbor: Informačné systémy, Softvérové inžinierstvo

Virtuálna FIIT

Projektová dokumentácia





OBSAH

OBSAH	ii
zoznam tabuliek	viii
zoznam obrázkov.....	ix
ZADANIE	xii
1 Úvod.....	1
1.1 Prehľad dokumentu	1
1.2 Skratky.....	1
1.3 Použitá anotácia UML	2
2 analýza problematiky tvorby Virtuálnej fiit.....	3
2.1 Analýza výsledkov podobných projektov.....	3
2.1.1 Analýza Bakalárskeho projektu – Bc. Ondej Ivančík	3
2.1.2 Analýza diplomového projektu – Bc. Gabriel Braniša.....	4
2.2 Analýza 2D plánov novej FIIT, vhodnosť pokračovania v doterajších 3D modeloch	9
2.2.1 Príklady výkresov .dwg:	11
2.2.2 Príklady výkresov .rvt:.....	14
2.3 Analýza O3D	15
2.3.1 Plugin O3d.....	15
2.3.2 Architektúra O3D	15
2.3.3 3D modely.....	16
2.3.4 Vývojárske prostredie pre O3D	16
2.3.5 Podporované platformy pre O3D	16
2.3.6 Hardvérové požiadavky pre O3D	17
2.4 Analýza X3D.....	18
2.4.1 Opis X3D	18





2.5	Loadovanie priestorov pre systém Virtuálnej FIIT	21
2.6	Analýza vývojového prostredia Virtuálnej FIIT	23
2.6.1	Editory vývojového prostredia	23
2.6.2	3D modelovanie.....	24
2.6.3	Debugovanie Javascript kódu	26
2.6.4	Existujúce riešenia zobrazenia 3D scén	26
2.6.5	Prehliadače pre systém Virtuálna FIIT	29
2.7	Analýza modelovacieho nástroja 3DS Max	30
2.8	Zhrnutie analýzy problematiky systému	33
3	špecifikácia riešenia.....	34
3.1	Špecifikácia požiadaviek na systém Virtuálnej FIIT	34
3.1.1	Základné požiadavky na systém a jeho vývoj	34
3.1.2	Hardvérové a softvérové požiadavky na systém	35
3.1.3	Charakteristika používateľov systému.....	35
3.1.4	Požiadavky na funkcionality systému.....	35
3.2	Diagram prípadov použitia.....	36
3.2.1	UC01 Zvolenie módu zobrazenia	37
3.2.2	UC02 Pohybovanie sa po budove	38
3.2.3	UC03 Zobrazenie informácií	40
3.2.4	UC04 Otvorenie a zatvorenie dverí	41
3.2.5	UC05 Použitie výťahu.....	42
3.2.6	UC06 Voľba poschodia.....	44
3.2.7	UC07 Prehliadanie modelu poschodia.....	45
3.2.8	UC08 Vyhľadávanie miestnosti.....	46
3.2.9	UC09 Navigácia medzi miestnosťami	48





3.3	Nefunkcionálne požiadavky na systém	50
4	Návrh systému virtuálnej fiit	51
4.1	Architektúra systému Virtuálnej FIIT	51
4.2	Komponenty systému Virtuálnej FIIT.....	52
4.2.1	Komponenty prezentačnej vrstvy.....	52
4.2.2	Komponenty aplikačnej vrstvy	53
4.2.3	Komponenty dátovej vrstvy.....	55
4.3	Návrh GUI systému	57
4.3.1	Uvítacia obrazovka.....	57
4.3.2	Obrazovka pre 3D mód	58
4.3.3	Obrazovka vyhľadávania miestností.....	58
4.3.4	Obrazovka vyhľadávania cesty	59
4.3.5	Obrazovka prehľadávania	60
4.3.6	Zmeny vo výkresoch	61
4.4	Databázový návrh	66
4.4.1	Logický dátový model	66
4.4.2	Fyzický model.....	67
4.5	Určenie priorít implementácie systému	69
5	Implementácia systému virtuálnej fiit.....	71
5.1	Implementácia používateľského rozhrania.....	71
5.1.1	Základná štruktúra.....	71
5.1.2	Hlavička dokumentov	72
5.1.3	Implementácia Javascriptov	72
5.1.4	CSS kaskádové štýly	73
5.1.5	Úvodná obrazovka	74





5.1.6	Navigácia a módy modelu	76
5.1.7	Základné rozloženie prvkov	76
5.1.8	Mód prehliadania a navigácie.....	77
5.2	Implementácia prototypu modelu.....	80
5.2.1	Modelovanie poschodia	80
5.2.2	Práca s modelom v O3D.....	81
5.2.3	Implementácia navigácie	82
5.2.4	Komunikácia s databázou	84
5.3	Implementácia navigácie – graf miestností	86
5.3.1	Umiestnenie.....	86
5.3.2	Štruktúra	86
5.3.3	Ďalšie pravidlá	87
6	opis realizácie	89
6.1	Modelovanie poschodí v 3DS Maxe.....	89
6.1.1	Proces vymodelovania finálneho modelu poschodia.....	89
6.1.2	Modelovanie špecifických objektov	90
6.1.3	Aplikovanie materiálov elementom komponentu	92
6.1.4	Materiály.....	93
6.1.5	Strecha	94
6.2	Organizácia komponentov v modeli	94
6.3	Implementácia O3D zobrazovacej časti	96
6.3.1	Implementácia menu.....	96
6.3.2	Implementácia vysúvacieho menu	98
6.3.3	Optimalizácia	102
6.4	Automatický update databázy z XML súborov	103





6.4.1	Spustenie synchronizácie.....	103
6.4.2	Postup synchronizácie	103
7	testovanie.....	105
7.1.1	UC01 Zvolenie módu zobrazenia	106
7.1.2	UC02 Pohybovanie sa po budove	107
7.1.3	UC03 Zobrazenie informácií	108
7.1.4	UC04 Otvorenie a zatvorenie dverí	110
7.1.5	UC05 Použitie výťahu.....	111
7.1.6	UC06 Voľba poschodia.....	112
7.1.7	UC07 Prehliadanie modelu poschodia.....	114
7.1.8	UC08 Vyhľadávanie miestnosti.....	115
7.1.9	UC09 Navigácia medzi miestnosťami	117
8	Testovanie konečnými používateľmi.....	120
8.1	Rozdelenie respondentov:	121
8.2	Plynulosť zobrazovania modelov	122
8.3	Spoľahlivosť a správna funkčnosť aplikácie	123
8.4	<i>Jednoduchosť ovládania a rozloženie grafických elementov</i>	124
8.5	VIZUÁLNA STRÁNKA APLIKÁCIE	125
8.6	Súhrn:	126
9	Vylepšenia a Ďalšie smerovanie ďalšej implementácie	128
9.1	Vernosť pôdorysu budovy.....	128
9.2	Vernosť 3d prostredia	128
9.3	Množstvo informácií a dáta	129
9.4	Vyhľadávanie.....	129
9.5	Integrácia	129





9.6	Pomocník - HELP	130
9.7	Nenaplnené požiadavky, resp. Čo sme nestihli	130
10	Záver	132
11	zoznam použitej literatúry	134





ZOZNAM TABULIEK

Tab. 1 Prípád použitia UC01 Zvolenie módu zobrazenia	38
Tab. 2 Popis prípadu použitia UC02 Pohybovanie sa po budove	39
Tab. 3 Popis prípadu použitia UC03 Zobrazenie informácií	41
Tab. 4 Popis prípadu použitia UC04 Otvorenie a zatvorenie dverí	42
Tab. 5 Popis prípadu použitia UC05 Použitie výťahu.....	43
Tab. 6 Popis prípadu použitia UC06 Voľba poschodia.....	45
Tab. 7 Popis prípadu použitia UC07 Prehliadanie modelu poschodia.....	46
Tab. 8 Popis prípadu použitia UC08 Vyhľadávanie miestnosti	48
Tab. 9 Popis prípadu použitia UC09 Navigácia medzi miestnosťami	49
Tab. 10 Aktuálny stav modelov poschodí.....	90





ZOZNAM OBRÁZKOV

Obr. 1 Základný pohľad na kolobeh simulácie	5
Obr. 2 Vybrané aktuálne technológie	6
Obr. 3 Bočný prierez budovou.....	11
Obr. 4 Pôdorys 1. podlažia budovy.....	12
Obr. 5 Prierez poschodiami budovy a pôdorys 4. podlažia budovy	13
Obr. 6 Ivančíkov model -1	14
Obr. 7 Ivančíkov model -2 – prierez	14
Obr. 8 Architektúra pluginu O3D.....	15
Obr. 9 Architektúra X3D aplikácie	19
Obr. 10 Hierarchia profilov X3D	20
Obr. 11 Postupné načítanie izieb.	22
Obr. 12 Obrazovka programu HTMLKit	23
Obr. 13 Obrazovka programu Eclipse.....	24
Obr. 14 Obrazovka programu Google Sketchup	25
Obr. 15 Obrazovka pluginu Firebug v prehliadači Firefox	26
Obr. 16 Obrazovka aplikácie The Forbidden City: Beyond Space & Time	27
Obr. 17 Obrazovka stránky virtualtravel.sk.....	28
Obr. 18 Demo Beach Scene	29
Obr. 19 Material Editor dialóg.....	31
Obr. 20 Export do COLLADA formátu	32
Obr. 21 Okno programu 3DS Max.	33
Obr. 22 Prípád použitia pre bežného používateľa.....	36
Obr. 23 Prípád použitia „Zvolenie módu zobrazenia“	37





Obr. 24 Prípád použitia „Pohybovanie sa po budove“	38
Obr. 25 Prípád použitia „Zobrazenie informácií“	40
Obr. 26 Prípád použitia „Otvorenie a zatvorenie dverí“	41
Obr. 27 Prípád použitia „Použitie výťahu“.....	42
Obr. 28 Prípád použitia „Voľba poschodia“	44
Obr. 29 Prípád použitia „Prehliadanie modelu poschodia“	45
Obr. 30 Prípád použitia „Vyhľadávanie miestností“ v navigačnom móde	46
Obr. 31 Prípád použitia „Navigácia medzi miestnosťami“	48
Obr. 32 Diagram komponentov.....	56
Obr. 33 Uvítacia obrazovka	57
Obr. 34 Obrazovka vyhľadávania miestností.....	59
Obr. 35 Obrazovka vyhľadávania cesty	60
Obr. 36 Obrazovka prehľadávania.....	61
Obr. 37 Pôvodná časť poschodia 4NP	62
Obr. 38 Upravená časť poschodia 4NP	63
Obr. 39 Pôvodná časť poschodia 4NP	63
Obr. 40 Upravená časť poschodia 4NP.....	64
Obr. 41 Pôvodná časť poschodia 4NP	64
Obr. 42 Upravená časť poschodia 4NP	65
Obr. 43 Pôvodná časť poschodia 4NP	65
Obr. 44 Dátový model	66
Obr. 45 Fyzický model	67
Obr. 46 Editor PSPad 4.5.4	71
Obr. 47 Implementovaná úvodná obrazovka.....	75
Obr. 48 Obrazovka módu prehliadania	78





Obr. 49	Obrazovka módu navigácie	79
Obr. 50	Chybné zobrazenie modelu v O3D – priehľadné steny	80
Obr. 51	Zobrazenie modelu v 3DS Max – priehľadnosť stien sa neprejavila	81
Obr. 52	GUI - Navigácia medzi dvomi miestnosťami.....	83
Obr. 53	Chyba pri navigácii – skracovanie cez neverejné priestory.....	84
Obr. 54	Diagram komunikácie prototypu s databázou	85
Obr. 55	Na spoji navigačných hrán nie je žiadny vrchol – komponent.	87
Obr. 56	Abstrakcia okien modelu na priehľadné plochy	91
Obr. 57	Nadstavenie parametrov komponentov dverí	92
Obr. 58	Dvere s priehľadnou výplňou	92
Obr. 59	Aplikovanie priehľadného materiálu elementu dverí	93
Obr. 60	Získanie materiálu odobratím jeho vzorky z komponentu.....	94
Obr. 61	Hierarchia komponentov poschodia	95
Obr. 62	Načítaná stránka so zasunutým menu	99
Obr. 63	Dočasne vysunuté menu (šírka vysunutia nastavená na polovicu).....	99
Obr. 64	Pevne vysunuté menu	100
Obr. 65	Prípady použitia.....	105





ZADANIE

Pamätáte si, ako vždy začína každý semester? Zistíte si, aký je váš rozvrh, no v ňom sú záhadne zakódované čísla miestností, v ktorých máte cvičenia. A ako dlho vám trvá, kým nájdete miestnosť, v ktorej sa nachádza váš vedúci? A ako zistíte, kedy má váš prednášajúci konzultačné hodiny? Alebo ktorý cvičiaci má v danej miestnosti cvičenie hneď po vás?

Toto je len pár z mnohých problémov. Ich riešenie momentálne spočíva v tom, že si musíte otvoriť ten správny informačný zdroj a v ňom informáciu nájsť a aj tak vám nakoniec nikto nepovie, kde je miestnosť BX04. Nebolo by to krásne, keby ste jednoducho počítaču zadali číslo miestnosti alebo meno človeka a on vás k nemu virtuálne zaviedol? A čo tak keby to všetko fungovalo, až budeme mať novú budovu FIIT?

Vašou úlohou bude:

- *zanalyzovať danú oblasť*
- *vytvoriť skutočný 3D model novej budovy FIIT*
- *navrhnuť a zimplementovať jeho interaktívne časti ako je napríklad otváranie a zatváranie dverí, informačné tabule, možnosť zadania otázky, nájdenie najlepšej cesty a navigovanie používateľa po nej*
- *to všetko by samozrejme nešlo bez databázy*
- *riešenie bude potrebné optimalizovať tak, aby ho používateľ mohol používať aj cez jednoduché webové rozhranie*
- *na záver nesmie chýbať testovanie a vyhodnotenie použiteľnosti*





1 ÚVOD

Predkladaný dokument obsahuje projektovú dokumentáciu vytvorenú v rámci predmetu Tvorba informačných systémov v tíme.

Cieľom tejto dokumentácie je vysvetliť riešenie celého projektu s názvom Virtuálna FIIT. Navrhovaný systém by mal slúžiť ako navigácia po novej budove fakulty spolu s informáciami o jednotlivých miestnostiach.

1.1 Prehľad dokumentu

Dokument je rozdelený na 4 časti:

1. úvod
 - počiatočné informácie o dokumentácii
2. analýza problému
 - analýza problémovej oblasti, rozoberá existujúce riešenia v danej oblasti, rozoberá možnosti grafického zobrazenia.
3. špecifikácia riešenia
 - popis funkcionálnych a nefunkcionálnych požiadaviek
4. návrh
 - návrh riešenia po klientskej aj serverovej stránke

1.2 Skratky

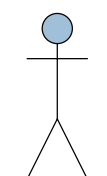
BP	bakalárska práca
DP	diplomová práca
X3D	eXtensible 3D
O3D	open 3D
3DS Studio Max	nástroj na tvorbu 3D modelov
XML	eXtensible Markup Language -jazyk pre uchovávanie štruktúrovaných dát
API	application programming interface
GPU	graphics processing unit
UCXY	use case XY – prípad použitia XZ





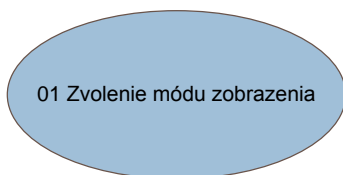
UML	Unified Modeling Language -grafický jazyk na v vizualizáciu, špecifikáciu, navrhovanie a dokumentáciu systémov
SVN	Subversion - je systém podpory revízií a verziovania
CVN	Concurrent Versions System –predchodca SVN
MMORPG	Massively multiplayer online role-playing game
NPC	non-player character

1.3 Použitá anotácia UML



Používateľ

používateľ



01 Zvolenie módu zobrazenia

prípád použitia



vzťah používateľa k prípadu použitia





2 ANALÝZA PROBLEMATIKY TVORBY VIRTUÁLNEJ FIIT

Táto časť dokumentácie sa venuje analýze problémovej oblasti, už existujúcich riešení, možnosti ďalšieho iného riešenia. Na úvod najskôr zanalyzujeme už vypracované práce študenta z bakalárskeho aj inžinierskeho štúdia. Ďalej sa budeme analyzovať 2D plány novej budovy FIIT a ich pokračovanie na 3D, povieme o možnom riešení v O3D, X3D. Následne uvedieme do problematiky loadovania priestorov a analýzou vývojového prostredia.

2.1 Analýza výsledkov podobných projektov

Z dostupných prác zaoberajúcich sa technológiami a problémami podobnými tomu nášmu sme rozobrali dve z nášho pohľadu najlepšie. Sú uvedené v samostatných kapitolách nižšie. Ostatné práce nepoužívali vhodné technológie a ich relevantnosť k nášmu problému je nízka.

2.1.1 Analýza Bakalárskeho projektu – Bc. Ondej Ivančík

Vo svojej práci využil štandard X3D (eXtensible 3D):

- jazyk na modelovanie 3D objektov, ktoré je možné po nainštalovaní vhodného pluginu (autorovi najviac vyhovoval BS Contact) zobrazíť aj v internetovom prehliadači
- vychádza z medzinárodného štandardu VRML (Virtual Reality Modeling Language)
- základom je XML, ktorý slúži na reprezentáciu 3D grafiky

V práci autor popisuje množstvo editorov a prehliadačov formátu X3D, odporúča Vivaty Studio a BS Contact, (prípadne OctagaPlayer).

Vytvorenie modelu:

- najskôr sa pokúsil vytvoriť model v 3DS Max Studiu, ale ten je vhodný len pre malé objekty (domy) a vymodelovať sa tam dá maximálne 1 poschodie
- nakoniec použil Revit Architecture, v ktorom vytváral model prehľadne v 2D priestore rozložený na podlažia, pričom tento nástroj umožňuje použiť predlohy technických výkresov formátu dwg. Podľa nich dokázal vymodelovať maximálne presný model, pričom v každom okamihu je možný 3D náhľad. Nevýhoda Revit Architecture bola v tom, že častokrát bol nestabilný a model ukladal až v 8 súboroch. Taktiež je to platený produkt, zadarmo sú 30 dňové verzie.
- model bolo potrebné zjednotiť a optimalizovať v AutoCad-e





- tiež bolo potrebné celý model importovať do 3DS Max-u, pretože priama konverzia dwg formátu na X3D nebola možná
- 3DS Max Studio dokázalo exportovať model do formátu vrml, v ktorom autor vo Vivaty Studiu nastavil svetlá, kamery, textúry a následne ho exportoval do požadovaného formátu X3D
- interakciu vytvoril taktiež vo Vivaty Studiu

V modeli (formát rvt) je možné podľa autora pokračovať, ale veľmi problematické sú konverzie do množstva formátov. Taktiež nástroj Revit Architecture je veľmi dobrý na veľké objekty, ale jeho používanie je komplikované a veľmi zložité vďaka jeho robustnosti. Taktiež model by bolo vhodné rozdeliť na menšie časti a použiť loading pri zobrazovaní modelu. Toto by mohli mnohé konverzie skomplikovať. Posledným problémom toho modelu bolo zvažujúce sa prvé podlažie, ktoré bolo v modeli vodorovné a nezodpovedalo realite. [9]

2.1.2 Analýza diplomového projektu – Bc. Gabriel Braniša

Analýza zadania diplomového projektu

Diplomová práca Bc. Braniša sa zaoberá tematikou evolučných multiagentových systémov z pohľadu 3D modelov a aplikovaním fyzikálnych vlastností na ne. Projekt je spracovaný ako návrhom softvérového simulátora na simuláciu programovateľných 3D rozumných agentov dynamické aspekty fyziky a jej zákonov, vlastností materiálov a podobne. Zadaním bolo teda návrh a implementácie softvérového nástroja pre modelovanie agentov.

Projekt sa zaoberá fázami vývoja softvéru, analýzou prostredia riešeného problému s popisom modelovej aplikácie, návrhom výslednej aplikácie a popisom implementácie výslednej aplikácie v technickej dokumentácii.

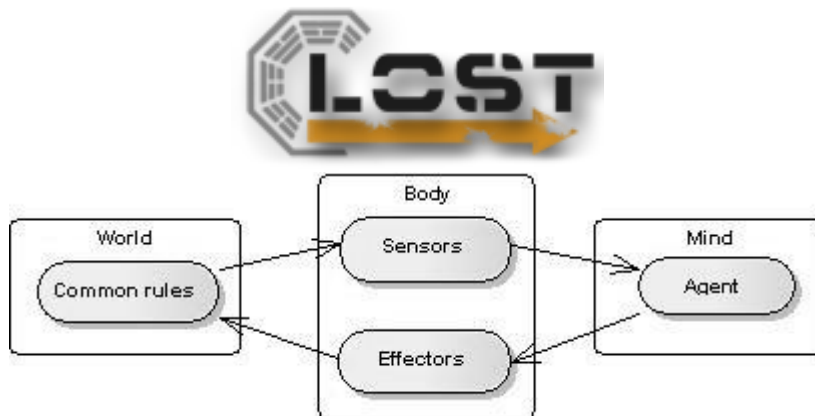
Významnejšie tematické prierezy

Po prečítaní práce som postrehol niekoľko bodov, ktoré podľa môjho názoru by mohli byť prínosné (resp. majú opodstatnený vzťah s našim projektom) a popisujem ich v tejto kapitole.

Simulačné procesy

Simulácia ako taká je zovšeobecnenie interakcie dvoch prvkov, tela a mysle (obr.1).





Obr. 1 Základný pohľad na kolobeh simulácie

Túto schému reprezentuje pojem „agent“ obsahujúci Disponuje senzormi a reflektormi. Ten na prostredie reaguje a tak vzniká životný cyklus agenta.

Multiagentové systémy

Multiagentové systémy sú simulátory umožňujúce spustiť životný cyklus určitých agentov v definovanom prostredí.

Zoznam najčastejšie používaných multiagentových simulátorov:

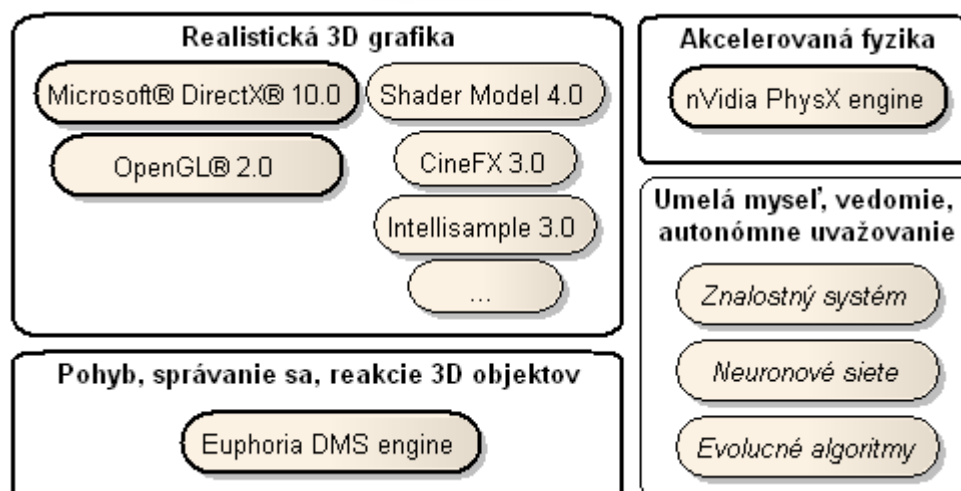
- Simspark
- Webots
- Endorphin

Simulačné technológie

Obrázok 2 zobrazuje vybrané aktuálne technológie v štyroch kategóriách:

- Realistická 3D grafika
- Akcelerovaná fyzika
- Pohyb, správanie sa, reakcie 3D objektov
- Umelá myseľ, vedomie, autonómne uvažovanie





Obr. 2 Vybrané aktuálne technológie

Časové úseky, vplyv oneskorení, regulácia

Simulácia reálneho sveta na počítači prebieha priblížením sa ku kontinuite času zvolením čo najmenšieho časového úseku medzi diskretnými hodnotami na časovej osi. Vznikajú tak kroky simulácie. Presnosť simulácie je tým väčšia, čím je kratší krok simulácie. Voľba veľkosti časového úseku kroku simulácie závisí od zložitosti simulovanej scény, od výkonu výpočtového prostriedku a od požiadavky zmeny 3D scény v reálnom čase.

Hodnoty aj pojmy:

- **Sim step**
 - Logické časové kvantum kroku simulácie
- **Time step**
 - Reálny čas určený pre výpočet scény v novom kroku simulácie
- **Server compute time**
 - Reálny čas strávený na výpočet scény v danom kroku simulácie
- **Synchro time (+/-)**
 - Časový rozdiel medzi *Time step* a *Server compute time*, ak je kladná hodnota, v jej rozsahu server čaká k zahájeniu ďalšieho kroku simulácie. Ak je hodnota záporná, vygeneruje sa upozornenie, že *Server compute time* presahuje hodnotu *Time step* a je potrebné vykonať opatrenia (zväčšiť výkon, zmenšiť zložitnosť simulovanej scény, alebo akceptovať trhavosť vykresľovanej scény)





- **Network delay**

- Oneskorenie pri prenose dát cez počítačovú sieť medzi serverom a klientom. Pokiaľ prebieha komunikácia na zahltenej linke alebo slabou prenosovou rýchlosťou so stratou dát, dochádza na strane klienta k neaktuálnosti údajov a neskorá odozva klienta sa na strane servera preukáže pomalými reakciami agenta. Z tohto dôvodu je zavedená možnosť synchronizácie, opísaná nižšie.

- **Client compute time**

- Reálny čas strávený na výpočet, resp. spracovanie vnemov a generáciu reakcií agenta. Ak je tento čas väčší ako doba medzi jedným a nasledujúcim krokom simulácie (informáciu o aktuálnom kroku simulácie sa dozvedá komunikáciu so serverom), môže dochádzať k nepresnostiam (riešenie vo zvýšení výkonu stroja klienta, aplikovanie synchronizácie)

Požiadavky, obmedzenia:

- **Výpočtový výkon**

- Záleží od použitej platformy, výkonnosti hardvéru, od módu spustených aplikácií

- **Zložitosť 3D scény**

- Množstvo fyzikálnych objektov navzájom kolidujúcich
- Detailnosť (textúry, materiál), komplikovanosť (zložité tvary) objektov
- Vizúálne efekty

- **Interakcia v reálnom čase**

- Plynulosť a prirodzenosť simulácie, možnosť kedykoľvek zasahovať do diania

- **Synchronizácia**

- Server nevstupuje do riešenia ďalšieho kroku simulácie, pokiaľ všetci pripojení klienti neukončili svoju činnosť / výpočty.

Čím je vyšší výpočtový výkon, tým je možné zobrazit' v reálnom čase zložitejšiu 3D scénu. Pokiaľ nezáleží na interakcií v reálnom čase, t.j. *Time step* je radovo väčší ako *Sim step*, je možné simulovať veľmi komplikované scény aj na menej výkonnom stroji. Využitie spočíva v učiacom procese agenta, kde nie je potrebná interakcia s používateľom.





Analyzovaný projekt priamo nesúvisí s tímovým projektom, nakoľko sa zaoberá simuláciami agentov v 3D modeloch. Analýza by bola prospešnejšia, ak by sa náš projekt virtuálnej budovy zaoberal aj implementáciu samostatných agentov. Napriek tomu, poukazuje na niektoré fakty, ktoré je potrebné zahrnúť do úvah o návrhu a implementácii ako:

- Využitie fyzikálnych modelov v budove pri interakcii
- Vhodnosť využitia jednotlivých grafických technológií
- Využitie interaktívnych agentov
- Faktory ovplyvňujúce prepojenie aplikácie so serverom [8]





2.2 Analýza 2D plánov novej FIIT, vhodnosť pokračovania v doterajších 3D modeloch

2D výkresy v AutoCAD-e:

- Podrobné a zložité, je potrebné mať aspoň min. znalosti v technickom kreslení.
- Obsahujú označenie miestností podľa blokov a poschodí.
- Pôdorys je komplikovanejší, pretože celé podlažie budovy je na stúpajúcom teréne, nachádzajú sa v ňom schodiská a prednáškové miestnosti, ktoré zaberajú viac ako 1 podlažie.
- Podlažia od 2. poschodia sú z časti rovnaké – stredová hala, chodby, tvar poschodia je rovnaký, rozdielne sú veľkosti miestností a polohy dverí.
- V BP Ivančička sa nachádza 3D model budovy, ktorý bol pravdepodobne zhotovený poskladaním výkresov z AutoCAD-u.

3D modely:

- Ivančík kreslil 3D model v Revit Architecture, pretože ten je vhodný na kreslenie veľkých budov, vychádzal z výkresov:
 - Kreslil postupne podlažia, model bol zjednotený, optimalizovaný v AutoCAD-e a konvertovaný na X3D. Formát X3D sa dá zobrazíť v prehliadači BS Contact, chodiť po ňom. Vo vnútri je namodelovaná časť vstupnej haly a priestory podlaží pri schodisku, všetky miestnosti by bolo potrebné dorobiť. Niektoré časti sú namodelované chybné. Autor uvádza, že by sa v modelovaní zo súboru typu .rvt dalo v modelovaní pokračovať.
 - Na konvertovania treba dávať pozor, pretože sa pri nich strácajú detaily a rôzne verzie súborov sú zobrazované inak.
- Kozák modeloval v 3D max štúdiu, ale iba jedno podlažie, ale mal vyriešené väčšie detaily.
- 3D model Zjaru sa nepodaril dobre zobrazíť, bol vidieť iba šedý tvar budovy. Problém bol pravdepodobne spôsobený obmedzeniami hardvéru, slabá grafická karta.
- Tiež je možné importovať Ivančíkov model FIIT-ky vo formáte 3ds do Google Sketchup-u, ale práca v ňom je na rozsiahlosť modelu, zdeformovanie množstvom





konverzií a importov zložitá a zobrazenie farieb (pravdepodobne „textúr“) bolo odlišné od zobrazenia v iných nástrojoch.

Vhodnosť pokračovania v doterajších 3D modeloch a nástrojoch:

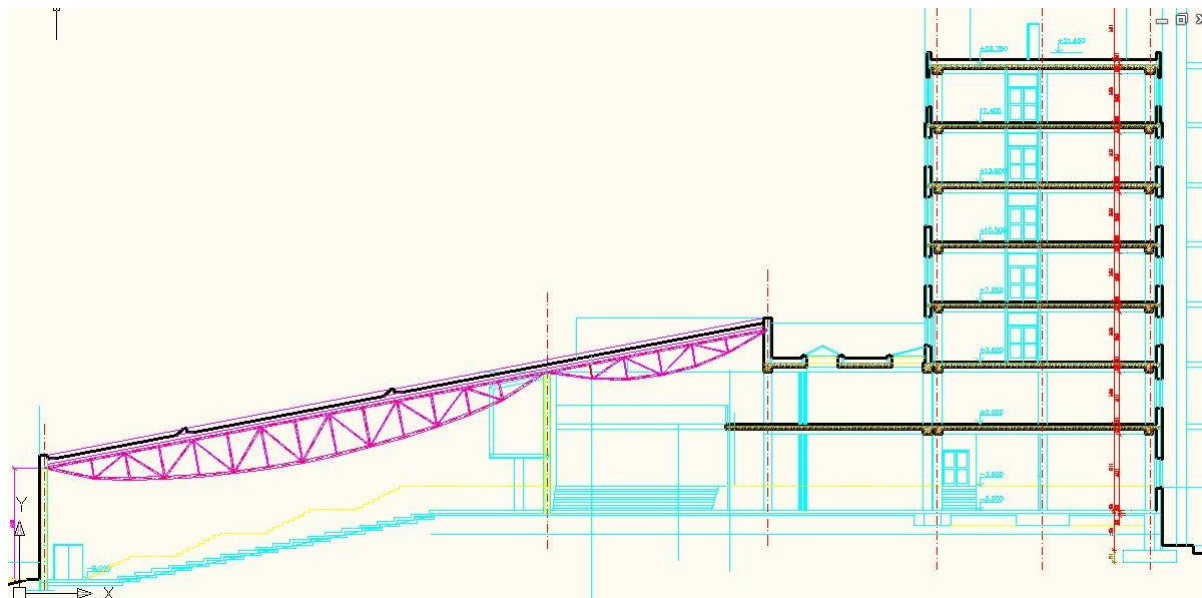
- Vzhľadom na techniku O3D, ktorú plánujeme použiť, tak je k nej určený a prispôsobený modelovací nástroj **Google SketchUp 7**. Pri tvorbe modelov umožňuje rozloženie objektov na komponenty, ktoré sú v O3D určené na prácu s animáciami. Taktiež je potrebná iba jedna konverzia modelu. Ak bude potrebné rozdeliť model na viac častí kvôli rozsiahlosti a požadovaným pohľadom, tak Google SketchUp 7 by mohol byť vhodný nástroj na tvorbu modelov, pretože po podlažiach sa bude dať model vytvárať ľahko a rýchlo. Plusom je jeho nenáročné a efektívne používanie.
- **Revit Architecture** – je v ňom vytvorený základ modelu budovy (rozložený aj na podlažia), v ktorom sa dá pokračovať, čo je veľká výhoda. Tento nástroj je určený na modelovanie veľkých budov, pri ktorých sa dá vychádzať z technických výkresov. Nevýhodou je jeho zložité používanie, množstvo konverzií modelov, kým by sa prevedie do nami požadovaného formátu dae (resp. o3dtgz) a tiež nevyhnutná konverzia cez ďalší robustný nástroj 3DS Max Studio. Používanie týchto dvoch nástrojov je pre komerčné účely platené, dostupné sú len 30 dňové verzie. Otázne je prípadné skonvertovanie vytvoreného modelu a jeho kvalita pri importovaní do nástroja Google SketchUp 7.

Keďže nie je požadovaná 100% podoba 3D modelu ku skutočnej budove kvôli niektorým verejnosti nedostupným informáciám, tak by nám mal postačiť na modelovanie Google SketchUp 7. Ten taktiež zachováva metriku modelu a umožňuje importovať technické výkresy vo formáte bmp, exportované z AutoCad-u. Dosahovaná presnosť postačuje pre nami požadované účely.



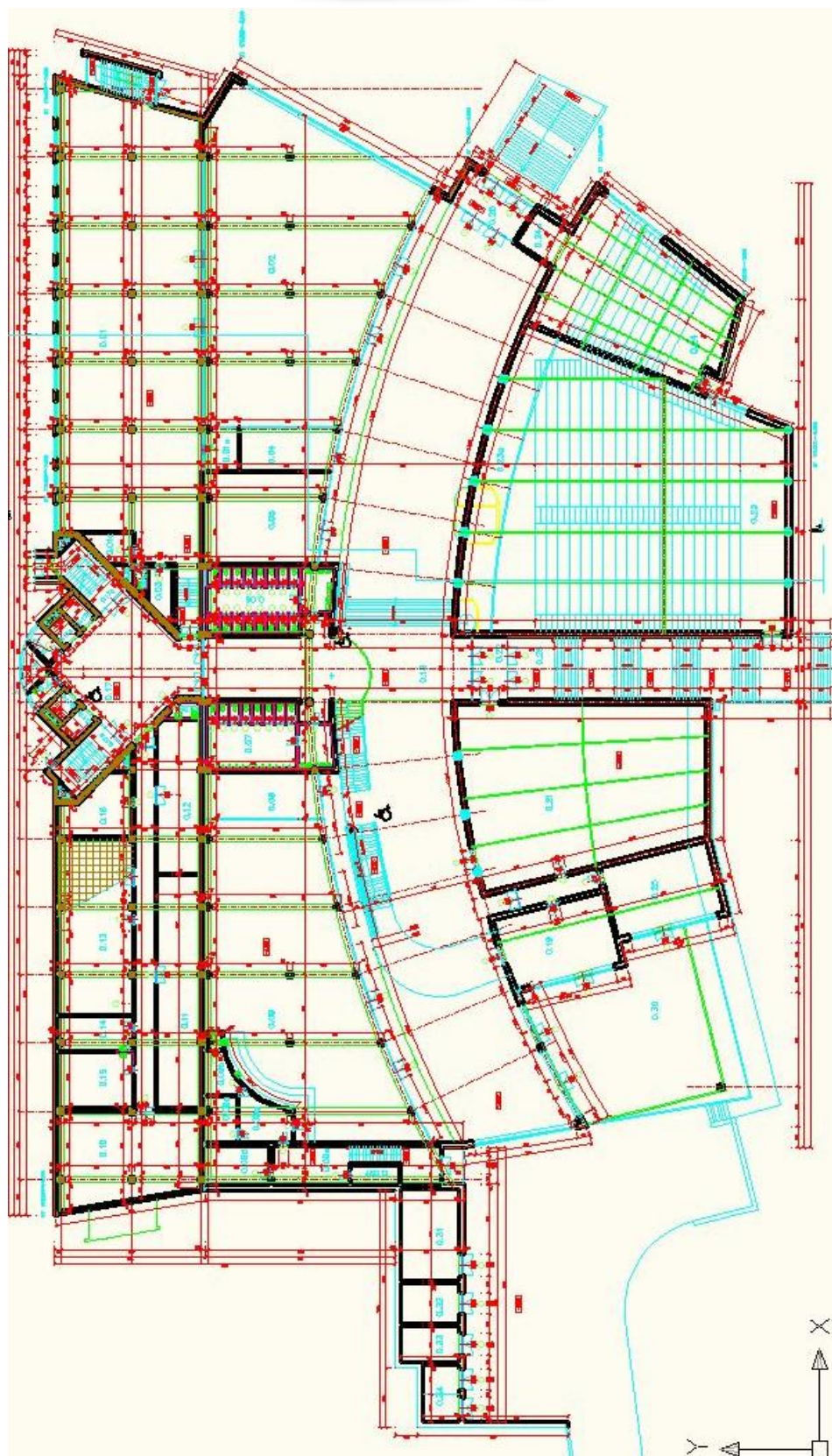


2.2.1 Príklady výkresov .dwg:



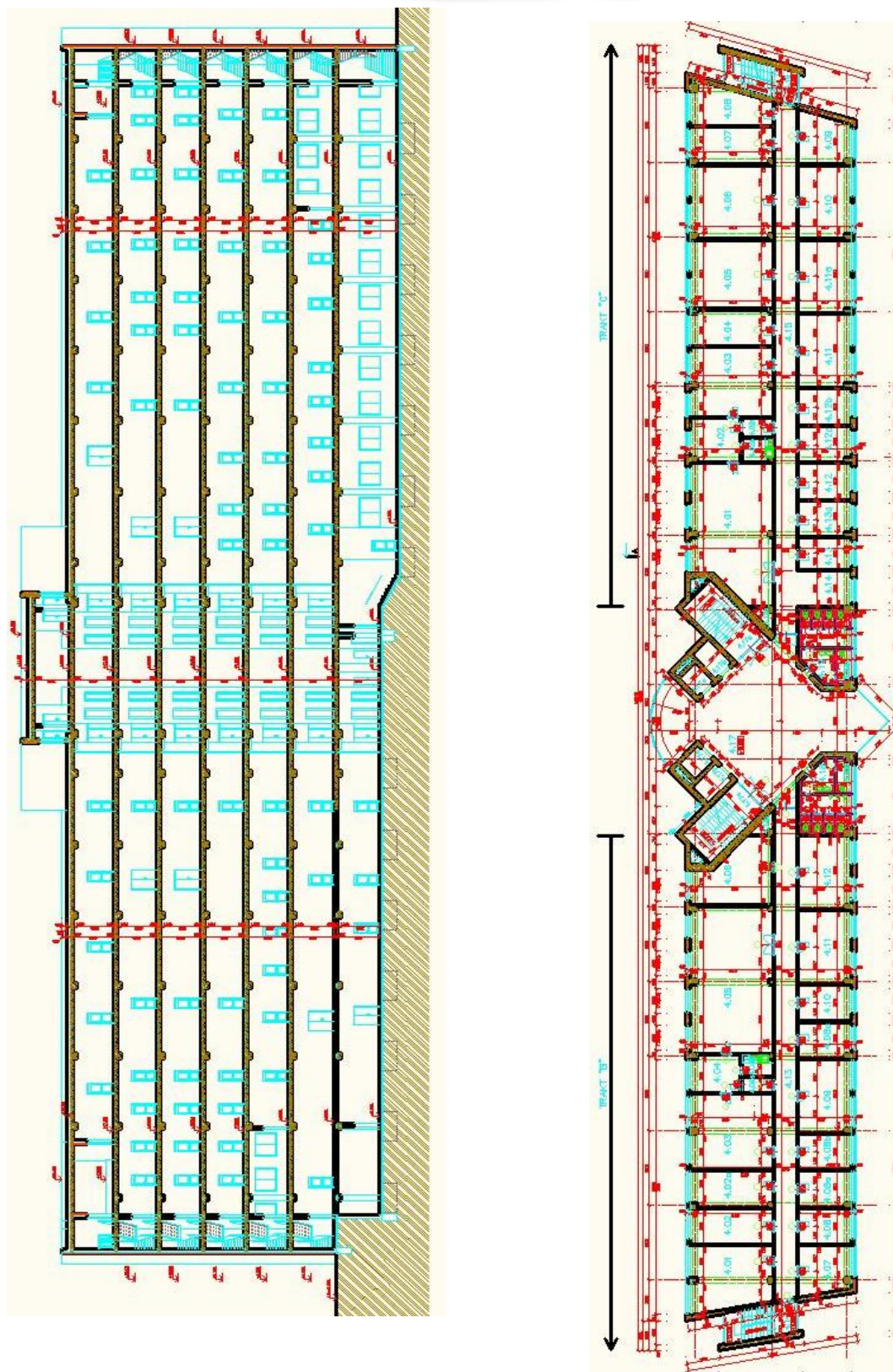
Obr. 3 Bočný prierez budovou





Obr. 4 Pôdorys 1. podlažia budovy



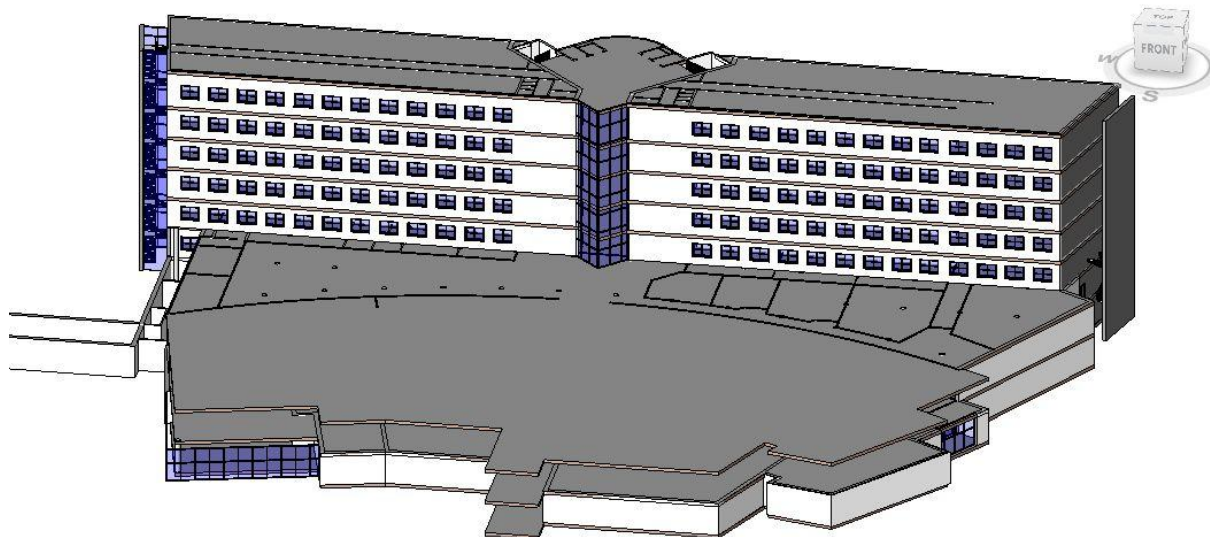


Obr. 5 Prierez poschodiami budovy a pôdorys 4. podlažia budovy

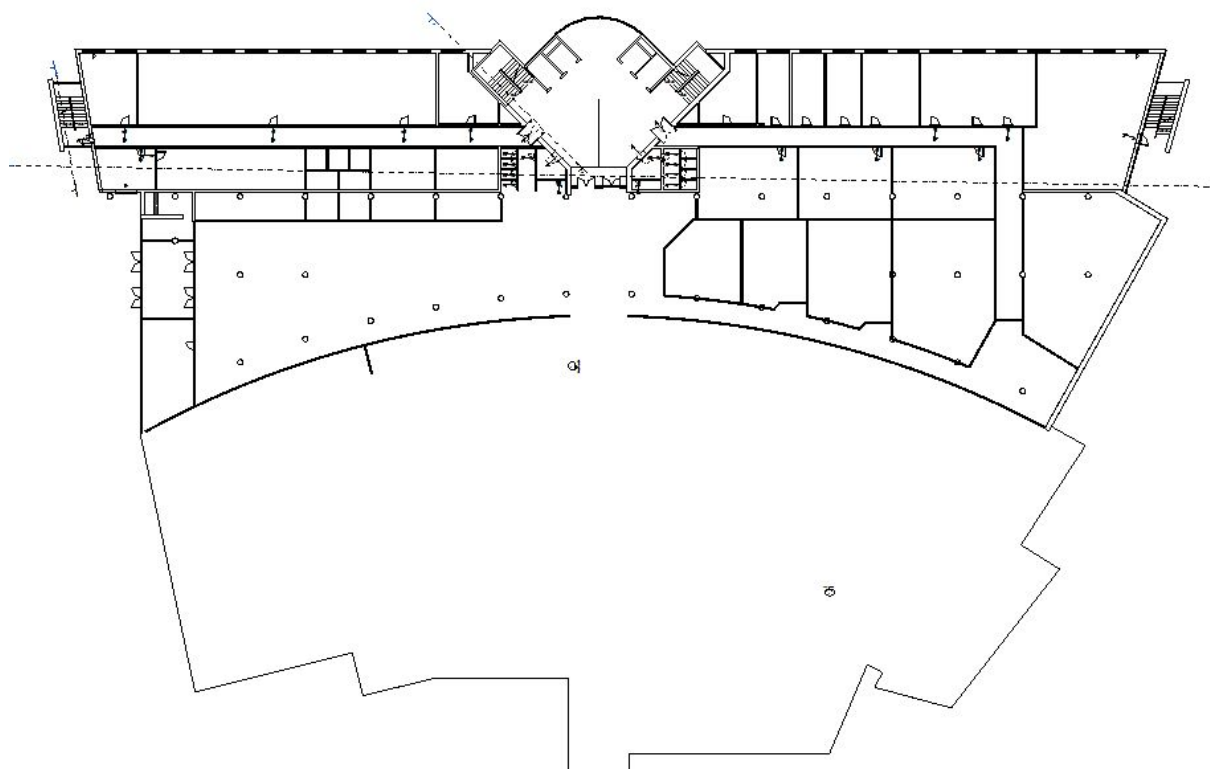




2.2.2 Príklady výkresov .rvt:



Obr. 6 Ivančíkov model -1



Obr. 7 Ivančíkov model -2 – prierez





2.3 Analýza O3D

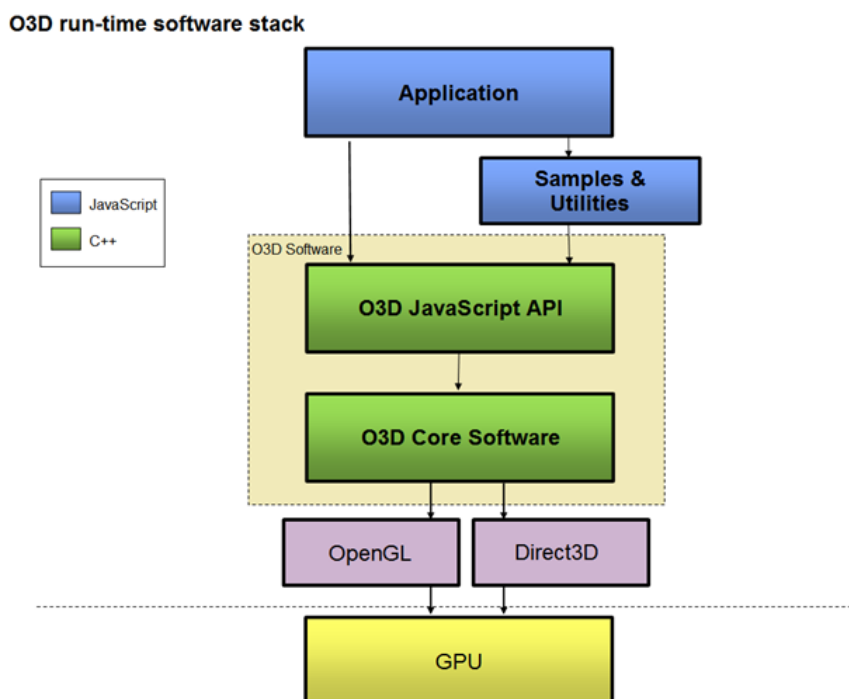
2.3.1 Plugin O3d

O3D [1] je nízkoúrovňové API určené na tvorbu interaktívnych 3D aplikácií bežiacich vo webovom prehliadači. Distribuuje sa formou pluginu, ktorý je používateľovi ponúknutý na nainštalovanie vo všetkých internetových prehliadačoch po navštívení webovej stránky s O3D obsahom. O3D je vyvíjané spoločnosťou Google ako opensource projekt. [3]

2.3.2 Architektúra O3D

O3D predstavuje rozhranie medzi internetovým prehliadačom a grafickými prostriedkami počítača, ktoré je naprogramované v jazyku C++. Vývojár k nemu pristupuje prostredníctvom funkcií volaných z JavaScriptu. O3D aplikácie sú väčšinou tvorené kombináciou HTML a JavaScript kódu.

O3D pristupuje ku grafickej karte prostredníctvom DirectX (Windows) alebo OpenGL (Mac a Linux). Pokiaľ nie je v operačnom systéme prístupná akcelerácia grafickou kartou, využije sa na renderovanie 3D scény procesor. Vďaka svojej architektúre poskytuje O3D plugin porovnateľný výkon ako majú bežné desktopové 3D aplikácie.



Obr. 8 Architektúra pluginu O3D





2.3.3 3D modely

O3D poskytuje dva spôsoby práce s 3D modelmi:

1. Vytvorenie modelu priamo v O3D

Pred samotným vytváraním objektov je najskôr potrebné pochopiť ich štruktúru v O3D. Každý objekt je definovaný svojim tvarom (shape). Tvar objektu je zložený z ľubovoľného počtu primitív (primitives), na ktoré sú aplikované materiály a efekty. Materiály a efekty môžu byť zdieľané viacerými primitívami pre šetrenie systémových prostriedkov. O3D poskytuje na vykresľovanie objektov dva zoznamy, z ktorých sa primitívy vykresľujú. Prvým je Performance DrawList, v ktorom sa nachádzajú nepriehľadné primitívy a druhým je Transparency DrawList pre priehľadné primitívy. Pri vytváraní objektov je kvôli vyššiemu výkonu vhodné primitívy správne rozdeľovať a nie ich všetky zaradiť do univerzálnejšieho Transparency DrawList.

2. Importovanie modelov

O3D aplikácie využívajú na opis 3D modelov formát COLLADA [2] (prípona „.dae“). COLLADA je otvoreným štandardom a jeho štruktúra je definovaná v XML schema. Tento formát je podporovaný najpoužívanejšími 3D modelovacími nástrojmi (3ds Max, Maya, Google SketchUp, Blender...). Samotný COLLADA súbor však nie je možné importovať do O3D aplikácie priamo, ale musí byť skontvertovaný do súboru s príponou „.o3dtgz“ pomocou COLLADA Converteru, ktorý sa nachádza na stránkach O3D projektu.

2.3.4 Vývojárske prostredie pre O3D

Keďže O3D je v súčasnosti novom technológiou (predstavená bola v máji 2009) neexistuje pre ňu momentálne žiadne vývojové prostredie, ktoré by ju priamo podporovalo. Používateľský kód je písaný v JavaScripte, takže na zvýraznenie syntaxe je možné použiť ľubovoľný editor (napríklad NotePad++, HTML-Kit). Debugovanie JavaScriptu je možné realizovať napríklad rozšírením Firebug pre internetový prehliadač Firefox. Jednou z pokročilých funkcií, ktorú O3D ponúka sú programovateľné shadre. Ich programovanie sa realizuje v jazyku založenom na HLSL a Cg. Na jeho zvýraznenie postačuje editor s podporou zvýraznenia syntaxe jazyka C.

2.3.5 Podporované platformy pre O3D

O3D je podporované hlavnými operačnými systémami a prehliadačmi:

Windows:

XP Service Pack 2, Vista (x86/x64) Service Pack 1

Internetové prehliadače: Firefox 2+, Internet Explorer 7.0+ (x86), Google Chrome





Podpora internetového prehliadača Opera by sa mala objaviť v najbližších verziách.

Mac:

Intel Mac s OS X v10.4 alebo novší

Internetové prehliadače: Firefox 2+, Safari 3+, Camino

Linux:

V súčasnosti je dostupná len 32-bitová verzia pluginu O3D, ktorú treba skompilovať zo zdrojových kódov, avšak s príslušnými knižnicami a 32-bitovým prehliadačom funguje aj na 64-bitovej distribúcii.

2.3.6 Hardvérové požiadavky pre O3D

Windows:

x86 kompatibilný procesor

Grafická karta s podporou pre DirectX 9, VertexShader 2.0 a PixelShader 2.0

Mac:

Ľubovoľný Intel Mac

Ak nie je grafická karta podporovaná, využije sa na renderovanie procesor

Realizácia projektu virtuálnej FIIT pomocou pluginu O3D sa ukazuje ako jedno z perspektívnejších riešení. Plugin O3D ponúka vysoký výkon na všetkých dostupných platformách, pokročilé funkcie pre prácu s prostriedkami grafickej karty (programovateľný pipeline), zdokumentované API a stále sa rozširujúcu komunitu. Okrem toho je jeho autor Google zárukou kvality a budúceho rozšírenia O3D pluginu na internete. Štandardizácia a otvorený zdrojový kód môžu spôsobiť, že sa O3D stane v budúcnosti súčasťou webových prehliadačov.

Nevýhodami výberu O3D na realizáciu projektu virtuálnej FIIT môžu byť zatiaľ nízka rozšírenosť a neexistencia IDE, ktoré by priamo podporovalo vývoj webových aplikácií v O3D.





2.4 Analýza X3D

2.4.1 Opis X3D

X3D je otvorený súborový formát a architektúra pre reprezentáciu a výmenu 3D scén a objektov. Tento, na XML založený formát, bol vytvorený organizáciou Web 3D Consortium a bol povýšený na ISO štandard. X3D vychádza zo staršieho štandardu VRML (a stále akceptuje jeho syntax), dokáže pracovať s textúrami, osvetlením, animáciami alebo systémom spracovania kolízií.

Príklad kódu v novšej (XML) syntaxe:

```
<Transform scale='0.91 0.6 0.3' translation='0.8 -0.65 0.5'>
  <Shape>
    <Appearance>
      <Material diffuseColor='0.749 0.694 0.651' />
    </Appearance>
    <Cylinder bottom='false' top='false' />
  </Shape>
</Transform>
```

2.4.1.1 X3D prehliadače

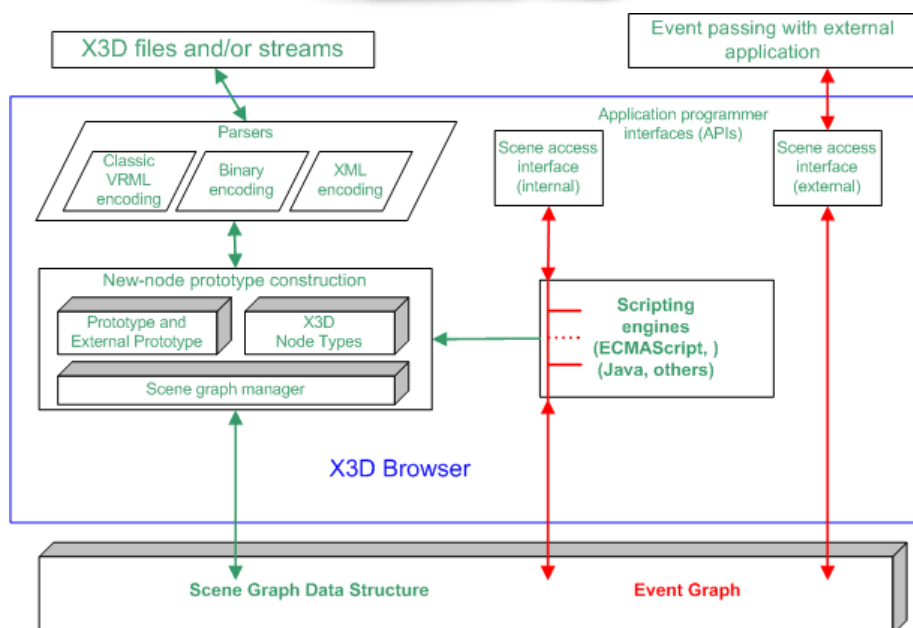
Súbor vo formáte X3D je len pasívny zápis, na vytvorenie viditeľnej trojrozmernej scény je potrebný tzv. X3D prehliadač (browser). Je to program, ktorý načíta vstup vo formáte X3D a zobrazí ho v reálnom čase. Okrem toho podporuje animácie a interakciu s používateľom. Prehliadače sú buď samostatné aplikácie, no väčšinou majú formu inštalovateľných rozšírení, tzv. pluginov do klasických internetových browserov. Možnosti konkrétneho X3D modelu ale aj hardvérová či platformová kompatibilita sú tak určené implementáciou daného prehliadača.

Web 3D Consortium ponúka vlastný browser Xj3D, vytvorený v Jave, ďalšie prehliadače sú napr. FreeWRL, Cortona, Octaga player a pod. [4]

2.4.1.2 Architektúra X3D aplikácie

Každá X3D aplikácia je vo všeobecnosti priestor obsahujúci grafické a zvukové objekty, ktoré môžu byť načítané a manipulované viacerými spôsobmi. Obrázok 9 zachytáva architektúru takejto aplikácie.





Obr. 9 Architektúra X3D aplikácie

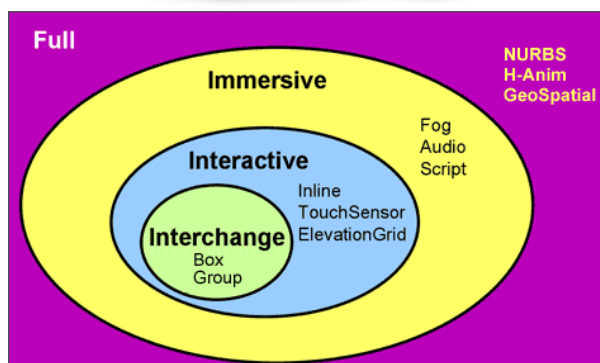
Z pohľadu tvorcu programu sú tu podstatné dve veci: X3D prehliadač načítava pasívne modely z X3D súborov. Prostredníctvom API prehliadača môže k objektom scény pristupovať iná aplikácia (vytvorená napr. v C, C++ alebo Java) a manipulovať s nimi. Používateľ pracuje len s prehliadačom, jeho akcie (pr.: kliknutie myšou) vyvolávajú udalosti (events), ktoré cez API môžu vyvolať reakciu externého kódu.

X3D nedefinuje žiadne fyzické zariadenia. Nepozná napríklad myš alebo 2D monitor (a teda nepracuje s rozlíšením). O súvisiacu funkcionality (napr. interakcia pomocou myši) sa stará prehliadač respektíve programátor aplikácie.

2.4.1.3 Profily X3D

X3D používa tzv. profily, ktoré sa dajú chápať ako sprístupnenie určitej podmnožiny z celkových možností X3D. Tvorcovia profilov sa takto snažili o čo najväčšiu kompatibilitu medzi prehliadačmi. Producent prehliadača nemusí podporovať všetky funkcie X3D, ale sa sústreďuje len na korektnú implementáciu určitej podskupiny funkcií. To môže byť veľmi výhodné napríklad pri X3D browseroch v mobilných zariadeniach. Platforma definuje štyri základné profily, od najobmedzenejšieho *Interchange profile* až po najvýkonnejší *Full profile*.





Obr. 10 Hierarchia profilov X3D

2.4.1.4 Vývoj v X3D

Základný postup vytvorenia aplikácie na platforme X3D má tri kroky, ktoré sa v zásade líšia len použitými nástrojmi.

Najprv je potrebné namodelovať 3D svet. Web 3D Consortium neposkytuje žiadny vlastný program, no väčšina súčasných nástrojov ako 3DS Max, Maya či Google SketchUp dokáže exportovať scény do X3D súborov.

Druhým krokom je pridanie interaktivity do namodelovanej scény. Samotné prehliadače ponúkajú niektoré základné druhy manipulácie ako otáčanie, približovanie/vzdďalovanie celej scény. Ak je táto funkcionlita postačujúca, tento krok možno vlastne vynechať. Consortium vyvinulo editor X3D-Edit, v ktorom sa zmeny robia najmä zásahom do XML syntaxe, no výsledok úprav možno vidieť vo vnorenom prehliadači Xj3D [6]. Napríklad cez tzv. TouchSensor a transformácie možno vytvoriť interaktívny model: objekty v scéne reagujú (transformáciou sa mení pozícia, tvar, priesvitnosť...) na akcie používateľa (napr. kliknutie na objekt s definovaným TouchSensorom). [5]

Odlíšny prístup k interaktivite ponúka program WireFusion. Importuje X3D súbory, ktoré musia byť pripravené vopred v niektorom modelovacom nástroji. Výstup má formu spustiteľného jar súboru alebo Java appletu, ktorý sa umiestni na www stránku. Je tak úplne vylúčený X3D browser, na spustenie stačí mať nainštalovanú (dnes už dosť bežnú) Java platformu. Proces pridania funkcionality do scény je jednoduchý a dosť intuitívny. Hlavné skriptovacie okno programu ponúka grafické znázornenie jednotlivých častí scény. Interakcia sa nastavuje prepájaním objektov pomocou orientovaných čiar a dodefinovaním špecifických reakcií (napr. zmena farby objektu, prehratie zvuku). [7]

Posledným krokom je sprístupnenie aplikácie. Pri offline využití stačí, ak má používateľ výsledný X3D súbor a nainštalovaný prehliadač (prípadne .jar súbor a Javu). Pri publikovaní na internetovej stránke sa obsah zapracuje do HTML kódu pomocou tagu object nasledovne:

X3D obsah na webovej stránke:

```
<object data="C:/MyWorld.x3d" type="model/x3d+xml"
height="360" width="300">
```





```
<param name="src" value="C:/MyWorld.x3d"/>
<param name="DASHBOARD" value="FALSE"/>
<param name="SPLASHSCREEN" value="FALSE"/>

<!-- Text, ktory sa zobrazi, ak plugin nie je nainstalovany -->
<div class="noX3dPluginInstalled">
  <h2>Plugin content...</h2>
</div>
</object>
```

V prípade Java appletu sa použije obdobný postup s tagom applet.

X3D obsah na webovej stránke vo forme Java appletu:

```
<applet code = "MyWorldCdeBase.class" jnlp_href = "MyWorld.jnlp",
width = "300" height = "300" />
```

X3D má vlastnosti, ktoré ho predurčujú ako dobrú možnosť na realizáciu projektu Virtuálnej FIIT. Ponúka stabilný výkon, mutiplatformovosť a pohodlnú prácu s existujúcimi nástrojmi. Konkrétny postup implementácie si predstavujem ako kombináciu nástrojov Google SketchUp (dostupný zadarmo v základnej verzii) na modelovanie a WireFusion na obohatenie o interakciu a vytvorenie Java aplikácie. Pri takomto riešení máme istotu, že sa výsledok bude správať rovnako na rôznych systémoch, čo pri X3D prehliadačoch nie je zaistené.

Jedinou nevýhodou je nutnosť naučiť sa pracovať s kompletne novou technológiou, ktorá ponúka odlišný pohľad na interaktívnu trojrozmernú grafiku, keďže žiadny z členov tímu realizujúceho projekt Virtuálnej FIIT nemá s touto platformou žiadne skúsenosti.

2.5 Loadovanie priestorov pre systém Virtuálnej FIIT

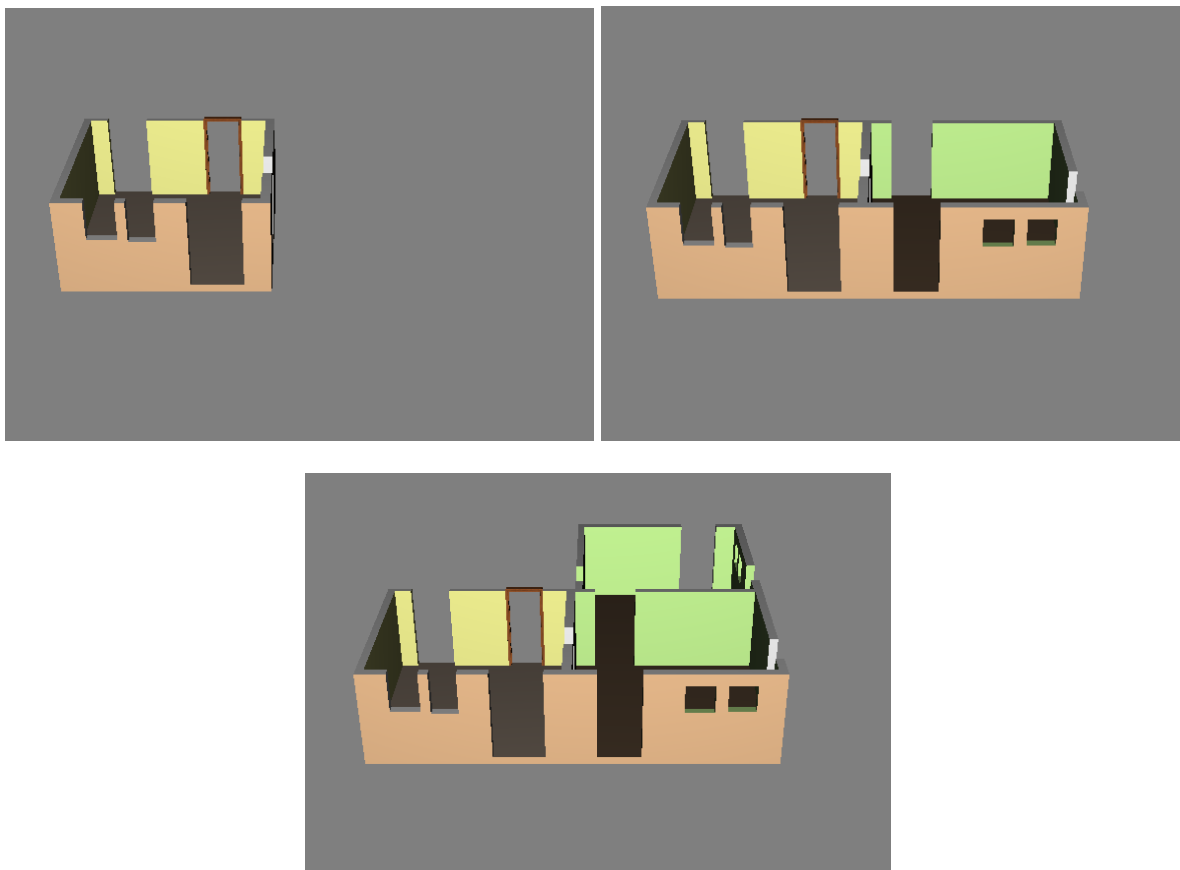
Načítanie modelu (súboru formátu o3dtgz) sa v O3D realizuje pomocou funkcie *loadScene*. Táto funkcia má tieto parametre:

- **client** – klient O3D objektu
- **pack** – balík obsahujúci všetky O3D objekty (sem sa načíta scéna), umožňuje riadenie práce s nimi
- **transform** – je uložený pod transformom rodiča
- **URL** – cesta a názov súboru, ktorý chceme načítať
- **callback** – funkcia, ktorá je spätne volaná po načítaní, obsahuje *pack*, *parent* a *exception* (výminku), ktorá má hodnotu *null*, ak načítanie bolo úspešné
- **opt_options** – dodatočný voliteľný parameter





Načítanie objektu je vhodné realizovať po nejakej udalosti, ja som si zvolil kliknutie na scénu. V Google SketchUp-e som namodeloval ako príklad 3 izby, ktoré sa po každom kliknutí postupne načítajú. Dôležité je uvedomiť si, kde sa objekty vykreslia vzhľadom na počiatok súradnicovej osi (bod 0,0,0). Ja som objekty v modelovacom nástroji vytvoril tak, aby mi ich pozície presne v 3D scéne sedeli. Tiež som sa pokúsil v prvej izbe vytvoriť komponent dvere, ktorý sa ale nenačítal (toto je tiež potrebné vyriešiť).



Obr. 11 Postupné načítanie izieb.





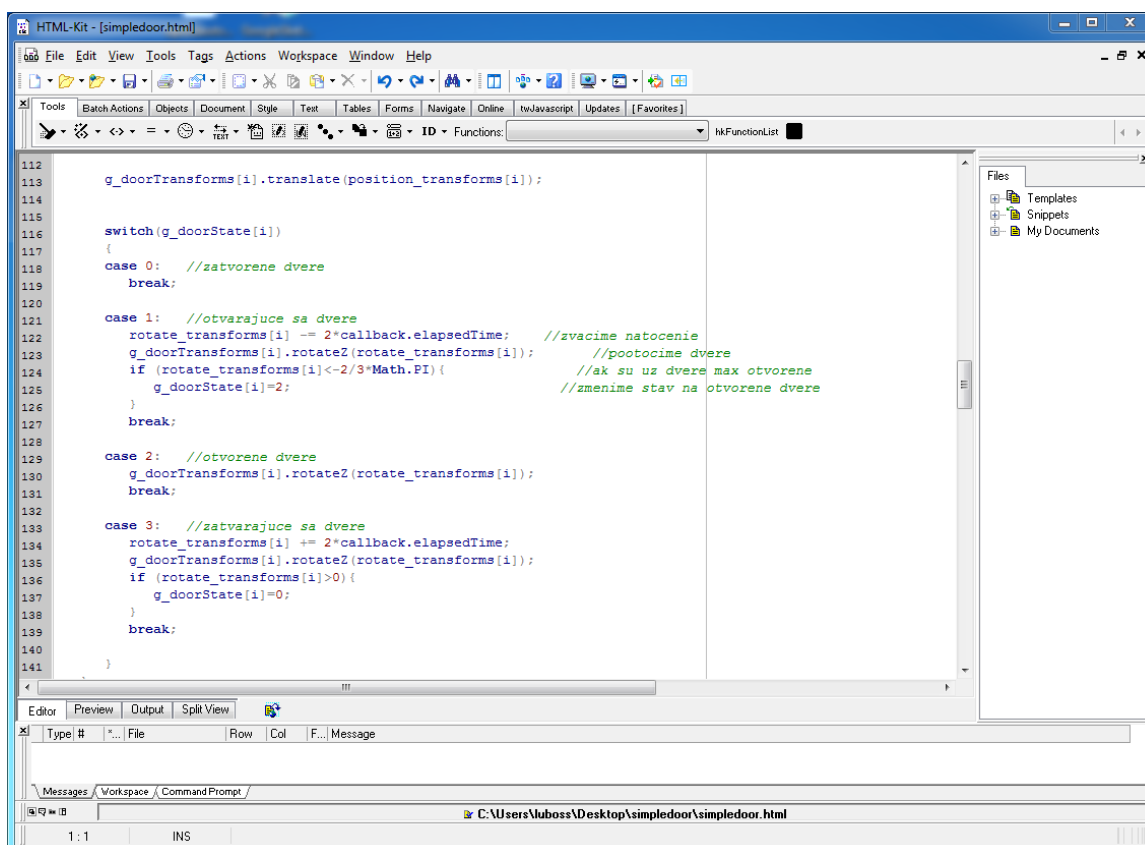
2.6 Analýza vývojového prostredia Virtuálnej FIIT

V tejto kapitole sú opísané programy, ktoré využijeme pri tvorbe O3D projektu.

2.6.1 Editory vývojového prostredia

HTMLKit

HTMLKit je rozšíriteľný editor podporujúci množstvo nielen webových formátov. Jednou z hlavných výhod, ktoré tento editor poskytuje, je zvýrazňovanie syntaxe aj viacerých formátov v jednom súbore. V našom projekte budeme pracovať hlavne s JavaScriptom obsiahnutým v HTML stránke, kde túto možnosť využijeme. HTMLKit obsahuje aj asistenciu pri písaní kódu, kde však chýba podpora pre O3D plugin. Ďalšou funkciou, ktorú editor ponúka, je rýchly náhľad vytvorenej stránky. Stránka je zobrazená na samostatnej záložke programu, kde je možné na zobrazenie vybrať medzi enginom Explorer alebo Gecko. Vytváraný kód netreba pred samotným náhľadom ukladať, čo môže byť výhodou, ak v kóde spravíme zmeny väčšieho rozsahu, s ktorými si nie sme úplne istí.



Obr. 12 Obrazovka programu HTMLKit





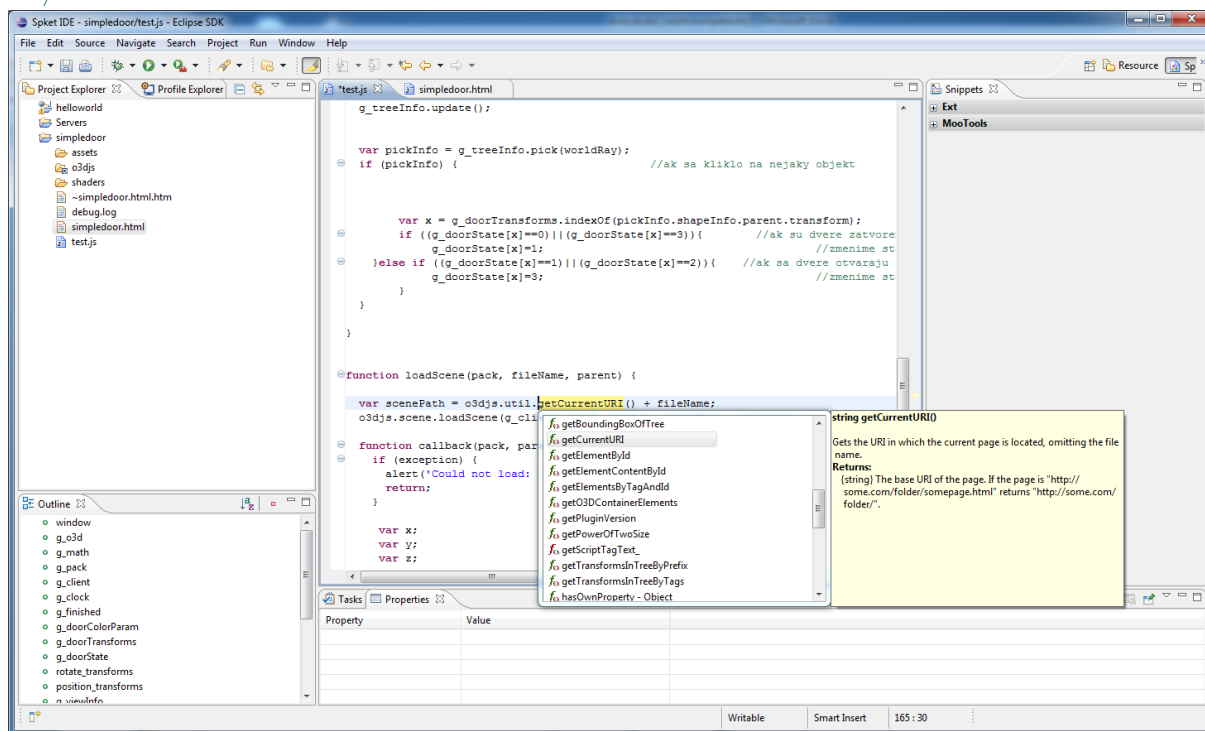
Eclipse

Eclipse je komplexné vývojové prostredie, ktorého možnosti je možné ešte rozšíriť množstvom ďalších pluginov. Pre náš projekt je najvhodnejšie použiť rozšírenia Web Developer Tools, Subversive SVN a Spket. Web Developer Tools obsahuje nástroje pre tvorbu webových stránok a aplikácií s podporou pre zvýrazňovanie syntaxe. Pomocou Subversive SVN je možné, ako názov napovedá, pripojiť sa na SVN server a spravovať verzie programu. Plugin Spket poskytuje asistenciu pri písaní kódu, ktorá je však v našom projekte obmedzená len na funkcie z externých JavaScript súborov (bez funkcií, ktoré sú obsiahnuté priamo v O3D plugine). Keďže budeme pracovať s množstvom externých JavaScript súborov (priechinok o3djs), je aj takto obmedzená asistencia výhodou. Pre jej fungovanie je potrebné na začiatku súboru importovať externé JavaScript súbory. Napríklad:

```

/*
 * @include "/simplifiedoor/o3djs/base.js"
 */

```



Obr. 13 Obrazovka programu Eclipse

2.6.2 3D modelovanie

Google Sketchup

Je editor 3D modelov, ktorý sa vyznačuje najmä svojím jednoduchým používateľským rozhraním. Jeho natívnym formátom pre ukladanie modelov je vlastný formát „.skp“. Umožňuje však aj export do iných formátov, ktorých zoznam je možné rozšíriť ďalšími



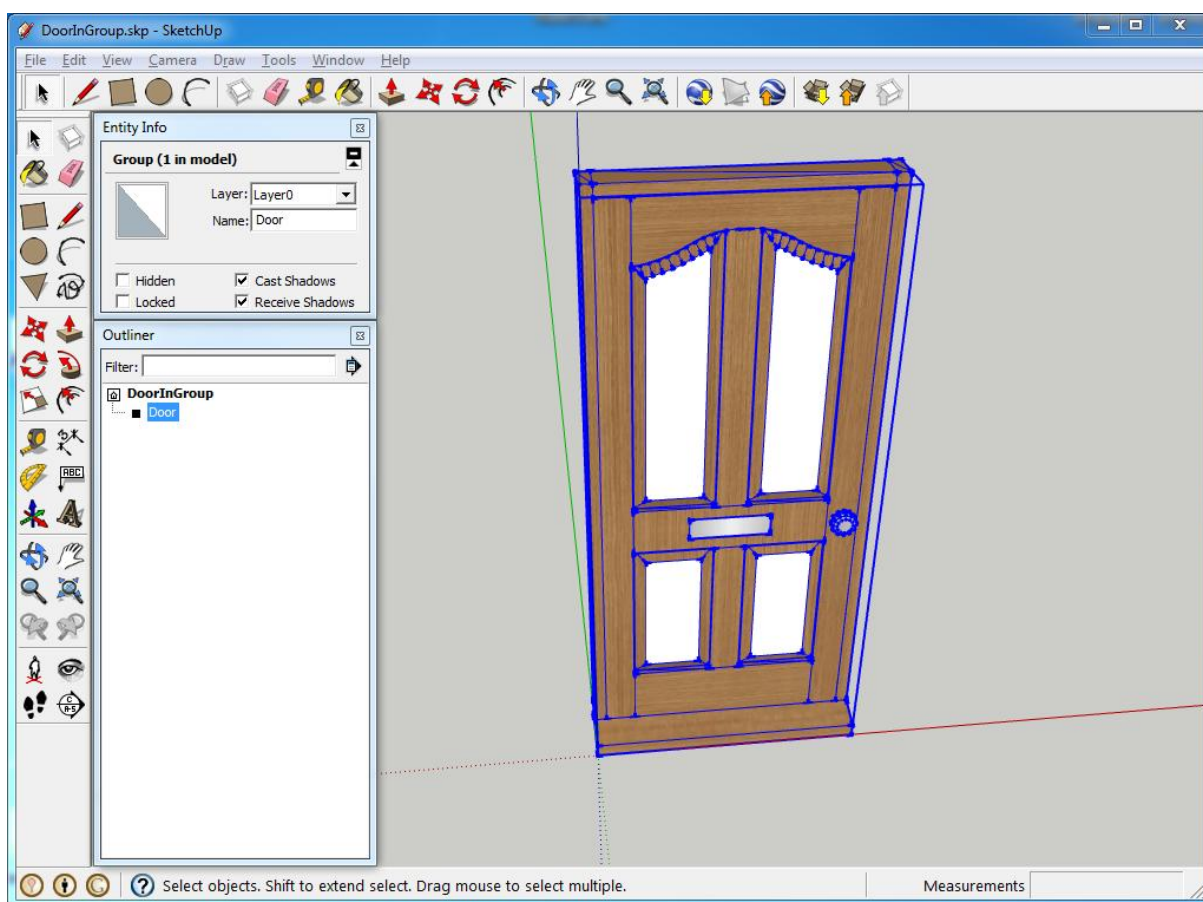


pluginmi. Podobne je možné rozšíriť aj jeho funkcionalitu. V našom projekte využijeme najmä export do formátu COLLADA (prípona „.dae“), ktorý sa po konverzii na formát JSON (prípona „.o3dtgz“) dá importovať do O3D projektu.

Pri importe sme sa však stretli s nasledujúcimi problémami:

Google Sketchup 7.1 – po konverzii do „.dae“ formátu sa stratili názvy komponentov. (boli premenované na „instance_0“, „instance_1“ atď.), čo značne komplikuje prácu s nimi v prostredí O3D. Ďalším problémom je, že ak je model rozdelený na komponenty, nie je ho možné v O3D zobraziť.

Google Sketchup 7.0 – táto verzia programu neexportovala modely do formátu „.dae“ priamo, ale podporovala export do formátu „.kmz“, čo je vlastne skomprimovaný „.dae“ súbor. V tomto prípade ostali názvy komponentov zachované aj v „.dae“ súbore, ale po konverzii do formátu JSON (prípona „.o3dtgz“), sa názvy komponentov v O3D projekte stratili. Na rozdiel od novej verzie však bolo možné model zobraziť aj keď bol rozdelený na komponenty.



Obr. 14 Obrazovka programu Google Sketchup

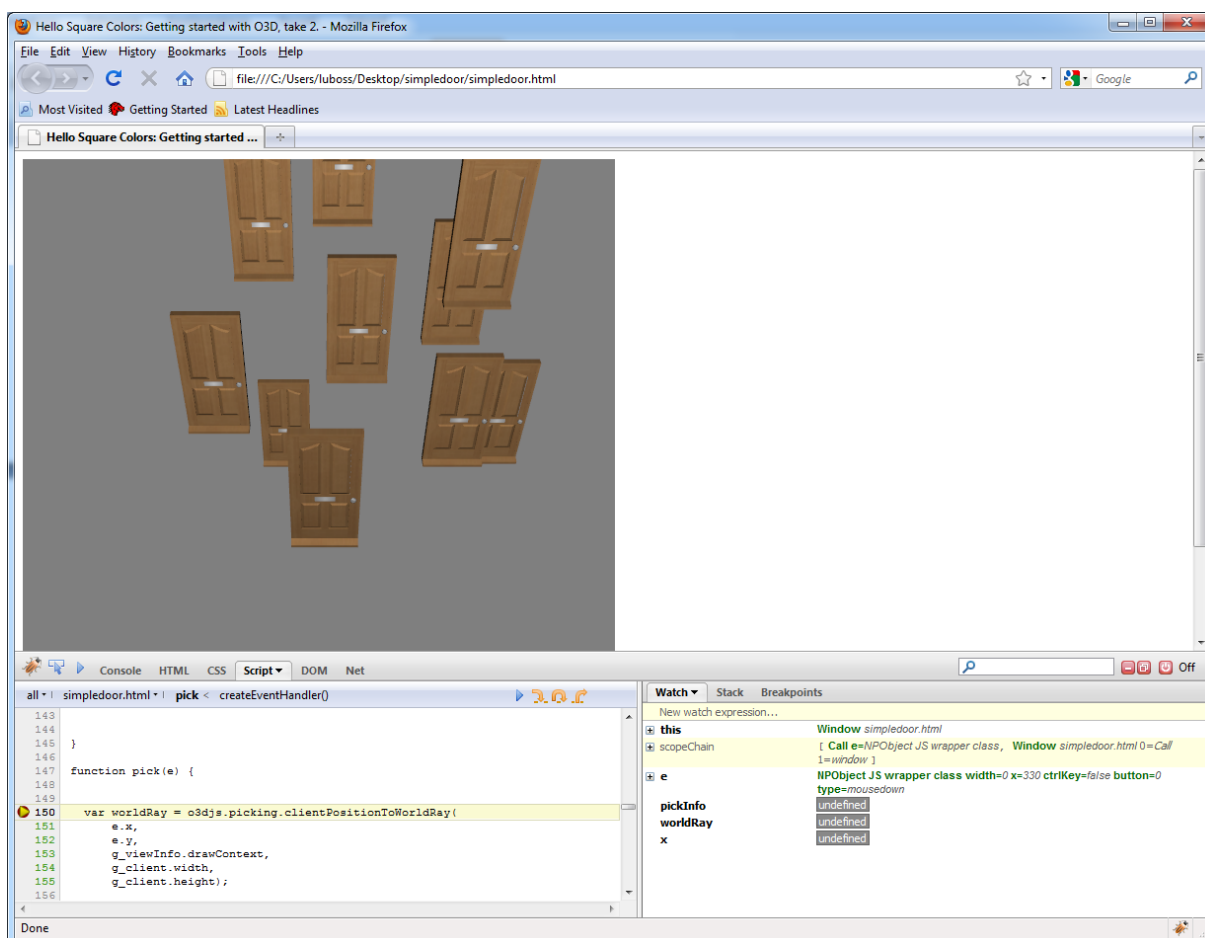




2.6.3 Debugovanie Javascript kódu

Firebug

Firebug je plugin do webového prehliadača Firefox. Medzi jeho najdôležitejšie funkcie patrí vykonávanie O3D programu po jednotlivých príkazoch, nastavovanie breakpointov a sledovanie premenných. Oproti iným debuggerom, ktoré sme skúsili sa vyznačuje najmä stabilitou a jednoduchosťou použitia.



Obr. 15 Obrazovka pluginu Firebug v prehliadači Firefox

2.6.4 Existujúce riešenia zobrazenia 3D scén

The Forbidden City: Beyond Space & Time

The Forbidden City [10] je multiplatformová desktopová aplikácia vytvorená spoločnosťou IBM, ktorá používateľom ponúka široké možnosti v prehliadaní virtuálneho Zakázaného mesta. Aplikácia má náučný a informačný charakter, pričom obsahuje aj prvky





MMORPG hier. Návštevník je po spustení aplikácie oblečený v dobovom kostýme, môže sa pohybovať po meste ako aj priestormi budov, rozprávať s inými návštevníkmi, prehliadať a skúmať artefakty, zúčastňovať sa prehliadok korešpondujúcich s významnými historickými udalosťami a príbehmi. Jeho pozícia spolu s vyznačenou cestou, ktorou sa pohyboval je vyznačená na zmenšenej mape mesta. Aplikácia je vytvorená programami WebSphere Application Server, Tivoli, ESB (Message Broker), DB2 Viper a IBM BladeCenters.

Aplikácia má dobre riešený spôsob zobrazovania informácií o budovách (pomocou NPC postáv) a prepínanie medzi módmi zobrazenia (self, first person, third person a bird's eye), ktorých princíp by sa dal využiť pri tvorbe virtuálnej FIIT.

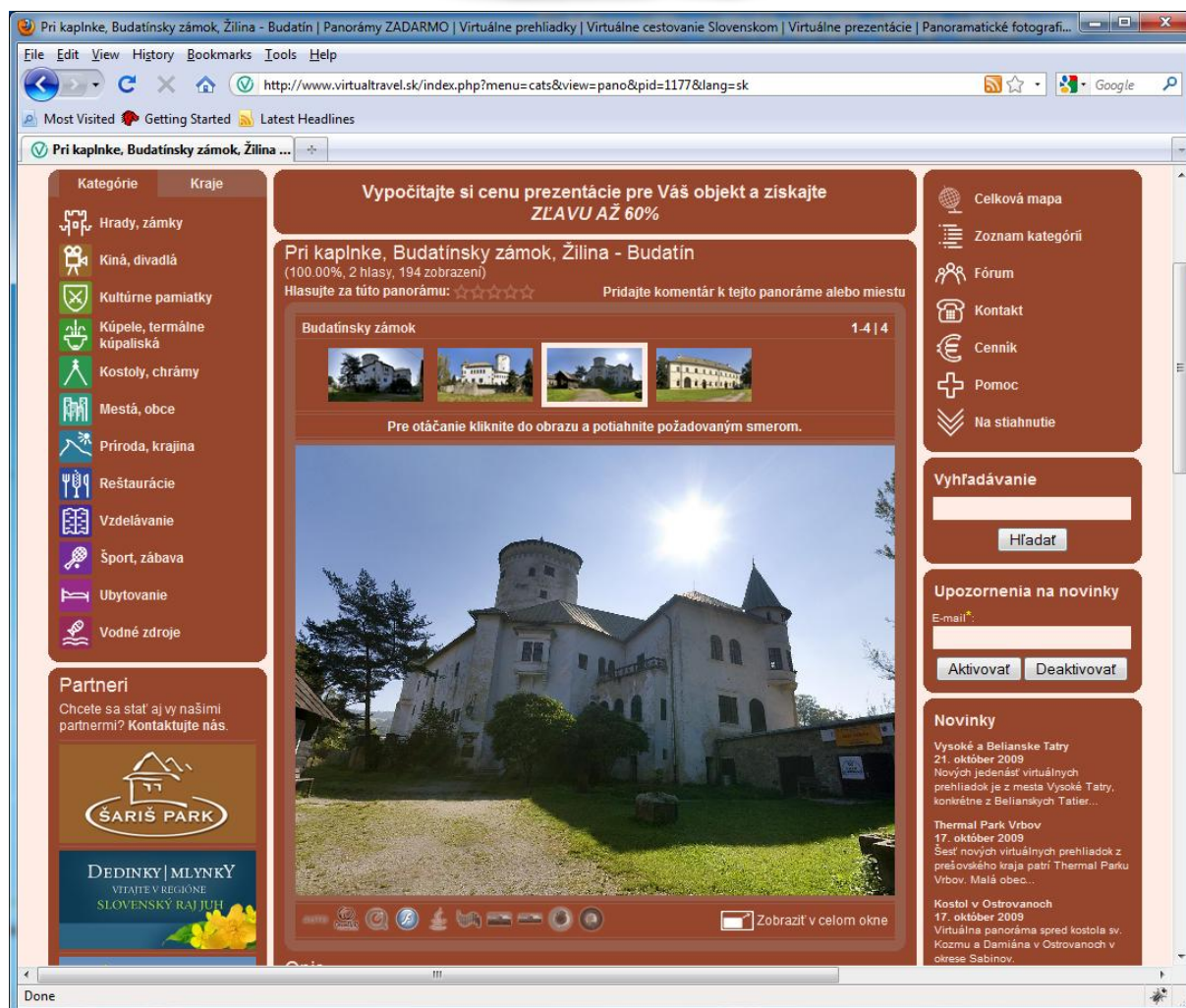


Obr. 16 Obrazovka aplikácie The Forbidden City: Beyond Space & Time

Webový portál virtualtravel.sk

Tento webový portál [11] neobsahuje 3D modely budov, ale panoramatické fotografie. Tie je v prehliadači možné otáčať o 360 stupňov a približovať, čím sa navodzuje dojem 3D obrazu. V súčasnosti je to asi najpoužívanejší spôsob zobrazovania 3D scén, keďže bežne rozšírenými technológiami, akou je napríklad Flash, nie je možné zobraziť skutočné 3D modely vo webovom prehliadači kvôli výkonnostným problémom.



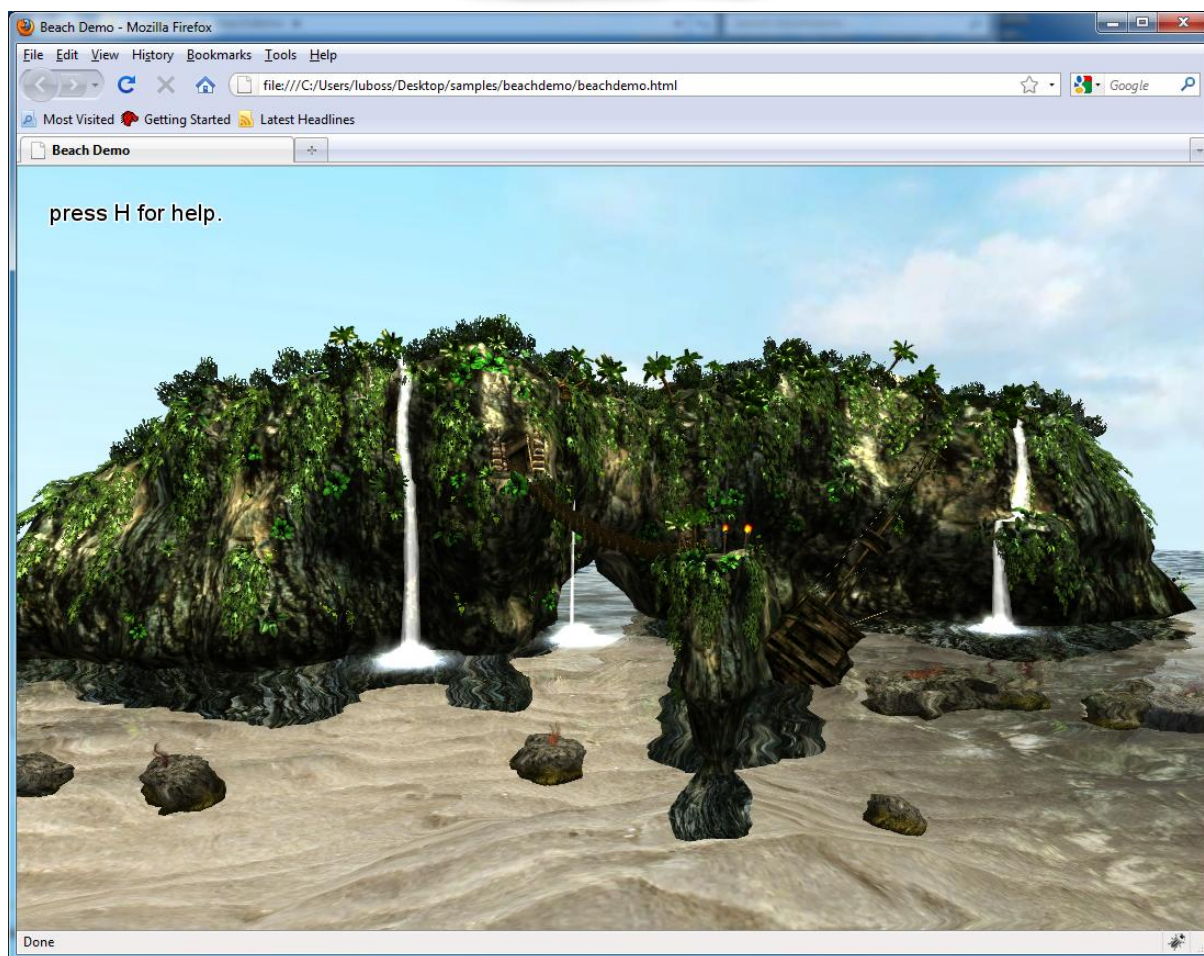


Obr. 17 Obrazovka stránky virtualtravel.sk

Beach Scene

Beach Scene [12] je demo, ktorým Google prezentoval O3D API pri jeho vydaní. Služi ako príklad toho, čo je v O3D možné vytvoriť. Demo obsahuje plnohodnotnú 3D scénu obsahujúcu model ostrova s množstvom odleskov (voda), častíc (vodopády) a detailov, v ktorej sa dá pohybovať. Google si na jeho tvorbu najal profesionálnych grafikov zo spoločnosti Crazy Pixel. Keďže Beach Demo je distribuované ako open source, je možné použiť časti jeho kódu pri tvorbe projektu (napr. načítavanie, meranie FPS, bočný panel, pohyb po scéne ap.)





Obr. 18 Demo Beach Scene

2.6.5 Prehliadače pre systém Virtuálna FIIT

Okrem webového prehliadača Firefox bude nami vytváraný projekt podporovať aj prehliadače Internet Explorer a Google Chrome.

Internet Explorer 8.0

Pre tento prehliadač bude zabezpečenie podpory asi najväčšou výzvou. I keď sa vo verzii 8.0 podpora štandardov značne zlepšila, už na začiatku vývoja sa stretávame s menšími problémami (napríklad chýbajúca funkcia IndexOf pre polia). Podporu chceme zabezpečiť najmä preto, lebo Internet Explorer má najvyšší podiel spomedzi webových prehliadačov. Druhým dôvodom je jeho integrácia v editore HTMLKit, kde je používaný na zobrazovanie náhľadov, čo značne zrýchľuje proces vývoja.





Google Chrome

Tento prehliadač má spomedzi plánovaných podporovaných najmenší podiel na poli webových prehliadačov. Jeho podporu chcem zabezpečiť z dôvodu, že na interpretáciu JavaScriptu využíva engine V8. Tento engine je zabudovaný aj v O3D plugine, ktorý ho umožňuje použiť aj v ostatných webových prehliadačoch namiesto štandardného. Po jeho vynútení by mal teoreticky náš O3D projekt fungovať pod všetkými prehliadačmi.

2.7 Analýza modelovacieho nástroja 3DS Max

3DS Max je komerčná aplikácia spoločnosti Autodesk. Je to nástroj na tvorbu pokročilých trojrozmerných modelov a animácií. Používa vlastný proprietárny formát (súbory s príponou .max), no dokáže importovať a exportovať veľké množstvo aj otvorených formátov.



Funkcionalita bola testovaná na 3DS Maxe vo verzii 2010.



Práca s aplikáciou

Hlavné okno programu v pôvodnom nastavení je rozdelené na štyri časti (pohľady z troch strán a perspektíva). Rozdelenie sa dá jednoducho zmeniť, napríklad na jediné okno v perspektíve ako v Google SketchUp, kliknutím na malý krížik v ľavom hornom rohu a výberom Configure..., v záložke Layout. V pravom hornom rohu je zobrazená malá kocka, pomocou nej sa otáča scéna v perspektíve po kliknutí a pohybe myšou so stlačeným ľavým tlačidlom. Scéna sa posúva pri stlačení koliesku myši.

Najčastejšie používaný postup pri modelovaní budovy FIIT bude vytváranie obdĺžnikových plôch, ich otáčanie a spájanie do jedného celku a aplikovanie materiálov.

Vytvorenie plochy: obdĺžniky (a ostatné tvary) sa vyberajú v pravom paneli v záložke Create. Pre plochu je určené tlačidlo Plane. Po zvolení sa kliknutím v dvoch miestach v hlavnom okne určia dva body uhlopriečky obdĺžnika, ten sa vykreslí vždy rovnobežný s osou X. Vytvorenej ploche treba určiť správne rozmery. To sa robí tiež v pravom paneli v záložke Modify. 3DS Max nepoužíva konkrétne jednotky (na rozdiel napríklad od SketchUp), meria v abstraktných tzv. units. Ak sa bude modelovať Virtuálna FIIT pomocou Maxu, je potrebné stanoviť si pomer jednotiek (napríklad 1 unit = 1 meter).


Po určení rozmerov plochu treba správne otočiť. Slúži na to tlačidlo Select and Rotate () v hornom paneli. Po kliknutí možno otáčať zvolený objekt ťahom myši. Ešte predtým sa treba uistiť, že tlačidlo Angle Snap Toggle () je zapnuté, čo umožní otáčať plochu o násobky 5 stupňov.

Spájanie plôch: správne orientované plochy treba ďalej spojiť do jedného celku. Predtým je vhodné ich k sebe priblížiť (Select and Move v hornom paneli ()), program potom pomôže pri úplnom zarovnaní. Kliknutím na Align () a zvolením dvoch objektov sa

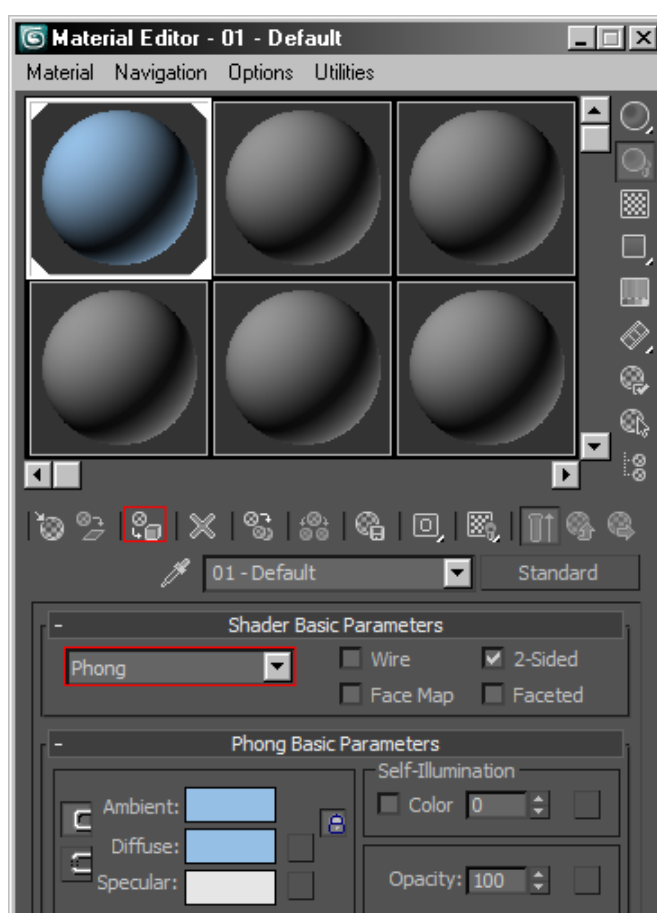




vyvolá dialóg, v ktorom sa určí ktoré osi oboch objektov sa majú zarovnať. Objekty zarovnané do jedného tvaru sa spoja do jedného objektu tak, že v záložke Modify (pravý panel) niektorého z objektov sa klikne na Attach a zvolí sa druhý objekt. Novovzniknutý útvar je vhodné pomenovať (pole na vrchu záložky).

Aplikovanie materiálu: materiál objektu (farba alebo textúra povrchu) sa vyberá v dialógu Material Editor (Obr. A), vyvolaný pomocou rovnomenného tlačidla v hornom paneli () . Okno ponúka množstvo nastavení pre materiál, pre potreby neskoršieho korektného exportu je podstatné v roletovom menu v časti Shader Basic Parameters zvoliť hodnotu Phong. Samotná aplikácia materiálu na povrch objektu sa robí pomocou tlačidla Assigne material to selection (na Obr. 19 v červenom štvorci).

Takýto postup je len jeden spôsob tvorenia 3D modelov, AutoDesk 3DS Max vo všeobecnosti ponúka veľké množstvo možností a nástrojov. Ich použitím možno vytvoriť aj veľmi komplikované realistické 3D prostredia.



Obr. 19 Material Editor dialóg

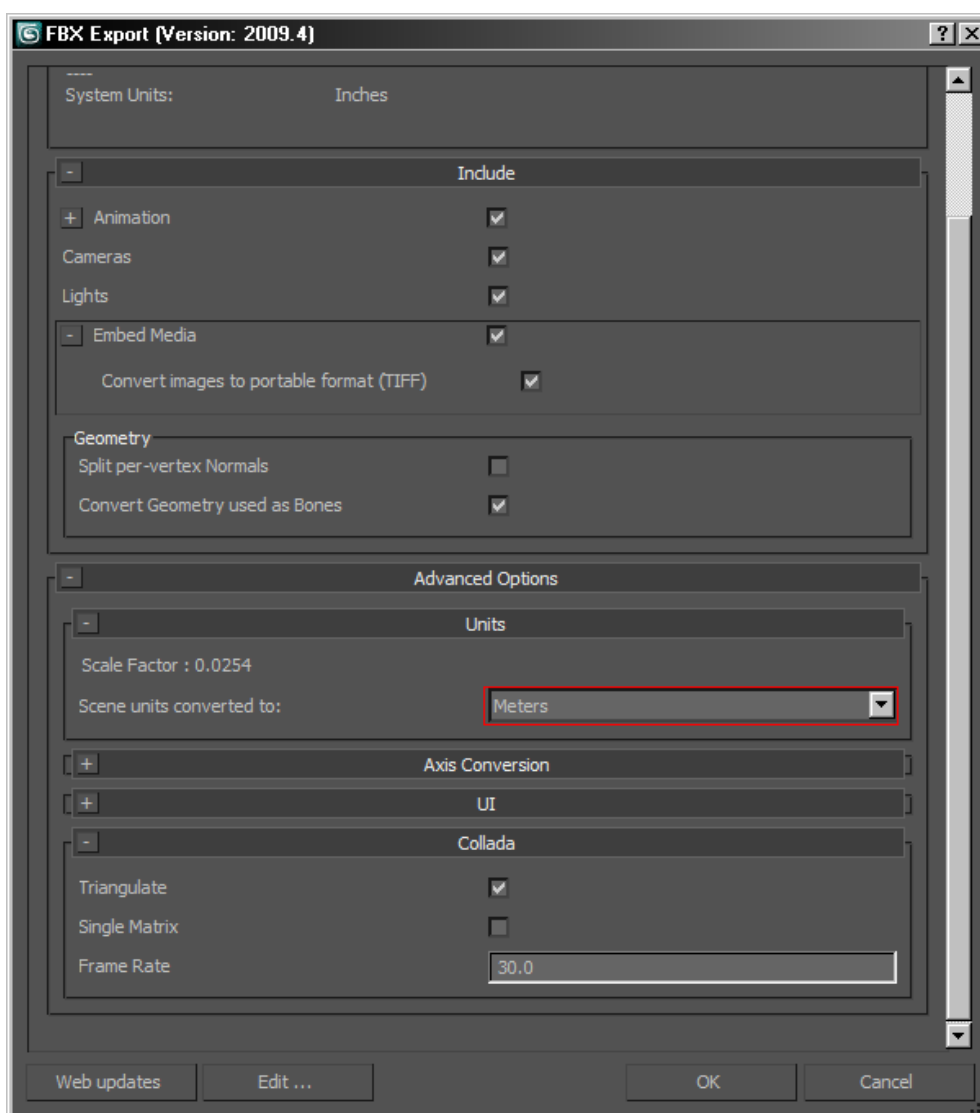




Export do COLLADA formátu

Aby bolo možné použiť modely a scény vytvorené v 3DS Maxe v O3D aplikácii, je potrebné ich uložiť v kompatibilnom formáte. Program podporuje export do formátu COLLADA (prípona .dae). Pred samotným exportom treba potvrdiť dialóg sumarizujúci parametre konverzie (Obr. 17). V ňom je potrebné určiť, ako sa preložia units na jednotky dĺžky.

Export bol otestovaný na viacerých jednoduchších modeloch a pri použití správneho shader algoritmu (Phong) fungoval korektne, všetky dáta boli zachované aj po ďalšej konverzii do O3D formátov.



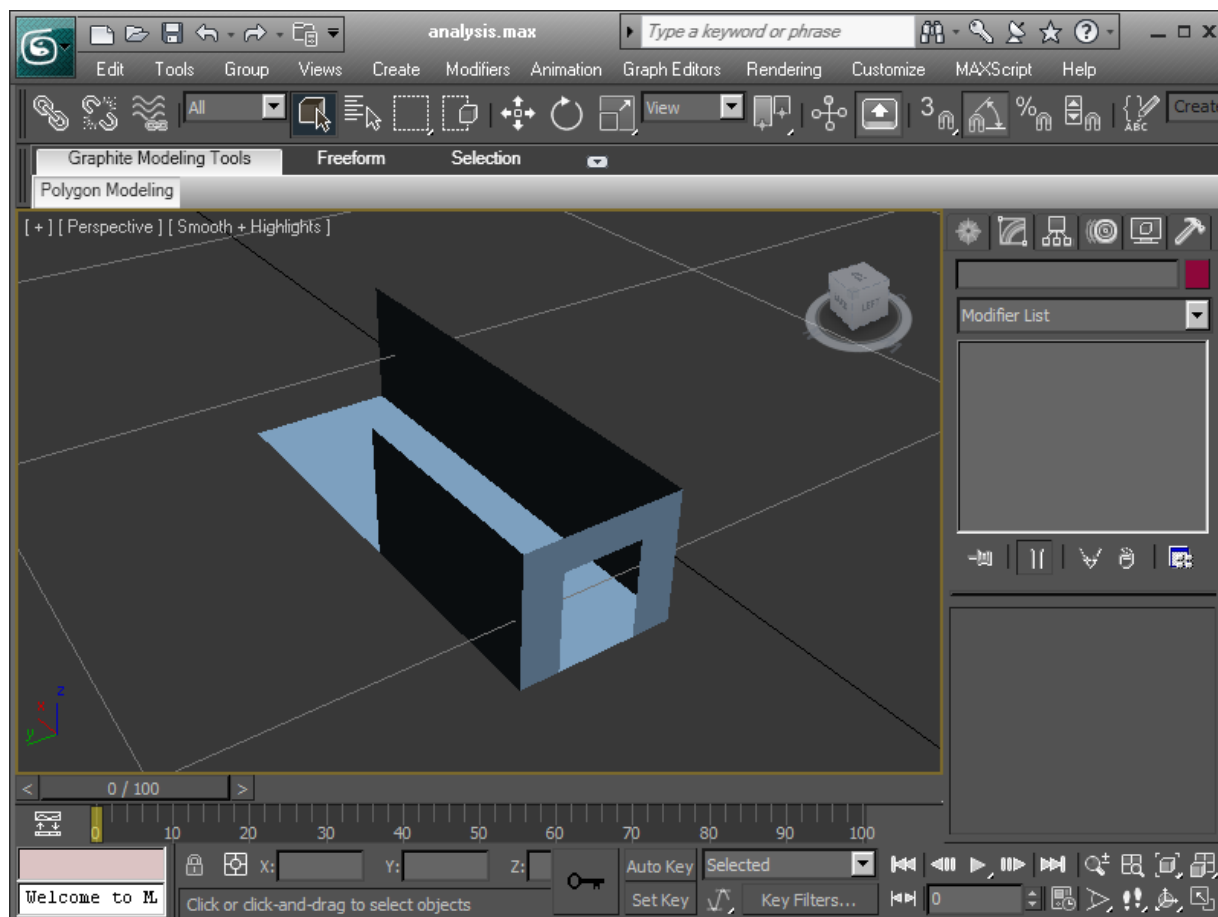
Obr. 20 Export do COLLADA formátu

Zhrnutie





AutoDesk 3DS Max je veľmi výkonný a komplexný nástroj, ktorý plne vyhovuje požiadavkám projektu Virtuálnej FIIT. Jeho veľkou výhodou je schopnosť korektnej konverzie do COLLADA súboru. Nevýhodou je len jeho komplexnosť a komerčný charakter.



Obr. 21 Okno programu 3DS Max.

2.8 Zhrnutie analýzy problematiky systému

Analyzovaním a zhodnotením dostupných nástrojov a prostriedkov sme dospeli k zloženiu, ktoré bude použité pri vývoji aplikácie Virtuálnej FIIT.

Budeme vytvárať vlastné modely budovy, keďže prostredia s predošlých prác majú viaceré nedokonalosti, ale najmä pre nekompatibilitu a problémy pri konverzii ich formátoch na tie, ktoré budeme využívať v našom projekte. Pri modelovaní uprednostníme program 3DS Max pre jeho pokročilé schopnosti a kompatibilitu s ostatnými prostriedkami vývoja.

Na implementáciu podstatnej funkcionality použijeme O3D ako rapídne sa vyvíjajúcu a dobre zdokumentovanú technológiu, ktorej primárne zameranie je prostredie internetu.

Na samotný vývoj bude použitá kombinácia prostredí Eclipse, Firefox a Firebug, ktorá sa v súčasnosti javí ako najlepšie spojenie pre takýto druh projektu.





3 ŠPECIFIKÁCIA RIEŠENIA

Tretia kapitola zachytáva špecifikáciu požiadaviek a ich riešenie. Najskôr boli vytvorené požiadavky na systém, ktoré vychádzali zo zadania úloh a našich tímových stretnutí. Následne boli tieto prvotné požiadavky došpecifikované funkcionalitou a podrobnejšími vlastnosťami, závislosťami. Vytvorené boli aj prípady použitia pomocou UML.

3.1 Špecifikácia požiadaviek na systém Virtuálnej FIIT

V tejto časti dokumentu sa popisuje špecifikácia požiadaviek na vyvíjaný 3D model budovy FIIT. Obsahuje ich analýzu a vyvodené závery, ktoré sú prenesené na grafický návrh a na funkcionalitu modelu. Ďalej opisuje prípady použitia modelu používateľom, jednotlivé role používateľov, do ktorých bude používateľ môcť vstupovať.

3.1.1 Základné požiadavky na systém a jeho vývoj

Základné požiadavky na systém vyplývajú z požiadaviek zadávateľa, ktorými sú:

- Vytvorený model má vychádzať z dobre analyzovaného problému tvorby 3D budov spolu s analýzou nástrojov určených na 3D vizualizácie a interakciu.
- Vytvorený model má byť skutočný 3D model novej budovy FIIT a má vychádzať z reálnych architektonických plánov novej budovy.
- Model má mať implementované interaktívne časti ako napríklad:
 5. otváranie a zatváranie dverí,
 6. informačné tabule,
 7. možnosť zadania otázky,
 8. nájdenie najlepšej cesty,
 9. navigovanie používateľa po navrhnutej ceste,
 10. informácie o navštívených miestach.
- Model má obsahovať databázu údajov, navrhnutú vhodne a efektívne.
- Riešenie 3D modelu má byť optimalizované tak, aby ho používateľ mohol používať aj cez jednoduché webové rozhranie pomocou najčastejšie používaných internetových prehliadačov (Opera, Internet Explorer, Mozilla Firefox a pod.).
- Model má byť riadne otestovaný.





- Výsledný 3D model má byť implementovaný takým spôsobom, aby jeho plynulé fungovanie nekladlo priveľké hardvérové nároky (prezentácia modelu má byť plynulá).
- Model má podporovať pripojenie niekoľkých používateľov naraz.
- Model by mal byť navrhnutý prehľadne a efektívne s jednoduchou navigáciou a s čo najjednoduchším intuitívnym ovládaním.

3.1.2 Hardvérové a softvérové požiadavky na systém

Zo základnej požiadavky „Riešenie 3D modelu má byť optimalizované tak, aby ho používateľ mohol používať aj cez jednoduché webové rozhranie pomocou najčastejšie používaných internetových prehliadačov vyplývajú nasledujúce hardvérové požiadavky:

- Model budovy má byť umiestený na serveri umožňujúcom klient - server spojenie.
- Model má podporovať desiatky klientskych pripojení.
- Vzhľadom na to, že 3D model je spracovaný na klientskom počítači musí byť klient vybavený postačujúcim hardvérom (samostatná GPU jednotka alebo postačujúca integrovaná GPU).
- Systém je dostupný cez internetový prehliadač. Preto ak je to potrebné, musí mať prehliadač inštalovaný zásuvný modul pre prezeranie 3D súborov podporujúcich webové rozhranie.

3.1.3 Charakteristika používateľov systému

Vytvorený systém podporuje role nasledujúcich používateľov:

- **Administrátor** – spravuje systém, obnovuje zdrojové súbory (model 3D budovy FIIT), je administrátorom databázy, ktorú pravidelne zálohuje a aktualizuje.
- **Užívateľ** – je osoba, pre ktorú je produkt určený. Vyberá si z možností zobrazenia, vyhľadáva cesty a prezerá zobrazené informácie. Využíva funkcionality systému, prezerá si systém.

3.1.4 Požiadavky na funkcionality systému

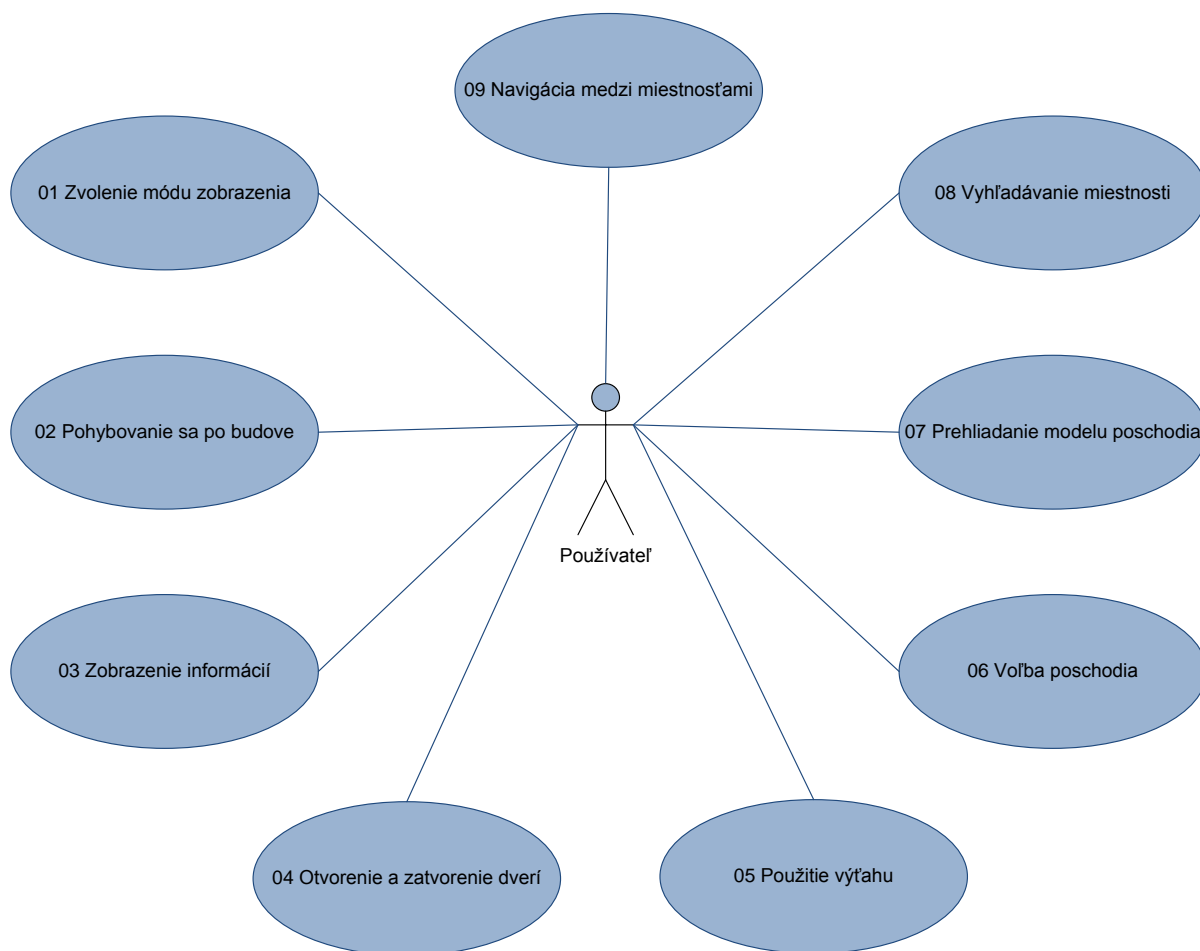
Vychádzajúc z kapitoly C.1.1 a podľa požiadaviek zadávateľa je celková funkcionality zameraná na používateľa. Práve ten je so systémom v najužšom kontakte a s ním súvisí väčšina interakcií.





3.2 Diagram prípadov použitia

Na obr. 22 je znázornený diagram prípadov použitia, ktorý poskytuje pohľad na služby, ktoré sú dostupné pre bežného používateľa systému virtuálnej budovy FIIT.

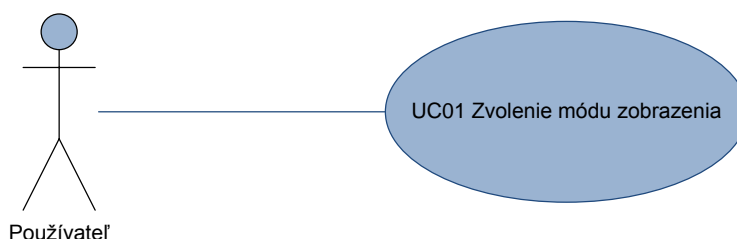


Obr. 22 Prípad použitia pre bežného používateľa





3.2.1 UC01 Zvolenie módu zobrazenia



Obr. 23 Prípád použitia „Zvolenie módu zobrazenia“

Informačný systém umožňuje používateľovi vybrať si z troch módov zobrazenia virtuálnej budovy FIIT, ktoré sú nasledovné: „3D“, „Prehliadanie“ a „Navigácia“. Voľba týchto módov je zobrazená v hlavnom okne spolu s virtuálnym modelom a je umiestnená v ľavom hornom rohu. Po výbere módu používateľom sa zobrazí príslušná voľba zobrazenia modelu a informačného panelu. Výber módu „3D“ znamená, že sa používateľ nachádza v modeli z pohľadu prvej osoby a pôsobí naňho gravitácia, pričom má dostupné prípady použitia 02 Pohybovanie sa po budove, 03 Zobrazenie informácií, 04 Otvorenie a zatvorenie dverí a 05 Použitie výťahu. Pri módoch „Prehliadanie“ a „Navigácia“ používateľ vidí celý model poschodia a nepôsobí naňho gravitácia, pričom v oboch má dostupné prípady použitia 03 Zobrazenie informácií a 07 Prehliadanie modelu poschodia. V móde „Prehliadanie“ môže ešte využiť prípad použitia 06 Voľba poschodia a v móde „Navigácia“ 08 Vyhľadávanie miestnosti a 09 Navigácia medzi miestnosťami. V žiadnom móde sa nedá prechádzať cez steny.

identifikátor	UC01		
názov	Zvolenie módu zobrazenia		
opis	Používateľ si môže zvoliť jeden z 3 módov zobrazenia modelu		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Nie sú		
výstupné podmienky	Používateľ má v hlavnom okne zobrazený vybraný mód modelu		
používatelia	Bežný používateľ		

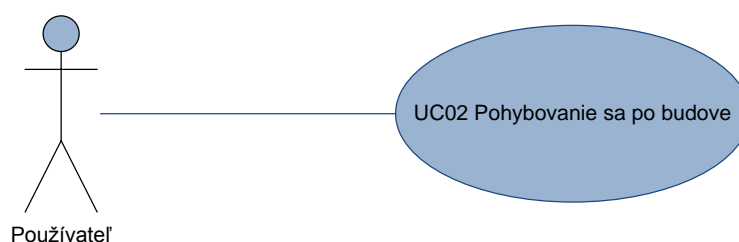




základná postupnosť	krok	činnosť
	1	Systém dáva používateľovi možnosť výberu z 3 módov zobrazenia
	2	Používateľ klikne na jeden z nich („3D“, „Prehliadanie“ alebo „Navigácia“)
	3	Ak bol zvolený „3D“ mód, tak systém zobrazí používateľovi model z pohľadu prvej osoby
	4	Pri móde „3D“ môže používateľ ďalej použiť UC02, UC03, UC04 a UC05
alternatívna postupnosť	krok	činnosť
	3a	Ak bol zvolený mód „Prehliadanie“ alebo „Navigácia“, tak systém nastaví pohľad, pri ktorom je vidieť celý model jedného poschodia pod miernym uhlom
	4a	Pri móde „Prehliadanie“ môže používateľ ďalej použiť UC03, UC06 a UC07
	4b	Pri móde „Navigácia“ môže používateľ ďalej použiť UC03, UC07, UC08 a UC09
poznámky	Vždy je zobrazený v hlavnom okne práve jeden mód virtuálneho modelu, východiskovo je to „3D“	

Tab. 1 Prípád použitia UC01 Zvolenie módu zobrazenia

3.2.2 UC02 Pohybovanie sa po budove



Obr. 24 Prípád použitia „Pohybovanie sa po budove“

Prípád použitia pohybovanie sa po budove je úzko spätý s 3D módom. Po výbere tohto módu si môže používateľ prezerať model budovy z pohľadu prvej osoby podobne ako je tomu v akčných hrách.





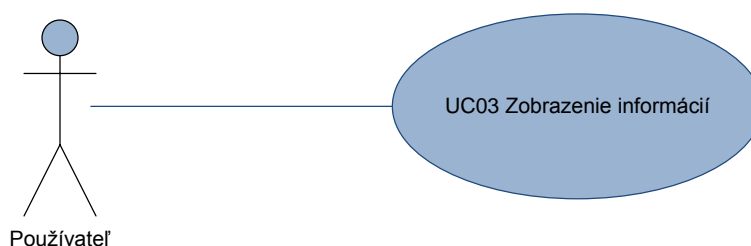
identifikátor	UC02		
názov	Pohybovanie sa po budove		
opis	Používateľ sa môže voľne pohybovať po modeli FIIT		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Požívateľ zmenil mód na 3D mód		
výstupné podmienky	Používateľ sa nachádza na novej pozícii alebo sa pozerá iným smerom		
používatelia	Bežný používateľ		
základná postupnosť	krok	činnosť	
	1	Systém nastaví pozíciu používateľa pred vchod do budovy	
	2	Používateľ šípkami alebo klávesmi W,S,A,D mení svoju pozíciu v budove, súčasným podržaním klávesy SHIFT sa mení pozícia o väčšiu vzdialenosť	
	3	Systém zobrazuje model z novej pozície alebo pohľadu používateľa	
alternatívna postupnosť	krok	činnosť	
	2a	Používateľ stlačením tlačidla myši a jej pohybom mení smer, ktorým sa pozerá	
poznámky	Ak sa pred používateľom nachádza stena alebo zatvorené dvere, tak sa jeho pozícia nezmení		

Tab. 2 Popis prípadu použitia UC02 Pohybovanie sa po budove





3.2.3 UC03 Zobrazenie informácií



Obr. 25 Prípad použitia „Zobrazenie informácií“

Pravá časť webovej stránky je vyhradená pre panel, ktorého obsah sa dynamicky mení. Jedným typom obsahu, ktorý sa v ňom zobrazuje sú informácie o budove, poschodiach, miestnostiach a pod. Tento informačný panel je permanentne zobrazený v 3D a prehliadacom móde. Jeho obsah sa mení napríklad prechodom do inej miestnosti (3D mód) alebo kliknutím na miestnosť (prehliadací a navigačný mód).

identifikátor	UC03		
názov	Zobrazenie informácií		
opis	Používateľovi sa v paneli zobrazia informácie o miestnosti alebo budove FIIT		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Používateľ sa nachádza v 3D móde, prehliadacom móde alebo navigačnom móde s už vyhľadaným modelom		
výstupné podmienky	Používateľovi sa zobrazia informácie o miestnosti alebo budove FIIT		
používatelia	Bežný používateľ		
základná postupnosť	krok	činnosť	
	1	Používateľ vykoná akciu vyvolávajúcu zobrazenie informácií o miestnosti (prechod do novej miestnosti v 3D móde alebo kliknutie na miestnosť vo zvyšných módoch)	
	2	Systém zmaže predchádzajúce informácie v paneli	

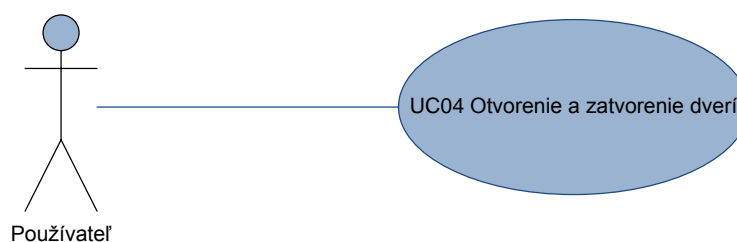




	3	Systém zobrazí aktuálne informácie o danej miestnosti v paneli
alternatívna postupnosť	krok	činnosť
	3a	Ak sa používateľ nachádza v 3D móde pred budovou, zobrazia sa informácie o celej budove
	3b	Ak sa používateľ nachádza v prehliadacom móde bez vybratej miestnosti, zobrazia sa informácie o zobrazenom poschodí
poznámky	V paneli sa budú zobrazovať štatistické údaje o budove, poschodiach (počet učiteľov, miestností, dverí a pod.) alebo miestnostiach (kapacita, typ miestnosti, informácie o užívateľoch spojených s danou miestnosťou a pod.)	

Tab. 3 Popis prípadu použitia UC03 Zobrazenie informácií

3.2.4 UC04 Otvorenie a zatvorenie dverí



Obr. 26 Prípad použitia „Otvorenie a zatvorenie dverí“

Tento prípad použitia nastáva len v 3d móde, ak chce používateľ prejsť z jednej miestnosti do druhej. Používateľ smie prejsť do druhej miestnosti len vtedy ak sú dvere medzi miestnosťami otvorené. Dvere je možné otvoriť kliknutím na ne.

identifikátor	UC04		
názov	Otvorenie a zatvorenie dverí		
opis	Interakcia používateľa v 3D móde s dverami		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Používateľ sa nachádza v 3D móde		

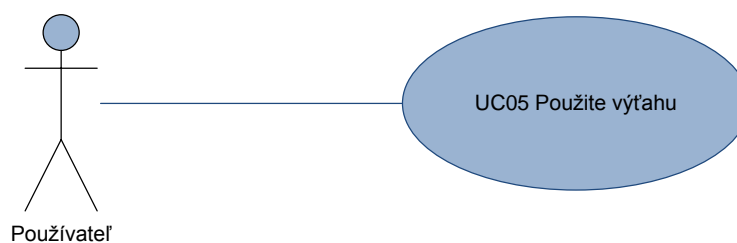




výstupné podmienky	Používateľovi môže/nemôže prejsť dverami	
používatelia	Bežný používateľ	
základná postupnosť	krok	činnosť
	1	Používateľ sa priblíži k dverám na dostatočnú vzdialenosť s priamou viditeľnosťou na dvere
	2	Používateľ klikne na dvere
	3	Ak boli dvere zatvorené, zobrazí sa animácia otvárajúcich sa dverí
	4	Ak sú dvere otvorené, užívateľ nimi môže prejsť
alternatívna postupnosť	krok	činnosť
	3a	Ak boli dvere otvorené, zobrazí sa animácia zatvárajúcich sa dverí
poznámky	Nie sú	

Tab. 4 Popis prípadu použitia UC04 Otvorenie a zatvorenie dverí

3.2.5 UC05 Použitie výťahu



Obr. 27 Prípad použitia „Použitie výťahu“

Vo vnútri modelu budovy používateľ prechádza medzi poschodiami buď po schodoch alebo môže použiť výťah. Vstup do výťahu je rovnaký ako vstup do miestnosti s dverami. V kabíne výťahu je panel s číslami poschodí. Po zvolení niektorého sa dvere výťahu zatvoria. Keď sa znova otvoria, užívateľ sa ocitá na inom poschodí.





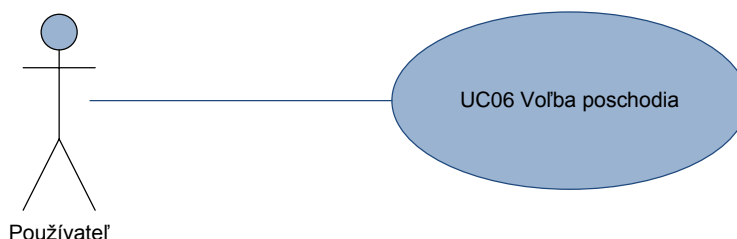
identifikátor	UC05		
názov	Použitie výťahu		
opis	Prechod medzi poschodiami pomocou výťahu		
priorita	3 = nízka	frekvencia	denne desiatky až stovky krát
vstupné podmienky	Zvolený je 3D mód		
výstupné podmienky	Používateľ sa premiestnil na iné poschodie		
používatelia	Bežný používateľ		
základná postupnosť	krok	činnosť	
	1	Používateľ sa priblíži k dverám výťahu na dostatočnú vzdialenosť s priamou viditeľnosťou na dvere	
	2	Používateľ klikne na dvere výťahu	
	3	Spustí sa animácia otvárajúcich sa dverí	
	4	Užívateľ vstúpi do kabíny.	
	5	Kliknutím na číslo poschodia na paneli zvolí cieľ	
	6	Dvere výťahu sa zatvoria na dobu dvoch sekúnd	
	7	Dvere výťahu sa otvoria	
	8	Užívateľ vystúpi z výťahu na inom poschodí, ako vstupoval	
alternatívna postupnosť	krok	Činnosť	
	6a	Ak užívateľ klikol na číslo poschodia, na ktorom sa nachádza, dvere ostávajú otvorené	
poznámky	Na príchod výťahu nie je potrebné čakať		

Tab. 5 Popis prípadu použitia UC05 Použitie výťahu





3.2.6 UC06 Voľba poschodia



Obr. 28 Prípad použitia „Voľba poschodia“

Používateľovi je umožnené zmenenie poschodia, v ktorom sa práve nachádza, ak má zvolený mód zobrazenia „Prehliadanie“.

identifikátor	UC06		
názov	Voľba poschodia		
opis	Používateľ si môže zmeniť zobrazené poschodie		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Požívateľ zmenil mód na prehliadací mód		
výstupné podmienky	Používateľ má v hlavnom okne zobrazené vybrané poschodie		
používatelia	Bežný používateľ		
základná postupnosť	krok	činnosť	
	1	Systém dáva používateľovi možnosť zmeniť zobrazené poschodie	
	2	Používateľ klikne na zobrazenie poschodia nad alebo pod ktorým sa práve nachádza	
	3	Systém nastaví pohľad, pri ktorom je vidieť celý model vybraného poschodia pod miernym uhlom, ak trvá načítanie dlhšie ako 1s, tak systém pred zobrazením poschodia vypíše „Loading...“	

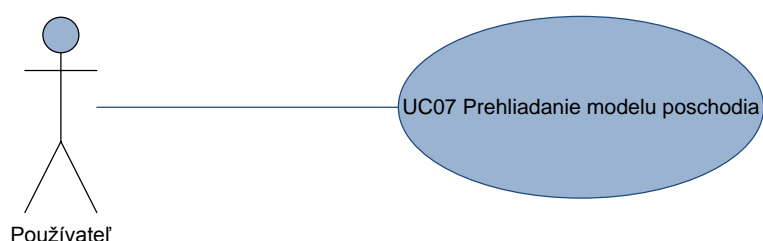




alternatívna postupnosť	krok	Činnosť
	3a	Ak sa používateľ nachádza na najnižšom poschodí a zvolí si zobrazíť poschodie pod alebo sa nachádza na najvyššom poschodí a zvolí si zobrazíť poschodie nad, tak systém na to používateľa upozorní a nezmení poschodie
poznámky	Nie sú	

Tab. 6 Popis prípadu použitia UC06 Voľba poschodia

3.2.7 UC07 Prehliadanie modelu poschodia



Obr. 29 Prípad použitia „Prehliadanie modelu poschodia“

Tento prípad použitia sa týka prehliadacieho a navigačného módu. V hlavnej časti okna sa nachádza model poschodia. Používateľ ho môže ľubovoľne otáčať a približovať.

identifikátor	UC07		
názov	Prehliadanie modelu poschodia		
opis	Používateľ si môže prezerať model z rôznych pohľadov		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Používateľ zmenil mód na prehliadací alebo navigačný mód		
výstupné podmienky	Používateľ sa pozerá na model poschodia z iného pohľadu		
používatelia	Bežný používateľ		

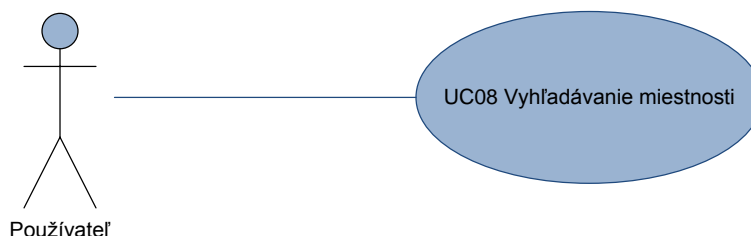




základná postupnosť	krok	činnosť
	1	Systém nastaví úvodný pohľad, pri ktorom je vidieť celý model poschodia pod miernym uhlom
	2	Používateľ stlačením tlačidla myši a jej pohybom mení uhol, pod ktorým sa na model poschodia pozerá
	3	Systém zobrazuje model z nového pohľadu
alternatívna postupnosť	krok	činnosť
	2a	Používateľ kolieskom na myši mení vzdialenosť od modelu
poznámky	Nie sú	

Tab. 7 Popis prípadu použitia UC07 Prehliadanie modelu poschodia

3.2.8 UC08 Vyhľadávanie miestnosti



Obr. 30 Prípad použitia „Vyhľadávanie miestnosti“ v navigačnom móde

Tento prípad použitia popisuje vyhľadanie miestnosti podľa zadaného kritéria a jednu z troch možností, ako získať informácie o danom mieste. Užívateľ zadáva podmienku, na základe ktorej sa nájdu vyhovujúce priestory. Aplikácia zahŕňa do výsledkov hľadania miestnosti, ktorých názov presne zodpovedá textu vloženému užívateľom (napríklad ak existuje izba CX50, výsledok ju bude obsahovať pre vstupný reťazec „CX50“). Okrem toho možno miestnosti vyhľadávať aj podľa mien osôb, ku ktorým sa viaže, napríklad pre sídlo Radoslava Zeleného možno zadať jeho celé meno, alebo len priezvisko, prípadne bez diakritiky (pre danú miestnosť budú akceptované reťazce ako „Radoslav Zelený“, „Radoslav Zeleny“, „zelený“ a pod.). Treťou skupinou možných vstupov sú kľúčové slová, teda výrazy, ktoré niečím charakterizujú miesto v budove. Každá miestnosť FIIT má priradenú množinu kľúčových slov, sú to výrazy ako prednáška, toaleta, študijné oddelenie a iné.

V prípade, že pre zadaný text sa nenájde žiadna zhoda, vypíše sa varovanie.





Z nájdených záznamov si užívateľ vyberá miestnosť kliknutím na jej názov v zozname výsledkov. Po výbere sa v hlavnom okne zobrazí model poschodia, na ktorom sa miestnosť nachádza a tá sa zvýrazní oproti zvyšku prostredia. Ak užívateľ klikne na miestnosť, program sa automaticky dostáva do prehliadacieho módu.

identifikátor	UC08		
názov	Vyhľadávanie miestnosti		
opis	Používateľ pomocou filtra vyhledá miestnosť a nechá si o nej zobrazit informácie		
priorita	1 = vysoká	frekvencia	denne niekoľko desiatok až stoviek krát
vstupné podmienky	Aktívny je navigačný mód		
výstupné podmienky	Používateľovi sa zobrazí zoznam miestností, ktoré vyhovujú kritériám, prípadne informácia, že žiaden vyhovujúci záznam sa nenašiel.		
používatelia	Bežný používateľ		
základná postupnosť	krok	činnosť	
	1	Užívateľ zadá do textového poľa v pravom paneli kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo	
	2	Užívateľ stlačí tlačidlo „Hľadať“	
	3	Systém zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam	
	4	Užívateľ kliknutím vyberie niektorú miestnosť zo zoznamu	
	5	Systém v hlavnom okne zvýrazní v modeli budovy zodpovedajúcu miestnosť a v pravom paneli zobrazí informácie o miestnosti. Mód sa automaticky mení na prehliadací.	
alternatívna postupnosť	krok	činnosť	
	3a	Ak bolo vyhľadávanie neúspešné, zobrazí sa hlásenie „Nebola nájdená žiadna miestnosť pre <input text>“, kde <input text> je text vložený užívateľom	

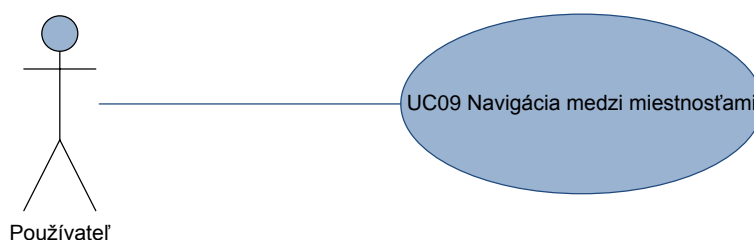




poznámky	Pri určovaní kritérií je nutné zadať vstup o dĺžke aspoň 3 znaky.
-----------------	---

Tab. 8 Popis prípadu použitia UC08 Vyhľadávanie miestnosti

3.2.9 UC09 Navigácia medzi miestnosťami



Obr. 31 Prípad použitia „Navigácia medzi miestnosťami“

Navigácia vo virtuálnej budove FIIT sa realizuje zvýraznením cesty medzi dvoma bodmi – miestnosťami. Prvé štyri kroky prípadu použitia sú zhodné s UC08. Užívateľ zaškrtnutím prepínacieho tlačidla Navigácia povolí v pravom paneli popri primárnom vyhľadávacom formulári ešte jeden podobný formulár. Spôsob vyhľadávania v ňom je totožný ako v primárnom formulári. Sekundárny formulár sa líši dvomi vlastnosťami: po kliknutí na miestnosť v zozname výsledkov sa nemení obsah hlavného okna. Formulár navyše obsahuje tlačidlo Nájst cestu, ktoré je zablokované, kým nie sú vybraté miestnosti v zoznamoch výsledkov oboch formulárov. Po jeho stlačení sa vypočíta najkratšia cesta medzi vybranými miestami.

Výsledok sa prezentuje v 3D modeli ako výrazná čiara vedúca priestormi budovy medzi vybranými koncovými bodmi.

identifikátor	UC09		
názov	Navigácia medzi miestnosťami		
opis	Používateľ pomocou filtra vyhledá dve miestnosti a dostáva znázornenie najkratšej cesty medzi nimi		
priorita	1 = vysoká	frekvencia	denne niekoľko desiatok až stoviek krát
vstupné podmienky	Aktívny je navigačný mód		
výstupné podmienky	Aplikácia používateľovi zobrazí trasu medzi dvomi bodmi virtuálneho interiéru.		





používatelia	Bežný používateľ	
základná postupnosť	krok	činnosť
	1	Užívateľ zadá do textového poľa v pravom paneli kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo
	2	Užívateľ stlačí tlačidlo „Hľadať“
	3	Systém zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam
	4	Užívateľ kliknutím na tlačidlo „Navigácia“ povolí sekundárny vyhľadávací formulár.
	5	Užívateľ zadá do textového poľa sekundárneho formulára kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo
	6	Užívateľ stlačí tlačidlo „Hľadať“ sekundárneho formulára
	7	V sekundárnom formulári sa zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam
	8	Užívateľ kliknutím vyberie jednu miestnosť v zoznamoch oboch formulárov
	9	Povolí sa tlačidlo „Nájsť cestu“
	10	Užívateľ stlačí „Nájsť cestu“
11	V hlavnom okne sa v modeli interiéru vykreslí čiara spájajúca dve vybrané miesta	
alternatívna postupnosť	krok	činnosť
	3a,7a	Ak bolo vyhľadávanie neúspešné, zobrazí sa hlásenie „Nebola nájdená žiadna miestnosť pre <input text>“, kde <input text> je text vložený užívateľom
poznámky	Pri určovaní kritérií je nutné zadať vstup o dĺžke aspoň 3 znaky.	

Tab. 9 Popis prípadu použitia UC09 Navigácia medzi miestnosťami





3.3 Nefunkcionálne požiadavky na systém

Systém nespracováva osobné informácie o jednotlivých užívateľoch, iba informácie o jednotlivých miestnostiach, rozvrhu miestností, obsadení jednotlivých miestností zamestnancami, otváracou dobou a podobne. Preto sa s nimi nemusí nakladať v zmysle zákona o ochrane osobných údajov. Z tohto dôvodu sa na systém nekladú nefunkcionálne požiadavky.

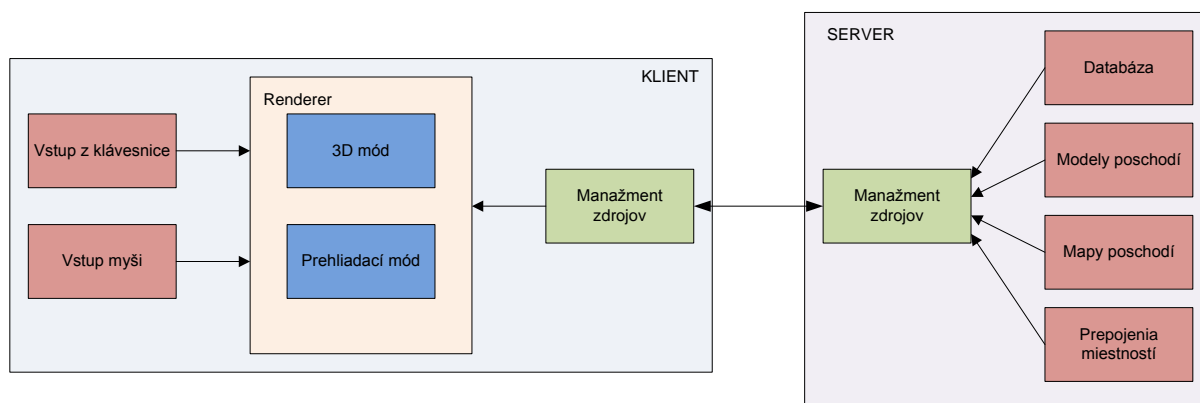




4 NÁVRH SYSTÉMU VIRTUÁLNEJ FIIT

Kapitola opisuje návrh systému 3D budovy FIIT. Obsahuje jeho architektúry a logický model, ako aj návrh databázy, s ktorou systém spolupracuje.

4.1 Architektúra systému Virtuálnej FIIT



Projekt virtuálnej FIIT je založený na klient-server architektúre. Najdôležitejším komponentom, ktorý tvorí jadro projektu je *Renderer*. *Renderer* sa skladá z dvoch častí – *3D mód* a *Prehliadací mód*.

V *3D móde* je zahrnutá funkcionálna otváranie dverí, pohyb po modeli z pohľadu prvej osoby, prechody medzi poschodiami, pohyb len v rámci mapy poschodia atď.

Prehliadací mód má na starosti zobrazovanie len jednotlivých poschodí, prepínanie medzi poschodiami, manipuláciu s poschodím (otáčanie, posúvanie, približovanie/odďaľovanie), vyznačovanie miestností, vyznačovanie ciest medzi miestnosťami, nastavovanie priehľadnosti, získavanie informácií o miestnostiach atď.

Obe časti *Renderera* prijímajú používateľské vstupy z komponentov *Vstup klávesnice* a *Vstup myši*. Používateľský vstup sa spracúva vždy len v aktuálnom móde. Ak sú pre beh *Renderera* potrebné dáta zo servera, *Renderer* si ich vyžiada cez komponent *Manažment zdrojov* (v klientskej časti). Príkladom môže byť napríklad vyžiadanie modelov poschodí po prvej inicializácii jedného z módov.

Na strane servera sa taktiež nachádza komponent *Manažment zdrojov*, ktorý spracúva požiadavky z klientskej časti a posiela vyžiadané zdroje. Klient môže požiadať o informácie o miestnostiach z databázy, modely poschodí, mapy poschodí alebo schému prepojenia miestností v poschodí. Tieto údaje sú mu následne prostredníctvom serverového *Manažmentu zdrojov* odoslané.





4.2 Komponenty systému Virtuálnej FIIT

Komponenty systému sú tvorené troma vrstvami – prezentačnou, aplikačnou a dátovou, podľa úrovne abstrakcie. Komponenty sú organizované do funkčných celkov (módov). V systéme rozlišujeme tri módy – 3D, prehliadací a navigačný mód, pričom každý obsahuje svoje vlastné komponenty alebo komponenty zdieľané s iným módom. (viď obr. 32)

4.2.1 Komponenty prezentačnej vrstvy

Prezentačná vrstva obsahuje komponenty slúžiace na priamu interakciu s užívateľom

Voľba módu

Komponent umožňujúci voľbu medzi 3D, prehliadacím a navigačným módom. Je tvorený 3 tlačidlami predstavujúcimi jednotlivé módy. Po kliknutí na jedno z tlačidiel sa inicializuje príslušný mód. Predvoleným módom je prehliadací mód.

Kliknutie na objekt

Komponent odchyťáva užívateľove kliknutia myšou na objekty (picking). Podľa typu objektu necháva spracovanie tejto akcie na komponenty Použitie výťahu a Otváranie dverí.

Pohyb myšou (3D mód)

Predstavuje odchyťávanie pohybov myšou a s tým súvisiacu zmenu cieľového bodu, na ktorý sa užívateľ pozerá.

Stlačenie klávesy

Predstavuje odchyťávanie stlačenia kláves a následnú zmenu polohy v 3D priestore. Ak sa zmenou polohy zmenila aj miestnosť, v ktorej sa používateľ nachádza, zavolá sa požiadavka na aktualizáciu informácií v informačnom paneli (komponent Zobrazenie informácií o miestnosti)

Voľba poschodia

Zabezpečuje možnosť zmeny poschodia v prehliadacom móde. Obsahuje dve tlačidlá, ktorých úlohou je zmeniť aktuálne poschodie na o jedno vyššie alebo o jedno nižšie.

Kliknutie na miestnosť

Tento komponent je spoločný pre prehliadací a navigačný mód. Zabezpečuje odchyťávanie kliknutí na objekty – miestnosti. Kliknutie na miestnosť v navigačnom móde spôsobí zmenu módu na prehliadací mód. Komponent okrem toho posiela požiadavku na aktualizáciu informácií v informačnom paneli (komponent Zobrazenie informácií o miestnosti)





Pohyb myšou (prehliadací a navigačný mód)

Predstavuje odchyťvanie pohybov myšou a s tým súvisiacu zmenu pohľadu, z ktorého sa používateľ pozerá na model poschodia.

Zadávanie hľadanej miestnosti

Komponent je tvorený formulárom zabezpečujúcim vyhľadanie miestnosti. Skladá sa z textového poľa pre zadanie výrazu, zoznamu nájdených miestností a tlačidla zahájenie vyhľadávania.

Zadávanie hľadanej cesty

Komponent je tvorený formulárom zabezpečujúcim vyhľadanie cesty. Skladá sa z dvoch textových polí pre zadanie výrazov, dvoch zoznamov nájdených miestností a tlačidla pre zahájenie vyhľadávania.

4.2.2 Komponenty aplikačnej vrstvy

Aplikačná vrstva tvorí jadro systému a jej úlohou je zabezpečiť funkcionality systému na nižšej úrovni (práca s dátami, výpočty ap.)

Inicializácia 3D módu

Obsahuje funkcionality potrebnú na zobrazenie modelu v 3D móde – inicializácia premenných, zabezpečenie načítania modelov a nastavenia pohľadu, zobrazenie informačného panela

Inicializácia prehliadacieho módu

Obsahuje funkcionality potrebnú na zobrazenie modelu poschodia v prehliadacom móde – inicializácia premenných, zabezpečenie načítania modelov poschodí a nastavenia pohľadu, zobrazenie informačného panela

Inicializácia navigačného módu

Obsahuje funkcionality potrebnú na zobrazenie modelu poschodia v navigačnom móde – inicializácia premenných, zabezpečenie načítania modelov poschodí a nastavenia pohľadu, zobrazenie vyhľadávacieho panela

Použitie výťahu

Komponent obsahuje funkcionality spojenú s používaním výťahu – privolanie výťahu, výber poschodia, načítanie príslušného poschodia, obrazovka načítania, animácia použitia výťahu

Otváranie dverí





Komponent obsahuje funkcionality spojenú s otváraním dverí – animácia otvorenia a zatvorenia dverí, rôzne typy dverí (obyčajné, dvojité, výťahové), umožnenie vstúpenia do miestnosti

Aktuálne nastavenia 3D módu

Obsahuje všetky informácie, ktoré sú potrebné na zobrazenie scény v želanom stave – aktuálna pozícia, pohľad, poschodie, miestnosť, načítané modely, stavy animácií, otvorené a zatvorené dvere

Zobrazenie modelu z FPS pohľadu

Obsahuje funkcionality potrebnú na zobrazenie 3D scény z FPS pohľadu na základe aktuálnych nastavení v komponente Aktuálne nastavenia 3D módu.

Zobrazenie informácií o miestnosti

Komponent zabezpečuje zobrazenie informácií o miestnosti v informačnom paneli na základe údajov z databázy. Je využívaný v 3D a prehliadacom móde.

Aktuálne nastavenia prehliadacieho a navigačného módu

Obsahuje všetky informácie, ktoré sú potrebné na zobrazenie scény v želanom stave – aktuálny pohľad, poschodie, vyznačená miestnosť, vyznačená cesta, načítaný model poschodia.

Zobrazenie modelu poschodia

Obsahuje funkcionality potrebnú na zobrazenie 3D scény poschodia na základe aktuálnych nastavení v komponente Aktuálne nastavenia prehliadacieho a navigačného módu.

Vyhľadanie miestnosti

Komponent zabezpečuje vyhľadanie výrazu (názvu miestnosti) v databáze cez komponent Rozhranie databázy a vrátenie zoznamu zodpovedajúcich miestností. Na základe vybranej miestnosti v zozname v komponente Zadávanie hľadanej miestnosti sa vyšle požiadavka na zvýraznenie miestnosti.

Vyhľadanie cesty

Komponent zabezpečuje vyhľadanie výrazov (názvov počiatkovej a cieľovej miestnosti) v databáze cez komponent Rozhranie databázy a vrátenie zoznamu zodpovedajúcich miestností. Na základe vybraných miestností v zoznamoch v komponente Zadávanie hľadanej cesty sa vyšle požiadavka na zvýraznenie cesty medzi miestnosťami.

Rozhranie databázy





Zabezpečuje funkcionálnosť pripojenia sa k databáze a komunikáciu s ňou cez SQL príkazy.

4.2.3 Komponenty dátovej vrstvy

Dátová vrstva predstavuje informácie a zdroje dát pre vyššie vrstvy

Model budovy

Predstavuje 3D model budovy uložený vo formáte „.o3dtgz“. Model je zložený z poschodí, miestností, dverí a výťahov, pričom každý komponent, ktorý obsahuje, má svoj unikátny názov.

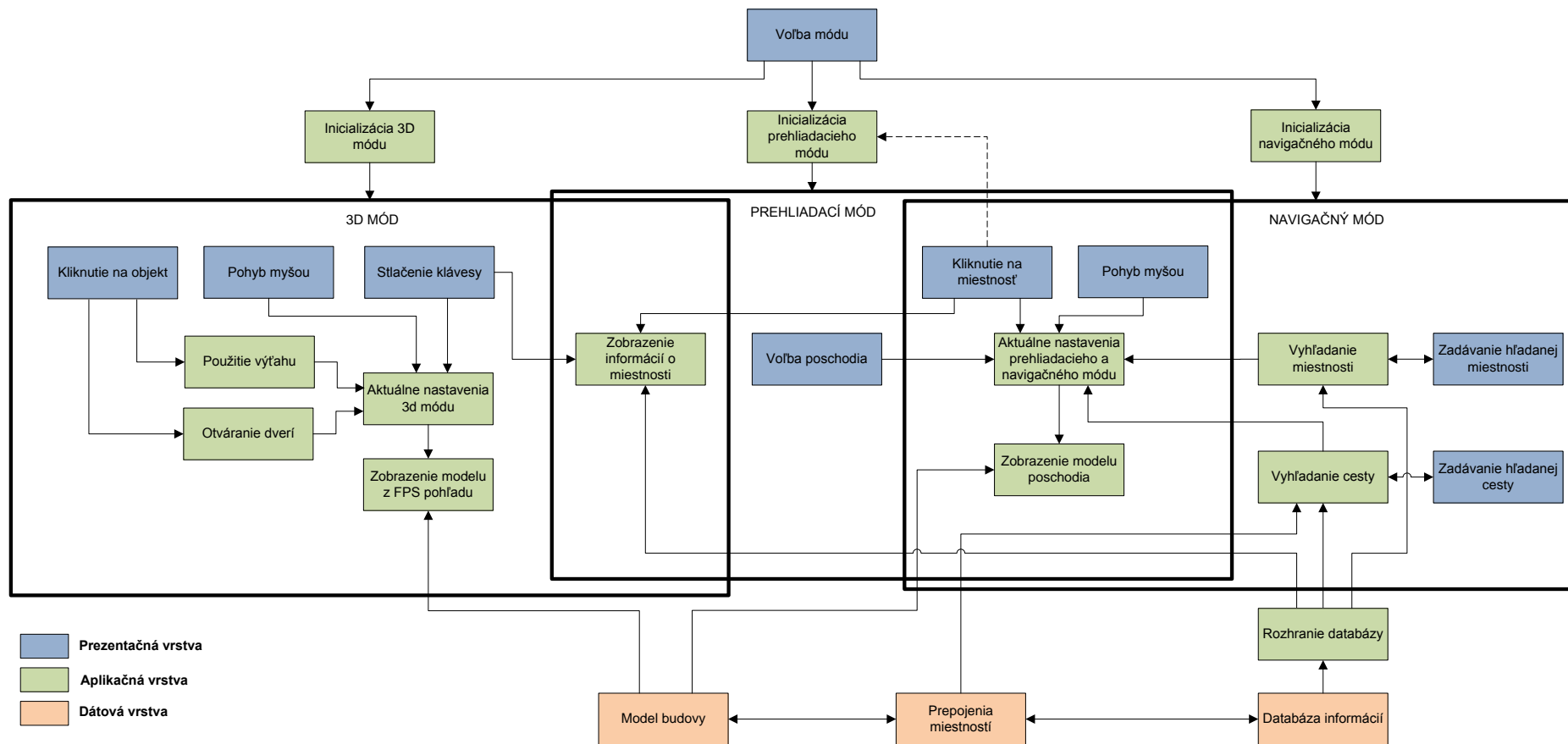
Prepojenia miestností

Predstavuje XML súbor, v ktorom sú uložené prepojenia jednotlivých uzlov a miestností. Prepojenia sú využívané pri vyhľadávaní cesty medzi miestnosťami.

Databáza informácií

Obsahuje informácie o miestnostiach (názov, typ, kapacita ap.) a učiteľoch.





Obr. 32 Diagram komponentov



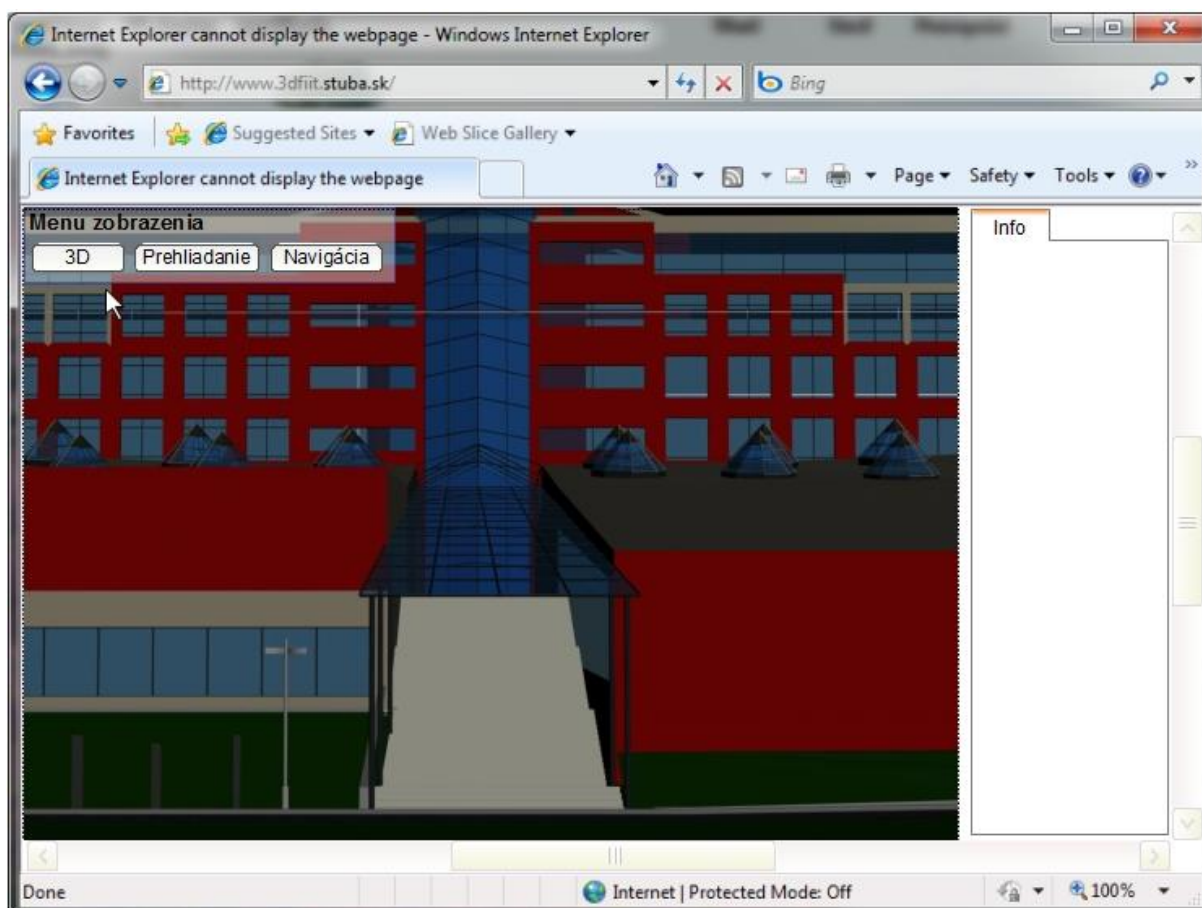


4.3 Návrh GUI systému

Podkapitola sa zaoberá návrhom používateľského rozhrania pre prípady použitia a celkového návrhu systému. Grafický návrh je vytvorený pre najpoužívanejší internetový prehliadač Internet Explorer 8 operačného systému Microsoft Windows XP.

4.3.1 Uvítacia obrazovka

Po pripojení na server a načítaní modelu sa v prehliadači zobrazí 3D model budovy s výberom zobrazenia modelu. Predefinovaným módom zobrazovania je 3D mód, ktorý umožňuje užívateľovi virtuálnu prechádzku po budove. Na pravej strane sa zobrazujú aktuálne informácie o objekte alebo miestnosti, v ktorej sa užívateľ virtuálne nachádza. Obrazovka je navrhnutá pre UC01 a je znázornená na obrázku 33.



Obr. 33 Uvítacia obrazovka





4.3.2 Obrazovka pre 3D mód

Obrazovka pre 3D mód je totožná s úvítacou obrazovkou na obr. 33, nakoľko po spustení aplikácie sa zobrazí práve tento mód. Obrazovka obsahuje tieto prvky:

- menu módu zobrazenia s výberovými tlačidlami,
- okno s modelom budovy,
- informačný panel (*textarea*) s aktuálnymi informáciami.

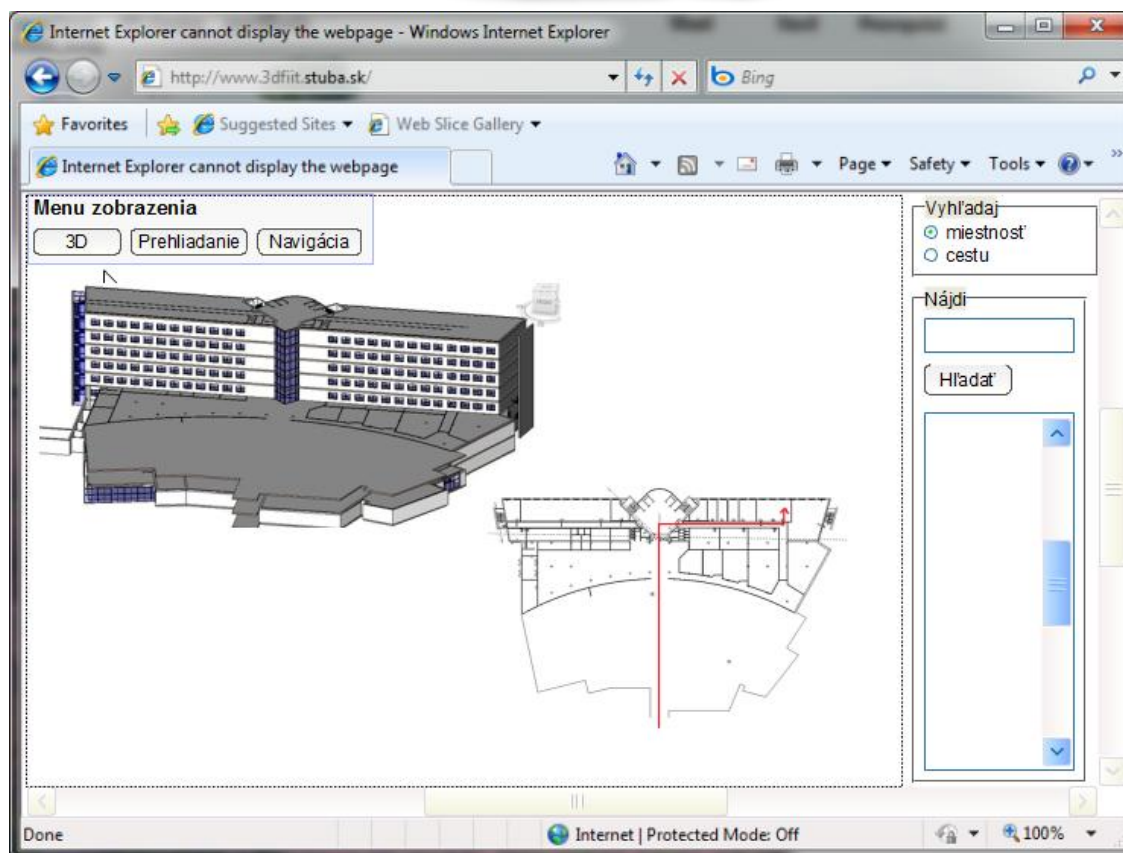
4.3.3 Obrazovka vyhľadávania miestností

Obrazovka pre vyhľadávanie miestností je aktivovaná výberom z ľavého horného menu, tlačidlom “navigácia” a výberom pre vyhľadávanie miestností. Obrazovka obsahuje prvky:

- menu módu zobrazenia s výberovými tlačidlami,
- menu pre výber možností hľadania,
- vstupný formulár pre hľadaný objekt,
- tlačidlo pre vyhľadanie výsledkov,
- textové pole s výsledkami.

Návrh obrazovky je na obrázku 34.





Obr. 34 Obrazovka vyhľadávania miestností

4.3.4 Obrazovka vyhľadávania cesty

Obrazovka pre vyhľadávanie cesty je aktivovaná výberom z ľavého horného menu, tlačidlom "navigácia" a výberom pre vyhľadanie cesty. Obrazovka obsahuje prvky:

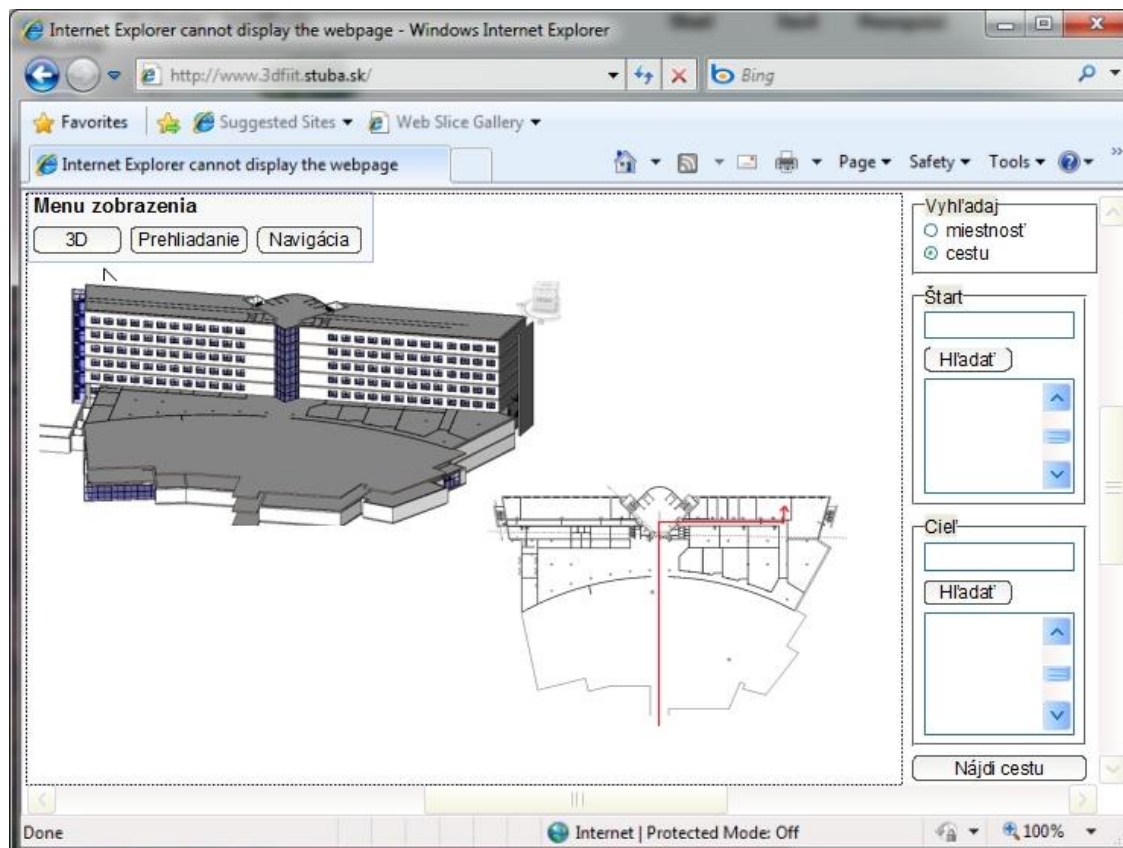
- menu módu zobrazenia s výberovými tlačidlami,
- menu pre výber možností hľadania,
- vstupný formulár pre hľadaný štartovací objekt,
- tlačidlo pre vyhľadanie výsledkov možných štartovacích objektov,
- textové pole s výsledkami možných štartov,
- vstupný formulár pre hľadaný cieľový objekt,
- tlačidlo pre vyhľadanie výsledkov možných cieľov,
- textové pole s výsledkami nájdených cieľov,





- tlačidlo pre výpočet a zobrazenie cesty.

Návrh obrazovky je na obrázku 35.



Obr. 35 Obrazovka vyhľadávania cesty

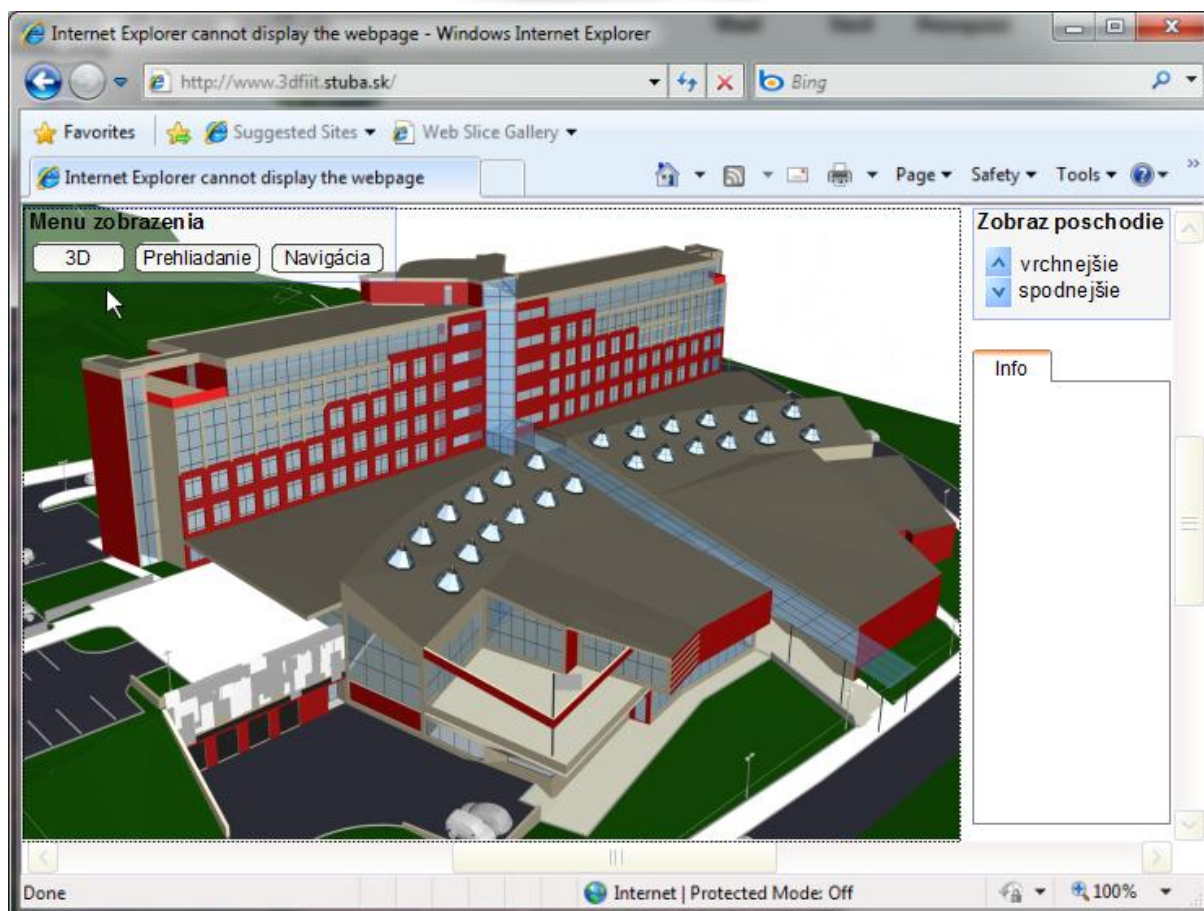
4.3.5 Obrazovka prehľadávania

Obrazovka pre prehľadanie miestností je aktivovaná výberom z ľavého horného menu, tlačidlom “prehľadanie”. Obrazovka obsahuje prvky:

- menu módu zobrazenia s výberovými tlačidlami,
- menu pre výber poschodia na zobrazenie,
- informácie o aktívnej miestnosti či objekte.

Návrh obrazovky je na obrázku 36.





Obr. 36 Obrazovka prehľadávania

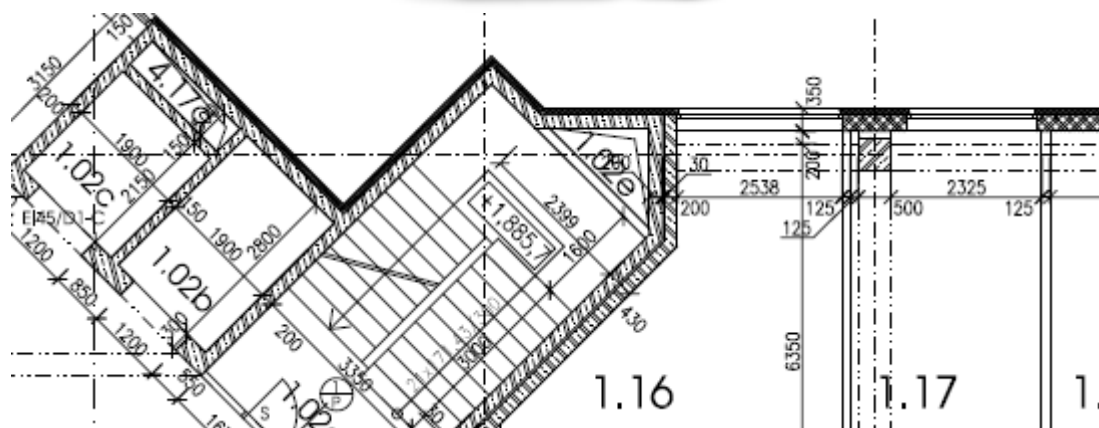
4.3.6 Zmeny vo výkresoch

Podľa označenia výkresov sa zmeny udiali v nasledovných 4 výkresoch poschodí:

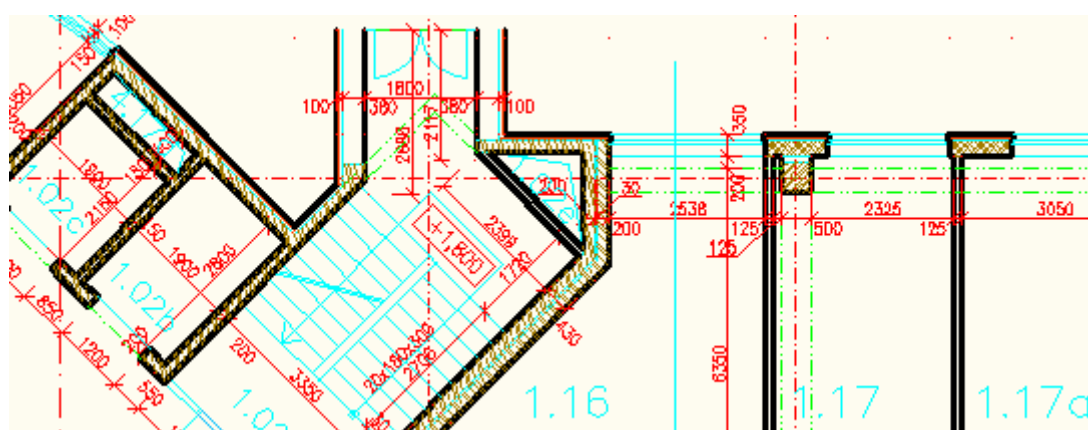
1.poschodie:

- sú zmenené z časti obidve krídla poschodia, predná časť zostala bez podstatných zmien
- v ľavom krídle boli zlúčené miestnosti 1.09a a 1.10a do miestnosti 1.10 a miestnosť 1.11a do 1.11. Opačne, miestnosť 1.12 bola rozdelená na miestnosť 1.12 a 1.12a. Drobné, z pohľadu abstrakcie modelovania nepodstatné, zmeny sa udiali aj v iných častiach poschodia.
- v stredovej časti v pravom schodisku bol zrušený zadný východ (Obr. 1. a 2.)





Obr. 1. Upravená časť poschodia 1NP

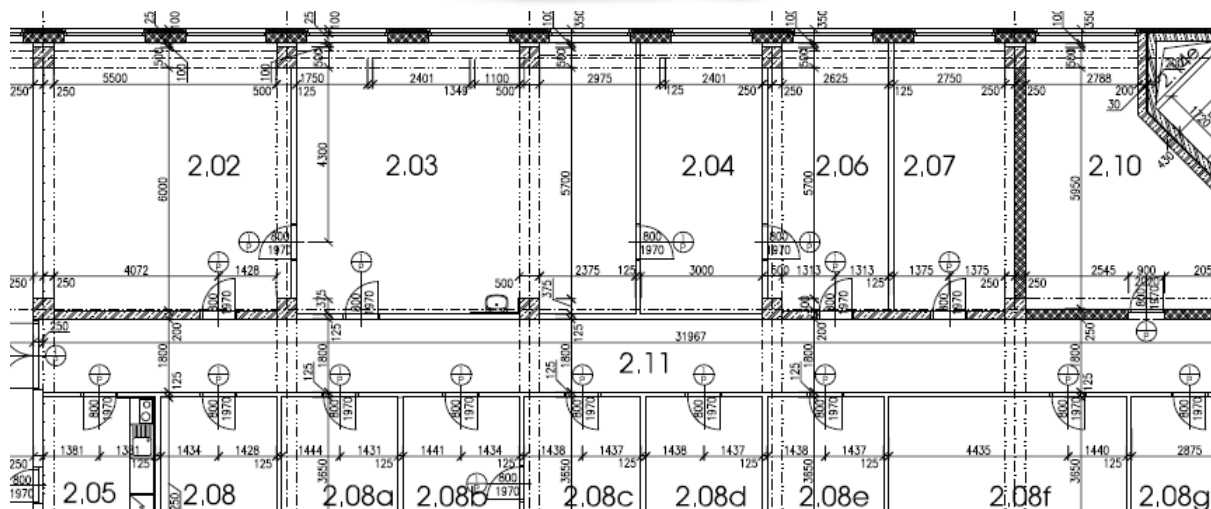


Obr. 37. Pôvodná časť poschodia 4NP

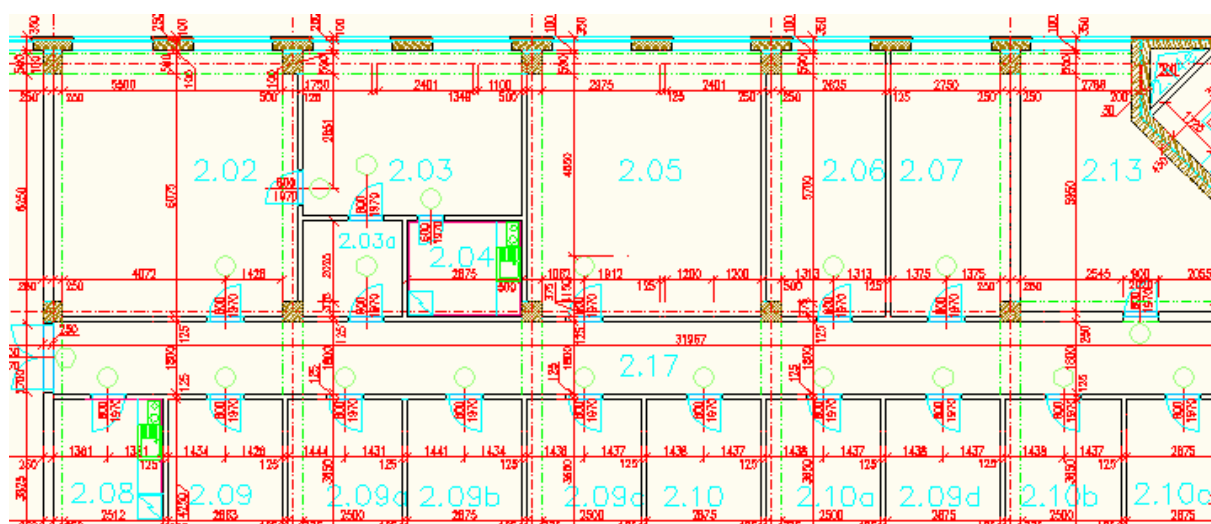
2.poschodie:

- sú zmenené obidve krídla poschodia
- v ľavom krídle je zrušená jedna kuchynka (Obr. X – miestnosť 2.03), odlišné číslovanie miestností, zlúčenie alebo rozdelenie niektorých miestností
- na tomto výkrese boli pridané tiež detaily týkajúce sa strechy, ale na prvý pohľad nie je zrejmé, čo z tieto čiary predstavujú





Obr. 38 Upravená časť poschodia 4NP

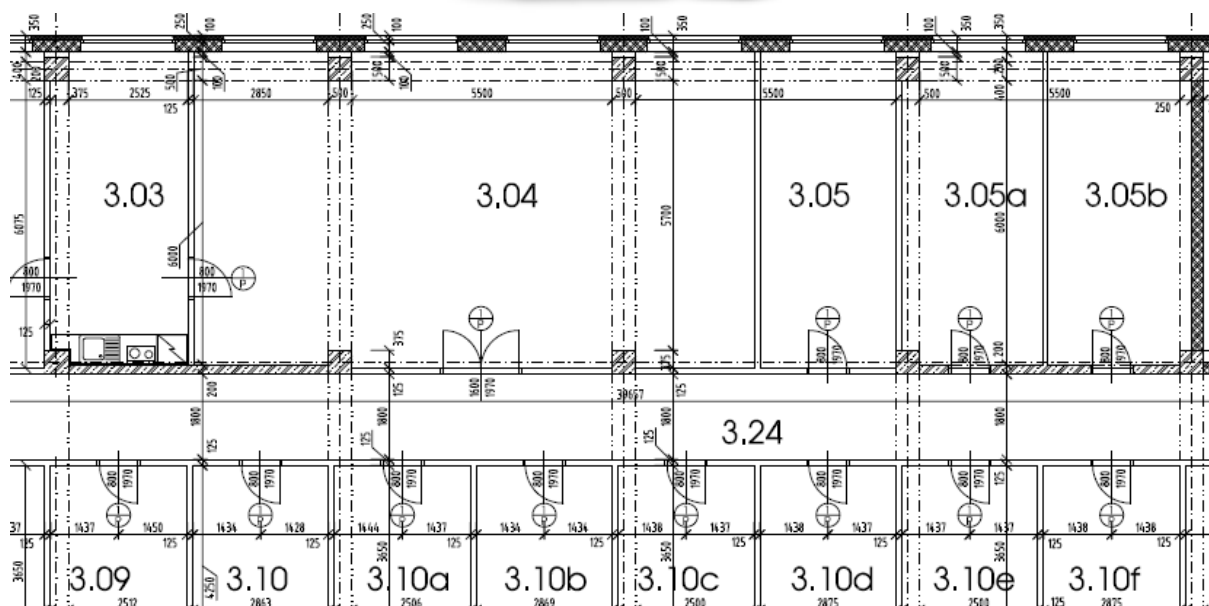


Obr. 39 Pôvodná časť poschodia 4NP

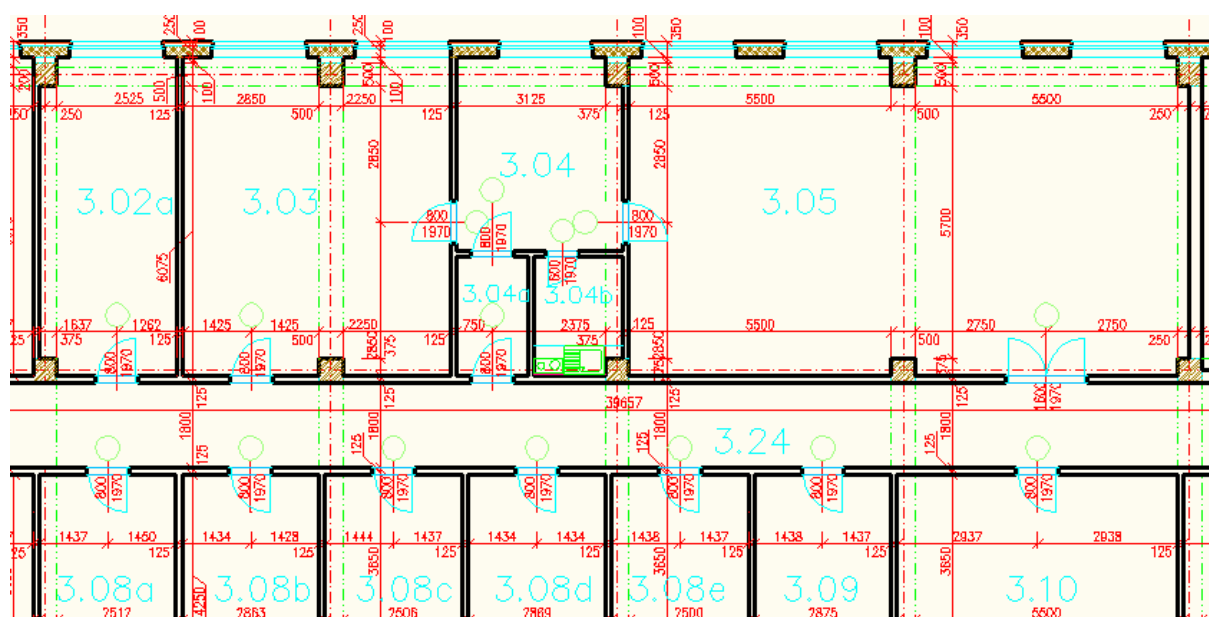
3.poschodie:

- sú zmenené obidve krídla poschodia
- v ľavom krídle je zmenená a posunutá kuchynka (Obr. X – miestnosť 3.03), odlišné číslovanie miestností, zlúčenie alebo rozdelenie niektorých miestností
- v pravom krídle nie sú niektoré miestnosti oddelené pevnou stenou, ale iba posuvnou odŕvavajúcou sa priečkou





Obr. 40 Upravená časť poschodia 4NP

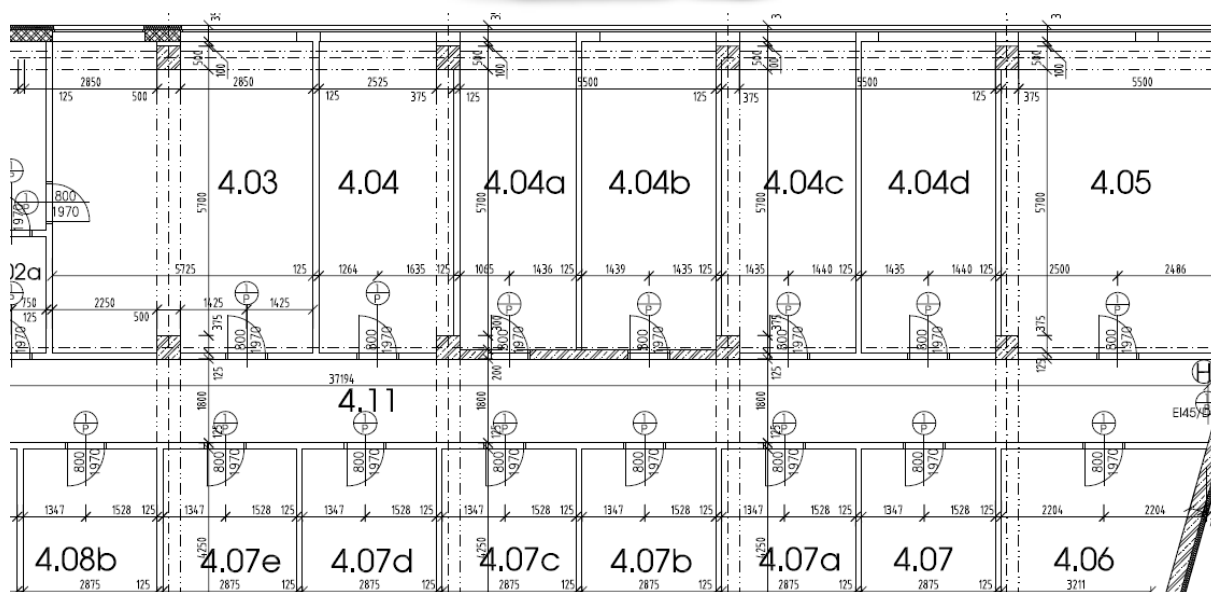


Obr. 41 Pôvodná časť poschodia 4NP

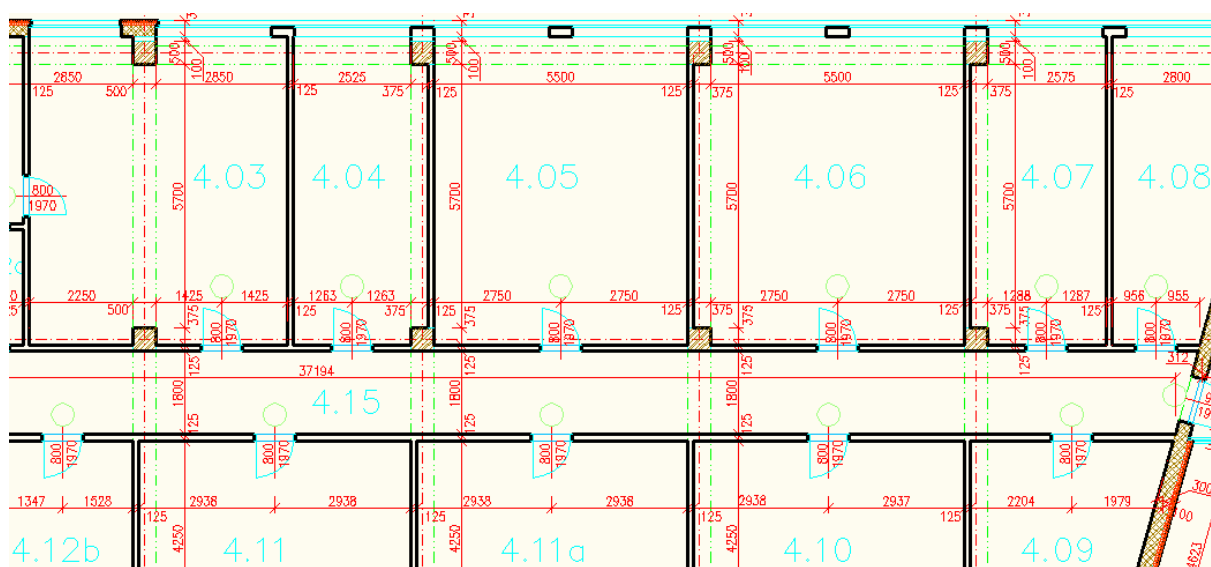
4.poschodie:

- je zmenené pravé krídlo poschodia
- zmeny sa týkajú odlišného číslovania miestností, niektoré miestnosti sú aj zlúčené do jednej, iné zase rozdelené do dvoch





Obr. 42 Upravená časť poschodia 4NP



Obr. 43 Pôvodná časť poschodia 4NP

Zmeny vo výške poschodí neboli z uvedených výkresov postrehnuté.

Poskytnuté nové výkresy sú vo formáte pdf a len čiernobiele. To je značná nevýhoda, pretože nie je možné vypnúť jednotlivé zobrazenie vrstiev výkresu a len dvoj-farebnosť sťažuje orientáciu vo výkrese. Podstatnou nevýhodou ovplyvňujúcou modelovanie je, že z pdf formátu nie je možné získavať rozmery meraním vzdialeností. Preto je nevyhnutné získať tieto pozmenené výkresy aj vo formáte dwg.

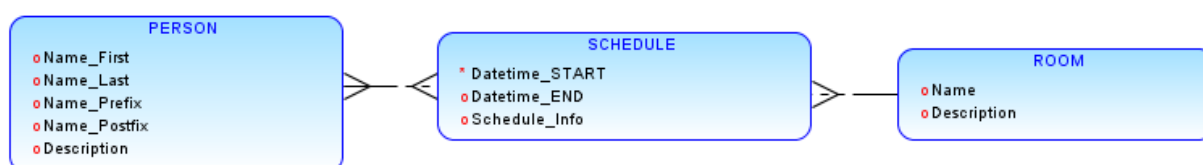




4.4 Databázový návrh

4.4.1 Logický dátový model

Dátový model aplikácie je veľmi jednoduchý. Rozhodli sme sa v ňom neimplementovať problém hľadania cesty medzi dvoma miestnosťami, preto obsahuje iba informácie o osobách, miestnostiach a rozvrhových akciách. Tieto informácie by mali byť dostatočné na to, aby bolo možné efektívne identifikovať miestnosť v ktorej sa hľadaná osoba nachádza. V prípade potreby je dátový model voľne rozšíriteľný a upraviteľný.



Obr. 44 Dátový model

Entita Person

Entita reprezentuje osoby v systéme. Každá osoba je popísaná nasledovnými atribútmi:

- Name_First – Meno osoby
- Name_Last – Priezvisko osoby
- Name_postfix – Prípona mena. Obsahuje tituly uvádzané za menom, prípadne iné znaky za menom
- Name_prefix – Predpona mena. Obsahuje tituly uvádzané pred menom, prípadne iné znaky pred menom

Entita Room

Entita reprezentuje miestnosti v systéme. Každá miestnosť je popísaná nasledovnými atribútmi:

- Name – Krátky názov miestnosti. V starej budove FEI/FIIT STU je to napríklad „de300“
- Description – Popis miestnosti. Napríklad „Aula Aurela Stodolu“

Entita Schedule





Entita reprezentuje rozvrh v miestnostiach v systéme (napríklad je to prednáška MSI konajúca sa 3. novembra 2009 od 15.00 do 17.00). Každá položka rozvrhu je popísaná nasledovnými atribútmi:

- Datetime_START – Začiatok rozvrhovej inštancie obsahujúci dátum a čas.
- Datetime_END – Koniec rozvrhovej inštancie obsahujúci dátum a čas.
- Schedule_Info – Popis rozvrhovej akcie (napríklad „Prednáška z predmetu MSI“)

Vzťah entít Schedule a Room

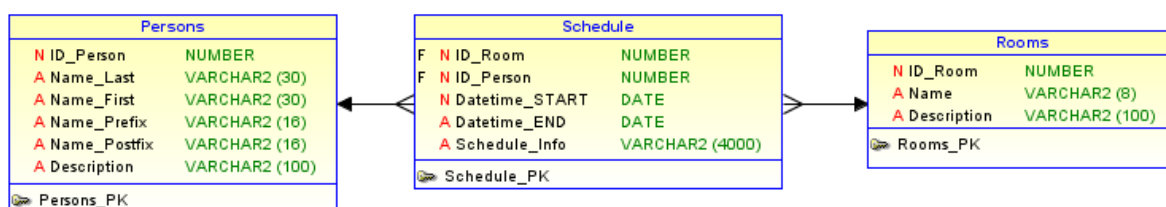
Vzťah popisuje vzťah medzi rozvrhovou akciou a miestnosťou, kedy pre každú miestnosť môže existovať nula, jedna alebo viacero rozvrhových akcií a každá rozvrhová akcia sa koná práve v jednej miestnosti.

Vzťah entít Schedule a Person

Vzťah popisuje vzťah medzi rozvrhovou akciou a osobou, kedy pre každú osobu môže existovať žiadna, jedna alebo viacero rozvrhových akcií a každá rozvrhová akcia môže mať pridelených nula, jednu alebo viacero osôb.

4.4.2 Fyzický model

Fyzický model vychádza z vyššie uvedeného dátového modelu.



Obr. 45 Fyzický model

Tabuľka Person

Tabuľka reprezentuje osoby v systéme, teda entitu Person z logického dátového modelu. Každá osoba je popísaná nasledovnými atribútmi:

- ID_Person – Jednoznačný číselný identifikátor osoby (primárny kľúč tabuľky Person – Person_PK)
- Name_First – Meno osoby. (Maximálne 30 znakov)
- Name_Last – Priezvisko osoby. (Maximálne 30 znakov)





- Name_postfix – Prípona mena. Obsahuje tituly uvádzané za menom, prípadne iné znaky za menom. (Maximálne 30 znakov)
- Name_prefix – Predpona mena. Obsahuje tituly uvádzané pred menom, prípadne iné znaky pred menom.
- Description – Popis osoby. Napríklad „Správca servera študent“.

Tabuľka Room

Tabuľka reprezentuje miestnosti v systéme, teda entitu Room z logického dátového modelu. Každá miestnosť je popísaná nasledovnými atribútmi:

- ID_Room – Jednoznačný číselný identifikátor miestnosti (primárny kľúč tabuľky Room – Room_PK).
- Name – Krátky názov miestnosti. V starej budove FEI/FIIT STU je to napríklad „de300“.
- Description – Popis miestnosti. Napríklad „Aula Aurela Stodolu“.

Tabuľka Schedule

Tabuľka reprezentuje rozvrh v miestnostiach v systéme (napríklad je to prednáška MSI konajúca sa 3. novembra 2009 od 15.00 do 17.00), teda entitu Schedule z logického dátového modelu. Každá položka rozvrhu je popísaná nasledovnými atribútmi:

- ID_Room – Jednoznačný číselný identifikátor miestnosti (primárny kľúč tabuľky Room). Tento atribút je cudzím kľúčom do tabuľky Room.
- ID_Person – Jednoznačný číselný identifikátor miestnosti (primárny kľúč tabuľky Person). Tento atribút je cudzím kľúčom do tabuľky Person.
- Datetime_START – Začiatok rozvrhovej inštancie obsahujúci dátum a čas.
- Datetime_END – Koniec rozvrhovej inštancie obsahujúci dátum a čas.
- Schedule_Info – Popis rozvrhovej akcie (napríklad „Prednáška z predmetu MSI“).

Tabuľka obsahuje zložený primárny kľúč – Schedule_PK. Skladá sa z kombinácie identifikátora miestnosti (ID_Room), identifikátora osoby (ID_Persono) a času a dátumu začiatku rozvrhovej akcie (Datetime_START).

Vzťah entít Schedule a Room





Vzťah je reprezentovaný atribútom ID_Room tabuľky Room a atribútom ID_Room tabuľky Schedule. Vzťah popisuje súvislosť medzi rozvrhovou akciou a miestnosťou, kedy pre každú miestnosť môže existovať nula, jedna alebo viacero rozvrhových akcií a každá rozvrhová akcia sa koná práve v jednej miestnosti.

Vzťah entít Schedule a Person

Vzťah je reprezentovaný atribútom ID_Person tabuľky Person a atribútom ID_Person tabuľky Schedule. Vzťah popisuje súvislosť medzi rozvrhovou akciou a osobou, kedy pre každú osobu môže existovať žiadna, jedna alebo viacero rozvrhových akcií a každá rozvrhová akcia má pridelenú práve jednu osobu.

Vzťah entít Room a Person

Vzťah je reprezentovaný atribútom ID_Room tabuľky Person a atribútom ID_Room tabuľky Room. Vzťah popisuje súvislosť medzi osobou a miestnosťou, kedy pre každú osobu môže existovať „domovská“ miestnosť. Táto informácia predstavuje miestnosť kde je daná osoba zastihnuteľná, ak práve nevykonáva rozvrhovú akciu.

Návrh systému popísaný v kapitole 4 je predpokladom pre dobrý základ následnej implementácie systému 3D budovy FIIT. Obsahuje všetky nevyhnutné časti. Popis architektúry systému poskytuje jasný pohľad na rozloženie jeho súčastí. Návrh GUI dáva prvú predstavu o vzhľade systému pre používateľa a nakoniec databázový návrh popisuje uchovávané dáta a ich vzťahy v dátovom a fyzickom modeli.

4.5 Určenie priorít implementácie systému

Určenie priorít implementácie častí systému vychádza hlavne z funkcionálnych požiadaviek. Stanovuje postup, ktoré súčasti systému sa implementujú skôr, pričom sa berie do úvahy ich dôležitosť, vzájomná previazanosť a náročnosť, resp. potrebný odhadovaný čas na ich vývoj. Špecifikované časti systému budú implementované v nasledujúcom poradí:

- prehliadací mód,
- databáza údajov,
- vyhľadávanie a navigácia,
- 3D mód.

Podrobnejší postup implementácie:

- vytvorenie modelu jedného poschodia,





- vytvorenie základného grafického menu (GUI),
- zakomponovanie modelu do GUI prostredníctvom O3D – základ prehliadacieho módu (stanovenie štruktúry a rozmiestnenia súborov projektu),
- picking miestností – odchytenie a určenie, na ktorú miestnosť bolo kliknuté
- prepojenie menu s databázou (dostupné zobrazenie informácií o miestnostiach)
- namodelovanie zvyšných poschodí
- voľba zmeny poschodia
- graf miestností, vyhľadávanie a zobrazenie najkratšej cesty (navigačný mód)
- 3D mód – loading poschodí a pohyb po budove
- interaktívne prvky – otváranie dverí, výťah (3D mód)
- zaznamenávanie aktuálnej pozície používateľa v 3D móde

Implementácia niektorých, za sebou idúcich častí, bude súčasná. Minimálna implementovaná funkcionálna požadovaná v 1. semestri vychádza z prototypu prehliadacieho módu (prvé 4 body podrobnejšieho postupu implementácie). Ďalej by bolo vhodné vytvoriť základné prepojenie prototypu systému s databázou. Pre vyhľadávanie a navigáciu je potrebné mať namodelované zvyšné poschodia. Implementácia týchto častí v 1. semestri bude závisieť od progresu projektu.





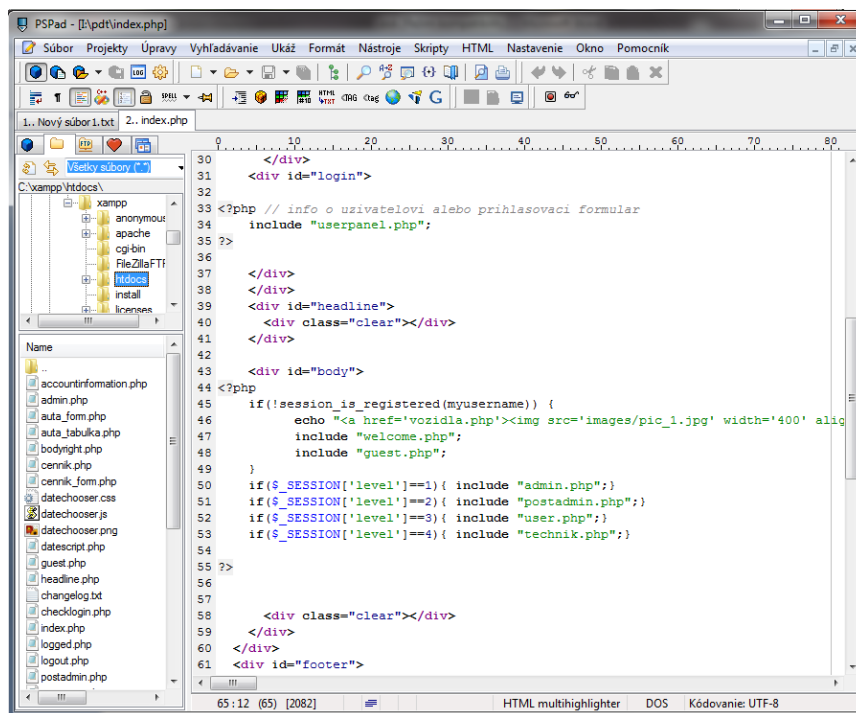
5 IMPLEMENTÁCIA SYSTÉMU VIRTUÁLNEJ FIIT

5.1 Implementácia používateľského rozhrania

Vzhľadom na dôležitú požiadavku „systém má byť klient - server aplikácia“ je systém realizovaný ako model umiestnený na serveri a prístupný pomocou adresy url [http:\\147.175.159.181\HTML]. Po zadaní adresy sa zobrazí uvítacia obrazovka s krátkym opisom o aplikácii.

Implementácia používateľského rozhrania je realizovaná pomocou:

- jazyka HTML, ktorý zabezpečuje základnú štruktúru systému a zabezpečuje prezentačnú vrstvu systému
- kaskádových štýlov CSS
- editoru PSPad 4.5.4 [www.pspad.com] (viď obr. 39), ktorý podporuje ftp pripojenie na server.



Obr. 46 Editor PSPad 4.5.4

5.1.1 Základná štruktúra

Základná štruktúra dokumentu je implementovaná pomocou jazyka HTML s využitím kaskádových štýlov. Rozlíšenie obrazovky, na ktorom bude model spustený, nie je vopred





známe. Preto je potrebné implementovať časti modelu (okno modelu a informačný panel) s relatívnymi veľkosťami.

5.1.2 Hlavička dokumentov

V každom dokumente .html je na začiatku dokumentu vložená hlavička s popisom kódovania stránky, jeho názvom a odkazom na súbor .css . Implementácia hlavičky je v ukážke nižšie

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>Virtual FIIT - model</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <link rel=StyleSheet href="../CSS/main.css" type="text/css">
</head>
```

5.1.3 Implementácia Javascriptov

Pre zobrazovanie O3D modelu v používateľskom rozhraní je potrebné do dokumentov html zahrnúť aj zdrojové kódy pre javascript, ktorý má za úlohu vykonávanie funkcií a operácií nad objektmi (obrázok, vyplňanie textových polí, inicializácia módov zobrazovania). Spôsob zahrnutia javascriptových funkcií do hlavičky html dokumentu je nasledovný:

```
<head>
.
.
.
<script src="../JS/o3djs/base.js" type="text/javascript"> </script>
<script src="../JS/RoomsGraph/Graph.js" type="text/javascript"> </script>
<script src="../JS/RoomsGraph/RoomGraphData.js" type="text/javascript"> </script>
<script src="../JS/ModeManager.js" type="text/javascript"> </script>
<script src="../JS/Mode3D.js" type="text/javascript"> </script>
<script src="../JS/ModeExplore.js" type="text/javascript"> </script>
<script src="../JS/ModeNavigate.js" type="text/javascript"> </script>
<script src="../JS/FormModes.js" type="text/javascript"> </script>
<script src="../JS/FormSearch.js" type="text/javascript"> </script>
<script src="../JS/FormInfo.js" type="text/javascript"> </script>
<script src="../JS/EngineFPS.js" type="text/javascript"> </script>
<script src="../JS/EngineExplore.js" type="text/javascript"> </script>
<script src="../JS/EngineExploreListeners.js" type="text/javascript"> </script>
<script src="../JS/Main.js" type="text/javascript"> </script>
.
.
.
</head>
```





5.1.4 CSS kaskádové štýly

Základná štruktúra a prvky stránok html je realizovaná pomocou CSS kaskádových štýlov. Tie upravujú vzhľad a rozloženie niektorých prvkov dokumentu ako napríklad tabuľky, obrázkov, pozadia obrazoviek (používateľského rozhrania), odkazov ako aj tlačidiel. Príslušnosť prvku ku konkrétnemu štýlu je implementované tagom `id='style_id_in_css_file'`. Príklad implementácie štýlov pre najdôležitejšie prvky dokumentu:

```
* {
    margin: 0;
    padding: 0;
    color: #102c57;
}
body {
    background: url(..../img/bg_main.jpg) repeat-x;
    font-size: 15px;
    color: #302e55;
}
h1, h2, h3, h4 {
    border: none;
    text-decoration: none;
    color: white;
}
hr{
    border: dashed #2c4c7c;
    border-width: 1px 0 0 0;
    height: 0;
    line-height:0px;
    font-size:0;
    margin:0;
    padding:0;
}
#logo3d{
    background: url(..../img/logofiito3d.png);
    width: 200px;
    height: 140px;
    margin: 0 0 0 0;
    /*border: dashed red;*/
}
#logolostbig{
    background: url(..../img/logolostbig.png);
    background-repeat: no-repeat;
    background-position: center;
    width: auto;
    height: auto;
    /*border: dashed green;*/
}
a:link {text-decoration: none; color: white;}
a:visited {text-decoration: none; color: white;}
a:active {text-decoration: none; color: white;}
a:hover {text-decoration: underline; color: white;}
```





Obrázky sú v dokumente implementované pomocou oddielov `<div></div>`, v ktorých je zadefinovaný zdrojový obrázok a ďalšie atribúty. Implementácia touto cestou zabezpečuje relatívnu pozíciu obrázka vzhľadom na dokument a automaticky sa prispôsobuje jeho pozícia rozlíšeniu obrazovky.

5.1.5 Úvodná obrazovka

Po zadaní neúplnej adresy URL , napr. <http://www.3dmodelfiit/>, server automaticky vyhľadá súbor s názvom *index.**, kde * znamená príponu html, htm alebo php. Preto dokument úvodnej obrazovky musí mať tento názov. Vzhľadom na to, že úvodná obrazovka nemusí byť implementovaná v dynamickom jazyku php a stačí ju implementovať ako html dokument, úplný názov súboru bude v tvare *index.html* .

Úvodná obrazovka má len informatívny charakter, preto na nej nebola implementované plná funkcionality modelu. Úvodná obrazovka podporuje zmenu prehliadača do režimu fullscreen. Implementácia úvodnej obrazovky je :

```
<body>
<div id="logolostbig">
<div id="logo3d"></div>

<div align="center">
<table border="0" align="center" style="vertical-align: middle;" height="200px"
cellsoacing="0" cellpadding="0">

<tr><td colspan="2" align="center"><font color="white" size="15">3D Virtual FIIT STU
</font><br><br></td>

<tr><td><div id="button"><a href="index_resize_flexible.html" target="_parent"
alt="flexible">Relative sized</div></td>
</tr>
</table>
<br>

Please, turn your browser to <u><b>Fullscreen mode</b></u>.<br>

<a href="javascript:void(0);" target=" parent" onClick="fullScreen('index.html');">
.: Switch to Fullscreen :.
</a>

</div>
<div align="right">

</div>
</div>
</body>>
```

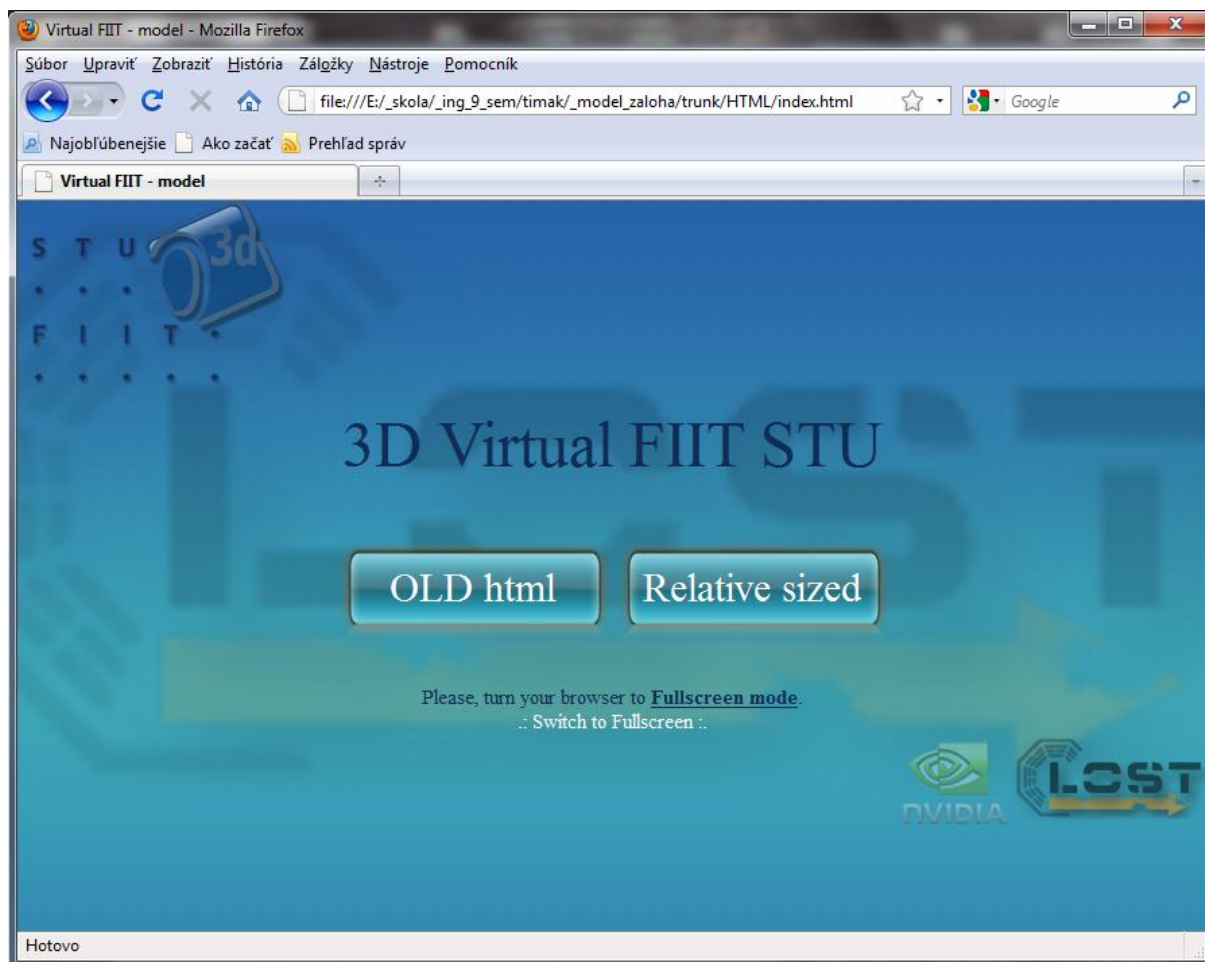




Funkcia skriptu pre prepnutie prehliadača do fullscreen módu:

```
<SCRIPT LANGUAGE="JavaScript">
function fullScreen(theURL) {
    window.open(theURL, '', 'fullscreen=yes, scrollbars=no');
}
</script>
```

Implementovaný dokument s využitím CSS kaskádových štýlov vyzerá nasledovne :



Obr. 47 Implementovaná úvodná obrazovka





5.1.6 Navigácia a módy modelu

Uvítacia obrazovka je prepojená s dokumentom pre používateľské rozhranie. To obsahuje rozloženie jednotlivých prvkov ako:

- Okno 3D modelu v O3D jazyku
- Tabuľku rozloženia
- Formuláre
- Tlačidlá pre navigáciu

Používateľské rozhranie je implementované ako jeden dokument html, nad ktorým sa funkcionality vykonáva pomocou javascriptov.

5.1.7 Základné rozloženie prvkov

Základné rozloženie prvkov je realizované pomocou tabuľky, ktorá má definovanú premenlivú šírku a výšku. Tieto atribúty sa prispôbujú rozlíšeniu prehliadača. Relatívna veľkosť je dosiahnutá použitím atribútu `width="100%" height="100%"`. Implementácia základného rozloženia je nasledovná:

```
<table width="100%" height="100%" border="0"> // zakladna deliaca tabulka
<tr> <td>
    <div id="o3d" style="width:100%;height: 100%;"/>
</td>
<td width="200" valign="top">
    <table width="100%" border="0" align="left" valign="top"> // tabulka menu
    <tr valign="top">
        <td>
        </td>
    </tr>
    <tr valign="top">
        <td valign="top">
        </td>
    </tr>
    <tr>
        <td valign="top" height="50">
        </td>
    </tr>
    <tr>
        <td>
        </td>
    </tr>
    <tr>
        <td>
        </td>
    </tr>
    </table>
</td>
</tr>
</table>>
```





Úrovňovo vyššia tabuľka rozdeľuje dokument na 2 časti a jej šírka s výškou sú 100% výšky a šírky rozlíšenia prehliadača. V prvej bunke je zadaná oblasť, v ktorej sa zobrazuje O3D model budovy FIIT. V druhej bunke tabuľky je samotné používateľské rozhranie.

5.1.8 Mód prehliadania a navigácie

Mód pre prehliadanie a navigáciu modelu majú základnú štruktúru ako v ukážke X znázornenej vyššie. Ako je spomenuté, dokument je spoločný pre každý mód a jednotlivé časti sú zviditeľňované alebo skrývané pomocou funkcií. Časti sú uzavreté tagmi `<div></div>`, u ktorých sa mení cez funkciu parameter *visibility*.

```
<div id="search" style="visibility:visible;">
  <div id="upper_form" style="visibility:hidden;">
  .
  .
  .
</div> <hr>

<div id="lower_form" style="visibility:hidden;">
.
.
.
</div>
</div>
```

Voľba módu zobrazenia je implementovaná ako súbor obrázkov, ktorým je zadané správanie pri kliku na nich tagom *onclick*.

```
<font color="white">Menu zobrazenia:</font>
<br>





```

Po kliknutí na tlačidlo sa volá príslušná funkcia, ktorá ma na starosti prácu s 3D modelom a zmenou atribútu *visibility*. Príklad takejto funkcie je uvedený nižšie.

```
function initModeExplore(){
```





```
document.getElementById("buttonExplore").src = "../img/explore3.png";
document.getElementById("buttonMode3D").src = "../img/3d.png";
document.getElementById("buttonNavigate").src = "../img/navigate.png";

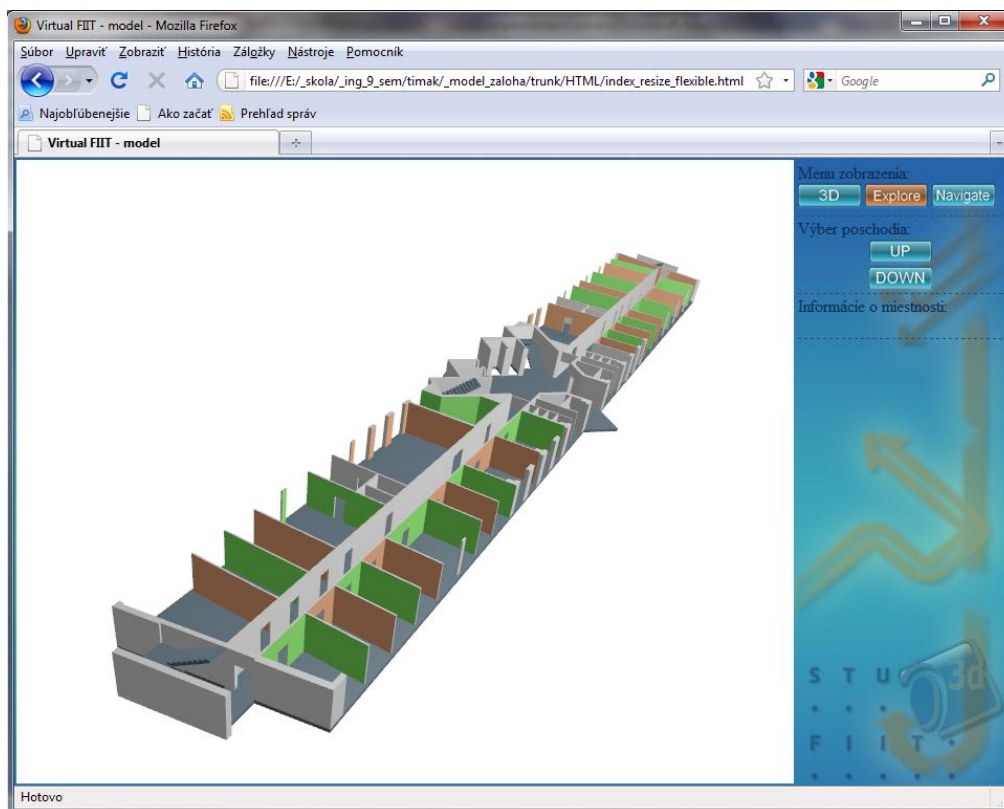
document.getElementById("info").style.visibility="visible";
document.getElementById("search").style.visibility="hidden";
document.getElementById("upper_form").style.visibility="hidden";
document.getElementById("lower_form").style.visibility="hidden";

g_modeManager.setCurrentMode("explore");

}
```

Do používateľského rozhrania sú implementované aj ostatné grafické prvky (pozadia a logá). Tie slúžia k estetickému vzhľadu rozhrania, ktoré je celé ladené v modrej farbe.

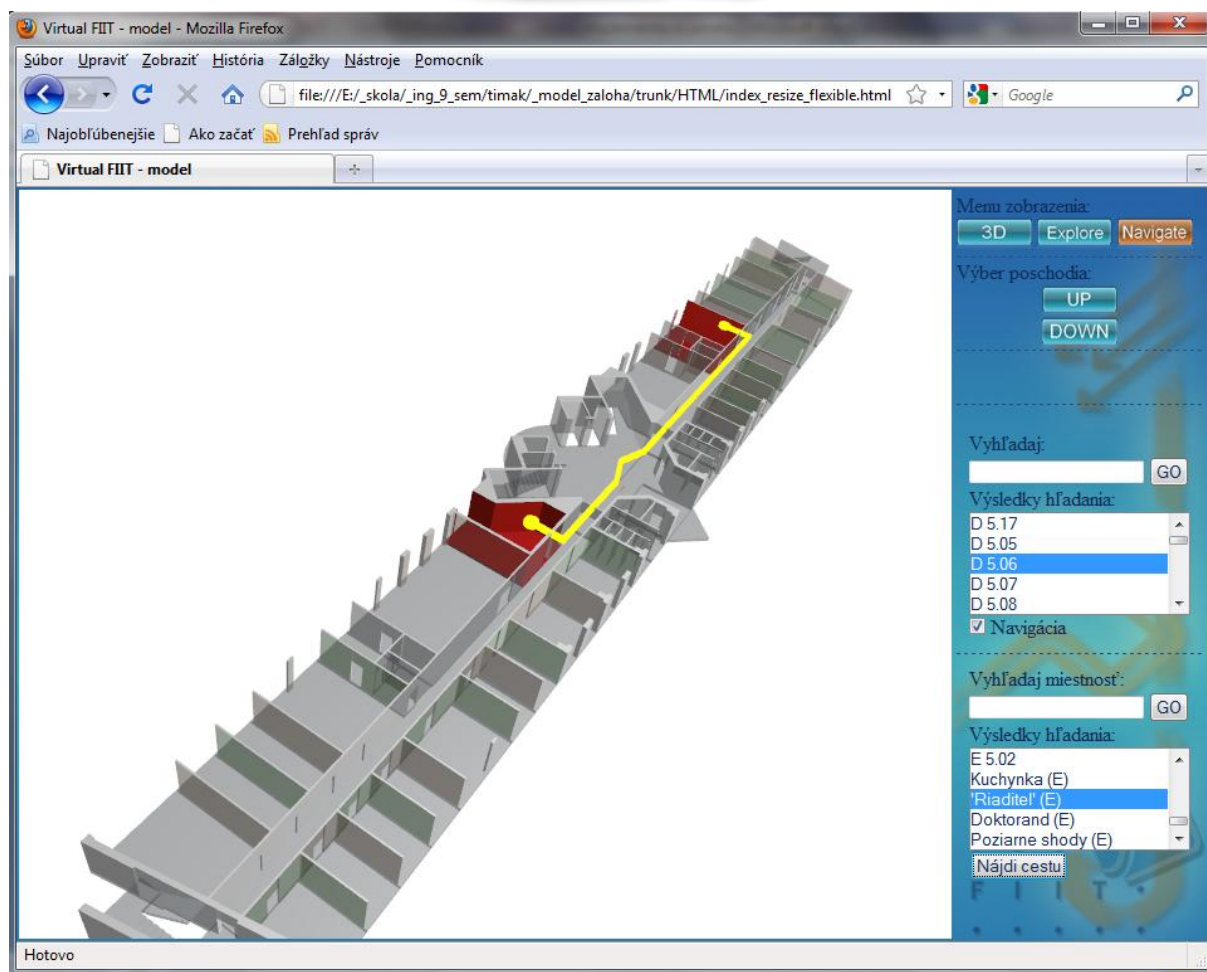
Mód prehliadania obsahuje navigáciu pre zmenu módu zobrazenia a obsahuje prvky pre voľbu poschodia modelu. Ostatné prvky ako textové polia pre navigačný mód sú skryté. Obrázku s 3D modelom je možné ovládať (posun, zoom, označenie miestností). Po označení konkrétnej miestnosti sa pod názvom „Informácie o miestnosti“ zobrazia informácie o miestnosti z databázy.



Obr. 48 Obrázok módu prehliadania

Po prechode na mód navigácie sa časti s atribútmi *visibility* nastavujú z hodnoty *hidden* na hodnotu *visible*. Výsledná obrazovka je na obrázku nižšie.





Obr. 49 Obrazovka módu navigácie





5.2 Implementácia prototypu modelu

Cieľom tvorby prototypu bolo overenie kritických častí projektu virtuálnej budovy FIIT. Overenie a oboznámenie sa s použitím navrhovanej technológie, s ktorou máme minimálne skúsenosti, bolo kľúčové a má zásadný vplyv na úspešnosť projektu.

5.2.1 Modelovanie poschodia

Pri modelovaní prototypu poschodia sa nástroj 3DS Max osvedčil ako postačujúci s dostatočnou funkcionalitou a výkonnosťou. Počas tvorby modelu nastali tieto problémy, ktoré sme museli vyriešiť:

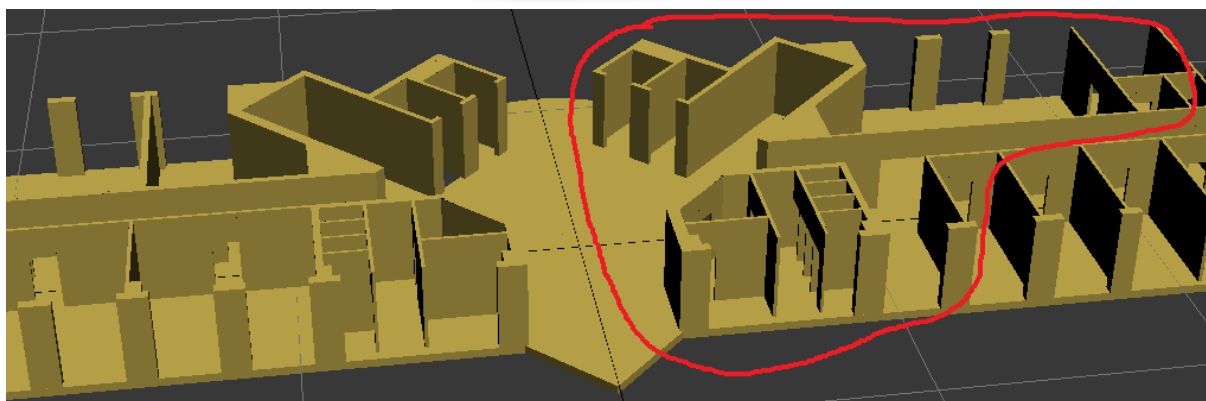
- priesvitnosť stien miestností pri zrkadlení zoskupených objektov
- zarovnávanie vyrezaných alebo pootočených objektov o špecifické uhly (10°, 16°, 45°)
- chybný export komplikovanejšieho modelu do formátu DAE v 3DS Max

Problém priesvitných stien nastal po zobrazení modelu v O3D (Obr. 1), pritom 3DS Max ich zobrazil správne (Obr. 2). Týkalo sa to stien, ktoré boli zrkadlené cez x-ovú os. Jednoduchý nápad riešenia bol párný počet zrkadlení cez kombináciu osí. Toto však nefungovalo. Nakoniec bolo nutné zrkadlený zoskupený objekt oddeliť, spätne zrkadliť a správne pootáčať jednotlivé časti objektu. Potom bolo zobrazenie celého modelu správne v 3DS Max aj v O3D. Odporúčam sa zrkadleniu podľa možností vyhnúť.



Obr. 50 Chybné zobrazenie modelu v O3D – priehľadné steny





Obr. 51 Zobrazenie modelu v 3DS Max – priehľadnosť stien sa neprejavila

Zarovnanie upravených objektov bolo vyriešené viacnásobným zarovnaním po jednotlivých osiach alebo druhou možnosťou bolo pootočenie celého modelu o požadovaný uhol. V najzložitejších prípadoch je možné použiť pomocný objekt.

Pri komplikovanejšom modeli boli v 3DS Max problémy s exportom do formátu DAE. Následne nebolo možné tento model zobraziť v O3D. Problém vyriešilo použitie nástroja OpenCOLLADA na konvertovanie modelu, ktorý bol doinštalovaný do 3DS Max.

Ďalšou podstatnou časťou overenia, ktorú nám prototyp modelu poskytol, je modelovanie poschodia skladaním objektov a nie nakreslenie pôdorysu a jeho vytiahnutie do priestoru. Spôsob vyskladania modelu z objektov je použiteľný a prináša aj výhodu ľahšej modifikácie pre zvyšné poschodia s rovnakým podlažím. Je síce časovo náročný, ale poskytuje požadované možnosti (komponentovanie miestností kvôli interaktívnemu vyhľadávaniu).

5.2.2 Práca s modelom v O3D

Implementáciu projektu sme sa rozhodli začať tvorbou prehliadacieho módu, v ktorom je zobrazený model poschodia. V tomto móde malo byť možné model poschodia otáčať a kliknutím do modelu vyberať konkrétne miestnosť. Pri implementácii sme často vychádzali z existujúcich príkladov (ukážok) práce s O3D, ktoré sa nachádzajú na oficiálnej stránke O3D.

Po úspešnom načítaní modelu poschodia do prostredia O3D sme sa rozhodli najskôr implementovať otáčanie modelu. Pri jeho tvorbe sme vychádzali z príkladu „Home configurator“, ktorý obsahoval niekoľko spoločných prvkov s našim projektom. Po dôkladnej analýze zdrojového kódu príkladu „Home configurator“ sme boli schopní vytvoriť prototyp s našim modelom poschodia s možnosťou otáčania a približovania.

V ďalšom kroku sme implementovali picking. Cieľom pickingu je zistiť názov komponentu (v našom prípade miestnosti), na ktorý užívateľ klikol. Vychádzali sme z príkladu nazvaného „Picking“. Po jeho úspešnej implementácii sa však vyskytli nezanedbateľné výkonnostné problémy, ktoré spôsobovali dlhé (rádovo sekundové) zisťovanie názvu





komponentu, na ktorý užívateľ klikol. Ako sme bližšou analýzou zistili rýchlosť pickingu závisí od dvoch hlavných faktorov.

Prvým faktorom bol spôsob práce s modelom. Pri implementácii otáčania modelu sme vzhľadom na počiatok súradnicovej sústavy otáčali celý model. Pre správne fungovanie pickingu bolo preto potrebné pri každom kliknutí obnoviť strom, z ktorého boli vyberané objekty, čo sa nepriaznivo odrazilo na dĺžke odozvy. Tento problém sme vyriešili tým, že sme namiesto otáčania modelu menili len pozíciu a smer kamery. Strom pickingu následne stačilo obnoviť len raz po načítaní modelu. Kvôli tejto zmene sme však museli prepísať celý kód otáčania modelu.

Druhým faktorom ovplyvňujúcim rýchlosť pickingu, bolo množstvo objektov, z ktorých sa zisťoval objekt, na ktorý bolo kliknuté. Tento problém sme vyriešili rozdelením modelu na tri časti:

1. komponenty miestností, na ktoré sa dá kliknúť
2. komponenty častí poschodia, na ktoré sa nedá kliknúť (rôzne priečky a horné krytie modelu)
3. komponenty navigácie

Po tom, ako sme na tvorbu pickingu využili len komponenty miestností, na ktoré sa dá kliknúť, sme zaznamenali asi o 50% lepšiu odozvu.

Ďalšou výzvou, s ktorou sme sa pri práci s modelom poschodia stretli, bola priehľadnosť. Ak sme v 3DS Maxe nastavili stenám priehľadnosť, v prostredí O3D sa priehľadnosť neprejavila. Ako sme neskôr zistili, O3D z neznámych dôvodov zaradilo materiály do „Performance Draw List“, ktorý priehľadnosť nepodporuje. Problém sme vyriešili manuálnym umiestnením materiálov do „ZOrdered Draw List“.

5.2.3 Implementácia navigácie

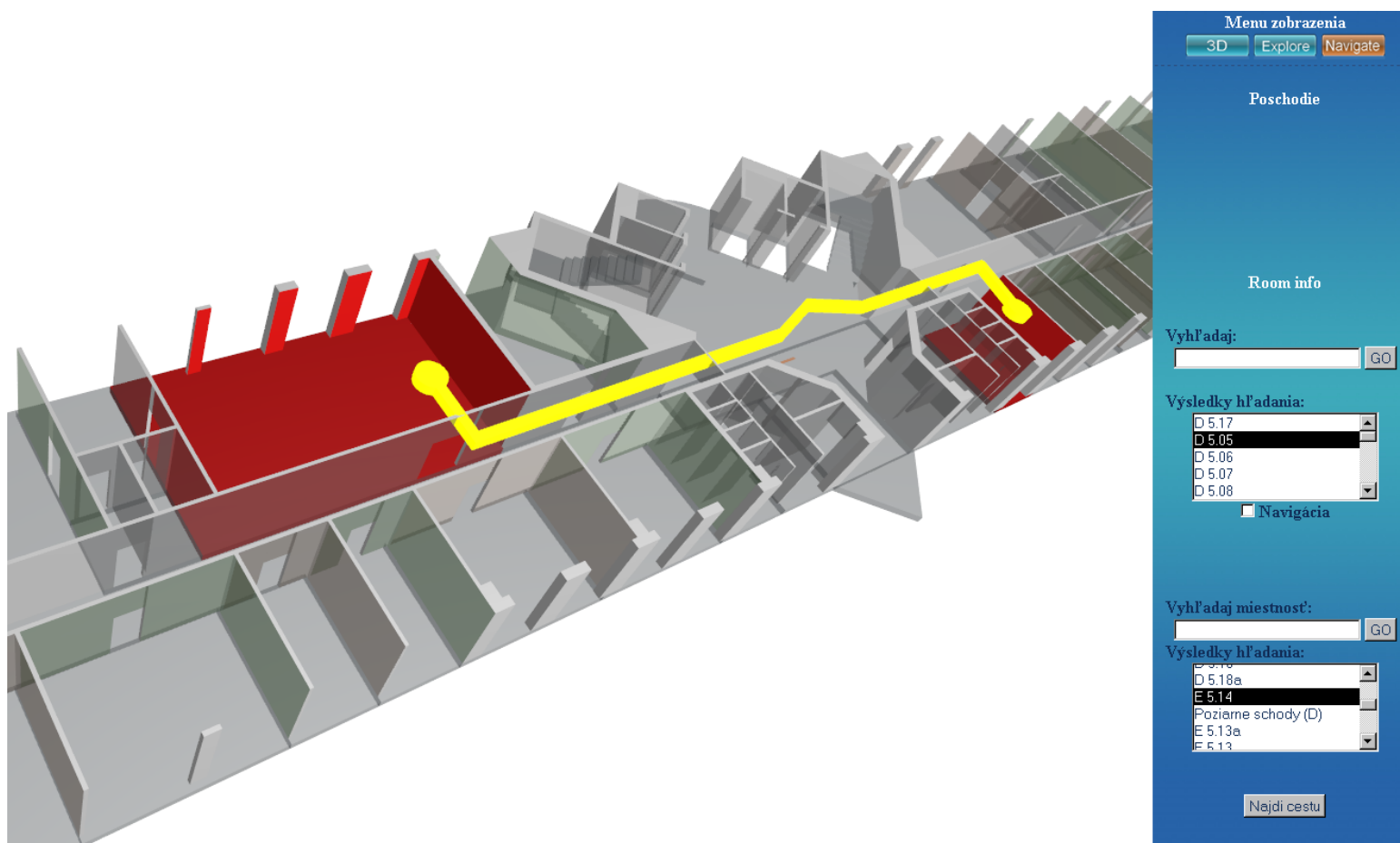
Jedným z troch funkčných pilierov Virtuálnej FIIT je navigácia v modeli budovy. Navigácia sa realizuje vyznačením cesty medzi dvomi bodmi – začiatkovej a cieľovej miestnosti – pomocou tzv. navigačných hrán (podobne ako na Obr. 3). Implementácia zahŕňala najmä dve dôležité funkcie: nájdenie najkratšej cesty v grafe miestností a zobrazenie výsledku.

Vyhľadávanie cesty sa robí nad grafom, ktorý zodpovedá topológii miestností vnútri budovy (uzly grafu sú miestnosti a sú spojené hranou, ak sa dá medzi nimi voľne prejsť, napr. dverami. Hrany majú navyše pridelenú váhu – dĺžka cesty v metroch). Tento graf je uložený v samostatnom XML súbore. Aplikácia pri načítaní stránky na strane klienta načíta súbor a vytvorí svoju programovú reprezentáciu grafu, využíva na to triedy `Graph`, `Node` a `Edge`. S týmto grafom sa pracuje počas behu programu (do ďalšieho načítania stránky). Zvláštnosťou tohto grafu je, že okrem skutočných miestností obsahuje ešte ďalšie, pomocné uzly. Tie sa nachádzajú na všetkých „križovatkách“ vnútri budovy, t.j. bodoch, z ktorých sa





možno vydať viacej ako dvomi smermi (napr. z chodby možno vstúpiť do viacerých miestností, preto je pomocný uzol na chodbách, vestibuloch a pod.)



Obr. 52 GUI - Navigácia medzi dvomi miestnosťami

Na samotné navigovanie bol použitý Dijkstrov algoritmus na nájdenie najkratšej cesty v grafe. Vyhľadanie cesty je funkciou grafu, realizuje ju metóda `Graph.GetShortestPath`, ktorá berie ako vstupy názov začiatkovej a koncovkej miestnosti a vracia pole objektov typu `Edge`.

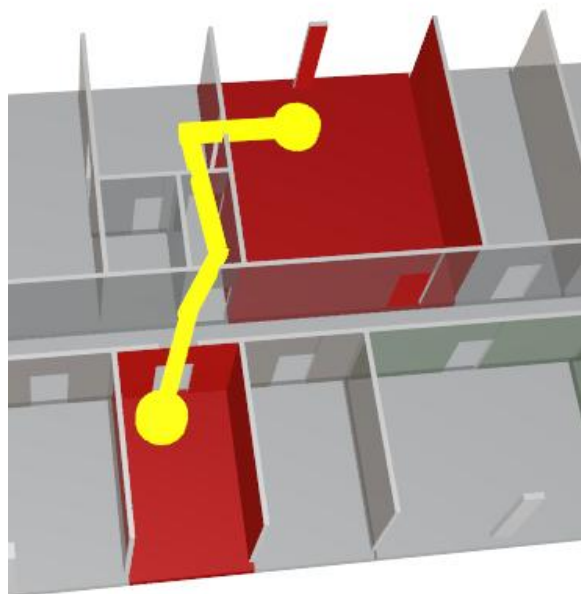
Po nájdení najkratšieho prechodu sa výsledok premietne do okna s O3D modelom. Pre každú hranu výsledného poľa sa vyberie zodpovedajúci komponent hrany v 3D modeli a zobrazí sa. Pri každej novej požiadavke navigácie sa hrany predošlej navigácie odstránia a zobrazia sa nové.

Počas implementácie sa nevyskytli žiadne ťažkosti. Prvé testovanie odhalilo menej závažnú chybu v návrhu: pri navigácii sa nájde najkratšia cesta medzi dvomi bodmi, tá však nie je vždy najvýhodnejšia. Táto nehoda vzniká pri miestnostiach, ktoré majú dva prístupy, z ktorých jeden je z chodby a jeden zo susednej miestnosti. V takom prípade môže najkratšia cesta viesť cez susednú miestnosť (ako na Obrázku 46). Takéto „skracovanie“ nie je korektné,





navigácia musí viesť cez verejné priestory. Tento problém bol vyriešený tak, že sa upravila váha koncovkej hrany vstupujúcej do izby z inej („neverejnej“) miestnosti. Váha bola zvýšená tak, aby pri navigácii z inej ako susednej miestnosti bol uprednostnený vstup z chodby, ale pri prechode medzi susedmi boli použité vnútorné hrany. Význam váhy hrany sa tým jemne zmenil z *vzdialenosti* na *dostupnosť*, na funkcionality to ale nemá žiadny vplyv.



Obr. 53 Chyba pri navigácii – skracovanie cez neverejné priestory

5.2.4 Komunikácia s databázou

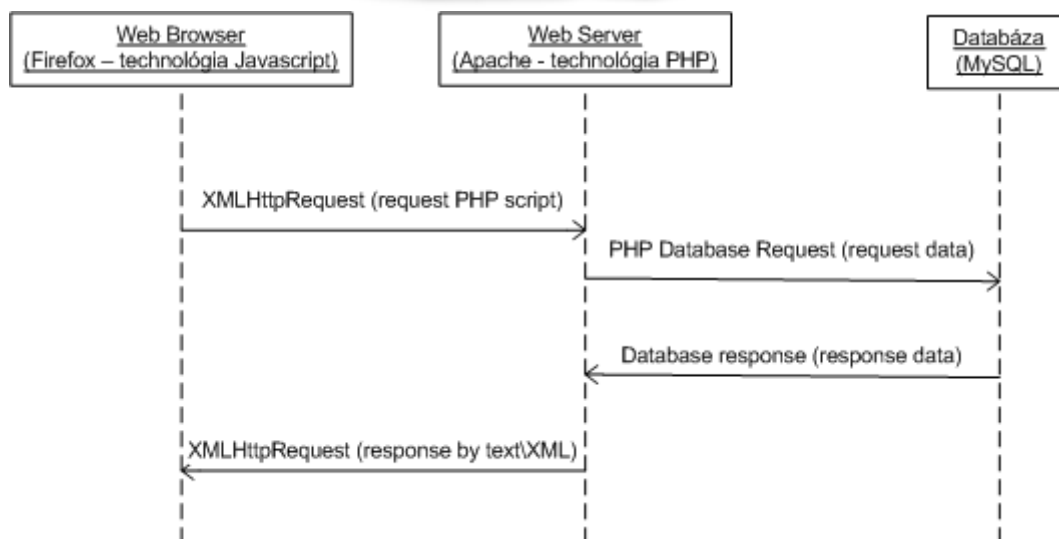
Tretím krokom k posúdeniu realizovateľnosti nášho projektu bolo odskúšanie spojenia prototypu s databázou údajov. Na tento účel sme sa rozhodli pre spojenie nasledovných technológií:

- MySQL – databázový server
Verzia – MySQL 5.1.41
- PHP – skriptovací jazyk na strane servera
Verzia – PHP 5.3.1
- Javascript – skriptovací jazyk na strane klienta

Spojenie technológií Javascript a PHP alebo iného skriptovacieho jazyka na strane klienta je tiež známe ako Ajax. Prístup využíva XMLHttpRequest objekt pre komunikáciu so serverom.

Ich komunikáciu znázorňuje obrázok 47.





Obr. 54 Diagram komunikácie prototypu s databázou

Proces komunikácie sme overili nasledovne:

1. Proces je spustený pri vzniku požiadavky na informáciu. Táto požiadavka je zachytená v skriptovacom jazyku na strane klienta. Konkrétne je reprezentovaná spustením javascript funkcie, ktorú pre tento príklad nazveme **setRoomInfo(text)**.
2. Funkcia **setRoomInfo(text)** má za úlohu vyplniť popis miestnosti, pričom ako parameter dostane názov miestnosti.
 - a. Vyplní xmlHTTP objekt. Tento objekt je vytvorený v javascripte na stránke a je perzistentný aj po skončení funkcie. Je potrebný pre komunikáciu so serverom a nie všetky prehliadače ho podporujú.
 - b. Vyskladá url. Tá obsahuje adresu PHP skriptu, ktorý bude volaný na strane servera, ale aj parameter **text**.
 - c. Definuje funkciu, ktorá má byť spustená pri zmene stavu requestu. Nazveme ju **setRoomInfoStateChanged()** Tá najzaujímavejšia zmena pre nás nastane, keď sa úspešne vráti zo servera s odpoveďou.
3. PHP skript na strane servera má za úlohu pripojenie k databáze a získanie výsledkov. Je tu definované pripojenie do databázy ako aj samotná štruktúra požadovaných dát (SQL select).
4. Funkcia **setRoomInfoStateChanged()** sa postará o získanie výsledkov z objektu ktorý sa vrátil ako odpoveď od PHP skriptu. Výsledný text premietne do prototypu.





5.3 Implementácia navigácie – graf miestností

Navigácia vo VFIT je vyvíjaná v troch krokoch:

- vytvorenie navigačných hrán
- zapísanie topológie hrán do samostatného XML súboru (`RoomGraph.xml`)
- implementácia načítania rozmiestnenia hrán a nájdenia najvýhodnejšej cesty v grafe podľa používateľových vstupov

Prvé dve fázy prebiehajú neustále počas vývoja; po vytvorení modelu ďalšej časti budovy sa doňho doplnia navigačné hrany a rozšíri sa `RoomGraph.xml`. Nasledujúci text hovorí práve o strednej fáze vývoja navigácie.

5.3.1 Umiestnenie

XML súbor je umiestnený v priečinku HTML spolu s webovou časťou projektu. Dôvodom je použitie JavaScript objektu `XmlHttpRequest` na načítanie grafu. V prehliadači Mozilla Firefox totiž nie je možné týmto spôsobom dostať obsah externého súboru, ak sa nachádza na inej doméne (za inú doménu je v prípade relatívneho odkazovania považovaný aj iný priečinok).

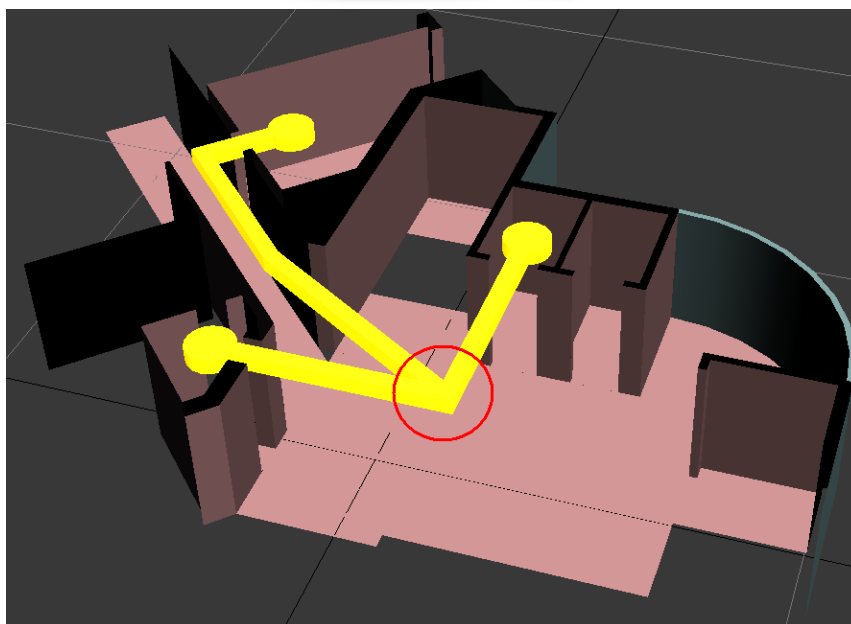
5.3.2 Štruktúra

Korektný obsah má štruktúru XML elementov naznačenú na nasledujúcom príklade:

```
<graph>
<!-- XXXXXXXXXXXXXXXXXXXXXXXX 1. poschodie XXXXXXXXXXXXXXXXXXXXXXXX-->
  <node id="room102" name="1.02" isRoom="true" />
  <node id="room102x" isRoom="false" />
  <edge id="edge_1" room1="room102" room2="room102x" length="3.3"/>
</graph>
```

Koreňový element `graph` obsahuje len dva typy nasledovníkov: `node` – vrchol grafu miestností, a `edge` – hrana spájajúca dva vrcholy. Atribút `isRoom` rozlišuje uzly grafu, ktorým zodpovedá skutočný komponent miestnosti v modeli (`true`) od pomocných uzlov (`false`). Pri `node`-och komponentov ich `id` má rovnakú hodnotu, ako je názov komponentu. Zároveň atribút `name` nesie názov zodpovedajúcej miestnosti (použité boli názvy z DWG plánov). Pomocné vrcholy (nemajú atribút `name`) sú doplnené všade tam, kde sa stretávajú komponenty hrán, podobne ako na obrázku 1. Pre pomenovanie týchto vrcholov nie je žiadne pravidlo, je len potrebné zachovať konzistenciu pri neskoršom definovaní hrán. Elementy hrán (`edge`) majú `id` podľa názvu príslušného komponentu hrany. `room1` a `room2` označujú koncové uzly hrany (atribút `id` elementu `node`. Oba uzly musia byť definované predtým, inak načítanie grafu skončí chybou), `length` je dĺžka v metroch (výnimky, kedy `length` nezodpovedá dĺžke hrany sú vysvetlené v).





Obr. 55 Na spoji navigačných hrán nie je žiadny vrchol – komponent.

Do RoomGraph.xml ale pribudne umelý vrchol, aby bolo možné vykonávať navigáciu.

Okrem uvedenej hierarchie XML elementov, je RoomGraph.xml štruktúrovaný aj na vyššej úrovni. Jediný účel tejto vyššej štruktúry je zvýšenie prehľadnosti pre implementátora.

- Vrcholy a hrany sú zoskupené po poschodiach, každé poschodie je oddelené výrazným komentárom (viď príklad vyššie). Poschodia idú porade od najnižšieho po najvyššie.
- V rámci sekcie jedného poschodia sú definované najprv uzly so zodpovedajúcimi komponentmi izieb, potom pomocné uzly a nakoniec hrany. Každá z týchto troch častí je navyše ešte rozdelená podľa traktov (krídel), väčšinou na ľavý, pravý a stredný trakt.

5.3.3 Ďalšie pravidlá

Pre zjednodušenie práce vývojára boli tiež zavedené postupy pre pomocné uzly, ktoré sú na dotykoch hrán dvoch poschodí (každá v inom modeli, napr. v jednom ako hrana dolu schodmi a v nižšom modeli ako hrana hore tým istým schodiskom. Po spojení modelov v aplikácii by sa mali dotknúť):

- V sekcii každého poschodia sú definované pomocné uzly smerom dole, najnižšie poschodie tak nebude mať žiadne. Naopak spojnice smerom hore je očakávaná v sekcii vyššieho podlažia.
- Ak sa súbor rozširuje o také poschodie, že v RoomGraph.xml zatiaľ nie je najbližšie vyššie poschodie, spojnice tohto chýbajúceho poschodia smerom





nadol sú dočasne vložené do sekcie definovaného nižšieho poschodia a komentárom jasne označené, že je potrebné ich presunúť po rozšírení XML súboru. Bez tohto kroku by načítanie grafu skončilo chybou.





6 OPIS REALIZÁCIE

6.1 Modelovanie poschodí v 3DS Maxe

6.1.1 Proces vymodelovania finálneho modelu poschodia

Najskôr je potrebné vymodelovať podľa dostupných .dwg výkresov model typu „planes“, pričom po samotnom vymodelovaní sú do modelu pridané ešte navigačné hrany, ktoré sú navyše presne nazvané na základe názvov miestností a blokov v .dwg výkresoch. Tento prvý krok vymodelovania poschodia je spravidla časovo najnáročnejší. Následne sa súbor modelu skopíruje a označí „components“, pričom nastáva fáza komponentovania navigačných hrán, miestností a zvyšných objektov. Pokračuje sa označením názvu komponentov miestností podľa označenia komponentov navigačných hrán. Nasleduje fáza aplikovania materiálov objektom modelu. Nakoniec je model optimalizovaný a je z neho vygenerovaná mapa poschodia, ktorá definuje pohyb po budove. Mapa musí byť ešte ručne upravená. Percentuálny odhad podielu časovej náročnosti jednotlivých fáz procesu je uvedený v Tab. 10. Tabuľka ďalej hlavne obsahuje stav jednotlivých častí (poschodí) modelu po dvoj – semestrálnej práci.

Model poschodia	Vymodelovaný	Skomponentovaný	Označené miestnosti	Optimalizovaný	Mapa poschodia	Stav	Odhad náročnosti
1PP	A	A	A	A	A	F	*****
1NP	A	A	A	A	A	F	*****
2NP	A	A	A	A	A	F	***
3NP	N	5NP copy	A	A	A	F	*
4NP	N	5NP copy	A	A	A	F	*
5NP	A	A	A	A	A	F	***
6NP	A	A	A	A	A	F	***
7NP (strecha)	A	A	A	A	A	F	**
2PP	N	N	N	N	N		**
Strecha (nad 1PP)	N	N	-	N	-		*





Odhad časovej náročnosti	75%	15%	4%	2%	4%	-	-
---------------------------------	------------	------------	-----------	-----------	-----------	---	---

Tab. 10 Aktuálny stav modelov poschodí

Vysvetlivky:

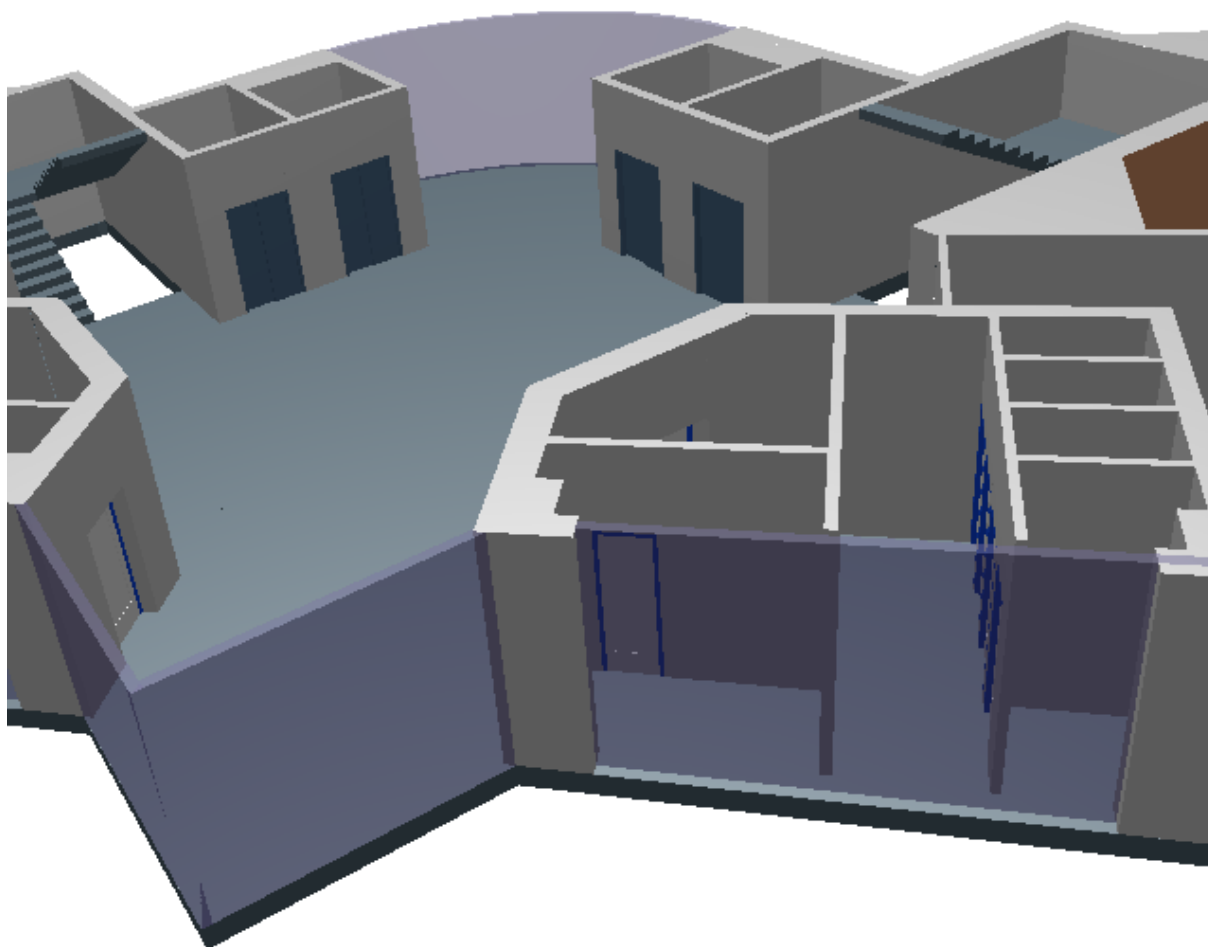
- PP – podzemné podlažie
- NP – nadzemné podlažie
- A – áno
- N – nie
- F – finálny model
- Čím väčší počet hviezdičiek (*), tým väčšia náročnosť modelu

6.1.2 Modelovanie špecifických objektov

6.1.2.1 Okná

Pri modelovaní okien je abstrahované od presných výkresov, pričom jednotlivé steny s oknami sú vyplnené plochami s aplikovaným priehľadným materiálom . Výhodou je efektívnejšie a rýchlejšie modelovanie pri zachovaní dobrého a moderného vzhľadu budovy.





Obr. 56 Abstrakcia okien modelu na priehľadné plochy

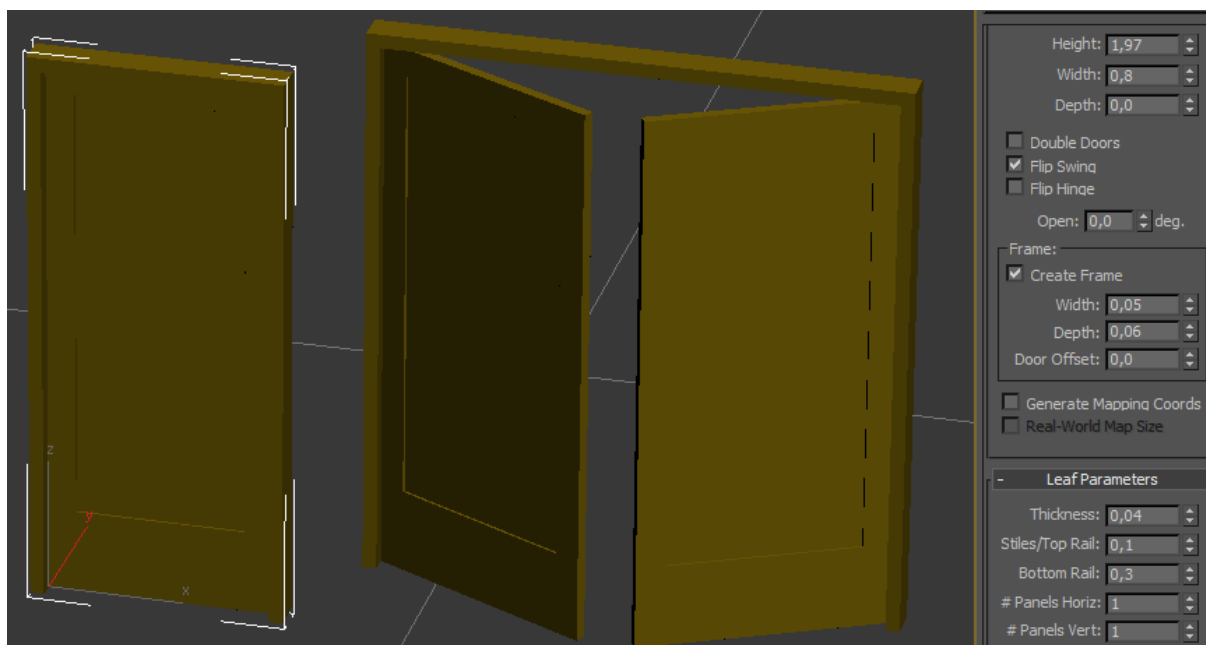
6.1.2.2 Dvere

Dvere sú do modelu vložené ako jeden komponent. Potom je nutné nastavenie všetkých potrebných parametrov ako:

- Rozmery dverí a rámu
- Typ dverí – jedno alebo dvojkridlové

Potom je nutná nevratná konverzia na „EditablePoly“ a oddelenie (Detach“) elementov tvoriacich dvere a rám. Pri dvojkridlových dverách je vždy nutné nastaviť pri druhom krídle počiatok ich súradnicovej sústavy k pántom dverí, aby ich otváranie v O3D prebiehalo korektné. Táto úprava ale po konverzii modelu do formátu o3dtgz nefungovala, a preto sme problém vyriešili odstránením chybného druhého krídla dverí a nahradzujeme ho prvým. Túto kópiu je nutné kvôli korektnému otáčaniu pri pántoch zárubní ešte prevrátiť o 180° v smere osi x a aj z.

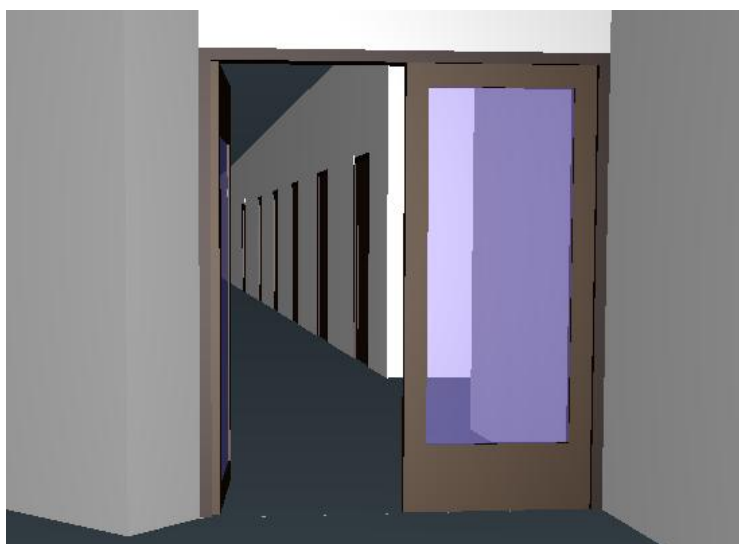




Obr. 57 Nadstavene parametrov komponentov dverí

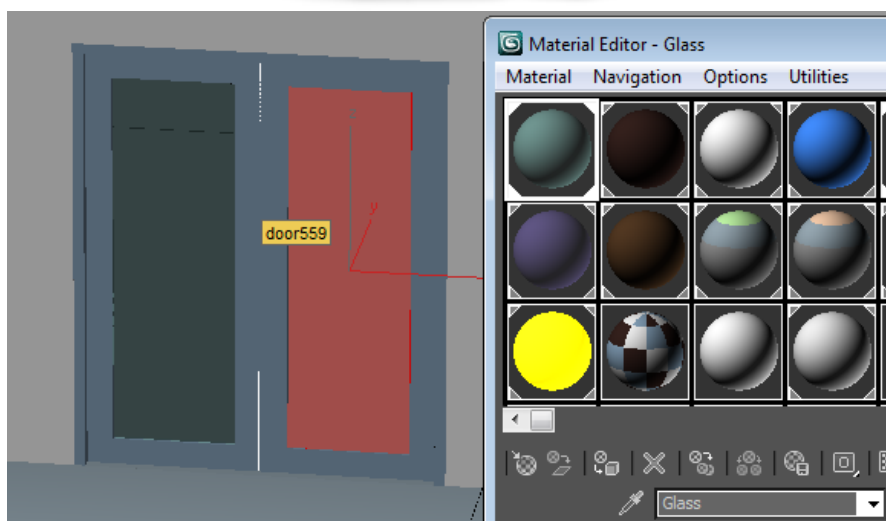
6.1.3 Aplikovanie materiálov elementom komponentu

Pre dosiahnutie priehľadnej výplne dverí v O3D (obr. 58) je nutné aplikovanie priehľadného materiálu daným elementom komponentu v 3DS Max (obr. 59). Potom je ešte nevyhnutné programovo zaradiť daný typ materiálu v O3D medzi priehľadné.



Obr. 58 Dvere s priehľadnou výplňou



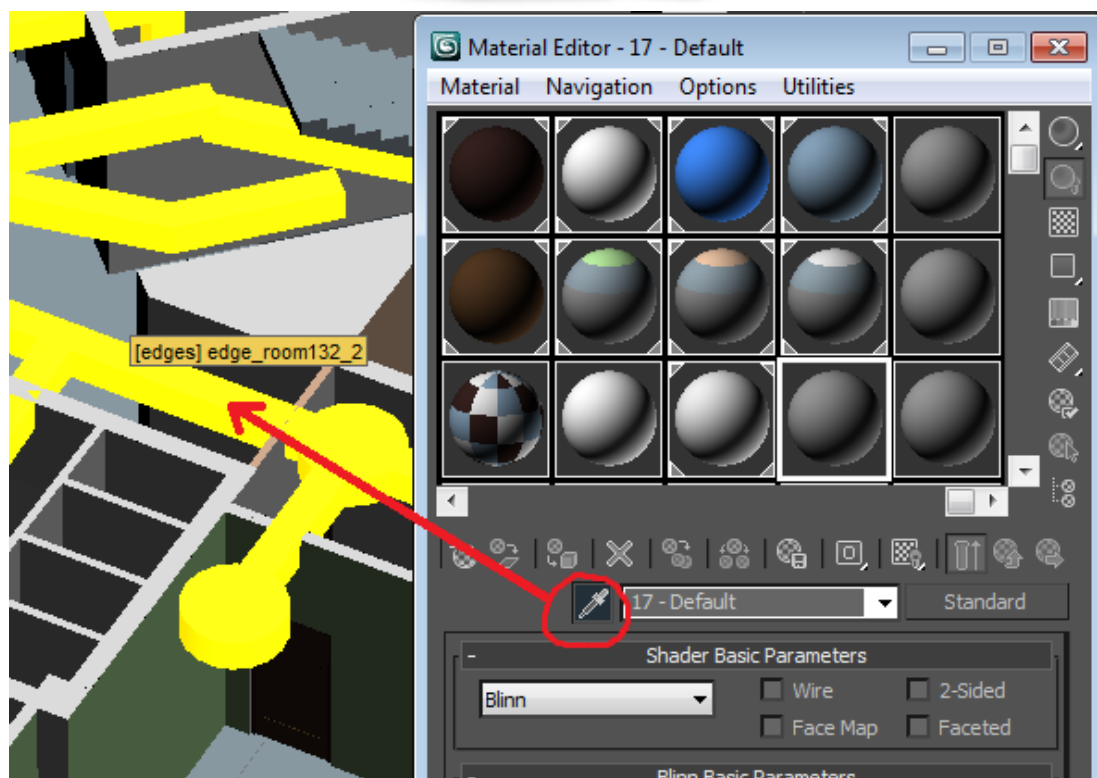


Obr. 59 Aplikovanie priehľadného materiálu elementu dverí

6.1.4 Materiály

Pri vytváraní „floor_xxP_components.max“ modelov poschodí z ich príslušných „floor_xxP_planes.max“ súborov je nutné definovať všetky používané materiály. Ručné zadávanie parametrov materiálov je zdĺhavé, a preto existuje aj jednoduchší spôsob – získanie parametrov materiálu odobratím jeho vzorky z komponentu (Obr. 609). Predtým je však nutné do modelu vložiť objekty, ktoré daný materiál majú aplikovaný. Toto môžeme realizovať „Merge“ (zlučovanie) funkcionalitou importu 3DS Maxu, kedy do daného modelu vložíme objekty z iných súborov modelov.





Obr. 60 Získanie materiálu odobratím jeho vzorky z komponentu

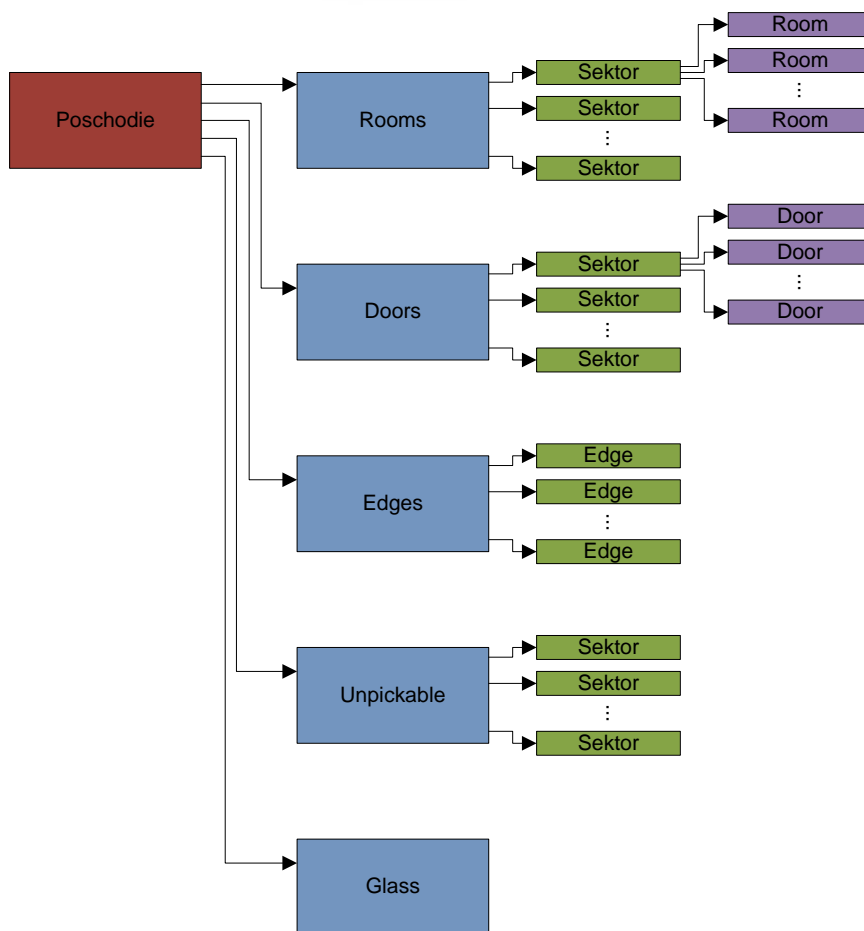
6.1.5 Strecha

Pri poschodiach, ktoré nemajú nad sebou iné poschodie, ale sú pokryté strechou, tak túto strechu budeme modelovať v rámci daného poschodia, pričom v prehliadacom a navigačnom móde (tlačidlá „Náhľad“ a „Naviguj“) bude strecha neviditeľná (skrytá) pre používateľa. Strecha sa týka poschodí 1.PP a 7.NP. Strecha sa budú v O3D zobrazovať len pri prepnutom zobrazovacom móde „3D“.

6.2 Organizácia komponentov v modeli

Organizácia komponentov v modeli je potrebná najmä z dôvodu zabezpečenia funkcionality zvýrazňovania vybraných miestností (Rooms), otvárania dverí (Doors), zobrazenia prepojení miestností (Edges) a spriesvitnenia presklených prvkov poschodia (Glass). Prakticky to znamená, že podobné komponenty sú už pri modelovaní sústredené do jednej skupiny. Všetky takéto zoskupenia komponentov tvoria jedno poschodie. K takémuto rozdeleniu sme sa rozhodli pre jednoduchšiu prácu v prostredí O3D, v ktorom sa dá pracovať aj len s jednotlivými podstromami modelu a nemusia sa pritom prehľadávať nerelevantné časti, čo zvyšuje rýchlosť aplikácie. Kvôli zabezpečeniu ešte vyššieho výkonu vytvárajúcej aplikácie sme sa rozhodli niektoré podstromy ďalej ešte deliť na sektory. Stromová hierarchia poschodia sa nachádza na obrázku 61.





Obr. 61 Hierarchia komponentov poschodia

Význam jednotlivých skupín:

Rooms

Nachádzajú sa v ňom jednotlivé izby rozdelené do sektorov. Z tejto skupiny sa robí picking (zistovanie či sa na miestnosť kliklo) a zvýrazňovanie daných miestností po kliknutí v prehliadacom a navigačnom móde. Sektory zvyšujú rýchlosť pickingu, keďže ak sa miesto kliknutia nenachádza v priestore sektoru, algoritmus nemusí zisťovať či bolo kliknuté na miestnosti nachádzajúce sa v danom sektore.

Doors

Obsahuje dvere rozdelené do sektorov. V princípe je členenie rovnaké ako u miestností. Picking dverí je využívaný na otváranie/zatváranie dverí v 3D móde.

Edges

V tejto skupine sa nachádzajú prepojenia miestností (tzv. hrany). Tieto prepojenia sú pri načítavaní modelu skrývané. Prepojenia slúžia v navigačnom móde na zobrazenie cesty medzi dvoma





miestnosťami. Keďže nikdy nie sú viditeľné všetky naraz (skryté komponenty sa nespracúvajú), nie je potrebné, aby boli členené do sektorov.

Unpickable

Obsahuje všetky časti modelu, ktoré sa nenachádzajú v ostatných skupinách. Ide najmä o rôzne priečky, stĺpy a časti múrov, ktoré nepatria žiadnej z miestností. Keďže takýchto objektov je v modeli pomerne veľa, rozhodli sme sa pre rozdelenie do sektorov.

Glass

V tejto skupine sa nachádzajú všetky presklené časti modelu (najmä okná). Komponenty tejto skupiny sú už pri načítaní modelu zaradené do `ZOrderedDrawList`, ktorý zabezpečuje priehľadnosť v ňom obsiahnutých komponentov. Keďže presklené časti sú v našom modeli zložené len z niekoľkých primitív, rozdelenie do sektorov by nemalo na výkon žiadny vplyv.

6.3 Implementácia O3D zobrazovacej časti

Časť stránky, v ktorej sa zobrazuje model virtuálnej budovy je implementovaná pomocou tagov `<div></div>` s bližšie špecifikovanými vlastnosťami v CSS kaskádových štýloch. V dokumente je táto časť implementovaná priamo do tabuľky základného rozloženia stránky. Implementácia je znázornená nižšie.

```
<div id="o3d" style="width:100%;height:100%;">
```

Lokálny štýl `width:100%; height:100%;` je definovaný na 100% šírky nadradeného prvku, čo je ľavá bunka základného rozloženia a takto je zaručená maximálna výška a šírka O3D okna.

Ďalej je tento `<div>` dodefinovaný v kaskádových štýloch nasledovne:

```
#o3d{
  margin-left: 0px;
  margin-top: 0px;
}
```

6.3.1 Implementácia menu

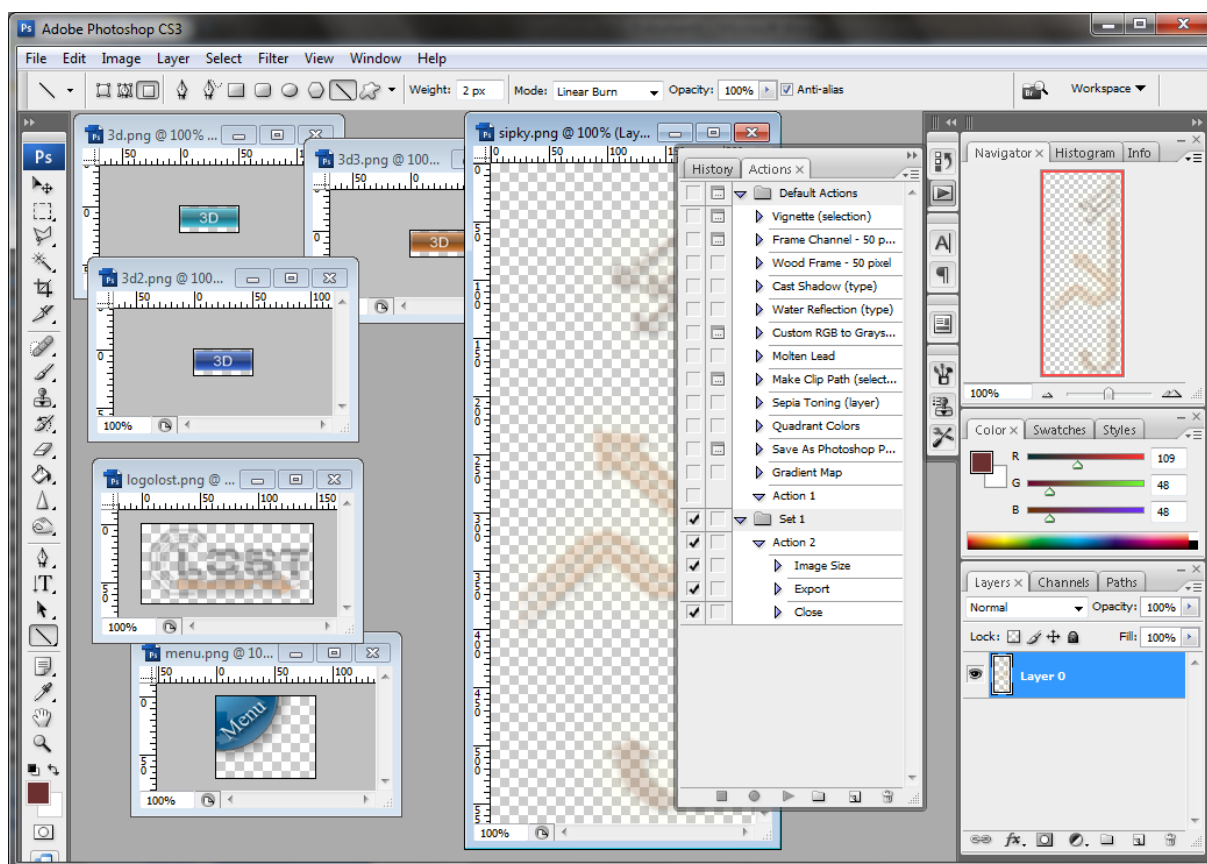
Implementácia postranného menu bola stručne popísaná v predchádzajúcej časti. Namiesto popisovania jednoduchých prvkov menu táto podkapitola rozoberá problémy spojené s implementáciou vylepšení a optimalizácie menu.





6.3.1.1 Grafické prvky menu

Všetky grafické prvky ktoré sa nachádzajú v aplikácii boli vytvorené pomocou programu Adobe Photoshop CS3. Pozadia boli vytvorené pomocou prechodov (gradients) v modrých odtieňoch. Tieto odtiene boli vybrané na základe faktu, že fakulta má z pomedzi farieb STU BA práve farbu modrú. Tlačidlá boli vytvárané od základov v niekoľkých rôznych farbách (kvôli zvýrazneniu umiestnenia kurzora nad tlačidlami).



6.3.1.2 Implementácia suggestions

Suggestions sú vybrané hodnoty textového poľa formulára, ktoré sú hľadané a automaticky dopĺňované do vyhľadávacieho poľa. Kvôli veľkému počtu nájdených výsledkov bolo potrebné naimplementovať štýly výsledkov efektívne. Vlastnosti štýlov sú v súbore *jquery.autocomplete.css* v adresári *JS/lib/jquery/*. Nájdené výsledky sú implementované ako prvky `` zoznamu `` a preto sú štýly definované ako pre samotný zoznam, tak aj pre jeho prvky. Aby výsledky neprečnievali za spodný okraj stránky a tak ju nerozširovali, bola prvkom *ac_result* (auto-complete_results) definovaná pevná výška o hodnote 350 pixelov. Ďalej sa nastavila gif animácia počas načítavania výsledkov a správanie pri prechode myšou.





```
.ac_results ul {
    width: 100%;
    list-style-position: outside;
    list-style: none;
    padding: 0;
    margin: 0;
    height: 350px;
    overflow: scroll;
}
.ac_results li {
    margin: 0px;
    padding: 2px 5px;
    cursor: pointer;
    display: block;
    width: 100%;
    font: menu;
    font-size: 9px;
    overflow: hidden;
}
.ac_loading {
    background : Window url('../img/spinner.gif') right center no-repeat;
}
.ac_over {
    background-color: #87BCC0;
    color: HighlightText;
}
}>
```

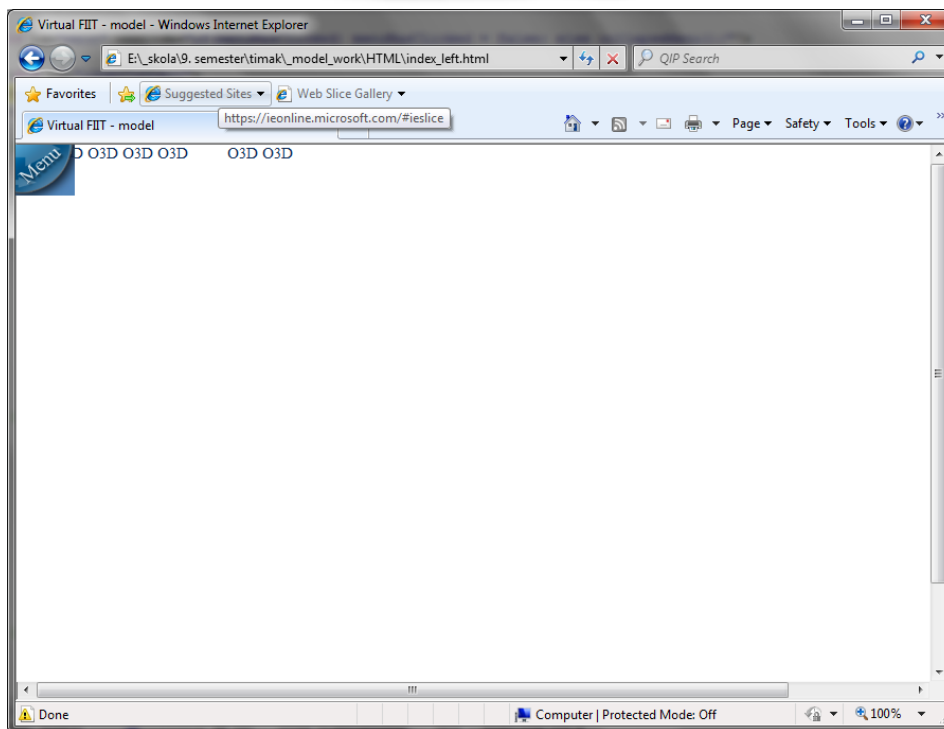
6.3.2 Implementácia vysúvacieho menu

Aby sa zabezpečilo čo najviac miesta pre zobrazovaciu plochu O3D modelu, bolo navrhnuté vysúvacie menu osadené na ľavej strane stránky. Menu bolo navrhnuté nasledovne:

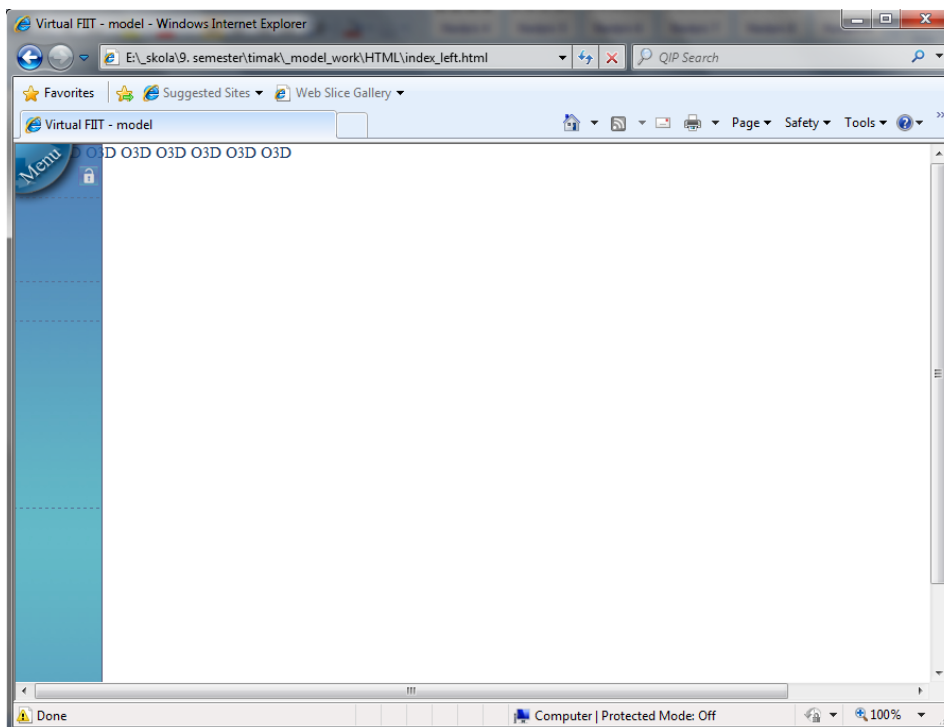
- Pri načítaní stránky sa zobrazí iba okno s O3D modelom a v ľavom hornom rohu tlačidlo menu
- Pri pozícii kurzora nad tlačidlom menu sa menu vysunie a bude nad O3D časťou
- Po prechode kurzora na O3D časť sa menu zasunie
- Menu sa dá zamknúť a O3D zmenší svoju šírku vzhľadom na menu

Príklady návrhu sú na obrázkoch nižšie.



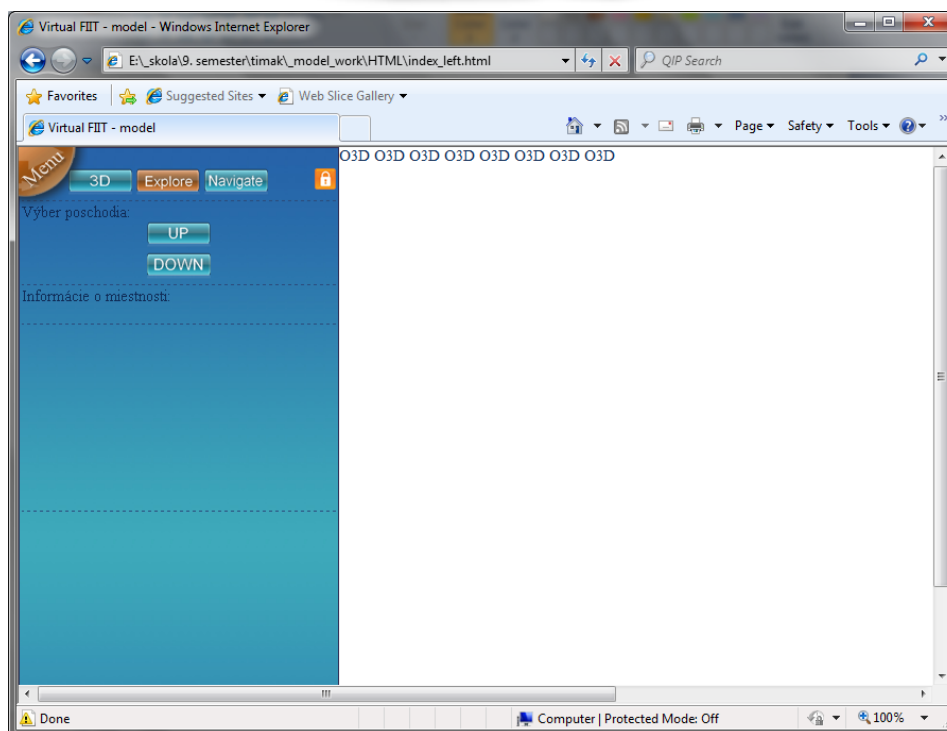


Obr. 62 Načítaná stránka so zasunutým menu



Obr. 63 Dočasne vysunuté menu (šírka vysunutia nastavená na polovicu)





Obr. 64 Pevne vysunuté menu

Prácu nad vysúvaním a manažment nad menu mali na starosti javascriptovské funkcie. Ich zdrojové kódy sú uvedené nižšie.

```

var menuWasClicked = false;
var isMenuExpanded = false;
var isMenuFocused = false;           //Ak je TRUE, tak isMenuExpanded musi byt tiez TRUE
var isMenuDocked = false;

function expandMenu()
{
if (!isMenuExpanded)
{
    var _div = document.getElementById('divSlidingMenu');
    _div.style.left = '-250px';
    setTimeout("document.getElementById('divSlidingMenu').style.left = '-140px';", 100);
    setTimeout("document.getElementById('divSlidingMenu').style.left = '0px';", 100);
    setTimeout("document.getElementById('menuHandle').style.left = '0px';", 100);
    isMenuExpanded = true;
}
}
function collapseMenu()
{
    if (isMenuExpanded)
    {
        var _div = document.getElementById('divSlidingMenu');
        _div.style.left = '-40px';
    }
}
    
```





```

setTimeout("document.getElementById('divSlidingMenu').style.left = '-140px';", 100);
setTimeout("document.getElementById('divSlidingMenu').style.left = '-220px';", 100);
setTimeout("document.getElementById('menuHandle').style.left = '0px';", 100);
isMenuExpanded = false;
isMenuFocused = false;
}
}
function manageDockStatus()
{
    var image = document.getElementById('dockImage');
    image.src = isMenuDocked ? "../img/unlock.png" : "../img/lock.png";
    var menuDiv = document.getElementById('divSlidingMenu');
    var menuHandleDiv = document.getElementById('menuHandle');
    var contentDiv = document.getElementById('innerContent');

    if(!isMenuDocked)
    {
        divSlidingMenu.style.filter = "alpha(opacity=100)"; /* IE's opacity*/
        divSlidingMenu.style.opacity = "1.00";
        menuDiv.style.position = "static";
        menuDiv.style.styleFloat = "left"; //IE
        menuDiv.style.cssFloat = "left";
        contentDiv.style.position = "static";
        contentDiv.style.styleFloat = "right"; //IE
        contentDiv.style.cssFloat = "right";
        menuHandleDiv.style.display = "inline"; //skryt to "usko"
        var mainDivWidth = document.getElementById('main').offsetWidth;
        var menuDivWidth = menuDiv.offsetWidth;
        contentDiv.style.width = (mainDivWidth - menuDivWidth) + "px";
    }
    else{
        divSlidingMenu.style.filter = "alpha(opacity=80)"; /* IE's opacity*/
        divSlidingMenu.style.opacity = "0.80";
        menuDiv.style.position = "fixed";
        menuDiv.style.styleFloat = "none"; //IE
        menuDiv.style.cssFloat = "none";
        menuDiv.style.left = '-220px'; //nechat vysunute
        isMenuExpanded = true;
        contentDiv.style.position = "relative";
        contentDiv.style.styleFloat = "none"; //IE
        contentDiv.style.cssFloat = "none";
        contentDiv.style.width = "100%";
        menuHandleDiv.style.display = "inline";
    }
    isMenuDocked = !isMenuDocked;
}
}

```

Posúvanie menu pri prechode na susedné oblasti mali na starosti funkcie *expandMenu* a *collapseMenu*. Zmena pozície sa nastavovala funkciami operujúcimi nad parametrami konkrétneho divu a to takto:





```
setTimeout("document.getElementById('divSlidingMenu').style.left = '-220px';", 100);
```

kde *divSlidingMenu* je názov divu v tele dokumentu obsahujúceho tabuľku s menu. Zamykanie a zmenu rozloženia mala za úlohu funkcia *manageDockStatus()*.

Pri zmene zdrojového kódu návrhu v hlavnom dokumente (index.html)

```
<div id="test">O3D O3D O3D O3D O3D O3D O3D O3D </div>
```

Zmena na:

=====

```
<div id="o3d"> </div>
```

sa OD3 časť zobrazovala stále na vrchu. Tento problém sa nepodarilo vyriešiť ani nastavovaním atribútov *position* v CSS kaskádových štýloch a ani v nastavovaní atribútov vo funkcii *manageDockStatus*.

Kvôli časovej náročnosti na vyriešenie tohto problému sa od návrhu upustilo a skúsili sa preto otestovať možnosti s použitím frame-ov s predpokladom, že frame je nadradený prvku div a nemôže prekračovať jeho hranice. Bohužiaľ, ani toto riešenie nezabránilo prelínaniu O3D divu nad všetky možné vrstvy a prvky.

6.3.3 Optimalizácia

Menu bolo optimalizované zmenšením tlačidiel na čitateľnú veľkosť a zmenou veľkosti textu v celom tele dokumentu. Vlastnosti textu sú v CSS štýloch vo všeobecných nastaveniach dokumentu a to

```
* {
  font-size: 11px;
  font-family: verdana;
  margin: 0;
  padding: 0;
  color: #102c57;
}
```

Ďalšou optimalizáciou bolo nastavenie pevnej šírky tabuliek pre rovnaké zobrazovanie v prehliadačoch a pevné nastavenie šírky prvkom tabuľky menu.

```
#upper_form {
  width: 185px;
  margin-left: 5px;
  margin-top: 0px;
}
```

```
#lower_form {
```





```
width: 185px;
margin-left: 5px;
}
#info{
width: 185px;
}
#updown{
width: 185px;
}
select {
width: 176px;
}
```

6.4 Automatický update databázy z XML súborov

Informácie o miestnostiach sa nachádzajú na dvoch miestach

- MySQL databáza
- XML – súbor „RoomGraph.xml“ - obsahuje informácie z databázy a nejaké informácie navyše (graf navigácie medzi miestnosťami.)

Oba spomínané zdroje majú charakter dátových zdrojov, ktoré obsahujú informácie o miestnostiach. Počas našej implementácie sme používali ako zdroj informácií XML súbor, v ktorom sú uložené všetky potrebné informácie.

6.4.1 Spustenie synchronizácie

Pre implementáciu prepojenia a automatickej aktualizácie sme použili technológiu PHP s napojením na mySQL. Súbor, ktorý obsahuje tento skript má názov „convertDataMysqlXml.php“.

Pre spustenie aktualizácie je potrebné zavolať PHP skript. Najjednoduchší spôsob je priamo sa odkázať na adresu PHP skriptu (v našom prípade otvoriť v internetovom prehliadači adresu „http://IP_Adresa_servera/PHP/convertDataMysqlXml.php“). Na danej adrese sa po vykonaní skriptu zobrazia informácie o stave jednotlivých záznamov – či bol synchronizovaný záznam upravený (UPDATED RECORD) alebo vložený (INSERTED RECORD).

6.4.2 Postup synchronizácie

Skript funguje nasledovne:

- Pre všetky záznamy z XML
 - vyhľadaj, či existuje korešpondujúci záznam v databáze (porovnaj podľa ID miestnosti)





- Ak záznam existuje – vykonaj update databázového záznamu
- Ak záznam neexistuje – vykonaj insert daného záznamu
- Vypíš vykonanú akciu pre používateľa

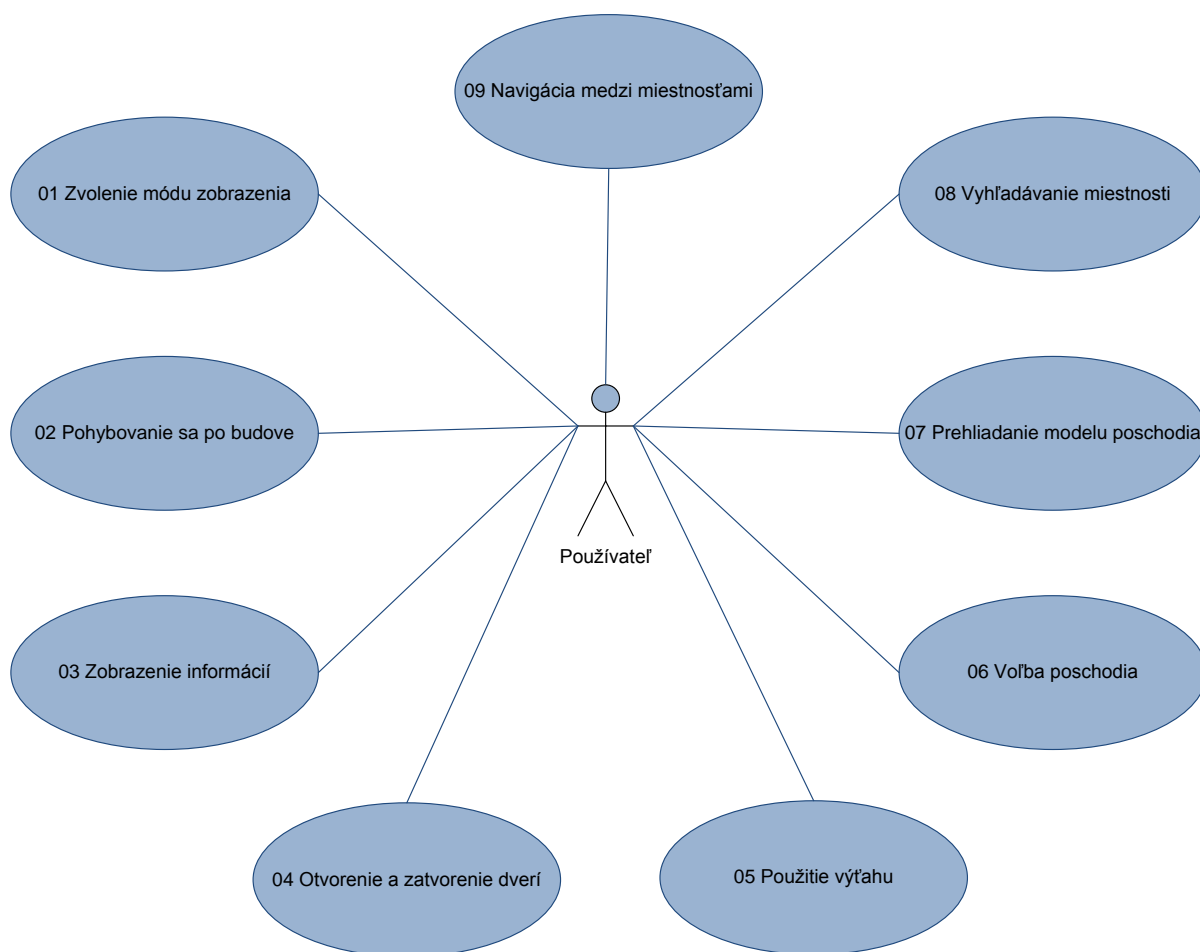
Synchronizácia prebieha iba jednosmerne. Zdrojom údajov je XML dokument a obnovujú sa iba údaje v MySQL databáze. Dôvodom je fakt, že v čase implementácie bol hlavným dátovým zdrojom XML súbor.





7 TESTOVANIE

Testovanie jednotlivých prípadov použitia pre bežného používateľa:

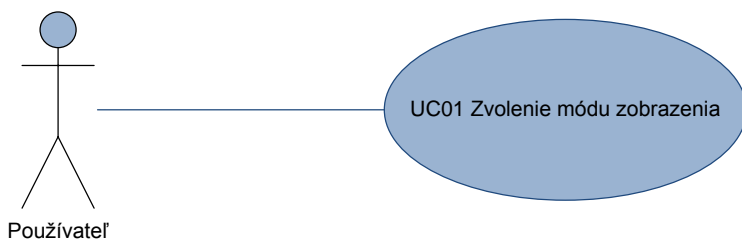


Obr. 65 Prípady použitia





7.1.1 UC01 Zvolenie módu zobrazenia



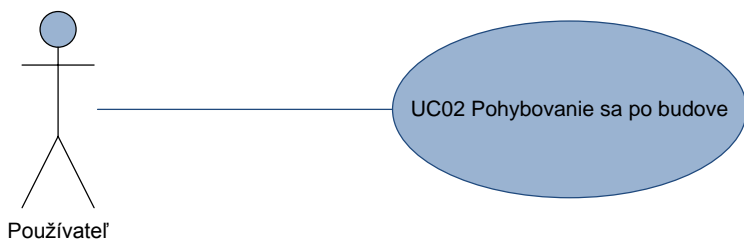
identifikátor	UC01			
názov	Zvolenie módu zobrazenia			
opis	Používateľ si môže zvoliť jeden z 3 módov zobrazenia modelu			
priorita	1 = vysoká	frekvencia	denne stovky krát	
vstupné podmienky	Nie sú			
výstupné podmienky	Používateľ má v hlavnom okne zobrazený vybraný mód modelu			
používatelia	Bežný používateľ			
Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	0	System dáva používateľovi možnosť výberu z 3 módov zobrazenia		
	1	Používateľ klikne na jeden z nich („3D“, „Prehliadanie“ alebo „Navigácia“)		





	1a	Používateľ vyberie mód 3d	Zobrazí sa 3d mód, pričom používateľ sa nachádza pri vstupe do budovy	OK
	1b	Používateľ vyberie mód prehliadanie	Zobrazí sa prehliadací mód	OK
	1c	Používateľ vyberie mód navigácia	Zobrazí sa navigačný mód	OK
Poznámka k testovaniu				
OK				

7.1.2 UC02 Pohybovanie sa po budove



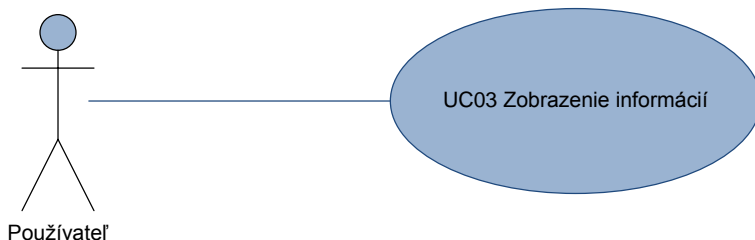
identifikátor	UC02		
názov	Pohybovanie sa po budove		
opis	Používateľ sa môže voľne pohybovať po modeli FIIT		
priorita	1 = vysoká	frekvencia	denne stovky krát
vstupné podmienky	Používateľ zmenil mód na 3D mód		
výstupné podmienky	Používateľ sa nachádza na novej pozícii alebo sa pozerá iným smerom		
používatelia	Bežný používateľ		





Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Systém nastaví pozíciu používateľa pred vchod do budovy		
	2	Používateľ pohybuje šípkami alebo klávesmi W,S,A,D	Zmena pozície v budove, súčasným podržaním klávesy SHIFT sa mení pozícia o väčšiu vzdialenosť	OK
	3	Systém zobrazuje model z novej pozície alebo pohľadu používateľa		OK
	2a	Používateľ stláča tlačidlá myši a pohybuje ňou	Zmena smeru, ktorým sa pozerá	OK
Poznámka k testovaniu				
OK				

7.1.3 UC03 Zobrazenie informácií



identifikátor	UC03
názov	Zobrazenie informácií





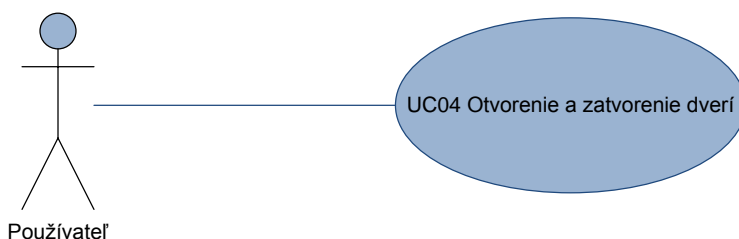
opis	Používateľovi sa v paneli zobrazia informácie o miestnosti alebo budove FIIT			
priorita	1 = vysoká	frekvencia	denne stovky krát	
vstupné podmienky	Používateľ sa nachádza v 3D móde, prehliadacom móde alebo navigačnom móde s už vyhľadaným modelom			
výstupné podmienky	Používateľovi sa zobrazia informácie o miestnosti alebo budove FIIT			
používatelia	Bežný používateľ			
Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Používateľ vykoná akciu vyvolávajúcu zobrazenie informácií o miestnosti	prechod do novej miestnosti v 3D móde alebo kliknutie na miestnosť vo zvyšných módoch	
	2	Systém zmaže predchádzajúce informácie v paneli		OK
	3	Systém zobrazí aktuálne informácie o danej miestnosti v paneli	Zobrazenie aktuálne informácie o danej miestnosti v paneli	OK
	3a	Používateľ sa nachádza v 3D móde pred budovou	Zobrazenie informácií o celej budove	
	3b	Používateľ sa nachádza v prehliadacom móde bez vybratej miestnosti	Zobrazenie informácie o zobrazenom poschodí	
	3b	Používateľ sa nachádza v prehliadacom móde bez vybratej miestnosti	Zobrazenie informácie o zobrazenom poschodí	





Poznámky k testovaniu
OK

7.1.4 UC04 Otvorenie a zatvorenie dverí



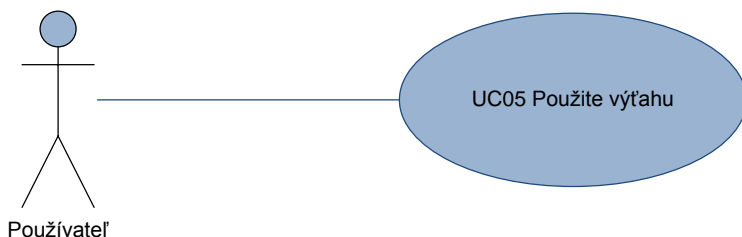
identifikátor	UC04			
názov	Otvorenie a zatvorenie dverí			
opis	Interakcia používateľa v 3D móde s dverami			
priorita	1 = vysoká	frekvencia	denne stovky krát	
vstupné podmienky	Používateľ sa nachádza v 3D móde			
výstupné podmienky	Používateľovi môže/nemôže prejsť dverami			
používatelia	Bežný používateľ			
Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Používateľ sa priblíži k dverám na dostatočnú vzdialenosť s priamou viditeľnosťou na dvere		





	2	Používateľ klikne na dvere		
	3	Dvere boli zatvorené	Zobrazenie animácie otvárajúcich sa dverí	
	4	Dvere boli otvorené	užívateľ nimi môže prejsť	
	3a	Dvere boli otvorené	Zobrazenie animácie zatvárajúcich sa dverí	
Poznámka k testovaniu				
OK				

7.1.5 UC05 Použitie výťahu



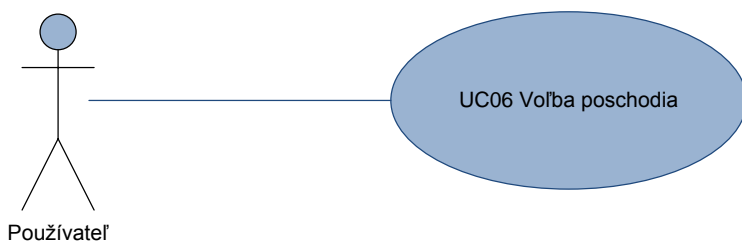
identifikátor	UC05		
názov	Použitie výťahu		
opis	Prechod medzi poschodiami pomocou výťahu		
priorita	3 = nízka	frekvencia	denne desiatky až stovky krát
vstupné podmienky	Zvolený je 3D mód		
výstupné podmienky	Používateľ sa premiestnil na iné poschodie		
používatelia	Bežný používateľ		





Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Používateľ sa priblíži k dverám výťahu na dostatočnú vzdialenosť s priamou viditeľnosťou na dvere		
	2	Používateľ klikne na dvere výťahu	Spustenie animácie otvárajúcich sa dverí	
	3	Otvorenie dverí	Užívateľ vstúpi do kabíny.	
	4	Kliknutím na číslo poschodia na paneli zvolí cieľ		
	5	Dvere výťahu sa zatvoria na dobu dvoch sekúnd	Dvere výťahu sa otvoria	
	6	Užívateľ vystúpi z výťahu na inom poschodí, ako vstupoval		
Poznámka k testovaniu				
OK				

7.1.6 UC06 Voľba poschodia



identifikátor	UC06
----------------------	------





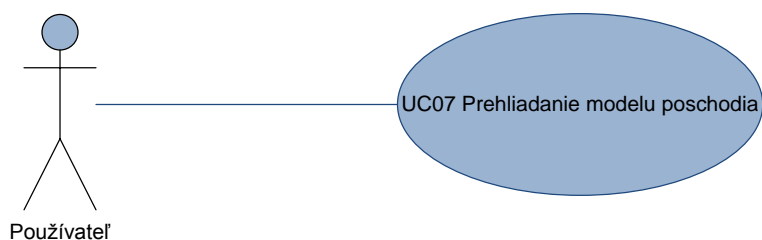
názov	Voľba poschodia			
opis	Používateľ si môže zmeniť zobrazené poschodie			
priorita	1 = vysoká	Frekvencia	denne stovky krát	
vstupné podmienky	Požívateľ zmenil mód na prehliadací mód			
výstupné podmienky	Používateľ má v hlavnom okne zobrazené vybrané poschodie			
používatelia	Bežný používateľ			
Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Systém dáva používateľovi možnosť zmeniť zobrazené poschodie		
	2	Používateľ klikne na zobrazenie poschodia nad alebo pod ktorým sa práve nachádza		
	3	Systém nastaví pohľad, pri ktorom je vidieť celý model vybraného poschodia pod miernym uhlom		
	4	Načítanie dlhšie ako 1s	systém pred zobrazením poschodia vypíše „Loading...“	





	3a	Používateľ sa nachádza na najnižšom poschodí a zvolí si zobrazíť poschodie pod alebo sa nachádza na najvyššom poschodí a zvolí si zobrazíť poschodie nad	system na to používateľa upozorní a nezmení poschodie	
Poznámka k testovaniu				
OK				

7.1.7 UC07 Prehliadanie modelu poschodia



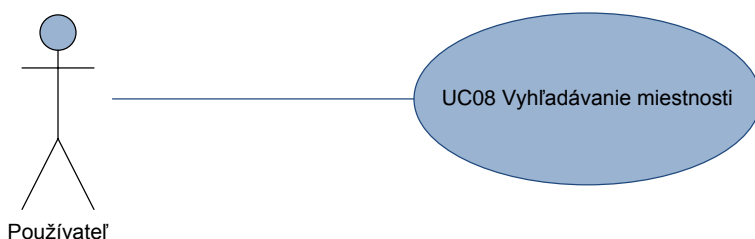
identifikátor	UC07		
názov	Prehliadanie modelu poschodia		
opis	Používateľ si môže prezerať model z rôznych pohľadov		
priorita	1 = vysoká	Frekvencia	denne stovky krát
vstupné podmienky	Používateľ zmenil mód na prehliadací alebo navigačný mód		
výstupné podmienky	Používateľ sa pozerá na model poschodia z iného pohľadu		
používatelia	Bežný používateľ		





Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Systém nastaví úvodný pohľad, pri ktorom je vidieť celý model poschodia pod miernym uhlom		
	2	Používateľ stlačením tlačidla myši a jej pohybom mení uhol, pod ktorým sa na model poschodia pozerá	Systém zobrazuje model z nového pohľadu	
	2a	Používateľ kolieskom na myši mení vzdialenosť od modelu		
Poznámka k testovaniu				
OK				

7.1.8 UC08 Vyhľadávanie miestnosti



identifikátor	UC08
názov	Vyhľadávanie miestnosti
opis	Používateľ pomocou filtra vyhľadá miestnosť a nechá si o nej zobrazit' informácie





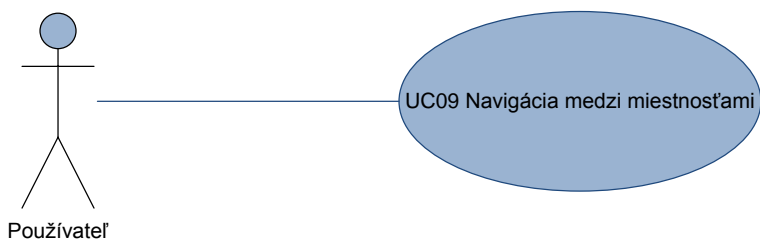
priorita	1 = vysoká	Frekvencia	denne niekoľko desiatok až stoviek krát	
vstupné podmienky	Aktívny je navigačný mód			
výstupné podmienky	Používateľovi sa zobrazí zoznam miestností, ktoré vyhovujú kritériám, prípadne informácia, že žiaden vyhovujúci záznam sa nenašiel.			
používatelia	Bežný používateľ			
Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Užívateľ zadá do textového poľa v pravom paneli kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo		
	2	Užívateľ stlačí tlačidlo „Hľadať“	Systém zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam	
	3	Užívateľ kliknutím vyberie niektorú miestnosť zo zoznamu	Systém v hlavnom okne zvýrazní v modeli budovy zodpovedajúcu miestnosť a v pravom paneli zobrazí informácie o miestnosti. Mód sa automaticky mení na prehliadací.	





	3a	Vyhľadávanie bolo neúspešné	Zobrazenie hlásenia „Nebola nájdená žiadna miestnosť pre <input text>“, kde <input text> je text vložený užívateľom	
Poznámka k testovaniu				
OK				

7.1.9 UC09 Navigácia medzi miestnosťami



identifikátor	UC09		
názov	Navigácia medzi miestnosťami		
opis	Používateľ pomocou filtra vyhledá dve miestnosti a dostáva znázornenie najkratšej cesty medzi nimi		
priorita	1 = vysoká	Frekvencia	denne niekoľko desiatok až stoviek krát
vstupné podmienky	Aktívny je navigačný mód		
výstupné podmienky	Aplikácia používateľovi zobrazí trasu medzi dvomi bodmi virtuálneho interiéru.		
používatelia	Bežný používateľ		





Testovanie				
postupnosť	krok	Akcia	Očakávaná reakcia	Skutočná reakcia
	1	Užívateľ zadá do textového poľa v pravom paneli kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo		
	2	Užívateľ stlačí tlačidlo „Hľadať“	Systém zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam	
	3	Užívateľ kliknutím na tlačidlo „Navigácia“ povolí sekundárny vyhľadávací formulár.		
	4	Užívateľ zadá do textového poľa sekundárneho formulára kritérium vyhľadávania – buď označenie miestnosti, meno osoby alebo kľúčové slovo		
	5	Užívateľ stlačí tlačidlo „Hľadať“ sekundárneho formulára	V sekundárnom formulári sa zobrazí zoznam miestností s položkami, ktoré vyhovujú zadaným podmienkam	





	6	Užívateľ kliknutím vyberie jednu miestnosť v zoznamoch oboch formulárov	Povolí sa tlačidlo „Nájsť cestu“	
	7	Užívateľ stlačí „Nájsť cestu“	V hlavnom okne sa v modeli interiéru vykreslí čiara spájajúca dve vybrané miesta	
Poznámka k testovaniu				
OK				





8 TESTOVANIE KONEČNÝMI POUŽÍVATEĽMI

Testovanie aplikácie prebiehalo vo viacerých fázach.

V prvej fáze prebiehalo testovanie samotnými vývojármi a bolo zamerané na odchytenie chybných funkcií. Všetky nájdené chyby v implementácii boli odstránené v krátkej dobe po ich identifikácii.

V druhej fáze prebiehalo testovanie osobami, ktorí neboli súčasťou vývojového tímu a nepoznali technologické pozadie aplikácie. Pre zjednodušenie odozvy od používateľov sme vytvorili jednoduchý dotazník, ktorý každý respondent vyplnil.

Dotazník obsahoval nasledovné oblasti:

Spôľahlivosť a správna funkčnosť aplikácie (chyby, padanie a pod.)

Plynulosť zobrazovania modelov (teda subjektívne hodnotenie FPS)

Jednoduchosť ovládania a rozloženie grafických elementov

Vizuálna stránka aplikácie

Každú z týchto oblastí mohol používateľ ohodnotiť bodmi v rozsahu 0-5, pričom 5 je najlepšie ohodnotenie.

Od používateľov sme chceli vedieť základné údaje a to pohlavie, vek a náklonnosť ku IT (konkrétna otázka mala tvar „Si IT thinking :) ?“)

Testovanie prebehlo na vzorke 50 respondentov s rôznymi charakteristikami.

Na začiatku dokumentu vidíme rozdelenie respondentov. Nasleduje analýza odpovedí na jednotlivé otázky, pri ktorých sú uvedené aj niektoré slovné poznámky používateľov. V závere nájdeme zhodnotenie testovania v prehľadnej tabuľke aj s dodatočnými štatistickými charakteristikami.

Výsledky testovania pokladám za relevantné, vzhľadom na rôznorodosť a množstvo používateľov, ktorí aplikáciu testovali. V poslednom grafe vidíme, že výsledky sa pohybujú okolo hodnoty 3,5 až 4 body z 5, čo pokladám za pozitívne.

Len výnimočne mali používatelia vážne pripomienky k projektu, zväčša išlo o chyby malé a týkajúce sa výzoru. Preto predpokladám, že ak na projekte bude pracovať ďalší tím ešte rok, bude projekt plne nasaditeľný a bude prínosom v reálnom svete.

Vzhľadom na fakt, že žiadny z členov tímu nemal skúsenosti s podobnými projektmi, tak zo stránky charakteru jeho tímu (školský 6-členný tím) ako aj technologického charakteru projektu (technológia O3D), považujem projekt za úspešný.

\

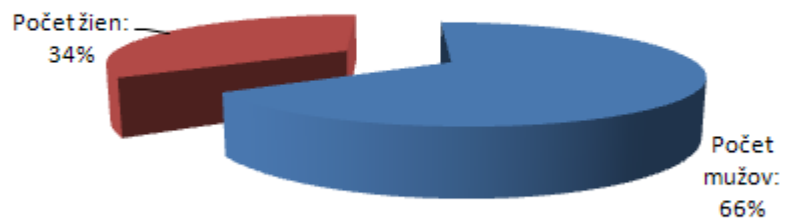




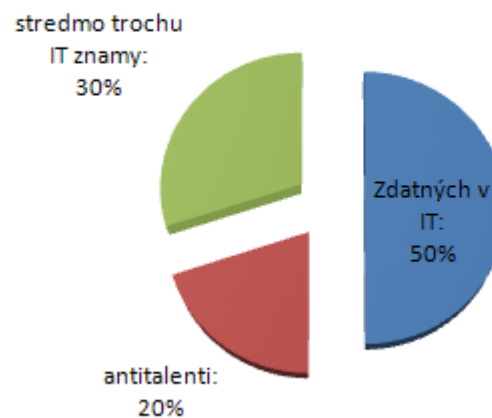
8.1 Rozdelenie respondentov:

Počet mužov:	33
Počet žien:	17
Zdatných v IT:	25
antitalenti:	10
stredmo trochu IT znamy:	15

Pohlavie respondentov



Znalosti v IT





8.2 Plynulosť zobrazovania modelov

Plynulosť zobrazovania modelov (teda subjektívne hodnotenie FPS - Ak viete prosím uveďte stručnú konfiguráciu PC)	0	5	2	4	5	4	3	3	3	5	4	5	4	5	3	4	5	0	5	2	4	5	4	3	3
	5	5	5	3	4	0	5	2	4	5	4	3	3	3	5	0	5	4	3	4	4	3	5	5	5

problémy:

1. procesor 1,7 Ghz, RAM 1,5Gb, hardisk 60GB, win XP
2. AMD Athlon 64 X2 Dual Core 4200+; 2,2 GHz; 2 GB RAM
3. nefungovalo mi celkom vyhľadavanie a zobrazenie, možno chybou notebooku
4. 3GHz procak; 1024 RAM; 256 grafika
5. Pomalé prvé načítavanie, a načítavanie pri prechode do miestnosti. PC : Intel Core 2 duo T750, NvidaGT 8xxx, 256M
6. C2D 3 GHz, Nvidia 8800 GTS, 4 GB RAM
7. Po načítaní išlo všetko svižne a plynulo. Po kliknutí na 3D
8. sa mi to celé asi na minútu zaseklo, ale potom to išlo zase plynulo. Procesor 1,81 GHz, 2GB RAM



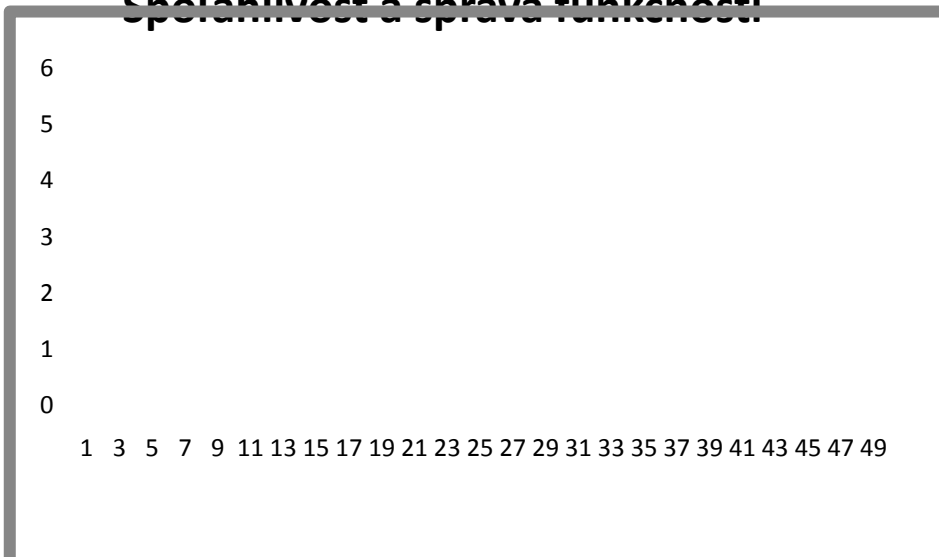
8.3 Spôľahlivosť a správna funkčnosť aplikácie

Spôľahlivosť a správna funkčnosť aplikácie (chyby, padanie aplikácie)	0	5	2	2	4	2	5	4	3	3	4	5	4	5	5	3	4	1	3	3	4	5	4	5	5
	1	4	2	4	5	3	4	5	4	5	5	2	0	3	4	1	2	3	4	5	4	5	5	2	2

Problémy:

1. nejde mi to otvoriť, chyba mi niečo nainštalované
2. po chvíľke hrania mi to ani raz nepadlo a okrem TO DO na nultom poschodí sa to tvárilo pekne :)
3. nefungovalo mi celkom vyhľadávanie a zobrazenie, možno chybou notebooku
4. keď som klikol na "3D" tak to nejako seklo..
5. Nenašiel som názvy miestnosti podľa ktorých sa dá hľadať. Teda neotestovaná funkcia nájdenia cesty
6. Vyhľadávanie funguje pochybne, keď si zadám presný názov miestnosti, zobrazí sa iba TODO na 0. poschodí, keď dám nezmysel, zobrazí sa TODO na NaN
7. musel som kvôli nej inštalovať O3D plugin, tak len 4 body ... Aj keď sa to asi inak nedá
8. Okrem toho prekryvanie, chôdze cez s tým som nenatrafil na nič.
9. ok

Spôľahlivosť a správa funkčnosti





8.4 Jednoduchosť ovládania a rozloženie grafických elementov

Jednoduchosť ovládania a rozloženie grafických elementov	0	5	4	3	3	3	3	4	4	4	4	4	4	4	5	5	4	3	3	5	4	5	4	5	3
	4	5	0	5	2	4	5	4	3	3	3	5	5	3	5	1	0	2	3	1	2	0	1	1	5

Problémy:

1. hmmm neviem
2. pekne, možno by to chcelo ešte trosku viac prepracovať :->
3. Až na chýbajúce popisy a názvy miestnosti je ovládanie intuitívne
4. Je nezvyk, že na otáčanie sa v 3D pohľade je potrebné držať stlačené tlačidlo myši, WSAD je relatívne k smeru pohľadu, čo pôsobí dezorientujúco, kliknutie na klávesnici
5. lepšie by sa to ovládalo, keby bolo vymenene ľavé a pravé tlačidlo na myši
6. ovládanie mohlo byť aj jednoduchšie
7. chvíľu mi trvalo, kým som v 3D prišiel na to, že treba stlačiť tlačidlo

Jednoduchosť ovládania a rozloženie grafických elementov





8.5 VIZUÁLNA STRÁNKA APLIKÁCIE

Vizuálna stránka aplikácie	4	5	5	4	3	4	4	4	5	4	4	3	5	4	4	4	4	3	4	5	2	2	1	5	3
	2	5	1	0	4	2	1	3	4	1	3	5	1	2	3	5	4	1	2	0	3	1	2	5	4

Názory:

1. Chýbajú textúry :)
2. OK, fajn, ujde to.....
3. Celkové hodnotenie - veľmi milé :)
4. Pekne spracovane textúry a priehľadnosť
5. Pekné, len možno by som spravil niečo také, že pri pohľade na určite poschodie by bol veľmi jemne naznačený aj exteriér celej budovy. Ale neviem, či by to nepôsobilo rušivo.
6. pekne
7. môže byť
8. v paneli by nemuseli byť tie čiarkované oddeľovače

Vizuálna stránka aplikácie

6

5

4

3

2

1

0

1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49



8.6 Súhrn:

	Priemer	Modus	Medián	Rozptyl	Smerodajná odchýlka
Spoľahlivosť a správna funkčnosť aplikácie (chyby, padanie aplikácie)	3,48	5	4	1,432	1,418
Vizuálna stránka aplikácie	3,18	4	4	1,480	1,465
Jednoduchosť ovládania a rozloženie grafických elementov	3,34	4	4	1,533	1,518
Plynulosť zobrazovania modelov (teda subjektívne hodnotenie FPS - Ak viete prosím uveďte stručnú konfiguráciu PC)	3,82	5	4	1,282	1,269

Priemer predstavuje aritmetický priemer

Modus je najčastejšie sa vyskytujúca hodnota sledovaného znaku X medzi hodnotami x_1, x_2, \dots, x_n

Medián je prostredná hodnota sledovaného znaku X medzi hodnotami x_1, x_2, \dots, x_n

Rozptyl sa definuje ako priemer druhých mocnín odchýlok od aritmetického priemeru.

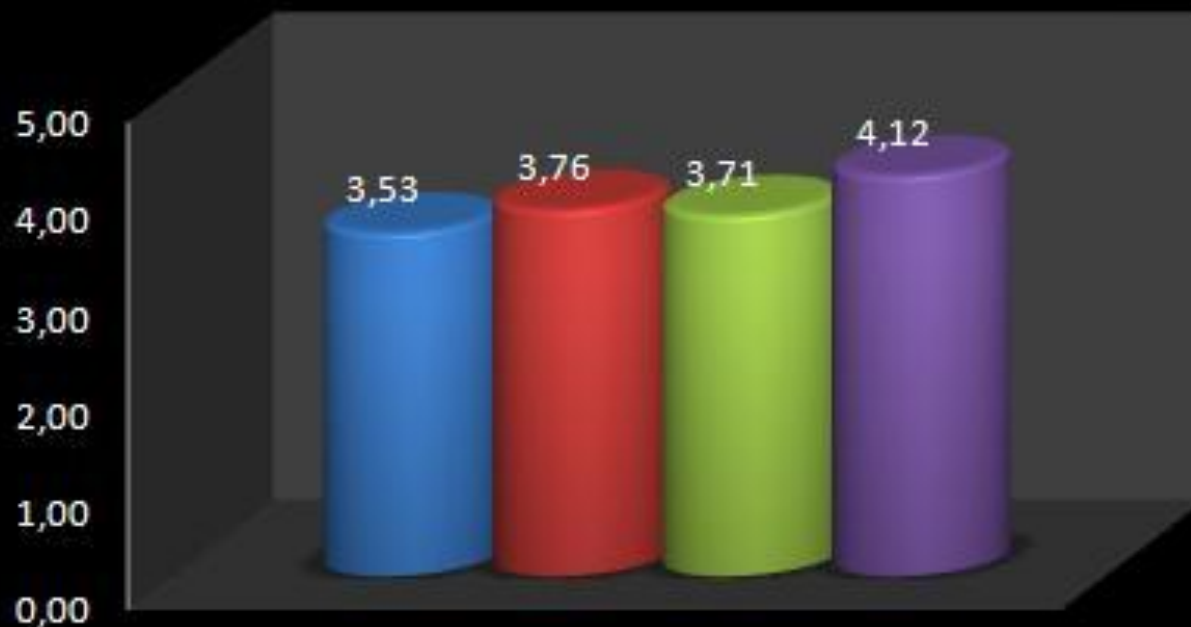
Smerodajná odchýlka je druhá odmocnina z rozptylu





Celkové hodnotenie

- Spoľahlivosť a správna funkčnosť aplikácie
- Plynulosť zobrazovania modelov
- Jednoduchosť ovládania a rozloženie grafických elementov





9 VYLEPŠENIA A ĎALŠIE SMEROVANIE ĎALŠEJ IMPLEMENTÁCIE

Táto kapitola sa pokúsi vykresliť možnosti ďalšej práce na projekte Virtuálna FIIT.

Projekt bol jednoročným projektom, realizovaný tímom 6 ľudí s prevažne malými skúsenosťami s prácou v tíme. Okrem toho bol projekt iba jednou z viacerých aktivít jeho účastníkov, preto vzhľadom na obmedzený čas sme neimplementovali všetky naše nápady. Táto časť dokumentu sa pokúsi opísať aspoň časť neimplementovaných nápadov a možných vylepšení aplikácie. Má slúžiť ako pomôcka pre nasledujúci potenciálny vývoj, napríklad ďalšieho tímu na predmete tímový projekt.

Možnosti vylepšení sú rôzne. Niektoré sú realizovateľné ľahšie, niektoré ťažšie a niektoré sú dokonca také, že za momentálnych podmienok nie sú realizovateľné vôbec (ako príklad uvediem automatické prepojenie na informačný systém AIS, keďže v čase písania dokumentu je AIS komerčným plateným systémom so zatvorenými zdrojovými kódmi a bez možnosti automatického napojenia).

9.1 Vernosť pôdorysu budovy

V čase implementácie projektu bola nová budova FIIT stále vo výstavbe. Plány, podľa ktorých sme modelovali budovu sa počas projektu menili a teda implementácia modelu bola iba približná. Podľa môjho osobného názoru nie je potrebné implementovať model v detailnejšej podobe. Momentálny stav mi pripadá dostatočný.

Iným problémom sú zmeny, ktoré môžu vzniknúť počas stavby. Ani v čase finalizácie projektu nebola budova FIIT postavená (bola vo výstavbe). Preto je možné, že sa zmenili niektoré miestnosti úplne, niektoré sa možno spojili alebo rozdelili. Pred znovu naštartovaním projektu by som pokladal za rozumné overiť správnosť implementácie modelu.

9.2 Vernosť 3d prostredia

3D prostredie INTERIERU je implementované v min. forme. Sústredili sme sa na tvorbu modelu budovy ako takého, a ostatným grafickým prvkom sme sa venovali len minimálne. Sústredili sme sa na implementáciu modelu budovy ako takého a nevenovali sme sa ostatným grafickým vylepšeniam. 3d priestor sa preto nemusí zdať pre používateľa príťažlivý. Pri ďalšom vývoji je možné poskytnúť prostredie, ktoré nebude radového používateľa odpudzovať ale naopak ho priláka.

Vylepšenia môžu zahŕňať zaoblenie stien, textúry na stenách, vybavenie miestností vo forme nábytku, počítačov, interaktívnych tabúl a podobne.

Takto by aplikácia mohla slúžiť aj pre ľudí, ktorí budovu nikdy nenavštívili a chceli by si ju pozrieť. Pre budúcich študentov a podobne.





9.3 Množstvo informácií a dáta

Momentálne aplikácia obsahuje iba minimum faktových informácií okrem virtuálneho modelu budovy. Tieto informácie je možné doplniť, aby mala aplikácia väčšiu silu a mohla perspektívne slúžiť ako komplexný informačný panel.

- Informácie o miestnostiach – Momentálne evidujeme iba názov miestnosti, jej jednoduchý popis a jej rozvrh vo forme HTML tabuľky. Je možné doplniť informácie o kapacite, charaktere, zariadení a podobne. Dáta v databáze majú charakter príkladov, nie reálnych dát.
- Informácie o ľuďoch – Momentálne aplikácia neobsahuje žiadne informácie o ľuďoch. Komplexné údaje o osobách by nám poskytli priestor na ich vyhľadávanie a následné vyhľadávanie cesty k nim. V spojení s informáciami o rozvrhových akciách by bolo možné osoby vyhľadávať dynamicky, teda nie nájsť iba domácu miestnosť danej osoby, ale nájsť miesto kde sa v danom čase osoba naozaj nachádza.
- Informácie o jedálňach - Momentálne aplikácia neobsahuje žiadne informácie o jedálňach. Môže obsahovať dynamické jedálenské menu, zobrazované napríklad formou interaktívnych tabúľ priamo v 3d prostredí.

9.4 Vyhľadávanie

Aplikácia obsahuje jednoduché vyhľadávanie cesty medzi dvoma ručne definovanými miestnosťami. Sústredili sme sa na funkčnú stránku vecí, preto v tejto časti je taktiež priestor na vylepšenia z pohľadu „prívetivosti k používateľovi“.

Za predpokladu, že doplníme potrebné údaje do databázy, je možné implementovať vyhľadávanie v aplikácii podľa veľa rôznych kritérií:

- Kontakty – v spojení s informáciami o rozvrhoch vyhľadávanie momentálnej polohy, vyhľadávanie koreňovej miestnosti - kancelárie
- Miestnosti – podľa vybavenia, typu miestnosti (kabinet, počítačová učebňa a podobne) alebo voľné miestnosti v daný čas. Pri dostatku informácií by sme mohli vytvoriť aj komplexný dopyt ako „Kde v škole môžem využiť školský počítač v daný čas (ktoré sú voľne dostupné počítačové učebne v daný čas)?“ Takáto komplexnosť si okrem iného vyžaduje aj možnosť vyhľadania viacerých miestností naraz.
- Rozvrhy – Vyhľadávanie miestností pre jednotlivé rozvrhové akcie (tu je priestor aj pre integráciu s rozvrhovým systémom AIS).

9.5 Integrácia

Keďže ide o informačný systém, vždy je potrebné získavať čo najnovšie informácie, podľa možnosti automaticky, bez potreby správy administrátorom.





Integrovať by bolo napríklad možné:

- AIS – integrácia informácií o ľuďoch, rozvrhoch, miestnostiach a podobne. Všetky tieto informácie AIS obsahuje, no je to komerčný uzavretý systém, ktorý je implementovaný v Českej republike. Momentálne sú možnosti získania dát veľmi malé a čo sa týka automatickej aktualizácie databázy, tá je nemožná. Postupne sa objavujú možnosti integrácie pomocou webových služieb. Odporúčam nakontaktovať sa na administrátorov systému a informovať sa o súčasnom stave.
- Dochádzkový systém – Ak by sa podarilo implementovať prepojenie na dochádzkový systém školy, bolo by možné zistiť a následne používateľa informovať o prítomnosti jednotlivých osôb v budove školy. Avšak vzhľadom na dôvernosť týchto informácií predpokladám, že integrácia bude ťažko realizovateľná.
- Automatická poloha – Ak by systém vedel zistiť polohu aplikácie, teda polohu používateľa, bolo by možné automaticky prednastaviť začiatkový bod vyhľadávania cesty.

9.6 Pomocník - HELP

Pre zvýšenie „prívetivosti k používateľovi“ by bolo potrebné implementovať komplexný systém pomocníka – HELP.

Ako jedna z možností je implementácia prepínača v rohu obrazovky (napríklad s obrázkom otáznika), na ktorom by bolo rozoznateľné, či je prepínač zapnutý alebo vypnutý. Ak by používateľ prepínač zapol, aplikácia by sa prepla do režimu help a používateľ by kliknutím na komponent aplikácie vyvolal text s help správou.

Tvorba komplexnej používateľskej príručky je síce potrebná, no nemôže nikdy nahradiť rýchlu nápovedu v aplikácií. Podľa môjho osobného názoru bude používateľskú príručku čítať maximálne 10% používateľov, preto je potrebné poskytnúť nápovedu v čo najjednoduchšej forme

9.7 Nenaplnené požiadavky, resp. Čo sme nestihli

Pri charakteristike nesplnených častí, ktoré sme si na začiatku naplánovali, vychádzame z bodov určenia priorít implementácie (kap. 4.5), ktoré vlastne obsahujú harmonogram realizácie projektu. Ako prvý fakt môžeme skonštatovať, že sa nám podarilo splniť takmer všetkých 11 bodov. Z toho 3 body sme splnili čiastočne. Konkrétne sa jedná o namodelovanie všetkých poschodí budovy, kedy sme nestihli namodelovať najspodnejšie poschodie. Toto poschodie obsahuje prevažne skladové miestnosti. Taktiež 3. a 4. poschodie nie sú namodelované, ale iba skopírované 5., pričom označenia miestností sú korektné. Rozhodli sme sa pre toto riešenie z nedostatku času a tiež z dôvodu, že dané poschodia sú si veľmi podobné. Avšak tie najkomplikovanejšie a najväčšie časti budovy sme vymodelovali a celkovo je vymodelovaných približne 90% priestorov. Aktuálny stav poschodí je podrobne popísaný v Tab. 6.1. Ďalej sme nestihli spomedzi interaktívnych prvkov implementovať výťah. Za nie úplne splnený bod je možné považovať zaznamenanie aktuálnej pozície používateľa





v 3D móde. Aktuálne sa používateľovi zobrazuje iba informácia o poschodí, na ktorom sa nachádza, namiesto presnejšej informácie o miestnosti. Za nadmieru splnený bod možno považovať pohyb v 3D móde, kde sme implementovali výškové mapy zamedzujúce prechodu cez steny. Aj tento bod má svoje nedostatky a potrebuje ošetriť prechádzanie cez zavreté dvere a tiež prechod medzi poschodiami, v ktorom sa niekedy vyskytne chyba. Úspešne splnené bolo aj prepojenie web aplikácie s databázou, ktoré ale nenapĺňuje úplne svoj význam, pretože databáza neobsahuje skoro žiadne dáta. Celkovo je nesplnenej časti minimálne k tej splnenej, o čom svedčí aj pekný a použiteľný výsledok.





10 ZÁVER

Pri práci na tímovom projekte sme vychádzali z požiadaviek na funkcionálnosť od zadávateľky projektu. Požiadavky sme dôkladne preskúmali a na ich základe sme vytvorili návrh, v ktorom sme sa na ne snažili pozrieť z pohľadu zrealizovateľnosti zadaného projektu. V týchto skorých fázach projektu tvorených najmä analýzou sme sa zhodli na tom, že najťažšie splniteľnou požiadavkou bolo, aby projekt fungoval v 3D cez webový prehliadač. V tom čase existovalo len niekoľko možných spôsobov riešenia, z ktorých každý mal svoje plusy, ale aj svoje mínusy. Ako sme postupne skúmali jednotlivé možnosti riešenia, museli sme väčšinu hneď zamietnuť – X3D a Flash boli príliš pomalé, WebGL pre neexistenciu stabilnej finálnej verzie nepoužiteľné a realizácia projektu prostredníctvom vlastného pluginu by bola príliš komplikovaná a neflexibilná. Projekt sme sa nakoniec rozhodli realizovať posledným možným spôsobom riešenia – O3D.

O3D v súčasnosti považujeme za rozumnú voľbu, ale napriek tomu vývoj v tomto API nebol vôbec jednoduchý. Na začiatku nás veľmi obmedzovala neexistencia žiadneho podporného vývojového prostredia, neskôr sa k tomu pridali problémy so zobrazovaním nami vytvorených modelov alebo pri implementácii požadovanej funkcionality (otváranie dverí, pohyb po budove, otáčanie modelov poschodí, informácie o miestnostiach atď.). Riešenia na tieto problémy bývali často jednoduché, ale kvôli nízkej rozšírenosti O3D, bolo veľmi obtiažne a niekedy až nemožné nájsť na tieto problémy odpovede.

Za ďalšiu výzvu sme považovali tvorbu 3D modelov, keďže sme s modelovaním predtým nemali žiadne skúsenosti. Problémy nastali už pri voľbe modelovacieho nástroja, pretože aj napriek ohlasovanej podpore formátu COLLADA niektorých nástrojov, export do tohto formátu neprebehol vždy tak, ako by mal (chýbajúce časti modelu, zlé názvy komponentov atď.). Model budovy sme nakoniec vytvorili v nástroji 3Ds Max s osobitným pluginom pre export do formátu COLLADA s názvom OpenCollada, keďže táto kombinácia sa ukázala ako jediné plne funkčné riešenie. Samotné modelovanie budovy bolo pre nás časovo veľmi náročné vzhľadom na naše skúsenosti s tvorbou zložitých modelov. Model sme navyše museli niekoľkokrát upravovať kvôli zmenám vo výkresoch rozostavanej budovy.

Pri realizácii projektu sme sa snažili dodržiavať architektúru a nami vytvorené metodiky súvisiace s projektom. Avšak nie vždy to bolo možné a s pribúdajúcimi skúsenosťami sa niektoré metodiky dočkali zmien v záujme jednoduchšieho a rýchlejšieho pokračovania v realizácii a implementácii. Medzi takéto podnety k zmenám patrili napríklad spôsob modelovania budovy alebo nové požiadavky na funkcionálnosť. Požiadavky sa postupom času mierne upravovali a konkretizovali, aby zodpovedali aktuálnemu stavu projektu. Ako tím sme požiadavky rešpektovali a snažili sa ich realizovať k spokojnosti zadávateľky projektu.

Výsledkom nášho dvojsemestrálneho úsilia je funkčná a použiteľná webová aplikácia, ktorá umožní ľubovoľnému záujemcovi jednoducho sa zorientovať v priestoroch novej budovy FIIT a v budúcnosti aj získať informácie o jednotlivých jej miestnostiach. Do jej tvorby boli zapojení všetci členovia tímu, kde mal každý svoju úlohu a zodpovednosť nad jej časťou. Každý člen tímu si svoje úlohy svedomite





plnil, čomu nasvedčuje aj stav vytvoreného projektu a pozitívne ohlasy ne neho. Ako tím sme radi, že nami vytvorený projekt má zmysel a praktické použitie, ktoré môže niekomu aj v skutočnosti ušetriť čas, ktorý by strávil blúdením po budove novej FIIT.





11 ZOZNAM POUŽITEJ LITERATÚRY

- [1] O3D API. Dostupné na internete: <http://code.google.com/intl/sk/apis/o3d/> (14.10.2009)
- [2] COLLADA. Dostupné na internete: <https://collada.org/mediawiki> (14.10.2009)
- [3] The O3D API Blog. Dostupné na internete: <http://o3d.blogspot.com/> (14.10.2009)
- [4] Web 3D Consortium. Dostupné na internete:
http://www.web3d.org/x3d/vrml/tools/viewers_and_browsers/ (3.11.2009)
- [5] X3D-Edit. Dostupné na internete: <https://savage.nps.edu/X3D-Edit/> (28.10.2009)
- [6] Xj3D. Dostupné na internete: <http://www.xj3d.org/> (28.10.2009)
- [7] WireFusion. Dostupné na internete: <http://www.demicron.com/wirefusion/> (28.10.2009)
- [8] Návrh prostredia pre simulácie evolučných, 3D multiagentových systémov, Diplomová práca, Bc. Gabriel Braniša, máj 2009
- [9] X3D a jeho možnosti pri tvorbe trojrozmerných interaktívnych scén, Bakalárska práca, Ondrej Ivančík, máj 2009
- [10] The Forbidden City: Beyond Space & Time. Dostupné na internete:
<http://www.beyondspaceandtime.org/FCBSTWeb/web/index.html> (3.11.2009)
- [11] Virtualtravel.sk. Dostupné na internete: <http://www.virtualtravel.sk/> (3.11.2009)
- [12] Beach Scene. Dostupné na internete:
<http://o3d.googlecode.com/svn/trunk/samples/beachdemo/beachdemo.html> (3.11.2009)

