



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

---

# Virtuálna FIIT

Projektová dokumentácia





## OBSAH

OBSAH .....	ii
zoznam tabuliek.....	vi
zoznam obrázkov .....	vii
ZADANIE .....	ix
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 .....	22
2.6	Analýza vývojového prostredia Virtuálnej FIIT .....	24
2.6.1	Editory vývojového prostredia .....	24
2.6.2	3D modelovanie.....	25
2.6.3	Debugovanie Javascript kódu .....	27
2.6.4	Existujúce riešenia zobrazenia 3D scén.....	27
2.6.5	Prehliadače pre systém Virtuálna FIIT.....	30
2.7	Analýza modelovacieho nástroja 3DS Max.....	31
2.8	Zhrnutie analýzy problematiky systému .....	34
3	Špecifikácia riešenia .....	36
3.1	Špecifikácia požiadaviek na systém Virtuálnej FIIT .....	36
3.1.1	Základné požiadavky na systém a jeho vývoj .....	36
3.1.2	Hardvérové a softvérové požiadavky na systém .....	37
3.1.3	Charakteristika používateľov systému .....	37
3.1.4	Požiadavky na funkcionality systému .....	37
3.2	Diagram prípadov použitia .....	38
3.2.1	UC01 Zvolenie módu zobrazenia .....	39
3.2.2	UC02 Pohybovanie sa po budove .....	40
3.2.3	UC03 Zobrazenie informácií .....	42
3.2.4	UC04 Otvorenie a zatvorenie dverí.....	43
3.2.5	UC05 Použitie výtahu .....	44
3.2.6	UC06 Voľba poschodia .....	46
3.2.7	UC07 Prehliadanie modelu poschodia .....	47
3.2.8	UC08 Vyhľadávanie miestnosti.....	48
3.2.9	UC09 Navigácia medzi miestnosťami.....	50





3.3	Nefunkcionálne požiadavky na systém .....	52
4	Návrh systému virtuálnej fiit.....	53
4.1	Architektúra systému Virtuálnej FIIT.....	53
4.1.1	Komponenty prezentačnej vrstvy .....	53
4.1.2	Komponenty aplikačnej vrstvy.....	54
4.1.3	Komponenty dátovej vrstvy .....	56
4.2	Návrh GUI systému .....	58
4.2.1	Uvítacia obrazovka.....	58
4.2.2	Obrazovka pre 3D mód .....	59
4.2.3	Obrazovka vyhľadávania miestností .....	59
4.2.4	Obrazovka vyhľadávania cesty .....	60
4.2.5	Obrazovka prehľadávania.....	61
4.3	Databázový návrh .....	63
4.3.1	Logický dátový model.....	63
4.3.2	Fyzický model .....	64
4.4	Určenie priorít implementácie systému.....	66
5	Implementácia systému virtuálnej fiit.....	68
5.1	Implementácia používateľského rozhrania .....	68
5.1.1	Základná štruktúra .....	68
5.1.2	Hlavička dokumentov.....	69
5.1.3	Implementácia Javascriptov .....	69
5.1.4	CSS kaskádové štýly .....	70
5.1.5	Úvodná obrazovka .....	71
5.1.6	Navigácia a módy modelu .....	73
5.1.7	Základné rozloženie prvkov .....	73





5.1.8	Mód prehliadania a navigácie.....	74
5.2	Implementácia prototypu.....	77
5.2.1	Modelovanie poschodia .....	77
5.2.2	Práca s modelom v O3D .....	78
5.2.3	Implementácia navigácie.....	79
5.2.4	Komunikácia s databázou.....	81
6	zoznam použitej literatúry.....	83





## ZOZNAM TABULIEK

Tab. 1 Prípád použitia UC01 Zvolenie módu zobrazenia.....	40
Tab. 2 Popis prípadu použitia UC02 Pohybovanie sa po budove .....	41
Tab. 3 Popis prípadu použitia UC03 Zobrazenie informácií .....	43
Tab. 4 Popis prípadu použitia UC04 Otvorenie a zatvorenie dverí.....	44
Tab. 5 Popis prípadu použitia UC05 Použitie výťahu .....	45
Tab. 6 Popis prípadu použitia UC06 Voľba poschodia .....	47
Tab. 7 Popis prípadu použitia UC07 Prehliadanie modelu poschodia .....	48
Tab. 8 Popis prípadu použitia UC08 Vyhľadávanie miestnosti .....	50
Tab. 9 Popis prípadu použitia UC09 Navigácia medzi miestnosťami .....	51





## 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.....	23
Obr. 12 Obrazovka programu HTMLKit.....	24
Obr. 13 Obrazovka programu Eclipse .....	25
Obr. 14 Obrazovka programu Google Sketchup.....	26
Obr. 15 Obrazovka pluginu Firebug v prehliadači Firefox.....	27
Obr. 16 Obrazovka aplikácie The Forbidden City: Beyond Space & Time .....	28
Obr. 17 Obrazovka stránky virtualtravel.sk.....	29
Obr. 18 Demo Beach Scene .....	30
Obr. 19 Material Editor dialóg.....	32
Obr. 20 Export do COLLADA formátu.....	33
Obr. 21 Okno programu 3DS Max.....	34
Obr. 22 Prípád použitia pre bežného používateľa .....	38
Obr. 23 Prípád použitia „Zvolenie módu zobrazenia“ .....	39





Obr. 24 Prípád použitia „Pohybovanie sa po budove“ .....	40
Obr. 25 Prípád použitia „Zobrazenie informácií“ .....	42
Obr. 26 Prípád použitia „Otvorenie a zatvorenie dverí“ .....	43
Obr. 27 Prípád použitia „Použitie výťahu“ .....	44
Obr. 28 Prípád použitia „Voľba poschodia“ .....	46
Obr. 29 Prípád použitia „Prehliadanie modelu poschodia“ .....	47
Obr. 30 Prípád použitia „Vyhľadávanie miestnosti“ v navigačnom móde .....	48
Obr. 31 Prípád použitia „Navigácia medzi miestnosťami“ .....	50
Obr. 32 Architektúra systému - diagram komponentov .....	57
Obr. 33 Úvítacia obrazovka .....	58
Obr. 34 Obrazovka vyhľadávania miestností.....	60
Obr. 35 Obrazovka vyhľadávania cesty .....	61
Obr. 36 Obrazovka prehľadávania .....	62
Obr. 37 Dátový model .....	63
Obr. 38 Fyzický model .....	64
Obr. 39 Editor PSPad 4.5.4 .....	68
Obr. 40 Implementovaná úvodná obrazovka.....	72
Obr. 41 Obrazovka módu prehliadania.....	75
Obr. 42 Obrazovka módu navigácie.....	76
Obr. 43 Chybné zobrazenie modelu v O3D – priehľadné steny .....	77
Obr. 44 Zobrazenie modelu v 3DS Max – priehľadnosť stien sa neprejavila .....	78
Obr. 45 GUI - Navigácia medzi dvomi miestnosťami .....	80
Obr. 46 Chyba pri navigácii – skracovanie cez neverejné priestory .....	81
Obr. 47 Diagram komunikácie prototypu s databázou .....	82







## 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

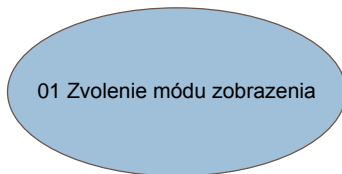
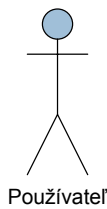
<b>BP</b>	bakalárska práca
<b>DP</b>	diplomová práca
<b>X3D</b>	eXtensible 3D
<b>O3D</b>	open 3D
<b>3DS Studio Max</b>	nástroj na tvorbu 3D modelov
<b>XML</b>	eXtensible Markup Language -jazyk pre uchovávanie štruktúrovaných dát
<b>API</b>	application programming interface
<b>GPU</b>	graphics processing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





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

### 1.3 Použitá anotácia UML



používateľ

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 pokračovaní 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) zobraziť 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.

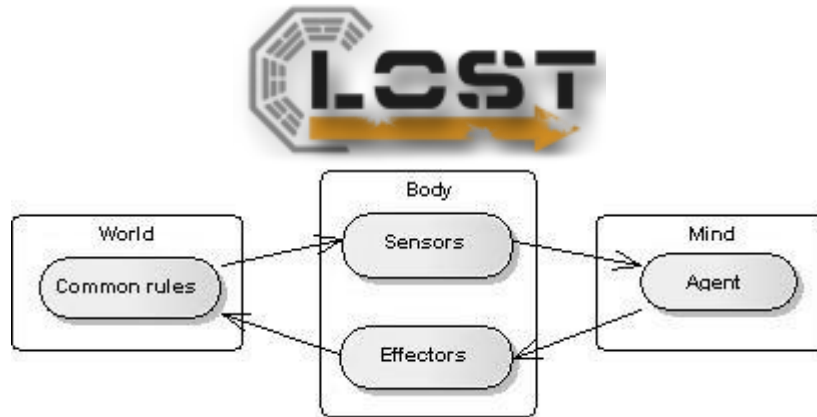
### Významnejšie tematické prierezy

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:

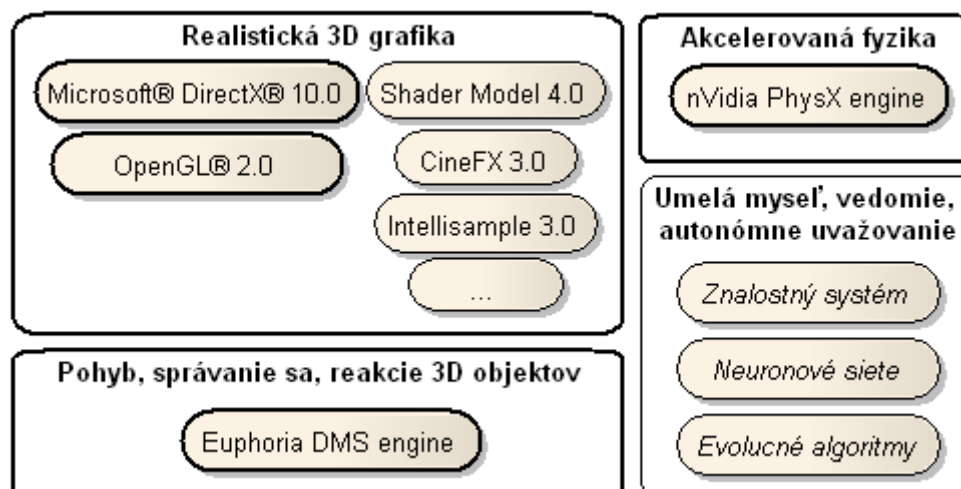
- Simspark
- Webots
- Endorphin

### 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
- Vizuálne 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 zobraziť 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 studiu, ale iba jedno podlažie, ale mal vyriešené väčšie detaily.
- 3D model Zjaru sa nepodaril dobre zobraziť, 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:

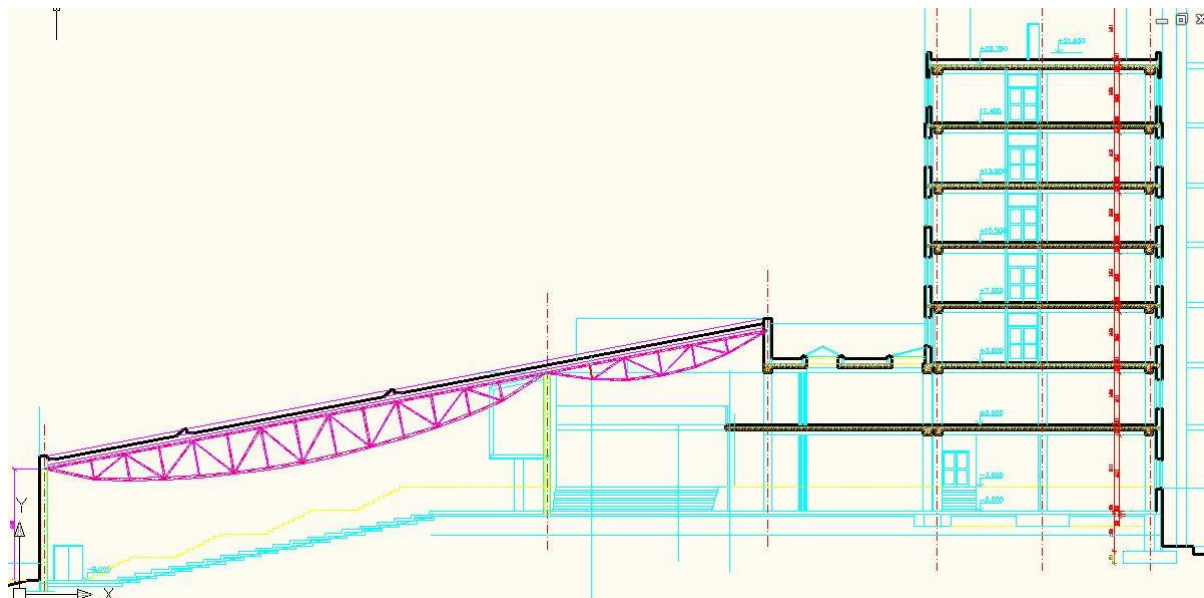
- Vzhľadom na techniku O3D, ktorú plánujeme použiť, tak je k nej určený a prispôsobený modelovací nástroj **Google SketchUp 7**. Pri tvorbe modelov umožňuje rozloženie objektov na komponenty, ktoré sú v O3D určené na prácu s animáciami. Taktiež je potrebná iba jedna konverzia modelu. Ak bude potrebné 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.



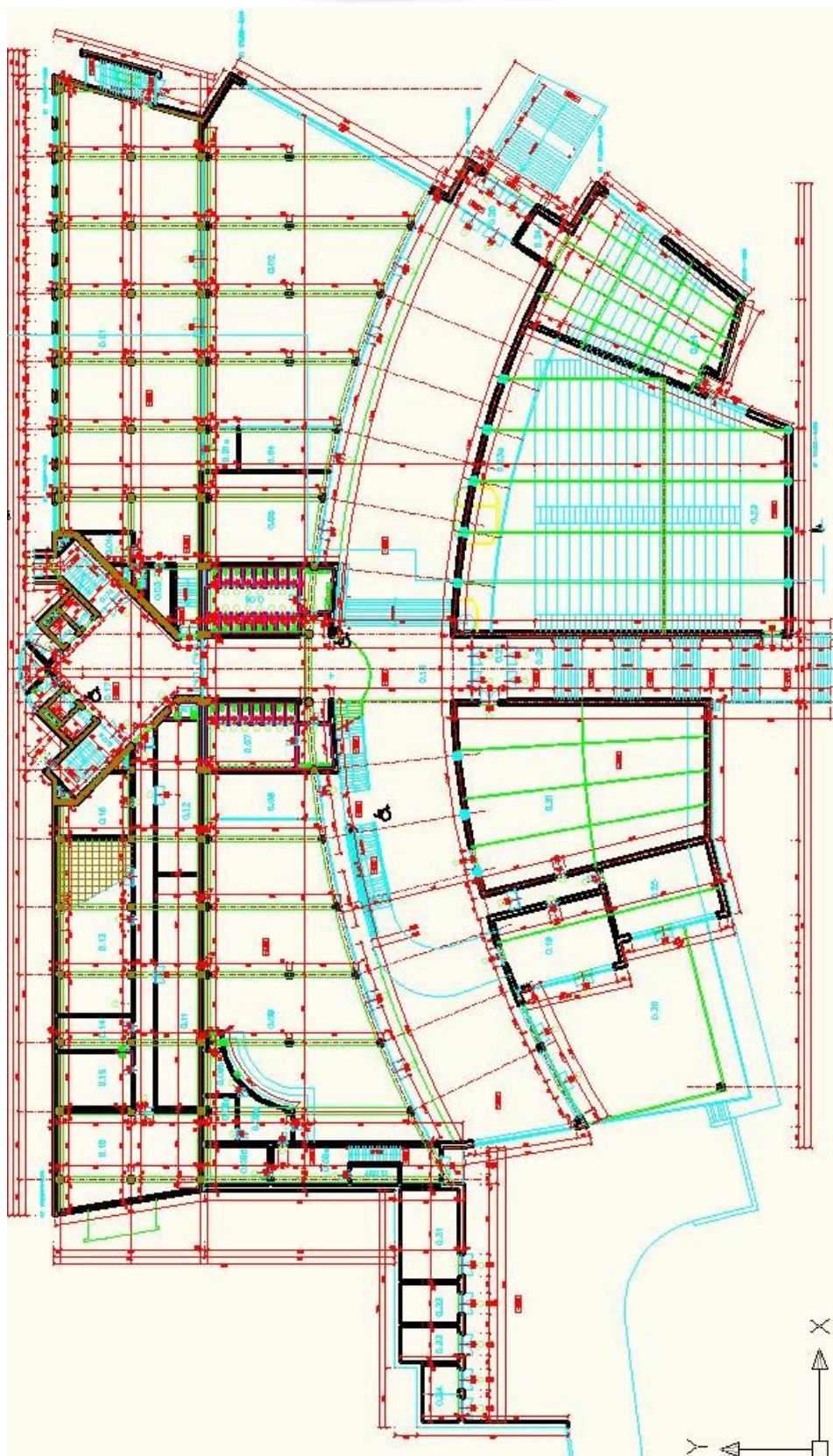


### 2.2.1 Príklady výkresov .dwg:



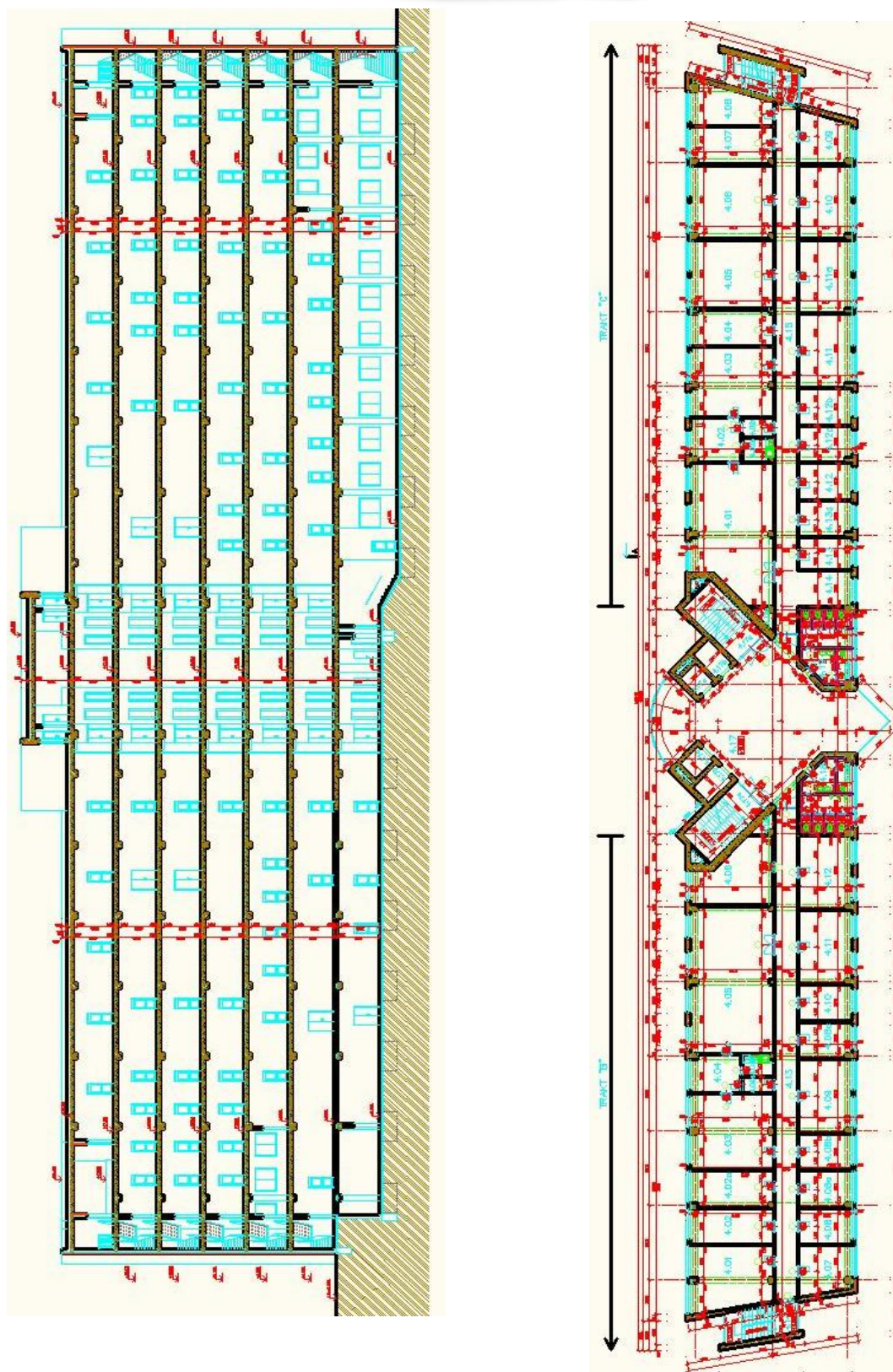
Obr. 3 Bočný prierez budovou





Obr. 4 Pôdorys 1. podlažia budovy



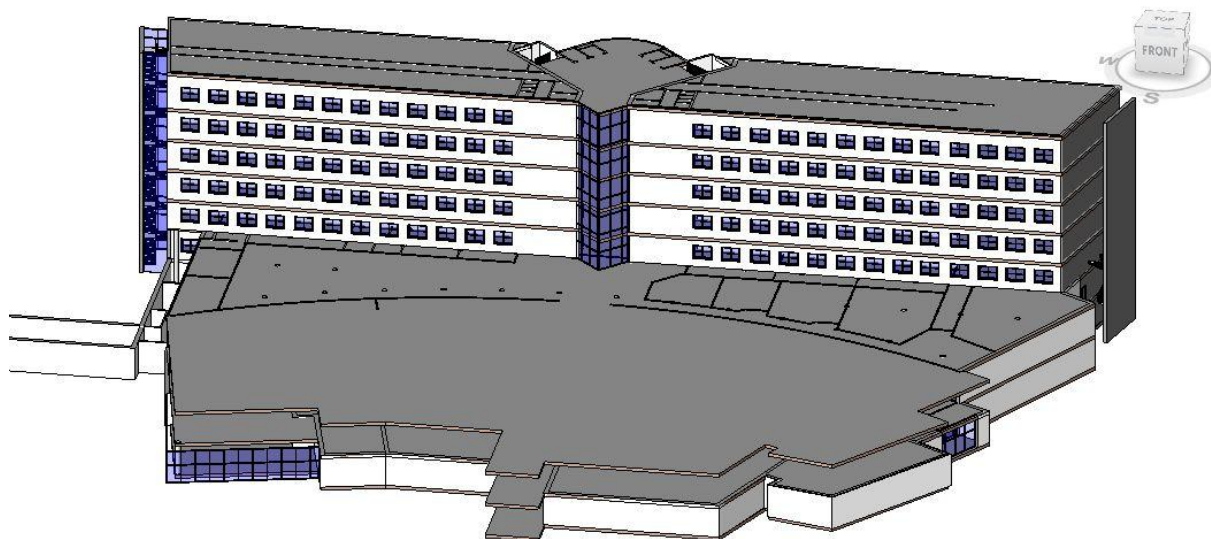


Obr. 5 Prierez poschodiami budovy a pôdorys 4. podlažia budovy

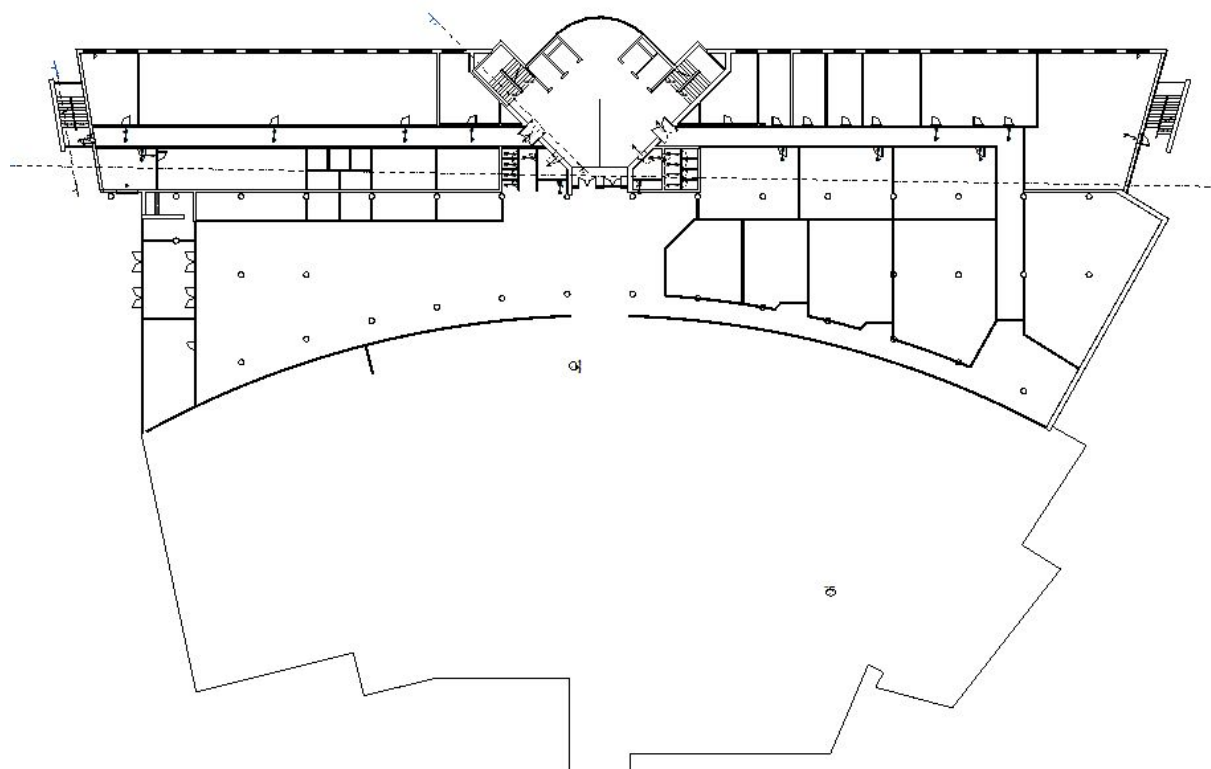




## 2.2.2 Príklady výkresov .rvt:



Obr. 6 Ivančíkov model -1



Obr. 7 Ivančíkov model -2 – prierez





## 2.3 Analýza O3D

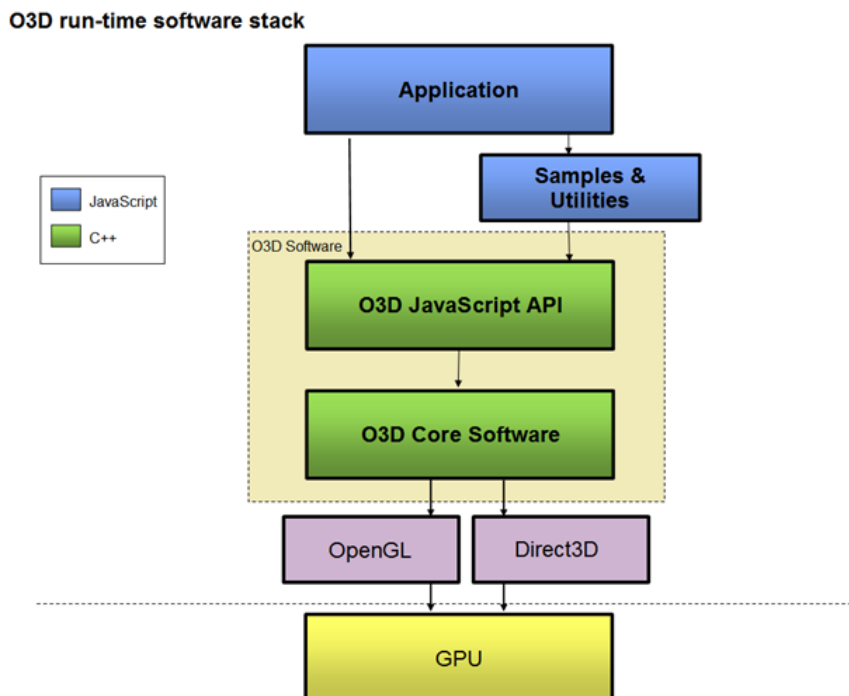
### 2.3.1 Plugin O3d

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 novej (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

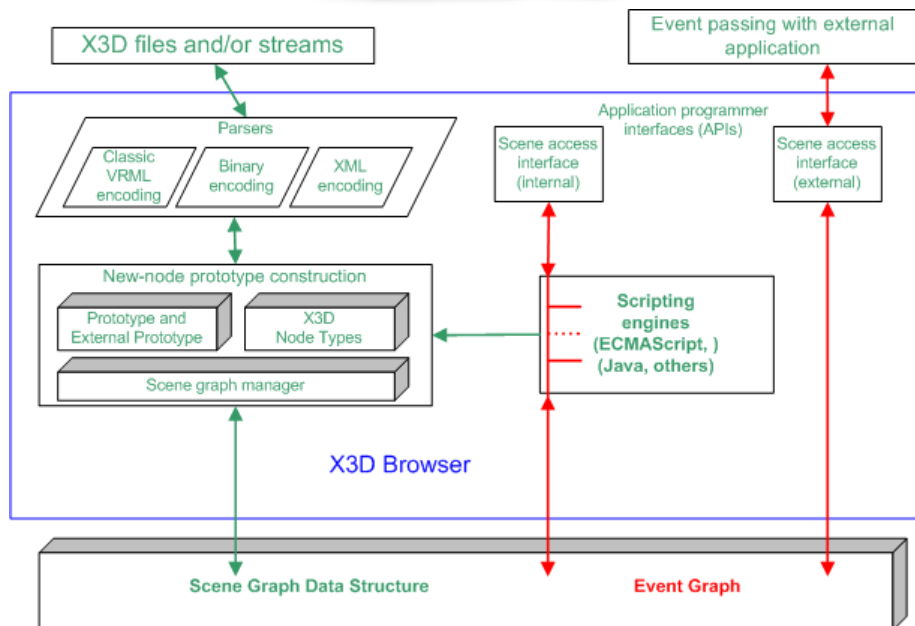
Súbor vo formáte X3D je len pasívny zápis, na vytvorenie viditeľnej trojrozmernej scény je potrebný tzv. X3D prehliadač (browser). Je to program, ktorý načíta vstup vo formáte X3D a zobrazí ho v reálnom čase. Okrem toho podporuje animácie a interakciu s používateľom. Prehliadače sú buď samostatné aplikácie, 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

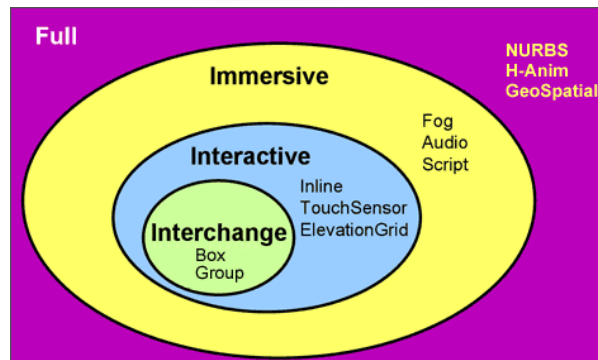
Z pohľadu tvorcu programu sú tu podstatné dve veci: X3D prehliadač načítava pasívne modely z X3D súborov. Prostredníctvom API prehliadača môže k objektom scény pristupovať iná aplikácia (vytvorená napr. v C, C++ alebo Java) a manipulovať s nimi. Používateľ pracuje len s prehliadačom, jeho akcie (pr.: 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 funkcionálna 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, priehľadnosť...) 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.



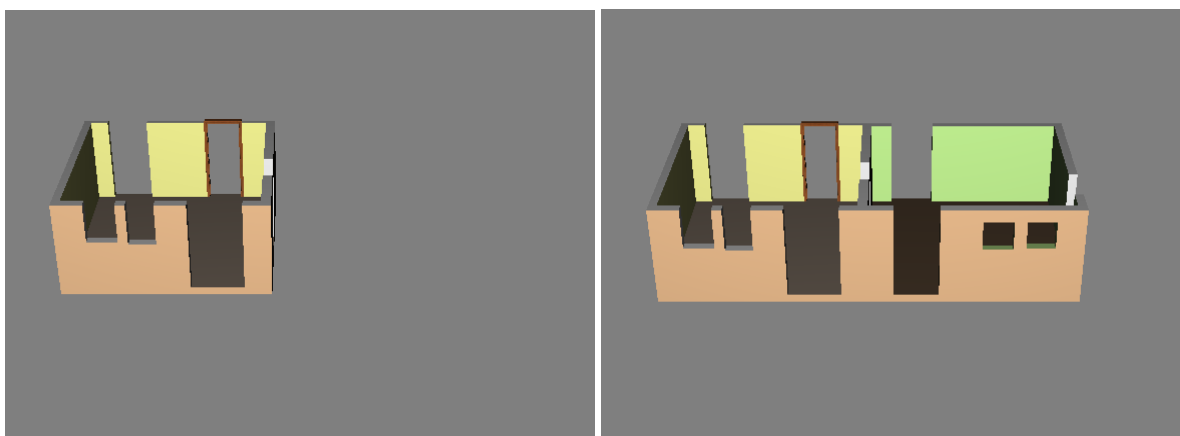


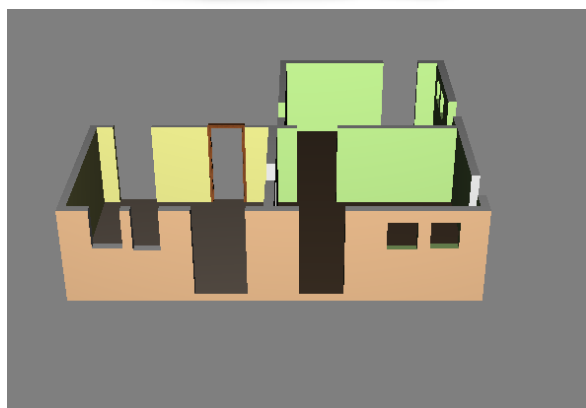
## 2.5 Loadovanie priestorov pre systém Virtuálnej FIIT

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.







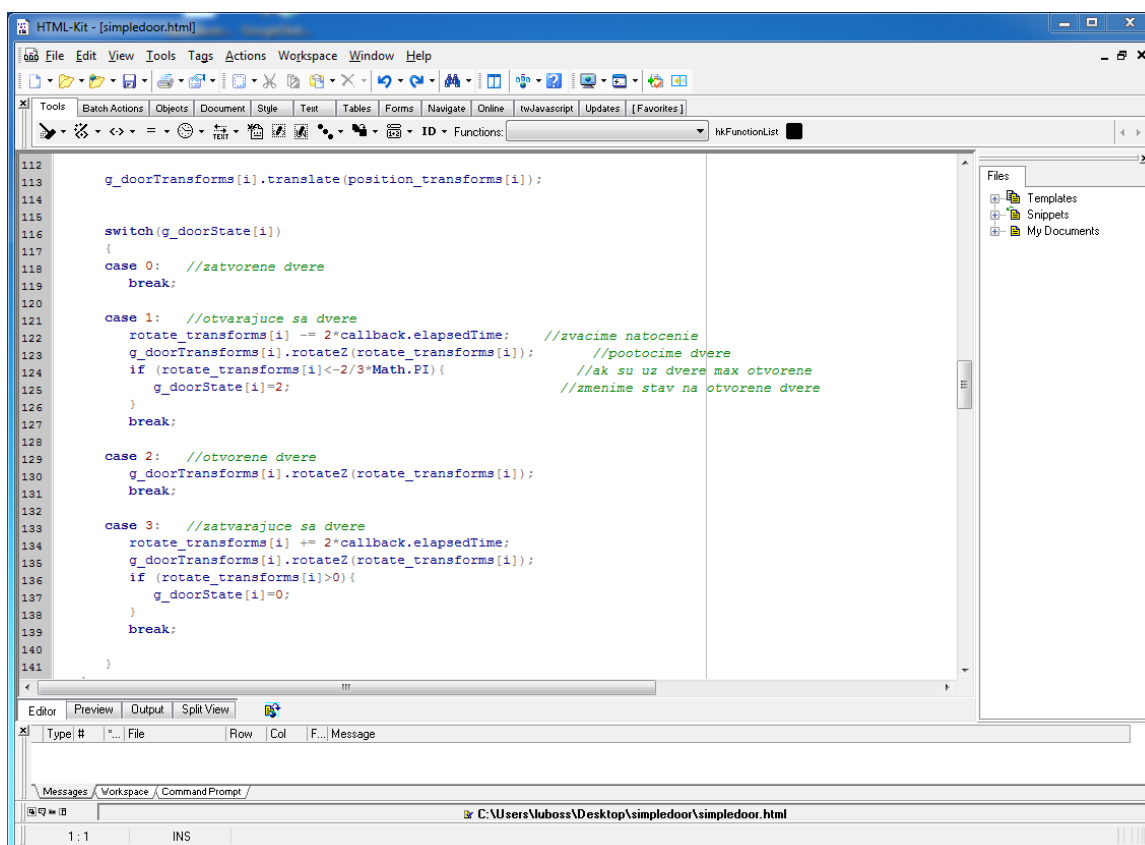
## 2.6 Analýza vývojového prostredia Virtuálnej FIIT

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





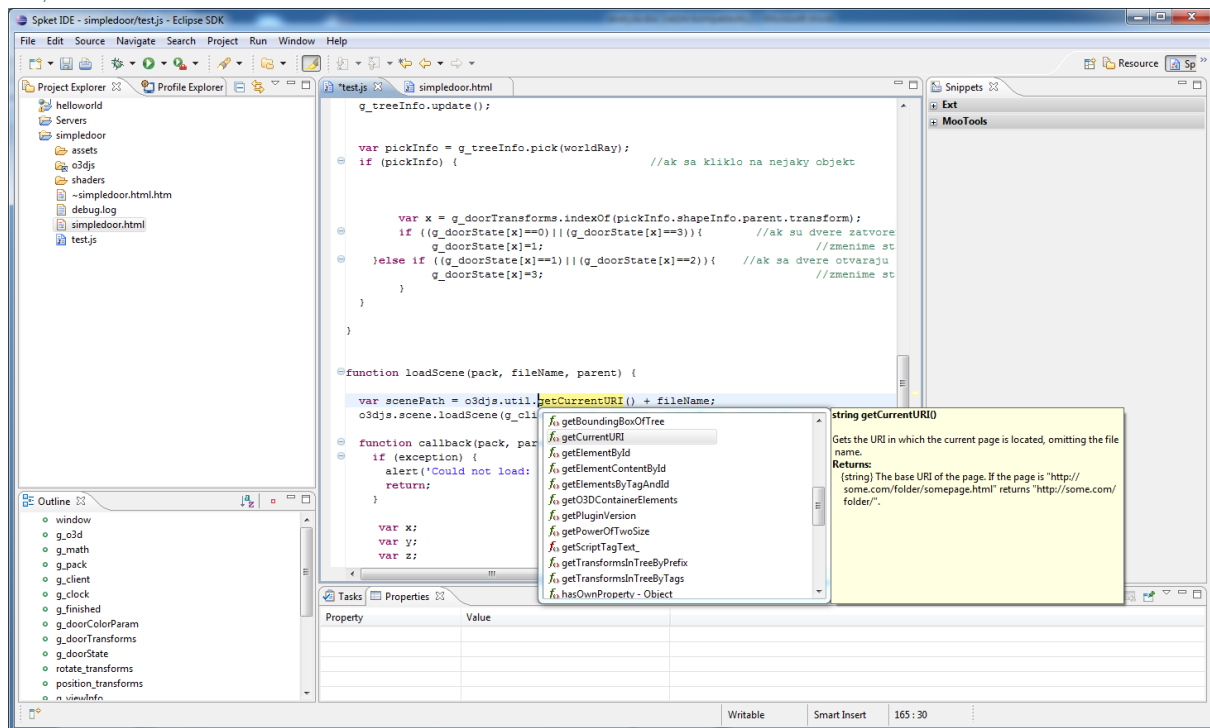
## Eclipse

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 zvyrazňovanie 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 (pričínok 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



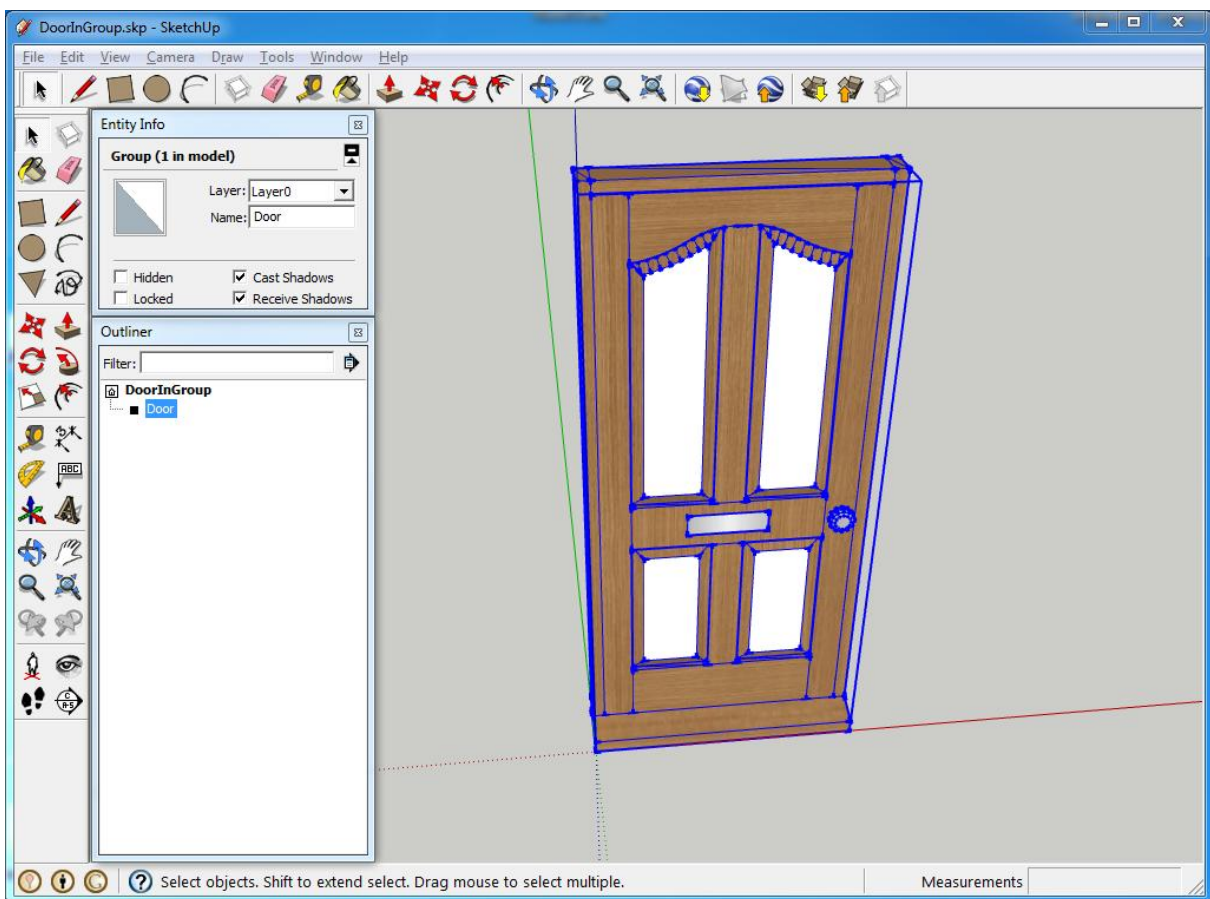


pluginmi. Podobne je možné rozšíriť aj jeho funkcionality. V našom projekte využijeme najmä export do formátu COLLADA (prípina „.dae“), ktorý sa po konverzii na formát JSON (prípina „.o3dtgz“) dá importovať do O3D projektu.

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 (prípina „.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 Obrázok programu Google Sketchup

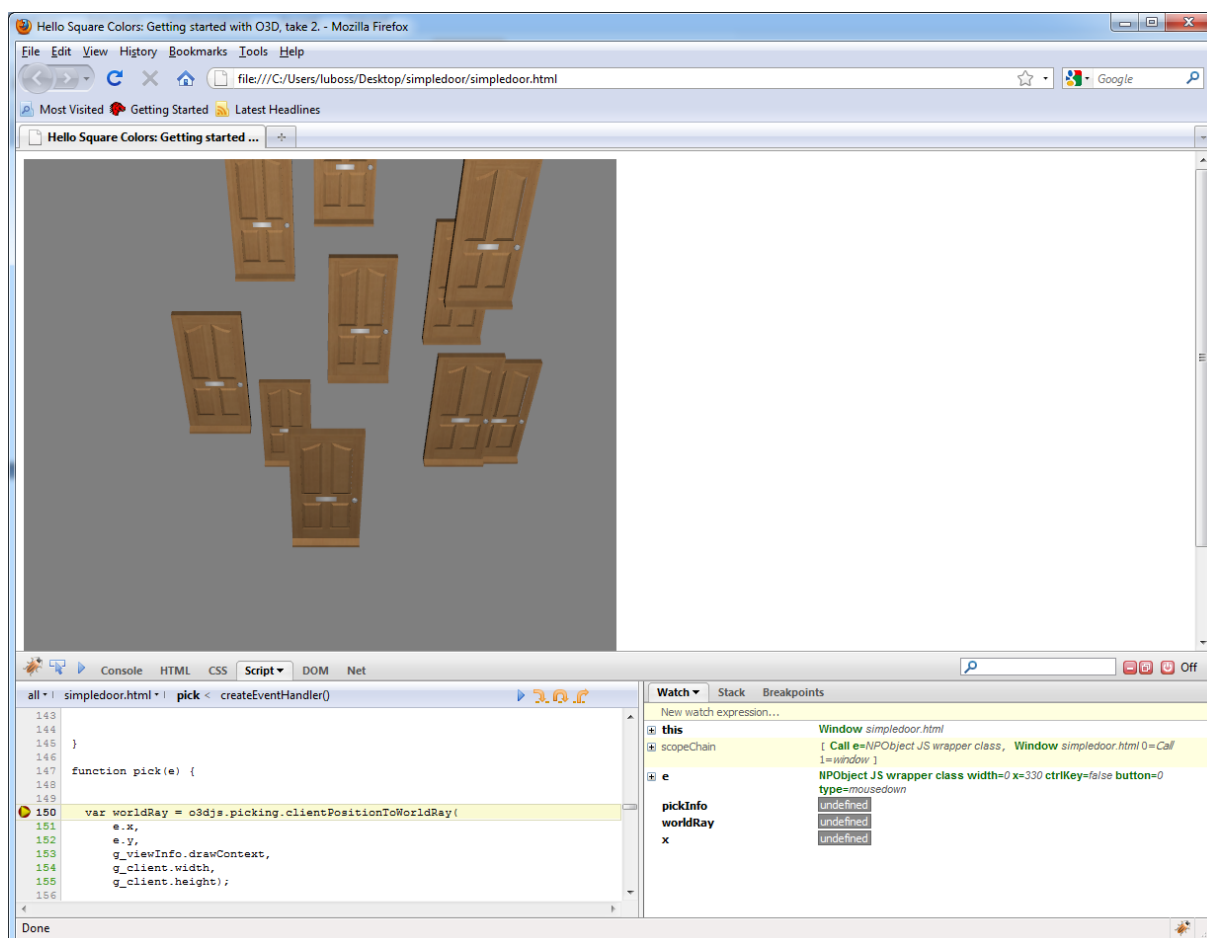




### 2.6.3 Debugovanie Javascript kódu

#### 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.

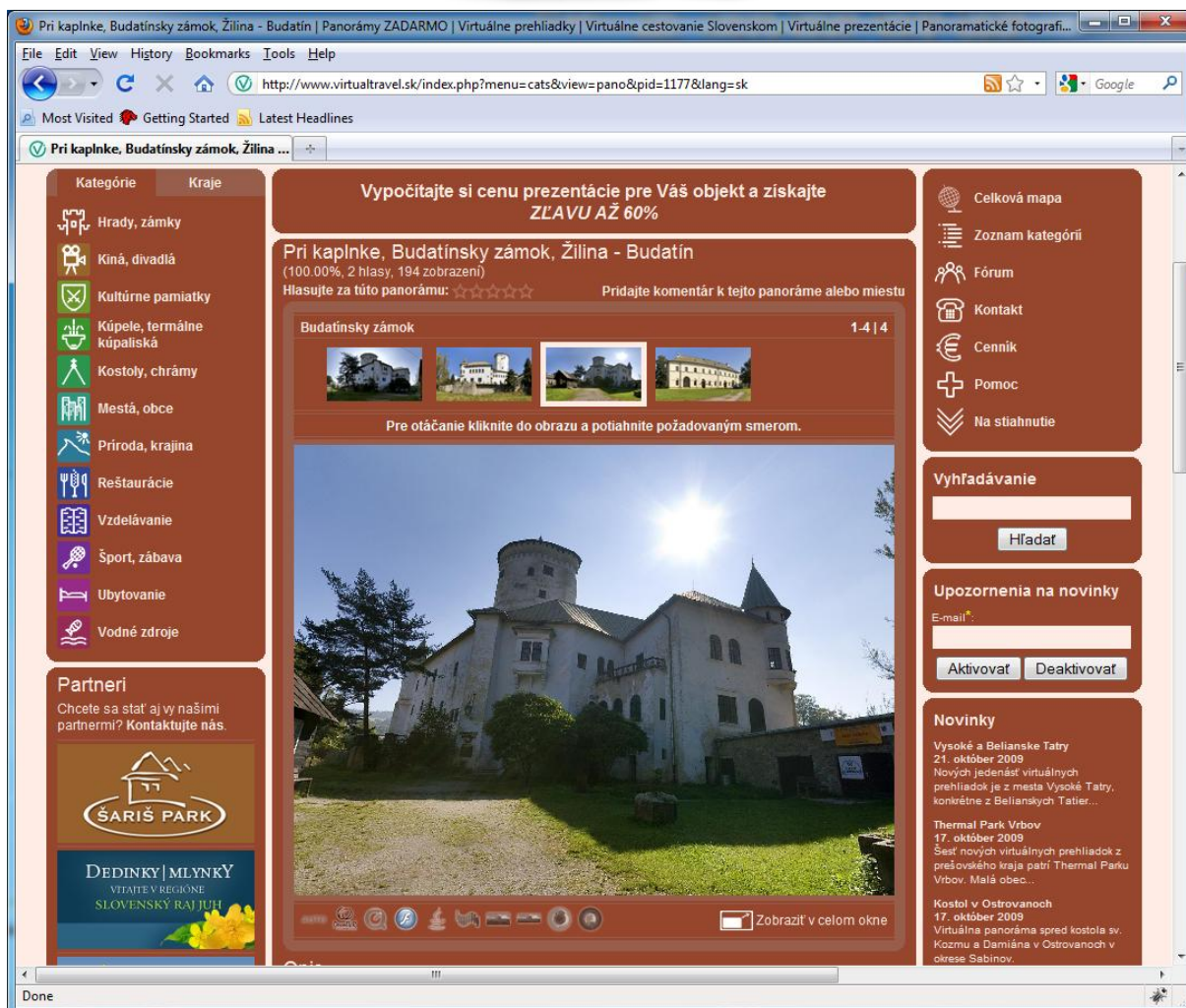


Obr. 16 Obrazovka aplikácie The Forbidden City: Beyond Space & Time

### 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.



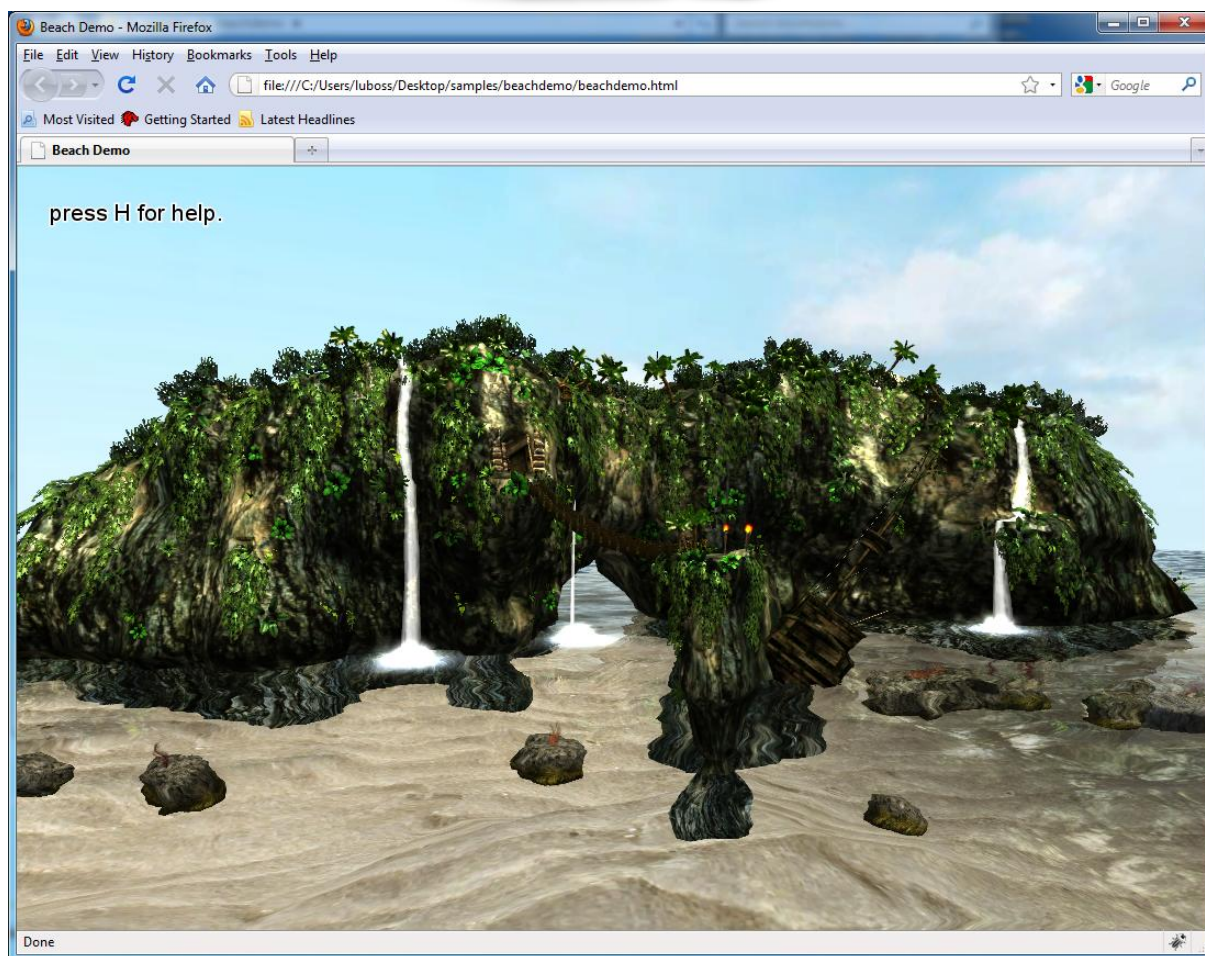


Obr. 17 Obrazovka stránky virtualtravel.sk

## 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



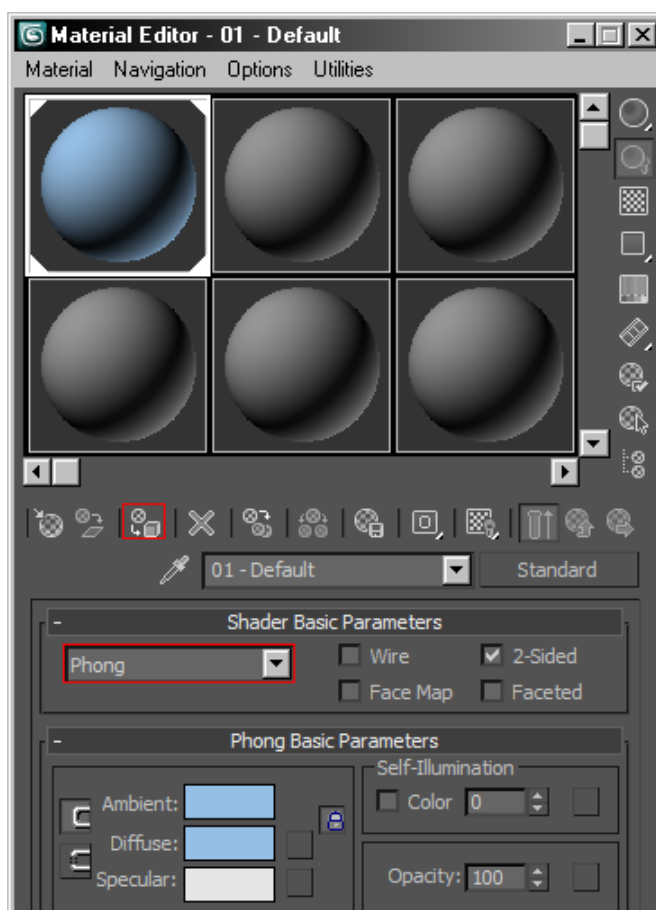




vyvolá dialóg, v ktorom sa určí ktoré osi oboch objektov sa majú zarovnať. Objekty zarovnané do jedného tvaru sa spoja do jedného objektu tak, že v záložke Modify (pravý panel) niektorého z objektov sa klikne na Attach a zvolí sa druhý objekt. Novovzniknutý útvar je vhodné pomenovať (pole na vrchu záložky).

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

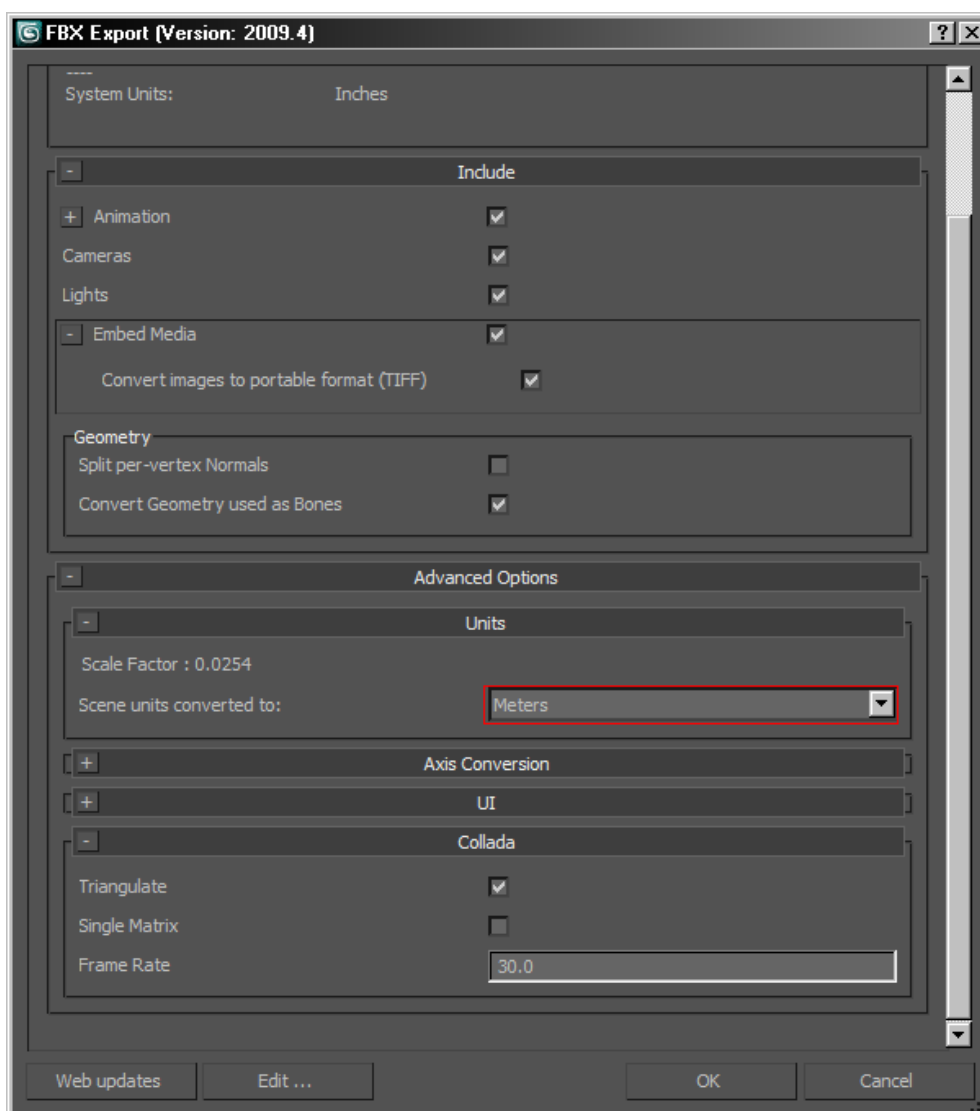
### Export do COLLADA formátu





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.



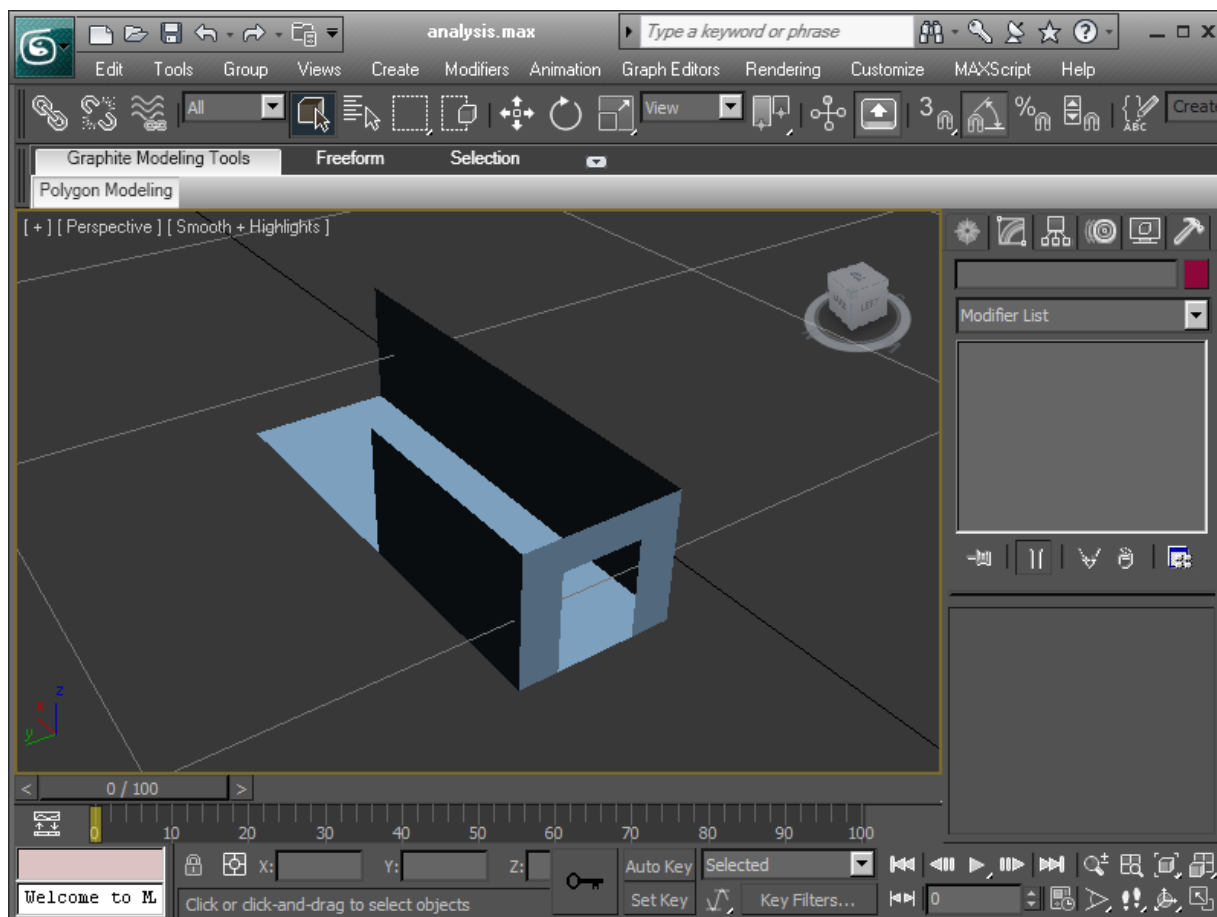
Obr. 20 Export do COLLADA formátu

## 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

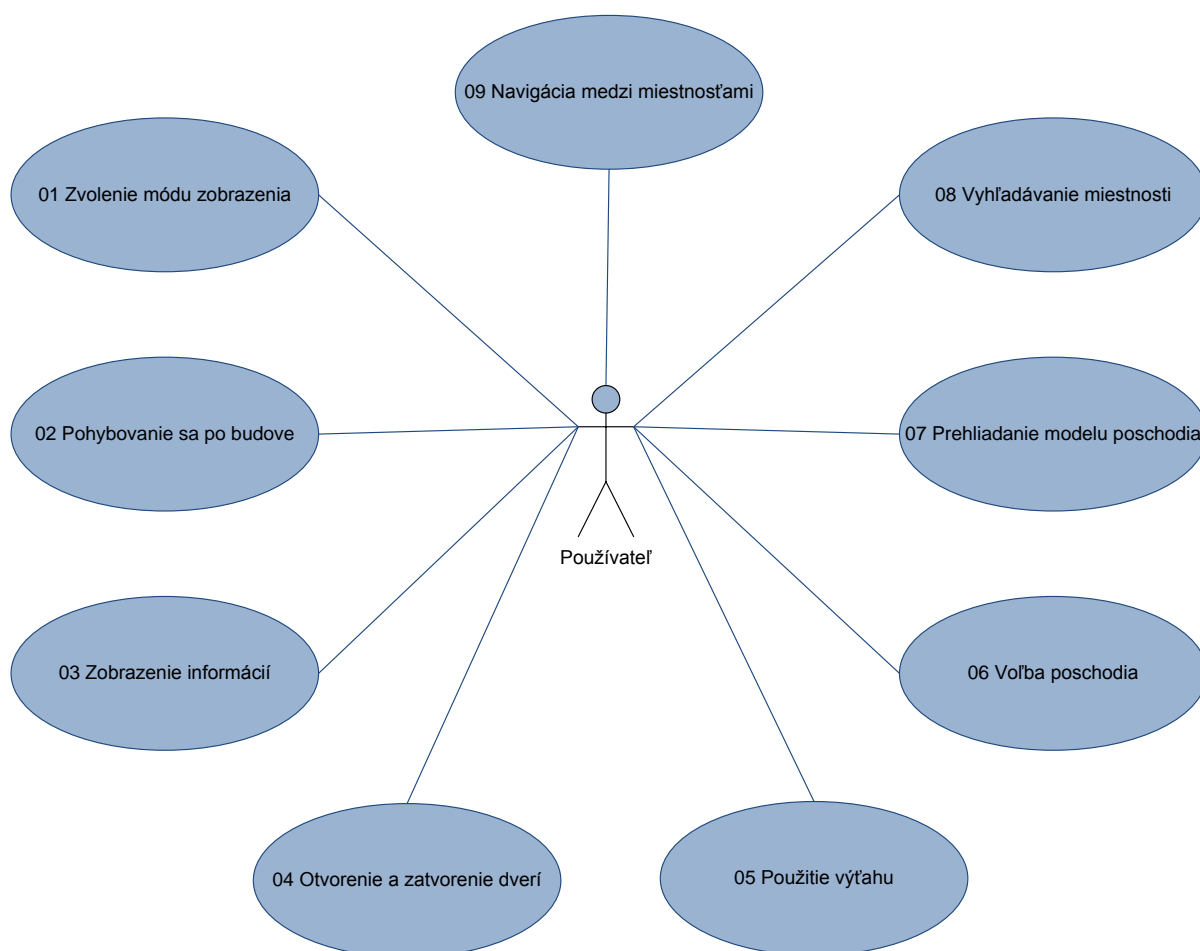
Vychádzajúc z kapitoly C.1.1 a podľa požiadaviek zadávateľa je celková funkcionality zameraná na používateľa. Práve ten je so systémom v najužšom kontakte a s ním súvisí väčšina interakcií.





### 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.

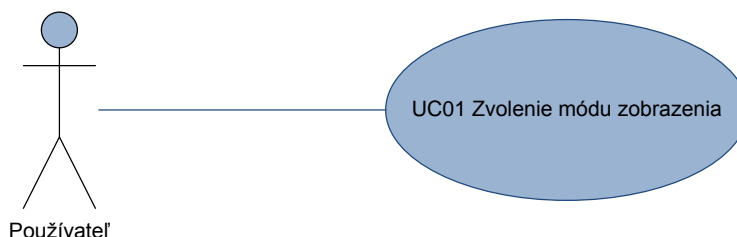


Obr. 22 Prípad použitia pre bežného používateľa





### 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.

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



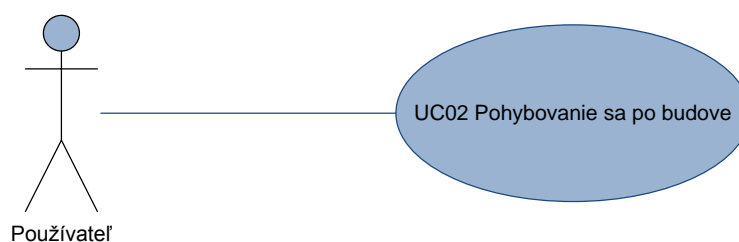




<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	1	Systém dáva používateľovi možnosť výberu z 3 módov zobrazenia
	2	Používateľ klikne na jeden z nich („3D“, „Prehliadanie“ alebo „Navigácia“)
	3	Ak bol zvolený „3D“ mód, tak systém zobrazí používateľovi model z pohľadu prvej osoby
	4	Pri móde „3D“ môže používateľ ďalej použiť UC02, UC03, UC04 a UC05
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	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
<b>poznámky</b>	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.





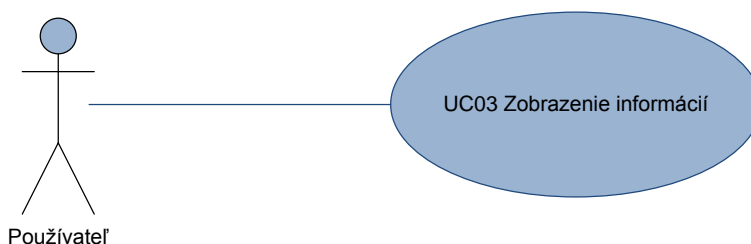
<b>identifikátor</b>	UC02		
<b>názov</b>	Pohybovanie sa po budove		
<b>opis</b>	Používateľ sa môže voľne pohybovať po modeli FIIT		
<b>priorita</b>	1 = vysoká	<b>frekvencia</b>	denne stovky krát
<b>vstupné podmienky</b>	Požívateľ zmenil mód na 3D mód		
<b>výstupné podmienky</b>	Používateľ sa nachádza na novej pozícii alebo sa pozerá iným smerom		
<b>používatelia</b>	Bežný používateľ		
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	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	
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	2a	Používateľ stlačením tlačidla myši a jej pohybom mení smer, ktorým sa pozerá	
<b>poznámky</b>	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).

<b>identifikátor</b>	UC03		
<b>názov</b>	Zobrazenie informácií		
<b>opis</b>	Používateľovi sa v paneli zobrazia informácie o miestnosti alebo budove FIIT		
<b>priorita</b>	1 = vysoká	<b>frekvencia</b>	denne stovky krát
<b>vstupné podmienky</b>	Používateľ sa nachádza v 3D móde, prehliadacom móde alebo navigačnom móde s už vyhľadaným modelom		
<b>výstupné podmienky</b>	Používateľovi sa zobrazia informácie o miestnosti alebo budove FIIT		
<b>používatelia</b>	Bežný používateľ		
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	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	

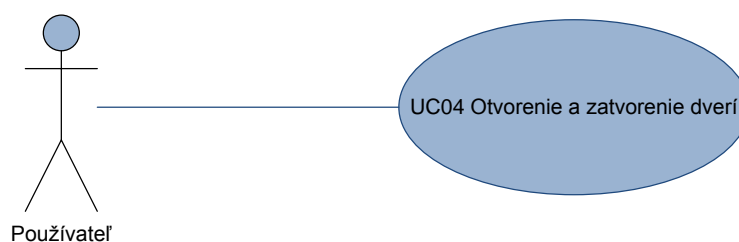




	3	Systém zobrazí aktuálne informácie o danej miestnosti v paneli
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	3a	Ak sa používateľ nachádza v 3D móde pred budovou, zobrazia sa informácie o celej budove
	3b	Ak sa používateľ nachádza v prehliadacom móde bez vybratej miestnosti, zobrazia sa informácie o zobrazenom poschodí
<b>poznámky</b>	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.

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

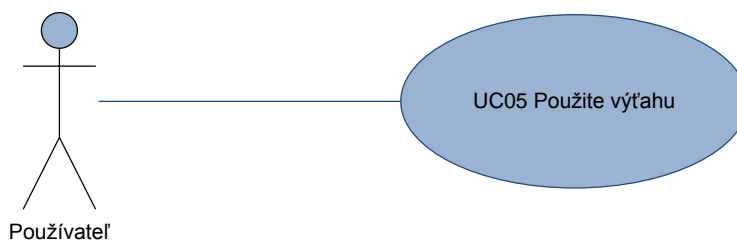




<b>výstupné podmienky</b>	Používateľovi môže/nemôže prejsť dverami	
<b>používatelia</b>	Bežný používateľ	
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	1	Používateľ sa priblíži k dverám na dostatočnú vzdialenosť s priamou viditeľnosťou na dvere
	2	Používateľ klikne na dvere
	3	Ak boli dvere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





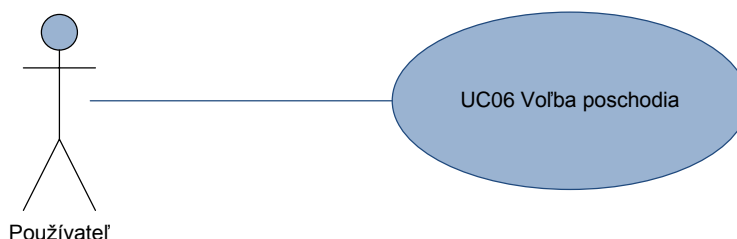
<b>identifikátor</b>	UC05		
<b>názov</b>	Použitie výťahu		
<b>opis</b>	Prechod medzi poschodiami pomocou výťahu		
<b>priorita</b>	3 = nízka	<b>frekvencia</b>	denne desiatky až stovky krát
<b>vstupné podmienky</b>	Zvolený je 3D mód		
<b>výstupné podmienky</b>	Používateľ sa premiestnil na iné poschodie		
<b>používatelia</b>	Bežný používateľ		
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	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	
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>Činnosť</b>	
	6a	Ak užívateľ klikol na číslo poschodia, na ktorom sa nachádza, dvere ostávajú otvorené	
<b>poznámky</b>	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“.

<b>identifikátor</b>	UC06		
<b>názov</b>	Voľba poschodia		
<b>opis</b>	Používateľ si môže zmeniť zobrazené poschodie		
<b>priorita</b>	1 = vysoká	<b>frekvencia</b>	denne stovky krát
<b>vstupné podmienky</b>	Požívateľ zmenil mód na prehliadací mód		
<b>výstupné podmienky</b>	Používateľ má v hlavnom okne zobrazené vybrané poschodie		
<b>používatelia</b>	Bežný používateľ		
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	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...“	

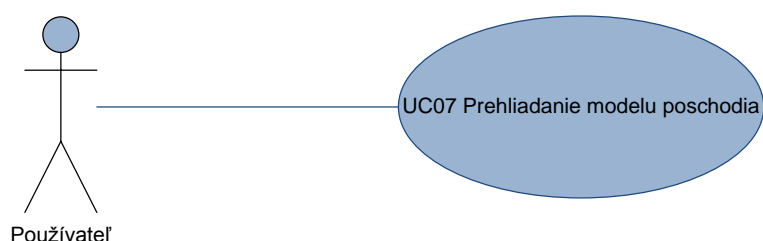




alternatívna postupnosť	krok	Činnosť
	3a	Ak sa používateľ nachádza na najnižšom poschodí a zvolí si zobrazíť poschodie pod alebo sa nachádza na najvyššom poschodí a zvolí si zobrazíť poschodie nad, tak systém na to používateľa upozorní a nezmení poschodie
poznámky	Nie sú	

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.

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



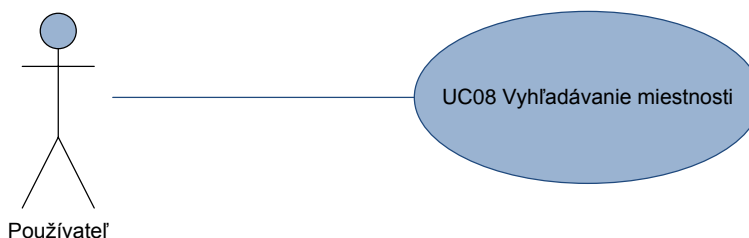




<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	1	Systém nastaví úvodný pohľad, pri ktorom je vidieť celý model poschodia pod miernym uhlom
	2	Používateľ stlačením tlačidla myši a jej pohybom mení uhol, pod ktorým sa na model poschodia pozerá
	3	Systém zobrazuje model z nového pohľadu
<b>alternatívna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

<b>identifikátor</b>	UC08		
<b>názov</b>	Vyhľadávanie miestnosti		
<b>opis</b>	Používateľ pomocou filtra vyhledá miestnosť a nechá si o nej zobrazit informácie		
<b>priorita</b>	1 = vysoká	<b>frekvencia</b>	denne niekoľko desiatok až stoviek krát
<b>vstupné podmienky</b>	Aktívny je navigačný mód		
<b>výstupné podmienky</b>	Používateľovi sa zobrazí zoznam miestností, ktoré vyhovujú kritériám, prípadne informácia, že žiaden vyhovujúci záznam sa nenašiel.		
<b>používatelia</b>	Bežný používateľ		
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	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 prehládací.	
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>činnosť</b>	
	3a	Ak bolo vyhľadávanie neúspešné, zobrazí sa hlásenie „Nebola nájdená žiadna miestnosť pre <input text>“, kde <input text> je text vložený užívateľom	

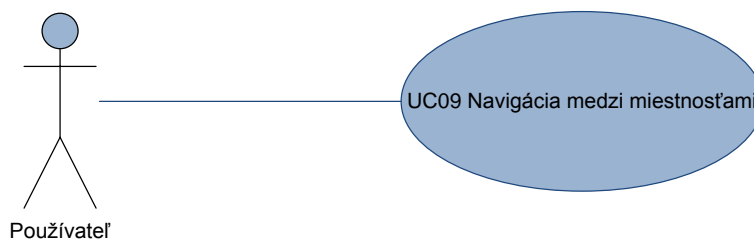




<b>poznámky</b>	Pri určovaní kritérií je nutné zadať vstup o dĺžke aspoň 3 znaky.
-----------------	---

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.

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





<b>používatelia</b>	Bežný používateľ	
<b>základná postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	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	
<b>alternatívna postupnosť</b>	<b>krok</b>	<b>činnosť</b>
	3a,7a	Ak bolo vyhľadávanie neúspešné, zobrazí sa hlásenie „Nebola nájdená žiadna miestnosť pre <input text>“, kde <input text> je text vložený užívateľom
<b>poznámky</b>	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

Architektúra systému je tvorená troma vrstvami systému – prezentačnou, aplikačnou a dátovou, podľa úrovne abstrakcie. Komponenty, ktoré obsahuje, 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.

#### 4.1.1 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 odchyťovanie 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 odchyťovanie pohybov myšou a s tým súvisiacu zmenu pohľadu, z ktorého sa užívateľ 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.

### **4.1.2 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 poschodia a nastavenia pohľadu, zobrazenie informačného panela

#### *Inicializácia navigačnom módu*

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





### *Použitie výťahu*

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

### *Otváranie dverí*

Komponent obsahuje funkčnosť 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 funkčnosť 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 funkčnosť 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 (názov 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 funkcionality pripojenia sa k databáze a komunikáciu s ňou cez SQL príkazy.

### **4.1.3 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.

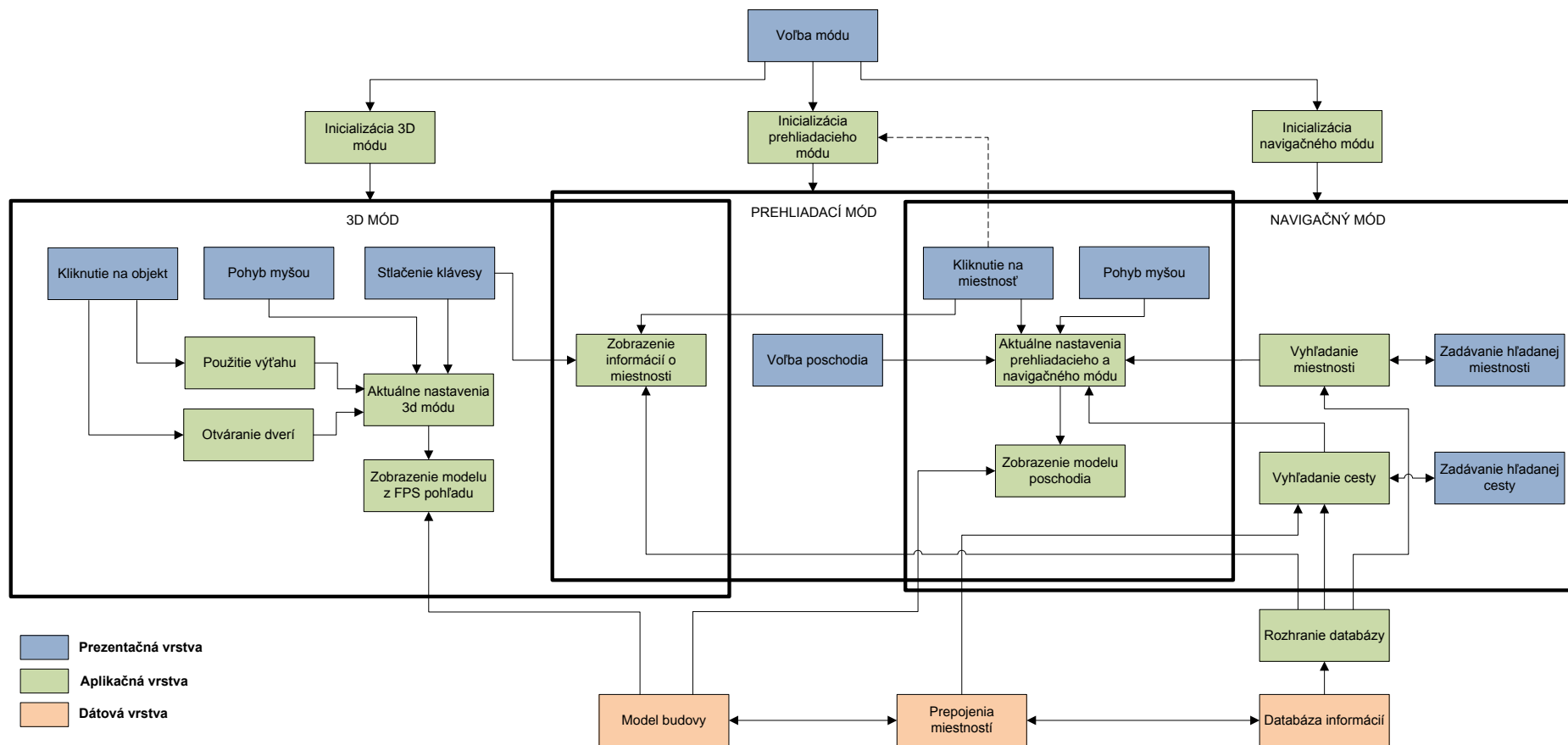
#### *Prepojenia miestností*

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 Architektúra systému - diagram komponentov



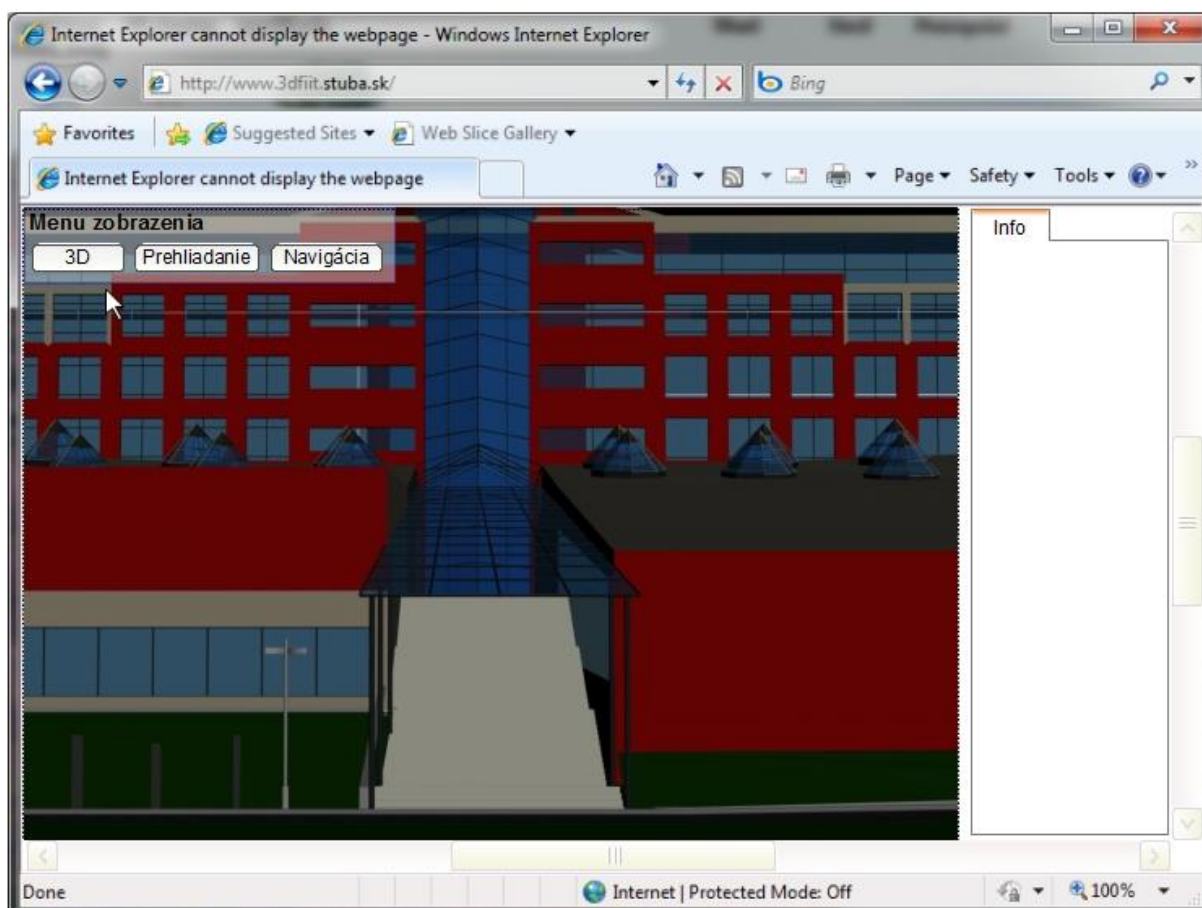


## 4.2 Návrh GUI systému

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.

### 4.2.1 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





#### 4.2.2 Obrazovka pre 3D mód

Obrazovka pre 3D mód je totožná s uvítacou 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.

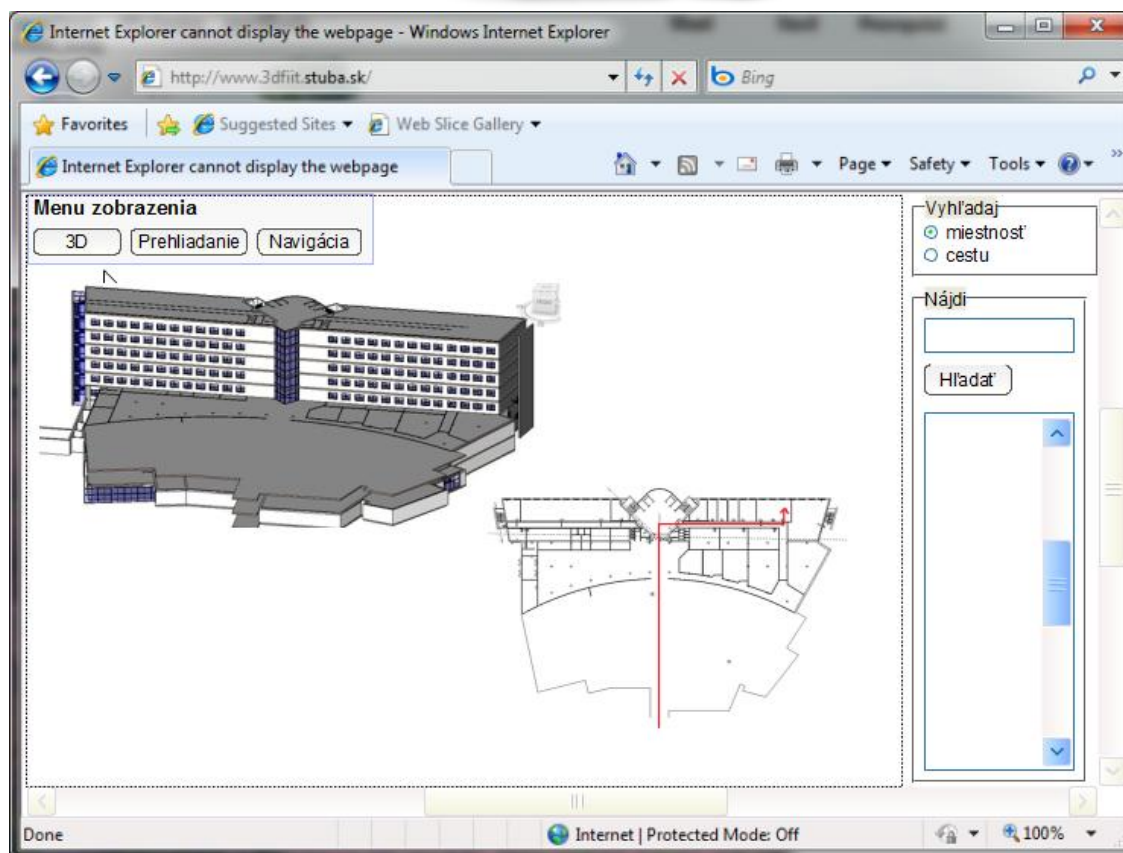
#### 4.2.3 Obrazovka vyhľadávania miestností

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

#### 4.2.4 Obrazovka vyhľadávania cesty

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

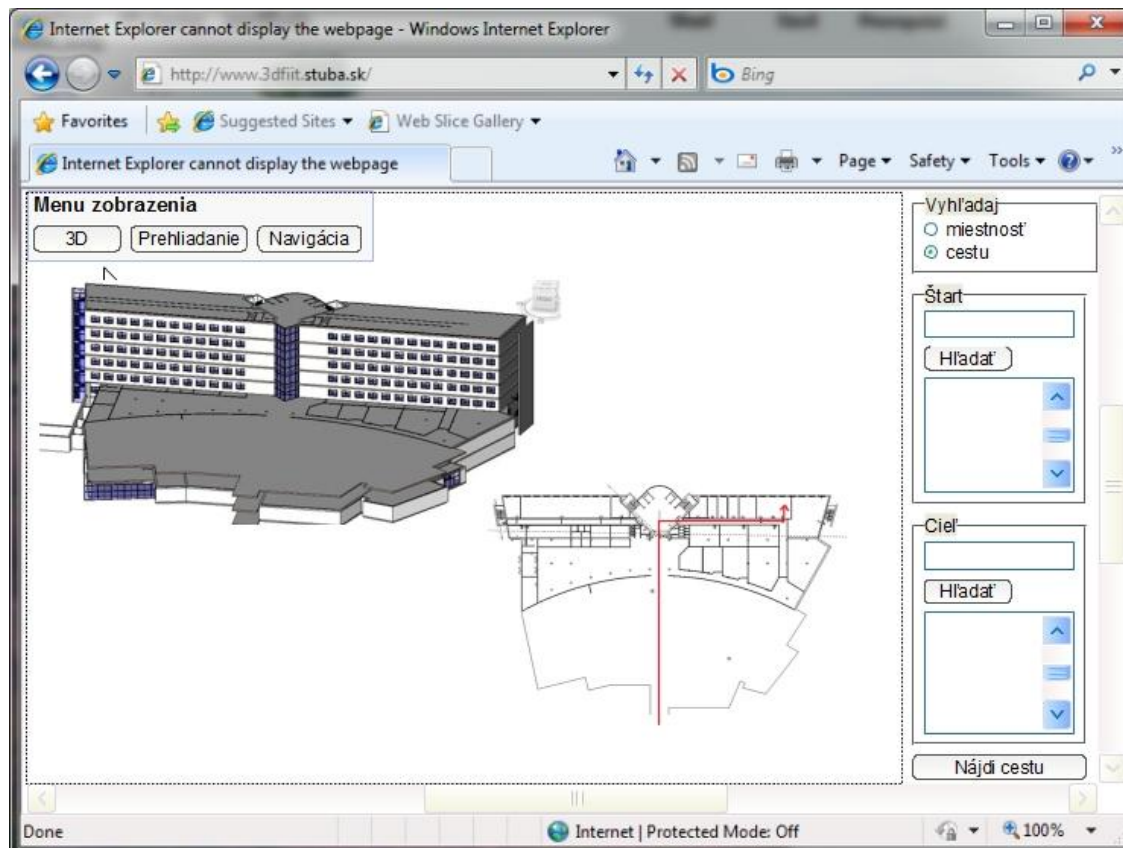
- menu módu zobrazenia s výberovými tlačidlami,
- menu pre výber možností hľadania,
- vstupný formulár pre hľadaný štartovací objekt,
- tlačidlo pre vyhľadanie výsledkov možných štartovacích objektov,
- textové pole s výsledkami možných štartov,
- vstupný formulár pre hľadaný cieľový objekt,
- tlačidlo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

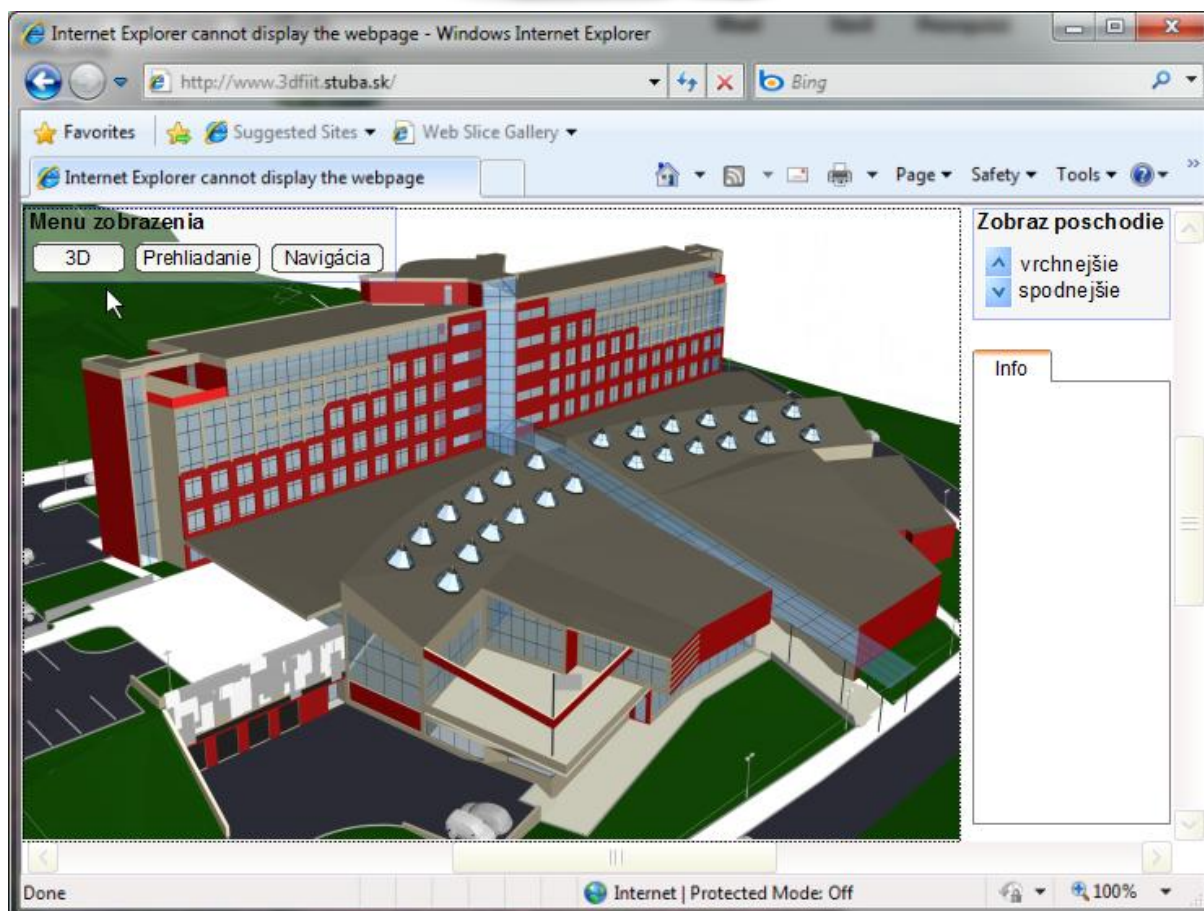
#### 4.2.5 Obrazovka prehľadávania

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

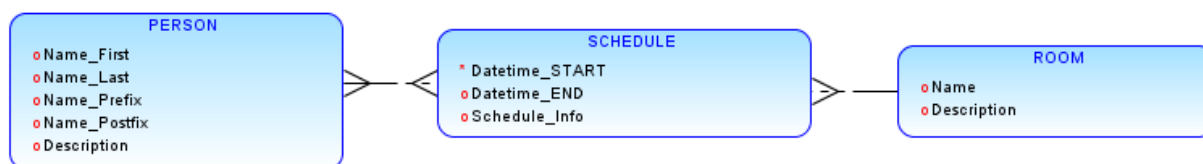




## 4.3 Databázový návrh

### 4.3.1 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. 37 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

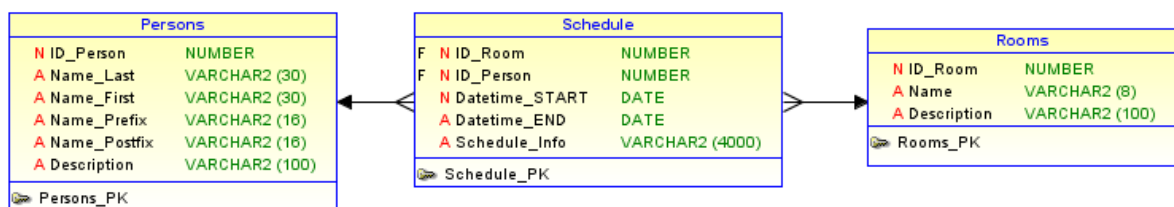
Vzťah popisuje vzťah medzi rozvrhovou akciou a miestnosťou, kedy pre každú miestnosť môže existovať nula, jedna alebo viacero rozvrhových akcií a každá rozvrhová akcia sa koná práve v jednej miestnosti.

### 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.

## 4.3.2 Fyzický model

Fyzický model vychádza z vyššie uvedeného dátového modelu.



Obr. 38 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.

### **4.4 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.





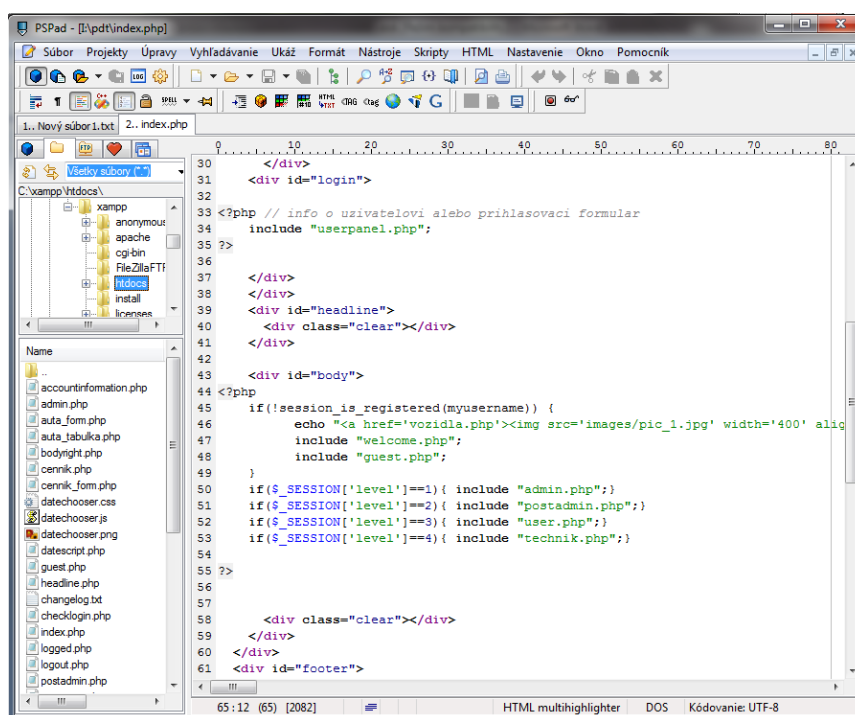
## 5 IMPLEMENTÁCIA SYSTÉMU VIRTUÁLNEJ FIIT

### 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. Po zadaní adresy sa zobrazí uvítacia obrazovka s krátkym popisom 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. 39 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 vyhledá 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="logoo3d"></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>>
```



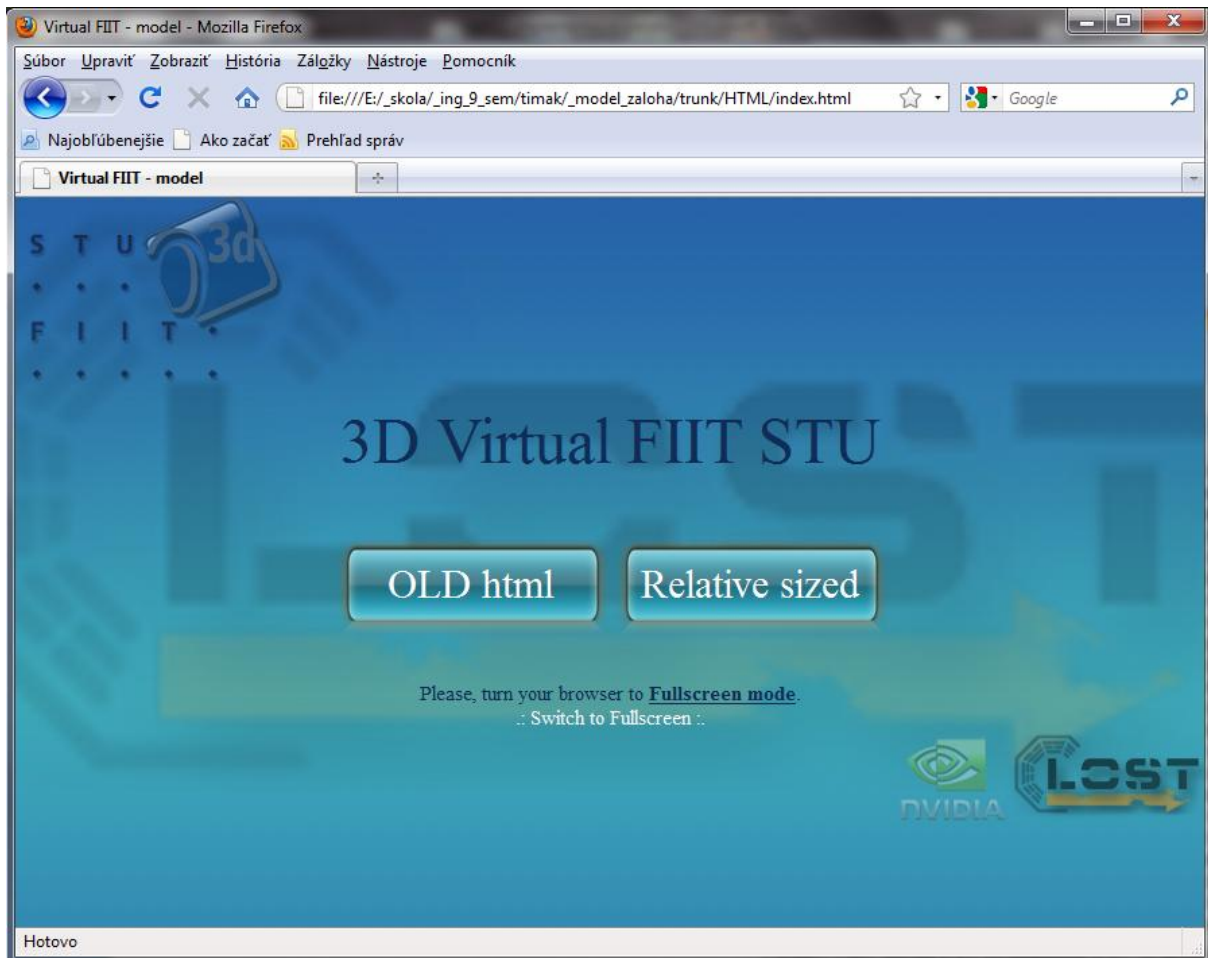




Funkcia skriptu pre prepnutie prehliadača do fullscreen módu:

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

Implementovaný dokument s využitím CSS kaskádových štýlov vyzerá nasledovne :



Obr. 40 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>
    </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 zadefinovaná 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 zadefinované 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(){
```





```

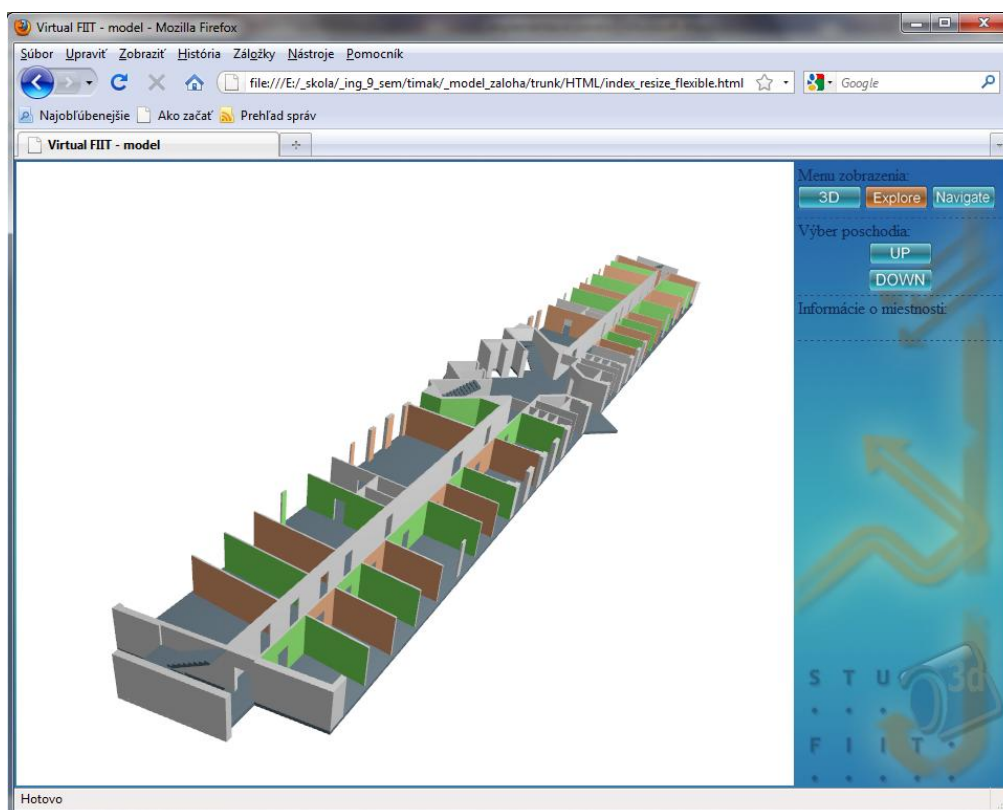
document.getElementById("buttonExplore").src = "../img/explore3.png";
document.getElementById("buttonMode3D").src = "../img/3d.png";
document.getElementById("buttonNavigate").src = "../img/navigate.png";

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.

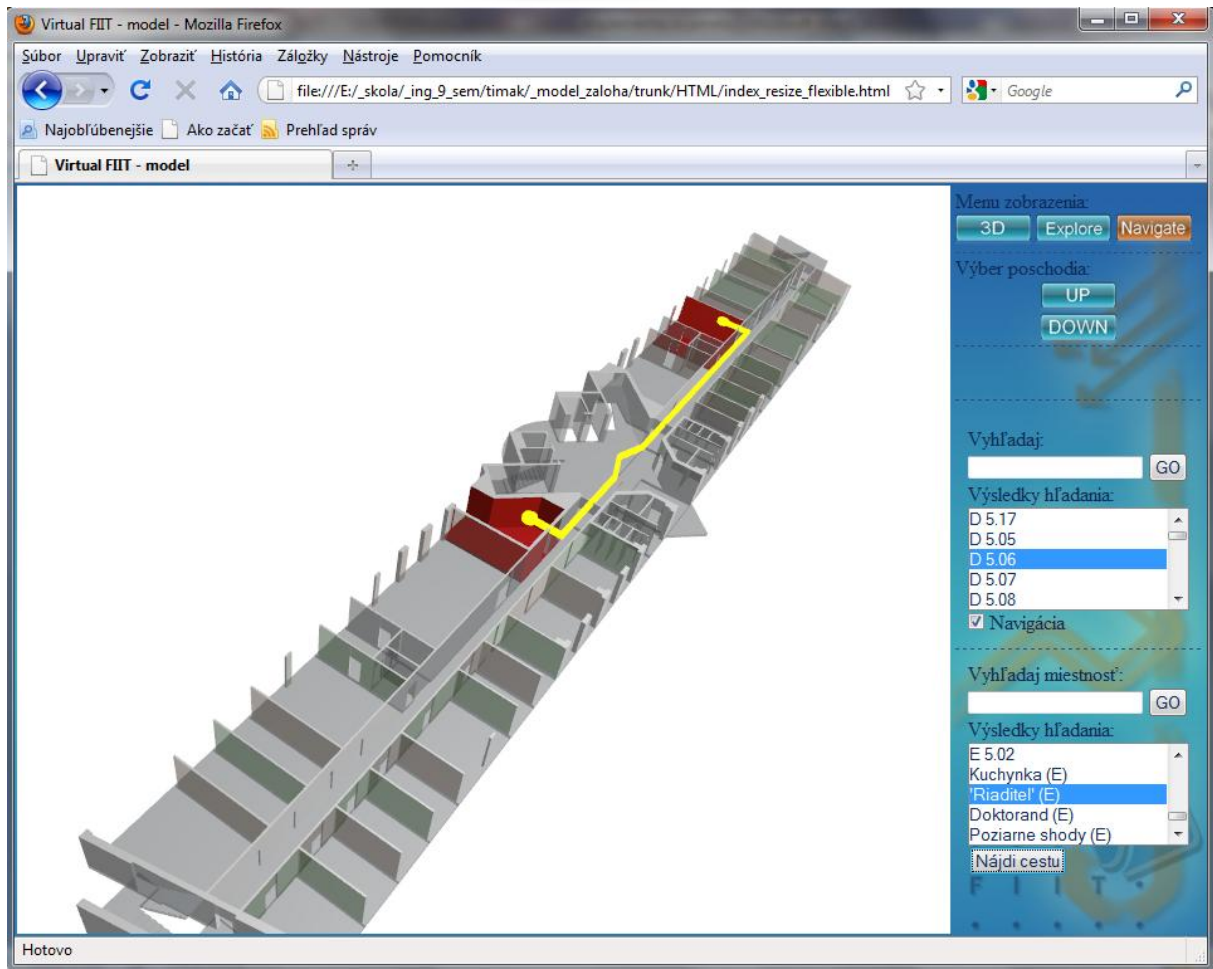
Mód prehliadania obsahuje navigáciu pre zmenu módu zobrazenia a obsahuje prvky pre voľbu poschodia modelu. Ostatné prvky ako textové polia pre navigačný mód sú skryté. Obrázku s 3D modelom je možné ovládať (posun, zoom, označenie miestností). Po označení konkrétnej miestnosti sa pod názvom „Informácie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



Obr. 41 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. 42 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:

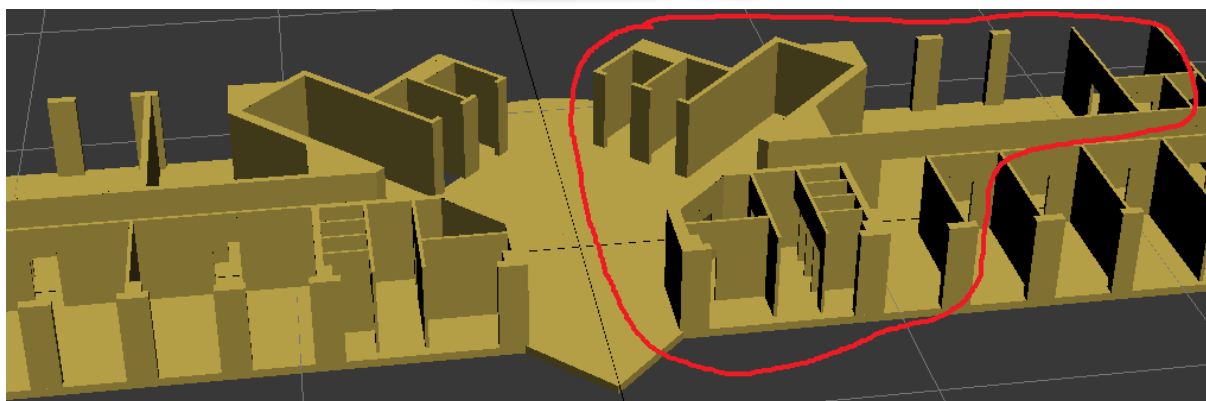
- priesvitnosť stien miestností pri zrkadlení zoskupených objektov
- zarovnávanie vyrezaných alebo pootočených objektov o špecifické uhly (10°, 16°, 45°)
- chybný export komplikovanejšieho modelu do formátu DAE v 3DS Max

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. 43 Chybné zobrazenie modelu v O3D – priehľadné steny





Obr. 44 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.

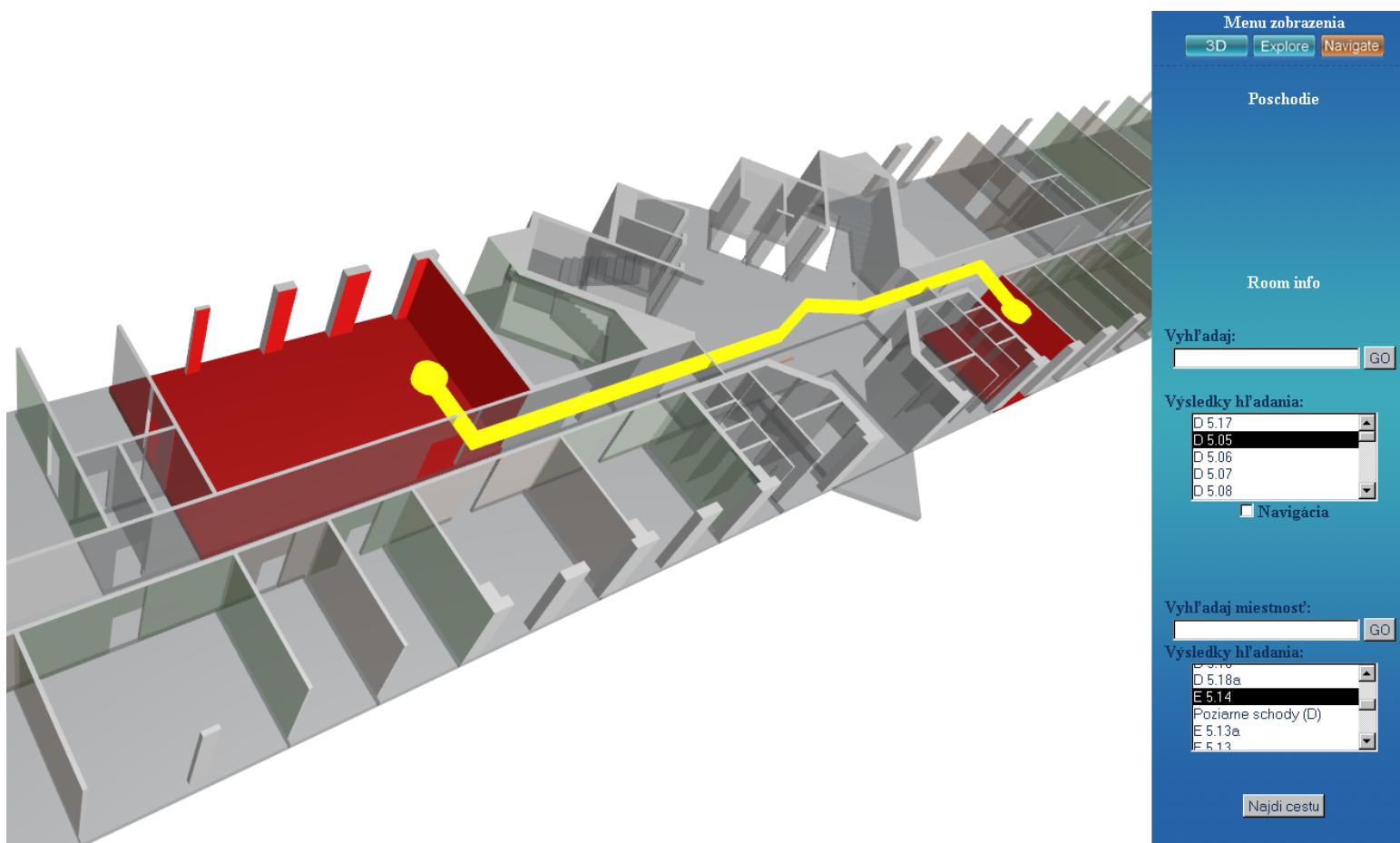
Vyhľadávanie cesty sa robí nad grafom, ktorý zodpovedá topológii miestností vnútri budovy (uzly grafu sú miestnosti a sú spojené hranou, ak sa dá medzi nimi voľne prejsť, napr. dverami. Hrany majú navyše pridelenú váhu – dĺžka cesty v metroch). Tento graf je uložený v samostatnom XML súbore. Aplikácia 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. 45 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`.

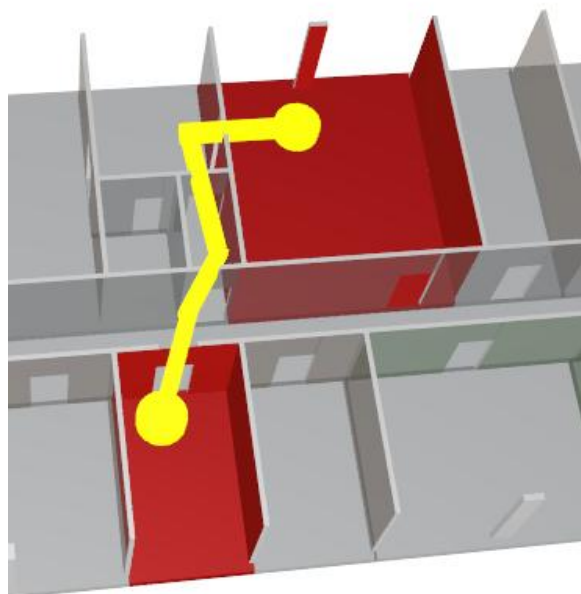
Po nájdení najkratšieho prechodu sa výsledok premietne do okna s O3D modelom. Pre každú hranu výsledného poľa sa vyberie zodpovedajúci komponent hrany v 3D modeli a zobrazí sa. Pri každej novej požiadavke navigácie sa hrany predošlej navigácie odstránia a zobrazia sa nové.

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 4). Takéto „skracovanie“ nie je korektné,





navigácia musí viesť cez verejné priestory. Tento problém bol vyriešený tak, že sa upravila váha koncovej 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. 46 Chyba pri navigácii – skracovanie cez neverejné priestory

#### 5.2.4 Komunikácia s databázou

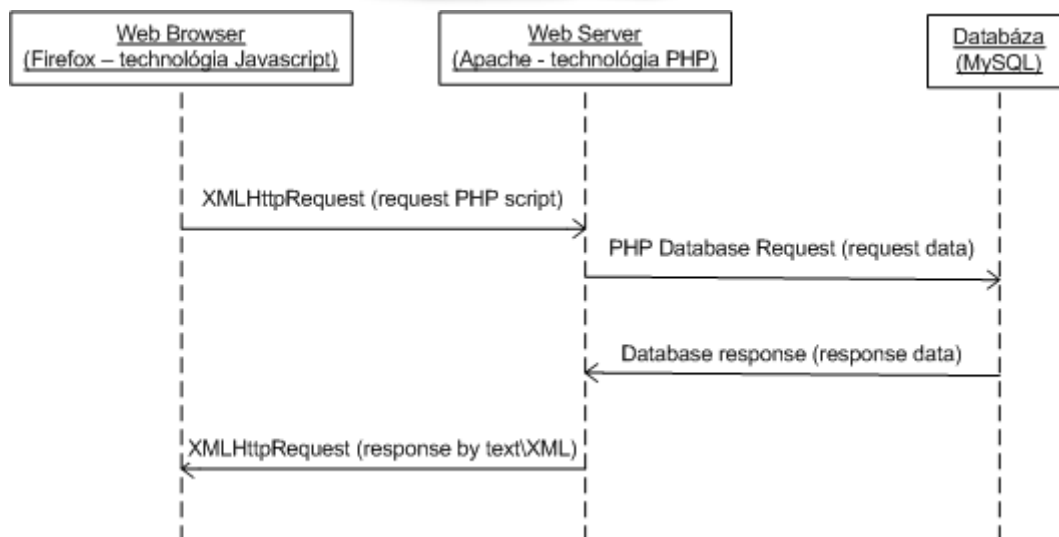
Tretím krokom k posúdeniu realizovateľnosti nášho projektu bolo odskúšanie spojenia prototypu s databázou údajov. Na tento účel sme sa rozhodli pre spojenie nasledovných technológií:

- 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





Obr. 47 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 xmlhttprequest 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.





## 6 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/](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)

