

Modelovanie 3D objektov vo virtuálnom meste

BAKALÁRSKA PRÁCA

Ján Šutara

**UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY**

Študijný odbor 9.2.1 Informatika

Vedúci práce : RNDr. Peter Borovský

BRATISLAVA 2007

Čestné vyhlásenie

Čestne vyhlasujem, že som túto prácu vypracoval samostatne s pomocou literatúry uvedenej v zozname použitej literatúry a podporou môjho vedúceho.

V Bratislave, dňa 15. júna 2007

.....
Ján Šutara

Pod'akovanie

Touto cestou by som chcel pod'akovať vedúcemu práce RNDr. Petrovi Borovskému, ktorý ma usmerňoval a pomáhal mi prekonávať komplikácie, ktoré sa v súvislosti s témou práce objavili.

Z okruhu mojich najbližších patrí moja vďaka najmä mojim rodičom, súrodencom a kamarátom, ktorí ma v mojej snahe podporovali. Špeciálne pod'akovanie patrí mojej sestre Márii Šimekovej za jej ochotu a obetavosť.

Abstrakt

Šutara Ján, Modelovanie 3D objektov vo virtuálnych mestách. Bakalárska práca. Katedra Informatiky. Fakulta Matematiky, Fyziky a Informatiky. Univerzita Komenského v Bratislave.

Vedúci bakalárskej práce : RNDr. Peter Borovský.

Bratislava 2007.

Práca sa zaoberá problematikou modelovania 3D objektov vo virtuálnych mestách. Prináša popis jazyka VRML, ktorý je základným štandardom pre prezentáciu 3D obsahu na Internete. Praktická časť obsahuje 3D modely s uvedením postupov ich vytvorenia. Práca je zameraná na geometrické modelovanie objektov. V konkrétnych modeloch a postupoch prináša prehľad o jednotlivých fázach modelovania. Výsledky modelovania sú 3D modely Rolandovej fontány, kostola Klarisiek, sochy Čumila a Schöne Náciho. Významnou časťou práce sú podrobné postupy modelovania a ich grafické zobrazenie a prezentácia práce prostredníctvom vytvorenej internetovej stránky projektu.

Kľúčové slová : virtuálna realita, VRML, geometrické modelovanie objektov

Abstract

Šutara Ján, Modeling 3D objects in virtual cities. Bachelor's thesis.

Department of Computer Science. Faculty of mathematics, Physics and Informatics,
Comenius University in Bratislava

Consultant : RNDr. Peter Borovský.

Bratislava 2007.

This thesis deals with the problem of modeling 3D objects in virtual city. In recent years computer graphics has made big progress in visualizing 3D models. Theoretical part of project describes language VRML, which is the standard for 3D presentation on the Internet. Applications in this thesis are based on a geometric city model comprising. The three-dimensional representation of 4 objects contain step-by-step procedures. The thesis puts the emphasis on the way of creation of the object application. The image of objects is good recognizing and understanding. For project presentation was the Internet page created.

Key words: virtual reality, VRML, geometric object modeling

Obsah

Obsah	6
Zoznam obrázkov	8
Zoznam príkladov	8
Zoznam tabuliek	9
Zoznam grafov	9
Úvod	10
1 VRML	12
1.1 Základné vlastnosti	14
1.2 Štruktúra súboru	16
1.2.1 Globálne vlastnosti	17
1.3 Navigácia	18
1.3.1 Stanovisko	19
1.3.2 Avatar	20
1.4 Jednoduché svety	23
1.4.1 Základné telesá	23
1.4.2 Transformácie	24
1.5 Povrch telies	26
1.5.1 Farba	26
1.5.2 Textúra	27
1.6 Všeobecné telesá	32
1.6.1 Množina plôch	33
1.6.2 Extrusion	34
1.6.3 Výšková mapa	35
1.6.4 Čiary a body	35
1.7 Svetlo	37
1.7.1 Zdroj rovnobežných lúčov	37
1.7.2 Bodový a smerový zdroj	38
1.8 Zvuk	39
1.9 Pozadie a hmla	41
1.10 Skupinové uzly	42
1.10.1 Group	42
1.10.2 Billboard	42
1.10.3 Anchor	43
1.10.4 Switch	43
1.10.5 LOD	43
2 Modely	45
2.1 Fázy modelovania	47
2.1.1 Popis objektu	47

2.1.2 Nafotenie snímok	47
2.1.3 Modelovanie	48
2.1.4 Textúrovanie	49
2.2 Vyrenderované obrázky modelov	51
2.2.1 Rolandova fontána	51
2.2.2 Klarisky	52
2.2.3 Čumil	53
2.2.4 Schöne Náci	54
Prezentácia práce na Internete	55
3.1 Logo	55
3.2 Popis webovej stránky	56
3.2.1 3D modely	56
3.2.2 Postupy	59
3.3 Anketa	60
3.3.1 Výsledky ankety	61
Záver	66
Zoznam použitej literatúry	67
Prílohy	
A Prehľad súborov na CD	68
B Postupy	69
C Informácie o objektoch	133

Zoznam obrázkov

1.1	Orientácia vo virtuálnom priestore	15
1.2	Geometrický význam parametra avatarSize	21
1.3	Umiestnenie a orientácia základných telies	24
1.4	Vysielanie lúčov bodového zdroja a reflektora	39
1.5	Geometrické šírenie zvuku v priestore	39
2.1	Multimediálna historická Bratislava	46
2.2	Zjednodušený model Rolandovej fontány	51
2.3	Zložitejší model Rolandovej fontány	51
2.4	Klarisky a časť Univerzitetnej knižnice	52
2.5	Klarisky – pohľad na vežu	52
2.6	Klarisky – nádvorie	53
2.7	Čumil bez dopravnej značky	53
2.8	Čumil s dopravnou značkou	54
2.9	Schöne Náci	54
3.1	Logo projektu	55
3.2	Informačná stránka o Rolandovej fontáne	57
3.3	Prezentačná stránka modelu Klarisiek	58
3.4	Ukážka z postu Čumila	59

Zoznam príkladov

1.1	Príklad použitia uzla WorldInfo	18
1.2	Príklad použitia uzla Viewpoint	20
1.3	Príklad použitia uzla NavigationInfo	22
1.4	Príklad jednoduchého umiestnia telesa	25
1.5	Príklad použitia uzla Material	27
1.6	Príklad použitia uzla ImageTexture	28
1.7	Príklad použitia uzla MovieTexture	29
1.8	Príklad použitia uzla PixelTexture a TextureTransform	31

Zoznam tabuliek

1.1	Doporučené logické členenie súboru VRML	16
1.2	Rozmerové parametre základných telies	23
1.3	Základné uzly stromovej hierarchie	25
1.4	Parametre uzla Material	26
1.5	Parametre uzla TextureTransform	31
1.6	Parametre uzla IndexedFaceSet	33
1.7	Parametre uzla Extrusion	34
1.8	Parametre uzla ElevationGrid	35
1.9	Parametre uzlov PointSet a IndexedLineSet	36
1.10	Ďalšie parametre uzla IndexedLineSet	36
1.11	Spoločné parametre svetelných uzlov	37
1.12	Špeciálne parametre uzla SpotLight	38
1.13	Špeciálne parametre uzla Sound	40
1.14	Parametre uzla AudiClip	40
1.15	Parametre uzla Background	41
1.16	Parametre uzla Fog	42
1.17	Parametre uzla Anchor	43
1.18	Parametre uzla LOD	44
2.1	Nafotenie snímok	48
2.2	Modelovanie	49
2.3	Textúrovanie	50
3.1	Počet respondentov v jednotlivých kategóriách	61

Zoznam grafov

3.1	Strávený čas	63
3.2	Celkový dojem	63
3.3	Vizualizácia	63
3.4	Orientácia	63
3.5	Obsahová časť	64
3.6	zjednodušená Rolandova fontána	64
3.7	zložitejšia Rolandova fontána	64
3.8	Klarisky	64
3.9	Čumil bez dopravnej značky	65
3.10	Čumil s dopravnou značkou	65
3.11	Schöne Näci	65
3.12	Postupy	65

Úvod

Virtuálna realita sa rýchlo dostáva z oblasti vedeckej fantasticky na obrazovky počítačov bežných užívateľov. Samotný pojem je spojením dvoch slov opačného významu (*virtual* = neskutočný, fiktívny; *real* = skutočný, pravý). Virtuálnu realitu možno vysvetliť ako prostredie, ktoré umožňuje prácu v trojrozmernom priestore vymodelovanom a spracovávanom počítačom.[01]

Základom virtuálnej reality sú postupy na tvorbu priestorových modelov a scén, manipuláciu s nimi, pohyb v trojrozmernom priestore a zobrazovanie v reálnom čase so snahou o čo najvernejšie zobrazenie. Pritom sa používajú základné postupy z oblasti počítačovej grafiky. Tieto metódy bývajú umocnené obrazovou, zvukovou a hmatovou interakciou.

Jednou z možností aplikácie týchto metód je budovanie virtuálnych miest. Virtuálne mestá majú budúcnosť, môžu vizualizovať konkrétne mestské štruktúry, byť nápomocné aj pri urbanistickom plánovaní, ovplyvňovať architektonickú podobu mesta. Konštrukcia objektov reálneho sveta vo virtuálnom prostredí si využíva rôzne postupy, keďže neexistuje jedna univerzálna metóda. Tejto problematike sa venujem vo svojej práci a to konkrétne geometrickému modelovaniu objektov.

Práca je rozdelená na tri časti. Prvá je teoretická, popisuje **jazyk VRML**, ktorý definuje zápis virtuálnych scén do súborov v textovom tvare. Je základným jazykom pre popis virtuálnych svetov na Internete.

Druhú časť tvorí kapitola **Modely**. Je súčasťou praktickej časti práce. Obsahuje vyrenderované obrázky modelov, ich popis a postupy použité pri modelovaní. Vytvoril som a popísal objekty historickej Bratislavy – Rolandovu fontánu, kostol Klarisiek, sochy Čumila a Schöne Náciho.

Tretiu kapitolu tvorí **Prezentácia práce na internete**. Kľúčové časti sú 3D modely a Postupy. Hlavný dôraz je kladený na podrobné popísanie postupov modelovania a výslednej podoby modelov. Prezentácia modelov na stránke obsahuje základné informácie o objekte, jeho fotografie umiestnenie na mape Bratislavy odkaz na jeho virtuálny model VRML model daného objektu a vyrenderované obrázky modelu, ktoré boli vytvorené v programe Caligari trueSpace 6.5. Návštevníci si môžu pozrieť

VRML model a vstúpiť tak do trojrozmerného priestoru. Súčasťou prezentácie je aj anketa, ktorej výsledky sú uvedené v samostatnej podkapitole tejto časti.

1 VRML

VRML je akronym pre **Virtual Reality Modeling Language**. Tento názov nesie verzia VRML 97, ktorá je medzinárodnou normou ISO (ISO/IEC 14772-1:1997) [01] pre popis statických a dynamických svetov. Definuje spôsob zápisu virtuálnych svetov do súborov v textovom tvare, preto je súčasne aj tzv. formátom, predpisom pre zapisovanie informácii určitého typu.

Použitím VRML môžeme vytvárať vlastné trojrozmerné virtuálne svety na Internete. Umožňuje budovať virtuálne izby, budovy, mestá, hory alebo planéty, ktoré môžeme naplniť hocičím, od virtuálneho nábytku cez autá až po ľudí.

Jeho dôležitou vlastnosťou je, že nevznikol ako produkt jednej firmy, ale je výsledkom spoločného úsilia firiem a odborníkov z celého sveta. Má teda základný predpoklad pre všeobecné prijatie ako univerzálny štandard pre virtuálnu realitu.

Začiatky jazyka VRML nájdeme vo firme Silicon Graphics, Inc. Jej programátori navrhli koncom 80. rokov knižnicu pre prácu s priestorovými objektmi nazvanú *Inventor*. Bola vybudovaná ako nadstavba nad známou a úspešnou základnou grafickou knižnicou GL. Po vzniku novej grafickej knižnice s názvom OpenGL bola vyvinutá aj nová aplikačná knižnica *OpenInventor*. Práve formát, v ktorom sa do súborov zapisovali telesá a scény pre knižnicu *OpenInventor*, sa stal základom jazyka VRML.

V roku 1995 definuje firma Silicon Graphics formát **VRML 1.0**, ktorý je rozšírením formátu *OpenInventor* o možnosti využívania priestorových dát v prostredí World Wide Web. Súčasne so vznikom tohto formátu je založená nezávislá skupina programátorov a návrhárov VAG - VRML Architecture Group, ktorá definuje tri základné požiadavky budúceho vývoja jazyka a to prostriedky pre popis statických svetov, prostriedky pre popis dynamických svetov, prostriedky pre spoluprácu viac užívateľov vo virtuálnom prostredí.

V apríli 1996 je vybraná z ôsmich návrhov spoločná špecifikácia firiem Silicon Graphics a Sony, ktorá má pracovný názov *Moving Worlds*. Táto špecifikácia sa stáva základom jazyka **VRML 2.0**. V tomto tvare sa začne jej upresňovanie, menenie, a dopĺňovanie v otvorenej diskusii prebiehajúcej na Internete (www-vrml@vrml.org).

Z neformálnej skupiny VAG sa stáva oficiálne združenie *VRML Consortium, Inc.*, ktoré začne spolupracovať s medzinárodnou štandardizačnou organizáciou ISO na vzniku medzinárodnej normy. Táto norma nesie pracovný názov DIS 14772-1 a v apríli 1997 získava meno **VRML 97**.

Norma VRML 97 je celý rok 1997 upravovaná do podoby medzinárodne prijateľnej normy ISO. Koncom roku je jazyk VRML oficiálne prijatý za štandard ISO s označením ISO/IEC 14772-1:1997 [01].

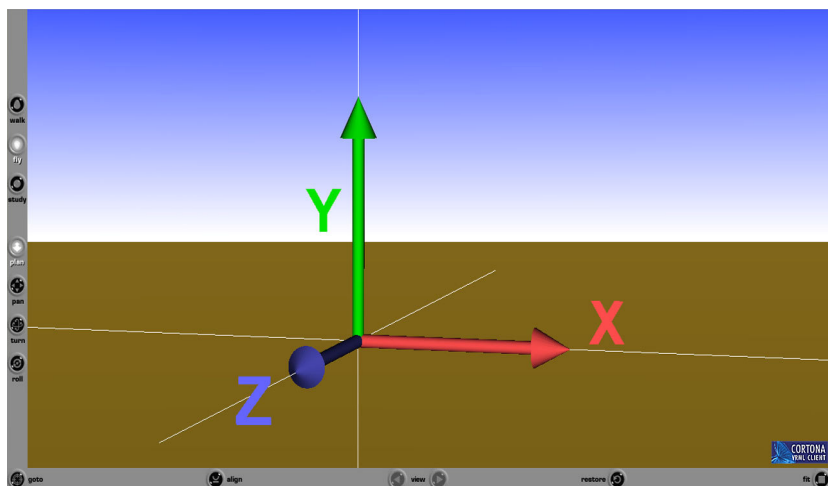
Z historického prehľadu môžeme vidieť, že pod skratkou VRML sa môžeme stretnúť s niekoľkými navzájom odlišnými verziami. Táto nejednotnosť je daňou za rýchlosť, s ktorou bol jazyk vyvíjaný a tiež za otvorenosť pri jeho upresňovaní. Pod názvom **VRML** je teda chápaná medzinárodná verzia VRML 97. Ak sa stretneme s názvom VRML 2.0, je to programátorské označenie VRML 97.

1.1 Základné vlastnosti

Základné vlastnosti jazyka VRML sú [01] :

- Virtuálne svety tvorené priestorovými objektmi sú kombinované s multimediálnymi prvkami ako sú obraz, video, zvuk.
- Pri tvorbe virtuálnych svetov sa dajú využívať prvky zapísané lokálne v súboroch i kdekoľvek na Internete.
- Virtuálne svety môžeme vkladať do WWW-stránok a medzi rôznymi virtuálnymi svetmi môžeme plynule prechádzať podobne ako prechádzame stránkami WWW.
- Prehľadným spôsobom je zaistená animácia, interakcia a manipulácia s virtuálnymi objektmi.
- Na popis statických a dynamických svetov sa používajú rovnaké prostriedky, preto sú statické svety veľmi ľahko rozširiteľné na dynamické a naopak.
- Súčasťou jazyka sú definície spôsobov pohybu užívateľa (chôdza, let, skúmanie objektov), podpora automatickej navigácie v prostredí, popis reakcie na správanie užívateľa.
- Jazyk VRML umožňuje spoluprácu programovacími jazykmi (Java, JavaScript)
- Popis virtuálnych svetov je ukladaný v textovom súbore, ktorý je dobre čitateľný.
- Veľkosť súborov môžeme znížiť kompresiou pomocou programu *gzip* a nemusíme sa ani explicitne starať o spätné dekódovanie.

Systém priestorových súradníc vo VRML používa pravotočivú súradnicovú sústavu pre svetové súradnice, je orientovaný podľa obrázku 1.1. Smer nahor je totožný s kladným smerom osi y . Podlaha alebo zem, po ktorej sa bežne pohybujeme je totožná s rovinou xz . Kladná polos x smeruje iniciálne doprava a kladná polos z smeruje k užívateľovi.



Obrázok 1.1 : Orientácia vo virtuálnom priestore. Prevzaté z [01]

1.2 Štruktúra súboru

V každom súbore v jazyku VRML môžeme nájsť niekoľko logicky odlišných častí, ukazuje ich tabuľka 1.1. Ich uvedené poradie v súbore nie je záväzné, je však dobrým programátorským a autorským zvykom.

#VRML V2.0 utf8	Hlavička súboru VRML.
WorldInfo { ... } Viewpoint { ... }	Úvodné, všeobecné informácie o virtuálnom svete.
Transform { ... } Group { ... } PositionInterpolator { ... }	Popis telies, ich vlastností, definícia prvkov potrebných pre animáciu a interakciu.
ROUTE ... TO ...	Prepojenie dynamických statických prvkov z predchádzajúcej časti.

Tabuľka 1.1 : Doporučené logické členenie súboru VRML. Prevzaté z [01]

V prvom riadku je vždy umiestnená **hlavička súboru**, ktorej tvar je nezmenený. Podľa nej rozpoznajú aplikačné programy typ a verziu súboru. V hlavičke je dokonca zapísaný aj spôsob kódovania tak, aby jeden súbor mohol obsahovať písmená rôznych medzinárodných abecied súčasne. Skratka utf8 znamená, že je použité kódovanie UTF-8 (známe aj pod názvom Unicode), ktoré pre zápis základnej sady znakov využíva osem bitov a pre znaky národných abecied dvojice až šestice bajtov pre jeden znak. Hlavička súboru začína znakom '#', ten s výnimkou prvého riadku označuje vo vnútri súboru komentár. Znamená to, že všetko za ním až do konca riadku je chápané ako poznámka, nie je to súčasťou virtuálneho sveta.

Za hlavičkou najčastejšie nasledujú údaje, ktoré popisujú **celkové vlastnosti virtuálneho sveta**. Patria tu uzly WorldInfo – obsahuje informácie o súbore a jeho autorovi, Viewpoint – zoznam zaujímavých miest vo virtuálnom svete a NavigationInfo – doporučený spôsob prechádzania svetom.

Tretia skupina tvorí **popis virtuálnych telies**, definuje ich tvar, farbu a ďalšie vlastnosti. Je väčšinou najrozsiahlejšou časťou súboru. Do tejto skupiny patrí aj zápis prvkov, pomocou ktorých sa virtuálny svet stáva dynamickým.

V poslednej časti súboru sa zapisuje prepojenie medzi aktívnymi a statickými prvkami. Tejto problematike sa táto práca nevenuje.

Každý z prvkov jazyka má svoje meno (napr. WorldInfo). Tieto hlavné prvky jazyka označujeme pojmom **uzol** (z anglického slova *node*). Môžeme mu priradiť individuálne meno príkazom DEF (Príklad 1.2). Každý uzol obsahuje niekoľko parametrov (pre WorldInfo sú to *title* a *info*). Pre zápis **parametrov** (z angl. slova *field*) platia nasledujúce pravidlá [01]:

- Na poradí parametrov vo vnútri uzla nezáleží.
- Každý parameter má preddefinovanú implicitnú hodnotu danú medzinárodnou špecifikáciou ISO. To umožňuje zapisovať iba tie parametre, ktoré majú hodnoty odlišné od implicitných.
- Počet medzier a riadkov vo vnútri definície uzla nie je dôležitý.
- Názvy parametrov začínajú vždy malými písmenami, názvy uzlov veľkými. Pokračovanie názvu môže potom obsahovať malé i veľké písmená.
- Parameter uzlu na rozdiel od uzlu samotného nemôže byť pomenovaný.

Vzdialenosti sa vo VRML zadávajú v metroch, čas v sekundách a uhly v radiánoch. Farba sa popisuje trojicou hodnôt RGB, každá je v rozsahu 0-1.

Súbory obsahujúce VRML svety majú príponu *wrl* (z anglického slova world). Túto príponu môžu mať aj v prípade, že ich obsah bol skomprimovaný programom *gzip*. Prehliadače sú schopné rozpoznať či ide o textový alebo komprimovaný tvar.

1.2.1 Globálne vlastnosti

WorldInfo je informačný uzol, ktorý sa umiestňuje na začiatok súboru a slúži na dokumentačné účely. Nesie globálnu informáciu o súbore.

Obsahuje dva parametre :

- ❖ *title* – zadávame v ňom jeden textový reťazec, ktorý môže byť prehliadačom zobrazovaný ako meno daného sveta.
- ❖ *info* - obsahuje ľubovoľný počet textových reťazcov, ktorých význam nie je bližšie určený. Textové reťazce zapisujeme do úvodzoviek. Ak v parametri info použijeme viac hodnôt, musíme ich vložiť do hranatých zátvoriek a oddeliť čiarkami.

```
#VRML V2.0 utf8
WorldInfo {
  title "Prázdny svet"
  info ["Autor : Ján Šutara",
       " Dátum : 10.6.2007"
  ]
}
```

Príklad 1.1 : Príklad použitia uzla WorldInfo

1.3 Navigácia

Pri navigácii vo virtuálnom svete musíme zobrať do úvahy, že väčšina počítačov je vybavená iba dvoma vstupnými zariadeniami – klávesnicou a myšou. Z týchto dvoch možností sa viac preferuje myš, pretože sa s ňou dokážeme pohybovať na ploche obrazovky rýchlejšie. Práca s myšou sa dá potom doplniť rýchlym prepínaním režimov pomocou klávesnice.

Práca s myšou by mala viesť uskutočniť tieto činnosti :

- presunúť sa v priestore na ľubovoľné miesto,
- pozrieť sa ľubovoľným smerom,
- vybrať objekt, s ktorým chceme pracovať,
- manipulovať s vybraným objektom – otáčať ho, presúvať, deformovať atď.

Prvé dve požiadavky sa týkajú polohy návštevníka virtuálneho sveta a pohyb myši predstavuje pohyb jeho nôh, resp. celého tela. Druhé dve aktivity súvisia s ručným ovládaním objektu a sú jednoduchšie.

S navigáciou vo virtuálnom prostredí súvisia pojmy *avatar* a *veliteľské stanovisko*. Avatar je pomenovanie virtuálneho dvojníka, ktorý predstavuje nás vo vnútri virtuálneho sveta. Toto pomenovanie pochádza z hinduistickej mytológie a označuje dočasnú telesnú schránku, do ktorej sa Boh vtelňuje pri svojej návšteve Zeme. Avatar má aj určité rozmery, tie mu bránia prechádzať extrémne malými priechodmi.

Veliteľské stanovisko je systém pre ovládanie avatara. V prehliadači je reprezentovaný ovládacím panelom aplikácie.

1.3.1 Stanovisko

Virtuálny svet môže obsahovať zoznam stanovísk (miest v trojrozmernom priestore) spolu so smerom pohľadu a prípadnými inými charakteristikami optiky, cez ktorú do sveta pozeráme. Takéto stanovisko sa nazýva **Viewpoint**.

Pri výbere nového stanoviska nás program automaticky dopraví z predchádzajúceho miesta na nové miesto. Najčastejšie sa to udeje preletom cez virtuálny priestor s postupnou zmenou smeru pohľadu.

Viewpoint má tieto základné parametre :

- ❖ *description* – zapisujeme tu textový reťazec, ktorý je pomenovaním daného stanoviska. Iniciálnou hodnotou je prázdny reťazec.
- ❖ *position* – určuje polohu avatara v ľubovoľných súradniciach. Zadáva sa trojicou celých čísel, kde prvé číslo reprezentuje hodnotu *x-ovej* súradnice, druhé *y-ovej* súradnice a tretie *z-ovej* súradnice. Jeho implicitná hodnota je 0 0 10.
- ❖ *orientation* – natočenie avatara podľa osi a uhla. Os sa zadáva vektorom s hodnotami v rozsahu [-1,1] a uhol je ľubovoľný. Preddefinovaná hodnota je štvorica čísel 0 0 1 0.
- ❖ *fieldOfView* – šírka pohľadu, s ktorým vidíme scénu. Je to hodnota v rozsahu $(0, \pi)$, implicitná hodnota je 45° ($\pi/4$), teda 0,785398. Je dobré si uvedomiť, že väčšia šírka pohľadu nám síce umožní zobrazit' viac telies, ale na okraji obrazovky spôsobí výrazné perspektívne skreslenia. Menšia šírka záberu je podobná rovnobežnému premietaniu, dovoľí nám zväčšiť obraz telies bez priblíženia sa ku nim.
- ❖ *jump* – povoľuje plynulý prechod na stanovisko. Je typu boolean a iniciálna hodnota je TRUE.

```

#VRML V2.0 utf8
Viewpoint {
  position      3 0 0
  orientation   0 1 0 1.57
  fieldOfView  1.57
}
DEF DALEKO Viewpoint {
  description   "Sprava ďalej"
  position      80 0 0
}

```

Príklad 1.2 : Príklad použitia uzla Viewpoint

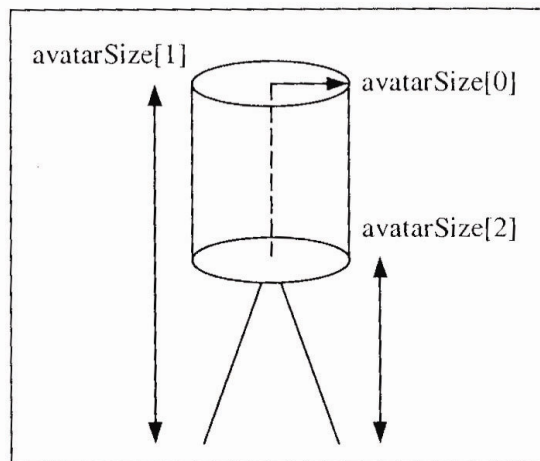
1.3.2 Avatar

V uzle `NavigationInfo` nastavujeme vlastnosti avatara a niektoré prvky veliteľského stanoviska. Priame použitie uzla s inými parametrami nie je časté, pretože preddefinované hodnoty obsahujú vlastnosti podobné človeku. Ak sa v súbore nachádza viac navigačných uzlov, použije sa prvý z nich, lebo iba jeden môže byť aktívny. Hovoríme, že uzol je viazaný na konkrétny stav prehliadača. Prepínanie medzi uzlami je možné a často nastane s prechodom na nové stanovisko.

Obsahuje nasledujúce parametre :

- ❖ *avatarSize* – nastavuje rozmery avatara trojicou kladných čísel. Prvé číslo určuje maximálnu povolenú vzdialenosť, s ktorou sa môže avatar priblížiť k prekážke. Druhé číslo udáva výšku očí nad okolitým terénom meraným v osi *y*. Tretie číslo je maximálna výška prekročenej prekážky. Avatar je schopný malú prekážku prekročiť a na väčšiu vystúpiť. Preddefinované hodnoty sú [0.25, 1.6, 0.75].

Geometrický význam týchto parametrov znázorňuje obrázok 1.2. Na tomto obrázku je avatar reprezentovaný súvislým valcom postaveným na pomyslených nohách. Prvý rozmer v parametre *avatarSize* určuje polomer valca, čiže avatarovu šírku (resp. jej polovicu), druhý umiestnenie jeho hornej podstavy, v jej strede sa nachádzajú oči. Tretí rozmer udáva výškové umiestnenie dolnej podstavy valca, pod ktorou sa nachádzajú avatarove nohy. Chodidlá stoja na zemi, t.j. v rovine $y=0$.



Obrázok 1.2 : Geometrický význam parametra avatarSize. Prevzaté z [01]

- ❖ *headlight* – predstavuje čelné svetlo avatara - baterku. Používa sa v neosvetlených virtuálnych svetoch. Toto svetlo sleduje smer pohybu avatara a osvetľuje bielym svetlom tú časť sveta, ktorú práve pozeráme. Má neobmedzený svetelný dosah. Parameter je typu boolean, iníciaľna hodnota je TRUE.
- ❖ *visibilityLimit* – nezáporné číslo, ktoré určuje dokiaľ avatar vidí. Parameter sa nastavuje z praktických dôvodov na niekoľko desiatok metrov, pretože potom prehliadač nemusí vzdialenejšie objekty vykresľovať.
- ❖ *speed* – nezáporné číslo, určuje rýchlosť avatara v m/s. Je dôležitá pri odhadovaní veľkosti stavieb, ktorú vieme zistiť podľa času stráveného chôdzou okolo virtuálnej stavby.
- ❖ *type* – obsahuje zoznam povolených metód riadenia avatara z veliteľského stanoviska. Avatarovi definujeme spôsoby presunu v priestore. Existujú tri základné pohyby : “WALK”, “FLY”, “EXAMINE”, sú doplnené dvoma ďalšími možnosťami : “ANY”, “NONE”.

Spôsoby presunu [01]

WALK – reprezentuje bežnú chôdzu, pri ktorej sa avatar pohybuje po zemi alebo podložke. Pôsobí na neho príťažlivosť v smere zápornej polosi y, teda dole. Je zapnutá detekcia kolízie s objektmi, takže avatar sa pri pokuse o prechod objektom zastaví.

FLY – vyznačuje sa rovnakým chovaním ako pri chôdzi s tým rozdielom, že príťažlivosť v smere y na avatara nepôsobí.

EXAMINE – avatar je najmenej obmedzovaný bežnými fyzikálnymi zákonmi. Režim je určený predovšetkým pre skúmanie objektov. Je vypnutá príťažlivosť, detekovanie kolízií a avatar môže okolo objektu krúžiť zo všetkých strán. Taktiež je ignorovaná aj rýchlosť pohybu *speed*.

ANY – nešpecifikuje spôsob ovládania, ale doporučuje prehliadaču, že môže povoliť užívateľovi prepínanie medzi predchádzajúcimi základnými spôsobmi. Keďže parameter type má iniciálne hodnoty [“WALK”, “ANY”], pri vstupe do virtuálneho prostredia je užívateľov avatar pripravený k chôdzi, ale je možné prepnúť do ľubovoľného z ďalších dvoch režimov. Ak hodnota ANY nie je zadaná, užívateľ môže voliť len medzi tými spôsobmi, ktoré sú v parametre type vymenované.

NONE – táto hodnota sa v parametre type objavuje samostatne a určuje, že ovládač má skryť všetky ovládacie prvky. Súčasne zamedzí pohyb avatara na pohyb myši.

```
#VRML V2.0 utf8
NavigationInfo {
  avatarSize      [0.30, 2.5, 1.5]
  headlight       FALSE
  visibilityLimit  20
  speed           1.5
  type            ["FLY","ANY"]
}
```

Príklad 1.3 : Príklad použitia uzla NavigationInfo

1.4 Jednoduché svety

Pri modelovaní jednoduchých svetov postupujeme nasledovne :

- Vymodelujeme hrubý tvar objektu (telesá) a určíme jeho veľkosť – *základné telesá a transformácie*.
- Vylepšíme povrch telesa – *farba, textúra*.
- V prípade, že nemôžeme použiť základné telesá, urobíme si vlastné – *plochy, čiary, body*.
- Pridáme efekty – *svetlá, zvuky*.
- Namodelované telesá rozmiestnime do priestoru – *skupinové uzly*.

Dôležitou vlastnosťou jazyka VRML je, že v roli parametrov uzlov môžu byť aj iné uzly. To nám dovoľuje stavať virtuálne svety nielen sekvenčne, ale aj tzv. *hierarchickou stavbou*, ktorá sa nazýva *stromová štruktúra*. Podľa vzájomnej polohy uzlov v strome potom hovoríme o vzťahoch *rodič, potomok, súrodenec*.

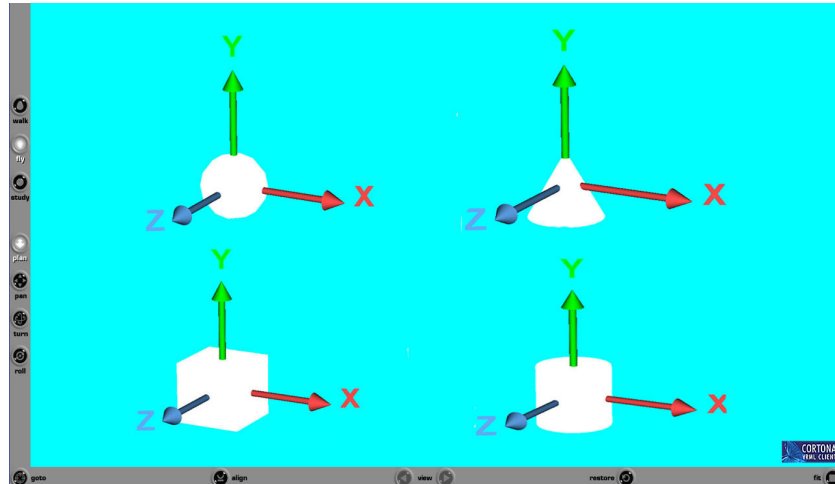
1.4.1 Základné telesá

K dispozícii máme štyri druhy geometrických telies – *guľa, kváder, kužeľ a valec*. Ich zápis, parametre a iníciaľne hodnoty uvádza tabuľka 1.2.

Teleso	Uzol	Parametre	Iniciálna hodnota	Význam parametrov
Guľa	Sphere { ... }	radius	1	polomer
Kváder	Box { ... }	size	2 2 2	dĺžky strán x, y, z
Kužeľ	Cone { ... }	bottomRadius height	1 2	polomer podstavy výška
Valec	Cylinder { ... }	radius height	1 2	polomer podstavy výška

Tabuľka 1.2 : Rozmerové parametre základných telies Prevzaté z [01]

Všetky telesá s výnimkou kužela sú iníciaľne umiestnené tak, aby ich ťažisko bolo v začiatku súradnicovej sústavy. Kužel je umiestnený tak, že v začiatku súradníc leží stred jeho osi. Orientáciu a umiestnenie v priestore ukazuje obrázok 1.3.



Obrázok 1.3 : Umiestnenie a orientácia základných telies

U kužela a valca máme ďalšie parametre typu boolean :

- ❖ *side* – povoľuje vykresľovanie plášťa,
- ❖ *bottom* – povoľuje vykresľovanie dolnej podstavy,
- ❖ *top* – povoľuje vykresľovanie hornej podstavy pre valec.

1.4.2 Transformácie

Na vloženie telies do virtuálneho sveta potrebujeme poznať ešte dva uzly, pomocou ktorých vytvárame stromové štruktúry a spájame jednoduchšie objekty a ich vlastnosti do jedného celku. Sú to uzly Transform a Shape. Tabuľka 1.3 uvádza ich zápis, parametre a iníciaľne hodnoty.

Funkcia uzla	Uzol	Parametre	Iniciálna hodnota	Význam parametrov
umiestnenie	Transform	scale	1 1 1	mierka (kladné hodnoty)
		rotation	0 0 1 0	os a uhol natočenia
		translation	0 0 0	posunutie
		children	prázdny uzol	zoznam potomkov
		scaleOrientation	0 0 1 0	os a uhol natočenia pred zmenou mierky
		center	0 0 0	bod, voči ktorému sa robí otočenie
spojenie vlastností	Shape	geometry	prázdny uzol	tvár povrchu
		appearance	prázdny uzol	vzhľad povrchu

Tabuľka 1.3 : Základné uzly stromovej hierarchie. Prevzaté z [01]

Uzol Transform všetkým svojim potomkom zapísaným do parametra *children*, nastavuje mierku – *scale*, natočenie – *rotation* a posunutie – *translation*. V tomto poradí sú vykonávané aj transformácie v rámci jedného uzla. Netreba pripomínať, že na ich poradí záleží, pretože je rozdiel či teleso najskôr posunieme a v jeho novej polohe ho otočíme okolo stredu súradnicovej sústavy alebo najskôr otočíme a potom posunieme. Ak chceme zmeniť pevné poradie transformácie (napríklad najprv teleso posunúť a potom otočiť) musíme umiestniť viac uzlov Transform do stromu nad seba, čím dosiahneme, že posunutie sa uskutoční v potomkovi a otočenie v rodičovi, pretože transformácie sa skladajú od detí k rodičom.

```
#VRML V2.0 utf8
Transform {
  scale      1 5 2
  translation 3 6 1
  children   Shape { geometry Box { size 4 8 2 }
}
}
```

Príklad 1.4 : Príklad jednoduchého umiestnia telesa

1.5 Povrch telies

Uzol Shape má okrem parametra *geometry* použitého v príklade 1.7.2 ešte druhý parameter *appearance*, ktorý má zásadný vplyv na vzhľad telesa. Do tohto parametra sa môže umiestniť iba jediný uzol s podobným názvom Appearance. Ten dovoľuje definovať dva možné vzhľady povrchu telies – **farbu a textúru**.

1.5.1 Farba

Farbu definuje parameter *material*, ktorý prislúcha uzlu Appearance. Cez tento parameter zadávame farbu, ktorá je rovnaká na celom povrchu. Môžeme do neho umiestniť iba uzol Material, ktorý označuje farebné charakteristiky povrchu. Všetky jeho parametre môžu mať hodnoty od 0 po 1. Ak nastavíme parameter *emissiveColor* na nenulovú hodnotu, je dobré všetky ostatné parametre vynulovať. Zjednodušený popis parametrov ukazuje tabuľka 1.8.1.

Parameter	iniciálna hodnota	význam
<i>diffuseColor</i>	0.8 0.8 0.8	základná farba povrchu v zložkách RGB, matný vzhľad
<i>ambientIntensity</i>	0.2	udáva osvetľovanie farby povrchu celkovým jasom priestoru
<i>specularColor</i>	0 0 0	farba dopadajúceho svetla, ktorú povrch odráža
<i>shininess</i>	0.2	ostrosť odrazu pre <i>specularColor</i>
<i>emissiveColor</i>	0 0 0	vyžarujúca farba povrchu
<i>transparency</i>	0	priehľadnosť; ak sa rovná 1, teleso zmizne

Tabuľka 1.4 : Parametre uzla Material. Prevzaté z [01]

```

#VRML V2.0 utf8
Transform {
  children Shape {
    appearance Appearance {
      material Material {
        diffuseColor 0.5 0.5 1
        emissiveColor 0.5 0.5 1
        transparency 0.3
      }
    }
    geometry Cylinder { height 4 }
  }
}

```

Príklad 1.5 : Príklad použitia uzla Material

1.5.2 Textúra

Najvernejší vzhľad telesa dosiahneme použitím textúry. Textúra je obrazový vzorec, ktorý nanášame na povrch telesa. Tomuto procesu sa hovorí *mapovanie textúry*. Textúru popisujeme parametrami *texture* a *textureTransform*.

Parameter *texture* dovoľuje pokryť povrch telesa ľubovoľným obrázkom, môžeme do neho umiestniť jeden z troch uzlov. Rozhodujeme sa na základe farebného vzoru, ktorý chceme použiť.

Máme tieto možnosti :

- pre obrázok uložený v samostatnom súbore použijeme uzol *ImageTexture*,
- opakujúcu sa kombináciu farieb zapísanú v uzle definuje uzol *PixelTexture*,
- ak chceme prehrávanú videosekvenciu uloženú v samostatnom súbore, použijeme uzol *MovieTexture*.

Každý z týchto uzlov má špecifické parametre, spoločné sú práve dva :

- ❖ *repeatS* – povoľuje opakovanie textúry vo vodorovnom smere,
- ❖ *repeatT* – povoľuje opakovanie textúry v horizontálnom smere.

Oba parametre sú typu boolean, ich iníciaľne hodnoty sú TRUE. Ak ich nastavíme na hodnotu FALSE, povrch telesa mimo textúry sa ofarbí farbou okraja textúry.

ImageTexture

Pre statické textúry, ktoré definuje uzol ImageTexture, môžu byť použité nasledujúce formáty :

- **JPEG** – je vhodný pre kódovanie plnofarebných fotografií, zanedbáva však niektoré detaily a farebné prechody a vôbec neobsahuje informáciu priehľadnosti,
- **GIF** – uchováva obrazy maximálne v 256 farbách a obsahuje jednu úplne prehl'adnú farbu,
- **PNG** – dokáže efektívne a bezstratovo kódovať plnofarebné obrazy vrátane informácii o priehľadnosti a to dokonca v rôznom stupni.

Na zapisovanie údajov o umiestnení a mene obrázku slúži parameter *url*. Tento parameter je skratkou používanou v sieti Internet a označuje adresu súboru či miesta prístupného po sieti (URL – *Uniform Resource Locator*). Jeho dobrou vlastnosťou je, že umožňuje zápis niekoľkých textových reťazcov, ktoré budú využité pre vyhľadanie konkrétneho súboru v prípade, že predchádzajúce adresy nebudú dostupné.

```
#VRML V2.0 utf8
Transform {
  children Shape {
    geometry Sphere {
      appearance Appearance {
        texture ImageTexture {
          url [ "obr.png",
              "http://www.nahrada.sk/obr.png" ]
          repeatS FALSE
          repeatT FALSE
        }
      }
    }
  }
}
```

Príklad 1.6 : Príklad použitia uzla ImageTexture

MovieTexture

Uzol `MovieTexture`, ktorý môžeme chápať ako pohyblivú textúru, používa súbory typu **MPEG-1**. Na zapisovanie údajov o umiestnení a mene obrázku mu slúži taktiež parameter `url`. Na rozdiel od statického obrázka má uzol `MovieTexture` ďalšie parametre týkajúce sa dynamiky prehliadania alebo skôr prehrávania sekvencie pohyblivých obrázkov :

- ❖ `loop` – povoľuje opakovanie prehrávania textúry donekonečna. Bežne je toto prehrávanie vypnuté, má iniciálnu hodnotu `FALSE`.
- ❖ `speed` – nastavuje rýchlosť prehrávania. Ak je hodnota `speed` väčšia ako 1, videosekvencia sa prehráva rýchlejšie ako pôvodná rýchlosť. Hodnota menšia ako 1 znamená spomalenie. Ak hodnotu nastavíme záporne, prehrávať sa bude od konca videosekvencie. Nulová hodnota spôsobí zobrazenie prvého snímku videosekvencie.

```
#VRML V2.0 utf8
Transform {
  children Shape {
    geometry Box { size 3 2 3 }
    appearance Appearance {
      texture MovieTexture {
        url "video.mpg"
        loop TRUE
        speed 1.2
      }
    }
  }
}
```

Príklad 1.7 : Príklad použitia uzla `MovieTexture`

PixelTexture

Textúra PixelTexture je dobrá pre definovanie jednoduchých, opakujúcich sa vzorov na povrchu telesa. Neodkazuje na žiadny vnútorný zdroj obrazových dát, ale definuje vzorec priamym vpísaním farieb do parametra *image*. Ten sa skladá z viacerých hodnôt. Prvé dve čísla znamenajú rozmery, t.j. počet ďalej definovaných farieb pixlov v osiach x a y. Tretie číslo určuje spôsob zápisu jednotlivých farieb a môže nadobúdať hodnoty :

- 0 – žiadna farba, je rezervovaná pre iníciaľne nastavenie parametra (0 0 0),
- 1 – odtiene šedej v rozsahu 0 (čierna) až 255 (biela),
- 2 – odtiene šedej doplnené informáciou o priehľadnosti (rozsah 0-255),
- 3 – farba zložená z troch zložiek RGB (každá v rozsahu 0-255)
- 4 – farba tvorená tromi zložkami RGB doplnená informáciou o priehľadnosti

Pri zápise farieb jednotlivých pixlov sa najčastejšie používa vyjadrenie v hexadecimálnej (šestnástkovej) sústave, kde 0x00 je rovné 0 a 0xFF znamená hodnotu 255. Napríklad zápis 0xFF40FF80 označuje pixel svetlofialovej farby (RGB = [255, 64, 255]), ktorý je na 50 % priehľadný.

Spôsob nanášania textúry je preddefinovaný pre každé základné teleso. Na guli sa textúra omotá okolo rovníka, na póloch dôjde k výraznému skresleniu textúry - okraje sa zlejú do jedného bodu. Na kváder je textúra mapovaná bez skreslenia, na každú stenu sa nanesie jedna kópia zo šiestich. Bez skreslenia je nanášaná textúra aj na valec. Obraz je práve jeden raz namapovaný na celý plášť a ďalej je nanesený na obe podstavy tak, že postavy tvoria kruh vpísaný vo štvorci textúry. Podobne je to aj u kužeľa, u jeho vrcholov však dochádza k skresleniu. U rotačných telies je ľavý okraj textúry nanášaný od okraja so zápornou súradnicou.

textureTransform

Do parametra *textureTransform* môžeme priradiť iba uzol *TextureTransform*. Ten definuje mapovanie textúry na povrch. Popisuje posunutie, otočenie a zmenu mierky textúry. Jeho parametre uvádza tabuľka 1.5.

Parameter	iniciálna hodnota	význam
translation	0 0	posunutie textúry
center	0 0	poloha vzťažného bodu
rotation	0	uhol otočenia okolo vzťažného bodu
scale	1 1	zmena veľkosti vzhľadom ku vzťažnému bodu

Tabuľka 1.5 : Parametre uzla *TextureTransform*. Prevzaté z [01]

Nakoľko hodnoty parametrov sa vzťahujú na definičný rozsah textúry a nie na teleso, všetky akcie s textúrou sú prevádzané opačným postupom. Posun textúry doprava dosiahneme nastavením *translation* na zápornú hodnotu (doľava), otočenie proti smeru hodinových ručičiek (tzv. kladný smer otáčania) spôsobí záporná hodnota parametra *rotation* a zväčšenie textúry dosiahneme zadaním *scale* menšieho ako 1.

```
#VRML V2.0 utf8
Transform {
  children Shape {
    geometry Cylinder { }
    appearance Appearance {
      texture PixelTexture {
        image 3 3 1
        0x00 0xFF 0xFF
        0xFF 0x00 0xFF
        0xFF 0xFF 0x00
      }
      textureTransform TextureTransform {
        scale 3 3
        translation 4 8
      }
    }
  }
}
```

Príklad 1.8 : Príklad použitia uzla *PixelTexture* a *TextureTransform*

1.6 Všeobecné telesá

Medzi doplňujúce uzly popisujúce všeobecnejšie tvary patria :

1. IndexedFaceSet - množina plôch
2. Extrusion – opláštenie
3. ElevationGrid – výšková mapa
4. IndexedLineSet - množina čiar
5. PointSet - množina bodov
6. Text – nápis [01]

Všetky tieto uzly zapisujeme podobne ako základné telesá do parametra *geometry* rodičovského uzla Shape.

1.6.1 Množina plôch

Je najviac používaným uzlom, pretože dovoľuje definovať úplne všeobecné teleso alebo ľubovoľne zakrivenú priestorovú plochu tým, že popíše malé plôšky, ktoré daný objekt pokrývajú. Každá plocha z množiny sa definuje zapísaním indexov (poradových čísel) vrcholov a nie konkrétnym uvedením ich súradníc. Ak je jeden vrchol zdieľaný viacerými plochami, stačí jeho súradnice zapísať len raz a zapamätať si jeho poradie v zozname vrcholov.

parameter	iniciálna hodnota	význam
coord	prázdny uzol	zoznam vrcholov uložených v uzle <code>Coordinate</code>
normal	prázdny uzol	zoznam normál uložených v uzle <code>Normal</code>
color	prázdny uzol	zoznam farieb uložených v uzle <code>Color</code>
texCoord	prázdny uzol	zoznam súradníc bodov pre mapovanie textúry uložených v uzle <code>TextureCoordinate</code>
coordIndex	[]	postupnosti indexov vrcholov jednotlivých plôch (zakončené číslom -1)
normalIndex	[]	postupnosti indexov normál pre jednotlivé vrcholy alebo jednotlivé plochy
colorIndex	[]	postupnosti indexov farieb pre jednotlivé vrcholy alebo plochy
texCoordIndex	[]	postupnosti indexov súradníc textúry pre jednotlivé vrcholy
colorPerVertex	TRUE	farby v parametre <code>colorIndex</code> sa vzťahujú na vrcholy, v opačnom prípade na plochy
normalPervertex	TRUE	normály v parametre <code>normalIndex</code> sa vzťahujú na vrcholy, v opačnom prípade na plochy
convex	TRUE	nápoveda prehliadaču, že všetky plochy sú konvexné
ccw	TRUE	vonkajšie strany sú zadávané postupnosťou vrcholov proti smeru hodinových ručičiek (<i>counterclockwise</i>)
solid	TRUE	plochy sú jednostranné, tvoria povrch telesa zobrazovaný iba z vonkajšej strany, v opačnom prípade sú obojstranné
creaseAngle	0	hodnota uhla, do ktorej sú dve susedné plochy neležiace v jednej rovine považované za oblú

Tabuľka 1.6 : Parametre uzla `IndexedFaceSet`. Prevzaté z [01]

1.6.2 Extrusion

Uzol Extrusion je určený pre popis takých objektov, ktorých povrch vznikne postupným presúvaním dvojrozmerného profilu po priestorovej trajektórii. Profil je tvorený lomenou čiarou, ktorá môže byť pri presune transformovaná. Počet jej bodov sa však nemení. Taktiež trajektória je definovaná jedinou lomenou čiarou.

Objekty, ktoré vzniknú týmto spôsobom nazývame *translačné* alebo *rotačné telesá*. Príkladom takéhoto telesa je váza, ktorej profilom je kružnica (dno) a trajektóriou je zvislá os vázy.

parameter	iniciálna hodnota	význam
crossSection	[1 1, 1 -1, -1 -1, -1 1, 1 1]	postupnosť bodov v rovine určujúca profil, t.j. obrysovú krivku
spine scale	[0 0 0, 0 1 0] [1 1]	postupnosť bodov v rovine určujúca trajektóriu
orientation	[0 0 1 0]	zoznam kladných mierok pre každú ďalšiu polohu obrysu zoznam otočení pre každú ďalšiu polohu obrysu
beginCap	TRUE	povolenie vykresľovania dolnej podstavy (profilu v začiatkovej polohe)
endCap	TRUE	povolenie vykresľovania hornej podstavy (profilu v koncovej polohe)
convex	TRUE	profil je konvexný
ccw	TRUE	profil je zadaný proti smeru hodinových ručičiek (<i>counterclockwise</i>)
solid	TRUE	plochy sú jednostranné
creaseAngle	0	medzný uhol pre určenie hladkého napojenia susedných plôch

Tabuľka 1.7 : Parametre uzla Extrusion. Prevzaté z [01]

1.6.3 Výšková mapa

Posledný špeciálny uzol pre manipuláciu s plochami je uzol `ElevationGrid`. Slúži k vytvoreniu základného tvaru krajiny definovaného pomocou pravouhlej siete vrcholov s príslušnými výškami. Môžeme ho využiť k popisu plasticky zdeformovaného povrchu obdĺžnikovej plochy. Prehľad základných parametrov tohto uzla je uvedený v tabuľke 1.8.

parameter	iniciálna hodnota	význam
<code>xDimension</code>	0	počet vrcholov siete v osi x
<code>zDimension</code>	0	počet vrcholov siete v osi z
<code>xSpacing</code>	1.0	kladná vzdialenosť medzi vzorkami v osi x
<code>zSpacing</code>	1.0	kladná vzdialenosť medzi vzorkami v osi z
<code>height</code>	[]	pole výšok všetkých vrcholov siete
<code>normal</code>	prázdny uzol	zoznam normál uložených v uzle <code>Normal</code>
<code>color</code>	prázdny uzol	zoznam farieb uložených v uzle <code>Color</code>
<code>texCoord</code>	prázdny uzol	zoznam súradníc bodov pre mapovanie textúry uložených v uzle <code>TextureCoordinate</code>
<code>colorPerVertex</code>	TRUE	farby v parametre <code>color</code> sa vzťahujú na vrcholy, v opačnom prípade na plochy
<code>normalPervertex</code>	TRUE	normály v parametre <code>normalIndex</code> sa vzťahujú na vrcholy, v opačnom prípade na plochy
<code>ccw</code>	TRUE	profil je zadaný proti smeru hodinových ručičiek
<code>solid</code>	TRUE	mapa je jednostranná
<code>creaseAngle</code>	0	medzný uhol pre určenie hladkého napojenia susedných plôch

Tabuľka 1.8 : Parametre uzla `ElevationGrid`. Prevzaté z [01]

1.6.4 Čiary a body

Uzly `IndexedLineSet` (množina čiar) a `PointSet` (množina bodov) sú geometrické útvary, ktoré síce nemajú charakter telies, ale dokážu elegantne a úsporne naznačiť tvar alebo vlastnosti virtuálnych objektov. Nasledujúce tabuľky 1.9 a 1.10 ukazujú základné parametre týchto dvoch uzlov.

parameter	iniciálna hodnota	význam
coord	prázdny uzol	zoznam bodov v uzle Coordinate
color	prázdny uzol	zoznam farieb bodov v uzle Color

Tabuľka 1.9 : Parametre uzlov PointSet a IndexedLineSet. Prevzaté z [01]

parameter	iniciálna hodnota	význam
coordIndex	[]	postupnosti indexov vrcholov jednotlivých čiar (zakončené číslom -1)
colorIndex	[]	postupnosti indexov farieb pre jednotlivé vrcholy alebo celé čiary
colorPerVertex	TRUE	farby v parametre colorIndex sa vzťahujú na vrcholy, v opačnom prípade na čiary

Tabuľka 1.10 : Ďalšie parametre uzla IndexedLineSet. Prevzaté z [01]

1.7 Svetlo

V reálnom svete je existencia svetla základnou podmienkou nášho videnia a inak tomu nie je ani vo virtuálnej realite. Vo VRML sú všetky zdroje svetla sú len abstraktnými napodobeninami skutočných svetelných zdrojov a preto nemajú vlastný geometrický tvar. Majú schopnosť prechádzať objektmi, čo spôsobuje, že telesá nevrhajú tieň. K dispozícii máme štyri uzly, ktoré definujú základné druhy svetelných zdrojov :

1. DirectionalLight – zdroj rovnobežných lúčov
2. PointLight – bodový zdroj
3. SpotLight – reflektor, smerový zdroj

Štvrtým je čelné svetlo avatara, ktoré sme si uviedli v kapitole 1.6.2, preto sa mu v tejto kapitole už nebudeme venovať.

Každý uzol má niekoľko parametrov, tabuľka 1.11 uvádza spoločné parametre všetkých.

parameter	iniciálna hodnota	význam
color	1 1 1	farba lúčov vyjadrená v RGB
intensity	1	iniciálna intenzita lúčov v rozsahu [0, 1]
ambientIntensity	0	príspevok zdroja k nepriamemu osvetleniu sveta v rozsahu [0, 1]
on	TRUE	zapnutie alebo vypnutie svetelného zdroja

Tabuľka 1.11 : Spoločné parametre svetelných uzlov. Prevzaté z [01]

1.7.1 Zdroj rovnobežných lúčov

Má jediný parameter *direction*, ktorý obsahuje vektor smeru lúča. Iniciálna hodnota (0 0 -1) definuje lúče v zápornom smere osi z. Lúče nemajú určený začiatok a ich intenzita je konštantná. Osvetľuje iba súrodenecké objekty, teda podstromy VRML, ktoré majú rovnaký rodičovský uzol ako tento zdroj svetla.

1.7.2 Bodový zdroj a smerový zdroj

Nie sú obmedzené iba na súrodenecké objekty ako zdroj rovnobežných lúčov. Majú definované umiestnenie v priestore – parameter *location*. Jeho iniciálna hodnota je 0 0 0. Tieto uzly osvetľujú celý svet, ale iba vo vymedzenom guľovom okolí – dosahu svetla. Definuje ho parameter *radius*, má iniciálnu hodnotu 100. Intenzita lúčov klesá od zdroja až po nulovú hodnotu na okraji guľovej hranice. Útlm sa vyhodnocuje výpočtom, v ktorom sú použité tri koeficienty, zapísané v parametri *attenuation*. Ak tieto koeficienty označíme ako $[a_0, a_1, a_2]$, útlm svetelného lúča vo vzdialenosti d od zdroja svetla vypočítame podľa vzorca

$$\frac{1}{a_0 + a_1 d + a_2 d^2}$$

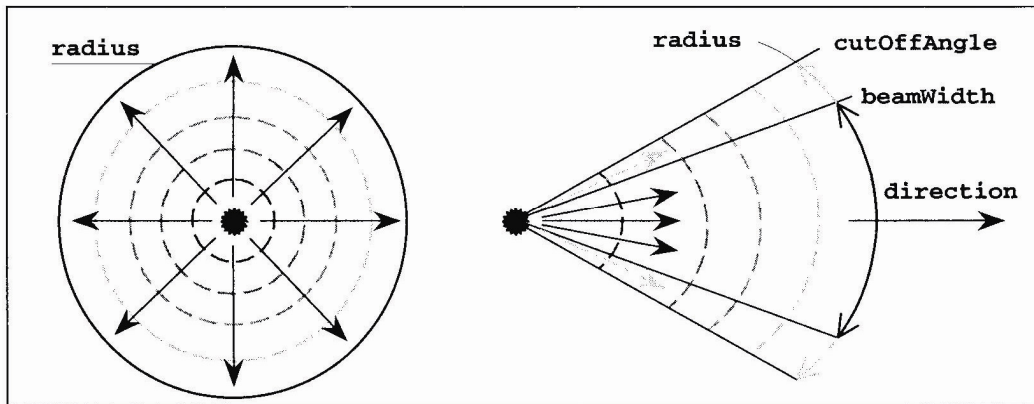
Ak je výsledok rovnice väčší ako jedna, útlm sa nastaví na 1. Trojica koeficientov $[0, 0, 0]$ sa vždy chápe ako $[1, 0, 0]$.

Uzol *SpotLight* obsahuje ešte špeciálne parametre, sú uvedené v tabuľke 1.12.

parameter	iniciálna hodnota	význam
direction	0 0 -1	smer osi svetelného lúča
beamWidth	1.570796	kladný uhol medzi osou a vrcholom vnútorného kužeľa, v ktorom je plný svetelný tok
cutOffAngle	0.785398	kladný uhol medzi osou a vrcholom vnútorného kužeľa, v ktorom klesá svetelný tok

Tabuľka 1.12: Špeciálne parametre uzla *SpotLight*. Prevzaté z [01]

Na obrázku 1.4 je zobrazené vysielanie lúčov u bodového aj smerového zdroja. Smerový zdroj si môžeme predstaviť ako dvojicu súosových kužeľov so spoločným vrcholom (obrázok 1.4 vpravo). Tok lúčov vo vnútornom kuželi (určený parametrom *beamWidth*) je homogénny, intenzita lúčov klesá iba s priamou vzdialenosťou od zdroja. Vo vonkajšom kuželi (parameter *cutOffAngle*) sa intenzita lúčov znižuje aj s rastúcim uhlom od osi kužeľa. Ak je vonkajší kužeľ menší ako vnútorný, reflektor generuje ostro ohraničený zväzok lúčov v priestore menšieho z kužeľov.

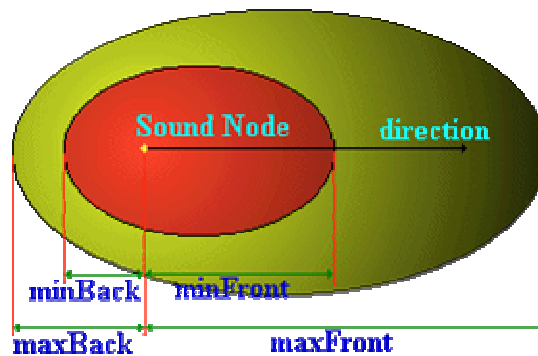


Obrázok 1.4 : Vysielaie lúčov bodového zdroja (vľavo) a reflektora (vpravo). Prevzaté z [01]

1.8 Zvuk

K zvýšeniu dojmu reality prispieva okrem svetla výraznou mierou aj zvuk. Jednotlivé zvuky dokážu presvedčiť návštevníka virtuálneho sveta o tom, že objekty sa správajú rovnako ako v skutočnosti. Pomocou zvuku sa dokážeme aj lepšie orientovať v priestore.

Podobne ako u svetla aj zdroj zvuku má definovaný dosah a útlm. Popis jeho geometrických vlastností ukazuje obrázok 1.5. Obe priestorové elipsy (elipsoidy) majú totožnú hlavnú os a ohnisko. V priestore menšej elipsy sa zvuk šíri bez akéhokoľvek útlmu s rovnakou hlasitosťou. Za hranicou vnútornej elipsy začína slabnúť s lineárne odstupňovanou intenzitou (v dB). Za hranicou vonkajšej elipsy už nepočuť nič.



Obrázok 1.5 : Geometrické šírenie zvuku v priestore. Prevzaté z [04]

Do virtuálneho prostredia dokážeme vložiť niekoľko zdrojov zvuku. Umiestnenie, smer a ďalšie charakteristiky jednoduchého zdroja zvuku sa definujú v uzle Sound. Ten má jediného potomka – uzol, ktorý obsahuje súbor so zaznamenaným zvukom a časový priebeh jeho prehrávania. Parametre zvuku sa nachádzajú v tabuľke 1.13.

parameter	iniciálna hodnota	význam
source	žiadny uzol	uzol so zvukovým záznamom
location	0 0 0	umiestnenie zdroja zvuku
direction	0 0 1	smer hlavnej osi oboch charakteristických priestorových elíps
minBack	1	vzdialenosť zadného vrcholu vnútornej elipsy
minFront	1	vzdialenosť predného vrcholu vnútornej elipsy
maxBack	10	vzdialenosť zadného vrcholu vonkajšej elipsy
maxFront	10	vzdialenosť predného vrcholu vonkajšej elipsy
intensity	1	intenzita zvuku v rozsahu [0, 1]
spatialize	TRUE	povolenie generovania priestorového zvuku podľa orientácie avatara
priority	0	priorita medzi viacerými zdrojmi zvuku v rozsahu [0, 1]

Tabuľka 1.13: Špeciálne parametre uzla Sound. Prevzaté z [01]

Parameter *source* môže obsahovať dva uzly MovieTexture a AudioClip. Prvý uzol je definovaný v kapitole 1.8.2.1. Uzol AudioClip obsahuje zvukové súbory, ktoré môžu byť vo formáte **WAW** alebo **MIDI**. Jeho parametre ukazuje tabuľka 1.14.

parameter	iniciálna hodnota	význam
url	[]	zoznam adries s umiestnením zvukového súboru
startTime	0	čas začiatku prehrávania
stopTime	0	čas ukončenia prehrávania
pitch	1.0	rýchlosť prehrávania (tempo)
loop	FALSE	povolenie prehrávania donekonečna

Tabuľka 1.14 : Parametre uzla AudioClip. Prevzaté z [01]

1.9 Pozadie a hmla

Uzol Background dokáže obraz virtuálneho sveta výrazným spôsobom zlepšiť a dodať mu priestorovú hĺbku. Používa sa na jednoduchú výplň pozadia farbou, škálu farieb meniacich sa nad horizontom alebo na panoramatické obrázky obklopujúce virtuálny svet zo všetkých strán. Je to samostatný prvok, ktorý je do sveta nanesený ako prvý. Vykresľuje sa bez ohľadu na svetelné zdroje, hmlu alebo videnie avatara. Parametre sú uvedené v tabuľke 1.15.

parameter	iniciálna hodnota	význam
skyColor	0 0 0	zoznam farieb pre postupné prechody na sférickej oblohe
skyAngle	[]	rastúca postupnosť uhlov pre farby oblohy
groundColor	[]	zoznam farieb pre postupné prechody na sférickej zemi (podlahe)
groundAngle	[]	rastúca postupnosť uhlov pre farby zeme
frontURL, leftURL, rightURL, backURL, topURL, bottomURL,	[]	zoznamy farieb s umiestnením obrázkov pre jednotlivé steny obklopujúcej kocky

Tabuľka 1.15 : Parametre uzla Background. Prevzaté z [01]

Uzol na popis hmly vo VRML sa nazýva Fog. Má tri parametre – farbu hmly (*color*), spôsob hustnutia hmly (*fogType*) a vzdialenosť od avatara, za ktorou sú všetky vzdialenejšie objekty úplne zakryté hmlou (*visibilityRange*). Jej použitie môže urýchliť výpočty vykresľovania virtuálneho sveta v prehliadači, pretože u vzdialenejších objektov nie je treba vyhodnocovať farby a textúry. Používa sa na jednoduchú výplň pozadia farbou, škálu farieb meniacich sa nad horizontom alebo na panoramatické obrázky obklopujúce virtuálny svet zo všetkých strán. Vykresľuje sa bez ohľadu na svetelné zdroje alebo videnie avatara. Parametre sú uvedené v tabuľke 1.16.

parameter	iniciálna hodnota	význam
skyColor	0 0 0	zoznam farieb pre postupné prechody na sférickej oblohe
skyAngle	[]	rastúca postupnosť uhlov pre farby oblohy
groundColor	[]	zoznam farieb pre postupné prechody na sférickej zemi (podlahe)
groundAngle	[]	rastúca postupnosť uhlov pre farby zeme
frontURL, leftURL, rightURL, backURL, topURL, bottomURL,	[]	zoznamy farieb s umiestnením obrázkov pre jednotlivé steny obklopujúcej kocky

Tabuľka 1.16 : Parametre uzla Fog. Prevzaté z [01]

1.10 Skupinové uzly

1.10.1 Group

Je najjednoduchším skupinovým uzlom. Má rovnaké parametre ako Transform okrem parametrov zameraných na transformáciu. Tento uzol je vlastne akýmsi kontajnerom na uzly. S jeho pomocou usporadúvame uzly v súbore do prehľadnejších skupín.

1.10.2 Billboard

Úlohou tohto uzla je otáčať všetkých potomkov, ktorý sú umiestnený v parametre *children*, tak, aby k avatarovi boli vždy čelom. Samotné otáčanie prebieha dynamicky - spolu s pohybom avatara. Objekty sú štandardne otáčané podľa svojej osi, t.j. osi určenej vektorom (0, 1, 0). Jej presné umiestnenie sa zabezpečí rodičovským uzlom Transform, ktorý môže zmeniť aj orientáciu osi.

Uzol je nenahraditeľný pri zobrazovaní zložitých prírodných objektov, ktorými môžu byť stromy alebo kry. Taktiež môže zobrazovať ľudské podstavy, zvieratá a vtákov. Všetky tieto objekty majú natoľko zložitú geometriu, že jej detailný popis by bol neúnosne rozsiahly a v reálnom čase ťažko spracovateľný.

1.10.3 Anchor

Tento uzol nám umožňuje umiestniť do virtuálneho sveta odkaz. Všetci jeho potomkovia (parameter *children*) sú citlivý na aktivitu kurzoru a stlačenie tlačidla myši. Pri aktivovaní ľubovoľného potomka je užívateľ prenesený do inej časti virtuálneho sveta. Môže to byť prechod na iné stanovisko (ViewPoint) v aktuálne prehliadanom svete, nahradenie aktuálneho sveta iným alebo aktivácia hypertextového odkazu (*http*), najčastejšie s WWW-stránkou. Parametre uzla sa nachádzajú v tabuľke 1.17.

parameter	iniciálna hodnota	význam
url	[]	zoznam adries iných svetov, mien stanovísk a WWW-stránok
parameter	[]	zoznam doplňujúcich parametrov predávaných prehliadaču po aktivácii potomka
children	[]	zoznam potomkov

Tabuľka 1.17: Parametre uzla Anchor. Prevzaté z [01]

1.10.4 Switch

Je skupinovým uzlom, ktorý združuje niekoľko potomkov, ale nedovolí im súčasné zobrazenie. Najčastejšie obsahuje uzly Group a Transform. Má tieto parametre :

- ❖ *whichChoice* – obsahuje celočíselnú hodnotu, ktorá reprezentuje poradie potomka, ktorý má byť zobrazený. Číslo -1 znamená, že všetci potomkovia sú skrytý. Nezáporná hodnota vyberá potomka v poradí, v akom je zapísaný do súboru.
- ❖ *choice* – zoznam potomkov

1.10.5 LOD (*Level Of Detail*)

Uzol sa navonok tvári ako jeden objekt, ale vo vnútri skrýva niekoľko potomkov – reprezentácii toho istého objektu v rôznych stupňoch presnosti. Potomkovia sa zapisujú v parametri *level* postupne od najdokonalejšie vymodelovanej reprezentácie až po najjednoduchšiu. V parametri *range* sú uložené medzné vzdialenosti, v ktorých dochádza k prepnutiu medzi dvoma reprezentáciami. Jeho hodnota by mala byť o jedna menšia ako

počet reprezentácii v zozname. Ak necháme parameter *range* prázdny, prehliadač si bude sám prepínať práve na tú reprezentáciu, ktorú je schopný na konkrétnom počítači vykresliť v dostatočne krátkom čase. Posledný parameter *center* definuje vzťažný bod, voči ktorému sa meria aktuálna vzdialenosť k avatarovi. Prehľad parametrov ukazuje tabuľka 1.14.5.

parameter	iniciálna hodnota	význam
level	[]	zoznam reprezentácii objektu s postupne klesajúcou presnosťou (množstvom detailov)
center	0 0 0	bod, voči ktorému sa meria vzdialenosť objektu od avatara
range	[]	rastúca postupnosť kladných vzdialeností indikujúcich prepnutie

Tabuľka 1.18: Parametre uzla LOD. Prevzaté z [01]

2 Modely

Pri vytváraní 3D modelov vybraných objektov som sa zameril na geometrické modelovanie. Všetky objekty sú buď významné turistické atrakcie alebo známe historické pamiatky mesta Bratislava.

Postupne som vytvoril 3D modely týchto objektov :

- **Rolandova fontána**
- **kostol Klarisky**
- **Čumil**
- **Schöne Náci**

Modely boli použité v projekte **Multimediálna historická Bratislava**, ktorý vznikol s podporou Ministerstva dopravy, pôšt a telekomunikácií (Projekt MDPT 456/131/20005). Jeho vedúcim je doc. Milan Ftáčnik. Cieľom projektu je umožniť digitálne sprítomnenie Starého mesta s využitím najnovších technológií – vrátane interaktívneho modelu virtuálnej 3D Bratislavy. Zameriava sa na podporu globálneho kultúrneho porozumenia a komunikovanie výnimočného komplexu bratislavských hodnôt. Konkrétnou podobou projektu je interaktívna multimediálna prezentácia, ktorá vznikla s použitím dát z Magistrátu, leteckých snímok spoločnosti EUROSENSE, archívu Slovenského rozhlasu a Bratislava Tourist Service v spolupráci MDPT, Starého mesta a UK Bratislava. [7]



Obrázok 2.1 : Ukážka z interaktívnej multimediálnej prezentácie projektu
Multimediálna historická Bratislava. Prevzaté z [7]

Pri modelovaní som zohľadňoval tieto atribúty :

- **Čas** – odhad časovej zložitosti, t.j. koľko času nám zaberie vymodelovať daný objekt
- **Veľkosť zdrojového súboru** – s tým súvisí aj počet vrcholov a stien daného modelu
- **Fotorealizmus** – podobnosť objektu s realitou
- **Presvedčivosť** – pri tomto atribúte objekt neporovnávame s reálnym objektom, ale nezávisle hodnotíme jeho presvedčivosť
- **Rýchlosť renderovania**
- **Prenositeľnosť zdrojového súboru**

2.1 Fázy modelovania

2.1.1 Popis objektu

Pri modelovaní objektov obyčajne vychádzame z nafotených snímok objektu a z nich rozpoznávame geometriu objektu a iné špecifické časti. Často sa však stáva, že fotografie skresľujú objekt. K tomu môže dôjsť v prípade, keď máme zlý prístup k objektu a sme nútený fotiť buď z veľkej diaľky alebo naopak zblízka. Rozpoznanie skutočného tvaru telesa potom zaberá veľa času a je neefektívne.

Týmto problémom sa môžeme vyhnúť spracovaním špecifikácie daného objektu, v ktorej si zapíšeme základné informácie o objekte :

- tvar objektu
- jednotlivé časti objektu, ich tvar, vzájomná poloha a umiestnenie na objekte. Ak sú určité časti objektu podobné, zapíšeme si ich počet.
- špecifické vlastnosti objektu, ktoré mu dodávajú originalitu a na ktoré by sme nemali pri modelovaní zabudnúť

Ďalšou možnosťou je získanie rôznych pomocných dát, to môžu byť letecké snímky objektu alebo katastrálne dáta, ktoré nám môžu poskytnúť pôdorysy daných objektov.

2.1.2 Nafotenie snímok

Pri fotení objektu je dobré, ak si stanovíme účel použitia daných snímok. Kategórie, do ktorých si fotky môžeme rozdeliť, sú :

- fotky obsahujúce informácie o geometrii objektu
- textúry a farebné informácie o objekte

Objekty boli fotené digitálnym fotoaparátom Canon PowerShot SD600. Nasledujúca tabuľka ukazuje porovnanie kvantitatívnych parametrov pri fotení jednotlivých objektov.

objekt	počet fotiek	rozlíšenie	čas fotenia	dátum
Rolandova fontána	72	2112x2816	40 minút	11. 10. 2006
Klarisky	94	2272x1704	48 minút	29. 11. 2006
Čumil	18	2272x1704	10 minút	13. 02. 2007
Schöne Náci	30	2272x1704	15 minút	13. 02. 2007

Tabuľka 2.1 : Nafotenie snímok

Poznámka : rozlíšenie 2112x2816 je 5,95 Megapixlov, rozlíšenie 2272x1704 je 3,87 Megapixlov.

2.1.3 Modelovanie

V tejto fáze modelovania som vytváral geometriu objektov. Modeloval som v programe trueSpace 6.5. Je to 3D grafický a animačný program firmy Caligari Corporation. Jeho výhodou je, že využíva vedľa hlavnej grafickej oblasti cez celú plochu displeja ďalšie nezávislé okná. Navyše môže byť v každom okne nastavený iný grafický režim vrátane plného dynamického zobrazenia v režime Direct3D alebo OpenGL. Ďalšou výhodou programu je, že každý objekt má svoj súradnicový systém. Užívateľ môže meniť jeho začiatok i orientáciu. [09]

Mojou úlohou bolo naučiť sa princípy modelovania v tomto programe a tie potom využiť a zdokonaľovať pri vytváraní požadovaných objektov. S programom som už mal skúsenosti. V roku 2006 som v ňom vymodeloval 3D model kostola v Štiavniku, ktorý bol mojím výstupným projektom predmetu Úvod do počítačovej grafiky.

Pri modelovaní daných objektov som sa snažil nachádzať čo najefektívnejšie postupy. Ich popis pre každý vymodelovaný objekt ich môžete nájsť v prílohe B.

objekt	čas modelovania	počet vrcholov	počet stien
Rolandova fontána - zjednodušená	5 hodín	946	804
Rolandova fontána – zložitejšia	7 hodín	1368	1170
Klarisky	18 hodín	934	649
Čumil	9 hodín	615	514
Čumil s dopravnou značkou	11 hodín	684	572
Schöne Náci	15 hodín	760	730

Tabuľka 2.2 : Modelovanie

Pozn. 1 : Pri zjednodušenej Rolandovej fontáne išlo modelovanie s nízkou detailov, z celkového času 5 hodín trvalo modelovanie fontány 2 hodiny (388 bodov, 233 stien). Modelovanie Rolanda trvalo 3 hodiny (595 bodov a 571 stien).

Pozn. 2 : Pri zložitejšej Rolandovej fontáne som modeloval s vysokou úrovňou detailov. Namiesto modelovania geometrie Rolanda som použil jednoduchý objekt s textúrou

Tabuľka 2.2 prináša porovnanie časov modelovania pre jednotlivé modely spolu s dosiahnutým počtom vrcholov a stien. Na tomto mieste treba spomenúť, že model Klarisiek má medzi objektmi špecifické postavenie nakoľko pri jeho modelovaní išlo o vytvorenie urbárneho modelu.

2.1.4 Textúrovanie

Textúry som pripravoval v programe Adobe Photoshop CS2. Textúrovanie objektov nebolo výskumom mojej práce, bolo však potrebné pre použitie objektov v projekte Multimediálna historická Bratislava.

Tabuľka 2.3 porovnáva pre jednotlivé modely čas prípravy textúr v programe Adobe Photoshop CS2 a čas textúrovania objektu v programe trueSpace 6.5. Posledný stĺpec v tabuľke uvádza percentuálne využitie nafotených snímok na textúrovanie, t.j. koľko fotiek zo všetkých nafotených sa využilo na otextúrovanie objektu. Číslo v zátvorke uvádza ich počet.

Špecifické postavenie medzi všetkými textúrovanými objektmi má model Klarisky. Na jeho otextúrovanie sme použili 133 textúr.

objekt	čas pripravovania textúr	čas textúrovania	využitie fotiek na textúrovanie
Rolandova fontána - zjednodušená	15 min.	15 min.	4,2%(3)
Rolandova fontána – zložitejšia	20 min.	25 min.	7%(5)
Klarisky	16 hod.	12 hod.	46,8% (44)
Čumil	20 min.	25 min.	11%(2)
Čumil s dopravnou značkou	25 min.	30 min.	22%(4)
Schöne Náci	25 min	20 min.	20%(6)

Tabuľka 2.3 : Textúrovanie

Pozn. 1 : Postavy Čumila a Schöne Náciho som netextúroval, objektu som zmenil farbu v materiálových vlastnostiach. Ostatné parametre materiálu som nechal na preddefinovaných hodnotách.

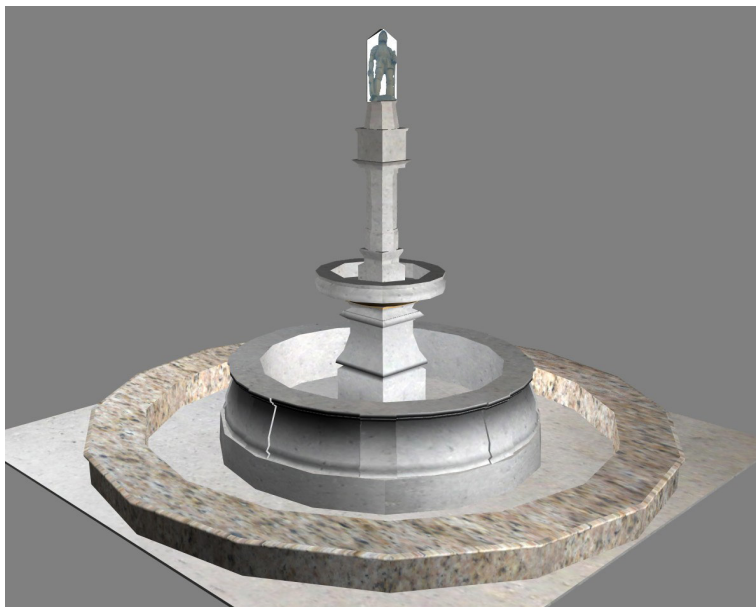
2.2 Vyrenderované obrázky modelov

Nasledujúce obrázky sú vyrenderované boli vyrenderované v programe Caligary trueSpace 6.5.

2.2.1 Rolandova fontána



Obrázok 2.2 : Zjednodušený model Rolandovej fontány



Obrázok 2.3 : Zložitejší model Rolandovej fontány

2.2.2 Klarisky



Obrázok 2.4 : Klarisky a časť Univerzitetnej knižnice



Obrázok 2.5 : Klarisky – pohľad na vežu



Obrázok 2.6 : Klarisky – nádvorie

2.2.3 Čumil



Obrázok 2.7 : Čumil bez dopravnej značky



Obrázok 2.8 : Čumil s dopravnou značkou

2.2.4 Schöne Náci



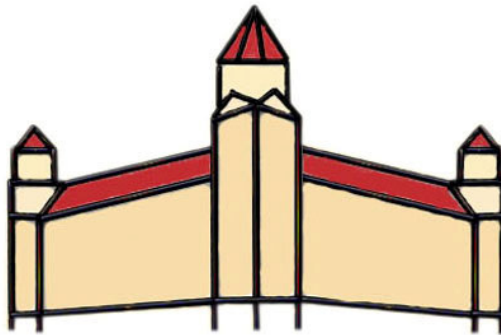
Obrázok 2.9 : Schöne Náci

3 Prezentácia práce na internete

Webová stránka <http://sprite.edi.fmph.uniba.sk/~jansutara/3dbratislava/> bola vytvorená pre študentov Fakulty matematiky, fyziky a informatiky Univerzity Komenenského, konkrétne poslucháčov predmetov *Grafické systémy, vizualizácia a multimédia* a *Úvod do počítačovej grafiky*, ktoré zastrešuje Katedra aplikovanej informatiky. Má napomáhať pri výučbe a tvorbe 3D modelov. Ponúka študentom postupy pri modelovaní konkrétnych objektov. Návštevníci tu ďalej môžu nájsť VRML modely objektov a vstúpiť tak do virtuálneho sveta.

3.1 Logo

Ako logo som zvolil **hrad s tromi vežami** (obrázok 3.1). Je symbolom Bratislavského hradu, dominanty hlavného mesta. Pozorovateľ si môže všimnúť, že štvrtá veža zostáva často skrytá a hrad sa týmto javí ako trojvežový. Aj takto ho poznajú a môžu vidieť obyvatelia Bratislavy.



Obrázok 3.1 : Logo projektu

3.2 Popis webovej stránky

Stránka je členená na tieto podstránky :

- **Domov** – obsahuje základné informácie o projekte, informácie o jeho logu, mapu stránky a odporúčania
- **3D modely**
- **Postupy**
- **3D web** - obsahuje odkazy týkajúce sa problematiky 3D modelovania.
- **Anketa**

Z hľadiska prezentácie projektu sú kľúčové 3D modely a Postupy. Je im venovaná samostatná kapitola.

Stránka je vytvorená pomocou týchto technológií a štandardov : XHTML, CSS, PHP, JavaScript, LightBox v2 a zodpovedá štandardu XHTML 1.1. Jej základom sú php skripty, pomocou ktorých sa generujú stránky.

Pomocou technológie LightBox v2 sa zobrazujú na stránke obrázky. Je to javascriptová technológia určená na prezentáciu obrazového materiálu na Internete. [8]

3.2.1 3D modely

Táto časť obsahuje namodelované objekty. Nachádzajú sa tu informácie o objekte, jeho VRML model, zdrojové súbory scény a vyrenderované obrázky. Zdrojové súbory aj vyrenderované obrázky boli vytvorené v programe Caligari trueSpace 6.5.


Prezentácia modelov na stránke sa delí do dvoch častí. Prvá obsahuje základné informácie o objekte a jeho fotografie. Ďalej tu návštevník nájde umiestnenie daného objektu na mape Bratislavy a odkaz na jeho virtuálny model. Obsah tejto sekcie je uvedený v prílohe C.

Na nasledujúcom obrázku je ukážka informačnej stránky o Rolandovej fontáne.


Rolandova fontána

Uhorský kráľ Maximilián II ju dal vybudovať v roku 1572 ako verejný zdroj vody. Práce sa zhostil kamenársky majster A. Lutringer, ktorý kráľa zobrazil ako obrneného rytiera na vrchole podstavca. Aj preto nesie fontána jeho meno. Je však známa aj pod menom Rolandova, podľa rytiera Rolanda - symbolu spravodlivosti, s ktorým ju začali dávať do súvislosti v 19. storočí. Poznávacie znamenie - „najslávnejšia bratislavská fontána“

zdroj : www.artattack.sk, visit.bratislava.sk



Na nasledujúcom obrázku môžete vidieť, kde sa nachádza Rolandova fontána.



Kredit: Eurosense a Peter Borovský.

- jednoduchý model Rolandovej fontány (veľkosť VRML súboru - 767 KB, textúry - 233 KB)
- zložitejší model Rolandovej fontány (veľkosť VRML súboru - 541 KB, textúry - 233 KB)

Obrázok 3.2 : Informačná stránka o Rolandovej fontáne


Druhá časť stránky slúži na prezentovanie 3D modelov. Nachádza sa tu VRML model daného objektu a vyrenderované obrázky modelu, ktoré boli vytvorené v programe Caligari trueSpace 6.5.

Na to aby si návštevník mohol pozrieť VRML model a vstúpiť tak do trojrozmerného priestoru, má na stránke ponuku nainštalovania si pluginu Cortona VRML client, ktorý patrí medzi najviac používané pluginy na Internete a je zadarmo.


Zaregistrovaní používatelia si môžu v tejto časti stiahnuť aj zdrojové súbory scény vytvorené v programe Caligari trueSpace 6.5.

Klarisky - model

Vyrenderované obrázky v programe Caligari trueSpace 6.5



VRML scéna



- Na prezeranie VRML scén je potrebné mať nainštalovaný plugin. Odporúčam Cortona VRML Client.
- Ako ovládať plugin Cortona VRML Client?

Zdrojové súbory - program Caligari trueSpace 6.5

➔ klarisky.sen

Obrázok 3.3 : Prezentačná stránka modelu Klarisiek

3.2.2 Postupy


V tejto časti sú uvedené postupy modelovania daných objektov. Postupy sú navrhnuté tak, aby bola oddelená technická stránka od filozofie konkrétneho postupu. V texte je to zabezpečené hypertextovými odkazmi, z ktorých každý má pre lepšiu orientáciu v texte aj svoju ikonu. Po každej vykonanej sade inštrukcii je zobrazený obrázok vytvoreného modelu. Ich znenie môžete nájsť v prílohe B.





Na obrázku 3.3 je ukážka z postupu Čumila.



<<<Čumil :: postup 2/6>>>

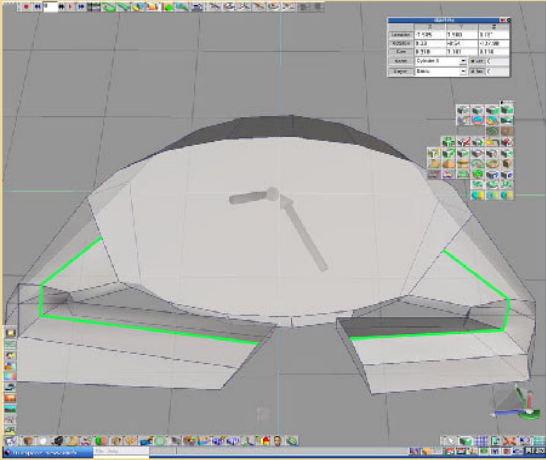
Ramená

☛ Čumilove ramená si môžeme predstaviť ako "predĺženie" valca v bokoch, ktoré v určitej vzdialenosti "zalomíme" a ťaháme späť.

☛ Označíme si vytvorené štvoruholníky na bokoch a **povyťahneme** . Ďalej cez **scale** a **rotate** stien upravujeme jednotlivé časti ramien a následne aplikujeme ďalšie povytiahnutia.

Pri modelovaní kĺbov často potrebujeme z jednej úsečky alebo viac pospájaných bodov spraviť jeden. Postupujeme nasledovne : označíme body, ktoré chceme spojiť a aplikujeme na ne funkciu **weld vertices** . Iné riešenie sa robí pomocou pridávania hrán  a mazania vrcholov . Na časti telesa ktoré chceme orezať vytvoríme rezovú rovinu a body ktoré z nej vytrčajú **vymažeme** .

Po zvládnutí kĺbov na rukách aplikujeme ďalšie povytiahnutia , **scale** a **rotate** extrudovaných stien. Nakoniec ramená ešte upravíme. Do stredu každej hornej steny **pridáme úsečku**  a bodom zvýšime z-ovú súradnicu o 0,050 .



➔ cumil - ramena.scn

<<<^>>>

Obrázok 3.4 : Ukážka z postu Čumila

3.3 Anketa

Súčasťou stránky je aj dotazník. Jeho hlavným cieľom je získať hodnotenie jednotlivých postupov, kvality modelov a celkového dizajnu stránky.

V prvej otázke respondenti ohodnotili úroveň svojich vedomostí a skúseností z problematiky počítačovej grafiky. Na základe odpovedí som ich zaradil do týchto kategórii :

- **Laik**
- **Začiatočník**
- **Mierne pokročilý**
- **Pokročilý**
- **Expert**

Ďalej som sa pýtal, či je respondent študent a aký študijný odbor študuje. Ostatní vyplnili svoje pracovné zameranie.

Nasledovali otázky o stránke, v ktorých sa hodnotili tieto parametre :

- celkový dojem zo stránky
- vizualizácia stránky
- orientácia na stránke
- obsahová časť

Tieto otázky boli doplnené otázkami o tom, či návštevníkov stránka obohatila a čo od nej očakávali.

Ďalej mohli respondenti vyjadriť svoj názor postupy, ohodnotiť kvalitu modelov a na záver som sa pýtal koľko času strávili na stránke.

3.3.1 Výsledky ankety

Ankety sa zúčastnilo spolu 65 respondentov, nasledujúca tabuľka uvádza počet uchádzačov v jednotlivých kategóriách. Vyhodnotenie uvádzam pre každú skupinu zvlášť.

Kategória	Počet respondentov
Laik	24
Začiatočník	25
Mierne pokročilý	13
Pokročilý	3
Expert	0

Tabuľka 3.1 : Počet respondentov v kategóriách

Laik

V tejto kategórii sa zúčastnili prevažne študenti iných vedných odborov, ktorí sa s problematikou 3D modelovania predtým nezaoberali. Tomu zodpovedal aj čas strávený na stránke v rozmedzí 10–30 minút. Celkový dojem z prezentácie bol hodnotený ako dobrý a zároveň kvalita modelovaných objektov ako výborná. Ako najlepšie boli hodnotené modely Klarisky a zložitejšia Rolandova fontána. Postupy boli hodnotené ako dobré a zrozumiteľné.

Prostredníctvom tejto stránky sa väčšina respondentov stretla s danou problematikou prvý krát. Vyjadrovali sa prevažne k obsahovej časti, napr. štylistike, histórii daných objektov ako aj ku logu, ktoré sa im páčilo.

Začiatočník

Anketu vyplnili prevažne študenti v odbore informatika. Čas strávený na stránke bol v rozmedzí 10-30 minút. Respondenti uviedli ako prínosy danej prezentácie, že získali prehľad o nástrojoch modelovania, hodnotili prehľadnosť, zdrojové kódy, dizajn, štýlovanie. Celkový dojem z prezentácie bol hodnotený ako dobrý, vizualizácia ako dobrá a použiteľná. S orientáciou na stránke nemali respondenti žiadne problémy. Kvalita

modelovaných objektov bola hodnotená prevažne ako dobrá. Postupy modelovania boli hodnotené ako dobré.

Respondenti v kategórii začiatočník uviedli aj zlepšovacie návrhy týkajúce sa samotného prevedenia stránky.

Mierne pokročilí

V tejto kategórii zúčastnili študenti v odboroch matematika, informatika – animácia. Strávili na stránke 10-30 minút. Celkový dojem z prezentácie bol hodnotený ako dobrý a vizualizácia stránky ako dobrá. Modelované objekty boli hodnotené ako dobré a model Klarisky prevažne ako výborný.

Medzi odporúčania respondentov patrilo použiť viac obrázkov ilustrujúcich uvedené Postupy. Tie boli zhodnotené ako prepracované zrozumiteľné. Návrhy na zlepšenie sa preto týkali tejto časti a to rozšírenia problematiky do hĺbky.

Pokročilí

Tu odpovedali študenti informatiky a new média dizajnu. Celkový dojem z prezentácie bol dobrý a vizualizácia bola hodnotená ako dobrá. Kvalita modelovaných objektov bola prevažne hodnotená ako dobrá.

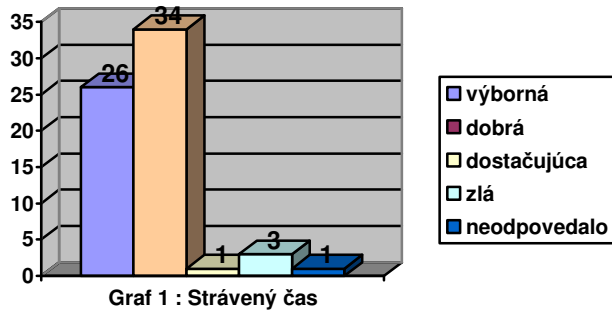
Na stránke strávili 10-30 minút. Návrhy na zlepšenie sa týkali konkretizácie o aký druh 3D grafiky sa jedná a sprehľadnenia Postupov.

Expert

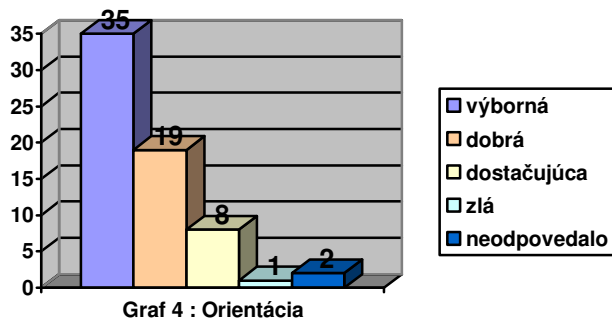
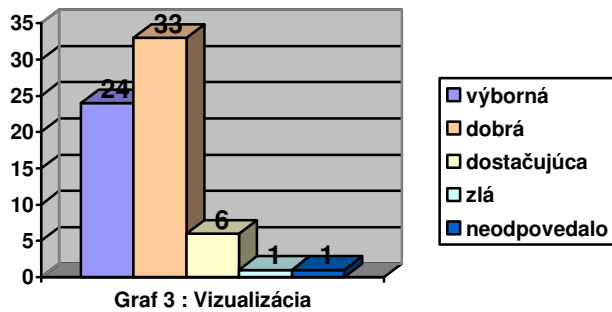
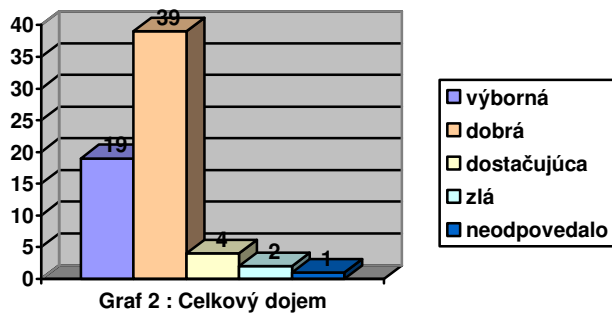
Do tejto kategórie sa nezaradil žiadny respondent.

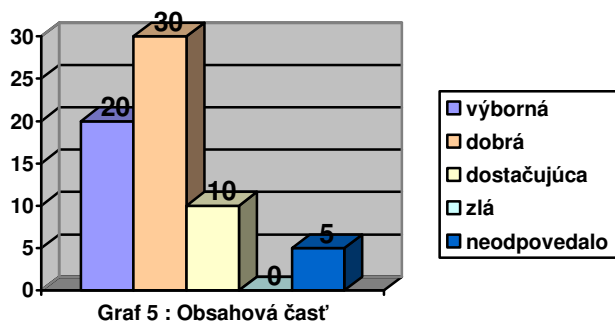
Celkové vyhodnotenie ankety

Strávený čas



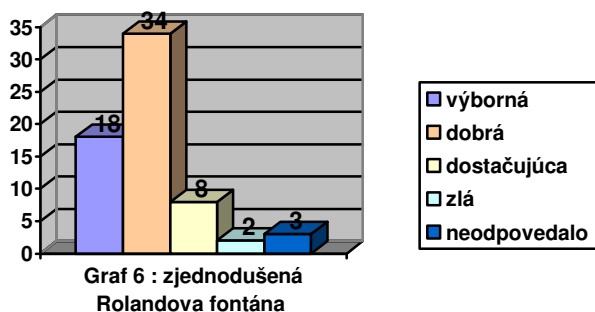
Hodnotenie stránky



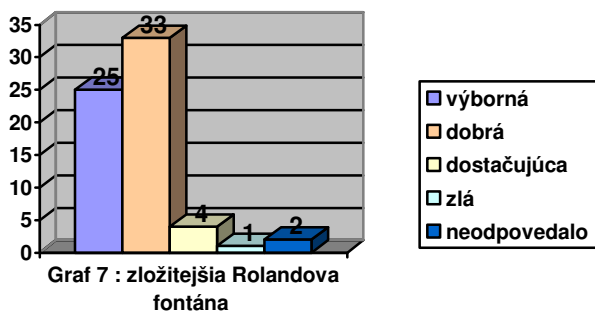


Graf 5 : Obsahová časť

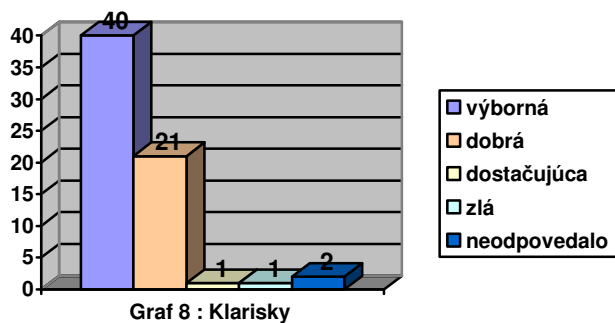
Kvalita modelov



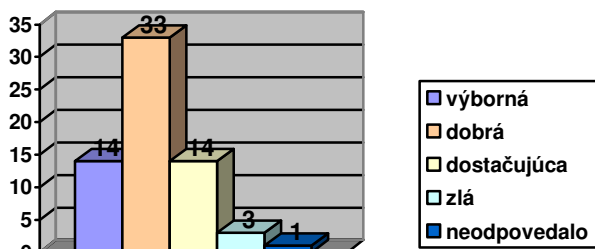
Graf 6 : zjednodušená
Rolandova fontána



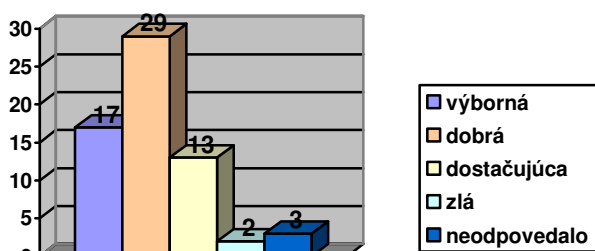
Graf 7 : zložitejšia Rolandova
fontána



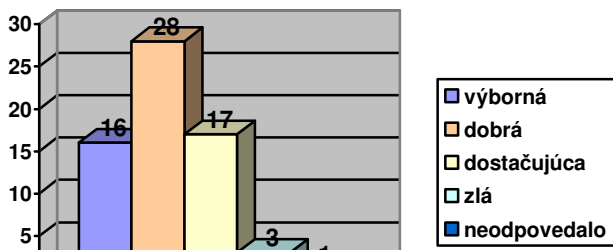
Graf 8 : Klarisky



Graf 9 : Čumil bez dopravnej značky

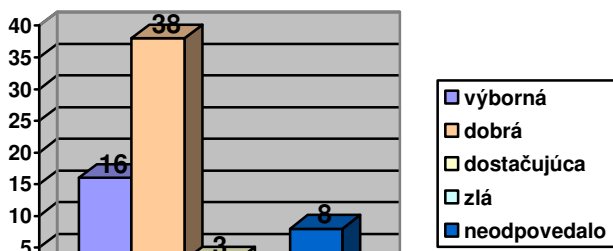


Graf 10 : Čumil s dopravnou značkou



Graf 11 : Schöne Näci

Postupy



Graf 12 : Postupy

Záver

Predkladaná práca sa zaoberá vytvorením 3D modelov objektov virtuálneho mesta. Ako ciele modelovania boli vybrané 4 skutočné objekty v Bratislave. Postupne som vytvoril 3D model Rolandovej fontány, kostola Klarisiek a sochy Čumila a Schöne Náciho.

Hlavným cieľom práce bolo na základe znalostí z 3D modelovania programovo spracovať vstupy a vytvoriť 3D modely vybraných objektov. K tomuto viedli čiastkové ciele, ktorými boli vypracovanie podrobnej dokumentácie postupov modelovania a vytvorenie internetovej prezentácie projektu.

Práca je rozdelená na teoretickú a praktickú časť. Teoretická obsahuje oboznámenie sa s problematikou, krátku históriu a popis jazyka VRML a jeho charakteristík. Praktickú časť tvoria vyrenderované obrázky 3D modelov, postupy vytvorenia virtuálnych 3D modelov a prezentácia projektu na internete.

Pri všetkých modeloch som zameral na geometriu. Na ich vytvorenie som použil program. Caligari trueSpace 6.5. Postupy modelovania obsahujú postupnú záväznosť jednotlivých fáz modelovania konkrétneho objektu a môžu slúžiť ako pomôcka pre užívateľov, ktorí sa venujú tejto problematike. Na príklade sa môžu zoznámiť s použitým programom a vytvoriť si v ňom vlastný 3D model.

Prezentácia projektu na vytvorenej internetovej stránke bola rozdelená do niekoľkých častí. Hlavné časti sú vytvorené modely a postupy modelovania. Každý model má vytvorené svoje podstránky, na ktorých uvádzam konkrétne výsledky modelovania daného objektu, vyrenderované obrázky. Postupy tvorí text a aktívne odkazy. Tieto názorne ilustrované jednotlivé postupy grafickým znázornením modelovanej situácie. Týmto umožňujú prehľadné sledovanie jednotlivých krokov.

Vlastný prínos práce je vo vytvorení 3D modelov objektov. Virtuálne 3D modely zachytávajú podobu svojich predlôh, je možné hodnotiť ich kvalitu a vernosť zobrazenia. S vytvorenými modelmi je možné ďalej pracovať. Všetky modely sa stali súčasťou projektu Multimediálna historická Bratislava.

Myslím si, že ciele určené na začiatku práce sa mi podarilo splniť. Vytvorením modelov som získal ďalšie skúsenosti s 3D modelovaním. Spracovaním postupov modelovania, som vytvoril postupnosť krokov, ktoré sa môžu využiť pri modelovaní podobných objektov.

Zoznam použitej literatúry

- [01] Jiří Žára : *VRML 97 Laskavý průvodce virtuálními světy*, Computer Press, 1999.
ISBN 80-7226-143
- [02] Jakub Zrzavý : *VRML tvorba dokonalých www stránek. Podrobný průvodce*.
Grada Publishing, spol. s.r.o., 1999. ISBN 80-7169-643-9
- [03] Šárka Gergelistová : *VRML v příkladech*, Nakladatelství BEN, Praha 2004.
ISBN 80-7300-138-1
- [04] Floppy's VRML97 Tutorial, <<http://web3d.vapourtech.com/tutorials/vrml97/>>,
15. 6. 2007
- [05] Petr Lindner, Miroslav Myška, Tomáš Tůma : *Velká kniha digitální fotografie*,
Computer Press, a.s., Brno 2003.
ISBN 80-251-0013-8
- [06] Jazyk VRML (ver. 2.0),
<http://hornad.fei.tuke.sk/predmety/pg/vrml/vrml_main.htm>, máj 2007
- [07] *Multimediálna historická Bratislava, PROMO CD*, vedúci projektu Milan
Ftáčnik, 15.6. 2007
- [08] Lightbox (ver. 2),
<<http://www.huddletogether.com/projects/lightbox2/>>, máj 2007
- [09] Caligary Corporation,
<<http://www.caligari.com/>>, máj 2007

Príloha A

Prehľad súborov na CD

Priložené CD obsahuje túto bakalársku prácu aj prílohami Postupy a Informácie o objektoch v elektronickej podobe vo formáte PDF. Nájdete ju v hlavnom adresári pod názvom *sutara.pdf*.

Adresár *zdroje* obsahujú offline súbory hypertextových odkazov na internete.

V adresári *fotky* nájdete fotografie vymodelovaných objektov rozdelené podľa ich názvov do jednotlivých podadresárov. Fotografie sú vo formáte JPEG. V podadresári *Klarisky – nádvorie* adresára *Klarisky* sú fotografie, ktorých autorom je RNDr. Peter Borovský. Sú vo formáte JPEG a NEF.

V adresári *render* nájdete vyrenderované fotografie vymodelovaných objektov rozdelené podľa ich názvov do jednotlivých podadresárov. Fotografie sú vo formáte JPEG.

V adresári *3d modely* sú podobne ako predchádzajúcom uložené zdrojové súbory namodelovaných scén podľa názvov jednotlivých objektov. Súbory sú formáte programu Caligary trueSpace 6.5. Ďalej tu nájdete aj zdrojové súbory objektov v jazyku VRML.

Príloha B


Postupy

B.1 Rolandova fontána




Zjednodušená fontána

Budeme ju modelovať s **nízkou úrovňou detailov**. Ide nám o zachytenie základných častí fontány bez detailov. Podobné požiadavky kladieme aj na model Rolanda, ktorý **umiestnime** na fontánu.











Podlaha




Do pracovnej plochy vložíme **kocku**  a **sploštíme** ju. Bude umiestnená na spodok fontány ako podlaha.

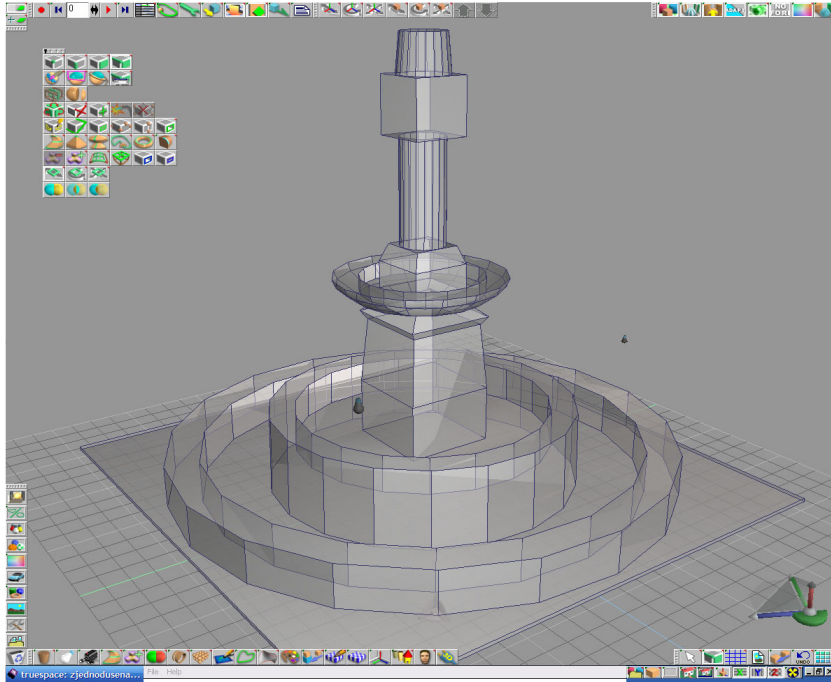
Vonkajší kruh

Vložíme **valec**  s počtom bočných strán 19 a **umiestnime** na podlahu. Ďalej vložíme do jeho stredu ďalší **valec** , upravíme šírku na 4/5 pôvodného a **odpočítame**  ho, čím urobíme do pôvodného valca otvor.

Vnútorň kruh

Vložíme ďalší **valec** , s počtom bočných strán 19, šírka bude 3/5 vonkajšieho, výška rovnaká a umiestnený bude v jeho strede. Do jeho hornej steny **vyrežeme**  menšiu stenu a **vyextrudujeme**  z nej vonkajší kruh. Do vnútorného kruhu vyrežeme kocku, použijeme na to funkciu **polygon bevel**  a **erase vertices**  a **vyextrudujeme**  ju 4x. Pri druhom extrudovaní vrchnú stenu **zúžime**. Pri treťom **rozšírime** a pri poslednom znova **zúžime**. Navrch **umiestnime guľu** , ktorú zvrchu v 3/4 **zrežeme** . **Vyrežeme**  do nej valcový otvor a z jeho stredu podobným spôsobom ako v predchádzajúcom **vyextrudujeme**  na 2x kocku. Do pracovnej plochy vložíme **valec**

 a **umiestnime** ho navrch. Podobným spôsobom **umiestnime** navrch **kocku**  a následne ďalší **valec** , ktorý zvrchu **zúžime**. Nakoniec všetky namodelované časti **spojíme**.





Obrázok B.1-1 : Model zjednodušenej fontány



Roland




Pri modelovaní som sa snažil zachytiť základné črty Rolanda (trup, ruky, nohy , meč, pancier, hlava) s **nižšou úrovňou detailov**.

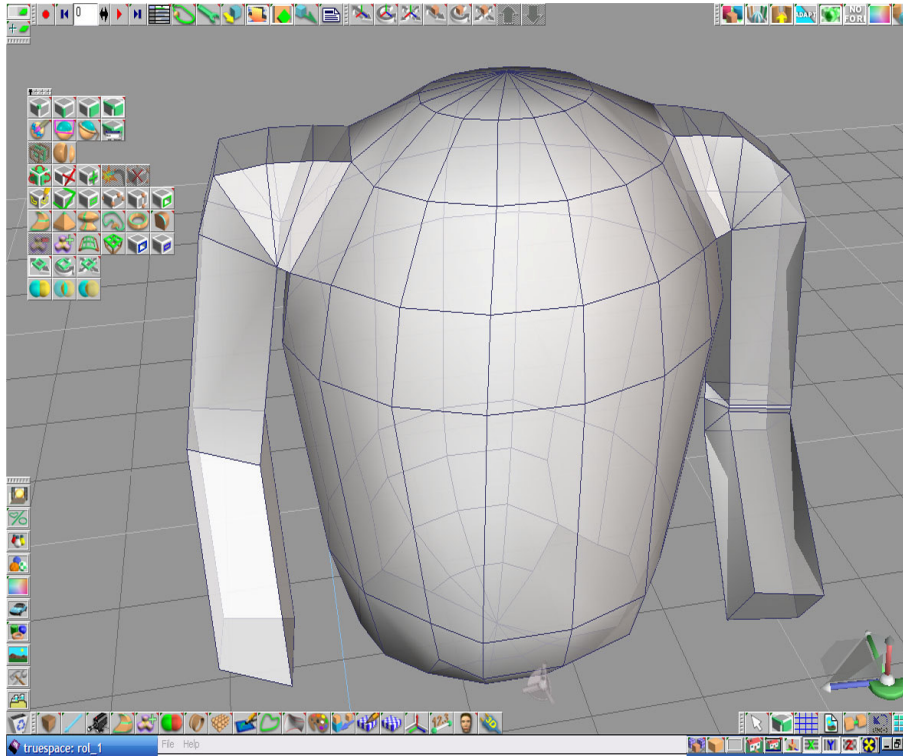
Trup a brucho

Do pracovnej plochy vložíme **rounded cylinder**  so 16 bočnými stenami v každej línii. Horizontálne v smere spredu dozadu ho **sploštíme** a dolnú časť **zúžime**. V tretej línii stien zhora si vyberieme bočnú stenu zľava a sprava, **rozšírime** ich a **vyextrudujeme**  z nich **ramená**.

Ramená



Pravé rameno **extrudujeme**  6x, ľavé 8x. V miestach, kde chceme urobiť klby, použijeme funkciu **weld vertices** , ktorá úsečku alebo viac pospájaných bodov zlúči

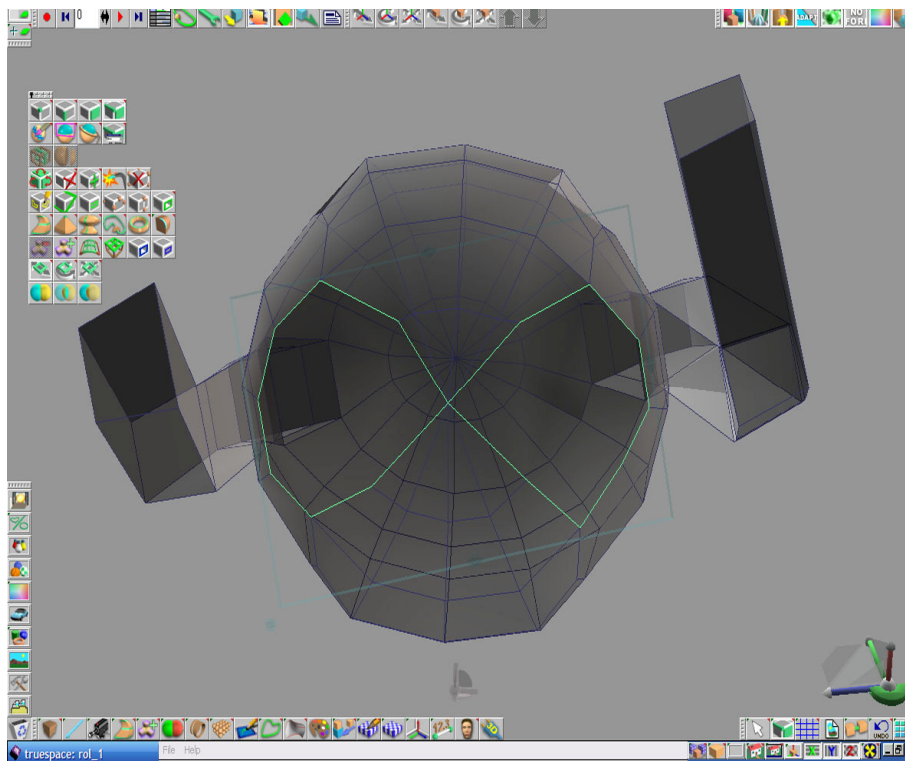
do jedného bodu a umiestni ho do stredu pôvodných bodov. Iné riešenie sa robí pomocou pridávania hrán  a mazania vrcholov . Na časti telesa, ktoré chceme orezať vytvoríme rezovú rovinu a body ktoré z nej vytrčajú **vymažeme** .



Obrázok B.1-2 : Trup a ramená Rolanda







Nohy

V prvých dvoch líniih zdola **vymažeme**  hrany po bokoch tak, aby spredu aj zozadu zostali v každej línii 4 menšie steny (viď obrázok B.1-3) a vzniknuté steny dvakrát **extrudujeme** .





Obrázok B.1-3 : Steny, z ktorých budeme extrudovať nohy Rolanda


Hlava

Na pôvodnom valci v prvej línii zhora **vymažeme**  vrchný bod v strede. Stenu, ktorá týmto vznikne **rozšírime** a **vyextrudujeme** . Do scény vložíme ďalší **rounded cylinder** , **zrotujeme** ho o 90 stupňov okolo *x-ovej* osi a **umiestnime** navrch trupu. Kliknutím RMB (práve tlačidlo myši) na objekt sa dostaneme do editovacieho módu, v ktorom môžeme robiť na Rolandovej hlave urobiť napríklad prilbu, fúzy alebo vlasy. Princíp modelovania týchto častí je rovnaký : na steny objektu pridávame hrany  alebo body , podľa potreby zmeníme ich polohu a niektoré steny **vyextrudujeme** .


Chodidlá

Spodné steny nôh **vyextrudujeme** , ich predné hrany **posunieme** dopredu, vrátime do rovnakej výšky ako pôvodná stena a ešte raz **vyextrudujeme** .

Meč

Vložíme **kocku** , upravíme ju na tvar dlhej tenkej tyče, **zrotujeme** a **umiestnime** Rolandovi do ľavej ruky (viď obrázok B.1-4).

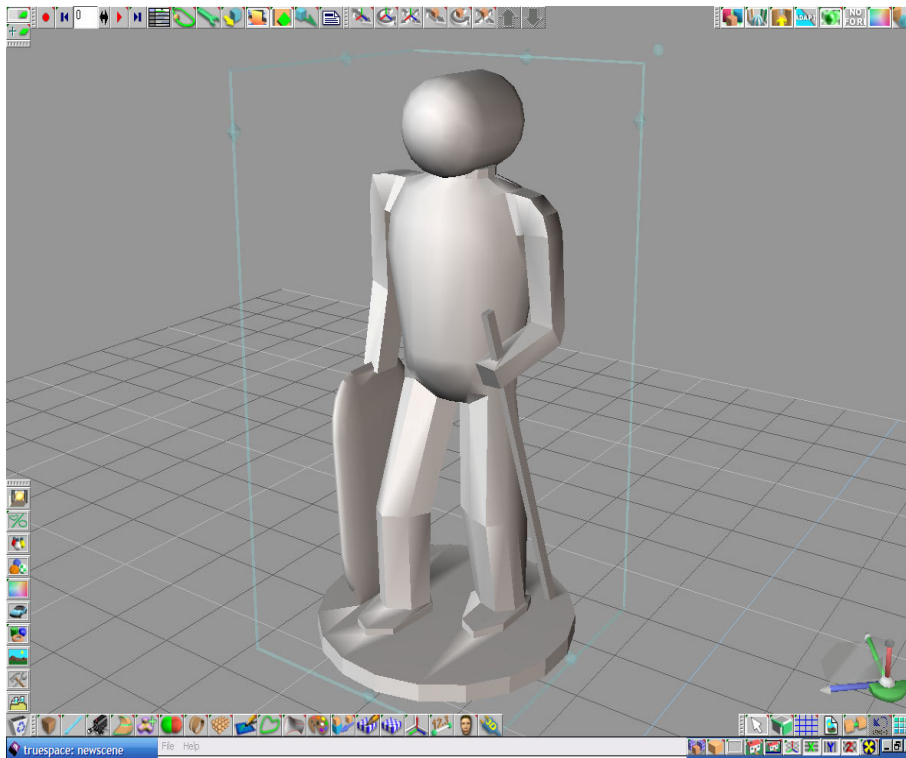
Erb

Vložíme **rounded cube** , **zrotujeme** okolo y-ovej osi o 90 stupňov, **sploštíme** (viď obrázok B.1-4) a **umiestnime** Rolandovi do pravej ruky.

Podstavec








Vložíme **valec** , **sploštíme** a **umiestnime** pod Rolanda.

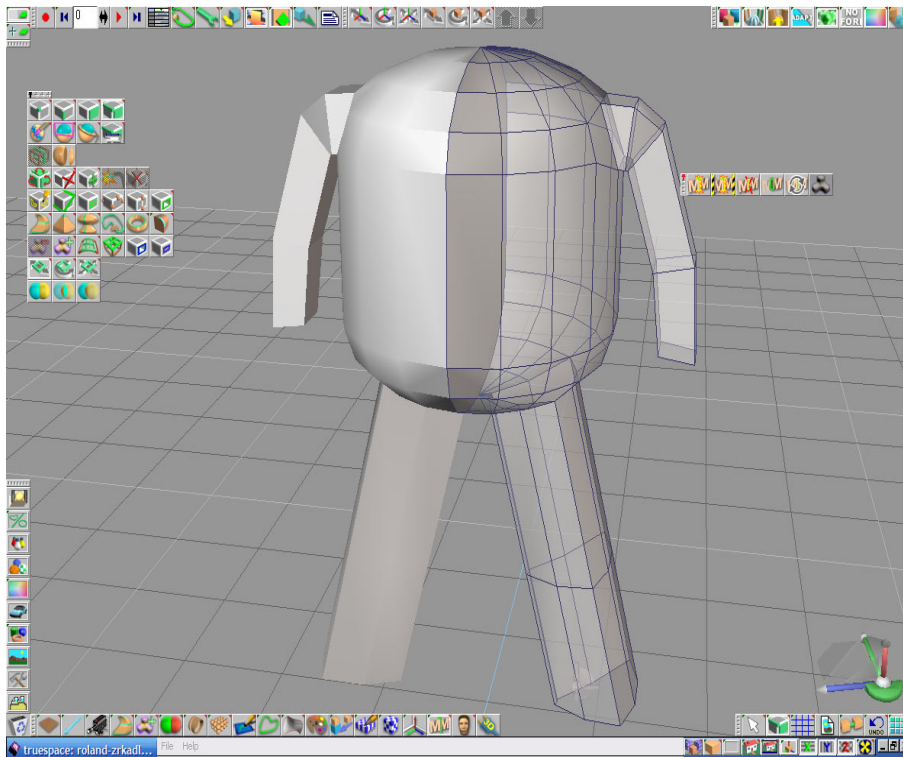
Na záver postupu všetky vymodelované časti **spojíme**.



Obrázok B.1-4 : Modelovaný Roland

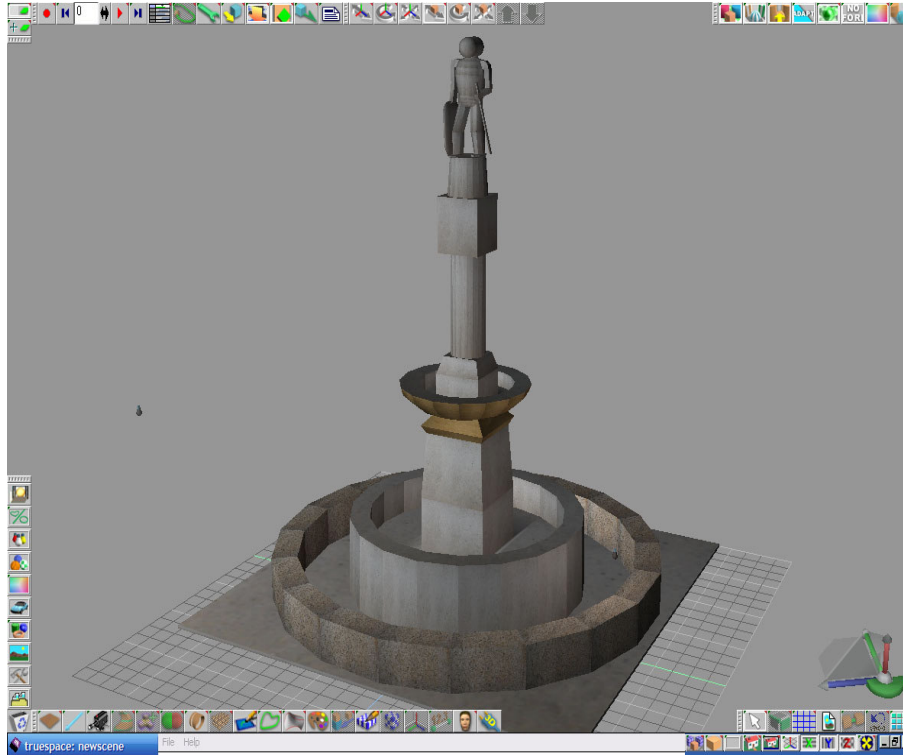
Zrkadlové modelovanie

Dá sa využiť na modelovanie symetrických objektov, kde môžeme zvýšiť efektívnosť modelovania. S **mirror modeler tool**  sa sústreďíme na polovicu symetrického objektu, zmeny potom zduplikujeme v druhej, zrkadlovej časti objektu. Tento prístup sa dá špeciálne využiť pri objektoch, kde sa obávame o zachovanie symetrie objektu. Postup pri modelovaní Rolanda je nasledovný : do pracovnej plochy vložíme **rounded cylinder**  a vertikálne ho **zrežeme**  na polovicu. **Označíme** zrezanú stenu a aktivujeme **mirror modeler** , čím sa rovina stane rovinou zrkadla. Z objektu **vyextrudujeme**  ruku a nohu podobným spôsobom ako v predchádzajúcom. Keď dosiahneme želaný tvar, LMB (ľavé tlačidlo myši) klikneme **erase mirror plane polygons** . Následne **create welded objects** , čím dostaneme podobný model ako v predchádzajúcom.



Obrázok B.1-5 : Ukážka využitia zrkadlového modelovania pri modelovaní trupu, nôh a ramien Rolanda

K zjednodušenej fontáne vložíme modelovaného, oba objekty **spojíme** a **otextúrujeme**.








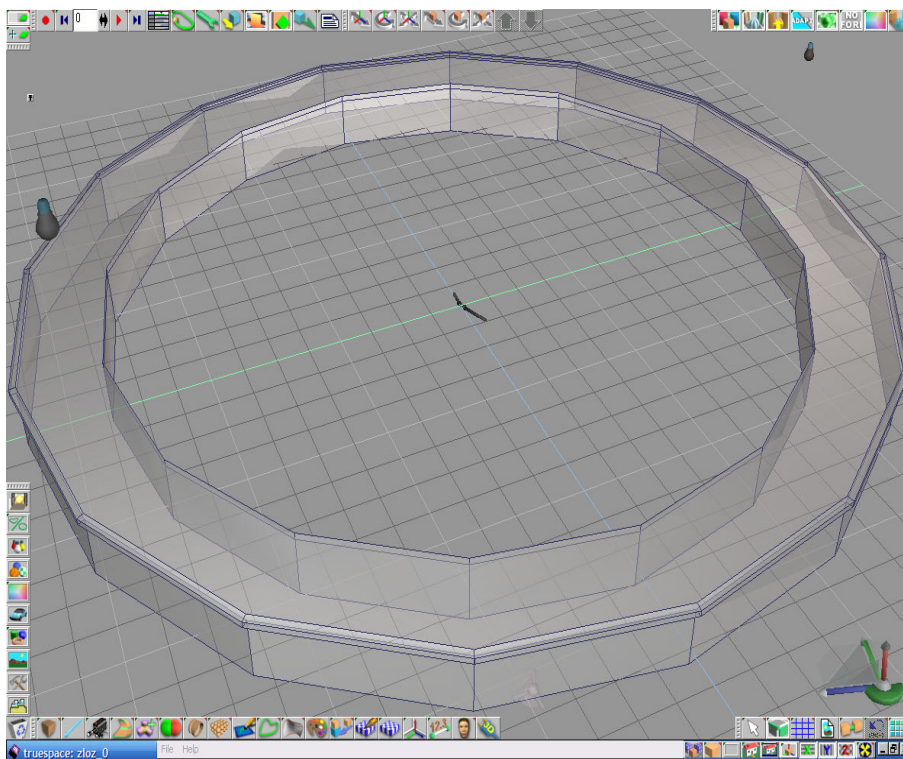
Obrázok B.1-6 : Zjednodušená fontána

Zložitejšia fontána

V tomto postupe budeme modelovať s **vyššou úrovňou detailov**. Objekt by mal byť rozpoznateľný zo vzdialenosti 10 metrov. Rolanda nebudeme modelovať, ale použijeme otextúrovaný objekt.







Vonkajší kruh fontány

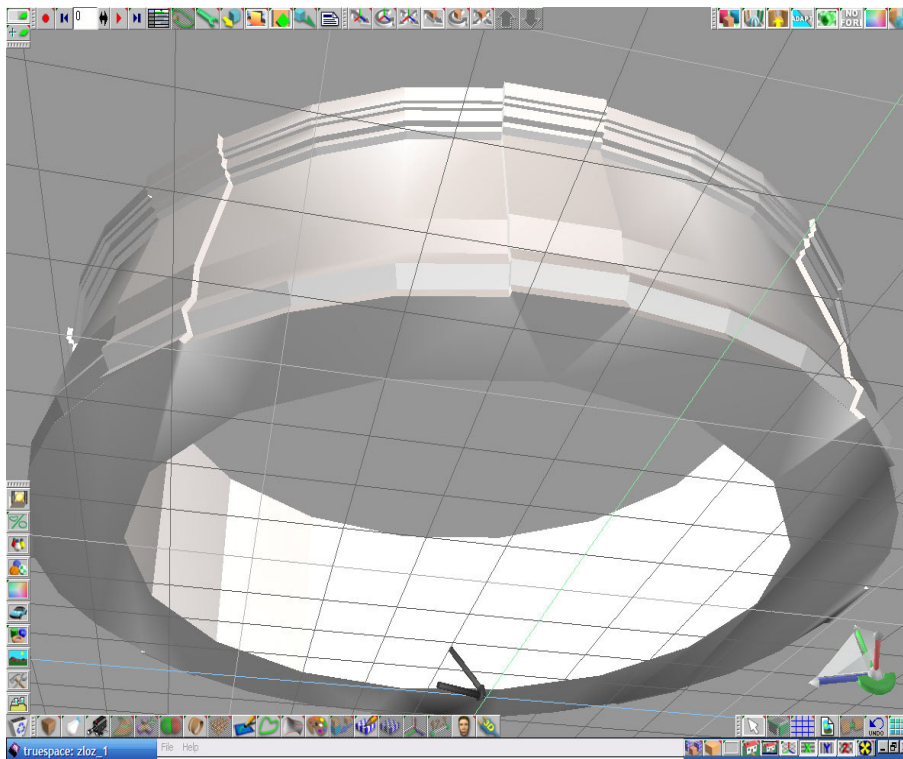
Do pracovnej plochy vložíme **valec**  so 16 bočnými stranami. Na hornej stene vyrobíme vyčnievajúci okraj nasledovne. **Označíme** hornú stenu valca a **vyextrudujeme**  ju. Vzniknutú stenu **rozšírime** a **umiestnime** do rovnakej výšky ako pôvodne extrudovaná stena. Opäť **označíme** hornú stenu valca a **vyextrudujeme** . Do stredu urobíme otvor. Vložíme do pracovnej plochy ďalší **valec**  a upravíme jeho šírku na 4/5 prvého. **Umiestnime** ho do stredu prvého valca a **odpočítame** .



Obrázok B.1-7 : Vonkajší kruh fontány



Vnútorňý kruh fontány









Do pracovnej plochy vložíme valec  so 16 bočnými stenami. Jeho šírku upravíme na $\frac{3}{5}$ vonkajšieho, výška zostáva rovnaká ako u vonkajšieho valca. Umiestnime ho do stredu. Na tento valec umiestnime ďalší, rovnako široký, ale sploštený valec  s 32 bočnými stenami. Každú štvrtú bočnú stenu vyextrudujeme  a na hornej stene vymažeme  pôvodné hrany extrudovaných stien. Označíme hornú stenu valca a vyextrudujeme  ju 8x. Pri 5. až 8. extrudovaní urobíme na bočných stenách vrstvy (viď obrázok B.1-8). Po získaní želaného tvaru urobíme  do valca podobne ako pri vonkajšom kruhu valcový otvor.

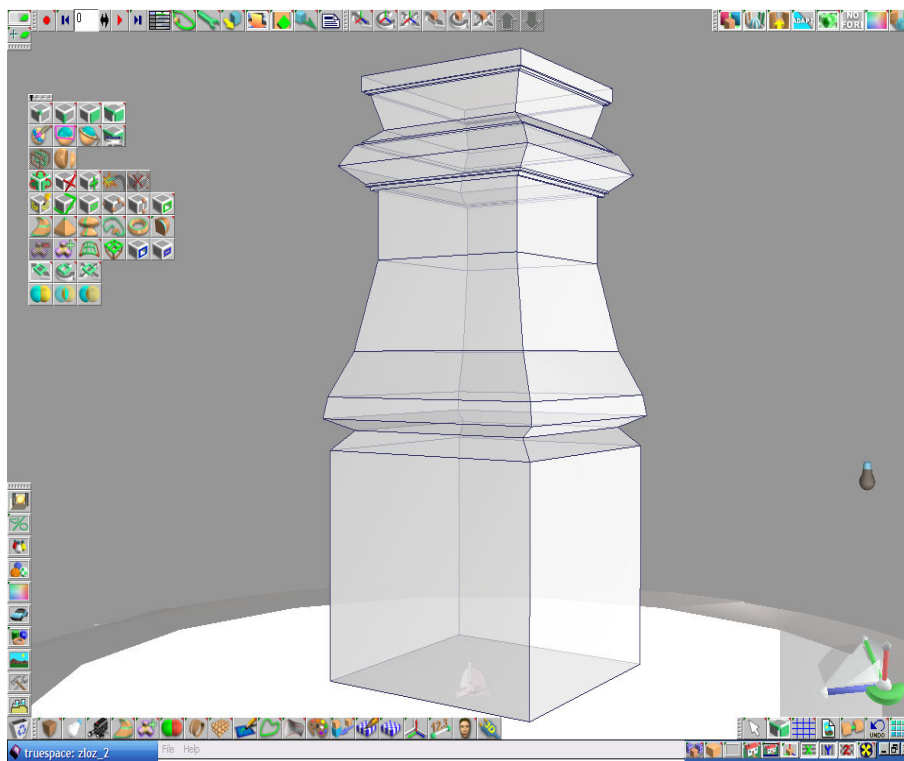


Obrázok B.1-8 : Vnútorý kruh fontány

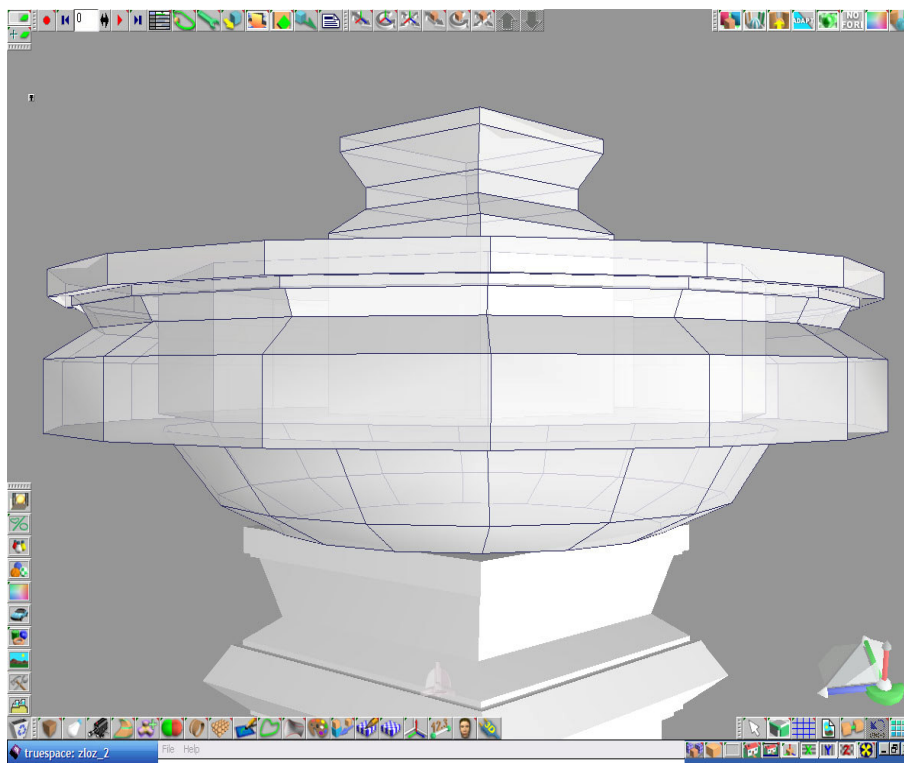
Nížší podstavec

Do stredu vnútorného kruhu fontány vložíme **kocku** , **označíme** hornú stenu a 15x **vyextrudujeme**  do tvaru zobrazeného na obrázku B.1-9.

Na kocku umiestnime **guľu** , ktorú do 3/4 zvrchu **zrežeme**  (môžeme na to použiť **kocku** , ktorú nad ňu **umiestnime**). Ďalej vložíme do scény **valec**  s počtom bočných stien 16 a **umiestnime** ho nad zrezanú guľu. Upravíme jeho šírku tak, aby zrezanú guľu trochu prečnieval. Hornú stenu **extrudujeme**  4x (viď obrázok dole). Potom v jeho strede **urobíme**  otvor v tvare valca. Do otvoru umiestnime **kocku**  a jej hornú stenu 4x **vyextrudujeme**  (jej tvar je zobrazený na obrázku B.1-10).









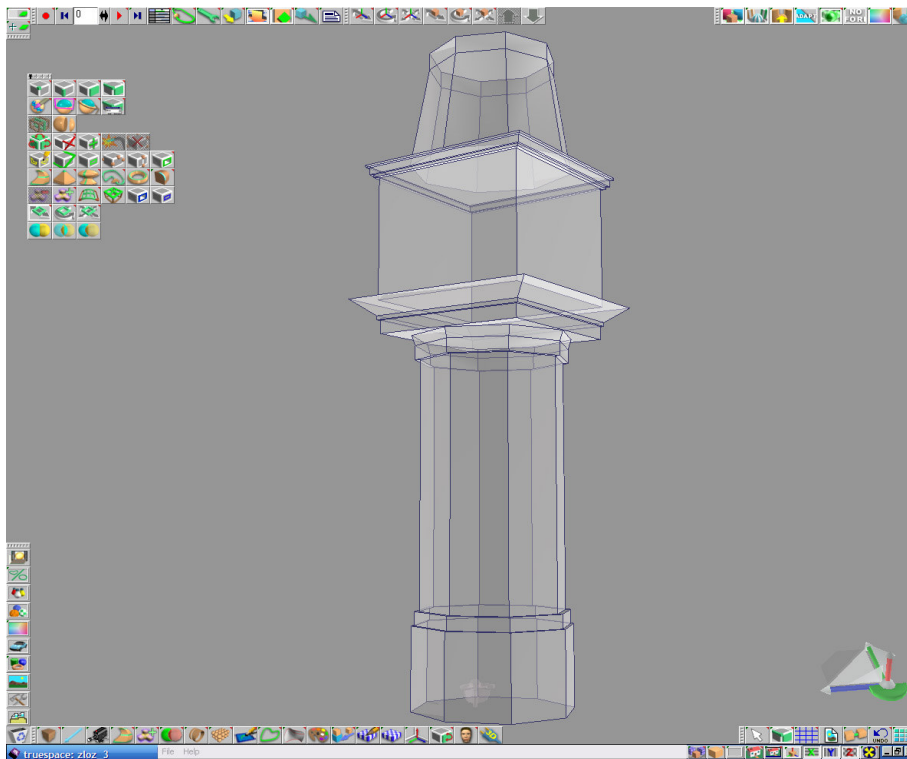
Obrázok B.1-9 : Nižší podstavec



Obrázok B.1-10 : Nižší podstavec hore

Vyšší podstavec

Na nižší podstavec umiestnime **valec**  s 8 stenami, šírka bude rovnaká ako šírka vrchnej steny kocky. Hornú stenu valca 4x **vyextrudujeme** , snažíme sa robiť vrstvy. Na valec umiestnime sploštenú **kocku** . Tú **vyextrudujeme**  7x podobne ako predchádzajúci valec. Navrch umiestnime ďalší **valec** , jeho hornú stenu zmenšíme a **vyextrudujeme** .



Obrázok B.1-11 : Vyšší podstavec

Podlaha


Do pracovnej plochy vložíme **kocku** , **splošíme** ju a **umiestnime** pod fontánu. Nakoniec tohto postupu všetky vymodelované časti **spojíme**.

Textúrovaný Roland

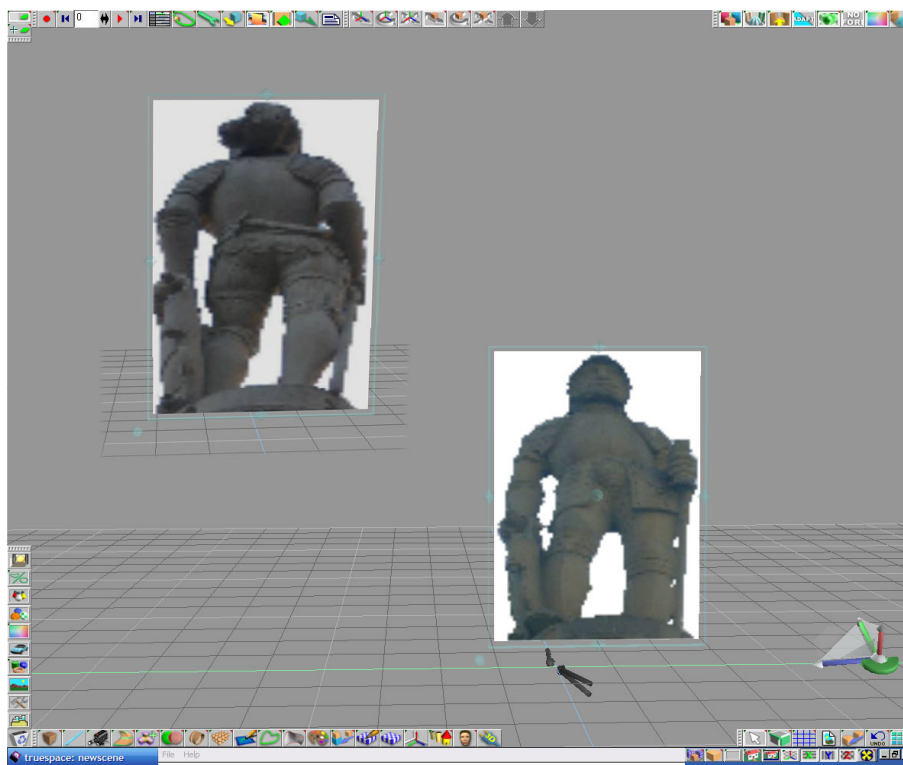
Pomocou snímok, ktorými **textúrujeme** objekty, môžeme urobiť dobrý model daného objektu a ušetriť čas venovaný modelovaniu. Veľkou výhodou je, že ak použijeme

kvalitné zábery, nemusíme modelovať určité časti objektu do detailov. Túto myšlienku som využil aj pri modelovaní Rolanda.

Zvislá rovina


Vložíme rovinu  do pracovnej plochy a zrotujeme ju o 90 stupňov okolo x-ovej osi. Jednu stenu otextúrujeme snímkou Rolanda spredu a na opačnú stranu dáme textúru Rolanda zozadu. Textúry sú typu PNG (Portable Network Graphics), pretože majú priesvitné pozadie. Formát JPEG (Joint Photographic Experts Group) priesvitne pozadie snímok neumožňuje.

Týmto prístupom sa dá dosiahnuť veľmi dobrý efekt s malou námahou. Nevýhodou je, že Roland nemá objem, je iba stena.



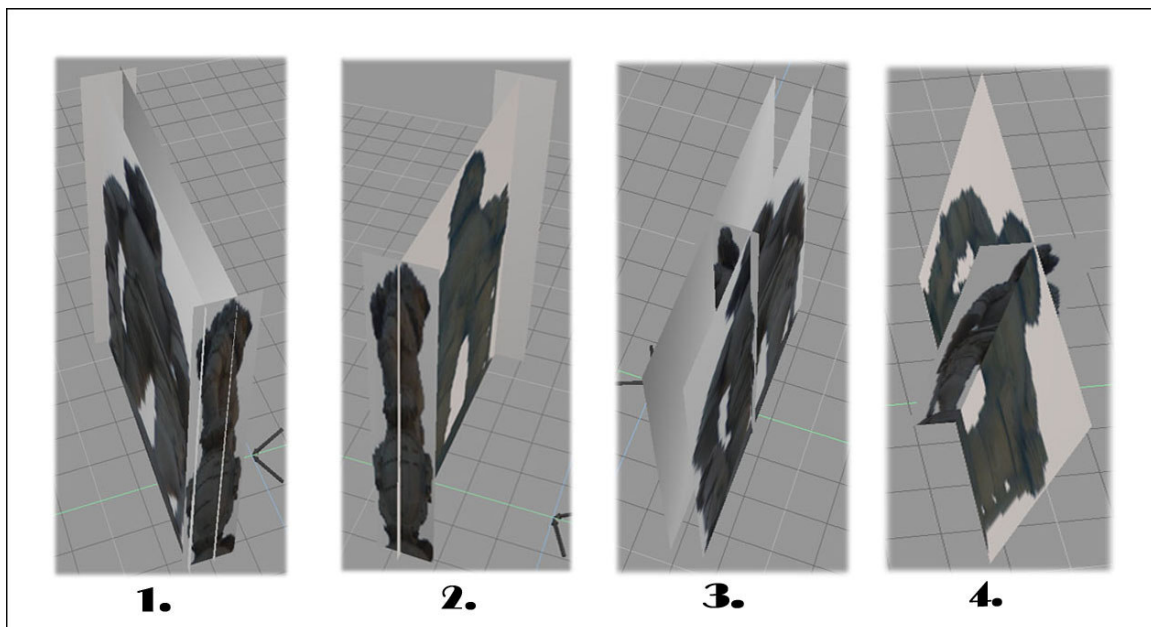
Obrázok B.1-12 : Textúrovaný Roland – zvislá rovina

Roviny

Do scény umiestňujeme **roviny**  a tie **textúrujeme** obrázkami Rolanda spredu, zozadu, zľava a sprava. Pre textúry platí to isté ako v predchádzajúcom odseku, musia byť typu PNG.


Roviny môžu byť umiestnené týmito spôsobmi :


1. **v tvare kvádra** – nevýhoda je, že pri otáčaní vidieť, že objekt má diery,
2. **v tvare kvádra, ale predná a zadná rovina sú spojené a strede** – lepší efekt ako prvý prístup, ale pri určitom pootočení objektu vidieť, že roviny sú len poukladané v priestore,
3. **v tvare kvádra, ale zľava a sprava je jedna rovina tiež umiestnená v strede** - podobný výsledok ako v bode 2,
4. **dve križujúce sa roviny** – zo všetkých pokusov najlepšie riešenie.

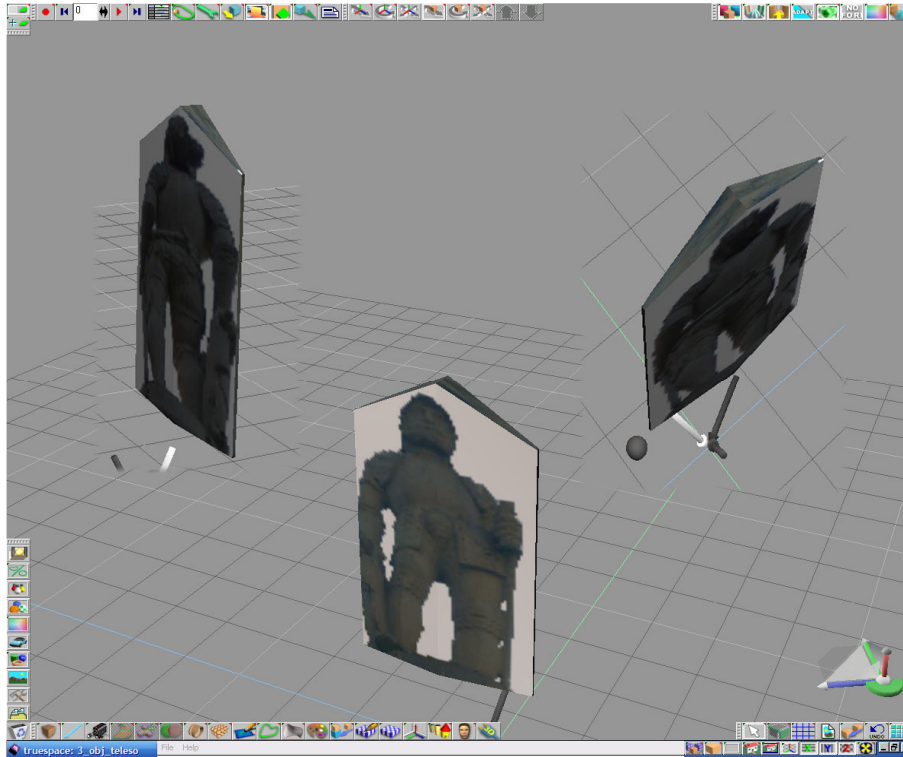


Obrázok B.1-13 : Textúrovaný Roland – roviny




Otextúrovaný objekt

Uplatňujeme podobný princíp ako pri rovinách s tým rozdielom, že sa snažíme vytvoriť vhodný objekt. Do scény vložíme **kocku**  a tú upravíme v editovacom móde

(aktivujeme ho kliknutím RMB na objekt) do tvaru na obrázku dole. Postupujeme nasledovne. **Zúžime** kocku v smere spredu dozadu. Do stredu hornej steny **pridáme**  úsečku a vrchné hrany bočných steny zľava a sprava **posunieme** dole. Potom upravenú kocku **textúrujeme**. Textúrujeme aj boky, aby sme zabránili priehľadnosti cez ne.

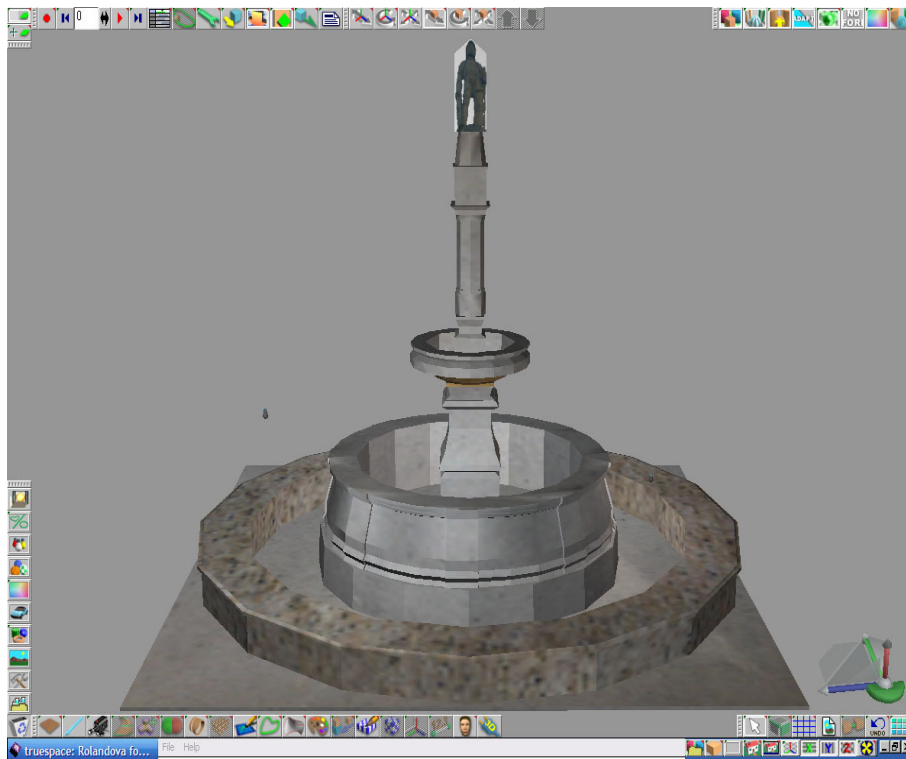


Obrázok B.1-14 : Textúrovaný Roland – otextúrovaný objekt

Do pracovnej plochy vložíme **valec**  s 8 bočnými stenami a **upravíme** (viď obrázok dole). Vrchnej a dolnej stene **prevrátíme**  normály. Zvislé steny **textúrujeme** so snímkami Rolanda spredu, zozadu, zľava a sprava s priesvitným pozadím. Zvolíme **cylindrickú projekciu** .

V oboch prípadoch tohto prístupu som dosiahol dobrú náhradu modelu.




Do scény zložitejšej fontány vložíme objekt textúrovaného Rolanda. Tento objekt **umiestnime** na vrch fontány a **vyškálujeme**, aby jeho rozmery boli v dobrom pomere ku fontáne. Fontánu ešte **otextúrujeme** - stačia tri základné textúry, ktoré nahrádzajú tri základné farby fontány pri pohľade z väčšej vzdialenosti.

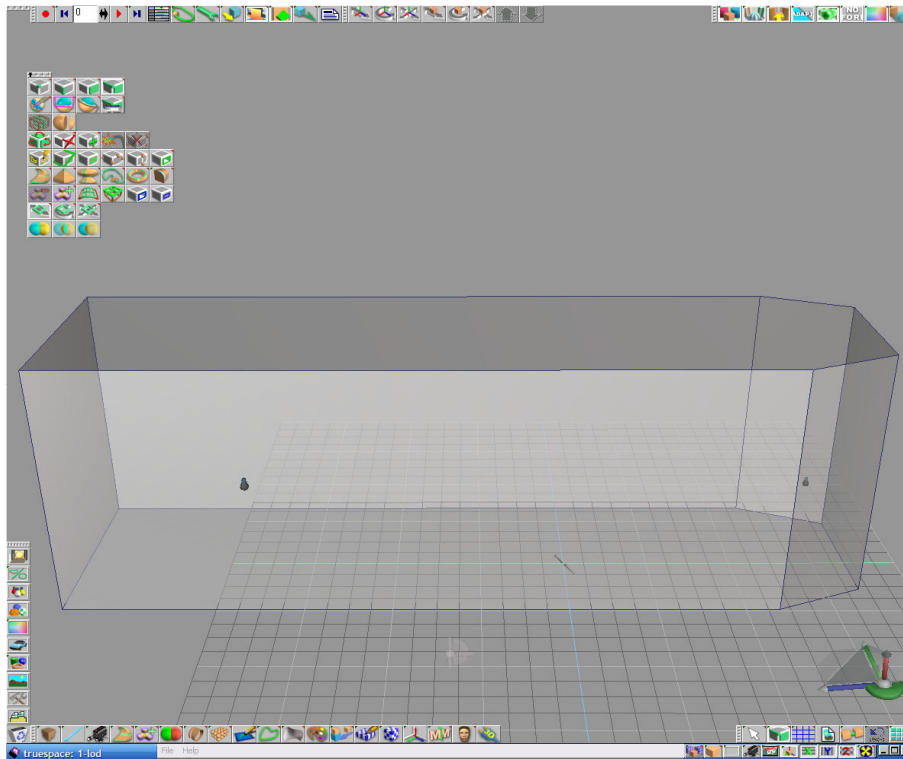


Obrázok B.1-15 : Zložitejšia fontána

B.2 Klarisky

Lod'





Do PP vložíme **kocku**  s rozmermi $x = 32,5$; $y = 9$; $z = 12$; jej zvislú stenu sprava **označíme**, **extrudujeme**  o 4 v smere x-ovej osi a horizontálne **zúžime** o $1/3$. Hrany pôvodne extrudovanej steny navrchu aj na podstave **vymažeme** .











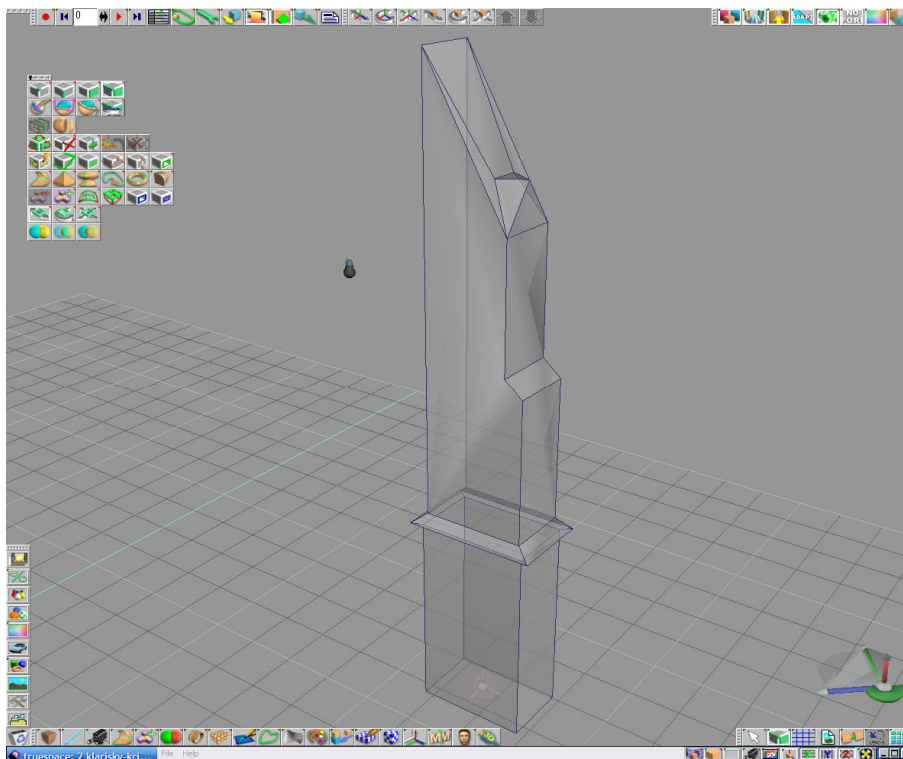
Obrázok B.2-1 : Lod' Klarisiek

Krídla

Prvé krídla na vonkajšej strane












Do PP vložíme **kocku** , z nej vymodelujeme prvé štyri krídla, ktoré **umiestnime** na vonkajšej strane lode v smere sprava doľava. Rozmery kocky sú $x = 1,25$; $y = 2$; $z = 3,8$. Vrchnú stenu kocky **označíme** a **vyextrudujeme** . Vzniknutú stenu **rozšírime** a **umiestnime** do rovnakej výšky ako pôvodne extrudovaná stena. **Vyextrudujeme**  je a **zúžime**. Opäť **extrudujeme**  2x a po druhom raze stenu zúžime - **posunieme**

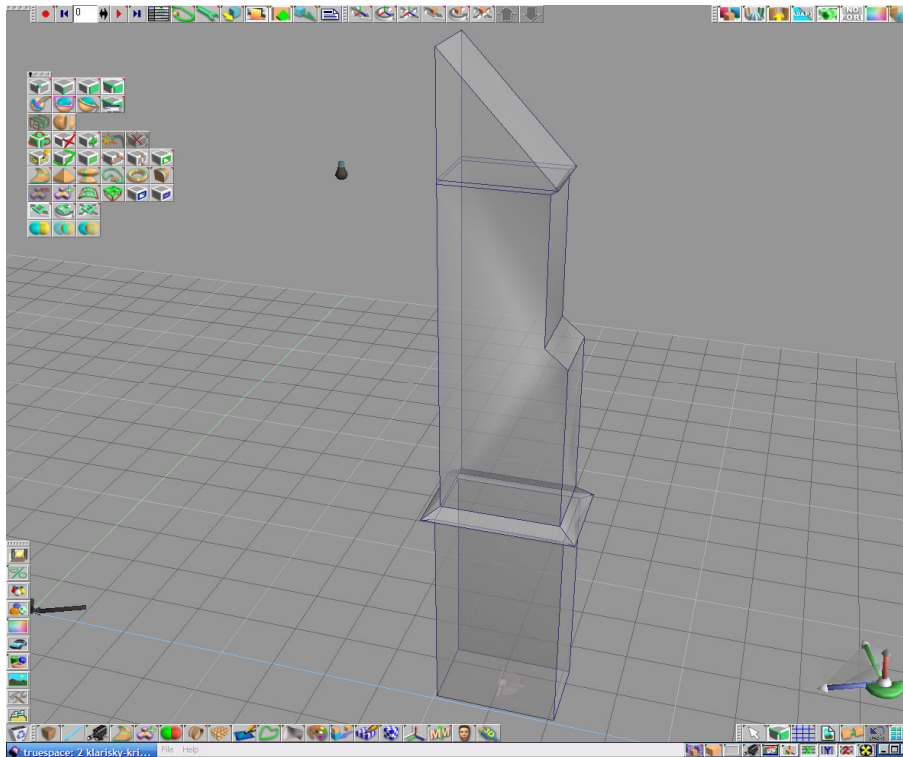
vonkajšiu hranu o 0,4 dovnútra. Hrany pôvodnej steny **vymažeme**  okrem hrany vpredu. Znova **extrudujeme**  o 2,3 v smere osi z a pôvodné hrany po bokoch okrem hrany vpredu **vymažeme**  ako v predchádzajúcom. **Označíme** si zvislú stenu vzadu hore a **predĺžime** jej hornú stenu v smere osi z o 1,8. To nám spôsobí zošíkmenie vrchnej steny, na ktorej vyrobíme v editovacom móde (RMB na objekt) "cuplík" vyčnievajúci von. Hrany pôvodnej steny **vymažeme**  okrem hrany vpredu. Asi v 1/4 zospodu **pridáme**  do vrchnej steny v horizontálnom smere hranu. Vo vzniknutom dolnom obdĺžniku **pridáme**  po bokoch ďalšie dve hrany tak, aby nám vznikol lichobežník, ktorý má základňu dlhšiu ako vrchnú úsečku. Do lichobežníka urobíme uhlopriečky. Vzniknutý stred **vysunieme**, na hornej hrane pôvodného obdĺžnika **vymažeme**  po bokoch menšie hrany a horné body lichobežníka **ukotvíme**  o vrchnú hranu zošíkmenej steny.









Obrázok B.2-2 : Prvý druh krídel z vonkajšej strany lode

Ďalší druh krídla na vonkajšej strane

Teraz si namodelujeme ďalší druh krídla, ktorým budeme pokračovať v dopĺňaní vonkajšej strany lode klarisiek smerom sprava doľava 5x. Postup je obdobný ako pri prvom krídle. Vložíme **kocku**  s rozmermi $x = 1,25$; $y = 2$; $z = 3,8$; vrchnú stenu kocky **označíme** a **vyextrudujeme** . Vzniknutú stenu **rozšírime** a **umiestnime** do rovnakej výšky ako pôvodne extrudovaná stena. **Vyextrudujeme**  ju a **zúžime**. Vrchnú stenu **extrudujeme** , na druhý krát jej vonkajšiu hranu **posunieme** o 0,4 dovnútra. Hrany pôvodne extrudovanej steny **vymažeme** , okrem hrany vpredu. Znova **extrudujeme**  vrchnú stenu o 2,3 v smere osi z a pôvodné hrany okrem hrany vpredu **vymažeme**  ako v predchádzajúcom. Ďalej postupujeme inak ako pri prvom krídle. Vrchnú stenu **extrudujeme**  a **predĺžime** jej vonkajšiu hranu dopredu. Opäť **extrudujeme**  v smere osi z o 1,8 a vrch zrežeme (viď obrázok B.2-3). Dá sa to urobiť viacerými spôsobmi, napr. **pridaním**  uhlopriečok po bočných stenách a **vymazaním**  vyčnievajúcej úsečky vpredu na hornej stene.






Obrázok B.2-3 : Ďalší druh krídel z vonkajšej strany lode








Prvé krídlo si **nakopírujeme**  4x a pomocou **move** a **rotate** umiestňujeme na správne miesta k vonkajšej stene hlavnej lode sprava doľava (vid' obrázok B.2-4). Pri prvých dvoch umiestneniach tohto krídla ešte celú zadnú stenu **vyextrudujeme** . Druhé krídlo **nakopírujeme**  5x a pokračujeme v jeho umiestňovaní popri vonkajšej stene (vid' obrázok B.2-4). Pri treťom a štvrtom umiestnení urobíme na krídle ešte jednu malú zmenu. Zospodu krídel **vymažeme**  prvé dve vrstvy štvoruholníkov, **označíme** si ďalšiu vrstvu v poradí a **vyextrudujeme**  dole na pôvodnú výšku krídla. Ak si chceme byť istý, že máme správne vzdialenosti medzi jednotlivými krídlami, môžeme si pomôcť pomocnými **kockami**  s upravenými rozmermi, ktoré budeme medzi krídla umiestňovať (vid' obrázok B.2-4).

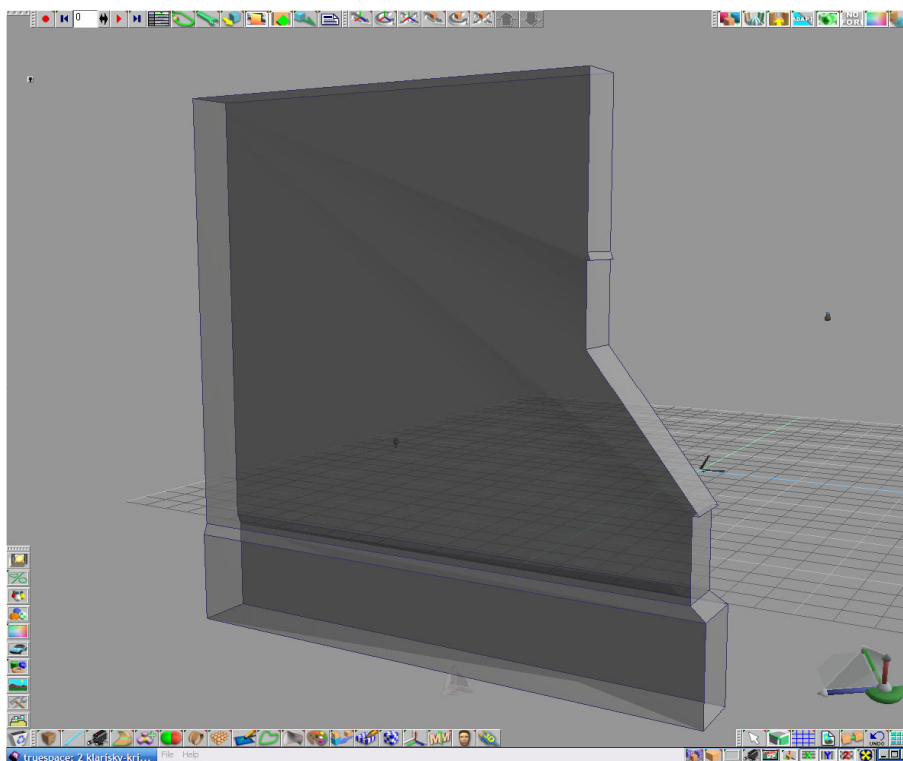


Obrázok B.2-4 : Vonkajšia stena Klarisiek

Posledné krídlo na vonkajšej stene




Do scény vložíme **kocku**  s rozmermi $x = 1,4$; $y = 13,3$; $z = 2,4$; vrchnú stenu **vyextrudujeme**  a zúžime - vonkajšiu a bočné hrany **posunieme** do vnútra. **Označíme** vrchnú stenu a **extrudujeme**  v smere osi z o 1,8. Teraz na zvislej hornej

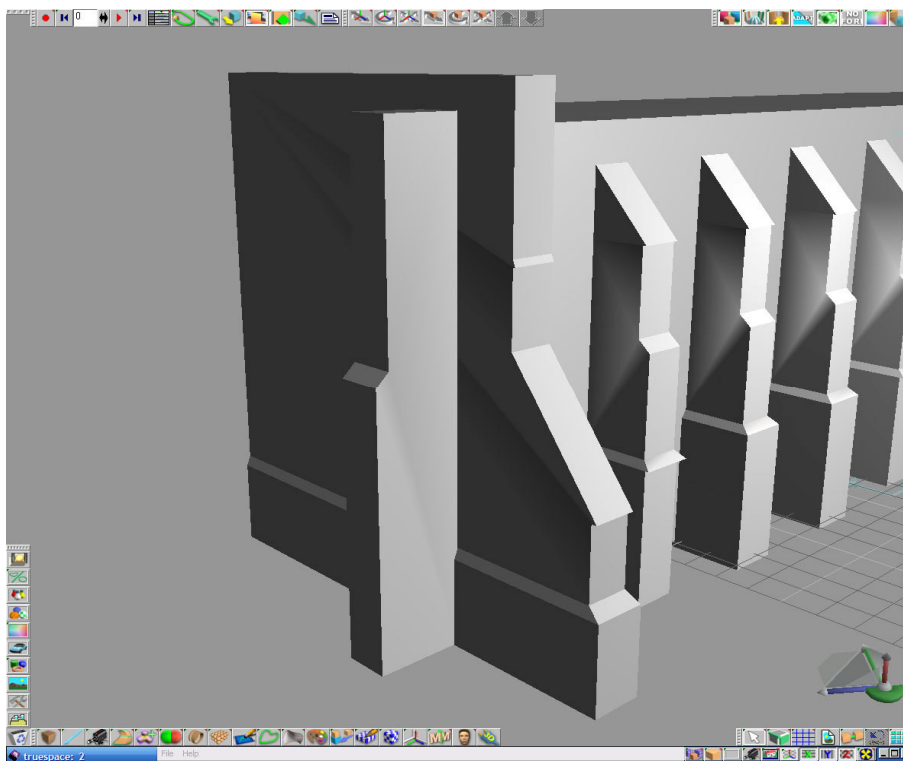
stene vpredu vyrobíme "výčnelok". **Pridáme**  do nej zvrchu v 1/8 horizontálnu úsečku, vzniknutý obdĺžnik **posunieme** hore, **vysunieme** dopredu a **zrotujeme** o 90 stupňov v smere osi x. Úsečku obdĺžnika bližšie ku objektu spojíme s hornou stenou pomocou funkcií **add edges**  a **weld vertices** . Vrchnú stenu znova **extrudujeme**  v smere osi z o 3,3 a **posunieme** jej prednú hranu dovnútra. Znova **extrudujeme**  a podobným spôsobom ako v predchádzajúcom vyrobíme ďalší výčnelok. Posledný krát **extrudujeme**  o 3,7 v smere osi z. Nakoniec **vymažeme**  bočné a zadné hrany niektorých extrudovaných stien (viď obrázok B.2-5).



Obrázok B.2-5 : Posledné krídlo vonkajšej strany










Zadné krídlo

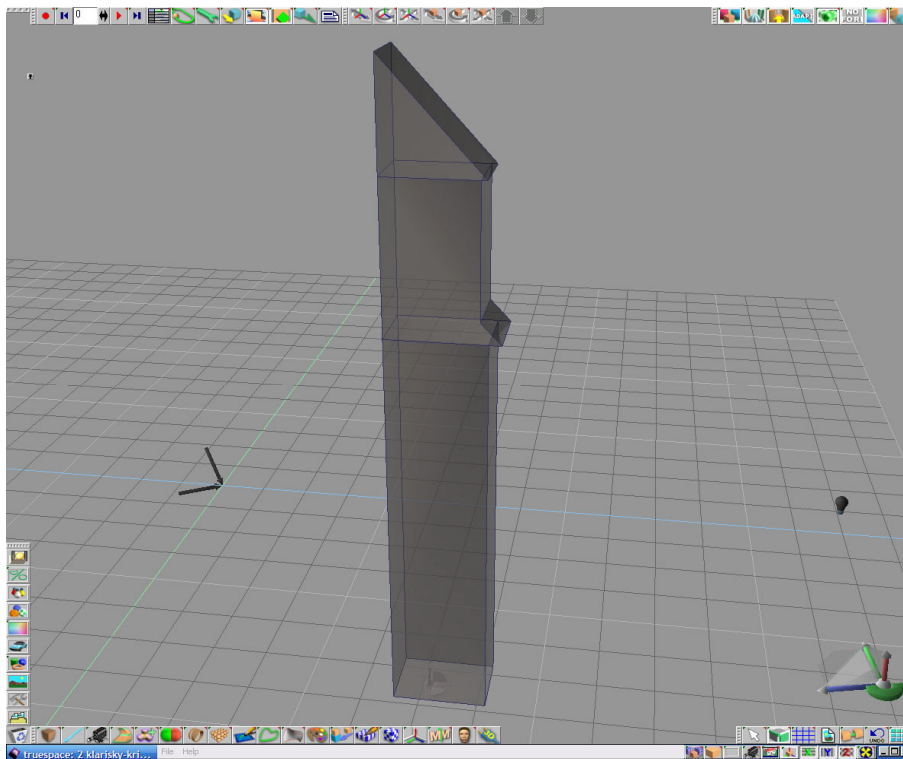
Do pracovnej plochy vložíme **kocku**  s rozmermi $x = 1,15$; $y = 2,3$; $z = 7$; vrchnú stenu **vyextrudujeme** , prednú hranu **posunieme** dovnútra a znova **vyextrudujeme** .





Obrázok B.2-6 : Zadné krídlo



Krídla na vnútornej strane

Teraz namodelujeme krídla na vnútornej strane lode. Do scény vložíme **kocku**  s rozmermi $x = 1,16$; $y = 1,7$; $z = 6,5$. Na zvislej prednej stene vyrobíme výčnelok. **Pridáme**  do nej zvrchu v 1/10 horizontálnu úsečku, vzniknutý obdĺžnik **posunieme** hore, **vysunieme** dopredu a **zrotujeme** o 90 stupňov v smere x-ovej osi. Úsečku obdĺžnika bližšie ku objektu spojíme s hornou stenou pomocou funkcií **add edges**  a **weld vertices**  podobne ako v predchádzajúcom. Vrchnú stenu **extrudujeme** , **posunieme** jej prednú hranu dovnútra. Znova **extrudujeme**  vrchnú stenu v smere osi z o 2,3 a vyrobíme výčnelok. Vrchnú stenu **vyextrudujeme**  a zrežeme podobne ako pri druhom krídle, **pridaním**  uhlopriečok po bočných stenách a **vymazaním**  vyčnievajúcej úsečky vpredu na hornej stene.



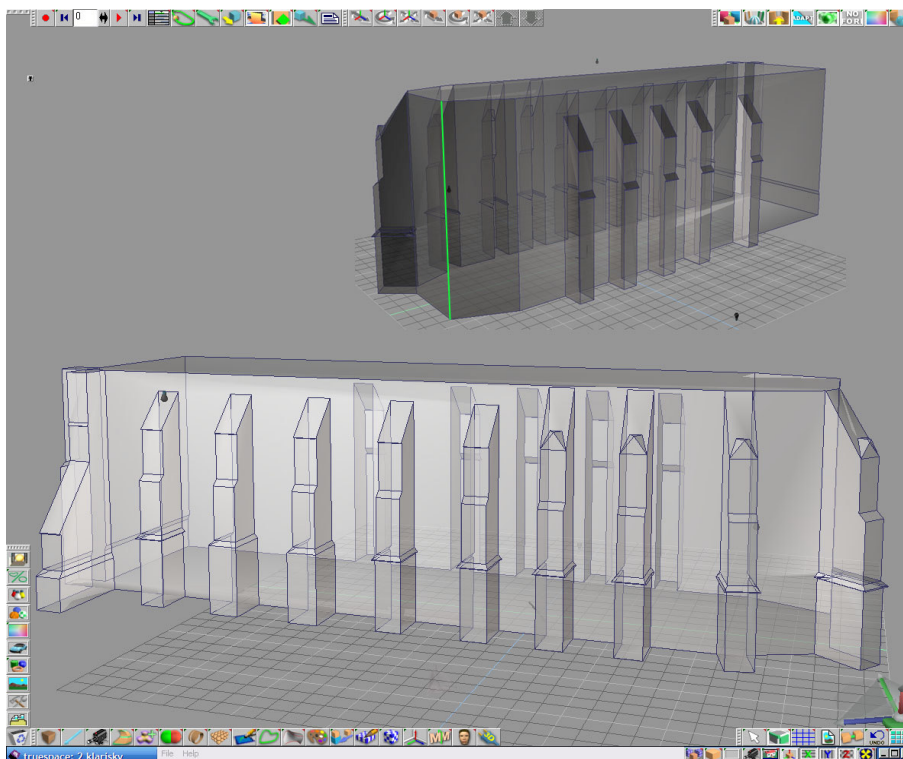
Obrázok B.2-7 : Krídla z vnútornej strany lode

Vymodelované krídlo **nakopírujeme**  5x a **umiestnime** pomocou funkcií **move** a **rotate** popri vnútornej stene lode (viď obrázok B.2-8). Rovnaké vzdialenosti medzi jednotlivými krídlami, dosiahneme pomocnými **kockami**  s upravenými rozmermi, ktoré budeme medzi krídla umiestňovať.

Všetky doteraz namodelované časti - loď aj krídla **spojíme**  a **zrežeme**  zvrchu aj zo spodu, aby sme dostali všetky body v podstave aj navrchu do rovnakej výšky (viď obrázok B.2-9).






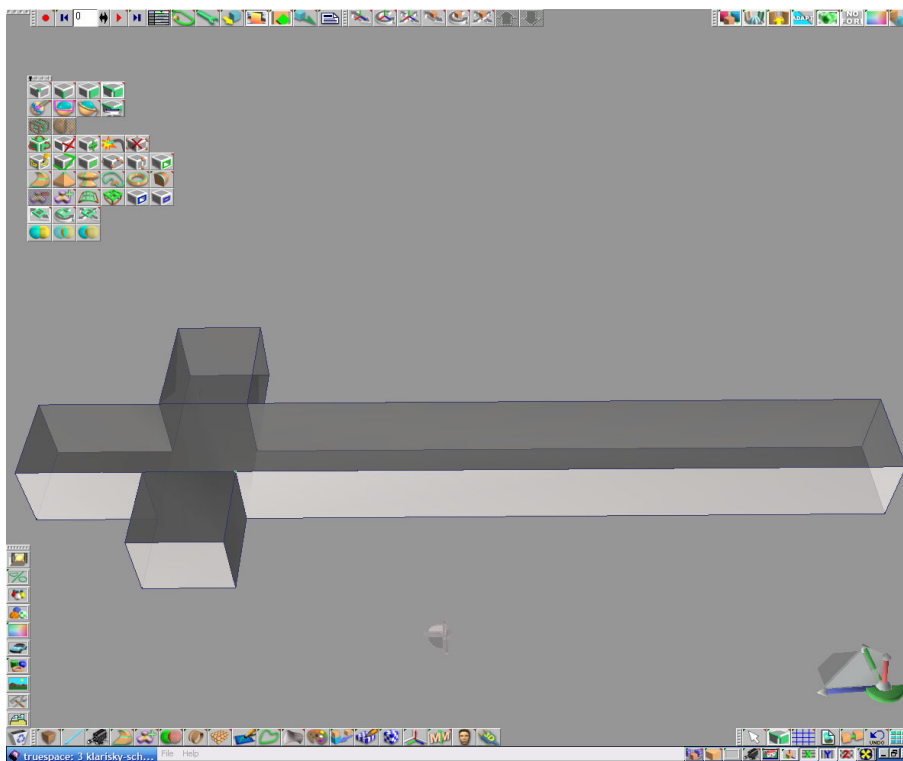
Obrázok B.2-8 : Umiestnenie krídel z vnútornej strany





Obrázok B.2-9 : Lod' Klarisiek spolu s krídlami

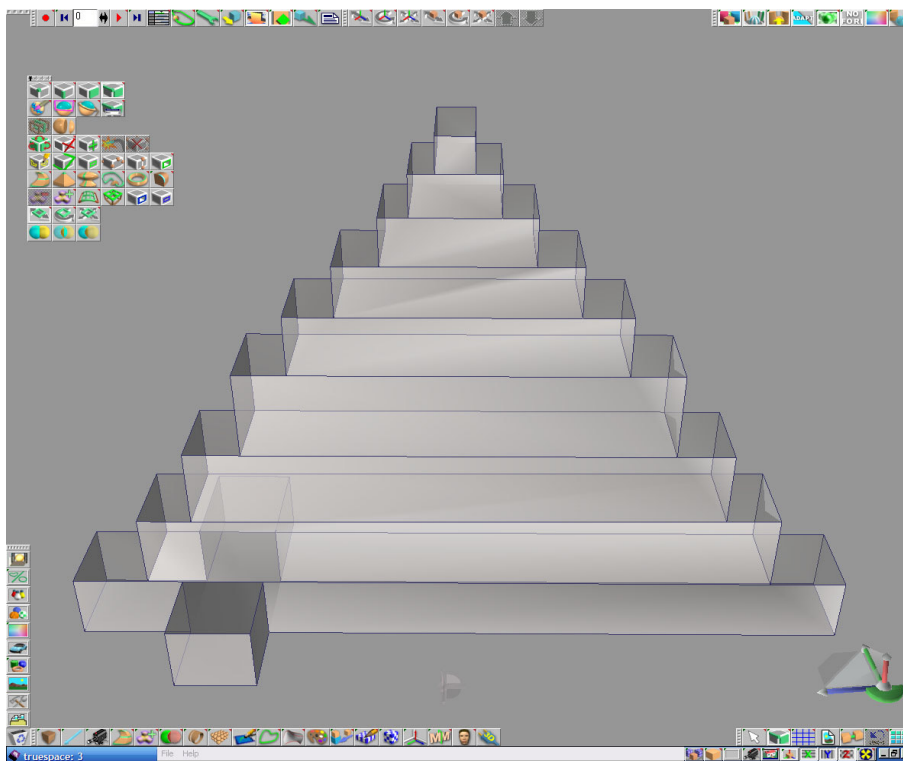
Schody

Do pracovnej plochy vložíme **kocku**  s rozmermi $x = 1,16$; $y = 10,8$; $z = 0,97$; vyrobíme z nej ležiaci kríž. Do zvislých stien napravo a naľavo **pridáme**  dve vertikálne hrany. Prvá bude vzdialená od vonkajšej steny 1,5 v smere Y a druhá o 2,65. Vzniknuté obdĺžniky **vyextrudujeme**  , sprava o 1 a zľava o 1,4.














Obrázok B.2-10 : Základný kríž pre modelovanie schodov

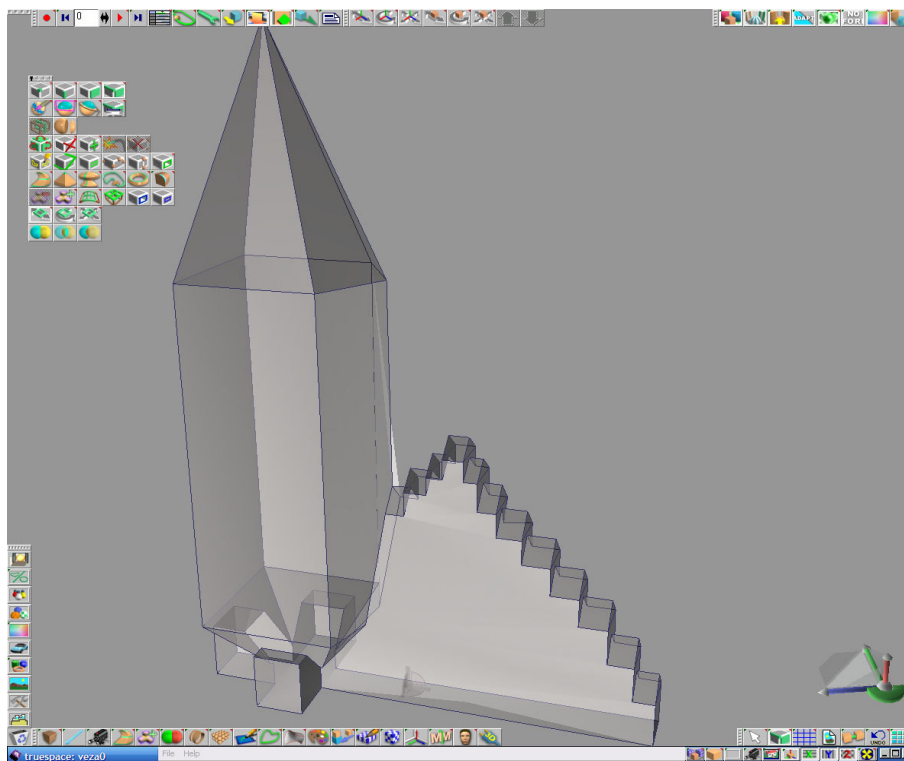
Na vrchnej stene začneme vyrábať schody. **pridáme**  do nej dve horizontálne hrany v smere sprava doľava v rovnakej vzdialenosti od krajov a vzniknutý obdĺžnik v strede **vyextrudujeme**  . Tento istý postup opakujeme ešte 7x pričom úmerne zväčšujeme vzdialenosti od krajov, aby nám na vrchu zostal schodík s rozmermi približne $x = 1,16$; $y = 0,5$.



Obrázok B.2-11 : Schody





Veža

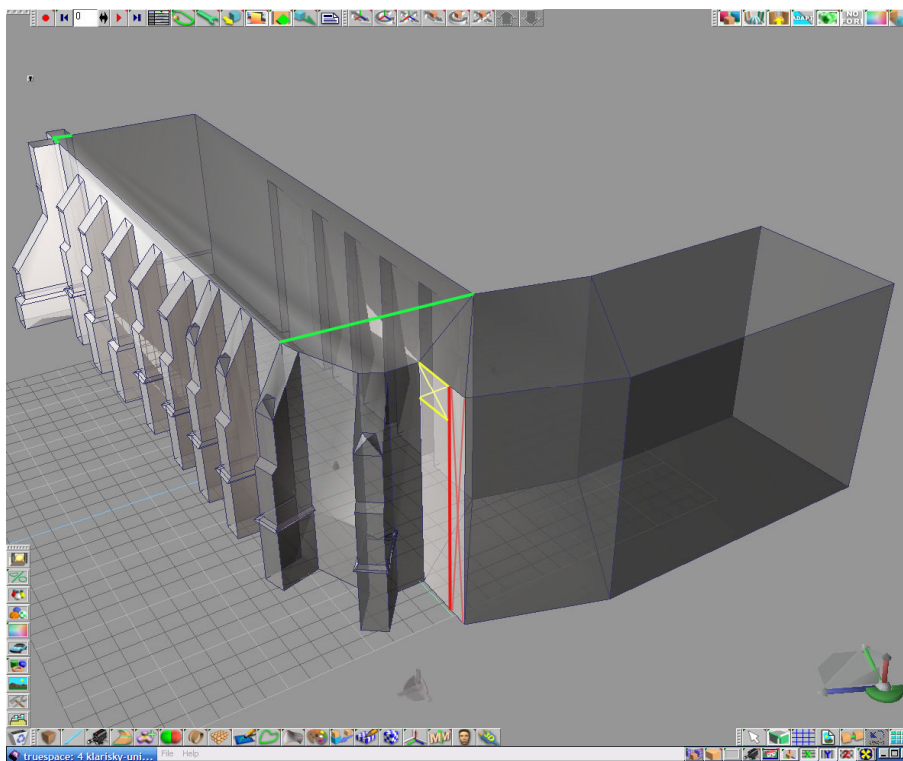
Do pracovnej plochy vložíme **kocku**  s rozmermi $x = 3,5$; $y = 3,5$; $z = 8,2$. **Označíme** jej zadnú stenu, **vyextrudujeme**  o 1 a na vyextrudovanej stene **spojíme**  vrchnú aj dolnú horizontálnu úsečku do jedného bodu. Vrchnú a spodnú hranu pôvodnej steny **vymažeme** . Ďalej **vyextrudujeme**  vrchnú stenu o 5,4 a dolnú stenu o 1,8 v smere osi z. Všetky body oboch vyextrudovaných stien **spojíme**  do jedného. Vymodelovanú vežu **spojíme**  z vonkajšej strany so schodmi, ktoré ešte pred pripojením **poposúvame** ku vnútornej strane. Na schodoch prirobíme striešky. **vyextrudujeme**  vrchnú stenu každého schodíka a na vyextrudovanej stene **spojíme**  vonkajšiu hranu a podobne aj vnútornú do jedného bodu. Bočné hrany striešky spojíme so susednými strieškami funkciami **add edges**  a **weld vertices** .



Obrázok B.2-12 : Veža







Univerzitná knižnica



Na pravej strane lode **označíme** zvislú úsečku - je vyznačená zelenou farbou na poslednom obrázku v druhej sekcii postupu a **zmeníme** jej Y súradnicu na 0. Ak máme objekt umiestnený na inom mieste ako v tomto postupe, k y-ovej súradnici pripočítame približne 3,7. **Označíme** si zošikmenú stenu, ktorá sa nám predchádzajúcim úkonom zväčší a **vyextrudujeme** . Kváder, ktorý nám vznikne zrežeme sprava na polovicu. **pridáme**  uhlopriečky na vrchnú a bočnú stenu a **vymažeme**  vyčnievajúcu zvislú úsečku na pravej strane. Stenu ešte dva krát **vyextrudujeme**  a následnou **zmenou polohy a rotácie** dosiahneme želaný tvar (vid' obrázok B.2-13).



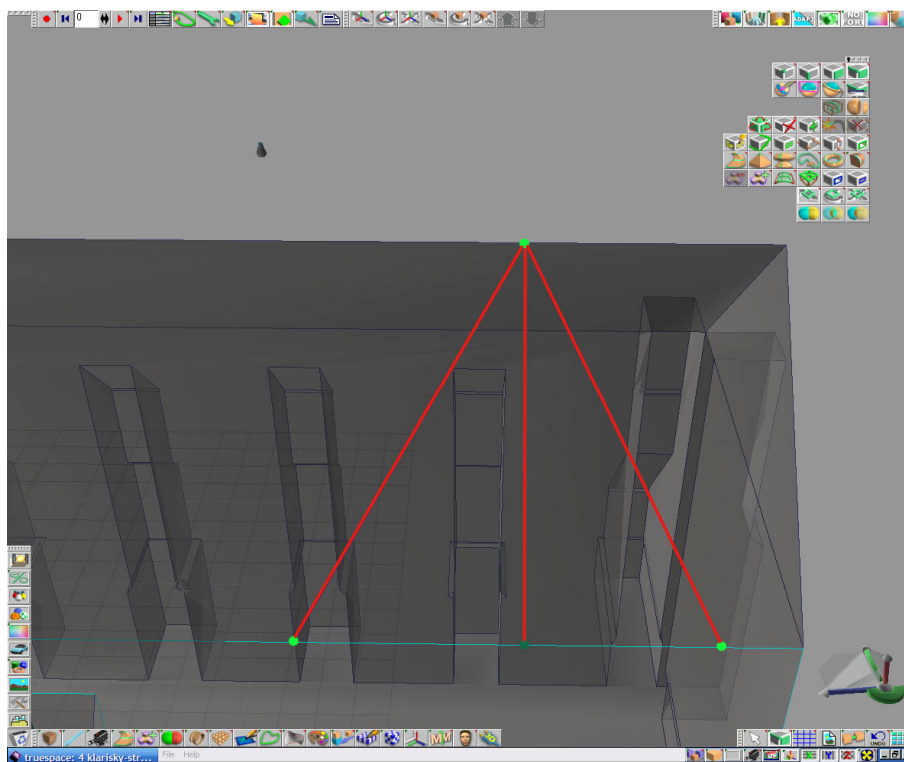
Obrázok B.2-13 : Univerzitná knižnica

Strecha nad klariskami

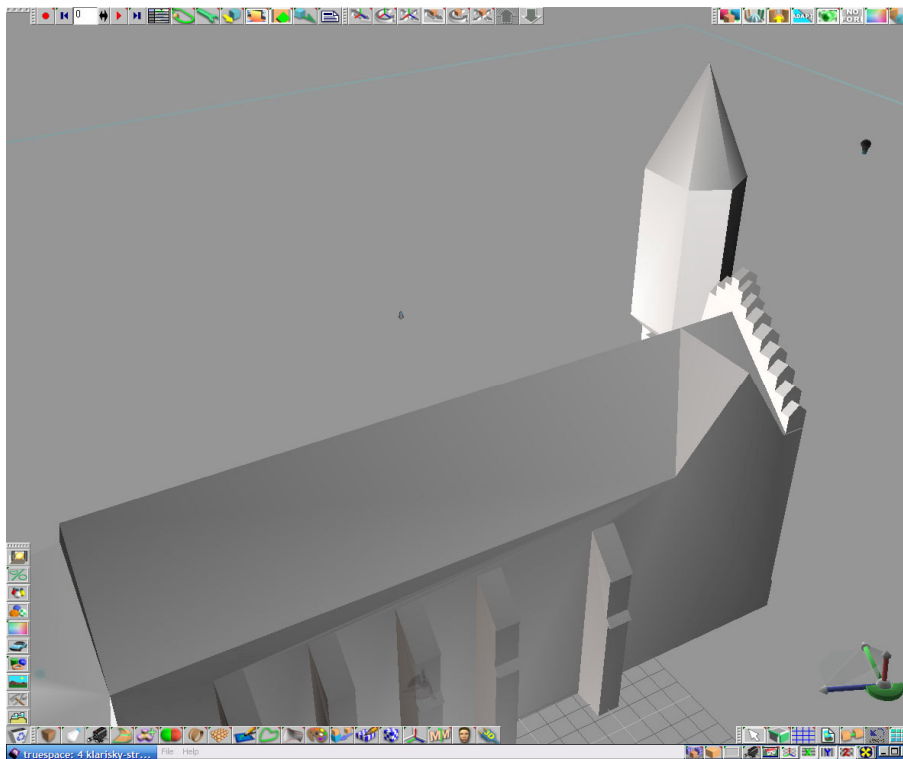
Nad hlavnou loďou vyrobíme strechu. Na jej vrchnej stene **urobíme**  tri úsečky tak, aby nám vznikol pravidelný obdĺžnik, ktorý budeme **extrudovať** . Najskôr **pridáme**  hranu pri poslednom krídle na vonkajšej stene, potom pri jednoduchom krídle na zadnej stene a nakoniec **pridáme**  ešte jednu hranu na opačnej strane (viď obrázok B.2-13 - zelená farba). Obdĺžnik **vyextrudujeme**  o 4,6 v smere *z-ovej* osi a **spojíme**  pravú aj ľavú úsečku po bokoch do jedného bodu.

Na ľavej vnútornej strane strechy urobíme menšiu strechu. Na dané miesto **pridáme**  štyri body (viď obrázok B.2-14 - zelená a tmavozelená farba) a **spojíme**  úsečkami (viď obrázok B.2-14 - červená farba). Polohu bodov zvolíme tak, aby bol spodný bod, ktorý je bližšie ku okraju, vzdialený od neho približne 1,2 a podstava celého trojuholníka mala dĺžku 6,4. Ďalšie dva body - tmavozelený a zelený na hornej hrane ležia v strede spomínaných bodov.







Tmavozelený bod **umiestnime** do rovnakej výšky ako vrchný bod trojuholníka. K ľavej strane strechy pripojíme vymodelovanú vežu a **spojíme** s objektom (viď obrázok B.2-15).







Obrázok B.2-14 : Bočná strecha Klarisiek

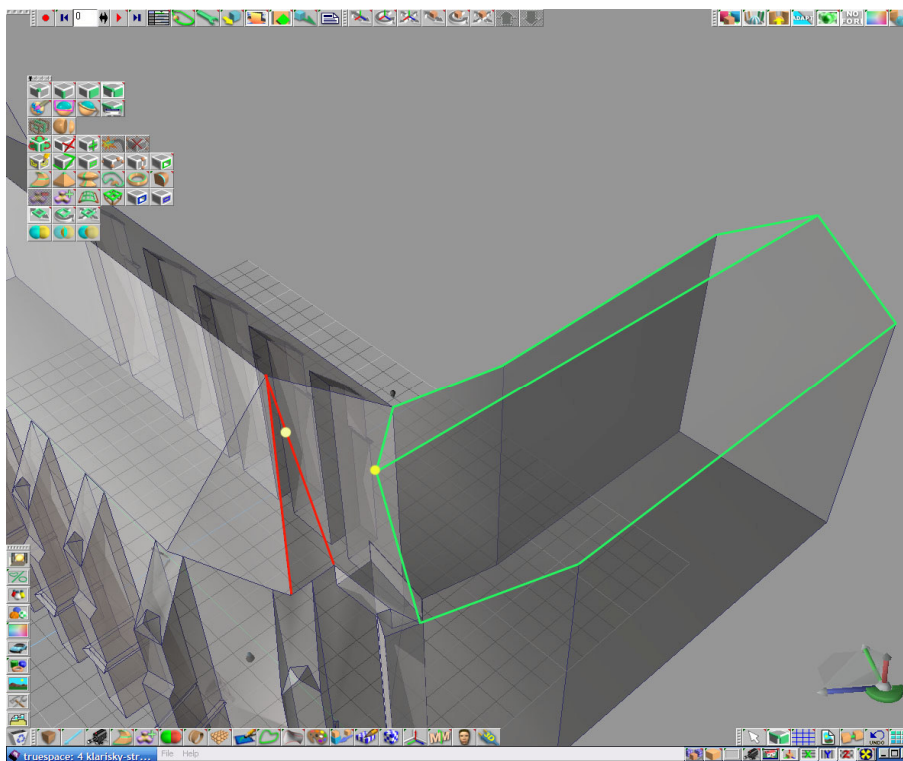


Obrázok B.2-15 : Klarisky



Na susednú stenu naľavo od steny vstupnej brány **pridáme**  vertikálnu úsečku (vid' obrázok B.2-13 - červená farba) a vzniknutý obdĺžnik (vid' obrázok B.2-13 - tmavočervený krížik) **vyextrudujeme** . Do tej istej steny **pridáme**  dve vertikálne úsečky (vid' obrázok B.2-13 - žltá farba) a vzniknutý obdĺžnik (vid' obrázok B.2-15 - tmavožltý krížik) taktiež **vyextrudujeme** . Bočné hrany vzniknutého kvádra pripojíme k objektu, použijeme funkcie **add edges**  a **weld vertices** .

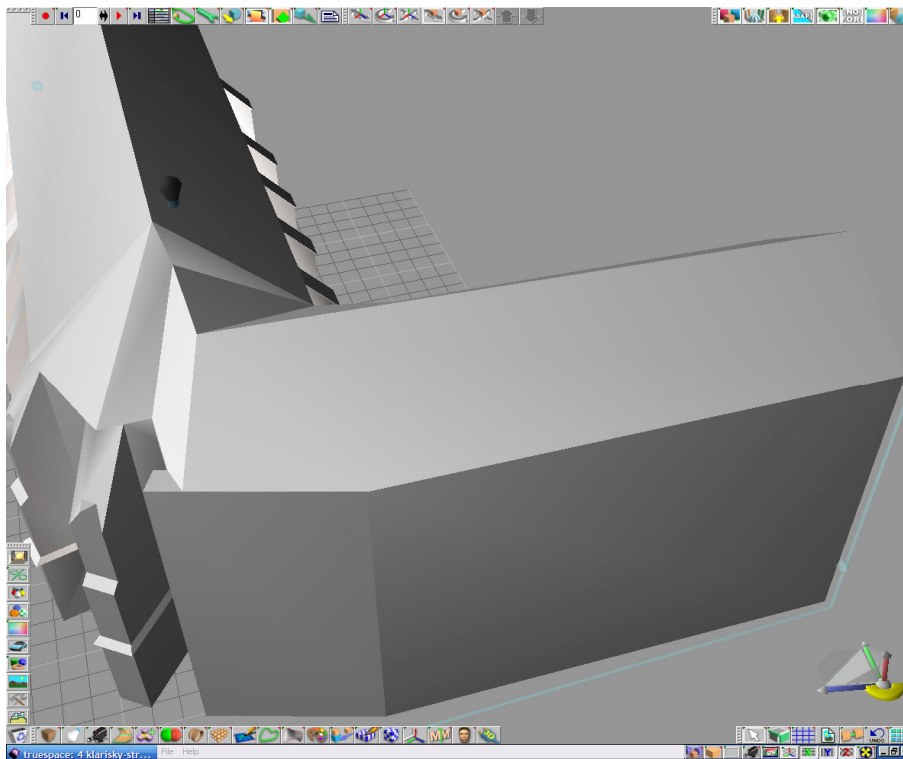
Strecha nad Univerzitnou knižnicou

Postupujeme obdobne ako pri modelovaní prvej strechy. Prvé dve vrchné steny zozadu Univerzitnej knižnice **extrudujeme**  v smere osi z o 3 a spojíme do jednej - **add edges**  a **weld vertices**  (vid' obrázok B.2-16 - zelená farba). Ďalej **spojíme**  pravý vrchný bod strechy nad klariskami s dvoma bodmi na spojnici Univerzitnej knižnice a klarisiek (vid' obrázok B.2-16 - červená farba).







Obrázok B.2-16 : Strecha nad Univerzitnou knižnicou pred dokončením

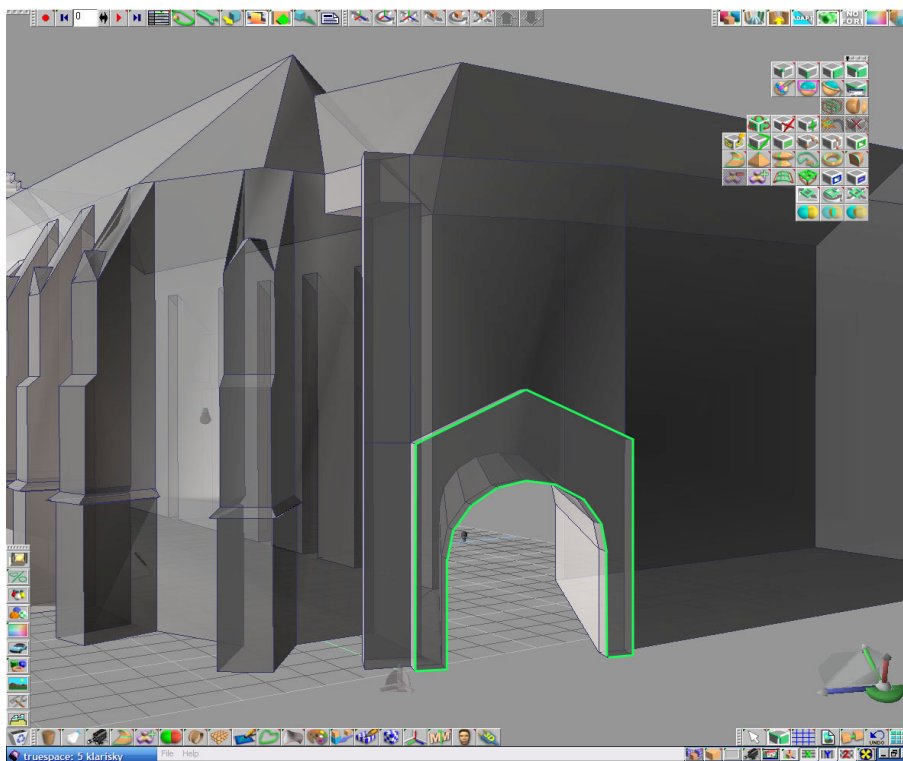
Na dokončenie strechy ešte **pridáme**  dva body na vyznačené miesta (viď obrázok B.2-16 - žltý a bledožltý bod) a **spojíme**  ich. Bledožltý bod **posunieme** mierne v smere osi y ku vnútornej strane objektu.



Obrázok B.2-17 : Dokončená strecha nad Univerzitnou knižnicou




Vstupná brána






Do PP vložíme valec  s počtom bočných hrán 20. **Zrotujeme** ho o 90 stupňov okolo osi x a o 56,7 stupňov okolo osi y. Upravíme jeho rozmery na $x = 4,47$; $y = 6,17$; $z = 11,06$ a **umiestnime** ho do spojnice medzi Klarisky a Univerzitnú knižnicu. Funkciou **object subtraction**  ho vyrežeme a tým urobíme vstupnú bránu do nádvorja. Okolo otvoru z vonkajšej strany urobíme "rám" - **pridáme**  hrany a **vyextrudujeme**  vzniknutú stenu (vid' obrázok B.2-18 - zelená farba).

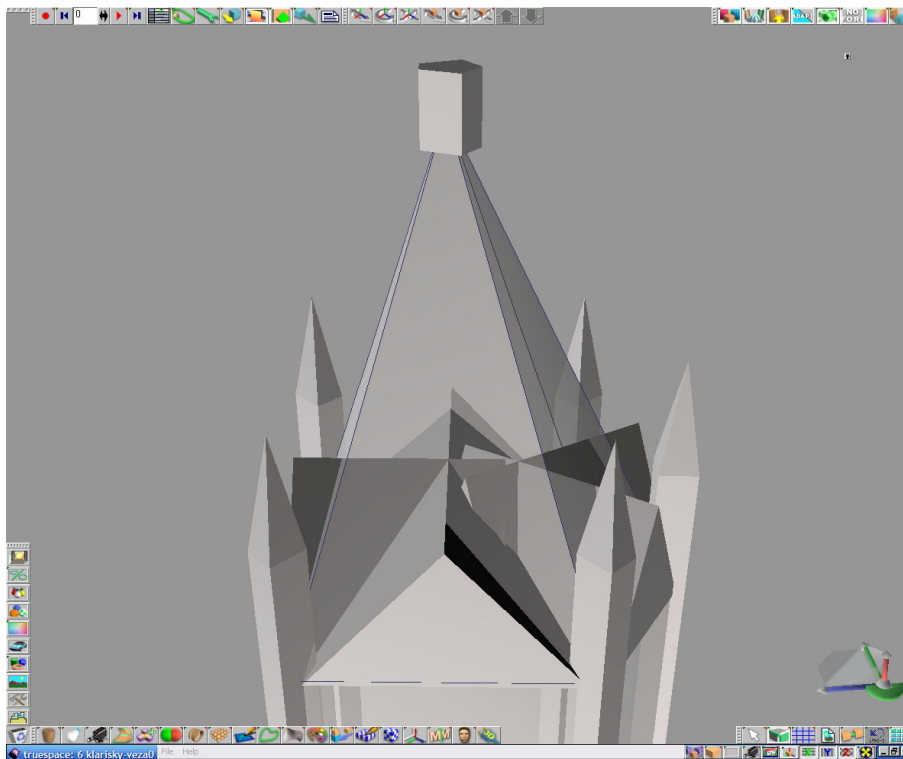


Obrázok B.2-18 : Vstupná brána







Domodelovanie veže

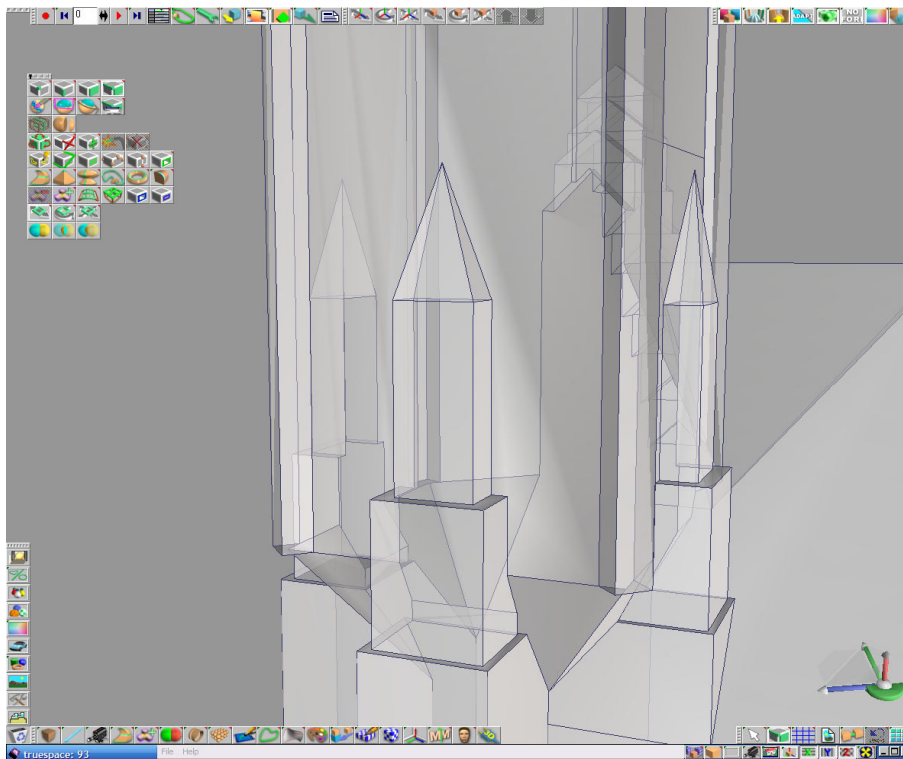
Do pracovnej plochy vložíme **kocku**  a funkciou **weld vertices**  spojíme jej vrchné hrany nachádzajúce sa oproti sebe do jedného bodu, čím vytvoríme striešku. Upravenú kocku **nakopírujeme**  3x a podľa toho ako ju **umiestnime** na veži (**move & rotate**), **upravujeme** jej rozmery - nemeníme výšku = 1,7; ale rozmery podstavy.

Do PP vložíme ďalšiu **kocku**  a **upravíme** jej rozmery na $x = 0,41$; $y = 0,34$; $z = 9,8$. Vrchnú stenu **vyextrudujeme**  a body **spojíme**  do jedného. **nakopírujeme**  ju 5x a **umiestnime** (**move & rotate**) ku bočným hranám okolo veže. Na vrch veže **umiestnime** zmenšený **valec**  s 5 bočnými hranami.



Obrázok B.2-19 : Dokončená strecha nad Univerzitnou knižnicou

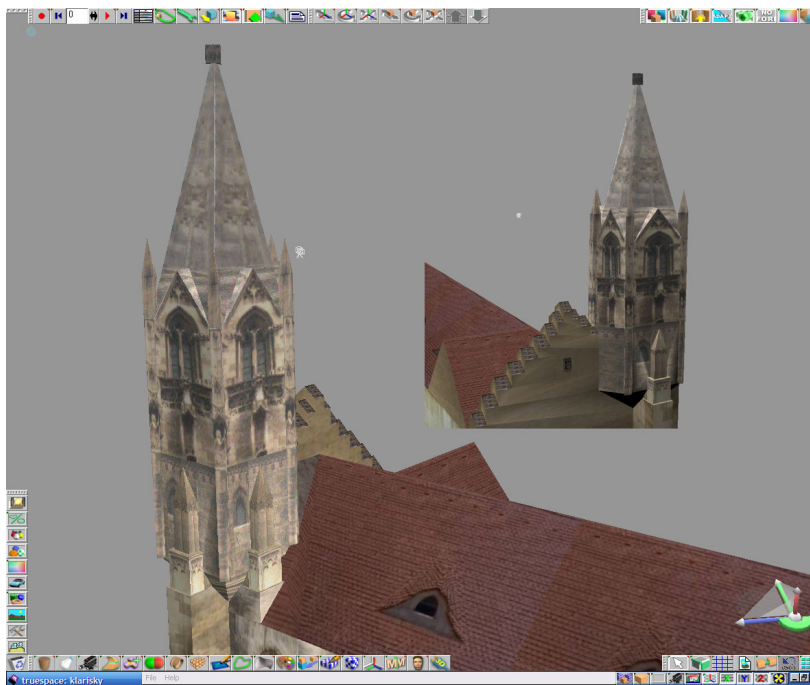
Namodelované objekty **pripojíme** ku veži. Do scény vložíme ďalšiu **kocku**  s rozmermi $x = 1$; $y = 0,8$; $z = 1,3$. Na vrchnej stene kocky funkciou **polygon bevel**  vyrežeme zmenšenú stenu, ktorú **vyextrudujeme**  v smere osi z o 2. Upravenú kocku **nakopírujeme**  3x, **umiestnime** do vyčnievajúcich múrov na spodku veže a **pripojíme**. Ich vrchnú stenu znova **vyextrudujeme**  a body **vertices**  do jedného.



Obrázok B.2-20 : Dokončená strecha nad Univerzitnou knižnicou

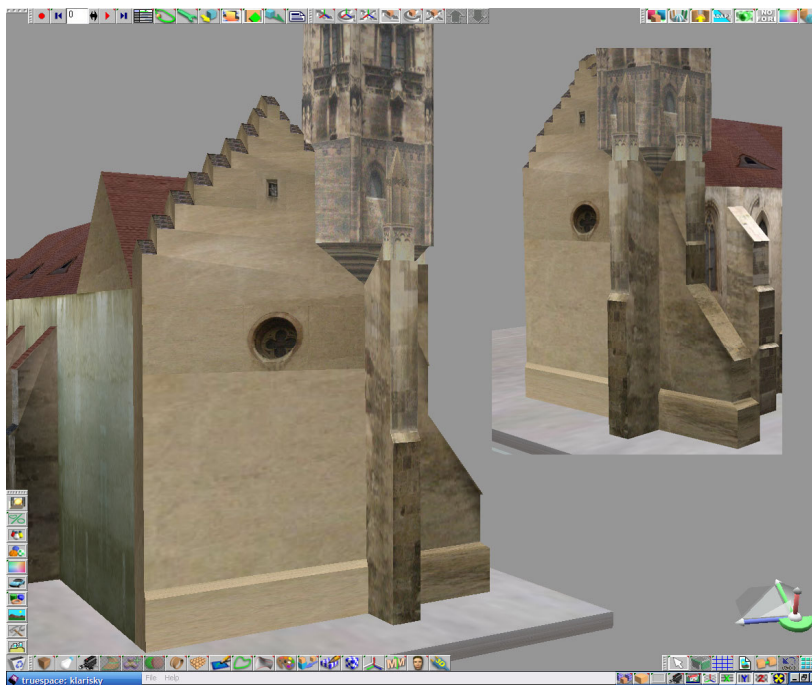
Textúrovanie

Veža



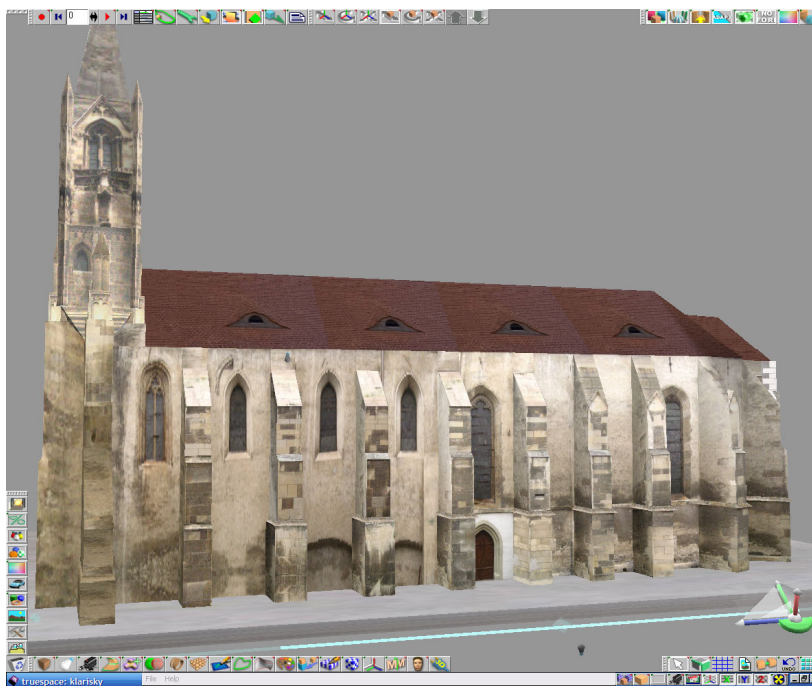
Obrázok B.2-21 : Veža

Zadná stena Klarisiek



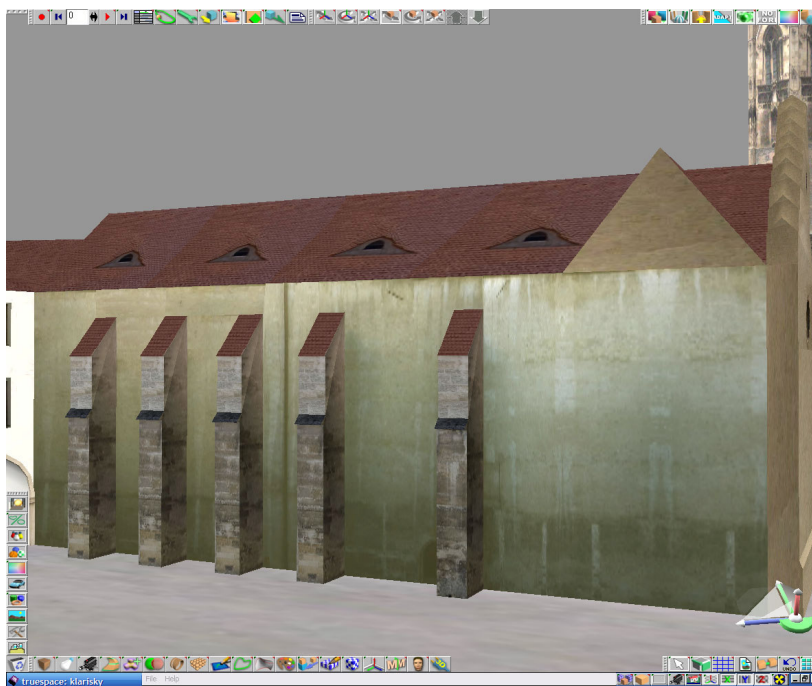
Obrázok B.2-22 : Zadná stena Klarisiek

Vonkajšia stena a strecha Klarisiek



Obrázok B.2-23 : Vonkajšia stena a strecha Klarisiek

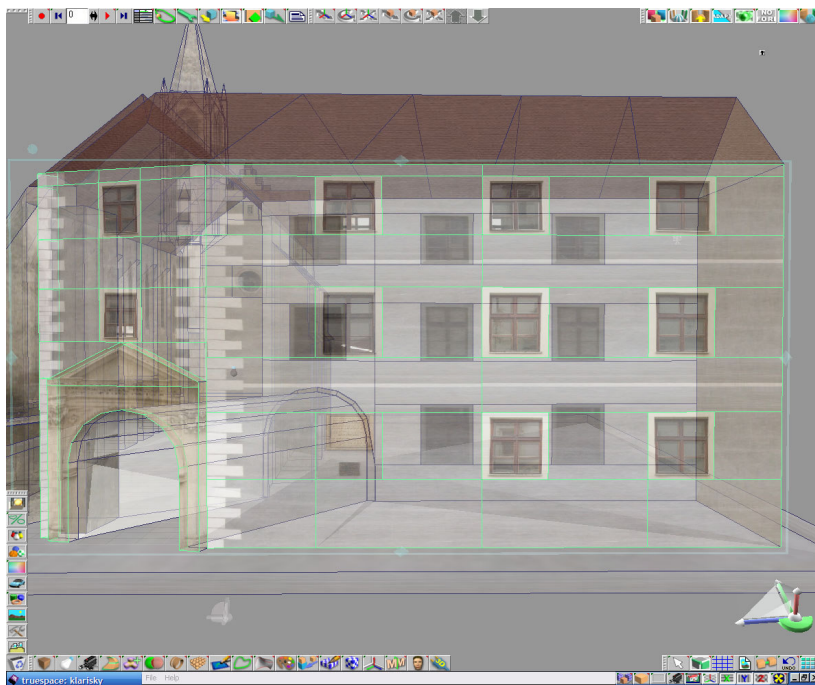
Vnútorňá stena Klarisieľ



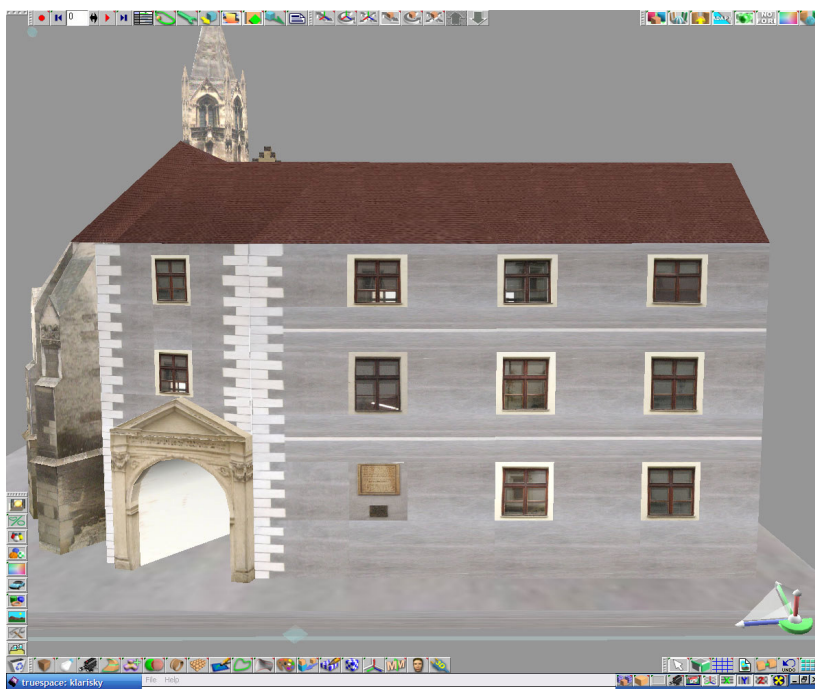
Obrázok B.2-24 : Vnútorňá stena Klarisieľ

Vonkajšia stena, strecha a vstupná brána Univerzitetnej knižnice

Na textúrovanie vonkajšej steny Univerzitetnej knižnice som zvolil "sieť". Dôvodom bolo, že sa nedá získať celý snímok steny, pretože ulička pred týmto objektom má do šírky menej ako 3 metre, jedinou možnosťou je nafotiť stenu po častiach. Pri takomto fotení vieme lepšie zabezpečiť aj kvalitu snímku. V princípe sa dá ešte zo všetkých týchto čiastkových fotiek urobiť v programe pre spracovanie fotografií jedna, ja som sa však rozhodol pre riešenie na základe urobenia siete na textúrovanom objekte.

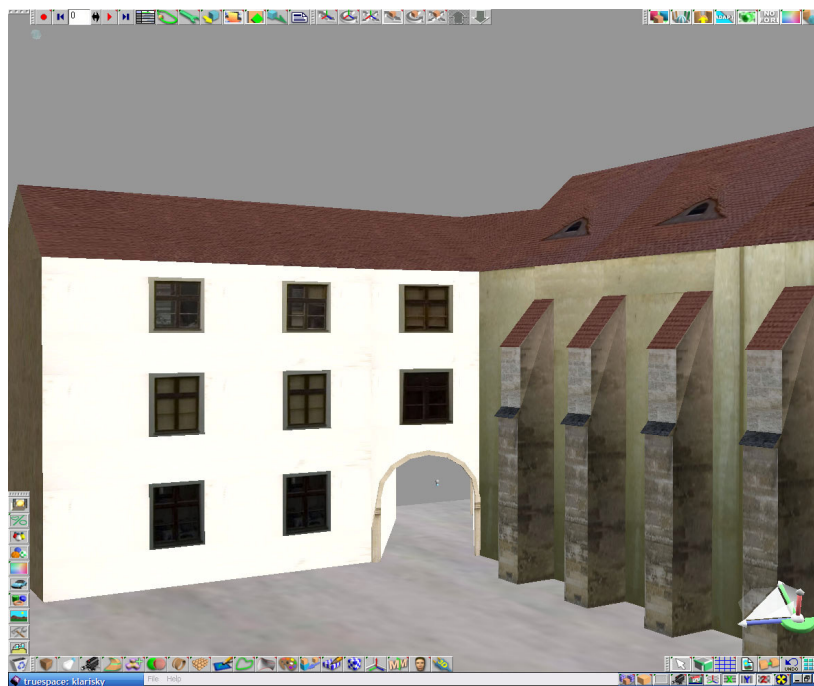


Obrázok B.2-25 : Vonkajšia stena Univerzitnej knižnice - sieť



Obrázok B.2-26 : Vonkajšia stena, strecha a vstupná brána Univerzitnej knižnice

Vnútoraná stena Univerzitetnej knižnice





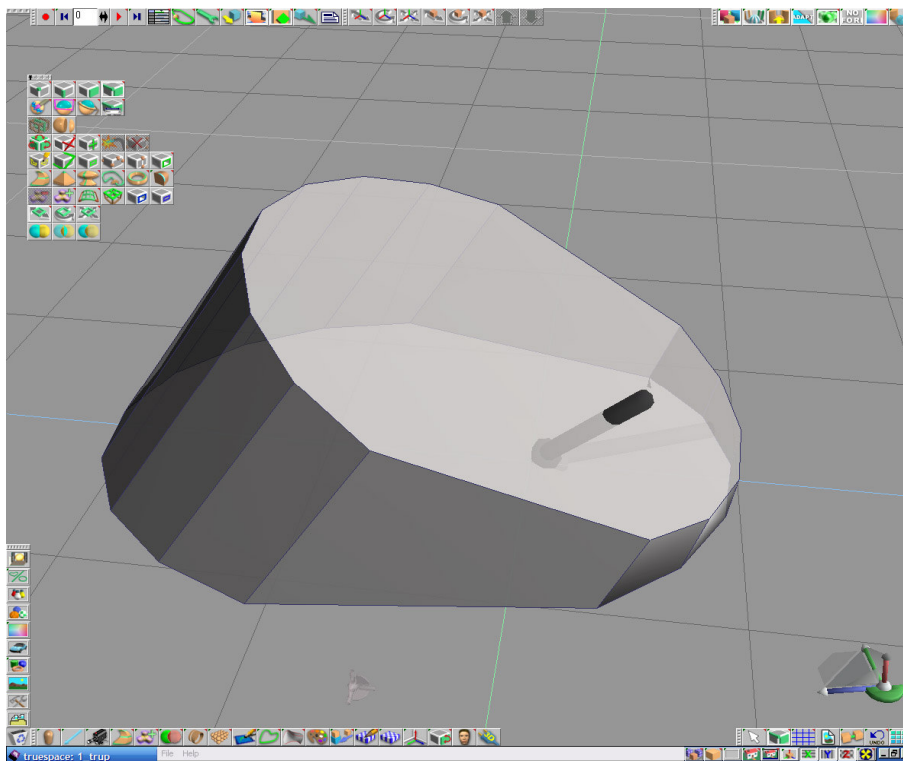
Obrázok B.2-27 : Vnútoraná stena Univerzitetnej knižnice

B.3 Čumil

Trup

Scéna, ktorú ideme modelovať, tvorí postava Čumila, otvor do kanála s poklopom, dlažba okolo kanála a dopravná značka „man in work“. Postupne budeme modelovať najskôr Čumila a následne jeho okolie. Keď si odmyslíme jemnejšie tvary na Čumilovi a budeme brať do úvahy len hlavné časti jeho tela, ľahko nahliadneme, že ho môžeme zjednodušiť na tvar splošteného, po bokoch rozťahnutého a zvrchu šikmo zrezaného valca. Z neho "vyrastá" hlava, ramená a iné časti. Tento valec si teraz namodelujeme.









Do pracovnej plochy vložíme valec , prejdeme do editovacieho módu (RMB na objekt), označíme hornú stenu valca a zrotujeme ju. Ďalej posunieme spodnú stenu valca, aby bočné steny spredu objektu smerovali kolmo na podstavu. Následne ju vyškálujeme tak, aby sme dostali bočné steny zozadu šikmé. Označíme na bokoch valca zľava aj sprava dve zvislé úsečky (najbližšie ku x-ovej osi) a vymažeme  ich (viď obrázok B.3-1).

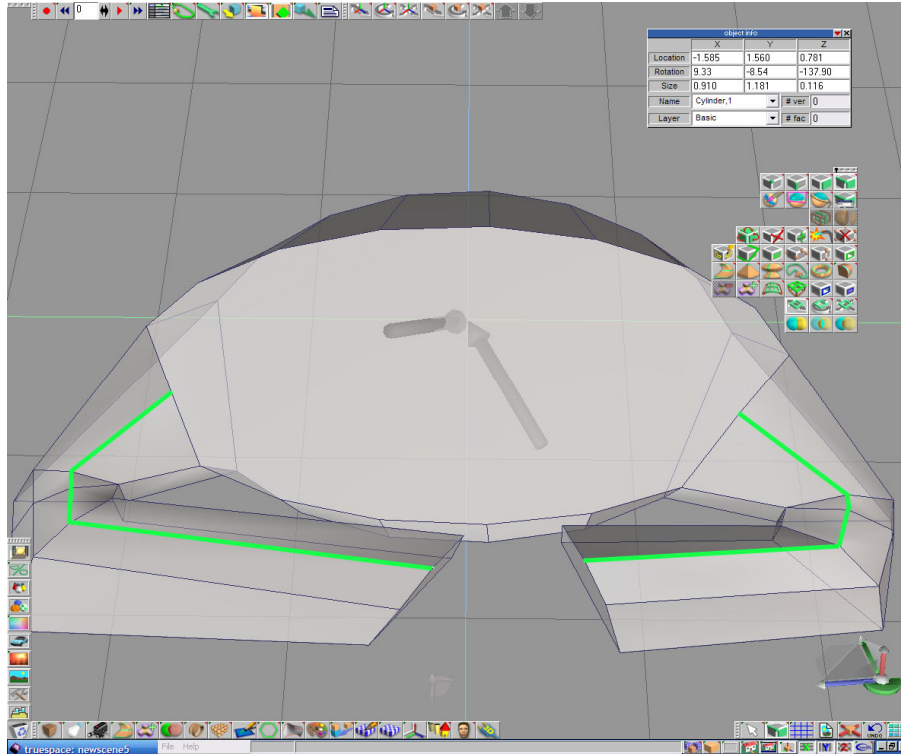


Obrázok B.3-1 : Trup

Ramená

Môžeme si ich predstaviť ako "predĺženie" valca v bokoch, ktoré v určitej vzdialenosti "zalomíme" a ťaháme späť.

Označíme si vytvorené štvoruholníky na bokoch a **extrudujeme** . Ďalej cez **scale** a **rotate** stien upravujeme jednotlivé časti ramien a následne **extrudujeme** . Pri modelovaní kĺbov často potrebujeme z jednej úsečky alebo z viac pospájaných bodov spraviť jeden. Postupujeme nasledovne. **Označíme** body, ktoré chceme spojiť a aplikujeme funkciu **weld vertices** . Iné riešenie sa robí pomocou **pridávania hrán**  a **mazania vrcholov** . Na telese, ktoré chceme orezať, vytvoríme **rezovú rovinu** a body ktoré z nej vytŕčajú **vymažeme** . Po urobení kĺbov aplikujeme ďalšie **povyťahnutia** , **scale** a **rotate** extrudovaných stien. Nakoniec ramená ešte upravíme - do stredu každej hornej steny **pridáme**  úsečku (viď obrázok B.3-2) a bodom **zvýšime z-ovú** súradnicu o 0,050 .














Obrázok B.3-2 : Ramená


Ruky a prsty

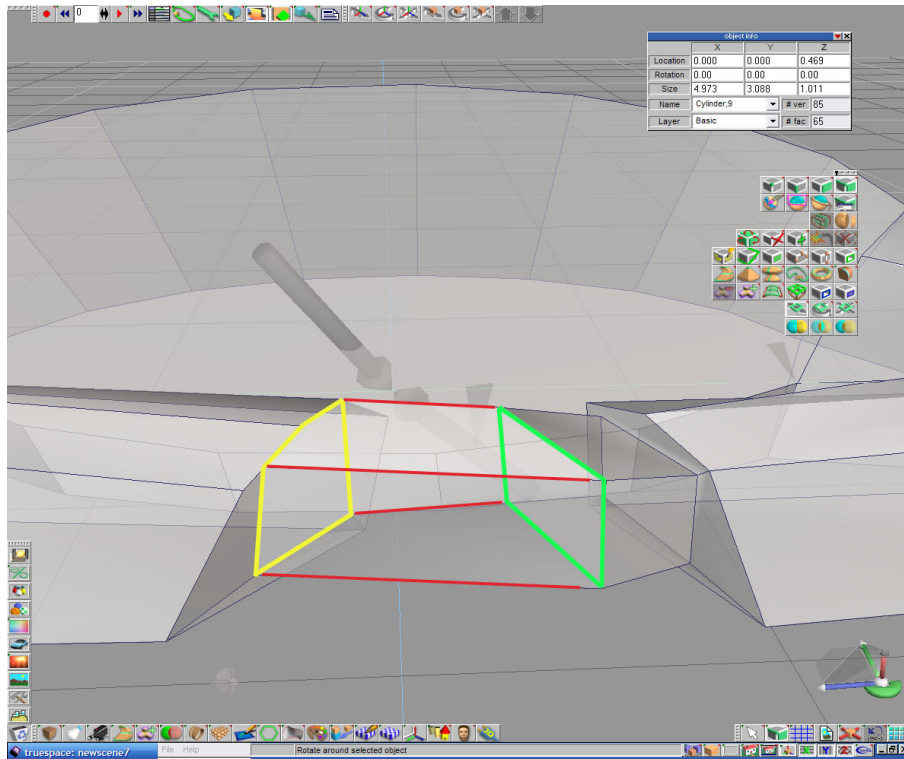
Ruky budeme modelovať z ramien. Keďže sa prekrývajú, budeme ich modelovať ako jeden objekt a odlíšime pravú a ľavú ruku.

Ruky

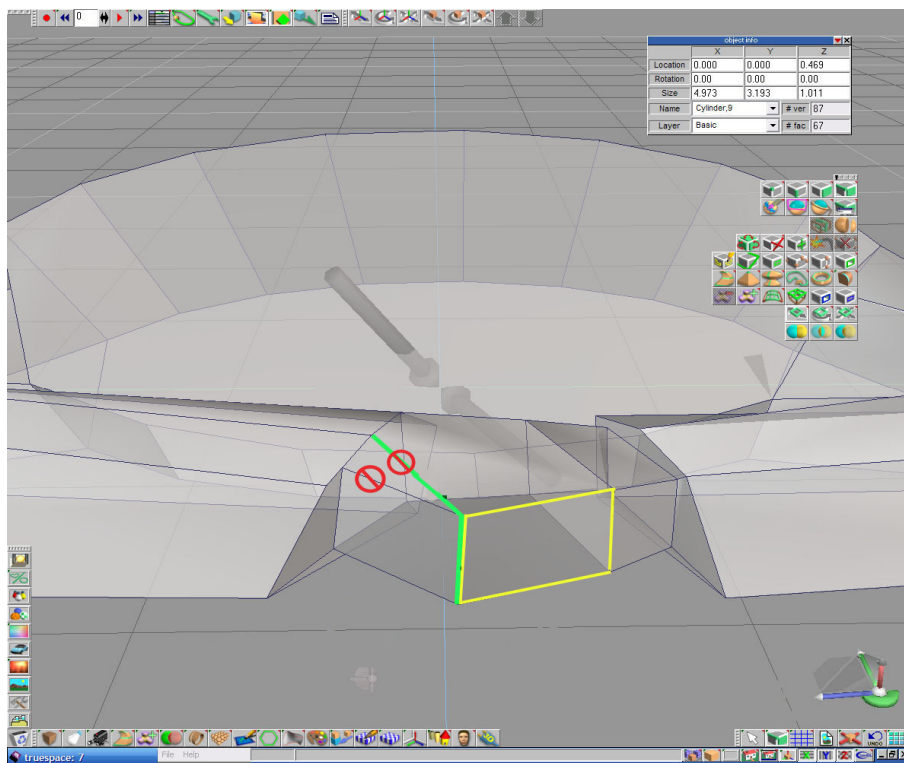
Funkciou **polygon bevel**  do koncov ramien vyrežeme "zmenšenú stenu", z ktorej budeme **extrudovať**  ruky. Jej vrchný bod **vymažeme**  a celý výrez **spojíme**  úsečkami s pôvodnou stenou. Upravíme tvar výrezov na obdĺžniky. Do vnútra obdĺžnika **pridáme**  úsečku, ktorá pôjde od vrchného vonkajšieho vrcholu na spodnú úsečku rovnobežne so zvislou hranou bližšou k trupu. Vznikne nám želaný obdĺžnik a pôvodný dolný vonkajší vrchol **vymažeme** . Výrez na ľavej ruke **extrudujeme**  a **rozšírime** (pozri obrázok B.3-3 - zelená farba).

Na pravej ruke výrezu odstránime vrchné vrcholy, prenesieme ich na ramená. Vytvoríme nové vrcholy na ramene pomocou funkcie **add vertices** . Tie majú rovnaké x a y polohové súradnice ako pôvodné pod nimi, **spojíme**  ich s nimi a pôvodné vrcholy **vymažeme**  (pozri obrázok B.3-3 - žltá farba). Vrcholy povytiahnutého výrezu ľavého ramena **spojíme**  s vrcholmi pravého výrezu. (pozri obrázok B.3-3 - červená farba).



Do hornej a zvislej steny **pridáme**  úsečku (vid' obrázok B.3-4 - zelená farba) a vniknutú zvislú stenu v strede **zrotujeme** a **posunieme dopredu** (vid' obrázok B.3-4 - žltá farba).

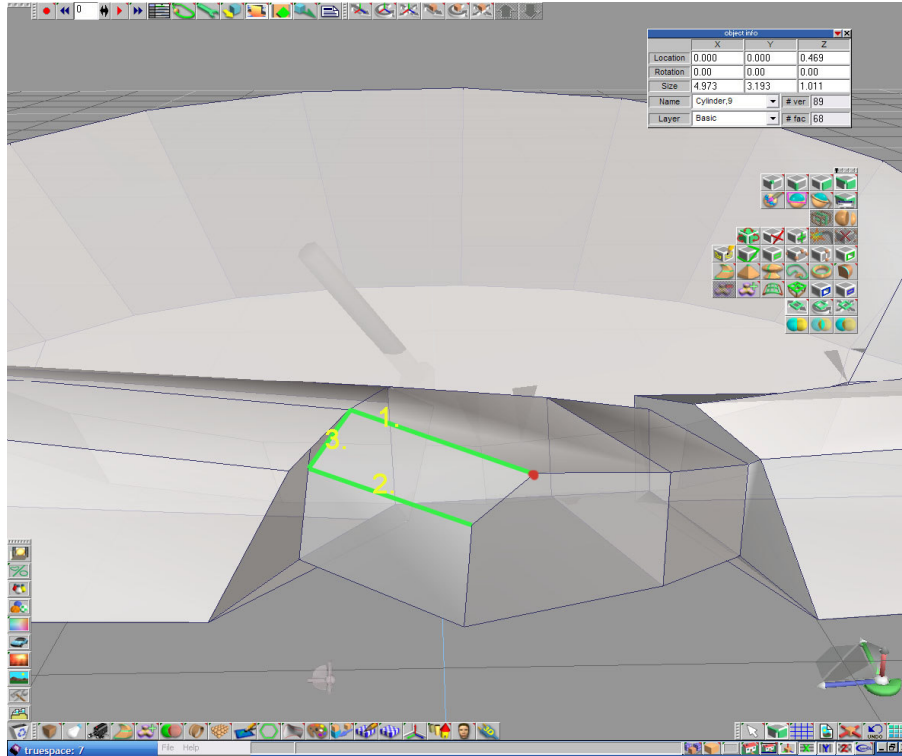


Obrázok B.3-3 : Dlane









Obrázok B.3-4 : Ruky 1

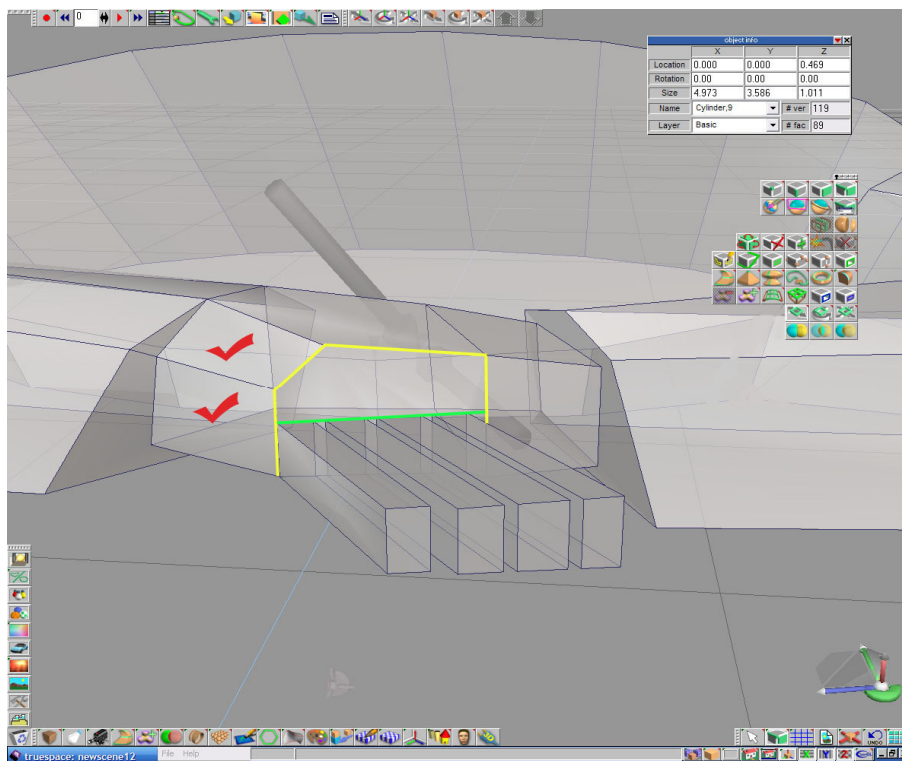
Označené úsečky **vymažeme** , ale najskôr **pridáme**  nové úsečky, v poradí znázornenom na obrázku dole. Pridáme 1 aj 2, potom mažeme a následne pridáme 3. Na vrchnej úsečke strednej zvislej steny nám vznikne bod (viď obrázok B.3-5 - červená bodka), ktorému **zvýšime** mu *z-ovú* súradnicu.



Obrázok B.3-5 : Ruky 2





Prsty na pravej ruke


Do strednej zvislej steny (viď obrázok B.3-6 - žltá farba) **pridáme**  horizontálnu úsečku rovnobežne s dolnou úsečkou. Dolný obdĺžnik rozdelíme pomocou funkcie **quad divide**  a **mazania vrcholov**  na 8 častí - 4 polovičné a 4 celé. **Označíme** stenu, ktorú ideme deliť a klikneme **quad divide**  toľko krát, koľko potrebujeme malých dielov, potom ostatné body a úsečky **vymažeme**  a necháme len deliace úsečky. **Označíme** si obdĺžniky z ktorých pôjdu prsty, sú to celé časti delenia a **extrudujeme** 

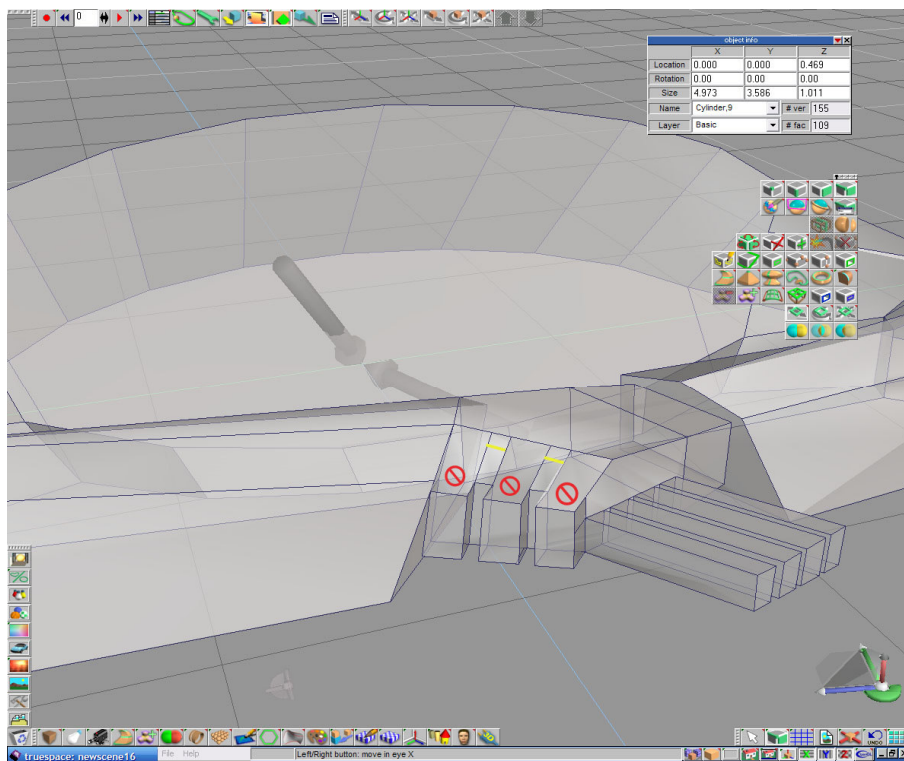


Obrázok B.3-6 : Prsty na pravej ruke

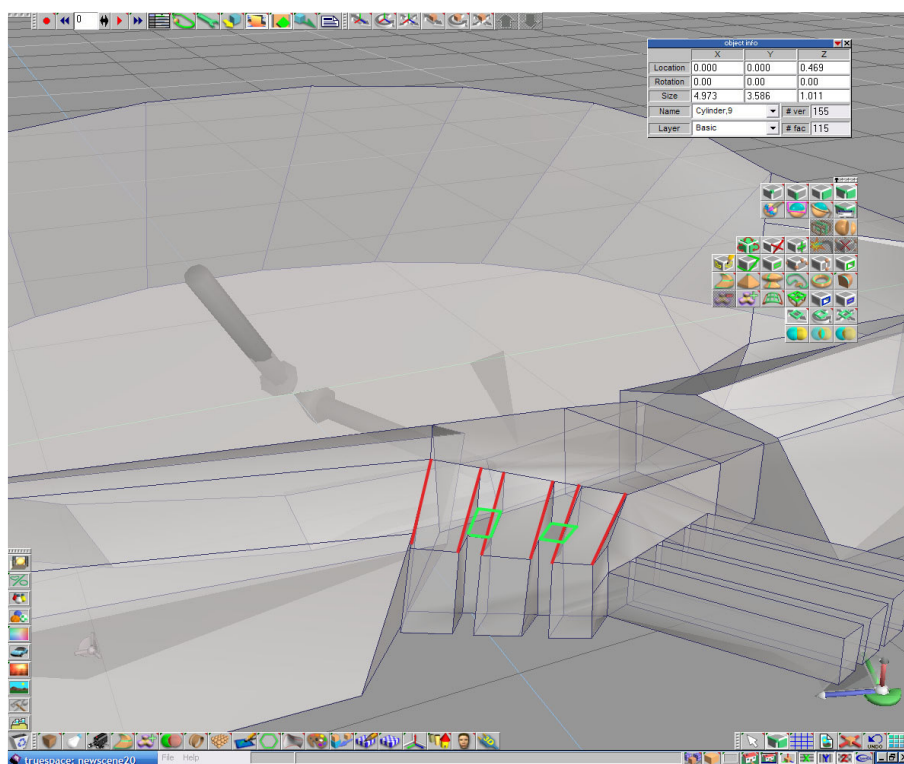
Prsty na ľavej ruke

Odfajknuté steny rozdelíme , podobne ako pri pravej ruke, na 6 častí - 3 celé a 2 polovičné (funkciu aplikujeme 3x). Označíme si 3 celé spodné obdĺžniky a **extrudujeme**  z nich prsty. **Vymažeme**  vrchné úsečky pôvodných obdĺžnikov (viď obrázok B.3-7 – prečiarknutý kruh) a **pridáme**  úsečky do obdĺžnikov medzierok medzi prstami približne do šestiny od vrchu (pozri obrázok B.3-7 - žltá farba).


Vzniknuté dolné obdĺžniky **posunieme** dole a **zrotujeme** (viď obrázok B.3-8 - zelená farba). Následne **pridáme**  úsečky tým vytvoríme prsty na ľavej ruke (viď obrázok B.3-8 – červená farba).



Obrázok B.3-7 : Prsty na ľavej ruke 1




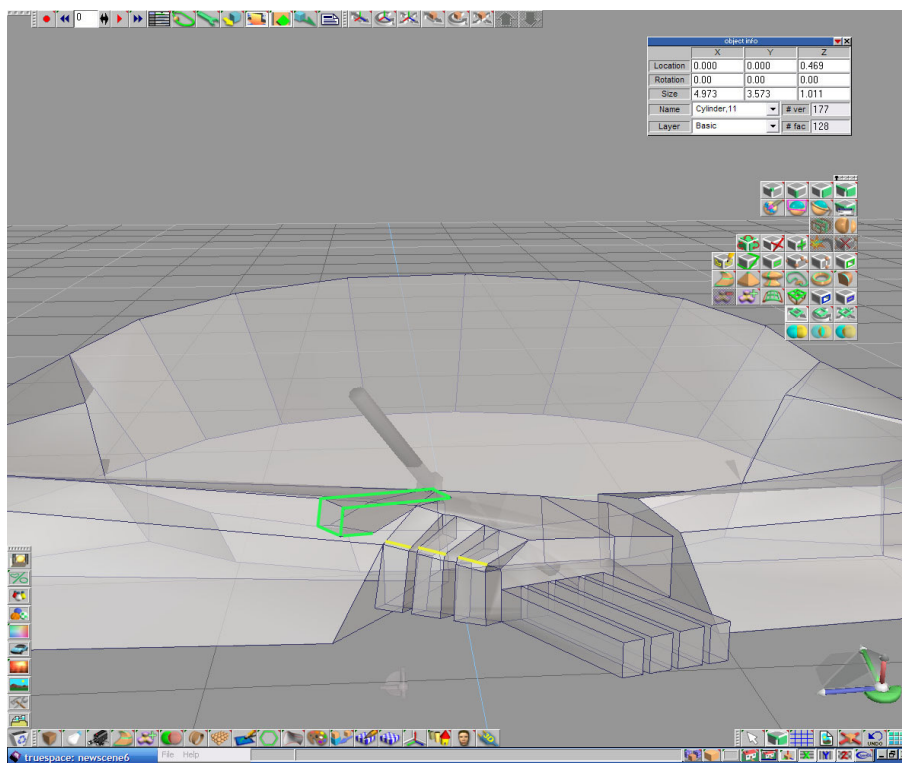
Obrázok B.3-8 : Prsty na ľavej ruke 2

Do obdĺžnikov, ktoré vznikli, **urobíme**  úsečky (pozri obrázok B.3-9 – žltá farba) a vrchné obdĺžniky **posunieme hore a zrotujeme**.

Ukazovák

Vložíme **kocku**  do PP a **upravíme** (vid' obrázok B.3-9 - zelená farba).






Premiestnime ju na ľavú ruku a **spojíme**  s Čumilom.





Obrázok B.3-9 : Ukazovák

Hlava







Hlava má tvar "zaguľateného" valca, ten k Čumilovi pripojíme a následne dorobíme na ňom čiapku a mimiku tváre.

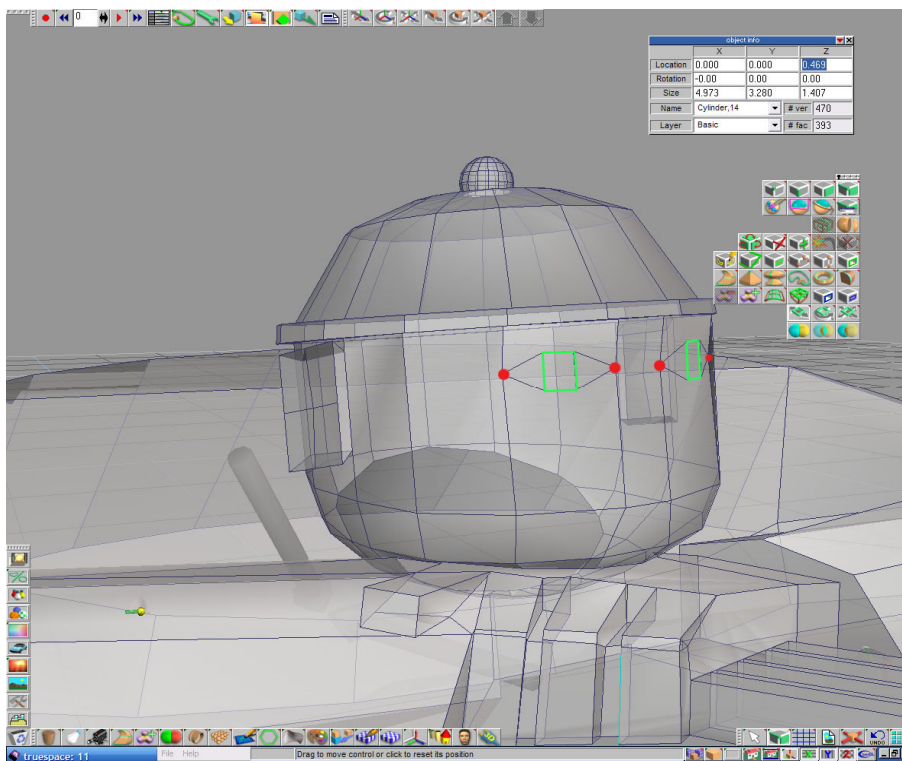
Vložíme **rounded cylinder**  do pracovnej plochy, **upravíme** jeho rozmery (viď obrázok B.3-10) a **premiestnime** na Čumila, aby vytrčal z trupu a zároveň zasahoval do rúk a **pripojíme**  k nemu. Následne valec zakloníme doprava - **vyberieme** všetky body a úsečky hornej podstavy valca a **posunieme** doprava. Obdĺžniky v štvrtej línii zhora zúžime, ich horné úsečky **posunieme** dole. Takto zmenšené steny **označíme** a **extrudujeme** . Ďalej prirobíme na čiapke **brmolec pripojením**  zmenšenej **gule**  na vrch hlavy.

Uši




Na ich vyrobenie nám stačí po bokoch hlavy v obdĺžnikoch nachádzajúcich sa pod čiapkou vyrobiť zmenšené obdĺžniky, **povyťahnuť**  ich a **rozšíriť**. Do týchto obdĺžnikov **pridáme**  dve stredné priečky - vodorovnú a zvislú a ich priesečník **posunieme** ku stredu hlavy.

Oči



Na obrázku dole vidíme aj umiestnenie očí. Vyrobíme ich pomocou **pridávania hrán** , **spájania vrcholov**  a **povyťahnutia** . V obdĺžnikoch, ktoré sme rozšírili na úkor predeľujúceho v strede, vyrobíme menšie **pridaním**  horizontálnych hrán. V ich strede vyrobíme ešte menšie **pridaním**  vertikálnych hrán (viď obrázok B.3-10 - zelená farba) a vonkajšie vertikálne hrany **spojíme**  do jedného bodu (viď obrázok B.3-10 - červená farba).





Obrázok B.3-10 : Umiestnenie očí

Tieto menšie obdĺžniky **uchopíme**  o zvislé hrany veľkých obdĺžnikov (vid' obrázok B.3-11 - zelená farba) a okolo očí **pridáme**  naokolo hrany (vid' obrázok B.3-11 - žltá farba). Vnútorne hrany vsunieme do tváre a malý obdĺžnik vo vnútri **extrudujeme** .

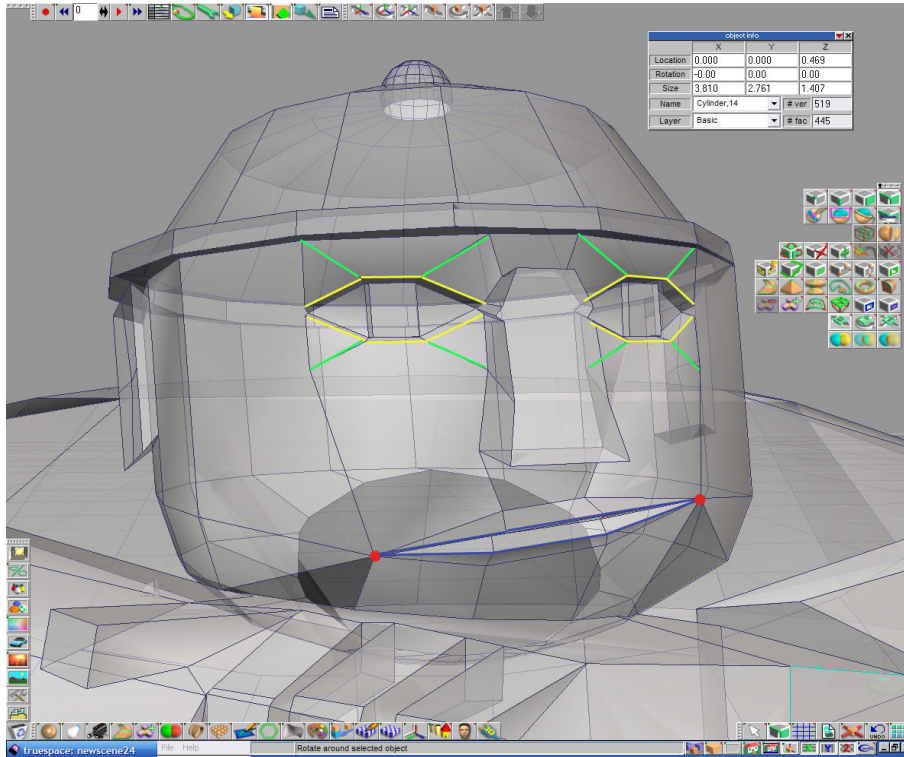
Nos

Vyrobíme ho na strednom obdĺžniku medzi očami **pridaním**  horizontálnych hrán, **zmenšením** vrchnej, **zväčšením** spodnej a **po vytiahnutí**  s následným **rozšírením** (vid' obrázok B.3-11).

Ústa


Spodné obdĺžniky vpredu tváre zúžime - dolné úsečky **posunieme hore**, **vyškálujeme** ich tak aby kopírovali trajektóriu úsmevu, **pridáme**  do nich stredné priečky a vonkajšie hrany zvrchu aj zospodu. Všetky body v oboch kútikoch úst **spojíme**  (vid' obrázok

B.3-11 - červená farba) a **vysunieme** pôvodné hrany (viď obrázok B.3-11 - modrá farba), aby nám vznikli pekné ružové pery.






Obrázok B.3-11 : Hlava

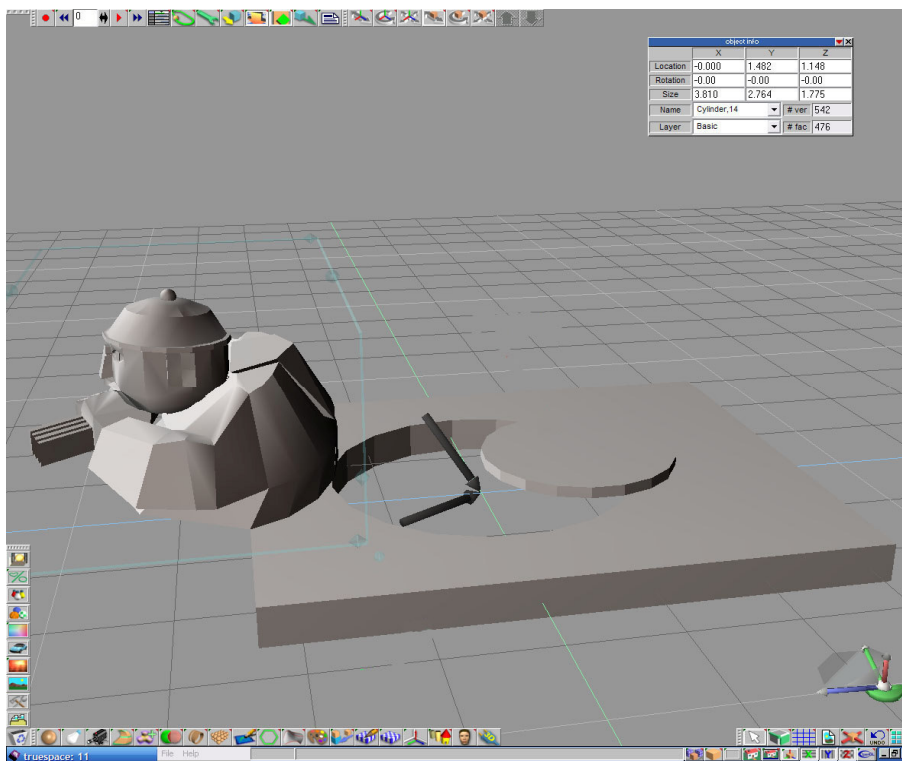
Kapucňa

Na chrbte zozadu, na dvoch stredných obdĺžnikoch a vrchnom, vyrobíme nový obdĺžnik, ktorý **extrudujeme**  a **rozšírime**.

Kanál

Pred modelovaním kanála **extrudujeme**  Čumilovu podstavu, aby sme ho mohli vložiť do kanála a ruky s prstami zarovnáme zospodu s ramenami.

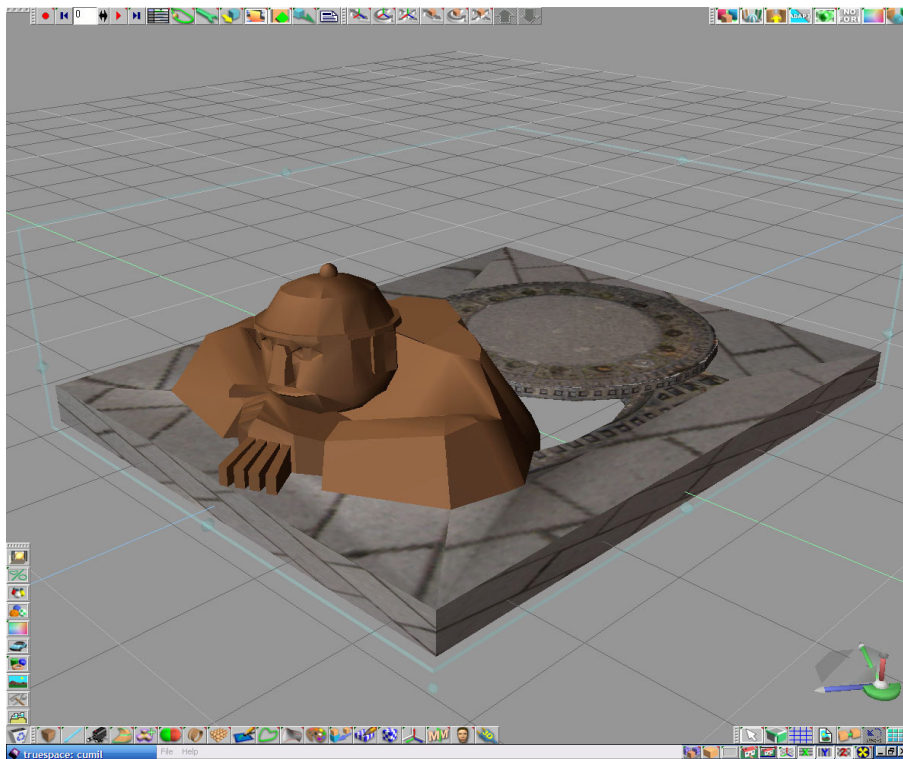
Do scény vložíme **kocku** , **upravíme** jej tvar (viď obrázok B.3-12) a pomocou funkcie **object subtraction**  z nej vyrežeme valcovú dieru. Výroba **poklopu** je jednoduchá, je to vlastne upravený valec .



Obrázok B.3-12 : Scéna

Čumil bez značky



Všetky namodelované objekty "zasadíme" do seba a spojíme ich. Zostáva nám už iba **otextúrovať** náš model. Budeme potrebovať textúru Čumila (použijeme farbu, RGB : 150, 98,59), poklopu a dlažieb. Tie sa dajú vyrobiť v grafickom editore, ja som použil Adobe Photoshop CS2.






Obrázok B.3-13 : Čumil

Čumil s dopravnou značkou

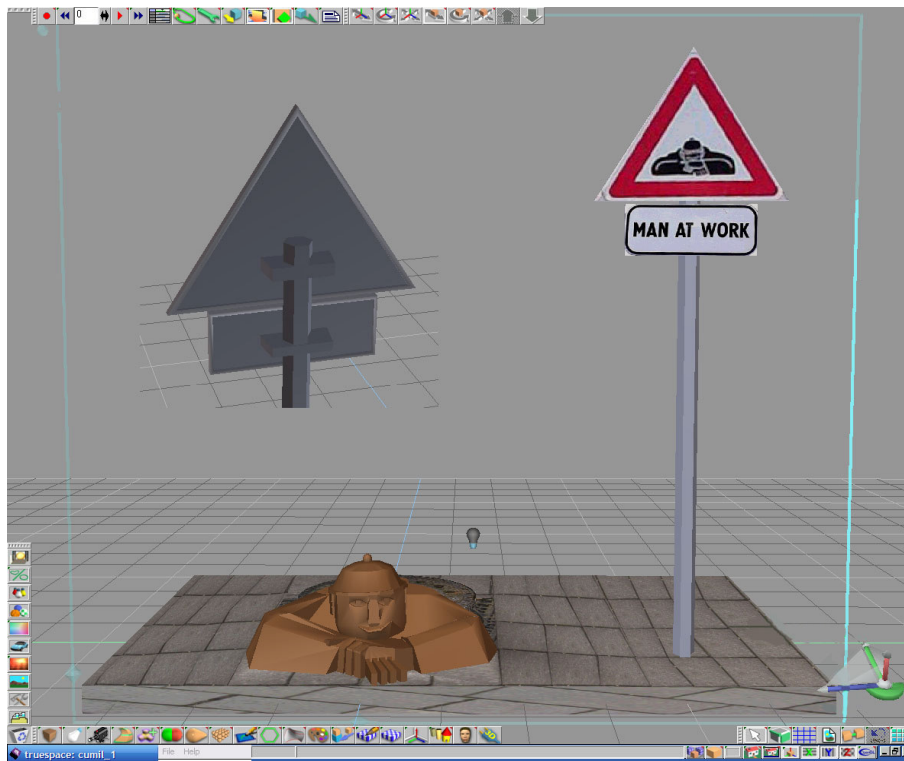
Aby bol náš model kompletný, domodelujeme dopravnú značku, ktorá ku Čumilovi neodmysliteľne patrí. **Rozšírime** podstavu a znovu **otextúrujeme**.

Pri modelovaní budeme používať hierarchiu objektov. Vložením a upravením **valca**  namodelujeme tyč značky a ďalej vkladáme do plochy **kocky** , ktoré upravujeme na potrebný tvar (viď obrázok B.3-14).

Postupne namodelujeme :

- **výstražný trojuholník** - zvrchu sme kocke spojili  úsečky do bodov a zozadu sme prirobili vnútro značky (**polygon bevel**  + **sweep**  dozadu)
- „**man in work**“ - tiež sme prirobali vnútro značky
- **dva úchytné kvádre značiek**



Otextúrujeme ich (tyč sme len ofarbovali, RGB : 169, 175, 199) a objekty spojíme.

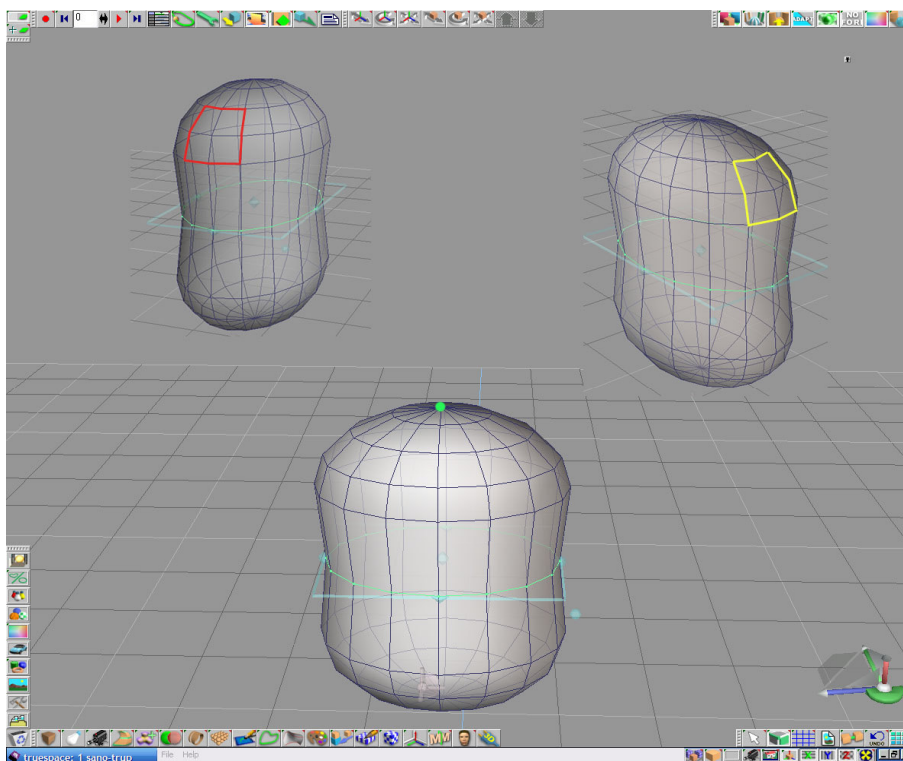


Obrázok B.3-14 : Čumil s dopravnou značkou

B.4 Schöne Näci



Trup

Do pracovnej plochy vložíme **rounded cylinder**  so 16 stenami po bokoch v každej línii a upravíme jeho rozmery na $x = 1,5$; $y = 1$; $z = 2$. Do stredu objektu po celom obvode pridáme  horizontálne hrany a vzniknutý kruh **zúžime**.













Obrázok B.4-1 : Trup

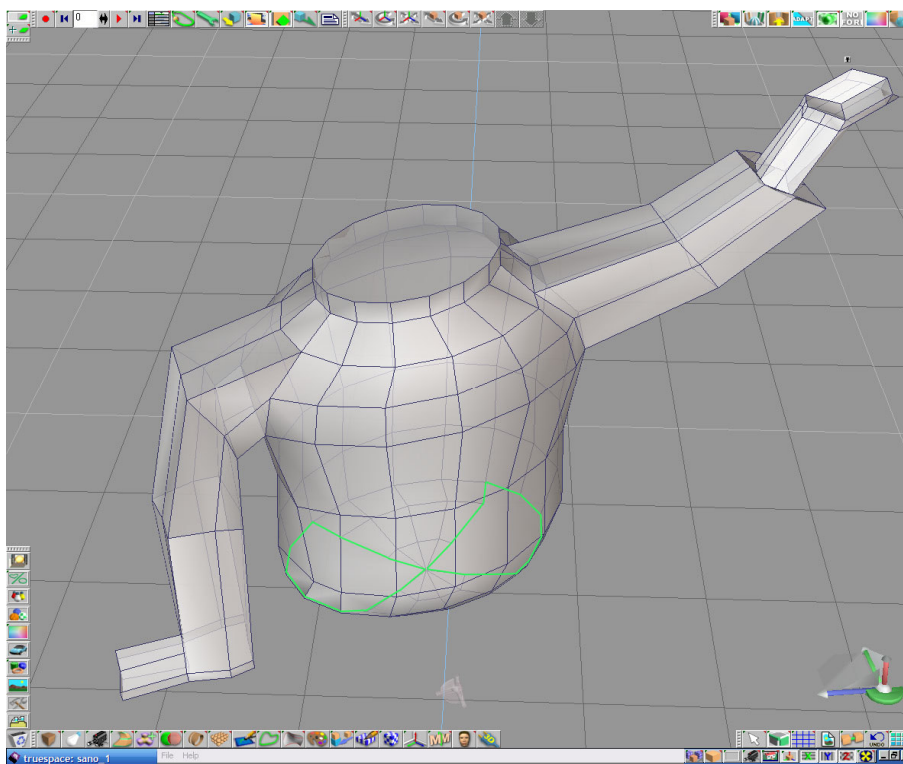
Krk

Na vrchu valca **vymažeme**  vyznačený bod (vid' obrázok B.4-1 - zelená bodka), vzniknutú hornú stenu **rozšírime a vyextrudujeme** .

Ramená

Vymažeme  bod aj hrany vo vyznačenom osemuholníku nachádzajúcom sa na pravej strane valca (vid' obrázok B.4-1 – žltá farba), **označíme ho a 2x vyextrudujeme** .


Pomocou funkcie **polygon bevel**  vyrežeme do vrchnej steny ramena menšiu stenu, z ktorej na 2x **vyextrudujeme**  ruku. Potom si **vyberieme** v hornej línii dve zvislé steny, stenu napravo a vzadu a spolu ich **vyextrudujeme** . (vid' obrázok B.4-1). Podobne vyrobíme rameno aj na ľavej strane. **Vymažeme**  bod aj hrany vo vyznačenom osemuholníku (vid' obrázok B.4-1 – červená farba), **označíme**  ho a 3x **vyextrudujeme** . Pri druhom extrudovaní **posunieme** stenu mierne dopredu a **zrotujeme**. **Označíme** si spodnú stenu ramena, **vyextrudujeme** , potom si **vyberieme** v spodnej línii dve zvislé steny naľavo a spolu ich **vyextrudujeme**  do ľavej ruky (vid' obrázok B.4-2). Celú postavu Schöne Náciho ešte mierne **zrotujeme** doprava.





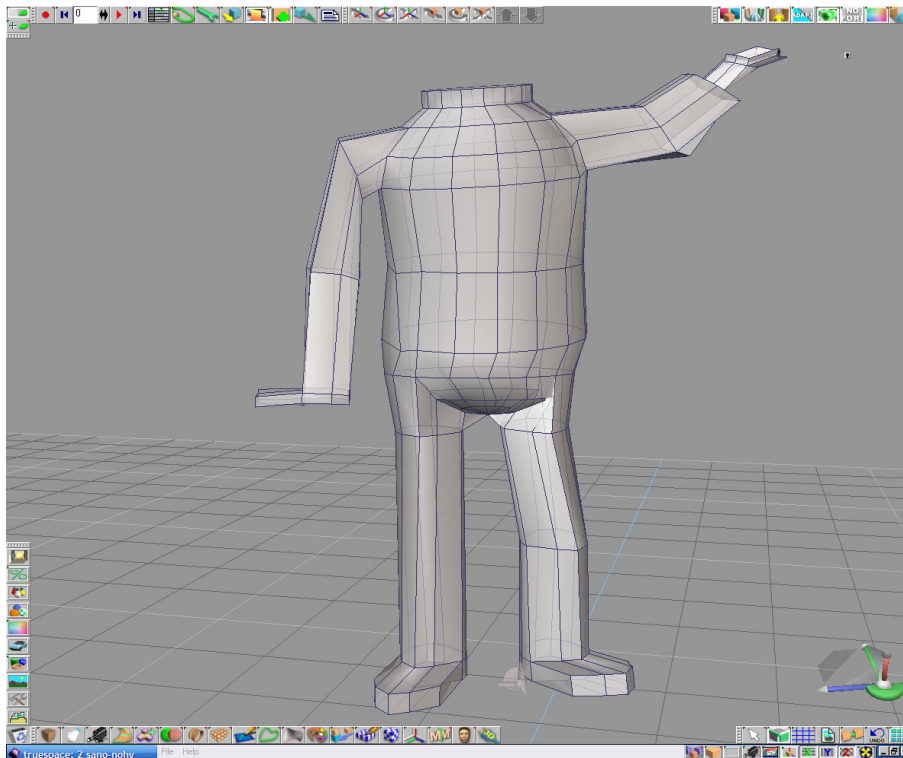
Obrázok B.4-2 : Ramená a ruky

Nohy

Zospodu valca si postupným **mazaním**  hrán vyrobíme dva desať uholníky (vid' obrázok B.4-2 – zelená farba) a **vyextrudujeme**  z nich nohy. Ľavý desať uholník

extrudujeme  2x a pravý 3x - pri druhom extrudovaní pravého **posunieme** vyextrudovanú stenu mierne dopredu, doprava a **zrotujeme**.


Na nohách ešte urobíme **topánky**. Spodnú stenu **vyextrudujeme**  v smere z-ovej osi o 1,13. Potom **vyberieme** predné zvislé steny v poslednej línii a **vyextrudujeme**  ich. Pomocou **scale** a **rotate** ich upravíme na želaný tvar (viď obrázok B.4-3).




Obrázok B.4-3 : Nohy

Prvá úroveň detailov






Podstava

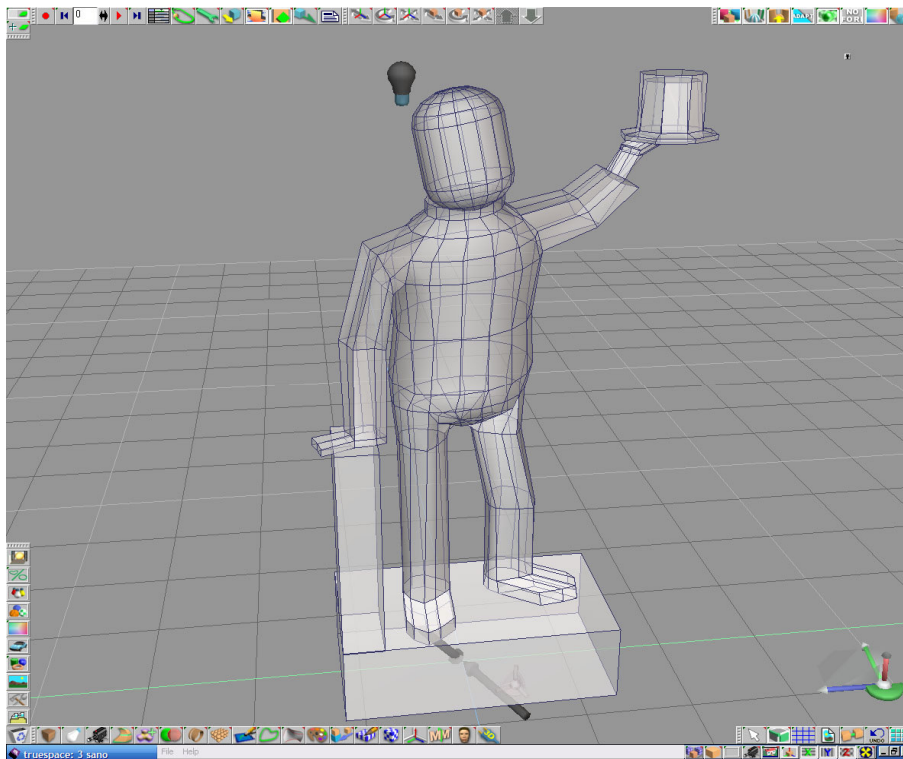
Do pracovnej plochy vložíme dve **kocky** , prvú s rozmermi $x = 2,5$; $y = 1,4$; $z = 0,73$ **umiestnime** pod nohy Schöne Náciho a druhá s rozmermi $x = 0,4$; $y = 0,6$; $z = 1,94$ bude podopierať ľavé rameno.

Hlava

Do PP vložíme **rounded cylinder** , upravíme jeho rozmery na $x = 0,8$; $y = 0,7$; $z = 1$; **zrotujeme** $x = -1,6$; $y = -10,5$; $z = -38,6$ a **umiestnime** na vrch trupu.

Klobúk

Do scény vložíme **valec** , upravíme jeho rozmery na $X : 0,9$; $Y : 0,8$; $Z : 0,04$; **označíme** si vrchnú stenu, **vyextrudujeme** , **zúžime** a **umiestnime** do rovnakej výšky ako pôvodne extrudovaná stena. Danú stenu následne **extrudujeme**  ešte raz v smere Z o $0,5$. **Označíme** si spodnú stenu valca, **vyextrudujeme** , **zúžime** (dôležité je, aby sme stenu zúžili viac ako v predchádzajúcom) a **umiestnime** do rovnakej výšky ako pôvodne extrudovaná stena. Danú stenu následne **extrudujeme**  ešte raz dohora v smere osi z o $0,5$.





Obrázok B.4-4 : Schöne Náciho – 1. úroveň detailov

Všetky doteraz vymodelované časti **spojíme** a daný model **otextúrujeme** (Schöne Náciho len ofarbíme, RGB : 123, 122, 127). Pri textúrovaní všetku ostatné materiálové vlastnosti ostali "default".


Druhá úroveň detailov

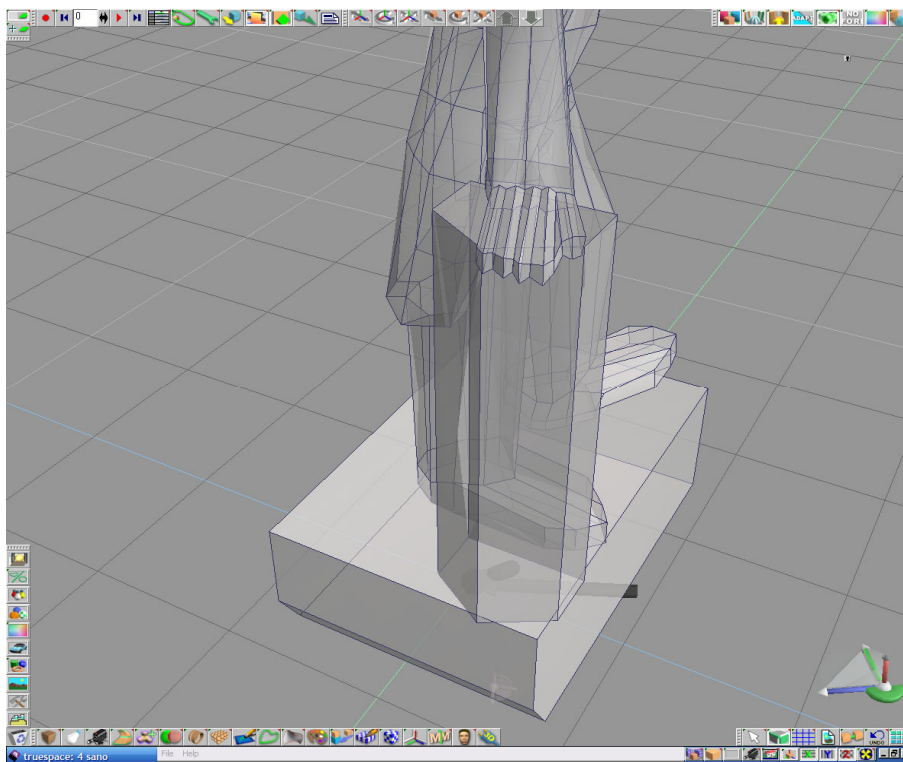
V nasledujúcich dvoch sekciách budeme už len vylepšovať predchádzajúci model.

Podstava


Spodnú podstavu upravíme **pridaním**  horizontálnej hrany do zvislej steny na ľavej strane kocky, **označíme** si ľavú hranu spodnej steny a **posunieme** doprava. Zvislú podstavu pod ľavým ramenom upravíme **pridaním**  vertikálnej hrany do zvislej steny na ľavej strane kocky, **označíme** si postupne ľavú hranu prednej a zadnej steny a **posunieme** doprava.

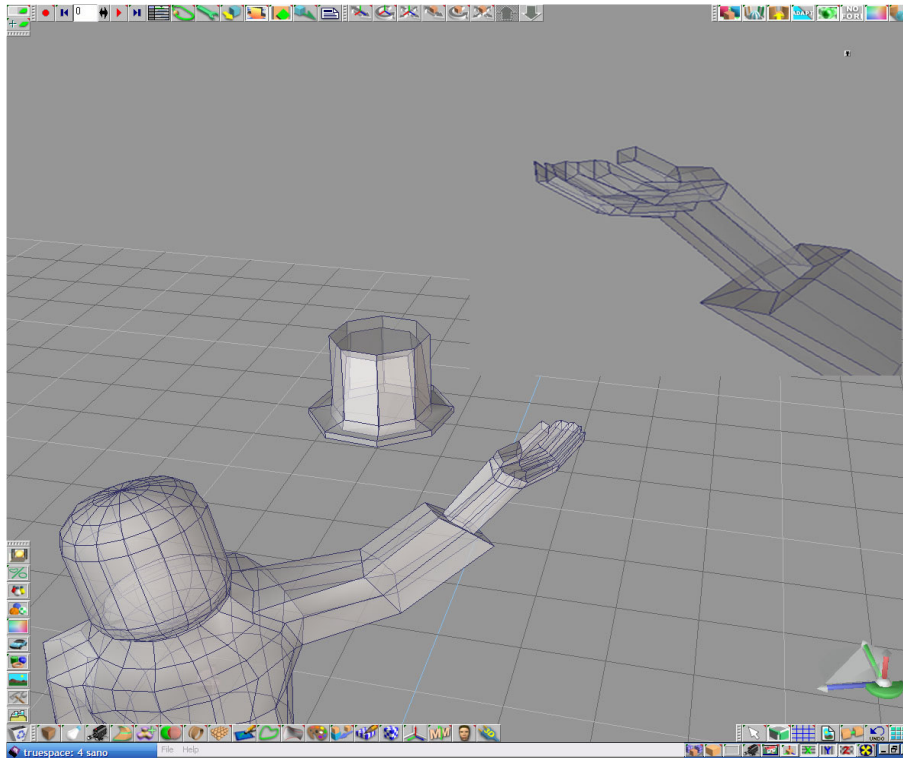
Prsty

Na obrázku B.4-5 môžeme vidieť aj prsty na pravej ruke - dostaneme ich postupným **pridaním**  hrán do dlane a následným **škálovaním** a **posúvaním**.






Obrázok B.4-5 : Upravenie podstavy a prsty na pravej ruke

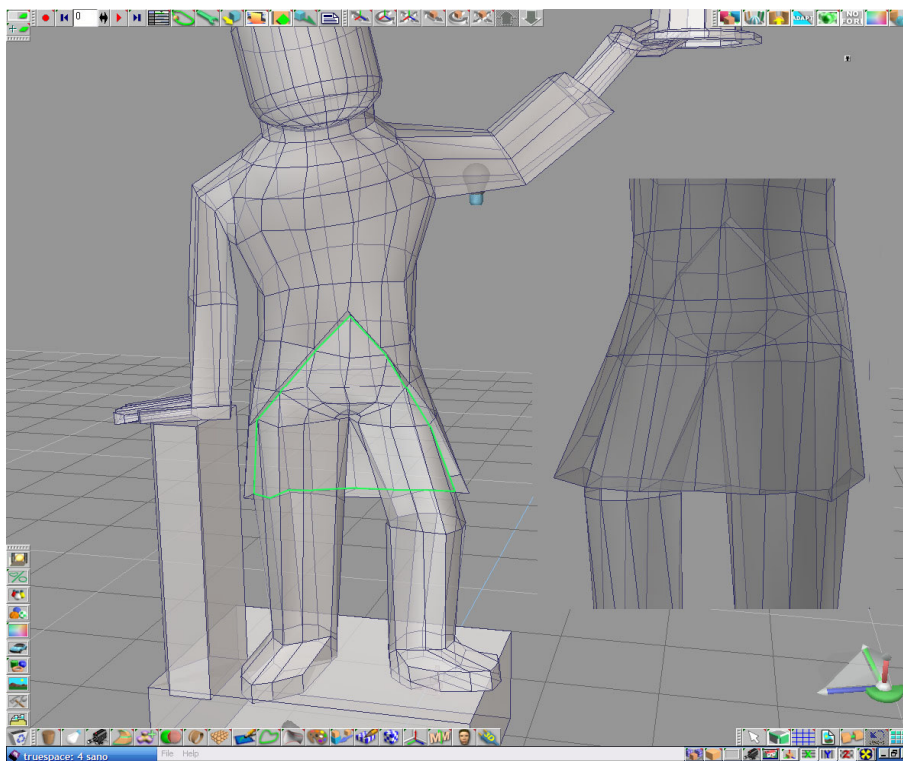
Rovnako postupujeme aj pri prstoch na ľavej ruke - dostaneme ich postupným pridávaním hrán do dlane a následným škálovaním, posúvaním a extrudovaním . Extrudovanie použijeme pri modelovaní palca.



Obrázok B.4-6 : Ľavá ruka

Plášť

Do trupu a nôh pridáme  na vyznačené miesta hrany (viď obrázok B.4-7 - zelená). Vzniknuté steny, pridané úsečky sú ich dolnými hranami, vyextrudujeme  ich a vrchné hrany spojíme pomocou funkcie **weld vertices**  s pôvodnými hornými hranami extrudovaných stien.



Obrázok B.4-7 : Plášť

V editovacom móde (RMB na objekt) ešte niektoré steny a hrany vhodne **vyškálujeme**, **poposúvame** a **zrotujeme**.



Daný model **otextúrujeme** (Schöne Náciho len ofarbíme, RGB : 123, 122, 127). Pri textúrovaní všetku ostatné materiálové vlastnosti ostali "default".

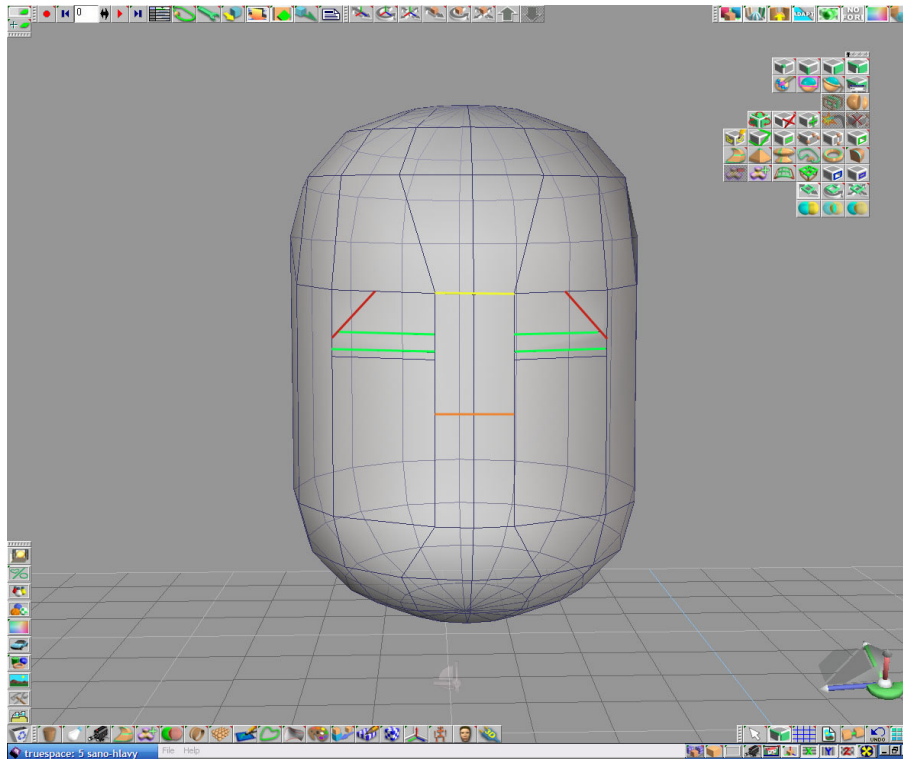
Tretia úroveň detailov

Hlava

Na vloženom zaguľatennom valci si postupne vymodelujeme **oči, nos, ústa a bradu**.

Oči




Na nasledujúcom obrázku môžeme vidieť ich umiestnenie. V obdĺžnikoch v strednej línii, ktoré sme rozšírili na úkor predelujúceho v strede, urobíme dva trojuholníky **pridaním**  šikmých úsečiek (viď obrázok B.4-8 – červená farba). Potom vyrobíme menšie obdĺžniky **pridaním**  horizontálnych hrán (pozri obrázok B.4-8 – zelená farba).

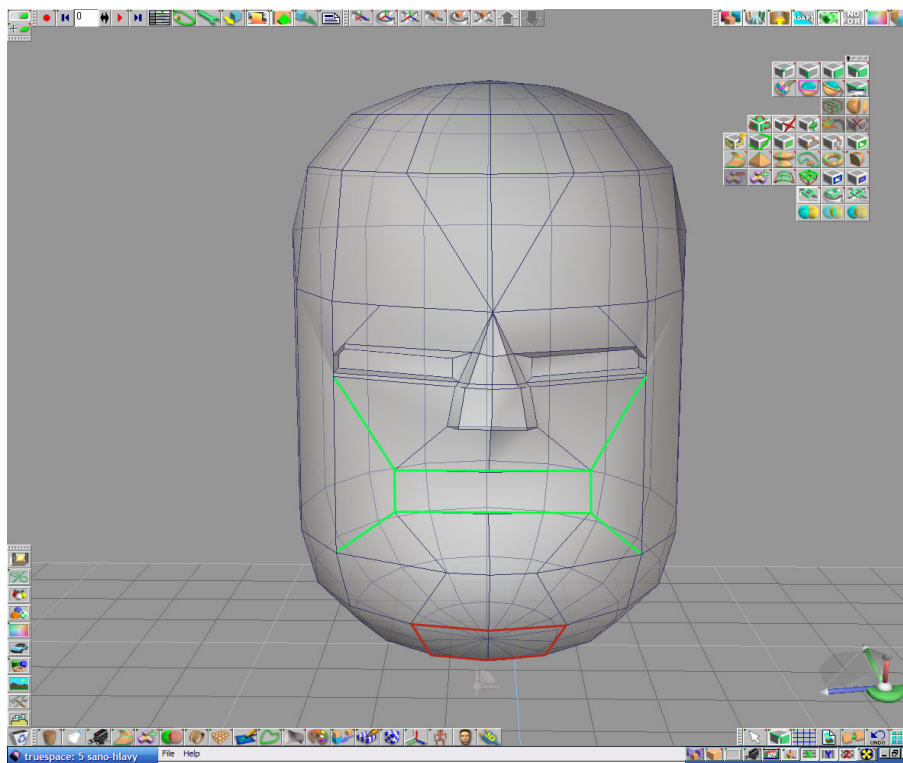


Obrázok B.4-8 : Oči

Tieto menšie obdĺžniky si **označíme, vyextrudujeme** , **zmenšíme a vsunieme** dovnútra valca.



Nos

Hornú horizontálnu úsečku označenú tmavo žltou farbou (vid' obrázok B.4-8) **spojíme**  do jedného bodu. **Pridáme**  dolnú úsečku oranžovej farby (vid' obrázok B.4-8) a vzniknutý trojuholník **vyextrudujeme**  a **zmenšíme** (vid' obrázok B.4-9).



Obrázok B.4-9 : Nos a ústa




Ústa



Pod nos **pridáme**  vyznačené úsečky, **vyškálujeme** ich a **umiestnime** do daného tvaru (vid' obrázok B.4-9 – zelená farba). Do vyznačeného obdĺžnika **pridáme**  strednú priečku a vonkajšie hrany zvrchu aj zospodu. Následne **vysunieme** pôvodné horizontálne hrany a vzniknú nám pery. Steny okolo úst **posunieme** dopredu, dosiahneme tým plnšie **líca**.

Brada

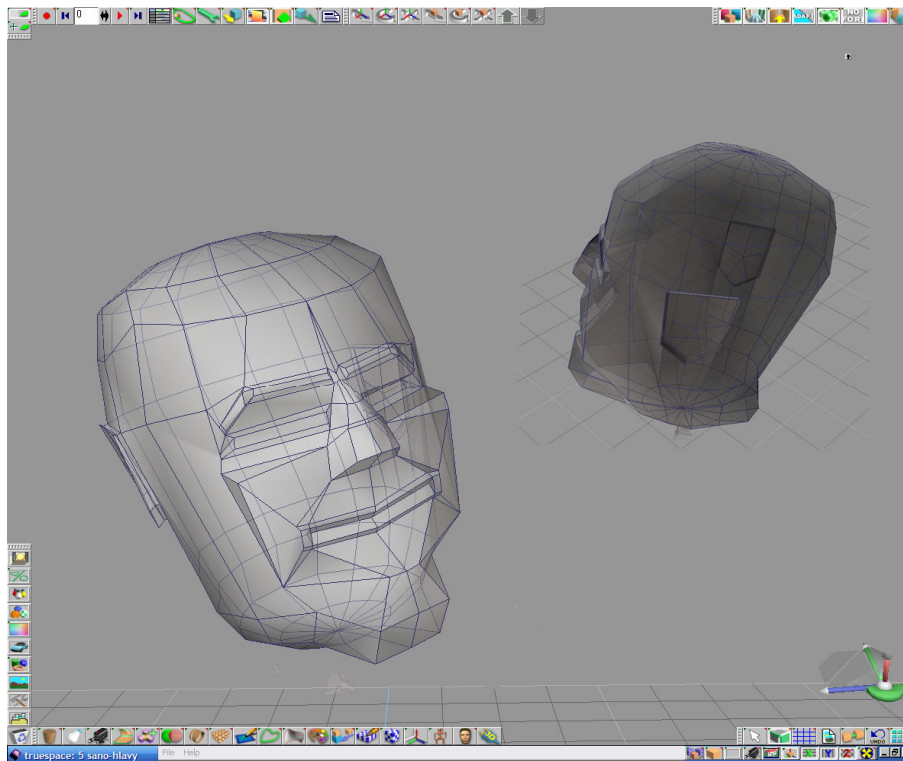
Dostaneme ju **vysunutím** červeného osemuholníka (vid' obrázok B.4-9 – červená farba) dopredu.

Uši

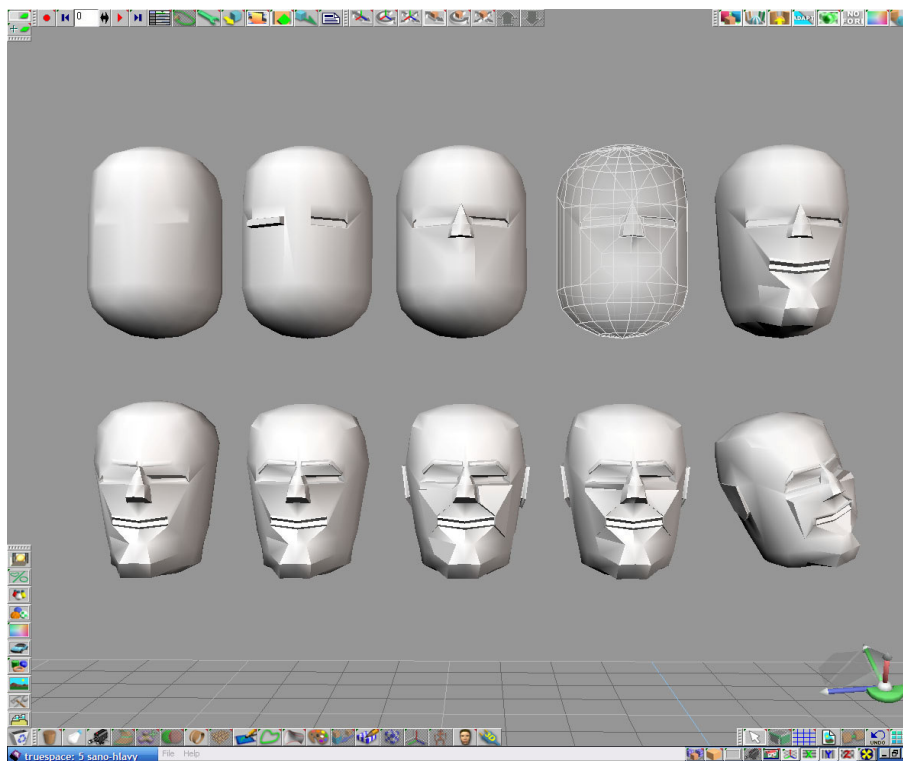
Po bokoch hlavy v obdĺžnikoch nachádzajúcich naľavo a napravo vyrobíme zmenšené obdĺžniky **pridaním**  horizontálnych hrán, ktoré **extrudujeme**  a **rozšírime**. Do týchto obdĺžnikov **pridáme**  ešte stredné priečky.

Nad očami urobíme **obočie** - **pridávaním** , **umiestnením** a **extrudovaním**  vhodných hrán a stien (vid' obrázok B.4-10).

Celú hlavu **zrotujeme** o $x = 5,85$; $y = -14,75$; $z = 30,93$.



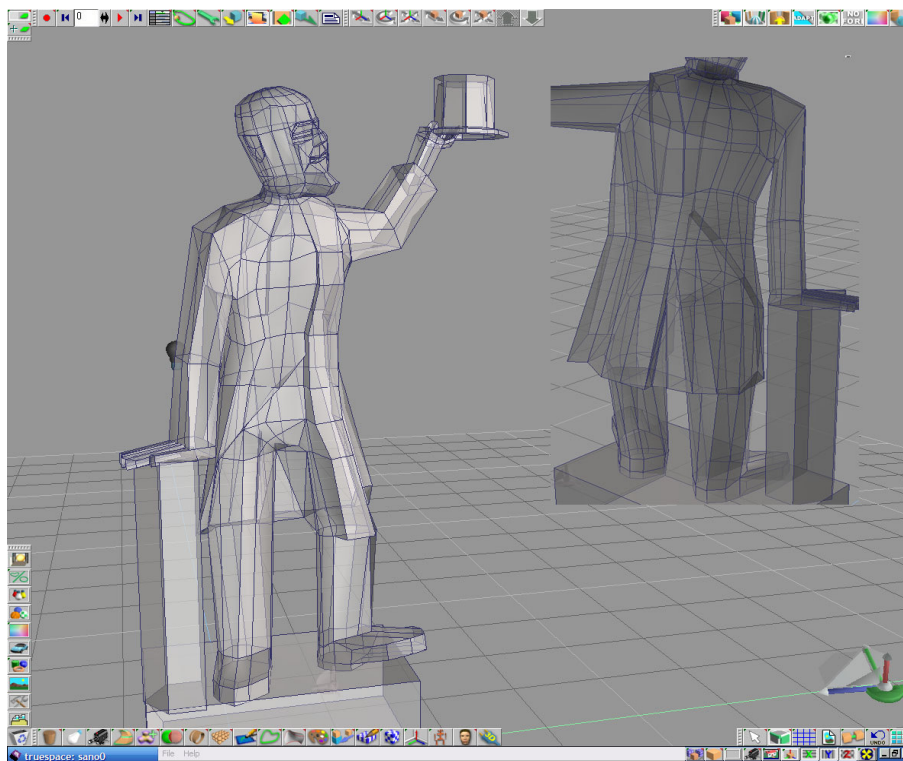
Obrázok B.4-10 : Hlava



Obrázok B.4-11 : Jednotlivé fázy modelovania hlavy

Úpravy na plášti

Pridaním  hrán, posúvaním, škálovaním dorobíme na plášti šál a zadný výstrih (viď obrázok B.4-12).



Obrázok B.4-12 : Schöne Náci – 3. úroveň detailov

Daný model **otextúrujeme** (Schöne Náciho len ofarbíme, RGB : 165, 168, 174). Pri textúrovaní všetku ostatné materiálové vlastnosti ostali "default".

Príloha C

Informácie o objektoch

C.1 Rolandova fontána

Uhorský kráľ Maximilián II ju dal vybudovať v roku 1572 ako verejný zdroj vody. Práce sa zhostil kamenársky majster A. Lutringer, ktorý kráľa zobrazil ako obrneného rytiera na vrchole podstavca. Aj preto nesie fontána jeho meno. Je však známa aj pod menom Rolandova, podľa rytiera Rolanda - symbolu spravodlivosti, s ktorým ju začali dávať do súvislosti v 19. storočí. Poznávacie znamenie - „**najslávnejšia bratislavská fontána**“

zdroj : www.artattack.sk, visit.bratislava.sk



Obrázok C.1-1 : Rolandova fontána spredu

Na nasledujúcom obrázku môžete vidieť, kde sa nachádza Rolandova fontána.



Kredit: EUROSENSE a Peter Borovský.

C.2 Klarisky

Tento kostol i kláštor je postavený na mieste, kde už v prvej polovici 13. storočia stál kláštor cisterciánok. Kláštor bol spustošený a mníšky ho opustili. Krátko po udelení mestských privilégií Bratislave (r. 1291) prichádzajú na pozvanie Ondreja III. do mesta klarisky. Gotický jednoloďový kostol je z prvej polovice 14. storočia. Päťboká hranolová veža bola postavená začiatkom 15. storočia. Z poslednej rokokovej úpravy okolo r. 1760 sa zachoval hlavný oltár s viacerými plastikami, bočné oltáre a kazateľnica. Po zrušení rehole klarisiek v roku 1782 sa kostol stal sídlom právnickej akadémie a gymnázia. V druhej polovici 20. storočia bol upravený na koncertnú sieň, sídli tu časť Univerzitnej knižnice.

zdroj : www.radaeuropy.sk



Obrázok C2-1 : Klarisky –
vonkajšia strana



Obrázok C2-2 : Klarisky – veža



Obrázok C2-3 : Univerzitná knižnica



Obrázok C2-4 : Klarisky – nádvorie.
kredit : Peter Borovský

Na nasledujúcom obrázku môžete vidieť, kde sa nachádzajú Klarisky.



Kredit : EUROSENSE a Peter Borovský.

C.3 Čumil

Čumil je bronzová postavička, ktorá sa nachádza na križovatke ulíc Rybárska brána, Panská a Laurinská. Vznikla dňa 26. júla 1997 a jej autorom je Viktor Hulík. Poznávacie znamenie – „najčastejšie fotografovaný Bratislavčan“.

zdroj : mesto.sk



Obrázok C3-1 : Čumil – pohľad zvrchu

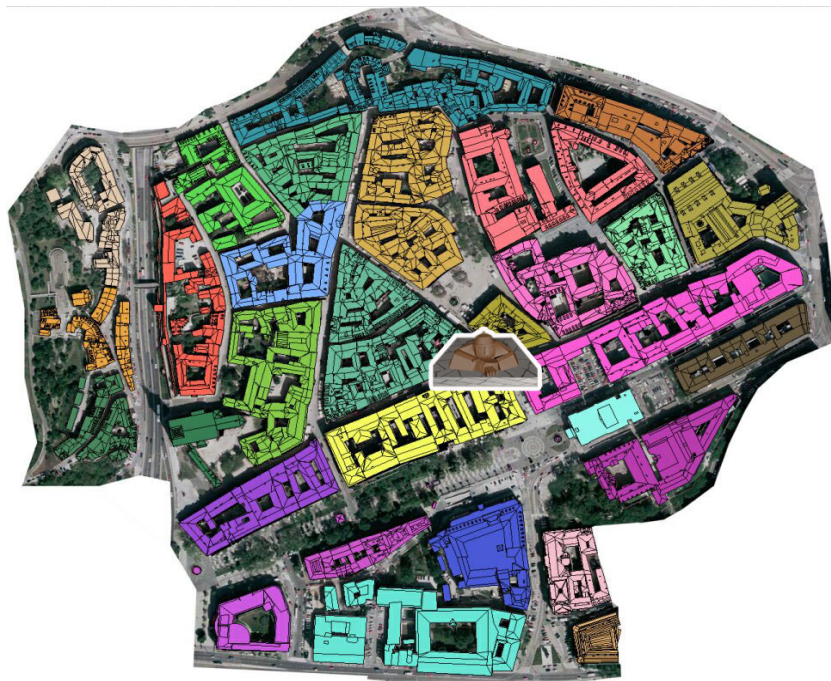


Obrázok C3-2 : Čumil – pohľad spredu



Obrázok C3-3 : Čumil – pohľad zozadu

Na nasledujúcom obrázku môžete vidieť, kde sa nachádza Čumil.



Kredit: EUROSENSE a Peter Borovský.

C.4 Schöne Náci

Narodil sa ako Ignác Lamar, ale väčšina Bratislavčanov ho pozná ako Schöne Náciho. Všetkých víta pri vstupne na legendárne Korzo, po ktorom chodil dennodenne celý svoj život. Jeho autorom je akademicky sochár Juraj Meliš. Poznávacie znamenie – „**legendárna postava starej Bratislavy**“.

zdroj : www.bkis.sk

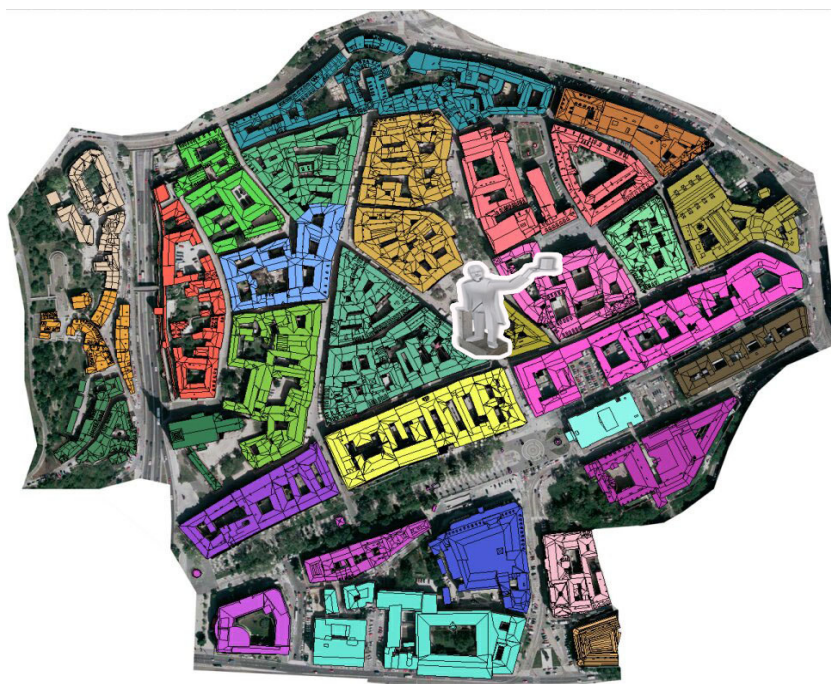


Obrázok C4-1 : Schöne Náci



Obrázok C4-2 : Schöne Náci - hlava

Na nasledujúcom obrázku môžete vidieť, kde sa nachádza Schöne Náci.



Kredit: EUROSENSE a Peter Borovský.