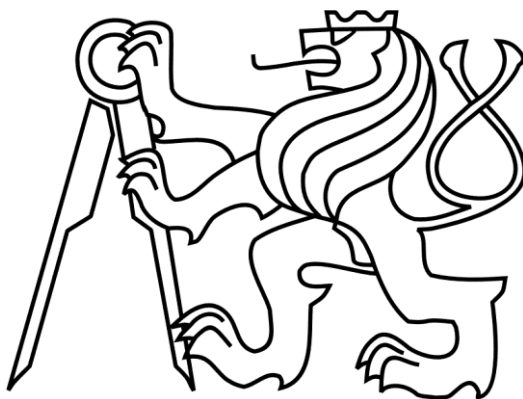


České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce



Bakalářská práce

## **Pravidla znovupoužitelnosti 3D modelů**

Michal Majerník

Vedoucí práce: Ing. David Sedláček

Studijní program: Softwarové technologie a management

Obor: Web a multimedia

26.5.2011



**Prohlášení:**

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 26. 5. 2011



## **Abstract**

The goal of this project is to create rules, which will lead to production of reused model for 4 kinds of reusability. In this work I will deal with formats, which are appropriate for saving a model and secure its reusability in the future. At the end I will test collected data from diploma works and in a case that all of them will not fulfill these rules I will create a schematic model, which I will also test.

## **Abstrakt**

Cieľom tohto projektu je vytvorenie pravidiel, ktoré povedú k tvorbe znovu použiteľného modelu pre 4 druhy použiteľnosti. V práci sa budem zaoberať formátmi, ktoré sú vhodné na uchovanie modelu a tým zabezpečiť jeho znovu použiteľnosť aj v budúcnosti. V závere otestujem zozbierané dáta z diplomových prác podľa vytvorených pravidiel a v prípade, že nebude ani jeden s modelov splňovať tieto pravidlá vytvorím názorný model, ktorý taktiež otestujem.



# Obsah

1	Úvod .....	1
2	Pravidlá znovu použiteľnosti 3D modelov .....	2
2.1	3D formáty.....	2
2.2	Zásady tvorby scény .....	3
3	Rendering .....	4
3.1	Vytvorenie náčrtov a rozdelenie práce .....	4
3.2	Zvolenie správneho programu .....	4
3.3	Vymodelovanie scény.....	5
3.3.1	Techniky modelovania:.....	5
3.4	Nasvietenie scény .....	10
3.5	Nastavenie rendru .....	11
3.6	UVW mapping, normálové mapy a tvorba materiálov .....	11
3.6.1	UVW.....	12
3.6.2	UNWRAP.....	13
3.7	Rendering .....	14
4	Animácia.....	15
4.1	Dôležité pojmy.....	15
4.2	Animácia charakteru .....	16
4.2.1	Animácia hlavy a jej mimiky .....	16
4.2.2	Animácia tela .....	17
4.3	Animácia objektu po trajektórii.....	18
4.4	Animovania s využitím simulačných programov alebo integrovaných nástrojov .....	19
5	Low-poly modelovanie .....	21
6	Unity3D.....	22
6.1	Rendering v Unity3D .....	22
6.2	Nasvietenie .....	22
6.3	Terén .....	22
6.4	Fyzika .....	22
6.4.1	Rigidbody(obr. 25) .....	22
6.4.2	Joints.....	23
6.4.3	Soft Bodies.....	23
6.4.4	Wheel Collider .....	23
6.4.5	Kolízie šiat.....	23

6.5	Programovanie .....	23
7	Testovanie zozbieraných dát .....	24
7.1	Živé divadlo.....	24
7.2	Prírodné prostredie vo VRML.....	26
7.3	Model zámockého parku vo Vlašimi .....	30
8	Messerschmit Bf109 .....	35
8.1	Rendering .....	35
8.1.1	Podklady .....	35
8.1.2	Modelovanie.....	35
8.1.3	Osvetlenie.....	35
8.1.4	Rendering .....	36
8.1.5	Mapping.....	36
8.1.6	Finálny render.....	36
8.2	Animácie .....	36
8.3	Low-poly modelovanie .....	36
8.4	Unity3D.....	36
9	Záver .....	37
	Literatúra .....	39
	A Zoznamy .....	41
	B Blueprinty .....	43
	C Obsah priloženého CD .....	45



# 1 Úvod

Táto práca je určená pre tých, ktorý by radi tvorili kvalitné modely, ktoré sa dajú použiť viac než jeden krát. Pri písaní práce som sa snažil o objasnenie všetkých dôležitých pojmov.

V kapitolách tejto práce sa oboznámite s tým ako správne vytvoriť alebo naanimovať model. Takýto model bude následne použiteľný v inom softwari alebo na iný účel než bol pôvodne vytvorený. Oboznámite sa s tým aké formáty sú najpoužívanejšie a najperspektívnejšie pre uloženie projektu. V každej sekcii som sa snažil o popísanie toho čo je a čo nie je možné exportovať a čo sa stane s modelom po otvorení v inom softwari.

Práca je rozdelená do šiestich sekcií podľa toho čo je s modelom možné urobiť. Prvou sekciou je rendering, ktorý je spojený s tvorením modelov rôznymi metódami. V druhej časti som popísal animovanie objektov. Tretiu sekciu tvoria zásady pre tvorbu herných modelov. Štvrtou časťou je pomerne nová internetová technológia Unity3D, využívaná k tvorbe hier alebo interaktívnych aplikácií. V predposlednej piatej časti som popísal chyby, ktoré boli spáchané na zozbieraných modeloch z diplomových prác a ich riešenie. Poslednou časťou je mnou vytvorený model podľa pravidiel, ktoré sa v tej práci nachádzajú a jeho otestovanie pre 4 druhy použiteľnosti.

Po vytvorení modelu podľa pravidiel popísaných v tej práci by mal byť model vhodný pre znovu použitie v akejkoľvek oblasti.

## 2 Pravidlá znovu použiteľnosti 3D modelov

Na začiatku každého 3D projektu je nejaká myšlienka či úloha s ktorou sa grafik musí vysporiadať. Existuje mnoho techník ako dosiahnuť požadovaný výsledok. Každý jeden grafik má zopár svojich vlastných techník, ktoré aplikuje na najrôznejšie úlohy. Kľúčom úspechu je použitie tej správnej a najefektívnejšej.

U každého projektu je dôležité mať adresárovú štruktúru, ktorá definuje zaradenie každého modelu a umožňuje jeho rýchle vyhľadanie. Finálny model by sa mal okrem normálneho formátu uložiť ešte aj do formátu, ktorý nie je závislý na verzii programu.

### 2.1 3D formáty

Formátov ktoré obsahujú 3D obsah je veľmi veľa, avšak formátov ktoré je možné otvoriť vo viacerých programoch už toľko nie je.

#### *FBX - fbx*

Formát FBX bol vyvinutý spoločnosťou Autodesk a poskytuje možnosť výmeny dát medzi rôznymi programami. Jedná sa v podstate o rozšírenie formátu COLLADA. Umožňuje prístup k dátam aj po inovácii softwaru.

Dokáže uložiť textúry, materiály, animácie, geometriu, svetlá, kamery, jednotky v ktorých bola scéna vytvorená a nastavenia svetelného okolia. Od FBX 2011 je možné scénu uložiť taktiež v ASCII. Viac informácií je možné nájsť na stránkach Autodesku (viď [1]).

#### *OBJ - .obj*

Je to geometricky definovaný formát vyvinutý spoločnosťou Wavefront Technologies. Dáta reprezentujú 3D geometriu – pozíciu každého vertex, UV koordináty textúry, normály a plochy, ktoré definujú každý polygon ako list vertexov a vertexov textúry. Informácie som čerpal zo stránok Paula Bourka (viď [2]).

#### *Collada - .dae*

Tento formát bol vytvorený pre prenos scén medzi rôznymi programami. Posledná verzia 1.5 bola vytvorená v roku 2008 a na rozdiel od formátu FBX sa tento formát od vtedy nezmenil.

Dokáže uložiť mesh, animácie, materiály, svetlá, kamery.

#### *VRML - .wrl*

Formát vznikol v roku 1994 a jeho najnovšia verzia je z roku 1997. Formát sa využíval k zobrazovaniu 3D modelov v prehliadačoch avšak s nástupom Unity3D jeho budúcnosť nevyzerá veľmi priaznivo.

Z dostupných formátov má najperspektívnejšiu budúcnosť formát FBX a pokiaľ nevyjde nová verzia formátu Collada tak ju nové programy prestanú podporovať a upadne do zabudnutia. OBJ sa oproti FBX pre archiváciu scény určite nehodí z dôvodu, že dokáže uložiť len veľmi málo informácií. A čo sa týka formátu VRML, podľa poslednej verzie z roku 1997, do zabudnutia už upadol. Preto odporúčam scény ukladať vo formáte FBX a zároveň aj vo formáte Collada a OBJ.

## 2.2 Zásady tvorby scény

Každý model by mal byť tvorený tak aby sa mohol znovu použiť a to bez ohľadu, na aký účel bol pôvodne vytvorený. Model by mal byť kompletný a jeho logické časti oddelené (napr. bicykel by sa mal skladať z kolies, volantu, konštrukcie, pedálov, prevodovky, atd.).

Postup tvorby scény by mal spĺňať nasledujúce body:

- Vytvorenie náčrtov a rozdelenie práce
- Zvolenie správneho programu
- Vymodelovanie scény
- Nasvietenie scény, pridanie kamier
- Nastavenie rendru
- UVW mapping, normálové mapy a tvorba materiálov
- Rendering
- Animácia
- Post produkcia

Použiteľnosť 3D modelov je rozmanitá a dala by sa uzavrieť do 4 skupín.

### 1. **Rendering**

- Najobsiahlejšia skupina, ktorej výsledkom je obrázok. Svoje uplatnenie využíva najčastejšie v reklame.

### 2. **Animácie**

- Výsledkom je postupnosť vyrenderovaných obrázkov, tzn. video. Animácie sa využívajú vo filmovom priemysle, reklame alebo v hrách.

### 3. **Low-poly modelovanie**

- V tejto práci je určite nutné spomenúť aj tento druh modelovania, ktorý sa najčastejšie uplatňuje v hrách.

### 4. **Unity3D**

- Unity3D je plugin pre webové prehliadače a umožňuje zobrazenie 3D obsahu v podobe 3D hier alebo najrozmanitejšieho interaktívneho obsahu.

### 3 Rendering

Rendering v preklade znamená interpretácia. V súvislosti s 3D grafikou to znamená vykreslenie vytvorenej scény pomocou zvoleného rendrovacieho systému. Kvalita výsledného rendru závisí od mnohých aspektov (hardware, modelovací a rendrovací software, schopnosti grafika), ktoré vplývajú na rôzne etapy jeho tvorby.

*Aspekty tvorby:*

Hardware – Lepší hardware umožňuje modely s vyšším stupňom detailu. Zvyšuje rýchlosť tvorby scény a znižuje render time.

Modelovací software – software by mal podporovať formáty FBX, Collada a OBJ.

Rendrovací software – medzi najpoužívanejšie systémy patrí RenderMan, V-ray, Mental-ray alebo Maxwell

Schopnosti grafika – každou minútou venovanou tvorbe, každou vytvorenou scénou, grafik naberá skúsenosti.

#### 3.1 Vytvorenie náčrtov a rozdelenie práce

Na začiatku každého projektu je nutné si ujasniť ako má vyzeráť výsledok. Základom úspechu je vytvorenie náčrtov, ktoré budú slúžiť ako podklad k práci. Nezáleží na tom ako je grafik zručný v kreslení, hlavným účelom je dať myšlienkam určité tvary, premyslieť si ako by mal výsledný render vyzeráť a čo by mal obsahovať.

Pri tímových projektoch sa práca rozdeľuje medzi skupiny ľudí, ktoré pracujú na jednom bode a po vypracovaní ho posúvajú ďalšej skupine.

#### 3.2 Zvolenie správneho programu

Po prvom kroku by mal byť jasný zámer projektu. K vytvoreniu jeho tvaru sa využíva najrozmanitejší software. Aký program by sa mal zvoliť záleží od projektu, avšak z hľadiska znovu použiteľnosti by mal podporovať formát FBX, Collada a OBJ. Existuje mnoho druhov softwaru a dali by sa rozdeliť do 2 veľkých kategórií.

##### 1. Komerčné

##### 2. Nekomerčné

Obe skupiny obsahujú podskupiny programov, ktoré sú určené ich použiteľnosťou.

##### 1. Modelovacie

Slúžia k vytvoreniu scény a rendru. Medzi najpopulárnejšie patria 3d studio MAX, Maya, Cinema 4D, Rhino 3D, Softimage XSI alebo Solid Works. Najpoužívanejším nekomerčným programom je Blender.

##### 2. Simulačné

Simulačné programy stavajú na fyzikálnych zákonoch a ich výsledok je blízko realite. Sú vhodné napríklad pre animovanie tečúcej vody alebo demolácie budovy. Často používaným programom je RealFlow

### **3. Sculptovacie**

Programy, pomocou ktorých sa dajú vymodelovať tie najmenšie detaily. Umožňuje pridať postave vrásky, žilky či jazvy v priebehu len pár minút. Najpoužívanejším programom je Z-Brush.

### **4. Post produkčné**

Za zmienku stojí aj tento druh programov, avšak v tejto práci sa im venovať nebudem. Slúžia na odstránenie posledných chýb na kráse projektu či pridanie najrozmanitejších efektov. Medzi najrozšírenejšie patria Adobe Photoshop, Adobe After Effects alebo Premiera.

## **3.3 Vymodelovanie scény**

Existuje mnoho modelovacích techník, ktoré sú vhodné pre rôzne situácie. Čo je kedy vhodné použiť sa nedá stopercentne povedať. Najefektívnejšiu techniku dokáže grafik určiť až po rokoch modelovania.

Pri renderingu nie je najdôležitejší model. Modelovanie je vo väčšine prípadov len 15% úspechu. Neznamená to avšak, že na modeli nezáleží. Práve naopak, tvorí základ pre ďalšie body. Bez dobre navrhnutého modelu je veľmi obtiažne použiť materiály, rôzne efekty alebo vytvoriť jemné detaily sculptingom. V 100% prípadoch to vedie k problémom, strate času a peňazí.

### **3.3.1 Techniky modelovania:**

#### **a. Modelovanie použitím primitívnych telies**

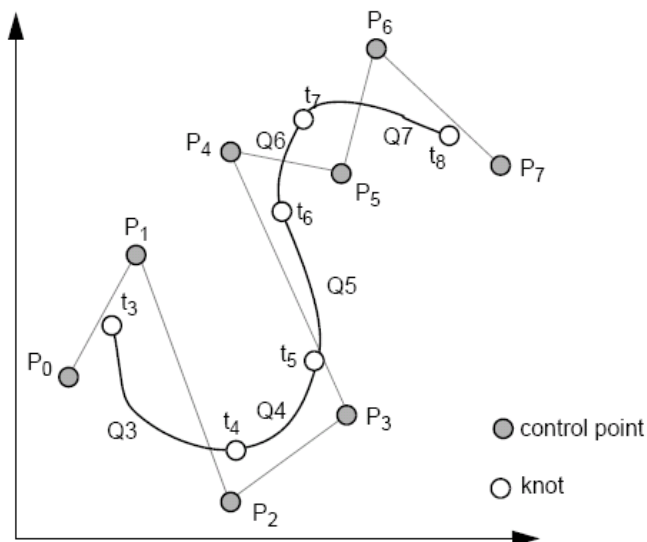
Každý program má nejakú základnú sadu telies. Medzi najčastejšie sa vyskytujúce patrí plocha (plane), kváder (Box), valec (Cylinder) a guľa (Sphere).

Modelovanie spočíva vo využívaní týchto telies bez prevedenia do editovateľného módu. Tvoria sa tým jednoduché modely, ktorých rendertime a pamäťové nároky sú oproti editovaným nižšie. Pivot telies by sa mal nachádzať v mieste ich otáčania aby sa dali použiť okrem renderingu aj napríklad v animáciách. Príkladom by mohli byť lopty, jednoduché stoly, stĺpy a podobné predmety vychádzajúce z primitívnych telies.

#### **b. Modelovanie s využitím NURBS kriviek**

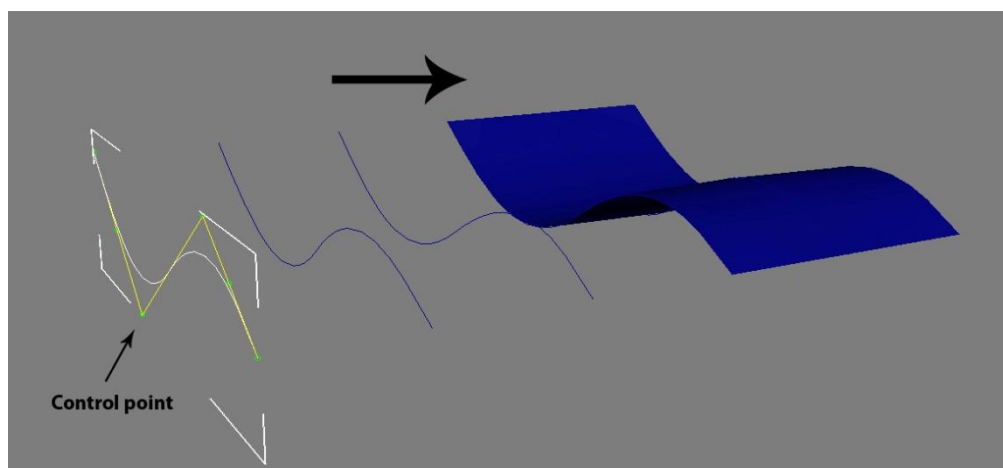
Táto technika sa využíva pri konštrukcii obecných tvarov v dizajne, v automobilovom priemysle, letectve alebo pri animáciách. Pri modelovaní sa využívajú rôzne geometrické tvary a tzv. B-splines (obr. 1), ktoré obsahujú n kontrolných bodov pre manipuláciu s nimi.

Výhodou NURBS kriviek je, že nevyžadujú až toľko operačnej pamäte a preto je manipulácia s veľkou scénou v prostredí softwaru ľahšia. Na druhú stranu render nie je rýchlejší pretože pri renderingu sa model prevádza na mesh.



Obrázok 1: Beziér krivka [3]

NURBS plocha sa tvorí tak, že sa vytvoria krivky, ktoré popisujú plochu a prepoja sa určitým nástrojom (obr. 2).

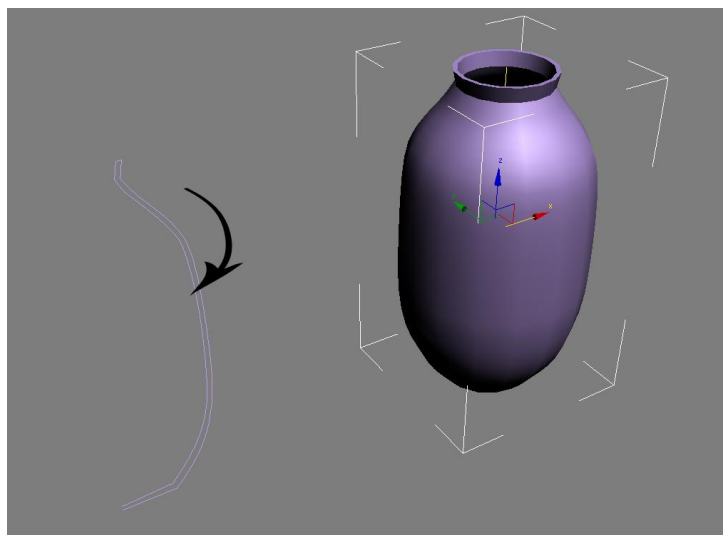


Obrázok 2: Tvorba NURBS plochy

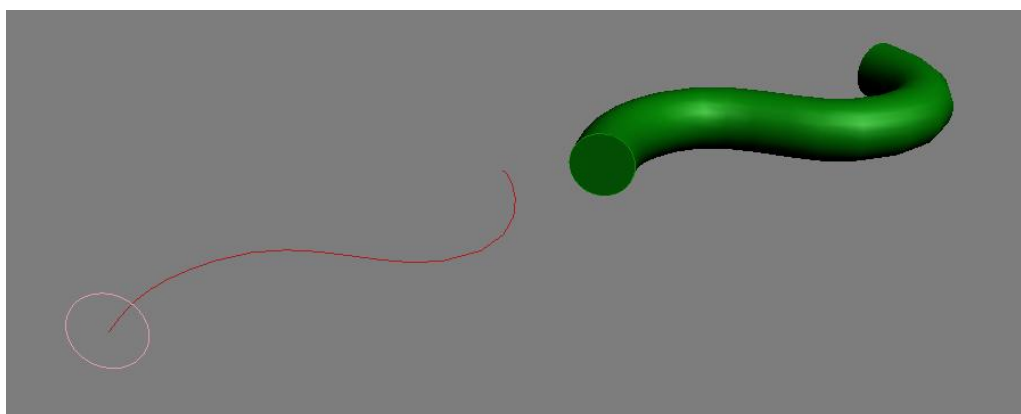
Modely tvorené touto technikou je možné vyexportovať do formátu FBX bez straty ich neskoršej editovateľnosti. Export do tohto formátu avšak v programe Blender nefunguje až tak dobre pretože prevedie NURBS objekt do mashu. Čo sa týka formátu Collada, tak tento formát bohužiaľ nedokáže uložiť NURBS plochy a pri exporte skončí chybou. Formát OBJ prevedie model do mashu.

#### c. Modelovanie pomocou 2D prvkov

Tento spôsob modelovania využíva rôzne transformácie s krivkami. Medzi tieto transformácie patrí napríklad orotovanie krivky okolo osy (Obr. 3) alebo pretiahnutie profilu krivkou (Obr. 4)



Obrázok 3: Orotovanie krivky okolo osy

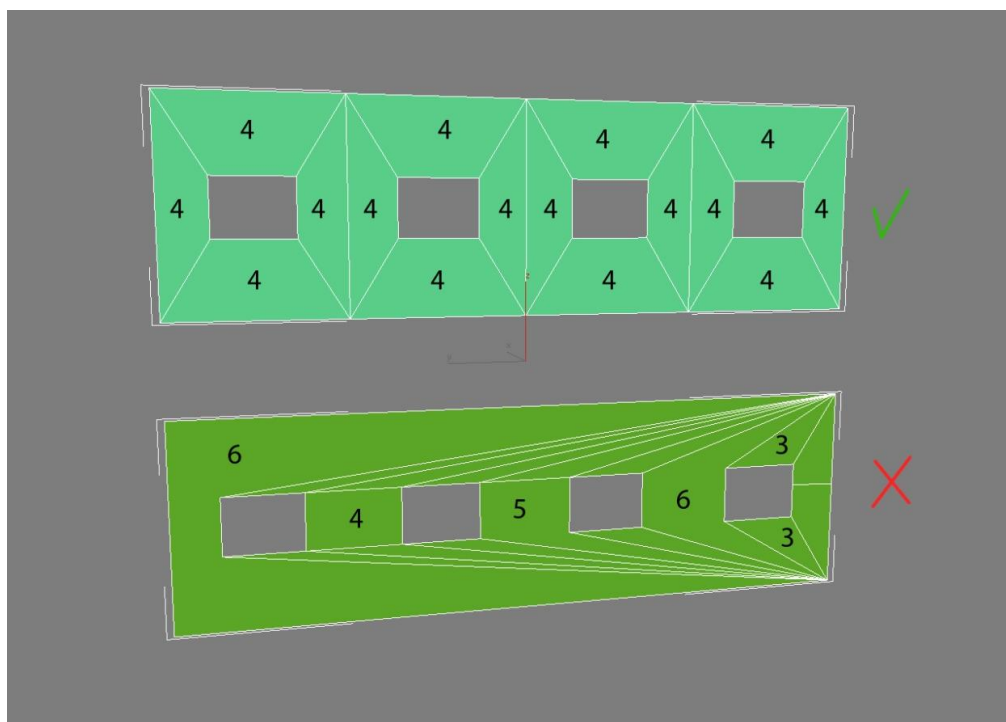


Obrázok 4: Pretiahnutie krivky profilom

Tieto dve techniky majú výhodu v rýchlej editácii modelov avšak veľkú nevýhodu pri exportovaní a to, že sa objekt prevedie do mashu a tým jeho výhoda s ľahkou editáciou končí. Často sa využívajú pre tvorbu lana, potrubí alebo váz.

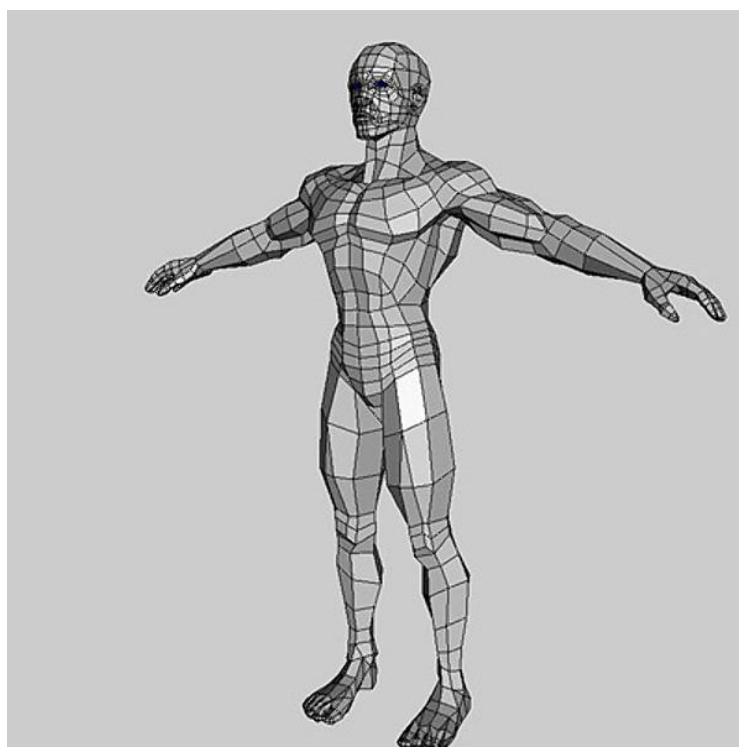
#### d. Polygonálne modelovanie

Pod názvom polygón sa skrýva plocha objektu. Základom správneho polygonálneho modelovania je pracovať so štvorhrannými polygónmi, snažiť sa čím viac obmedziť trojuholníkové polygóny a eliminovať polygóny s väčším počtom plôch (obr. 5). Jedným z dôvodov je, že počítač rád pracuje s číslami, ktoré sú v tvare  $2^n$ . Ďalším dôvodom je, že topológia objektu je prehľadná a uľahčí sa sub-division modelovanie a textúrovanie. Medzi bežné chyby patria aj prekrývanie polygónov alebo obrátené normály.



Obrázok 5: Príklad správnej a nesprávnej topológie

Topológia znázorňuje rozloženie a tok polygónov. Pre objasnenie a správne určenie toku polygónov je potrebné zakaždým určiť štruktúru objektu a na čo je určený. U charakteru sa napríklad topológia a jej tok určuje podľa svalstva (obr. 6).



Obrázok 6: Topológia charakteru [4]

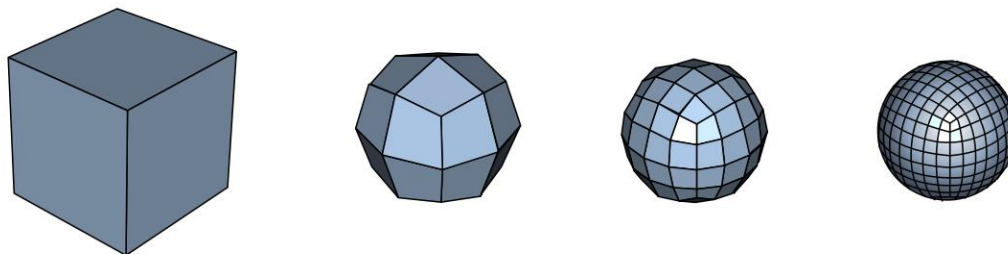
Dôležité je myslieť aj na budúcnosť modelu. To, že auto sa momentálne nepohybuje neznamena, že kolesá a karoséria majú tvoriť jeden model. Pri každom jednom modeli je potrebné oddeľovať funkčné časti. Ďalším možným príkladom nesprávneho delenia na časti



je rozdelenie domu na desiatky kvádrov. To čo je v normálnom svete oddelené by malo zostať oddelené aj v tom virtuálnom.

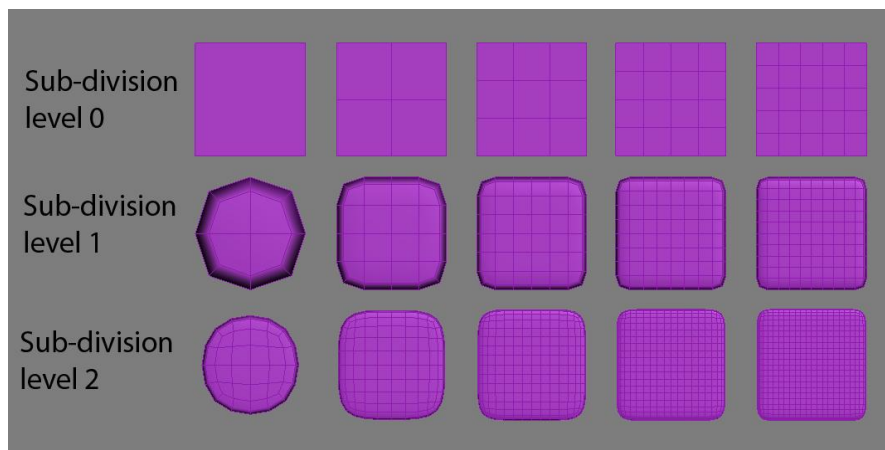
e. Sub-division modelovanie

Je to metóda, ktorá spočíva v delení polygónu na menšie a tým vo vyhladzovaní povrchu objektu (obr. 7).



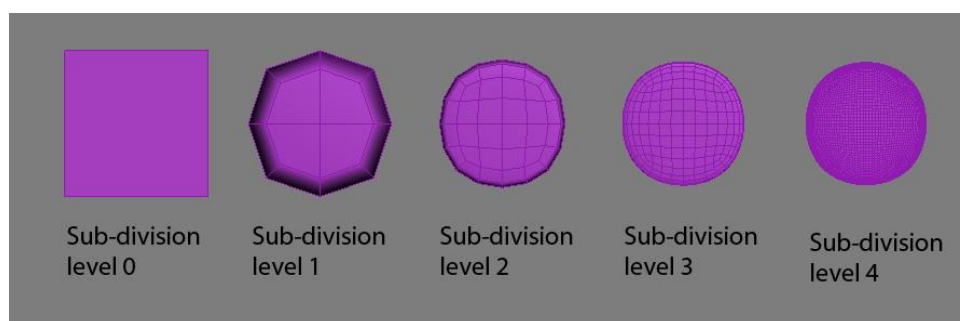
Obrázok 7: Sub-division modelovanie [5]

Miera vyhladenia sa ovplyvňuje topológiou objektu a sub-division levelom. Čím je topológia hustejšia tým sa objekt viac podobá na objekt s levelom 0 (Obr. 8).



Obrázok 8: Sub-division level – vliv topologie na level

Každým levelom sa počet polygónov dvojnásobne stupňuje. Ako je vidno na obrázku 9 zvyšovanie levelu je nepraktické použiť väčšie ako 3. Model vyzerá rovnako avšak má omnoho viac polygónov. Táto technika je vhodná pre finálne úpravy modelu alebo pre zhustenie topológie, napríklad pri tvorbe terénu.



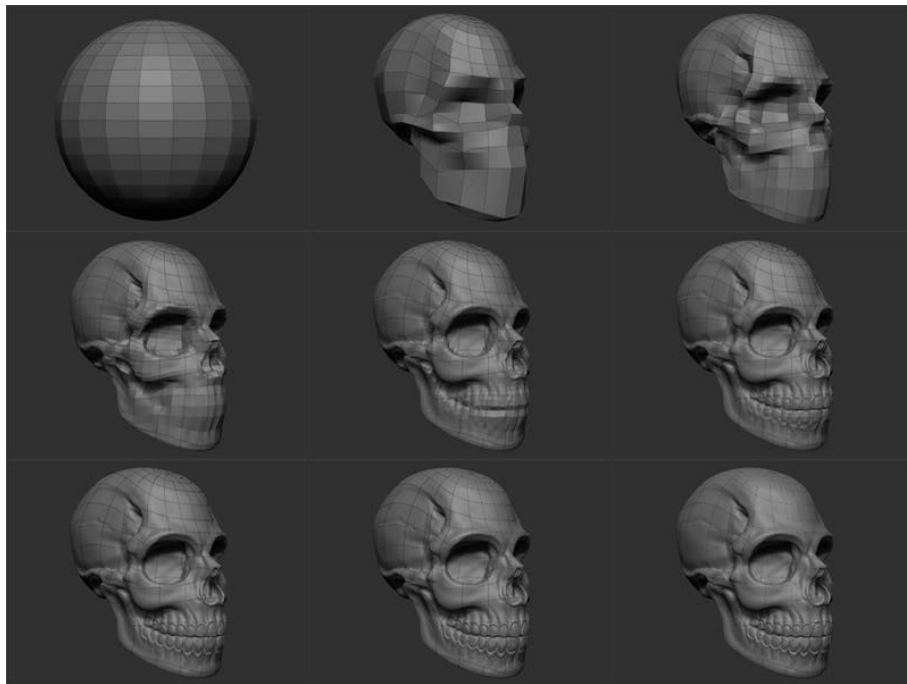
Obrázok 9: Sub-division level – vliv levelu na topológiu

Formát FBX na rozdiel od formátu Collada a OBJ dokáže uložiť sub-division modifikátor a po importovaní objektu je možné ho upravovať.

f. Sculpting

Sculptovací software(Z-Brush) dokáže importovať objekty vo formáte OBJ. Pre úpravu modelu využíva desiatky rôznych nástrojov (napr. push, pull , smooth, pinch, atď.). Pri modelovaní sa môže vychádzať zo základného modelu, ktorý je jednoduchým modelom bez väčších detailov alebo sa môže celý model vytvoriť priamo v prostredí softwaru. Technika modelovania je rovnaká ako pri modelovaní z hliny, t.j. vtlačaním, vytlačaním či vyhladzovaním hliny sa modeluje povrch telesa.

Pri tejto technike sa postupuje tak, že v momente keď je vymodelovaná maximálna úroveň detailu, zvýši sa hustota topológie a pokračuje sa znovu modelovaním ďalšej úrovne detailu (obr. 10). Tento úkon sa opakuje až do doby, keď je s modelom grafik spokojný. Po dokončení modelovania sa na záver vytvorí normálová mapa, ktorá sa aplikuje na model v prostredí modelovacieho softwaru.



Obrázok 10: Iterácie sculptingu [6]

### 3.4 Nasvietenie scény

Neoddeliteľnou súčasťou každej scény je nasvietenie. Mnoho začínajúcich grafikov si myslí, že jedno svetlo v scéne je dosť. Avšak nie je! Ideálny počet svetiel neexistuje a záleží len na scéne a na tom, čo sa grafik snaží vytvoriť. Dobré nasvietená scéna je základom k ďalšiemu postupu a tvorí 25% úspechu.

Ďalšou veľkou chybou je nasvecovania scény s použitými materiálmi. Z pravidla sa používa sivý materiál pre všetky objekty v scéne. Dôvod je veľmi prostý a to, že svetlo mení materiál, stmavuje ho alebo zvýrazňuje, upravuje jeho farbu či tvorí odlesky. Je jednoduchšie pre jedno nasvietenie aplikovať rôzne materiály na rôzne objekty ako pre hotové materiály nájsť práve jedno nasvietenie, ktoré by vyhovovalo každému objektu v scéne.

Pri najjednoduchších scénach sa používajú minimálne 3 svetlá. Z týchto troch svetiel je jedno hlavným zdrojom svetla, t.j. má najsilnejšiu intenzitu. Zvyšné svetlá dopĺňajú svetelnú scénu zjemňovaním/potlačovaním tieňov, zvýrazňovaním odleskov alebo rôznymi svetelnými efektami(napr. glow, lens effects). Dobrým príkladom správneho nasvietenia je štúdiové fotenie automobilov(obr. 11)



Obrázok 11: Nasvietenie v profesionálnom štúdiu [7]

Formáty FBX a Collada exportujú svetlá rovnako a podporujú len niekoľko druhov svetiel. Formáty nepodporujú napríklad svetlá, ktoré boli do softwaru pridané, t.j. svetlá od rendrovacích systémov. Pri importovaní do iného programu sa svetlá zmenia na druh aký daný software podporuje. Z hľadiska znovu použiteľnosti je najlepším riešením použiť podporované svetlá, avšak z hľadiska kvality výsledného rendru je lepšie svetlá vytvoriť znovu.

### 3.5 Nastavenie rendru

Táto časť obsahuje množstvo nastavení a testovania. Je dôležité znížiť čas redrovania na čo najmenší a to z dôvodu, že keď prídu na rad materiály a testovanie s tým spojené nebude nutné čakať hodiny na výsledok.

Nastavenia rendru záležia na použitom rendrovacom systéme. Najpoužívanejšou technikou, ktorú môžem len a len odporučiť je metóda "pokus, omyl". Pri tejto metóde je potrebné si dávať pozor na pomer render time/kvalita. Pre testovanie materiálov v ďalšom kroku je potrebné nechať nastavenia rendru na nízkom rozlíšení.

Po správnom nastavení rendru odporúčam vyrobiť si jeden render v lepšom rozlíšení aby bolo jasné, že pri rendrovaní finálneho obrázku sa nezmení nič iné než jeho rozlíšenie.

### 3.6 UVW mapping, normálové mapy a tvorba materiálov

Poslednou, finálnou časťou je tvorba textúr a materiálov. To, že celá scéna je nádherne nasvietená je veľmi pekné, to je však ešte len 65% projektu hotového. Tento bod je

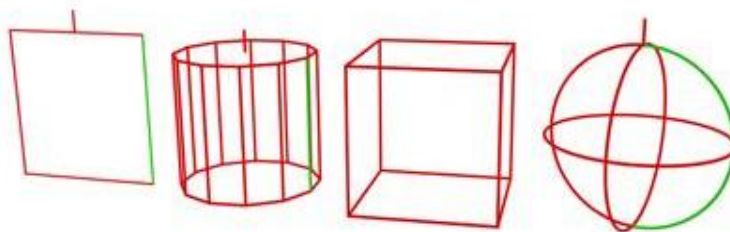
najdôležitejšou časťou celého projektu a zároveň najneobľúbenejšou časťou asi každého grafika.

V tomto kroku sa sivý materiál každého objektu nahradí správnym materiálom alebo textúrou a prípadne sa na model aplikuje normálová mapa. Pripomínam, že rozlíšenie rendru musí byť nastavené na čo najnižšie, aby sa na výsledky skúšobných rendrov nečakalo príliš dlho.

Formát FBX a DAE dokáže uložiť akúkoľvek mapu, ktorá nie je procedurálna. Formát OBJ vyexportuje minimálne dva súbory, prvý má informácie o objekte (koordináty vertexov a UV súradnice) a druhý obsahuje informácie o materiáli objektu. Ďalšími možnými súbormi sú mapy, na ktoré sa odkazuje v súbore s materiálmi.

### 3.6.1 UVW

Každý objekt, ktorý obsahuje akúkoľvek textúru musí mať nadefinované UVW koordináty, t.j. program musí vedieť ako má danú textúru namapovať na objekt.



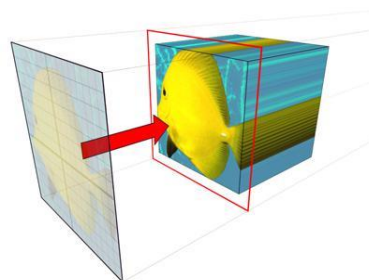
Obrázok 12: druhy gizma [8]

K tomu, ako sa má na objekt namapovať textúra, slúži gizmo (obr. 12). Gizmom určujeme ktoré súradnice textúry prídu na aké miesto.

#### **Mapovanie s použitím rôznych typov gizma**

##### a) Planar mapping (obr. 13)

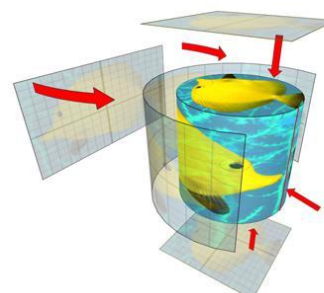
Projektuje textúru len na jednu stranu. Toto mapovanie je vhodné ak nie je potrebné vidieť nič iné než jednu stranu objektu (napr. terén alebo pracovná plocha monitoru).



Obrázok 13: Planar mapping [8]

##### b) Cylindrical mapping (obr. 14)

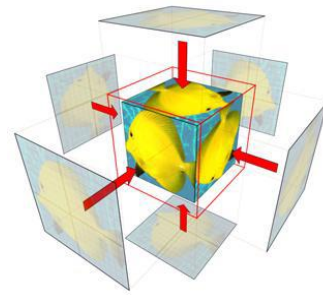
Projektuje mapu tak, že obalí objekt dookola textúrou. Tento spôsob je vhodný pre objekty v tvare cylindru (napr. fľaša, plechovka alebo sud).



Obrázok 14: Cylindrical mapping [8]

c) Box mapping (obr. 15)

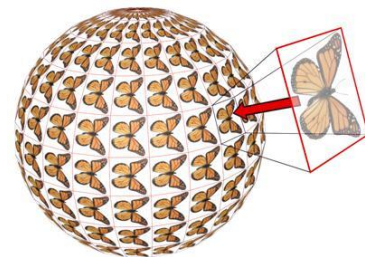
Projektuje mapu zo šiestich strán na objekt, tzn. že využíva ako keby šesť planárnych gízm. Toto mapovanie je vhodné na objekty v tvare kvádra (napr. dvere, stôl alebo skrinka).



Obrázok 15: Box mapping [8]

d) Face mapping (obr. 16)

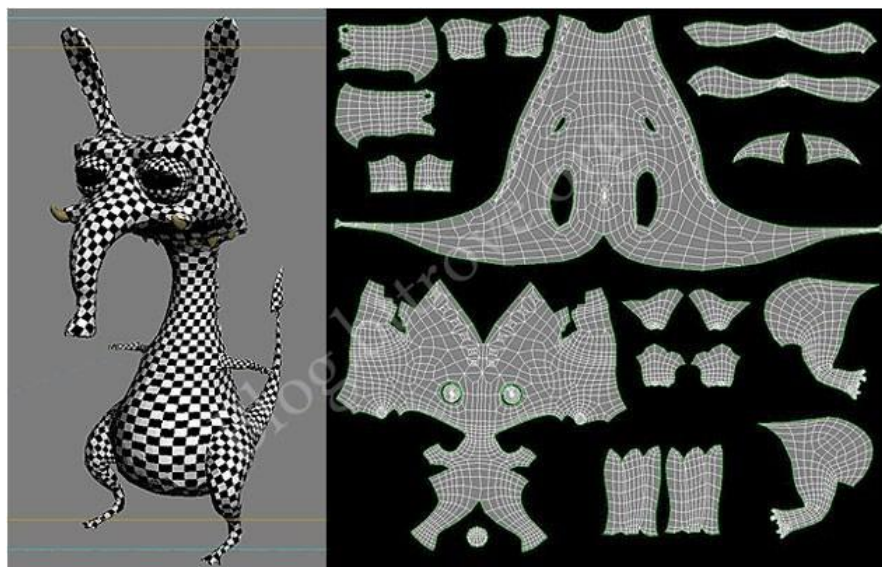
Tento druh mapovania aplikuje textúru na každý polygón objektu, tzn. každý polygón má svoje vlastné planárne gizmo. Vhodné to môže byť napríklad na vytapetovanie izby.



Obrázok 16: Face mapping [8]

### 3.6.2 UNWRAP

UNWRAP slúži k vytvoreniu jedinečnej projekcie textúry na objekt. Pri tomto mapovaní záleží predovšetkým na kvalite mashu. U komplexných modeloch sa objekt rozdelí na logické časti (napr. u človeka sú to končatiny, hlava a trup) a z nich sa vytvorí jedinečná projekcia každej časti (obr. 17). Jediné čo je treba dodržať je, že mierka v akej sú tieto projekcie vytvorené musí byť rovnaká.



Obrázok 17: UNWRAP [9]

### **3.7 Rendering**

V tomto bode je už všetko pripravené na to aby sa mohol spustiť finálny render, ktorý už máme pripravený z bodu 3.5. Pri rendrovaní zložitých obrázkov alebo filmov sú často využívané render farmy, čo je vlastne zoskupenie výkonných počítačov využívajúce off-line batch processing (spracovanie v dávkach). Po vyrendrovaní práca grafika samozrejme ešte nekončí a nasleduje postprodukcia.



## 4 Animácia

Animovanie je najťažším odvetvím 3D grafiky. To, aby animácia bola úspešná vyžaduje roky tréovania, učenia sa a pozorovania okolitého sveta očami animátora. Úspech závisí z veľkej časti aj na modely a na jeho topológii (viď. bod 1.3). Rovnako ako u renderingu aj u animácie existuje mnoho techník. Techniky sa odvíjajú od toho čo práve animuje.

### 4.1 Dôležité pojmy

#### *Pivot*

Pivot znamená bod objektu, okolo ktorého sa objekt otáča. Taktiež určuje pozíciu objektu od stredu súradnicového systému. Je dôležité aby bol pivot umiestnený v mieste v ktorom sa objekt otáča (napr. dvere by mali mať stred otáčania v pántoch) a ak sa neotáča, pivot by mal byť v strede objektu.

#### *Keying*

Kľúčovanie znamená vytvorenie časovej známky s informáciami (position, rotation, scale, size, value atd.) o objekte, materiály, zvuku alebo prostredí.

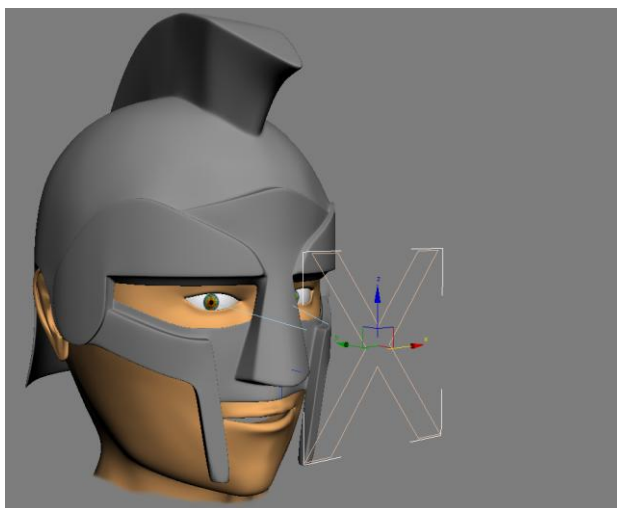
#### *Linkovanie*

Linkovanie je vytváranie závislostí medzi objektmi, tzn. ak je napríklad objekt A nalinkovaný na objekt B a pohybujeme, rotujeme alebo zmenšujeme objekt B, rovnaké zmeny sa prejavajú aj na objekte A. To avšak neplatí naopak, tzn. linkovanie je jednosmerné. Tieto závislosti sa často využívajú pri modelovaní technických zariadení (časti ramena žeriavu alebo oká reťaze).

#### *Helper*

Pojem helper označuje pomocný objekt, ktorý je nalinkovaný na kosti alebo na objekty. Obvykle je helper zobrazovaný ako kocka. Helpe sa nedajú vyrendrovať a slúžia len ako pomôcka na uľahčenie práce s objektami. Po pridaní týchto pomocných objektov už nie je napríklad potrebné hľadať kosti v objekte, ale stačí posúvať helprami a tým manipulovať s jednou alebo viacerými časťami kostry.

Ako helpe je možné využiť taktiež aj iné objekty a nastaviť priehľadnosť pri rendrovaní. Často sa využíva kruh, napríklad pre rotáciu trupu. Ďalšou možnosťou je využiť text. Využíva sa napríklad písmeno "X" pre vyjadrenie miesta kam sa postava pozerá (obr. 18).

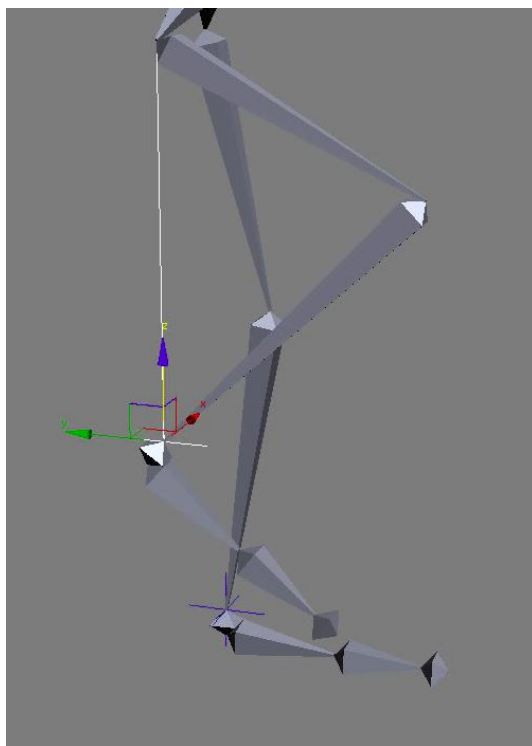


Obrázok 18: animácia pohľadu

Pri exportovaní treba mať na pamäti, že 2D objekty sa nedajú vyexportovať do žiadneho z troch spomínaných formátov.

## IK solvers

IK solver vytvára inverzné kinematické riešenie pre rotovanie a pohyb kostí (obr. 19).



Obrázok 19: IK solver

## 4.2 Animácia charakteru

Pri animácii charakteru si treba uvedomiť čo je to za charakter (človek, zviera, príšera) a ako sa pohybuje (plazí sa, chodí po dvoch alebo dvanástich nohách).

U človeka je rozdiel pri animácii ženy alebo muža, pri animácii dieťaťa, starého alebo mladého človeka. Záleží na tom čo má postava oblečené (ťažké brnenie alebo len ľahké tričko a nohavice), na tom či je zranená alebo má nejaké nadprirodzené schopnosti.

U zvierat a rôznych vymyslených tvoroch záleží na stavbe ich tela (kostra, svalstvo). Najlepším zdrojom informácií sú dokumentárne filmy alebo kniha biológie.

### 4.2.1 Animácia hlavy a jej mimiky

#### 4.2.1.1 *Morpher*

Morpher je nástroj, ktorý využíva 3d studio max pre animáciu mimiky. Tento nástroj využíva iné objekty s rovnakým počtom vertexov (bodov) a napodobňuje ich. Postupuje sa tak, že sa urobia kópie základného objektu a prevedú sa zmeny (obr. 19). Na základný objekt sa aplikuje morpher a vyberú sa upravené kópie. Na stupnici od 0-100 sa potom už len určí do akej miery sa základný objekt zmení na jeho kópiu. Takýmto nástrojom disponuje aj Maya avšak pod názvom Blend shape.

*POZOR: Zmeny nesmú zahŕňovať pridávanie nových polygónov alebo vertexov.*



Formát FBX a Collada dokážu vyexportovať teleso s aplikovaným modifikátorom Morpher a to aj s vytvorenou animáciou. Keďže formát OBJ nedokáže vyexportovať animácie, výsledný súbor bude obsahovať len mash bez akéhokoľvek modifikátoru.

#### 4.2.1.2 Linkovanie

Ďalšou možnosťou je nalinkovať vertexy na helper a pohybovaním helpru meniť tvar objektu. Táto technika je jednoduchšia, avšak pri zložitejších animáciách nepraktická.

Z hľadiska znovu použiteľnosti má tento spôsob linkovania veľkou nevýhodou a to, že scénu s nalinkovanými vertexami nie je možné vyexportovať.

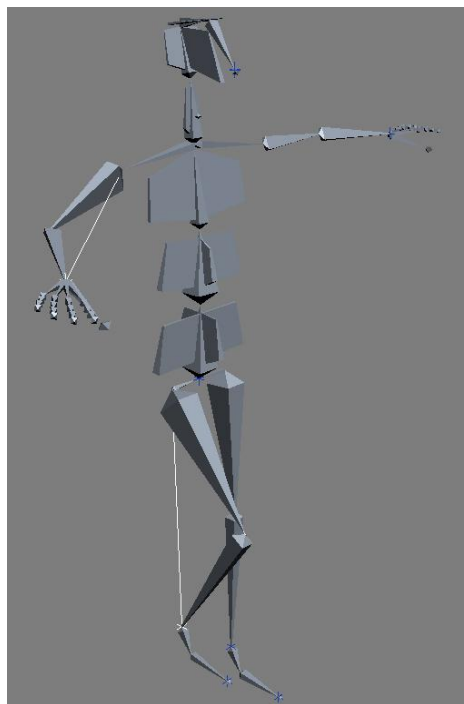
### 4.2.2 Animácia tela

To aby sme mohli pohybovať postavou potrebujeme viac než len správne vytvorený model. Prvým krokom je vytvorenie kostry. Kostra sa tvorí podľa toho jak sa má model pohybovať, tzn. v mieste kde sa niečo ohýba sa musia spájať dve a viac kostí. Druhým krokom je priradenie kostry modelu. V programe 3d studio max znamená tento krok aplikáciu modifikátoru skin (kože) na model a priradenie kostí tomuto modifikátoru. V treťom kroku sa upravujú obálky kostí (envelopes), tak aby pri ohýbaní nevznikali nežiaduce deformácie.

#### 4.2.2.1 Bones (obr. 21)

To ako sa tvorí kostra závisí od softwaru. V programe 3ds max je v mieste kde sa spájajú viac než dve kosti nutné použiť linkovanie aby kosti držali pokope a pohybovali sa spolu s modelom. Naopak v Mayi tento problém nie je a namiesto linkovania je možné kosti napájať na kĺby.

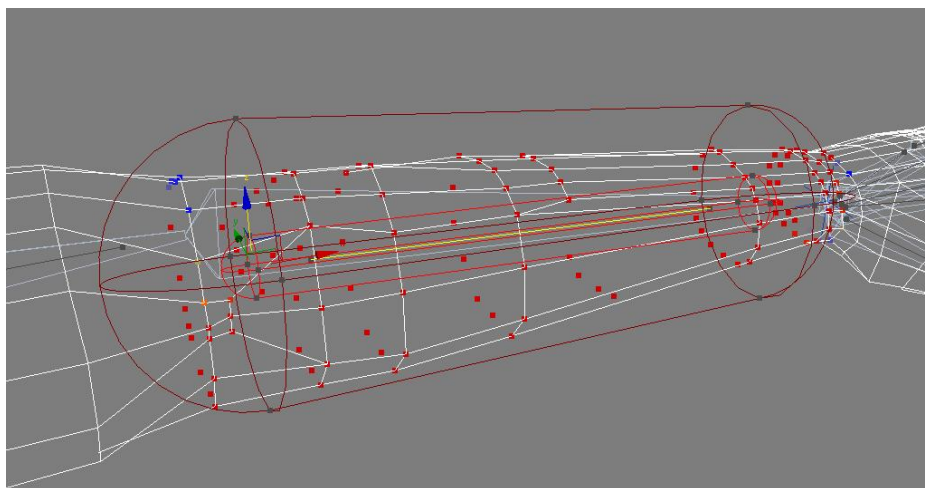
Export kostí nefunguje veľmi dobre. Kostí sa prevedú na pomocné objekty (Dummy), zmení sa vplyv na objekt a stáva sa, že pri pohybe kosťami neovplyvňujú vôbec nič.



Obrázok 20: Kostra postavy

#### 4.2.2.2 Koža a jej obálky (obr. 20)

Počet obálok sa odvíja od počtu použitých kostí. Obálky vyhradzujú oblasť vplyvu kostí na kožu. Váha vplyvu je určená farebným znázornením od červenej (najvyšší vplyv) po modrej (najslabší vplyv).



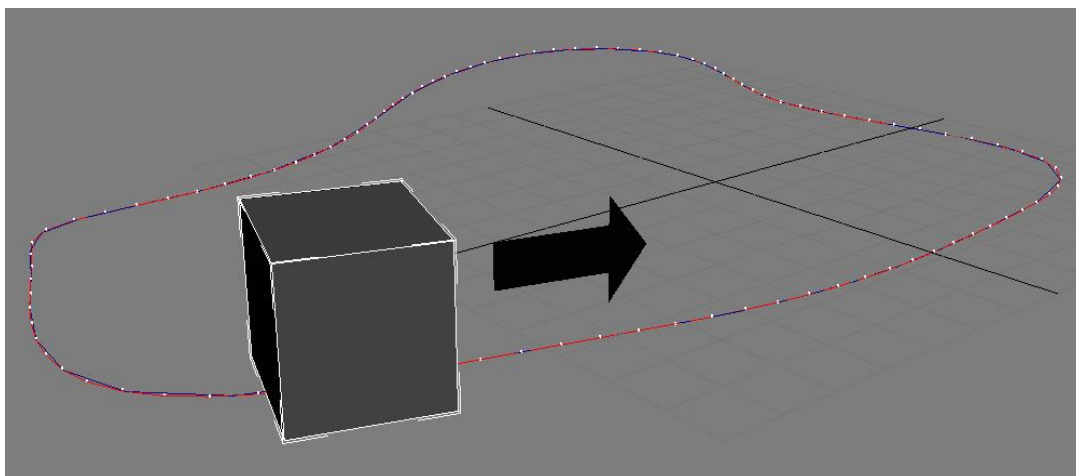
Obrázok 21: Obálka kosti

Váha vertexov sa dá meniť veľkosťou/umiestnením obálky a taktiež sa dá meniť aj pre každý samostatný vertex pomocou Weight Table. Je veľmi dôležité aby každý vertex mal presne takú váhu aby ohýbanie časti modelu deformovalo len to čo má.

Export kože do formátu FBX a COLLADA je možný, avšak nefunguje to úplne správne. Skin aj s nastaveniami sa exportuje, ale tým, že kosti sa prevedú na pomocné objekty, nie je možné meniť ich obálky. Po importovaní je bohužiaľ nutné vytvoriť kosť znovu, a tým sa sťažuje možnosť znovu použitia modelu.

### 4.3 Animácia objektu po trajektórii

Pre tento typ animovania stačia len dve veci: trajektória a objekt, ktorý sa bude po nej pohybovať (obr. 22). Týmto spôsobom sa ľahko animuje prelet krajinou alebo závody áut.

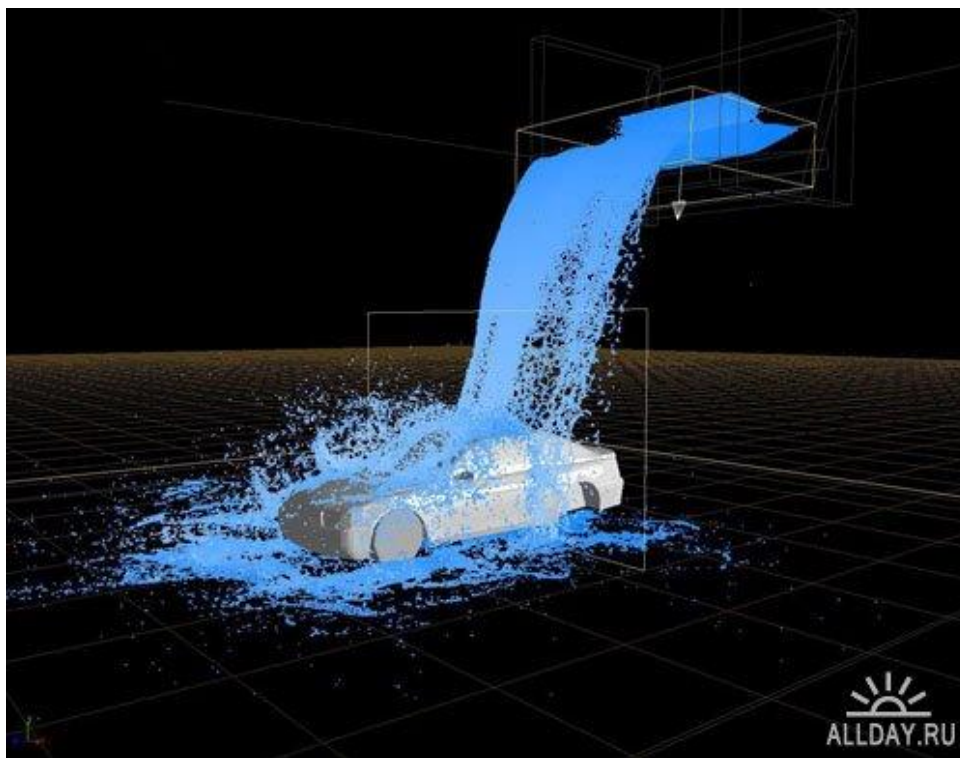


Obrázok 22: Animácia objektu po trajektórii

Do vyexportovaného súboru sa uloží objekt, trajektória a animácia. Nevýhodou je, že formáty FBX a Collada nedokážu vyexportovať krivku, z ktorej bola trajektória vytvorená. Na druhej strane väčšina programov podporuje vytvorenie krivky z trajektórie a tak je možné stratené dáta vrátiť.

#### 4.4 Animovania s využitím simulačných programov alebo integrovaných nástrojov

Pomocou simulačných programov sa zjednodušuje animovanie rôznych fyzikálnych javov (gravitácia, vietor) a ich účinku na objekty. Využitie týchto programov je obrovské, napr. na animáciu vody, vetru, ohňa alebo erupcií. Jedným z najpoužívanejších simulačných programov je RealFlow (obr. 23).



Obrázok 23: animácia vodopádu v programe RealFlow [10]

Program RealFlow využíva pre importovanie a exportovanie objektov a animácií formát Next Limit SD. Spoločnosť Next Limit Technologies, ktorá je tvorcom tohto programu, vytvorila taktiež plug-iny pre podporu tohto formátu u najpoužívanejších 3D programov. Objekty je možné vložiť do prostredia programu aj pomocou formátu OBJ. Nevýhodou tohto formátu je, že nedokáže uložiť animácie, t.j. pre export sa musí použiť už formát Next Limit SD.

Mnoho grafických programov má integrované systémy pre simulácie. V programe 3ds max sú to reactory a particle system.

##### *Reactor*

Reactor je sada nástrojov, ktoré umožňujú simulovať komplexné fyzikálne scény. Umožňuje simulovať dynamiku tela, textil, tekutiny, vietor alebo motor. Tento nástroj dovoľuje priradiť modelom ich váhu, flexibilitu alebo trenie. Okrem iného môže definovať vzťahy medzi objektami.

Často sa reactor využíva na deštruktívne účely, tzn. rozbitie skla, demolíciu budovy, alebo na rozpučenie auta tankom.

Reactor je skvelá vec, avšak formáty FBX a Collada ho nepodporujú.

### *Particle system*

Particle system sa dá definovať ako veľká kolekcia podobných objektov, ktoré sa správajú podobným spôsobom. Príkladom časticového systému je napríklad sneh, dážď ale taktiež voda, dym, alebo dokonca aj davy ľudí.

Čo sa týka exportu častíc, tak bohužiaľ rovnako ako u reactoru, formáty FBX a ani Collada particle system nepodporujú.

## 5 Low-poly modelovanie

Tento štýl modelovania sa snaží o tvorenie modelov s čo najmenším počtom polygónov. Počet polygónov je dôležitý parameter v hernom priemysle. Čím je počet polygónov nižší, tým sú nároky na výpočetnú techniku nižšie. Postup tvorby a techniky sú rovnaké ako v bode 1.3.1.

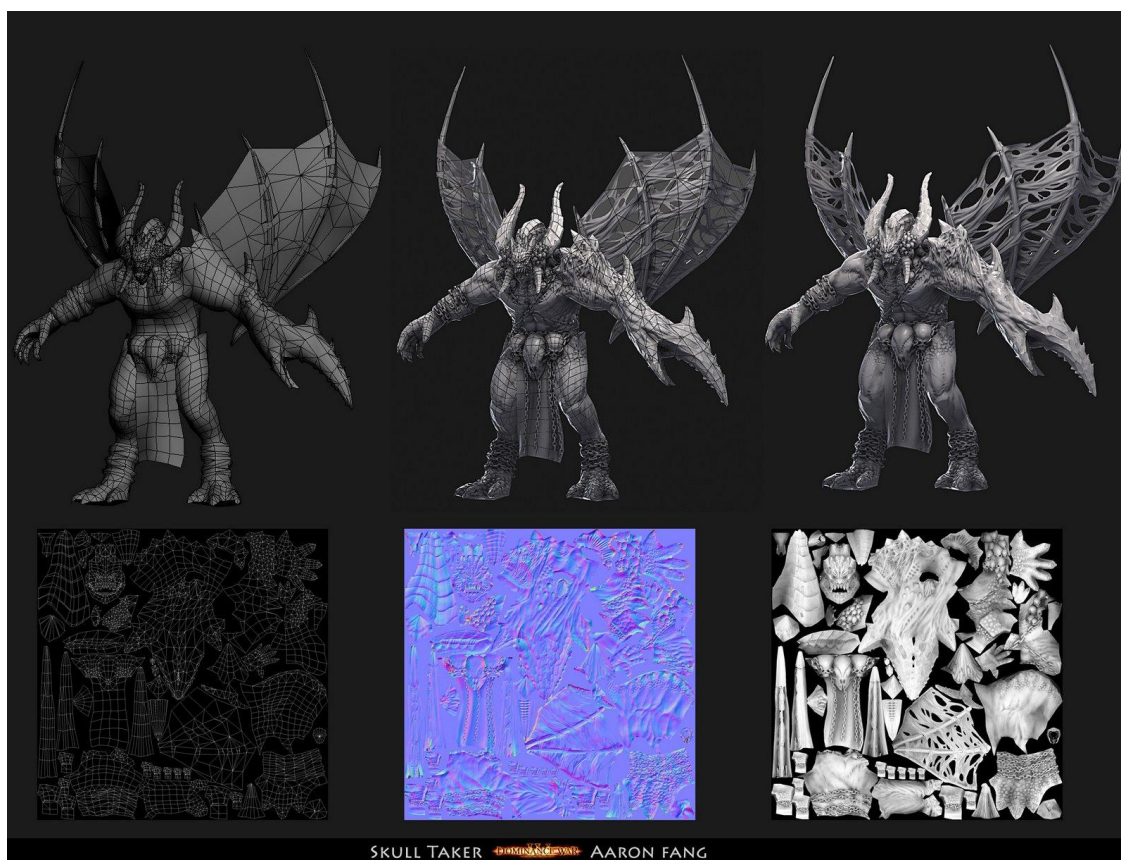
Pred importom musí byť model namapovaný a naanimovaný. Animácie sa prevádzajú na mieste z dôvodu, že pri vývoji sa objektu definuje jeho pohyb a jeho rýchlosť.

### *Normal mapping*

Hlavným účelom normáloveho mapovania je získať efekt dvoch miliónov polygónov za použitia len desiatich tisícov polygónov. Postupuje sa tak, že na low-poly model sa aplikuje jeho high-poly normálová mapa, získaná z jeho detailnej verzie.

Najčastejším spôsobom získania detailnej high-poly normálovej mapy je práve sculptovanie (viď. bod 1.3.1 - f). Model sa musí vyexportovať po jednotlivých objektoch, t.j. každý jeden objekt musí byť samostatný OBJ súbor. Do sculptovacieho programu sa importuje low-poly model vo formáte OBJ a v iteráciách zvyšovania hustoty meshu sa vytvárajú najjemnejšie detaily.

Normálovú mapu (obr. 24) je najvhodnejšie uložiť vo formáte .tif, ktorý je bezstratový. Po jej získaní sa aplikuje na low-poly model. Táto mapa nahradí pôvodnú low-poly normálovú mapu a vytvorí dojem high-poly modelu.



Obrázok 24: Ukážka normálovej mapy [11]

## 6 Unity3D

Unity3D je integrovaný vývojový nástroj pre tvorbu 3D hier alebo interaktívneho obsahu ako napríklad obchodné centrum alebo prototyp vozidla.

Hlavnou výhodou tohto nástroja je možnosť používania modelov z najpoužívanejších 3D programov (3d studio MAX, Maya, Cinema 4D alebo Blender) a ich zobrazenie priamo vo webovom prehliadači. Unity dokáže mimo iné importovať formáty FBX, OBJ alebo .max(3ds max) či .ma(Maya). Modely by mali byť rovnako zložité ako pri hrách to znamená low-poly prípadne mid-poly modely.

### 6.1 Rendering v Unity3D

Unity3D využíva k spracovaniu GPU namiesto CPU, ako to bolo zvykom napríklad u Flashu. To je hlavný dôvod prečo je možné v prehliadači používať zložité modely, zrovnateľné s modernými počítačovými hrami.

Tento engine využíva kombinovanie geometrie do skupín pre zníženie vykresľovacích volaní a disponuje stovkou vstavaných shaderov, ktoré definujú ako objekt vyzerá a ako reaguje na svetlo.

### 6.2 Nasvietenie

Nasvietenie je možné editovať priamo v Unity editore. Engine umožňuje použiť *dual lightmappig*: jedna mapa je použitá pre objekty, ktoré sú viac vzdialené a druhá pre tie, ktoré sa nachádzajú bližšie.

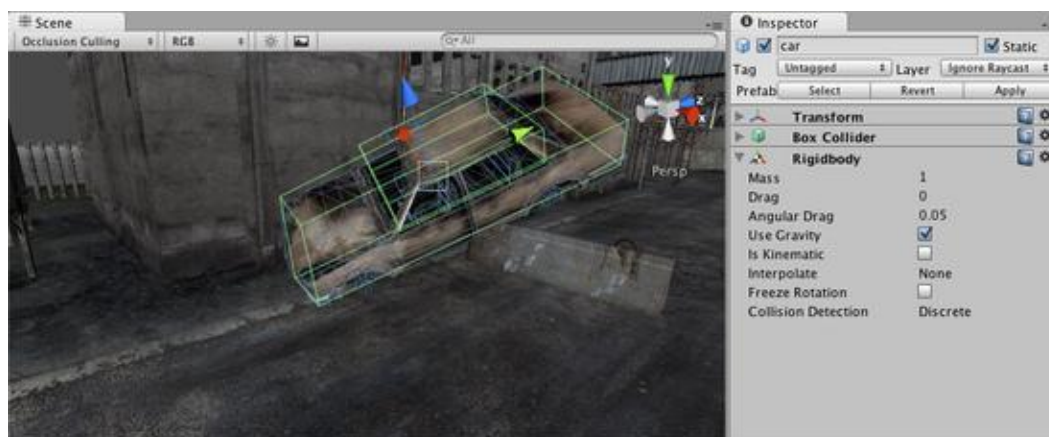
### 6.3 Terén

Terén sa môže tvoriť priamo v Unity editore a to modelovacou metódou sculpting (viď. bod 1.3.1 - f). Následne je možné na terén naniesť kamene, stromy alebo trávu. Pre tvorbu textúr sú v editore nástroje podobné photoshopu a na povrch sa nanášajú štetcom.

### 6.4 Fyzika

#### 6.4.1 Rigidbody(obr. 25)

Objekty, ktorým bola pridaná táto fyzika podliehajú gravitácii a iným silám skrze skriptovanie.



Obrázok 25: rigidbody [12]



### 6.4.2 Joints

Joints spájajú objekty dohromady a tvoria medzi nimi fyzikálne závislosti.

#### *Hinge*

Hinge (kĺb) spája dva objekty dohromady a vytvára medzi nimi vzťah. Výborne sa to hodí pre dvere alebo reťaz.

#### *Spring*

Spring (pružnosť) spája dva objekty dohromady a vytvára medzi nimi pružný vzťah. Znamená to asi toľko, že keď sa jeden objekt vzdiali od druhého, tento vzťah ho pritiahne späť.

#### *Character Joint*

Je to kĺb s možnosťou limitovať ho v akejkoľvek osi. Avšak ak na kĺb bude pôsobiť väčšia sila (Break Force) alebo väčší moment (Break Torque) než je stanovený, kĺb bude zničený a už nebude ďalej nič obmedzovať.

### 6.4.3 Soft Bodies

Unity3D dovoľuje vytvoriť z obyčajného objektu soft body objekt. Znamená to, že objekt sa bude schopný deformovať pri interakcii s okolím. Výborným príkladom je basketbalová lopta, ktorá sa pri styku s podlahou deformuje.

### 6.4.4 Wheel Collider

Wheel Collider je detektor kolízií určený predovšetkým pre kolesá vozidla. Tento detektor má vstavanú detekciu kolízií a šmykový model.

### 6.4.5 Kolízie šiat

Unity má dva druhy fyzikálne simulovaných šiat.

#### *Interactive Cloth*

Tento komponent simuluje reakciu látky na objekty v scéne. Obsahuje mnoho nastavení pre správne určenie materiálu látky, ako napríklad trenie, hustota ale aj napätie vo vnútri látky. Komponent je vhodný pre akúkoľvek látku v scéne.

#### *Skinned Cloth*

Tento komponent simuluje šaty charakteru a ich interakciu s okolím. Pracuje spolu s SkinnedMeshRenderer aplikovanom na objekte, ktorý nosí šaty. Jediné čo Skinned Cloth robí je, že si vezme výstup z SkinnedMeshRenderer a k nemu potom simuluje vertexy šiat.

## 6.5 Programovanie

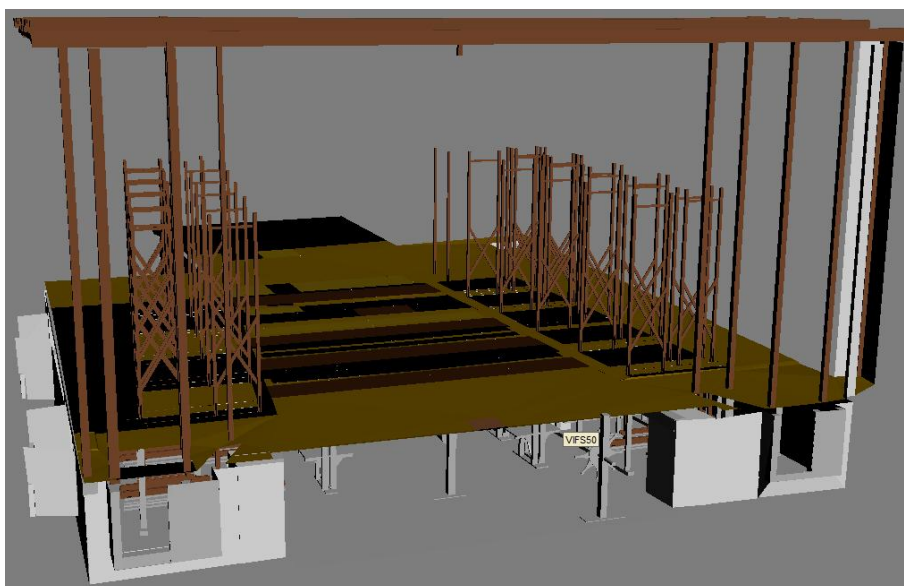
Pri tvorbe hier sa nikdy nedá vyhnúť troche kódu. Unity podporuje JavaScript, C# a Boo. Všetky 3 programovacie jazyky sú rovnako rýchle a môžu spolu kooperovať. Zároveň môžu využívať knižnice .NET, ktoré podporujú databáze, regulárne výrazy, XML, networking atď. Okrem iného Unity editor podporuje aj debugging. Informácie som čiastočne čerpal so stránok Unity3D (viď [13]).

## 7 Testovanie zozbieraných dát

Pre zdroj testovania som využil modely z diplomových prác. Všetky modely boli uchované vo formáte VRML, ktorý ako som spomenul na začiatku tejto práce, nie je vhodný pre uchovávanie modelov. Pre import tohto formátu som využil program 3ds max a verziu 2011, ktorá ako posledná ešte podporuje tento formát. To že formát nie je vhodný pre import sa prejavilo veľmi rýchlo a to v podobe nutnej úpravy VRML kódu. Síce sa model po úpravách importoval bez problémov ale treba mať na pamäti, že nie všetci, ktorí budú s modelom pracovať, sú schopní sa zorientovať v tisícoch riadkoch kódu a dokázať ho upraviť.

### 7.1 Živé divadlo

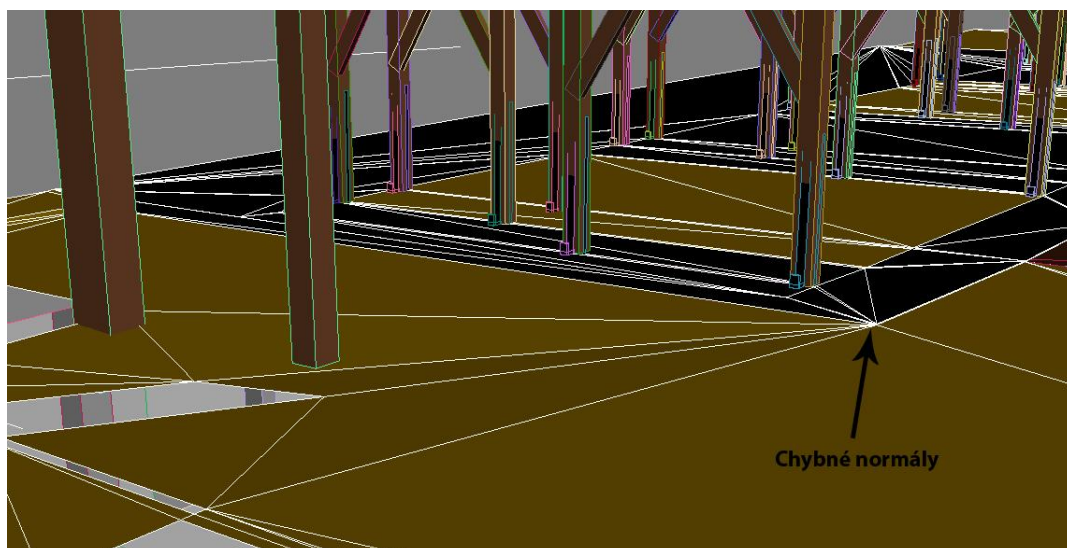
Model divadla bol vytvorený v roku 2005 a uložený vo formáte VRML. Skladá sa z 2184 objektov a 101 551 polygónov. Ako je vidno na obrázku 26, model nevyzerá až tak zle, ale skutočnosť je úplne iná. Takmer celé divadlo je poskladané z primitívnych telies, normály sú pootáčané, pivoty nie sú tam kde majú byť a mash je úplne neprehľadný. Po pravde ak by som mal niekedy tento model mapovať alebo animovať tak si ho vytvorím radšej znovu a ušetrím si hodiny/dni trápenia.



Obrázok 26: Živé divadlo

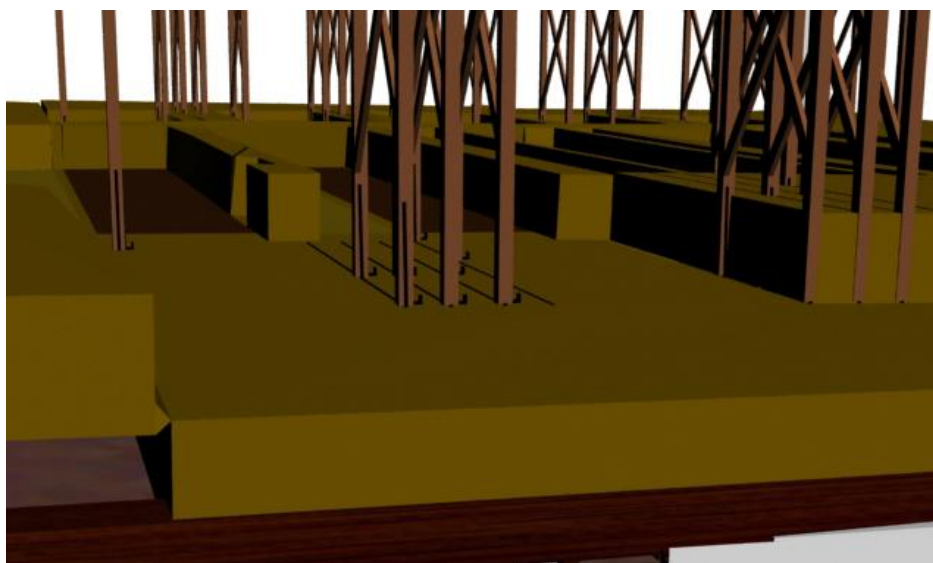
Na obrázku 27 je vidno ako sú na podlahe pootáčané normály (polygóny čiernej farby). Otočené normály spôsobil import formátu VRML do programu 3ds max. Otočené normály spôsobia, že polygóny nebudú vyrendrované a taktiež modifikátory sa nebudú správať tak ako majú.





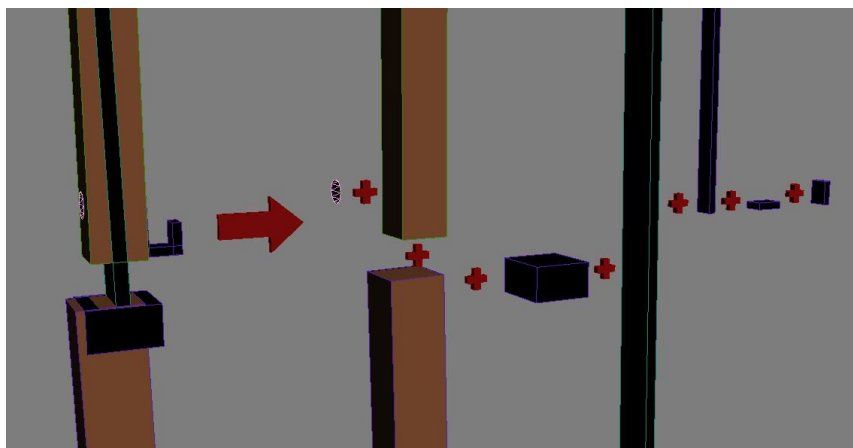
Obrázok 27: Chybné normály a mash

Keďže podlaha pódia je vytvorená ako obyčajná plocha, nemá hrúbku. Možným modifikátorom, ktorý by sa dal použiť na objekte je shell, ktorý dodá objektu hrúbku. Avšak pretože na objekte sú otočené normály, modifikátor nepracuje správne a vytvorí niečo iné než bolo plánom (obr. 28). Ďalším dôležitým problémom je mash podlahy. Topológia objektu je úplne nezmyselná a chaotická, čo má za následok sťaženie mapovania a v tomto prípade aj nárast počtu polygónov.



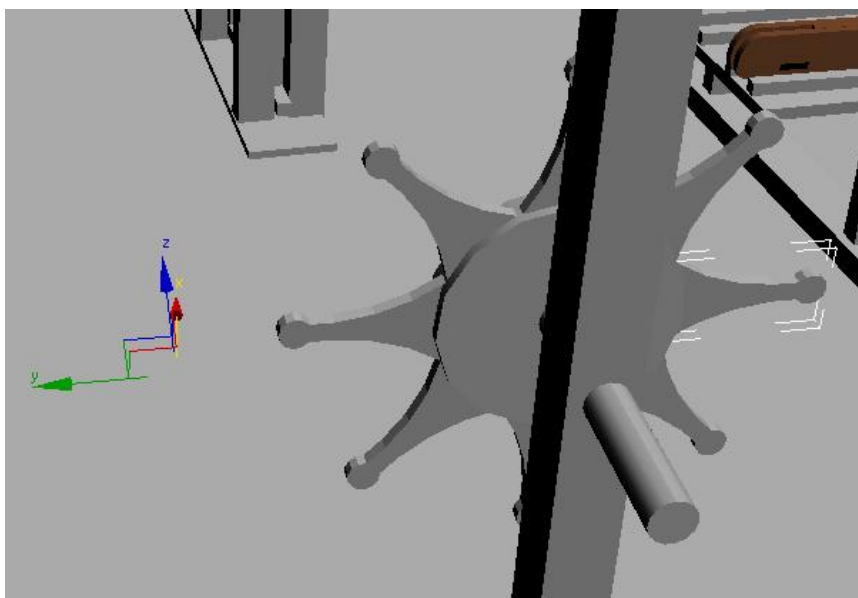
Obrázok 28: Vliv normál na modifikátor shell

Čo sa týka vytvorených modelov tak väčšina sa skladá z primitívnych telies. Na obrázku 29 je ukážka stĺpu. Tento stĺp je úplne nezmyselne vytvorený z ôsmich objektov a pritom by sa dal ľahko vyrobiť z piatich. Keďže celý model divadla je tvorený týmto štýlom je možné takýmto šetrením objektami vo výsledku dosiahnuť namiesto takmer 2200 objektov len napríklad 1200.



Obrázok 29: Stĺp, vytvorený z primitívnych telies

Poslednou chybou sú pivoty objektov. Nachádzajú sa mimo objektov, na miestach kde nemajú čo hľadať. Na obrázku 30 je príklad zle umiestneného pivotu. Model kola sa skladá z deviatich častí namiesto jednej a pivoty každej jednej časti sa nachádzajú úplne mimo objektu. Toto rozmiestnenie pivotov znemožňuje jeho animovanie a celkovú manipuláciu s ním.



Obrázok 30: Chybné umiestnenie pivotu

Čo sa týka vyexportovania modelu divadla, tak bol prevedený bez akýchkoľvek problémov do všetkých troch formátov.

## 7.2 Prírodné prostredie vo VRML

Diplomová práca z roku 2004 obsahuje 3 modely lodí uložených vo formáte VRML. Import do programu 3ds max 2011, ktorý ako posledný ešte podporuje tento formát, nebol hneď možný a bolo nutné previesť úpravy kódu (odstránenie LOD, prototypu a taktiež interaktívnej časti modelu).

### *FURIE*

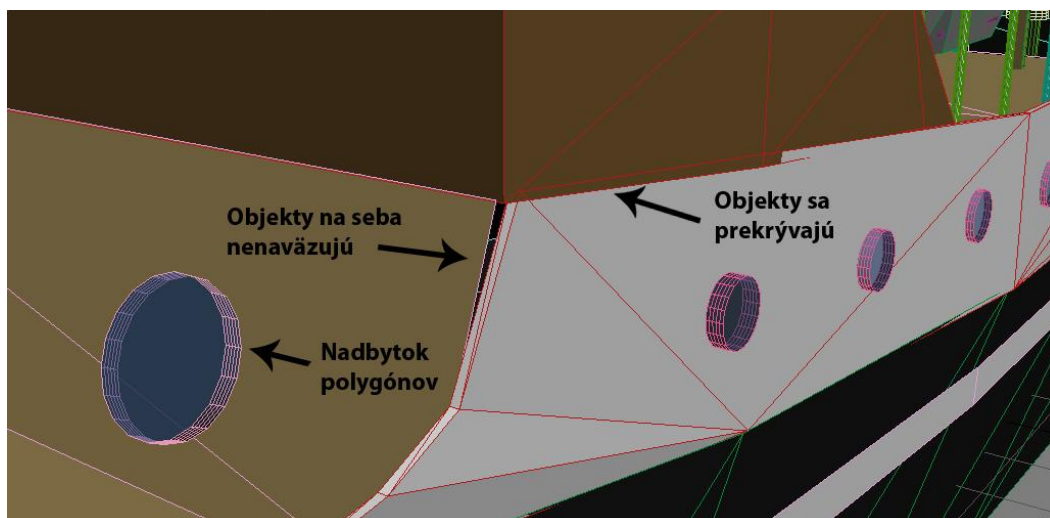
Prvým modelom je loď s názvom FURIE. Skladá sa z 263 objektov a 10580 polygónov. Topológia jednotlivých častí vyzerá omnoho lepšie než u Živého divadla. Pri importe sa objekty, ktoré prešli editáciou triangulovali, čo miestami zdeformovalo objekt inak než bolo pravdepodobne mienené. Na

obrázku 31 je znázornená takáto deformácia zadnej časti lode. Po triangulovaní 5-hrannej plochy vznikli 3 trojuholníky a plocha sa podľa nich zdeformovala nie na stred ale na ľavú stranu lode. Riešením by bolo vytvoriť dva 4-hranne polygóny a podľa nich zdeformovať názov lode.



Obrázok 31: Zdeformovaná zadná časť lode

Trup lode bol vytvorený z viacerých častí, čo spôsobilo, že sa jednotlivé časti prekrývajú alebo na seba nenaväzujú a tvoria sa medzi nimi medzery (obr. 32). Na obrázku 32 sa nachádza príklad zlého šetrenia polygónmi. Lodné okná sú vytvorené pomocou primitívneho telesa valec, a sú úplne zbytočne rozdelené piatimi čiarami po ich obvode.

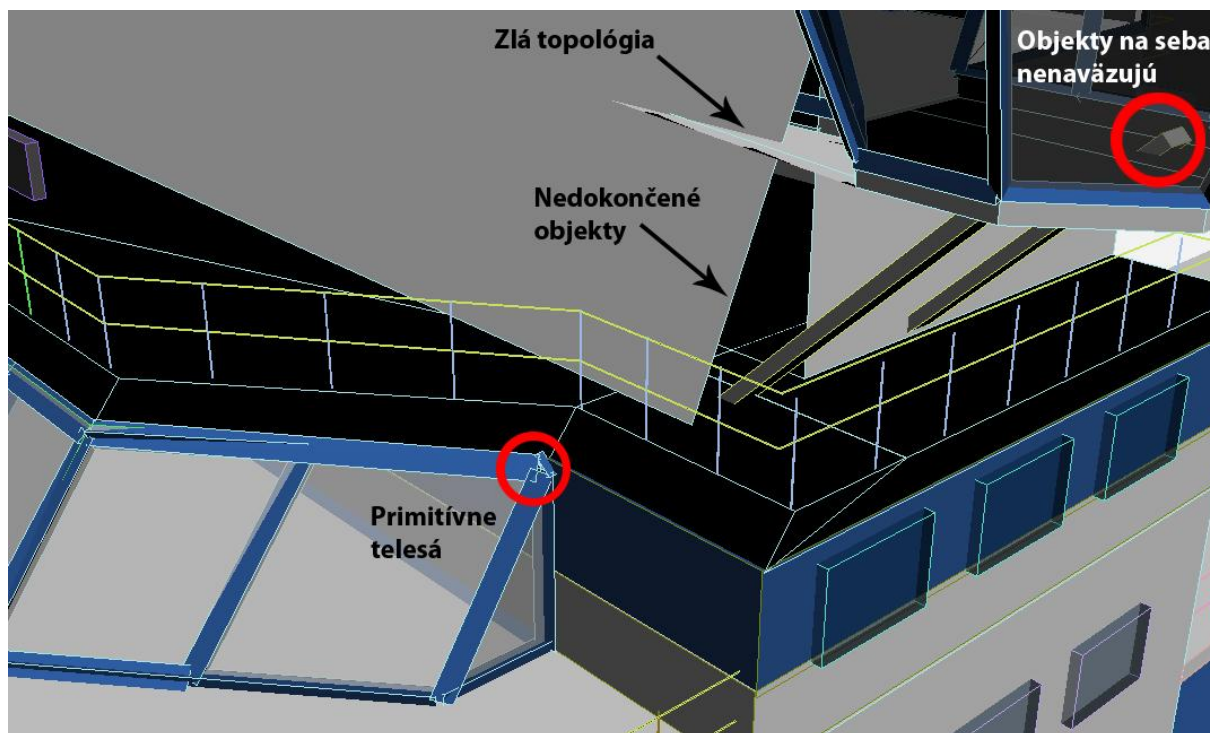


Obrázok 32: Nepresné umiestnenie objektov a nadbytok polygónov

Tento model nevyzerá až tak zle a po úpravách topológie, pridaní hrúbky niektorým častiam modelu by bol vhodný do nejakej jednoduchšej hry. Keďže sa jedná o jednoduchý model bez akýchkoľvek modifikátorov, bol model vyexportovaný do všetkých troch formátov bez problémov.

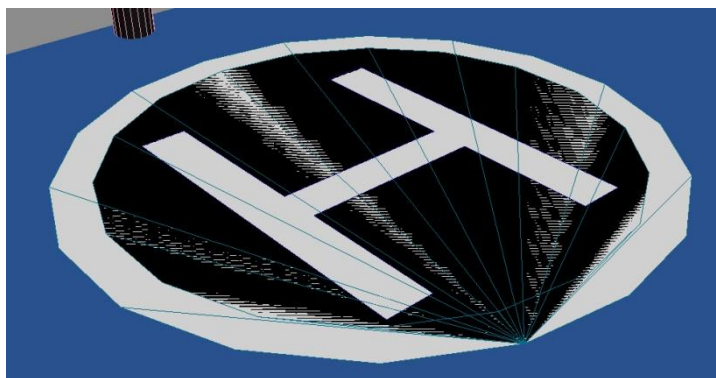
## KONTIO

Druhou loďou je ľadoborec KONTIO, vytvorený z 978 objektov a z 10645 polygónov. Import prebehol rovnako ako u prvej lode. Pri importe bola prevedená triangulácia, ktorá zvýraznila chybnú topológiu plavidla. Model je už o niečo rozsiahlejší a aj chýb je tam o dosť viac. Model bol tvorený z veľkej časti pomocou primitívnych telies ako je vidno aj na obrázku 33. Jednotlivé časti modelu na seba nenaväzujú a niektoré z nich dokonca obsahujú diery.



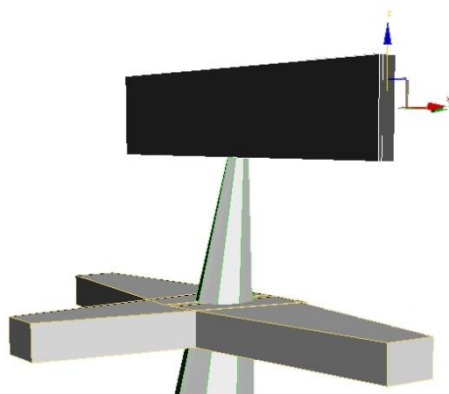
Obrázok 33: Časté chyby

Na obrázku 33 a 34 je zobrazená chybne spravená topológia. Pôvodne (pred trianguláciou) boli polygóny vytvorené miestami aj zo šiestnástich hrán namiesto zo štyroch. Po triangulovaní sa vytvorili polygóny, ktoré sú spojené v jednom bode a tým sa vytvorili plochy aj tam kde nemajú byť.



Obrázok 34: Triangulovaný objekt

Tvorca modelu asi nemyslel na možnosť animovania lode a pivoty nie sú na správnych miestach. Napríklad pivoty radarov (obr. 35) sú na strane objektu a nie v jeho strede (mieste otáčania).



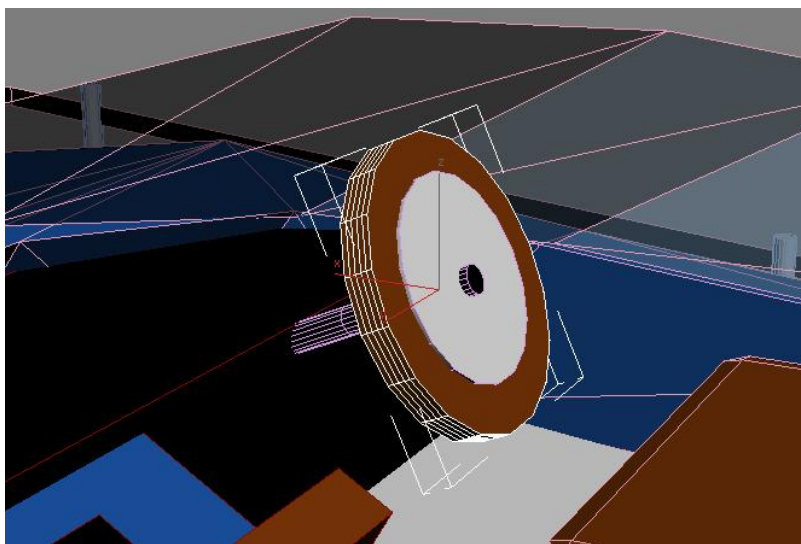
Obrázok 35: Chybné umiestnený pivot

Model je použiteľný, ale len s potrebnými úpravami topológie objektov, domodelovaním chýbajúcich častí a upravením pivotov. Po nutných úpravách by sa model dal využiť napríklad v hrách. Rovnako, ako predošlé plavidlo, export lode prebehol bez problémov .

### *MCLUN*

Posledným, tretím modelom je loď s názvom MCLUN. Toto plavidlo sa skladá z 83 objektov a 4564 polygónov. Jedná sa o doposiaľ najlepší model, ktorý už využíva editované objekty častejšie ako u predošlých modelov. Avšak ani tu nie je model dokonalý a má pár chýb, ktoré avšak nie sú až tak závažné, aby sa s ním nedalo neskôr pracovať.

Na obrázku 36 je zobrazené nadmerné použitie polygónov na volante plavidla. Je to rovnaký problém ako u lodných okien na prvom plavidle (obr. 32).



Obrázok 36: Nadmerný počet polygónov

Čo je na tomto plavidle pozitívne je, že pivoty objektov sú tam kde majú byť a model je teda ľahko použiteľný pre animovanie. Model je vhodný pre jednoduchšie hry. Export do všetkých troch formátov prebehol bez problémov.

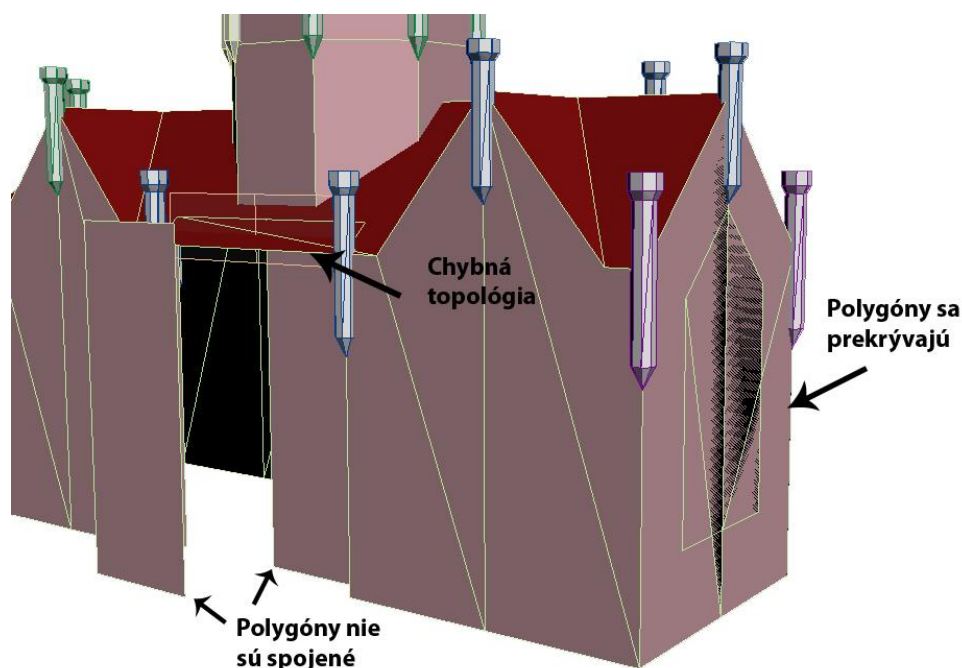


### 7.3 Model zámockého parku vo Vlašimi

V roku 2009 bol vytvorený zámocký park v jazyku VRML a obsahuje mnoho modelov, avšak o niektorých nemá zmysel hovoriť pretože sú veľmi jednoduché. U siedmich zložitejších modelov prebehol import bez problémov avšak mash objektov vyzerá veľmi zle. Chyby, ktoré na objektoch vznikli, boli pravdepodobne spôsobené importovaním a zlou topológiou objektu (polygóny s väčším počtom hrán ako štyri).

#### *BranaDo*

Model s názvom branaDo je vytvorený z 1614 polygónov, 27 objektov a 34 pomocných objektov (helpers). Topológia modelu je na veľmi zlej úrovni. Polygóny sa dosť často prekrývajú (obr. 37) a nie sú spojené. Aj napriek tomu, že je objekt v dezolátnom stave je veľmi rýchlo opraviteľný. Dôvodom je jednoduchosť modelu a časté opakovanie objektov (vežička).

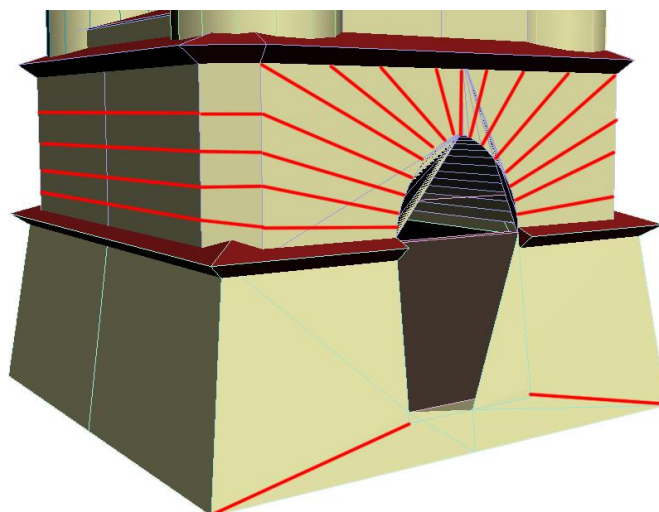


Obrázok 37: chyby na modely branaDo

Po nutných úpravách je model vhodný do jednoduchej hry alebo interaktívnej aplikácie. Export do všetkých troch formátov prebehol bez problémov.

#### *BranaVI*

BranaVI je model druhým modelom vo Vlašimi. Je vytvorený z 268 objektov avšak 172 z nich sú len pomocné. Počet polygónov je veľmi podobný ako u predošlého modelu a to 1864. Model má rovnaké chyby ako ten predošlý (chybná topológia, polygóny sa prekrývajú a nie sú pospájané). Niektoré časti ako napríklad vežičky, podpery alebo strechy vyzerajú dobre a nie je na nich čo vytknúť. Naopak prvé dve poschodia vyzerajú strašne ale po chvíli mazania a dopĺňovania polygónov je možné ho opraviť. Na obrázku 38 sú znázornené problémové objekty a ich možné riešenie.

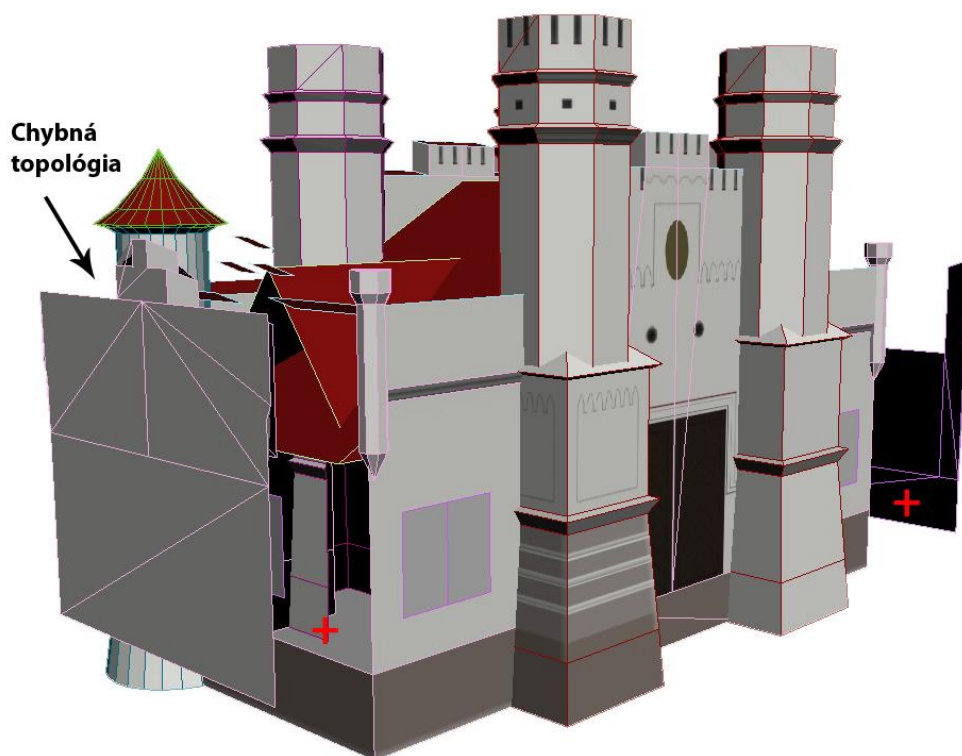


Obrázok 38: Problémové objekty a ich možné riešenie

Ako je vidno objekty boli pôvodne vytvorené z viac než štvorhranných polygónov. Po vymazaní nadbytočných polygónov a doplnení správnych čiar by po úprave vznikli už len štvorcové polygóny.

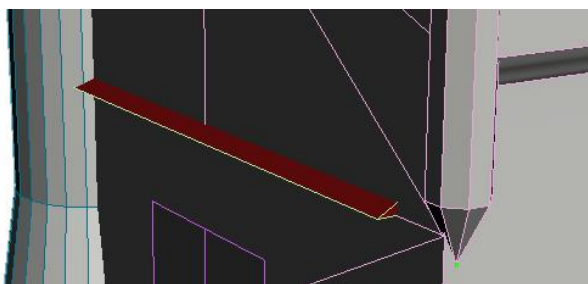
### *BranaZn*

Tretí model je pomenovaný ako branaZn a je vytvorený z 29 objektov a 1554 polygónov. Oproti predošlým dvom modelom vyzerá tento model na prvý pohľad o trochu lepšie, avšak je na tom rovnako ako ostatné. Topológia objektov je celkom v poriadku s výnimkou obvodových stien budovy, a prekvapením je, že sú rozdelené na 3 samostatné objekty (obr. 39). Tak ako u predošlých modelov aj tu sú polygóny nepospájané a miestami sa prekrývajú.



Obrázok 39: Chybná topológia a chybné vymodelované steny modelu branaZn

Ďalšou chybou je vyčnievajúca strecha z obvodovej steny budovy (obr. 40).



Obrázok 40: Vyčnievajúca strecha

Ak sa opravia chyby, ktoré sa na modely nachádzajú je možné ho ďalej použiť napríklad v jednoduchej hre ale dovedy je ako ostatné objekty nepoužiteľný.

### *Hrad*

Model hradu ma 727 polygónov a 30 objektov. Jedná sa o doposiaľ najjednoduchší model, ktorý je až na jeden objekt celkom dobre vymodelovaný. Daný objekt (obr. 41) vyzerá na prvý pohľad veľmi odstrašujúco ale je opraviteľný za pár minút. Ako aj u predošlých modelov aj tu je potrebné pospájať polygóny, ktoré sú od seba oddelené.

Model rovnako ako prvé 3 nie je vhodný na znovupoužitie. Po úprave topológie objektu by už bolo možné ho použiť v hre.

Uloženie modelu do formátov FBX, Collada a OBJ prebehlo opäť bez problémov.

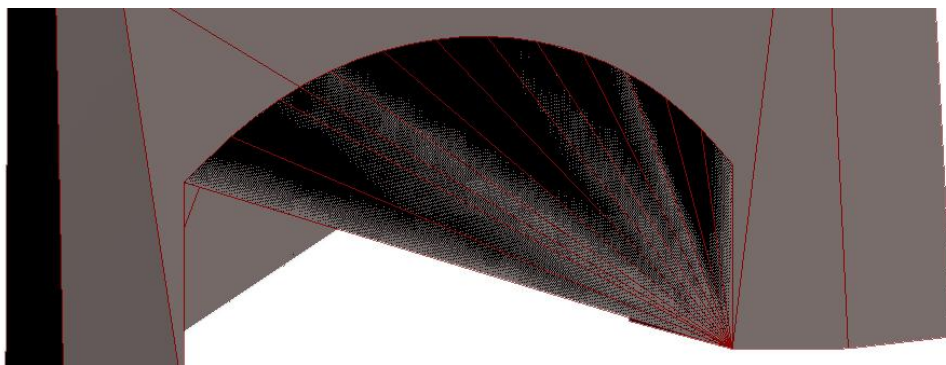


Obrázok 41: Zlá topológia modelu hrad

### *MostZn*

Ešte o niečo jednoduchším modelom je štvrtý model mostu z názvom mostZn. Je zložený len z 22 objektov a 462 polygónov. Bohužiaľ ani takto jednoduchý model nie je bez chyby. Opäť je tu problém s topológiou (obr. 42) a tým, že bol model triangulovaný sa vytvoril efekt podobný obrázku 34. Chyby sú ľahko opraviteľné, avšak ihneď použiteľné to nie je. Po nutných úpravách je model použiteľný napríklad v hrách alebo v Unity3D. Export modelu do 3 odporúčaných formátov prebehol bez komplikácií.



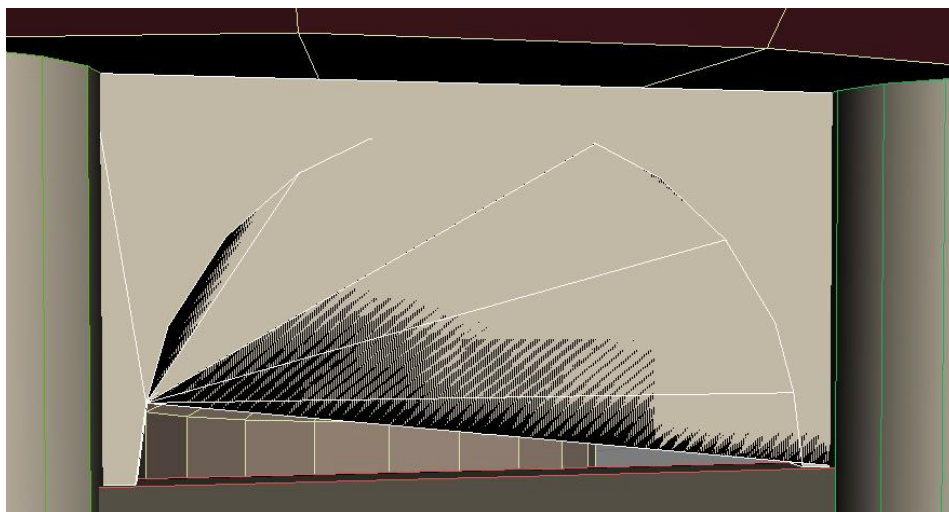


Obrázok 42: Chybná topológia mostu

## Pavilon

Model malého altánku s názvom pavilon je vytvorený z 57 objektov a 2410 polygónov. Ako je už pomaly zvykom u modelov vo Vlašimi aj tu nie sú polygóny prepojené, topológia objektov je rozhodená a objekty, ktoré by mali byť súčasťou jedného celku sú rozdelené do viacerých objektov.

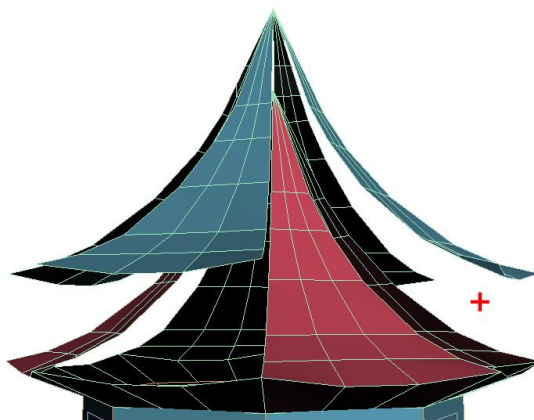
Na obrázku 43 je zobrazený už bežný problém a to zlá topológia. Polygóny boli pred exportovaním do VRML formátu vytvorené z viacerých hrán a to spôsobilo, že polygóny sa pri importe chybné vytvorili. Riešenie by mohlo vyzerať obdobne ako na obrázku 38.



Obrázok 44: Chybná topológia pavilónu

Ako som už naznačil objekty nepozostávajú z toho z čoho by mali. Na obrázku 44 je zobrazená strecha altánku, ktorá je rozdelená podľa farby na objekty. Je úplne zbytočne vytvárať ďalší objekt, keď stačí vytvoriť len dvojfarebný materiál pre jeden objekt.

Po upravení spomenutých chýb je možné model využiť v hernom priemysle ale bez oprav je nepoužiteľný.

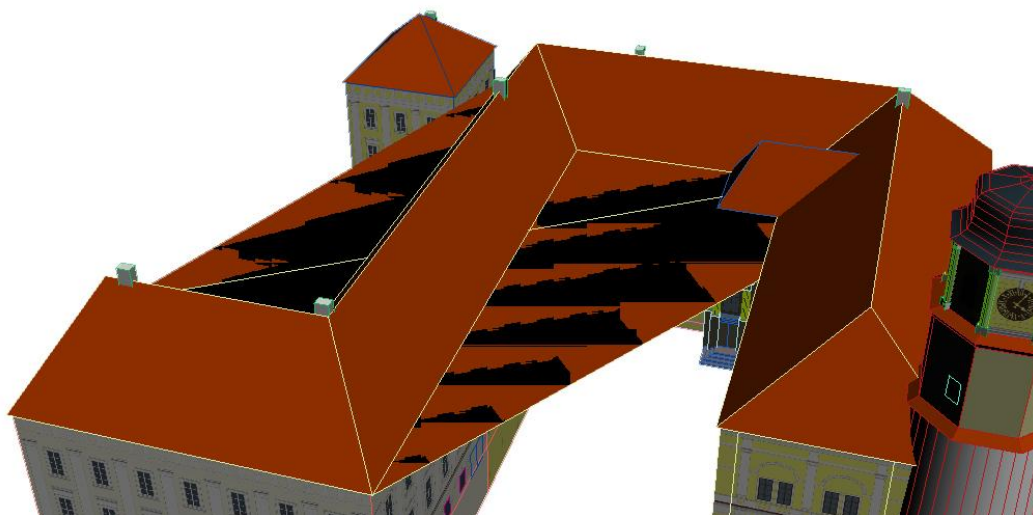


Obrázok 43: Strecha pavilónu vytvorená z dvoch objektov

## Zamek

Poslednou siedmou časťou je zámok vo Vlačimi a pozostáva zo 108 objektov a 2021 polygónov. Chyby, ktoré je potrebné odstrániť pred znovu použitím sú rovnaké ako u predošlých modelov, t.j. objekty s nespojeným polygónmi a zlá topológia.

Na obrázku 45 je znázornená chybná topológia strechy zámku. Tým, objekt nie je rozčlenený na dostatok polygónov, t.j. nachádzajú sa tam polygóny s väčším počtom hrán ako štyri, sa strecha po importovaní chybné vykreslila. Jedinou potrebnou úpravou, ktorá by danej situácii predišla, by bolo rozdelenie spodku strechy na štyri časti, alebo jeho úplné vymazanie.



Obrázok 45: chybná topológia zámku

Tak ako u predchádzajúcich objektov, aj tu je model bez nutnej úpravy strechy nepoužiteľný. Po uskutočnení nápravy je model vhodný do hier.

## 8 Messerschmit Bf109

V poslednej kapitole budeme hovoriť o modele nemeckého vojenského lietadla z druhej svetovej vojny (obr. 46). Lietadlo je zložené z 63 objektov a 6665 polygónov. Jedná sa o low-poly model vhodný do hier alebo pre unity3D. Aby bol model vhodný a znovu použiteľný pre rendering a animácie bol model vytvorený podľa pravidiel kapitolách 3 a 4. Keďže sa jedná aj o low-poly model, pri jeho tvorbe som musel zohľadniť aj pravidlá popísané v kapitolách 5 a 6.



Obrázok 46: Messerschmit Bf109

### 8.1 Rendering

Lietadlo bolo vytvorené v programe 3ds max 2011 a vyrendrované za pomoci rendrovacieho systému V-ray.

#### 8.1.1 Podklady

Vytvárať existujúce lietadlo bez referencií je úplne nemožné a tak aj pri tomto modeli som musel vyhľadať fotky a blueprinty (viď príloha B). Pri nastavovaní blueprintov mi pomohli aj skutočné rozmery lietadla. Všetky tieto informácie som našiel na stránkach Cybermodeler (viď. [14])

#### 8.1.2 Modelovanie

Ako techniku modelovania som zvolil polygonálne modelovanie, ktoré je popísané v bode 3.3.1.d. Začal som modelovať s najväčšou časťou (trupom lietadla), aby šli ostatné časti na trup lepšie napasovať a nevznikali nepresnosti. Všetky časti sú vytvorené zo štvorhranných polygónov podľa obrázku 5. Topológiu som tvoril rovnomerne aby sa na jednej strane netvorili veľké medzery a na druhej príliš úzke miesta. Tento model má byť vhodný pre hry a Unity3D takže som sa snažil obmedziť počet polygónov tak ako to len šlo.

#### 8.1.3 Osvetlenie

Rozloženie svetiel som zvolil podobné ako na obrázku 11. Na začiatku som si vytvoril kameru z pohľadu pre ktorý som nastavoval osvetlenie. Hlavné svetlo som umiestnil od pohľadu kamery smerom k lietadlu, prvé vedľajšie svetlo som umiestnil nad model, druhým som jemne osvetlil chvost lietadla a tretie a štvrté svetlo som umiestnil po stranách lietadla.

#### 8.1.4 Rendering

V nastaveniach rendru som nastavil nepriame osvetlenie s nízkymi hodnotami a malé rozlíšenie rendru aby som pri testovaní materiálov nečakal príliš dlho na výsledok.

#### 8.1.5 Mapping

Predposlednou časťou bolo mapovanie popísané v bode 3.6. Väčšina objektov bola namapovaná pomocou Unwrapu (viď obr. 17) a následne otextúrovaná v programe Photoshop. Textúry som tvoril z referenčných fotografií a s použitím rôznych nástrojov v programe.

#### 8.1.6 Finálny render

Pre finálny render som už len zvýšil hodnoty nepriameho osvetlenia a rozlíšenie rendru.

### 8.2 Animácie

Aby bol objekt ľahko animovateľný musí splňovať všetky náležitosti popísané v bode 4. Pivot každého objektu ktorý je statický má pivot v strede objektu inak sa nachádza v mieste otáčania. Pre uľahčenie animácie je každý objekt nalinkovaný na helper typu point (bod). Tento bod sa nachádza v strede objektu a umožní jednoduchšiu manipuláciu s objektom.

### 8.3 Low-poly modelovanie

Po vytvorení low-poly modelu a jeho uložení do formátu OBJ, som do programu Z-Brush importoval trup lietadla na ktorom som vytvoril pár detailov. Následne som vygeneroval normálovú mapu, ktorú som priradil materiálu trupu. Výsledný efekt je na obrázku 47. Pri porovnaní zmien s predchádzajúcim obrázkom sú zmeny, ktoré boli prevedené, zjavné. Len pripomínam, že počet polygónov zostal rovnaký a jediné čo pribudlo sú detaily.

Export a aj import modelu prebehol bez problémov a teda až doposiaľ je model znovu použiteľný.



Obrázok 47: Model s aplikovanou normálovou mapou

### 8.4 Unity3D

Posledným krokom je otestovanie modelu pre Unity3D. Import modelu do prostredia softwaru prebehol bez akýchkoľvek problémov. Jedinou nutnou úpravou, umožňujúcou s modelom ďalej pracovať, bolo vytvorenie prázdneho objektu do ktorého sa model vložil. Pre lepšie otestovanie modelu som vytvoril v jazyku javascript skript s ovládaním lietadla. Na záver som modelu priradil ešte skript, ktorý umožňuje manipulovať s modelom rovnako ako v prostredí programu 3ds max. Oba skripty fungujú bez akýchkoľvek problémov, čiže model lietadla splňuje aj posledný bod a teda je znovu použiteľný.

## 9 Záver

Pravidlá, zostavené v tejto práci sú návodom ako vytvoriť kvalitný model, ktorý bude použiteľný aj v budúcnosti. V tejto práci nebolo možné vymenovať všetky problémy, ktoré sa môžu vyskytnúť. Snažil som sa však popísať bežné problémy, ktoré sú spojené s 3D tvorbou, ich riešenie a prípadne aj spôsob ako sa im dá vyhnúť. Pravidlá som rozdelil do 4 skupín použiteľnosti a pre lepšiu demonštráciu problémov som využil obrázky. V závere práce som otestoval modely z diplomových prác. Najčastejším problémom objavujúcim sa u modelov bola chybná topológia. Medzi ďalšie problémy jednoznačne patrili zle rozdelené objekty, nekompletné objekty, zbytočné plytvanie polygónmi a chybné umiestnenie pivotov. Keďže pravidlám nevyhovoval ani jeden z modelov vytvoril som model 100% znovu použiteľného lietadla. Model som vytvoril tak aby bol použiteľný pre hry respektíve Unity3D. Aby som dokázal, že je model použiteľný aj v hrách vytvoril som ešte skript s ovládaním a následne nechal vygenerovať náznak hry. Práca by mohla pokračovať vytvorením kompletnej hry pre plug-in Unity3D.



## Literatúra

- [1] Autodesk FBX Help for 3ds Max. *download.autodesk.com*. [Online] 2010.  
[http://download.autodesk.com/us/fbx/20112/3dsmax/\\_index.html](http://download.autodesk.com/us/fbx/20112/3dsmax/_index.html).
- [2] **Bourke, Paul**. Object Files (.obj). *paulbourke.net*. [Online] 2011.  
<http://paulbourke.net/dataformats/obj/>.
- [3] **Northwood, Chris**. Computer Graphics and Visualisation. *pling.org.uk*. [Online] 2009.  
<http://www.pling.org.uk/cs/cgv.html>.
- [4] Heroic Male Body 3d model. *turbosquid.com*. [Online] 2005.  
<http://www.turbosquid.com/FullPreview/Index.cfm/ID/273487?referral=web3Dservice>.
- [5] Catmull-Clark\_subdivision\_of\_a\_cube.svg. *upload.wikimedia.org*. [Online] 2006.  
[http://upload.wikimedia.org/wikipedia/commons/3/33/Catmull-Clark\\_subdivision\\_of\\_a\\_cube.svg](http://upload.wikimedia.org/wikipedia/commons/3/33/Catmull-Clark_subdivision_of_a_cube.svg).
- [6] **AL, Karthik**. Finished Work - 3D Stills. *www.cgтанtra.com*. [Online] 2008.  
<http://www.cgтанtra.com/forums/showthread.php?t=12248>.
- [7] Bacht studio technics in Beijing . *www.bacht.net*. [Online] 2006.  
[http://www.bacht.net/english/aktuell01\\_e.html](http://www.bacht.net/english/aktuell01_e.html).
- [8] UVW Map Modifier. *3dmax-tutorials.com*. [Online] [http://www.3dmax-tutorials.com/UVW\\_Map\\_Modifier.html](http://www.3dmax-tutorials.com/UVW_Map_Modifier.html).
- [9] **Primachenko, Yaroslav**. Making Of Zombie In The Closet. *blog.bttrove.org*. [Online] 2011.  
<http://blog.bttrove.org/2187>.
- [10] realflow.jpg. *randommodel.files.wordpress.com*. [Online]  
<http://randommodel.files.wordpress.com/2010/09/realflow.jpg>.
- [11] **Fang, Aaron**. NormalMap\_P.jpg. *2.bp.blogspot.com*. [Online] 2009.  
[http://2.bp.blogspot.com/\\_nmF398SGnUk/Se6\\_axMe8fl/AAAAAAAAARg/\\_LBSGlovils/s1600-h/NormalMap\\_P.jpg](http://2.bp.blogspot.com/_nmF398SGnUk/Se6_axMe8fl/AAAAAAAAARg/_LBSGlovils/s1600-h/NormalMap_P.jpg).
- [12] Rigidbody. *unity3d.com*. [Online] 2011.  
<http://unity3d.com/support/documentation/Components/class-Rigidbody.html>.
- [13] Unity3D. *unity3d.com*. [Online] 2005. <http://unity3d.com/>.
- [14] Bf 109, HA 1112M Buchon, S-199. *cybermodeler.com*. [Online] 2011.  
[http://www.cybermodeler.com/aircraft/bf109/bf109\\_all.shtml](http://www.cybermodeler.com/aircraft/bf109/bf109_all.shtml).





## Príloha A

### Zoznam použitých obrázkov

Obrázok 1: Beziér krivka [3] .....	6
Obrázok 2: Tvorba NURBS plochy .....	6
Obrázok 3: Orotovanie krivky okolo osy .....	7
Obrázok 4: Pretiahnutie krivky profilom .....	7
Obrázok 5: Príklad správnej a nesprávnej topológie.....	8
Obrázok 6: Topológia charakteru [4].....	8
Obrázok 7: Sub-division modelovanie [5] .....	9
Obrázok 8: Sub-division level – vliv topologie na level .....	9
Obrázok 9: Sub-division level – vliv levelu na topológiu .....	9
Obrázok 10: Iterácie sculptingu [6] .....	10
Obrázok 11: Nasvietenie v profesionálnom štúdiu [7].....	11
Obrázok 12: druhy gizma [8] .....	12
Obrázok 13: Planar mapping [8].....	12
Obrázok 14: Cylindrical mapping [8] .....	12
Obrázok 17: UNWRAP [9].....	13
Obrázok 15: Box mapping [8] .....	13
Obrázok 16: Face mapping [8].....	13
Obrázok 18: animácia pohľadu.....	15
Obrázok 19: IK solver.....	16
Obrázok 20: Kostra postavy.....	17
Obrázok 21: Obálka kosti .....	18
Obrázok 22: Animácia objektu po trajektórii .....	18
Obrázok 23: animácia vodopádu v programe RealFlow [10] .....	19
Obrázok 24: Ukážka normálovej mapy [11] .....	21
Obrázok 25: rigidbody [12].....	22
Obrázok 26: Živé divadlo .....	24
Obrázok 27: Chybné normály a mash .....	25
Obrázok 28: Vliv normál na modifikátor shell.....	25
Obrázok 29: Stĺp, vytvorený z primitívnych telies .....	26
Obrázok 30: Chybné umiestnenie pivotu .....	26
Obrázok 31: Zdeformovaná zadná časť lode.....	27
Obrázok 32: Nepresné umiestnenie objektov a nadbytok polygónov .....	27
Obrázok 33: Časté chyby .....	28
Obrázok 34: Triangulovaný objekt .....	28
Obrázok 35: Chybne umiestnený pivot .....	29
Obrázok 36: Nadmerný počet polygónov .....	29
Obrázok 37: chyby na modely branaDo .....	30
Obrázok 38: Problémové objekty a ich možné riešenie.....	31
Obrázok 39: Chybná topológia a chybne vymodelované steny modelu branaZn.....	31
Obrázok 40: Vyčnievajúca strecha .....	32
Obrázok 41: Zlá topológia modelu hrad .....	32

Obrázok 42: Chybná topológia mostu .....	33
Obrázok 44: Chybná topológia pavilónu .....	33
Obrázok 43: Strecha pavilónu vytvorená z dvoch objektov .....	33
Obrázok 45: chybná topológia zámku .....	34
Obrázok 46: Messerschmit Bf109 .....	35
Obrázok 47: Model s aplikovanou normálovou mapou .....	36

## **Zoznam použitých skratiek**

**2D** Two-Dimensional

**3D** Three-Dimensional

**VRML** Virtual Reality Modeling Language

**NURBS** Non-uniform rational B-spline

**CPU** Central processing unit

**GPU** Graphics processing unit

## **Zoznam použitého softwaru**

3ds max 2011

Autodesk Maya 2011

Z-Brush 4

Blender

RealFlow 5

Photoshop CS4

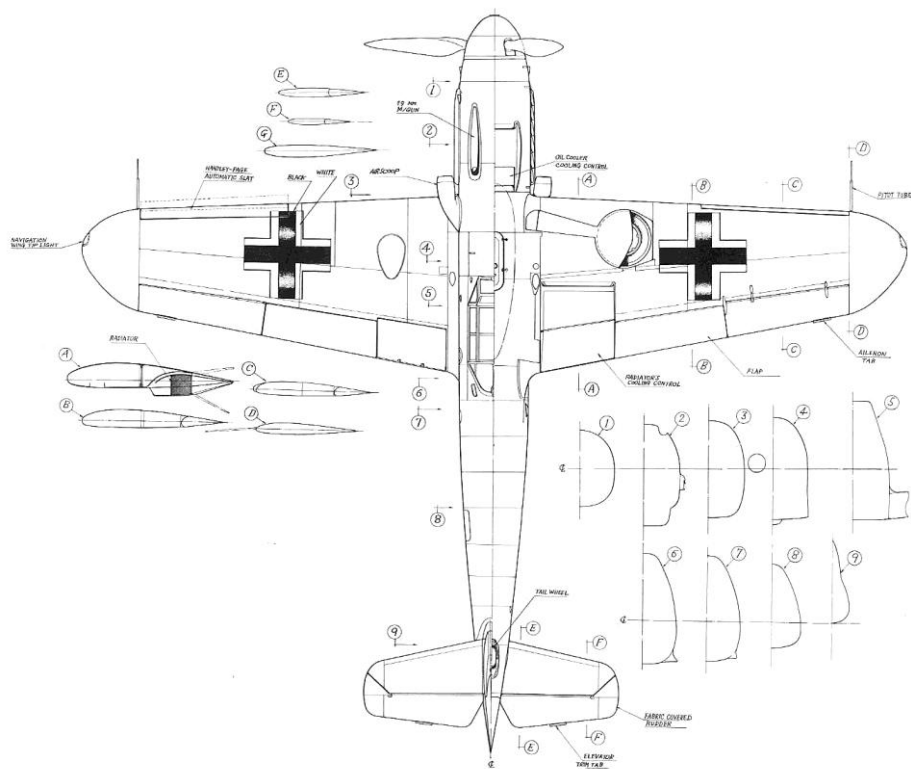
Unity3D

Microsoft Word 2007

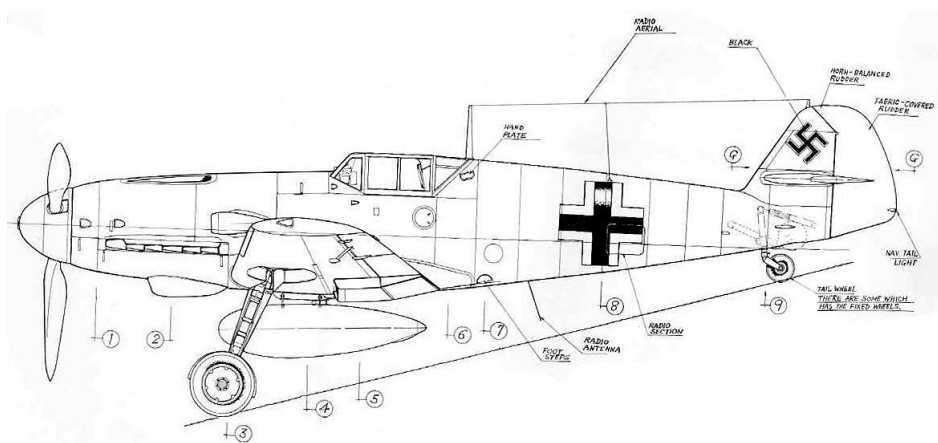
## Príloha B

### Blueprinty

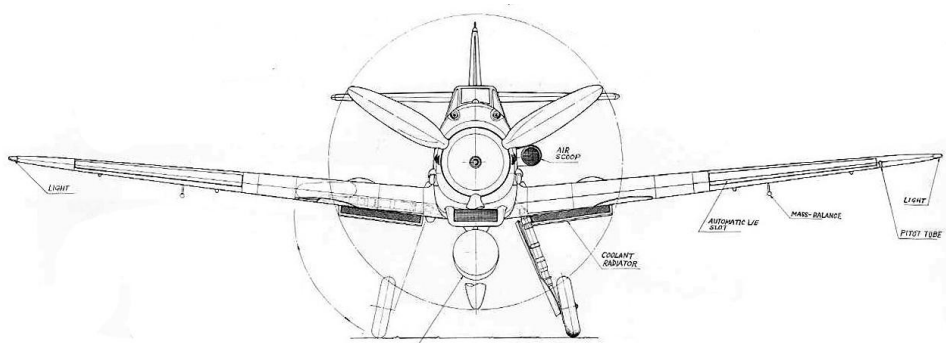
#### 1. Pohľad zhora a zospodu



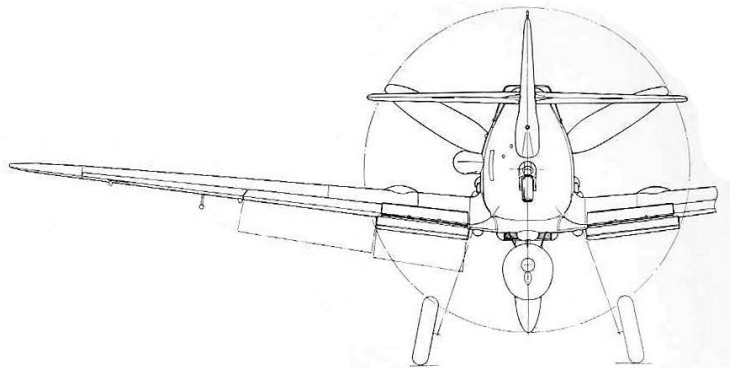
#### 2. Pohľad z boku



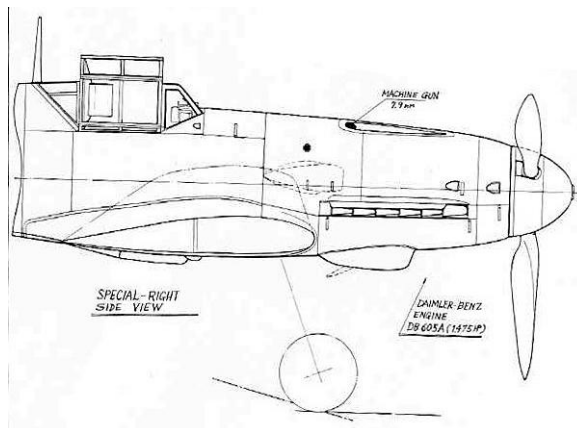
#### 3. Pohľad spredu



4. Pohľad zozadu



5. Pohľad z boku 2



## Príloha C

### Obsah priloženého CD

**Model** – Adresár obsahuje model vojenského lietadla, blueprinty a textúry.

**Testovacie scény** – Adresár obsahuje testované modely.

**Text** – Obsahom tohto adresáru je text tejto bakalárskej práce.

**Unity** – Adresár obsahuje aplikáciu s lietadlom.

V **roote** priloženého CD sa taktiež nachádza súbor **README.TXT**, ktorý obsahuje informácie k spusteniu aplikácie a informácie o obsahu CD.