



## Pelihahmon toteuttaminen

Viestintä  
3D-animaatio ja -visualisointi  
Opinnäytetyö  
19.5.2010

---

Senja Heikkinen

## TIIVISTELMÄSIVU

Koulutusohjelma Viestinnän koulutusohjelma	Suuntautumisvaihtoehto 3D-animaatio ja visualisointi	
Tekijä Senja Heikkinen		
Työn nimi Pelihahmon toteuttaminen		
Työn ohjaaja/ohjaajat Kristian Simolin		
Työn laji Opinnäytetyö	Aika 19.5.2010	Numeroidut sivut + liitteiden sivut 41+5
<p>TIIVISTELMÄ</p> <p>Tässä toiminnallisessa opinnäytetyössä toteutettiin pelihahmo lähtien liikkeelle suunnittelusta ja käyden läpi mallinnusprosessia aina valmiiseen reaaliaikaisessa ohjelmassa toimivaan hahmomalliin asti. Tavoitteena oli tutkia, mitä kaikkea on otettava huomioon, kun rakennetaan hahmomallia reaaliaikaiseen pelimoottoriin, joka tässä opinnäytetyössä oli Unreal 3.</p> <p>Ohjelmat, joita opinnäytetyön teossa pääasiassa käytettiin, olivat Autodesk 3Dsmax mallintamiseen, Autodesk Mudbox tiheäverkostoisen mallin veistämiseen, Adobe Photoshop tekstuurien tekoon ja Unreal Development Kit reaaliaikaisen version kokoamiseen.</p> <p>Suunnitteluvaiheessa tutkin, mitä kaikkea on hyvä ottaa huomioon ennen mallinnusvaiheen aloittamista. Mallinnusvaiheesta kävin läpi, kuinka rakensin hahmomallin päämääränä tietty polygoniraja. Kerron myös UV-kartoituksesta ja teksturoinnista ja hieman luuranko-mallin rakentamisesta. Lopuksi kasasin hahmon Unreal 3:n Unreal Development Kit editorin sisällä toimivaksi reaaliaikaiseksi hahmomalliksi.</p> <p>Lopputuotoksena liitteenä on asennusohjelma, joka asentaa tiedoston, jossa hahmoa voi tarkastella reaaliajassa.</p>		
Teos/Esitys/Produktio <i>Reaaliaikainen 3D-hahmo</i> (UDK-tiedostot, asennettava ohjelma (.exe) ja videokuvakaappauksia)		
Säilytyspaikka Metropolia Ammattikorkeakoulu, Tikkurilan toimipiste		
Avainsanat pelihahmo, 3Dsmax, UDK		

Degree Programme in <b>Media</b>		Specialisation <b>3D animation and visualisation</b>
Author <b>Senja Heikkinen</b>		
Title <b>Making a Game Character</b>		
Tutor(s) <b>Kristian Simolin</b>		
Type of Work <b>Bachelor's Thesis</b>	Date <b>19.5.2010</b>	Number of pages + appendices <b>41+5</b>
<p>The objective was to design and model a game character that would run in a real-time game engine. The aim was to gain better understanding of the modeling process and the workflow of making a game character.</p> <p>3Dsmax was used for the modeling, Mudbox for creating a high-polygon version of the model for the normal maps and Unreal Development Kit (UDK) to combine all elements together in a real-time environment. Different aspects of what is required to make a good, working character were identified. The thesis itself follows the path of making a game character from design to a working real-time model.</p> <p>It was discovered that there are many things you need to take into account when modeling a game character for real-time purposes, like the limits in the polycount and the sizes of the textures. And to make the character work inside the UDK the way you want, there are many things that are good to know beforehand.</p> <p>Optimizing graphics, 3D and other elements without compromising quality, is one key element of building up a good game character. Understanding the workflow and planning the process well beforehand is important for faster and more successful modeling.</p>		
Work / Performance / Project <b>A real-time 3D character (UDK files, installer (.exe) and screencapture videos)</b>		
Place of Storage <b>Helsinki University of Applied Sciences, Tikkurila Campus Library</b>		
Keywords <b>game character, 3Dsmax, UDK</b>		

# SISÄLLYS

1	JOHDANTO.....	2
2	TYÖN TAVOITTEET .....	3
2.1	Vahvan perusteen rakentaminen alusta alkaen .....	4
2.2	Tekninen toteutus.....	4
3	KESKEISIÄ KÄSITTEITÄ.....	6
4	PELIHAHMON LUOMINEN JA TOTEUTTAMINEN .....	7
4.1	Hahmon suunnittelu.....	7
4.1.2	Hahmon taustan suunnittelu .....	8
4.1.3	Ulkonäön suunnittelu .....	8
4.2	Huomioonotettavaa ennen mallintamisen aloittamista .....	9
4.2.1	Mallikuvan tulkinta .....	10
4.2.2	Topologian suunnittelu .....	10
4.2.3	Unreal Development Kit.....	13
4.3	Mallintaminen .....	14
4.3.1	Low-poly.....	15
4.3.1	High-Poly .....	21
4.4	Yksityiskohtaisemman geometrian illuusio normaalikartalla.....	23
4.5	UV-kartat ja tekstuurit.....	25
4.5.1	UV-kartat .....	25
4.5.2	Tekstuurit .....	27
4.6	Riggaus .....	30
4.7	Reaaliaikainen hahmomalli.....	32
4.7.1	UDK ja mallinnetun hahmon siirtäminen Unreal Editoriin .....	33
4.7.2	Vaatesimulaatio UDK:ssa .....	34
4.7.3	Normaalikartat ja materiaalit UDK:ssa.....	35
4.7.4	Animaatio.....	37
4.7.5	Lopullinen reaaliaikainen malli.....	37
5	POHDINTAA .....	38
	LÄHTEET .....	41
	LIITTEET	

## 1 JOHDANTO

Opinnäytetyöni käsittelee pelihahmoksi sopivan hahmon suunnittelua ja mallinnusprosessia, sekä hahmon sijoittamista Unreal 3:n pelimoottoriin. Pelihahmon mallintaminen on prosessi, jossa täytyy huomioida monenlaisia asioita. Tulen käymään vaiheet läpi alusta alkaen hahmon suunnitteluprosessista aina lopulliseen reaaliaikaiseen hahmomalliin, joka pyörii reaaliajassa Unreal 3 -pelimoottorissa.

Tavoitteenani on käydä läpi, mitä kaikkea pitää ottaa huomioon, kun rakentaa hahmomallia ulkopuolista moottoria varten ja miten asioita sovelletaan käytännössä. Samalla mallinnan hahmon niin hyvin kuin osaan ja käyn läpi mallinnusprosessia.

Ohjelmina tulen käyttämään Autodesk 3D Studio Maxia mallintamiseen, Adobe Photoshopia teksturointiin, Autodesk Mudboxia mallin yksityiskohtien luontiin, sekä Unreal Development Kitiä, eli UDK:ta reaaliaikaiseen toteutukseen. Helpotan myös UV-kartoitusprosessia käyttämällä ilmaisohjelmaa RoadKill.

Työskentelyvaiheet jakautuvat karkeasti kolmeen vaiheeseen: suunnitteluun, mallinnukseen ja pelieditorissa tehdyn materiaalin kasaamiseen toimivaksi kokonaisuudeksi.

Suunnitteluvaiheessa käyn läpi asioita, jotka helpottavat tulevaa mallinnusurakkaa, esimerkiksi hahmomallin topologian suunnittelua erityisesti kasvojen osalta. Hyvin

suunniteltu projekti on aina helpompi toteuttaa. Käyn myös läpi lyhyesti hahmon suunnitteluprosessia.

Mallinnusvaiheen tavoitteena on käydä läpi kaikki vaiheet, mitä hahmon tekeminen valmiiksi sisältää. Tämä tarkoittaa prosessikuvausta siitä, kuinka lähdin rakentamaan hahmoa kohti ennalta määrättyä polygonirajaa. Teen hahmosta kaksi eri versiota, harva- sekä tiheäverkostoiset (low- ja high-poly)-mallit. Mallinnuksen lisäksi käyn läpi UV-kartoitusta, teksturointia sekä riggausta.

Pelihahmon toteuttamiseen toimivaksi reaaliajassa vaatii tutustumista pelimoottoriin ja siihen, kuinka se toteuttaa asioita. 3D Studio Maxissa tehty pelihahmon mallintaminen, teksturointi ja animointi on vain osa tehtävää työtä, jotta hahmo toimisi reaaliaikaisessa ympäristössä. Mallinnohjelmasta pelieditoriin viety hahmomalli on pelieditorissa kuin eloton nukke, kunnes editorin sisällä siihen luodaan elämä taas uudestaan. Pelieditorissa täytyy tehdä vielä paljon asetuksia, jotta hahmo saa oikeanlaiset materiaalit ja animaatiot paikoilleen, ja näitä asetuksia varten täytyy jo mallinnusvaiheessa ottaa erilaisia asioita huomioon.

Opinnäytetyön projektiosuuden lopputuotos tulee olemaan reaaliaikaisessa sovelluksessa toimiva hahmo.

## 2 TYÖN TAVOITTEET

Tavoitteenani on pureutua syvemmin työkaluihin ja asioihin, mitä pelialalla – ja miksei muuallakin, käytetään hahmon mallinnukseen ja kaikkeen muuhun siihen liittyvään, kuten teksturointiin. Pyrin hiomaan osaamistani ja oppimaan uusia keskeisiä asioita, jotka liittyvät hahmomallin rakentamiseen topologiasta lähtien. Lähtötasoltani olen suhteellisen kokenut hahmojen mallintamisessa, ehkä enemmän low-poly-versioissa, sekä hahmojen teksturoinnissa. Normaalikarttojen teosta sekä Mudboxin käytöstä minulla ei ole kokemusta. Tavoitteenani on tehdä hahmomalli parhaan osaamiseni mukaan niin teknisesti, kuin visuaalisestikin.

Opinnäytetyöni aihetta motivoi kiinnostukseni peliteollisuutta kohtaan. Olen jo kauan

ollut kiinnostunut peleistä ja pelien 3D-grafiikka on aina kiehtonut minua, siksi aion syventää osaamistani tekemällä opinnäytetyön aiheesta. Tavoitteenani on kasvattaa tietotaitoani mallintamisesta ja reaaliaikaisista sovelluksista ja siitä, kuinka nämä kaksi toimivat yhteen.

## 2.1 Vahvan perusteen rakentaminen alusta alkaen

Aion oppia ymmärtämään hahmon mallinnusta syvällisemmin. Tähän kuuluu topologiaan tarkempi paneutuminen, edge-looppien ja lihaksiston yhteys, johon käytän avukseni anatomiakirjoja sekä erilaisia tutoriaaleja ja mallikuvia kokeneiden 3D - mallintajien hahmojen topologiasta. On tärkeää myöhempien vaiheiden osalta tehdä hahmo alusta alkaen ajatuksella, jotta vaikeuksilta vältyttäisiin.

Itse hahmo tulee olemaan hieman erilainen ja sisältämään joitain elementtejä, mitä en ole aiemmin tottunut mallintamaan. Tällä tavalla saan hieman haastetta ja joudun rakentamaan hahmomallin, joka olisi ennen tuottanut minulle ongelmia.

## 2.2 Tekninen toteutus

Opinnäytetyön tavoitteena on, että mallintamani hahmomalli toimii reaaliaikaisessa pelimoottorissa, joka tässä tapauksessa on Unreal 3 -pelimoottori. Tarkastelin, miten suurista polygonimääristä on kyse kyseisellä pelimoottorilla tehdyissä peleissä kuten Gears of War. Julkaisussa Character Modeling 2 (Lanning, Petroc & Baysal 2007, 39) todetaan Gears of War pelin käyttävän hahmomalleihin noin 10 000 – 15 000 polygonia. Opinnäytetyötäni varten asetan tavoitteekseni samanlaisen luvun hahmoni polygonimääräksi.

Miten paljon polygoneja käytetään hahmomalleihin, riippuu hyvin monesta eri asiasta. Ensimmäinen rajoittava tekijä on pelimoottori ja alusta, jolla peli tulee pyörimään. PlayStation 3 -tasoista grafiikkaa ei voi esimerkiksi pyörittää käsikonsoleilla kuten PlayStation Portable tai Nintendo DS. Seuraavaksi huomioon on otettava pelin tyyli; tuleeko ruudulla näkymään satoja hahmoja samaan aikaan, vai vain muutama hahmo

kerrallaan. Myös tekstuurikarttojen koot täytyy ottaa huomioon pelimoottoria varten. Tekstuurikartoista kerrotaan lisää luvussa 4.5.2.

Mallinnan hahmon käyttäen ohjelmaa 3D Studio Max. Mallinnusprosessi tulee koostumaan low- sekä high-poly -malleista. Low-poly -malli on se versio, joka tulee näkymään reaaliajassa. High-poly -mallia käytetään rakentamaan normaalikartat, jotka luovat illuusiota yksityiskohtaisemmasta mallista valon osuessa siihen. Mallintamisesta ja normaalikartoista kerrotaan tarkemmin luvuissa 4.3 ja 4.4.

Tavoitteena on saada hahmomalli pyörimään reaaliaikaisena sovelluksessa, eli tässä tapauksessa Unreal 3 –pelimoottorilla käyttäen tekemiseen Unreal Development Kitiä, eli UDK:ta. Tarkoituksena on tutustua Unreal Development Kitin editoriin, jolla kasaan lopullisen mallin reaaliaikaiseksi paketiksi.



### 3 KESKEISIÄ KÄSITTEITÄ

EDGEFLOW / POLYFLOW = jatkumo millä tavalla edget tai polygonit kulkevat hahmossa tai objektissa

HIGH-POLY = Tiheäverkostoinen malli. Mallinnuksessa käytetty tarpeeksi polygoneja yksityiskohtien luontiin

LOW-POLY = Harvaverkostoinen malli 3D:ssä. Yksinkertainen versio hahmosta/objektista

MATERIAALI = tekstuuriin, normaalikarttojen ja monen muun yhteistulos

MATERIAL ID = Joka polygonille määriteltävä numero, jonka mukaan eri materiaaleja voi sijoittaa samaan objektiin

NORMAALIKARTTA = Valo- ja varjokartta. Rakennettu high- ja low-poly-malleista ja luo low-poly-mallin pinnalle illuusiota yksityiskohtaisemmasta rakenteesta valon osuessa siihen

POLE = 3, 5 tai useamman edgen kohtauspäätteen. 4 edgen kohtauspäätettä pidetään yleensä normina, joten sitä ei kutsuta Poleksi

POLYGON = Neljän pisteen(verteksin) kokonaisuus, pisteiden väliin syntyvä pinta

REAALIAIKAINEN 3D-GRAFIikka = Välitöntä interaktiivista katsojan kanssa. Katsoja voi halutessaan kääntää kamerakulmaa tai tehdä muuta interaktiivista 3D-grafiikan kanssa. Tämä ei esimerkiksi onnistu esirendattujen animaatiopätkien kanssa

RIGGAUS = Luurakenteen tekeminen hahmolle

SMOOTHING GROUPS = Joka polygonille määriteltävä ryhmä, jolla määrätään polygonipintojen tasaisuutta

TEKSTUURI = Kaksiulotteinen kuva, joka kääritään 3D-mallin ympärille. Tekstuuri voi esimerkiksi olla kuva ovesta, lattiasta tai vaikka maalattu kuva hahmon osista

TOPOLOGIA = Polygonirakenteen kulku koko hahmossa tai objektissa

TRIGON = 3D-ohjelmat jakavat polygonit automaattisesti kolmioihin. Neljän kulman polygonista tulee kaksi Trigonina. Yleensä 3D-ohjelmat eivät automaattisesti näytä Trigoneita, vaan niiden näyttämiseksi pitää muuttaa asetuksia

UDK = Unreal Development Kit, käyttää Epi Gamesin Unreal 3 -pelimoottoria

UV-KARTTA = Liitetään geometriaan. Kertoo kaksiulotteiselle tekstuurille miten levittyä 3D-objektin pinnalle

## 4 PELIHAHMON LUOMINEN JA TOTEUTTAMINEN

Tämä luku käsittelee opinnäytetyöni projektiosuutta. Käyn vaihe vaiheelta läpi pelihahmon toteuttamista suunnitteluprosessista alkaen reaaliajassa toimivaan hahmomalliin saakka.

Pelihahmon mallintaminen on lähtökohtaisesti samanlaista kuin minkä tahansa muun animaatiohahmon mallintaminen, eroja tulee vastaan lähinnä rajoituksissa. Pelihahmon tulee toimia reaaliaikaisesti, mutta toisin kuin animaatiohahmoa, sitä ei samalla tavalla esirenderöidä. Riippuen alustasta, jolle hahmoa rakennetaan, rajoitukset ovat suuremmat tai pienemmät. Esimerkiksi mobiilisovelluksissa toimiville hahmoille on hyvin tiukat rajoitteet esimerkiksi polygonimäärissä, kun taas suuremman luokan tietokone- tai nykykonsolipelien (PlayStation 3, Xbox 360) polygonimäärät ovat todella korkeat ja käytettävien reaaliaikaisten erikoisefektien määrä suurempi.

Hahmoja rakennettaessa täytyy myös priorisoida. Pelin päähenkilöille, pelistä riippuen, sallitaan suurempi polygonimäärä kuin sivuhenkilöille. Mitä useammin ja mitä lähempää hahmoa tullaan pelin aikana katsomaan, sitä tärkeämpää on tehdä siitä elävämpi ja yksityiskohtaisempi.

Koska samanlaista esirenderöintiä ei pelihahmon kohdalla tehdä, kuin animaatiofilmeissä, ja jälkikäteen ulkonäköön ei muutoksia voi samalla tavalla tehdä, täytyy toimia pelimoottorin puitteissa. Tämä tarkoittaa, että kaikki efektit, joita lopulliseen teokseen tulee, tapahtuu pelimoottorin ominaisuuksien rajoissa. Pelimoottori tässä opinnäytetyössä on Unreal 3, josta lisää luvuissa 4.2.3 ja 4.7.

### 4.1 Hahmon suunnittelu

Opinnäytetyön prosessi alkaa hahmon suunnittelusta, jonka jälkeen mietitään toteutusta mallinnuksen kohdalla. En suunnittele hahmoa mihinkään olemassa olevaan peliin, vaan käytän siinäkin omaa mielikuvitusta, sillä pidän hahmosuunnittelusta enkä halua rajoittaa mielikuvitustani. Jo hahmon aikaisessa suunnitteluvaiheessa miellyn vahvaan persoonallisuuteen, joka tulee näkyä hahmon ulkoasussa ja tiettyihin visuaalisiin elementteihin jotka olen halukas mallintamaan. Halusin hahmosta naisen,

koska vahvat naishahmot ovat kiinnostavia, ja peleissä alkaa olla enemmän ja enemmän naishahmoja pääosissa. Pelien tunnettuja sankarinaishahmoja ovat muun muassa Bayonetta ja Final Fantasy XIII:n päähenkilö Lightning, puhumattakaan Lara Croftista, joka on pelimaailman suurimpia naistähtiä.

Lähdin hahmosuunnittelussa siitä lähtökohdasta, että hahmo on tietyn tyyppinen vahva naishahmo. Tämän jälkeen pohdin suuntaa-antavaa taustatarinaa tarkempaa visuaalista suunnitteluvaihetta varten. Yleensä oikeissa isoissa peliprojekteissa hahmoilla olisi valmiina taustatarinat ja muut, ennen kuin sen visuaalista ulkomuotoa suunniteltaisiin.

#### 4.1.2 Hahmon taustan suunnittelu

Jotta hahmon visuaalisen ilmeen sekä animaatioiden suunnittelu olisi johdonmukaista myöhemmissäkin vaiheissa, suunnittelin hahmolle lyhyen taustatarinan ja mietin mahdollista genreä pelille.

Jo alun perin halusin, että hahmo sijoittuu fantasiamaailmaan, joten sen perusteella muodostui mielessäni maailma kuninkaineen, ritareineen ja lohikäärmeineen. Genreksi ajattelin seikkailu- tai roolipeliä.

Suunnittelin pikaisen taustatarinan ennen visuaalisen ilmeen suunnittelua, jotta olisi jokin lähtökohta mistä edetä. Hahmosta tuli soturinainen, jonka tulee näyttää elegantilta, naiselliselta, mutta tarpeeksi uhkaavalta.

Hahmo ei saanut olla ilman nimeä, joten nimesin hänet Erika van Gerebrandiksi. Samalla sain opinnäytetyölle projektinimen: Erika, jota käytän myös pohjana työtiedostojen nimeämiselle.

#### 4.1.3 Ulkonäön suunnittelu

Seuraavaksi täytyi suunnitella hahmon ulkonäkö. Halusin myös lisätä elementtejä, jotka vihjaavat Erikan kytköstä lohikäärmeisiin. Tein useita luonnoksia, joista mikään ei tuntunut olevan juuri sitä ilmettä mitä hahmolta halusin. Joistain puuttui eleganssia,

joistain vakuuttavaa soturimaisuutta. Pikkuhiljaa löysin kuitenkin elementtejä, joita pidin mielenkiintoisina ja sopivina, joten aloin rakentaa lopullista ratkaisua näiden ympärille. Saatuani viimein kasaan haluamani lopputuloksen tein hahmosta puhtaaksi piirretyn version (kuva 1).

Lopullisena pitämäni kuva (kuva 1) ei kuitenkaan jäänyt lopulliseksi sillä suunnittelun ja mallinnuksen välissä halusin lisätä hahmolle lisää haarniskaa jalkoihin sekä yläruumiiseen, jotta hahmo näyttäisi vaikuttavammalta. Pitäydyin kuitenkin perus fantasiahaarniskassa, jossa ulkonäkö voittaa käytännöllisyyden.



Kuva 1: Puhtaaksi piirretty kuva hahmosta

## 4.2 Huomioonotettavaa ennen mallintamisen aloittamista

Hahmon suunnittelun jälkeen ennen mallintamisen aloittamista on hyvä suunnitella tuleva urakka etukäteen. Hyvä suunnitelma nopeuttaa mallinnusprosessia ja vähentää mahdollisia tulevia ongelmia.

### 4.2.1 Mallikuvan tulkinta

Kun suunnitellaan pelihahmoa, on hyödyllistä käyttää konseptitaiteita apuna niin paljon kuin mahdollista ja tutkia minkälaisista palasista hahmo koostuu. Hahmo kannattaa paloitella jo paperilla osiin ja tulkita mallikuvasta mitkä osat ovat yhtenäisiä, mitkä irrallaan. Selkeimmät irto-osat ovat Erikassa esimerkiksi haarniska.

Kirjassa Character Modeling 2 (Lanning ym. 2007) kerrotaan mallikuvan tutkimisen ja pilkkomisen helpottavan mallinnusprosessia. Samalla selviää, mitkä osat voidaan uudelleen kierrättää vanhoista malleista sekä eritellä orgaaniset ja kovet pinnat toisistaan (Lanning ym. 2007, 23).

Jos hahmo on monimutkainen, sitä ei kannata mallintaa yhtenäiseksi objektiksi, vaan sen voi rakentaa osissa. Osissa rakentaminen helpottaa mallinnusta niin, että hahmoon voi keskittyä yksi osa kerrallaan ja tarvittaessa osat voidaan piilottaa pois näkyvistä.

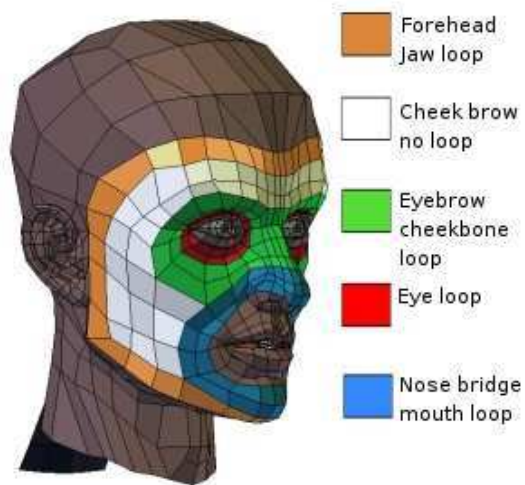
### 4.2.2 Topologian suunnittelu

Aloitin hahmon mallinnuksen osittain valmiiksi suunnitellun topologian pohjalta, sillä suunnitelman tekeminen helpottaa mallinnusurakkaa suuresti. En suunnitellut koko mallia täydellisesti etukäteen, vain vaikeimmat ja erikoisimmat alueet ajan säästämiseksi. Tärkeää oli kuitenkin saada sujuva ja hyvin animoituva rakenne aikaiseksi. Tärkeämmiksi alueiksi luokittelen monimutkaiset ja eniten liikkuvat osat, eli jalkojen ja keskiruumiin väli, kinalot ja yläruumis sekä kasvot. Näiden alueiden hyvä toteutus takaa paremman deformaation animaatio-vaiheessa.

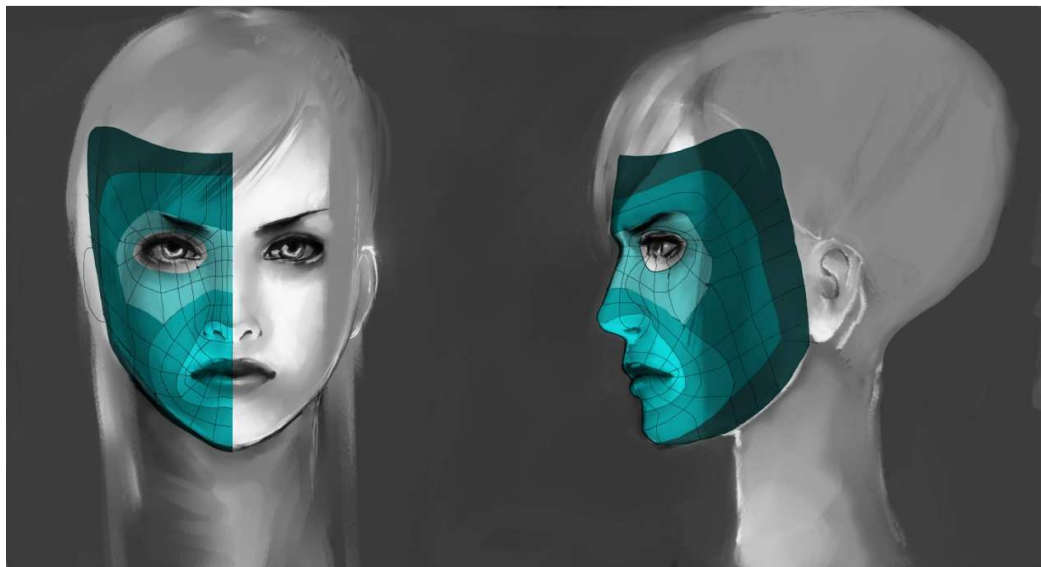
Jos hahmo on orgaaninen, mahdollisimman lähellä alastonta ihmisruumista, silloin mallintamista helpottaa anatomian ja lihasryhmien tuntemus. Tiesin, että hahmoni

keskiruumis tulee olemaan ihonmyötäistä kangasta, jolloin tutkin sen alueen lihaksistoa ja etsin internetistä mallikuvia hyvännäköiseen ja toimivaan topologiaan. Päädyin kuitenkin lopulta erottelemaan lihaksia vasta high-poly-vaiheessa.

Tein hahmosta sivu-, etu- sekä takakuvat, joihin hahmottelin tulevaa topologiaverkostoa mielestäni hankalimpiin paikkoihin. Tarkimmat topologiaohjeet tein kuitenkin kasvoista.



Kuva 2: kasvojen topologia lihasryhmien mukaan (Blender Artist Forum 2008)



Kuva 3: Kasvojen topologian suunnitelma mallinnusta varten.

Kasvojen mallintamista varten tein erityistarkan piirroksen tulevasta topologiasta. Etsin internetin kautta mallikuvia hyvästä kasvojen topologiasta ja päädyin käyttämään

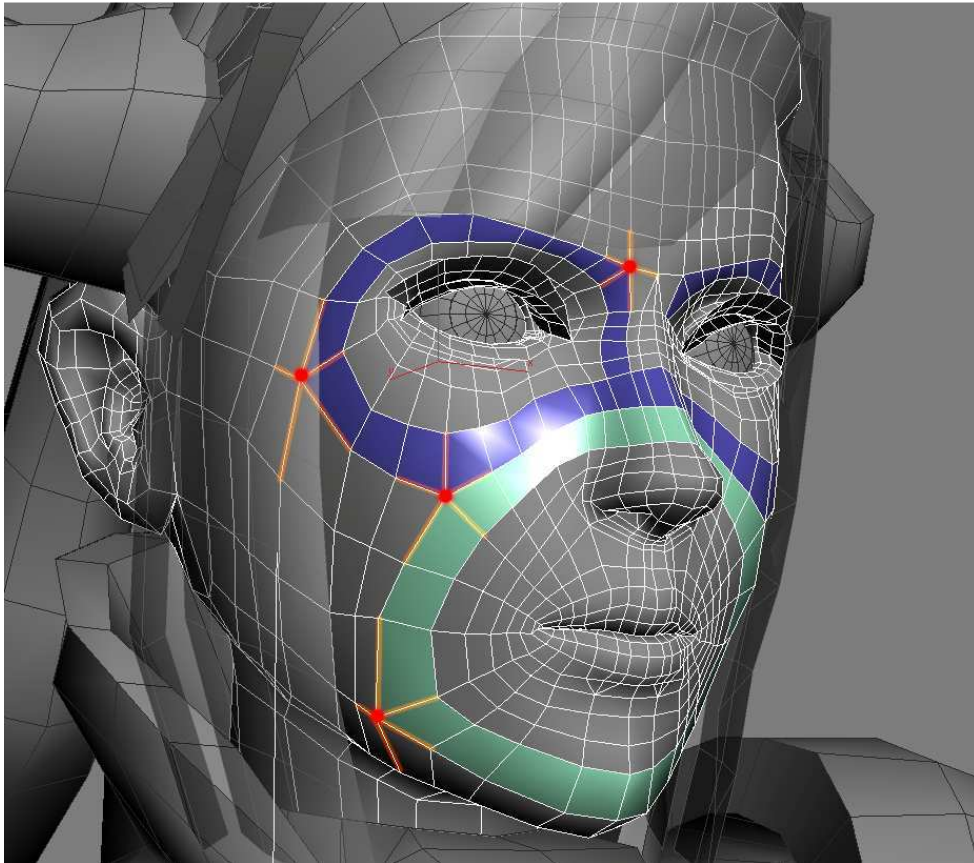
versiota, jossa kasvot oli jaettu alueisiin (kuva 2). Kyseinen malli mukailee kasvojen lihaksistoa ja siten myös animoituu myöhemmin hyvin. Tein mallikuvien perusteella suunnitelman toteutettavasta verkostosta kasvoissa (kuva 3).

Ihmiskasvoissa on paljon lihaksia, joilla välitämme tunnetiloja. Kasvoanimaation uskottavuutta ja sulavuutta tukee hyvin toteutettu topologia, jossa polyflow mukailee kasvojen lihasryhmiä.

Kasvojen mallintamisen voi tehdä monella eri tavalla ja aloitustapoja on monia. Yksi tapa on mallintaa laatikko ja lisätä siihen pikkuhiljaa tarvittavaa geometriaa samalla, kun muotoilee sitä pään muotoon. Toinen tapa on aloittaa tasosta(plane) leikkaamalla siihen etunäkymästä halutunlainen geometria, jonka jälkeen verteksit kohdistetaan syvyysuunnassa oikeille kohdilleen. Olen kokeillut monia eri tapoja pään mallinnuksessa ja suosikkini monimutkaisille päille on aloittaa mallintaminen pienestä paikalleen sijoitellusta polygonista, joita lisätään vieri viereen luoden polygoniketjuja. Tämä prosessi käydään läpi esimerkiksi Modeling Joan of Arc –tutoriaalissa (Roger). Tämä tapa mallintaa kasvot mahdollistaa mielestäni parhaimman hallinnan topologiaan ja varsinkin hyvin tehtyjen mallikuvien kanssa se on nopein tapa saada sellaiset kasvot hahmolle kuin haluaa.

Mallikuvia tehdessä on hyvä miettiä tarkkaan sitä, miten polyflow käyttäytyy. Polyflow´n hahmottamista helpottavat polet, eli kolmen, viiden tai useamman edgen kohtauspaikat. Nämä kohtauspaikat määrittelevät polyflow´n kulkua. Esimerkkikuvassa (kuva 4) olen merkinnyt punaisella muutaman pole-pisteen. Nämä ovat pari tärkeintä ja selvintä pistettä hahmoni kasvojen topologiassa. Kuvan tarkoituksena on havainnoida miten paljon polet vaikuttavat polyflow´n kulkuun: ne ovat kääntymiskohtia, jotka määrittävät topologiassa polyflow´n kulun. Polet on hyvä pitää mielessä, kun suunnittelee orgaanisen hahmon topologiaa.

Polen sijoituspaikkoja miettiessä kannattaa lähteä siitä, minne ne haluaa sijoittaa, missä niitä tarvitaan, kertoo Subdivision Modeling–keskustelupalstalla nimimerkki SomeArtist. Hän myös jatkaa, ettei ole väärää paikkaa polen sijoittamiselle, kunhan polet sijoittaa paikkoihin jossa ei tapahdu paljon deformaatiota (SomeArtist 2007.)



Kuva 4: Punaisilla pisteillä merkityt polet ja niiden vaikutus polyflow´n.

#### 4.2.3 Unreal Development Kit

Koska opinnäytetyön suunniteltu lopputulos on reaaliaikaisesti toimiva hahmomalli, tulen käyttämään siihen tarkoitukseen Unreal 3 -pelimoottoria. Unreal Development Kit eli UDK on ilmainen ladata ja käyttää, sekä siihen on paljon tutoriaaleja ja apuvideoita olemassa. Toinen mahdollinen vaihtoehto olisi ollut Source Engine, mutta UDK on uudempi, sekä siihen löytyy 3D Studio Maxille ja parille muulle 3D-ohjelmalle kätevä plugin, jolla voi mallinnusohjelmissa tehtyjä tiedostoja siirtää UDK:n sisälle helposti.

Tämä tarkoittaa, että minun täytyy pureutua myös Unreal Development Kitin editoriin ja tutkia, miten 3D Studio Maxilla toteutettu hahmo saadaan pyörimään toisessa ohjelmassa – reaaliajassa.

Otin selvää, mitkä ovat UDK:n mitat normaalikokoiselle hahmolle. Epic Games foorumilta selvisi, että keskimäärin hahmot ovat 96 yksikköä korkeita (Unit



Measurements? 2009). Mallinsin Erikan noin 110 yksikköä pitkäksi, jotta hahmo olisi hieman suurempi kuin pelaaja mutta sopivan mittainen katseltavaksi.

Opinnäytetyöni lopputuote tulee olemaan UDK:lla reaaliajassa toimiva hahmomalli, josta tulen tekemään tietokoneelle asennettavan tiedoston. En tee opinnäytetyöhön Erikasta pelattavaa hahmoa, vaan ulkopuolelta katseltavan hahmon. Jatkan ehkä myöhemmin projektin parissa ja teen Erikasta ohjastettavan versionkin.

Paneudun syvemmin UDK-editoriin ja siihen mitä ongelmia kohtasin luvussa 4.7 Reaaliaikainen hahmomalli.

### 4.3 Mallintaminen

Mallintaminen on varmaan aikaa vievin osa-alue opinnäytetyöstäni. Tulen käymään prosessia läpi jakaen alueen kahteen osuuteen, low-polyyn ja high-polyyn.

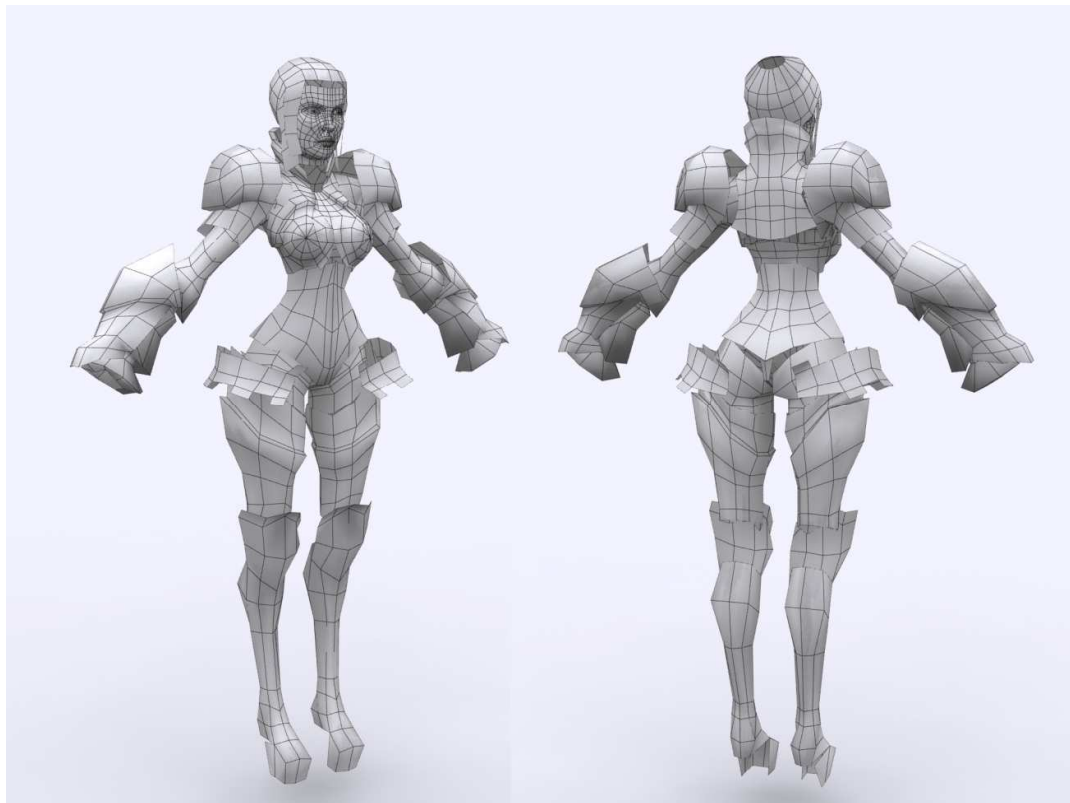
Mallinnan hahmon puhtaalta pöydältä, eli en käytä aiemmista 3D-malleistani osia. Peliteollisuudessa on ideaalista saada kokoon hahmoja nopeasti, joten siellä käytetään mahdollisimman paljon hyödyksi vanhojen mallien osia rakennettaessa uutta, mainitsee Lanning ym. (2007, 25). Opinnäytetyössäni kyse ei kuitenkaan ole mahdollisimman nopeasta työskentelystä vaan sen teosta mahdollisimman hyvin. Tekemällä opinnäytetyöni hyvin, tulen saamaan hahmostani paljon hyviä osia käyttöön tulevia projekteja varten.

Kun aloitan hahmon mallinnuksen, rakennan hahmosta ensin karkean luonnosmaisen mallin pysytellen suunnitelmien mukaisessa topologiassa pääpiirteittäin. Vältän liian yksityiskohtaista geometriaa aluksi, koska tärkeämpää on saada ensin perusmuoto haltuun, jonka jälkeen tarvittavia yksityiskohtia voi alkaa lisäilemään. Tämä helpottaa suunnittelussa polygonirajassa pysymistä. Mallinsin hahmon osiin ja jätin saumat paikkoihin, joita henkilökohtaisesti pidän vaikeina paikkoina yhdistää ja paikkoihin, jotka ovat aiheuttaneet aiempina mallinnuskertoina ongelmia. Osissa hahmon muotoa on mielestäni helpompi hallita. Kun olen saattanut hahmon muodot ja geometrian hyvälle mallille, yhdistän irralliset osat sulavaksi kokonaisuudeksi. Paras esimerkki tästä on hahmon torson ja jalkojen yhdistymiskohta. Alue on monimutkainen ja koen, että kahtena eri palasena se on helpompi hallita ja suunnitella toimivaksi.

### 4.3.1 Low-poly

Low-poly on harvaverkostoinen versio hahmosta, ja tulee olemaan se versio, joka pyörii reaaliaikaisena. Low-poly toimii myös pohjana high-poly-versiolle, eli suurin mallinnusprosessi tapahtuu juurikin low-polyn kohdalla tässä projektissa.

Tavoitteenani on saada hahmo mahtumaan noin 10 000-15 000 polygonin sisään. Tämä on ensimmäinen kerta, kun mallinnan hahmon tässä asteikossa, joten aloitin urakan tarkastelemalla saman kokoluokan valmiita hahmoja saadakseni silmämääräisen kuvan polygonitiheydestä. 10 000 polygonia on mielestäni suuri määrä ja haasteena on jakaa se tasaisesti pitkin hahmomallia tuoden tarvittavia muotoja esille.



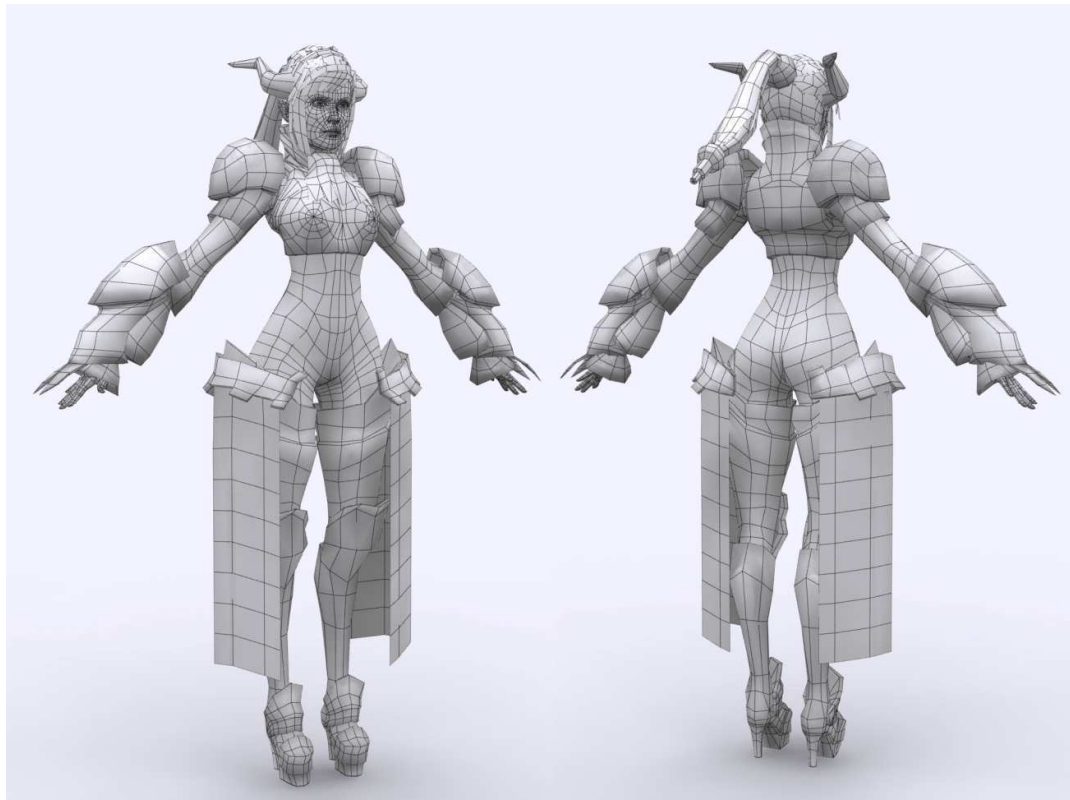
Kuva 5: Karkeita muotoja, noin 2 000 polygonia.

Aloitin mallinnusprosessin torsosta, josta etenin varovaisesti eteenpäin rakentaen ensin karkeita muotoja ja pikkuhiljaa lisäen tarvittavia yksityiskohtia (kuva 5). Mallinnuksen alkuvaiheessa hahmo oli mahdollisimman monesta kohtaa erillisinä palasina, koska halusin pystyä helposti muotoilemaan jokaista osaa erikseen vaikuttamatta toisiin osiin.

Tällä tavoin kokosin ikään kuin palapelimäisesti hahmolle muotoja, etsien samalla sopivia mittasuhteita. Vaikka minulla on tarkat mallikuvat piirrettynä valmiiksi, näyttää hahmo kolmiulotteisena aina hieman erilaiselta kuin paperilla suunniteltu versio.

Alustavia karkeita muotoja mallintaessa huomasin haluavani muuttaa hahmon mittasuhteita sopuisemmiksi ja visuaalisesti miellyttävämmiksi. Hahmon entinen sarjakuvatyylinen ohut vyötärö väistyi realistisemman tieltä (kuva 6). Päätin muuttaa vartalon mittasuhteita, koska hahmon kasvot olivat realistisimmat mitä olen tähän asti mallintanut ja hahmo näytti paljon paremmalta sopuisemmalla vartalolla.

Mallintaessa otin huomioon ensimmäisenä taitekohdat, joihin lisäsin tarvittavan määrän geometriaa, jotta hahmo animoituisi hyvin myöhemmin. Pidin myös tärkeänä kauniisti sulavaa topologiaa, jossa wireframe viivat virtaavat kauniisti ja loogisesti. Pidin pääosin hahmomallin neljän kulman polygoneina, mutta muutamaan paikkaan oli pakko jättää kolmikulmisiäkin. Pidin kuitenkin huolen, ettei kolmikulmisiä polygoneja jää taivekohtiin tai epätasaiselle pinnalle.

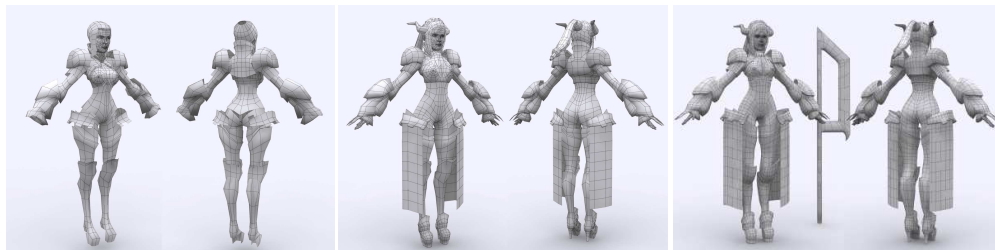


Kuva 6: noin 6 000 polygonia

Kasvoista halusin realistiset ja hyvin suunnitellut mallikuvat auttoivat sekä nopeuttivat mallinnusprosessia huomattavasti (kuva 2). Ainoaksi ongelmaksi ilmaantuivat sivukuvan viivat, jotka olin suunnitellut kulkeväksi liian väljästi. Mallinnettu pää ei näyttänyt halutulta, mutta hienosäädöllä ja välien tiivistämisellä kasvoista tuli juuri sen näköiset kuin olin halunnutkin. Mallinsin pään eri lihasryhmittymät ensin irralleen toisistaan, kuten tein vartalonkin kohdalla. Tämä auttoi eri alueiden muotoon asettelussa ennen kuin lopuksi yhdistin ne yhdeksi kokonaisuudeksi.

Hahmon käsiä ja korvia lukuun ottamatta mallinsin hahmon täysin omien mallikuvieni pohjalta. Käsiin käytin Athey Moravetsin Hand Modeling and Unwrapping – tutoriaalia (Moravets 2010), joka osoittautui erittäin käteväksi. Korvan n allinnuksessa apuna käytin Michel Rogerin Joan of Arc -tutoriaalia (Roger).

Kun hahmon mittasuhteet ja pohja oli mallinnettu, aloin lisäämään geometriaa paikkoihin jotka sitä tarvitsivat. Tarkastelin tässä vaiheessa hahmon siluettia pyöristäen kulmia lisägeometrialla. Esimerkkinä käsihaarniskat ja näiden pyöreät muodot, joihin mallinsin lisägeometriaa saadakseni pyöreämmän siluetin (vertaa Kuvia 5, 6 ja 8). Saman toimenpiteen tein jokaiselle hahmon osalle.



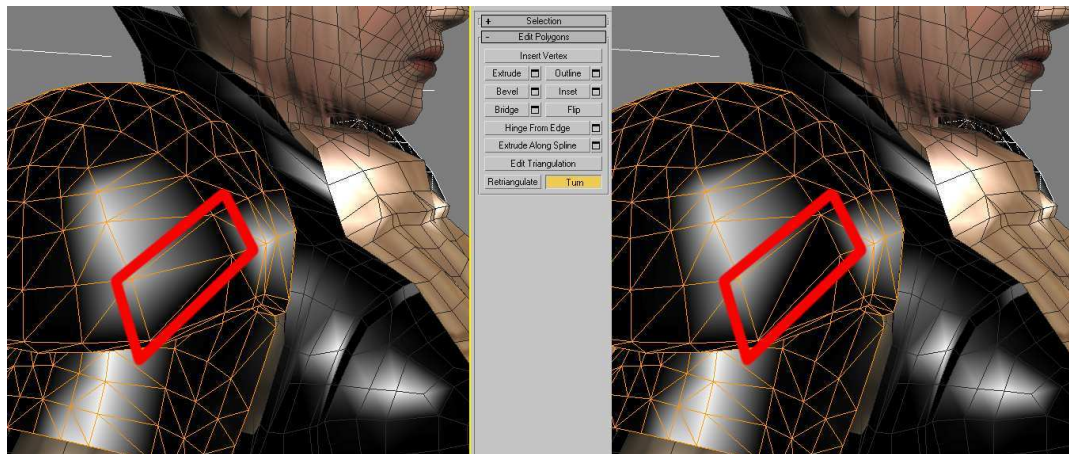
Kuvat 5, 6 ja 8

Mallinsin hahmoa pikkuhiljaa yksityiskohtaisemmaksi pitäen silmällä 10 000-15 000 polygonin rajaa. Koska tämä oli ensimmäinen mallintamani hahmo, jolle asetin polygonirajan tähän mittakaavaan, olin erityisen varovainen siitä, ettei turhia yllätyksiä polygonimäärään syntynyt. Lisäsin geometriaa aina sinne, missä sitä tarvitsin, jonka jälkeen katsoin kokonaispolygonimäärää ja lisäsin sitä muuallekin. Sama toistui, kunnes olin tyytyväinen lopputulokseen.

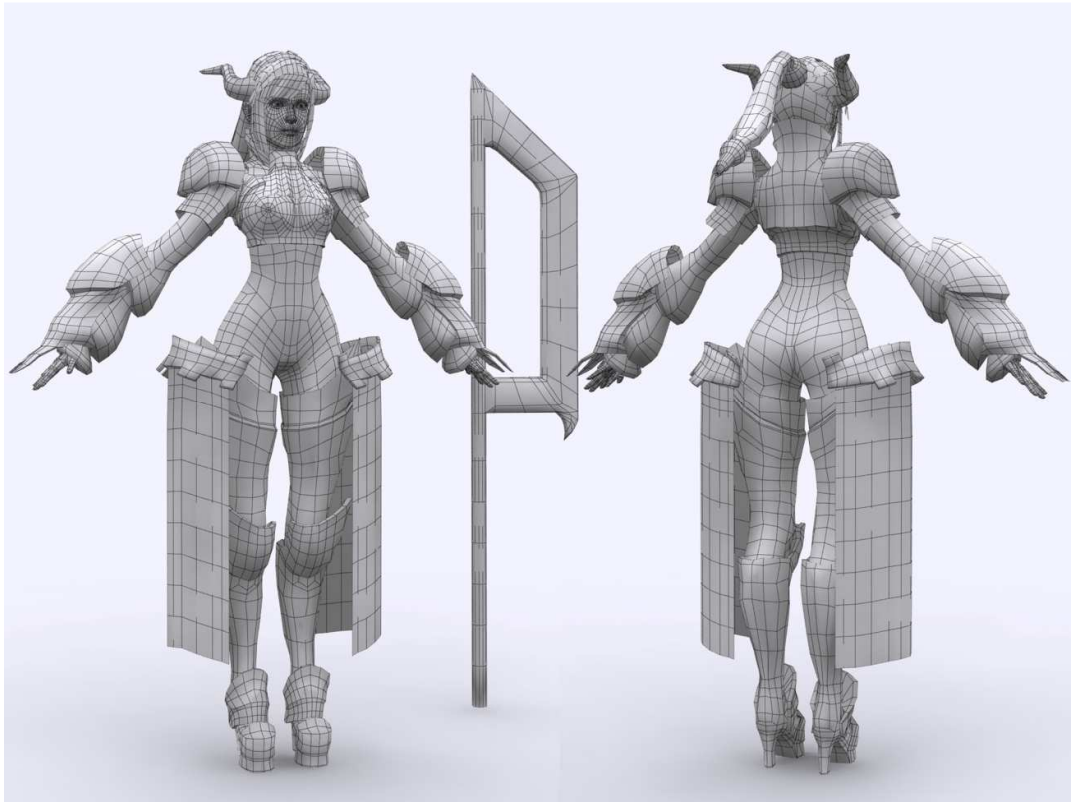
Vaikka alkuperäinen suunnitelma polygonimäärälle oli päälle 10 000 polygonia, lopulta saavuin 9 500 polygonin pintaan. En löytänyt enää paikkoja, joissa lisägeometrialle olisi ollut tarvetta, siksi hahn on polygonimäärä hieman alitti tavoitemäärää. Hahmollani on

aika yksinkertaiset hiukset verrattuna siihen, mitä olen tottunut hahmoille yleensä mallintamaan. Jos olisin halunnut tuoda lisää yksityiskohtaisuutta hiuksiin, olisin saavuttanut helposti 10 000-15 000 polygonin keskimääräisen rajan. Mielestäni on kuitenkin parempi saavuttaa haluttu lopputulos vähemmällä polygonimäärällä, kuin pakottaa hahmo tietyn rajan sisälle lisäämällä turhaa geometriaa.

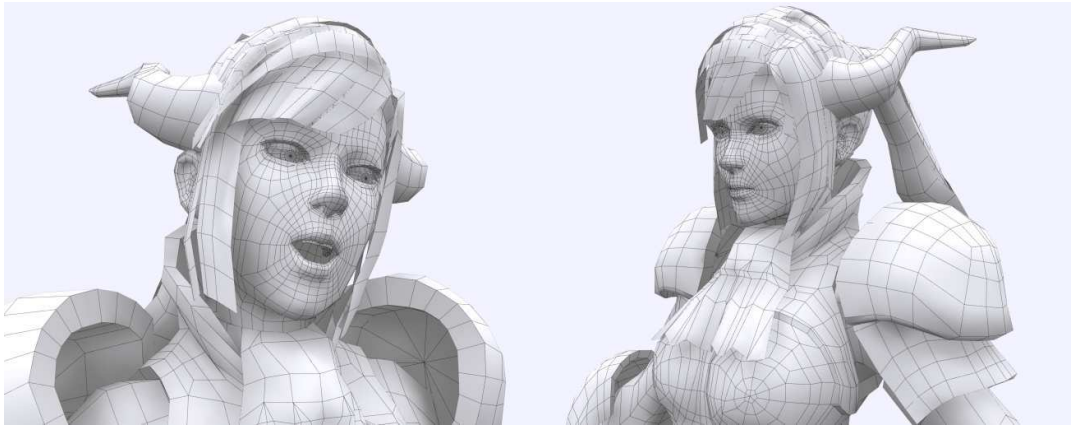
Kun olin saanut mallinnettua hahmoa tarpeeksi pitkälle, tässä tapauksessa melkein valmiiksi, keskityin valitsemaan hahmolle smoothing groupeja ja tarkistamaan mihin suuntaan nelikulmaisten polygonien sisäiset trigonit osoittavat. Smoothing groupsien määrittäminen tarkoittaa jokaiselle polygonille valittavaa arvoa, joka määrittelee miten terävän kulman ne luovat. Esimerkkinä kuva 9, jossa olkapäiden haarniskan reunan polygonit on määritelty teräviksi kulmiksi ja kaareva pinta pehmeiksi kulmiksi. Trigonien suunnan tarkistaminen ja muokkaaminen vaikuttaa siihen, miten geometria ja valo käyttäytyvät polygonin pinnalla (kuva 7). Mitä low-polympi hahmo on, sitä tärkeämpää on tarkastaa kolmioiden suunta. Hahmoni kohdalla käännsin suurimmaksi osin kasvojen alueen kolmioita, varsinkin nenän alueelta, sillä siellä oli eniten ongelmia monimuotoisen geometrian takia.



Kuva 7: Polygonien sisäisten kolmioiden kääntäminen vaikuttaa siihen, kuinka valo pinnalla käyttäytyy.



Kuva 8: lopullinen, noin 9 500 polygonia sisältävä malli



Kuva 9: lähikuvaa kasvoista ja pään alueesta

Hiukset ja silmäripset mallinsin käyttäen yksiulotteisia tasoja(plane). Ne eivät ehkä näytä kovin hienoilta sellaisenaan(kuva 9), mutta kun materiaaliin lisätään alpha-, eli läpinäkyvyyskartta(kuva 19) ja muut tekstuurit, saavat hiukset oikean muotonsa (Kuva 10).



Riippuen hahmon hiustyylistä alpha-kartan käyttö yksinkertaisten tasojen kanssa on säästäväisempi tapa mallintaa hiukset kuin se, että mallintaisi joka hiussortuvan erikseen. Hahmollani on suhteellisen yksinkertaiset hiukset eikä plane-tasoja ole kovin montaa päällekkäin. Mitä realistisemmat hiukset haluaa saada tällä menetelmällä aikaan, sitä enemmän tasoja on päällekkäin ja sekaisin. Erikalle mallinsin plane-tasoja pitkin päätä, joille tulen määrittelemään alpha-kartalla osittaisen läpinäkyvyyden. Näiden tasojen tarkoituksena on tuoda moniulotteisuutta hiuksiin kerrosten muodossa (kuva 10). Erikan kallo on kiinteä pinta hiuksia, joiden päälle mallinnan toisen samankaltaisen kerroksen geometriaa. Tähän kerrokseen tulen teksturoimaan osittain läpinäkyviä hiussortuvia. Saman tein etuhiuksille, jossa pääläella kulkee ohuita hiussortuvia jakauksen kohdalta. Jos näitä kerroksia lisäisi hahmon hiuksiin enemmän, saisi helposti aikaan tuuhean kampauksen.

Silmäripsset tein kolmesta tasosta polygoneja: yksi taso alaluomen ripsille ja kaksi yläluomelle tuomaan tuuheutta. Kaksi riviä eri suuntiin taivutettuja tasoja tekee ripsistä katseltavat joka suunnasta, eikä tule vastaan kulmaa jossa ripsiä ei näkyisi, niin kuin voi käydä jos käyttää vain yhtä taivuttamatonta tasoa.



Kuva 10: Hiukset muodostuvat päällekkäisistä tasosta joissa on osittaista läpinäkyvyyttä

### 4.3.1 High-Poly

Tässä luvussa käyn läpi yksityiskohtaisemman mallin veistämistä Mudbox-ohjelmalla. Lopputuloksena on tarkoituksena saada malli käytettäväksi pohjana normaalikartalle.

Työskentelytapoja on erilaisia: jotkut mallintavat ensin high-poly-version, jota käytetään low-poly-mallin pohjana. Jotkut mallintavat ensin suuntaa-antavan mallin low-polyyna, jonka jälkeen käyttävät sitä pohjana high-poly-versiolle, ja lopuksi mallintavat low-poly-version uudestaan veistettyä high-poly-mallia mukaillen, kuten nimimerkki Slipgatecentral videossaan Making of NOX-2292 näyttää (Slipgatecentral 2009). Minä aloitin mallintamalla ensin low-poly-version valmiiksi ja jatkamalla siitä high-poly-versioon.

En aiemmin ollut käyttänyt Mudbox-ohjelmaa, mutta ohjelmaan sisälle pääsy tuntui kuitenkin helpolta sekä luontevalta ja jo parin tutoriaalivideon jälkeen osasin käyttää ohjelmaa tarpeeksi hyvin.

Toin low-poly mallin Mudboxiin palasina. Mudbox osasi erotella jokaisen irrallisen objektin ja niitä oli helppo hallita. Tästä oli apua ongelmaan johon törmäsin: monta miljoonaa polygonia ruudulla samanaikaisesti aiheutti tietokoneen muistin loppumisen.

Mudboxilla työskentely tuntuu veistämisen ja piirtämisen yhdistelmältä. Työskentelyn alussa hahmon geometriaa moninkertaistetaan, jotta yksityiskohtia voidaan lisätä. Tämä tarkoittaa, että pohjamallissa olevat geometriset epätasaisuudet, kuten kolmikulmaiset tai pahimmassa tapauksessa yli neljäkulmaiset polygonit, tuottavat ongelmia. Myös kapeista ja pitkistä polygoneista tulee veistovaiheessa ongelmallisia. Törmäsin tähän ongelmaan tehdessäni high-poly-versiota hahmon aseesta, jossa oli paljon pitkittäisiä polygoneja. Pitkittäisiä polygoneja pitkin yksityiskohtien tekeminen sujui, mutta poikittaisista vedoista tuli sotkuisia. Ratkaisin ongelman Lisäämällä poikittaista geometriaa aseeseen. Vältin kuitenkin suurimmat ongelmat, kun pyrin alun perin rakentamaan hahmoni tasaisesti pelkistä nelikulmaisista polygoneista, kolmikulmaisia jätin paikkoihin, joissa niistä en uskonut koituvan ongelmia, kuten piilotettuihin nurkkiin tai tasaisille pinoille.

Toinen mahdollinen ongelma tehokkuuden kannalta on epätasaisesti jaettu geometria. Jos hahmo koostuu yhdestä osasta ja geometriaa on mallinnettu harvemmaksi jossain kohti samaa osaa, voi tulla hankaluuksia työstää yksityiskohtia tälle harvageometriselle



pinnalle. Mitä pienempiä yksityiskohtia haluaa työstää, sitä enemmän geometriaa täytyy pinnalla olla. Harvageometriselle pinnalle täytyy geometriaa moninkertaistaa useammin yksityiskohtia varten, ja samaan aikaan polygonimäärät moninkertaistuvat tiheämpigeometrisillä seuduilla aiheuttaen mahdollisesti tietokoneen muistin loppumisen. Hahmon osien täytyy siis koostua suhteellisen samankaltaisista geometrian tiheyksistä.

Päädyin tekemään useamman high-poly-mallin, koska halusin muokata low-poly-mallia kesken kaiken. Tein muun muassa käsistä paksummat ja muutin kehon suhteita niin paljon, että oli vaikeaa muuttaa jo työstettyä high-poly-mallia samaan muottiin. Tämän takia kannattaa olla varma low-poly-mallin lopullisuudesta ja mittasuhteita ennen kuin siirtyy high-poly vaiheeseen. Jos kuitenkin työskentelytapa eroaa ja tarkoituksena on alusta alkaen ollut rakentaa low-poly-malli uudestaan high-poly version työstämisen jälkeen, tätä ongelmaa ei ole.

Lopulta sain viimeisteltyä veistetyin mallin kahdessa eri tiedostossa työskentelyn helpottamiseksi ja nopeuttamiseksi ruudulla olevien miljoonien polygonien takia (kuva 11).



Kuva 11: Osa veistettyä High-poly kokonaisuutta

#### 4.4 Yksityiskohtaisemman geometrian illuusio normaalikartalla

Tässä luvussa selitän, mitä normaalikartat ovat ja kuinka käytin niitä opinnäytetyössäni. Käyn myös läpi ongelmia joita kohtasin tämän prosessin aikana.

Normaalikartta rakentaa illuusion yksityiskohtaisemmasta geometriasta (Kuva 12). Low-poly-mallin rinnalle rakennetaan High-poly-malli, jonka yksityiskohtien informaatio lasketaan normaalikartalle kuvaksi ja asetetaan Low-poly mallin päälle. Kun valaisu osuu malliin, muodostuu harhakuva yksityiskohtaisesta mallista. Normaalikartta toimii kuten bump -kartta, mutta käyttää kahden värin sijaan kolmea väriä (Omernick 2004, 195.).



Kuva 12: Normaalikartan vaikutus käytännössä: ensimmäinen kuva ilman normaalikarttaa ja toinen kuva normaalikartan kanssa.

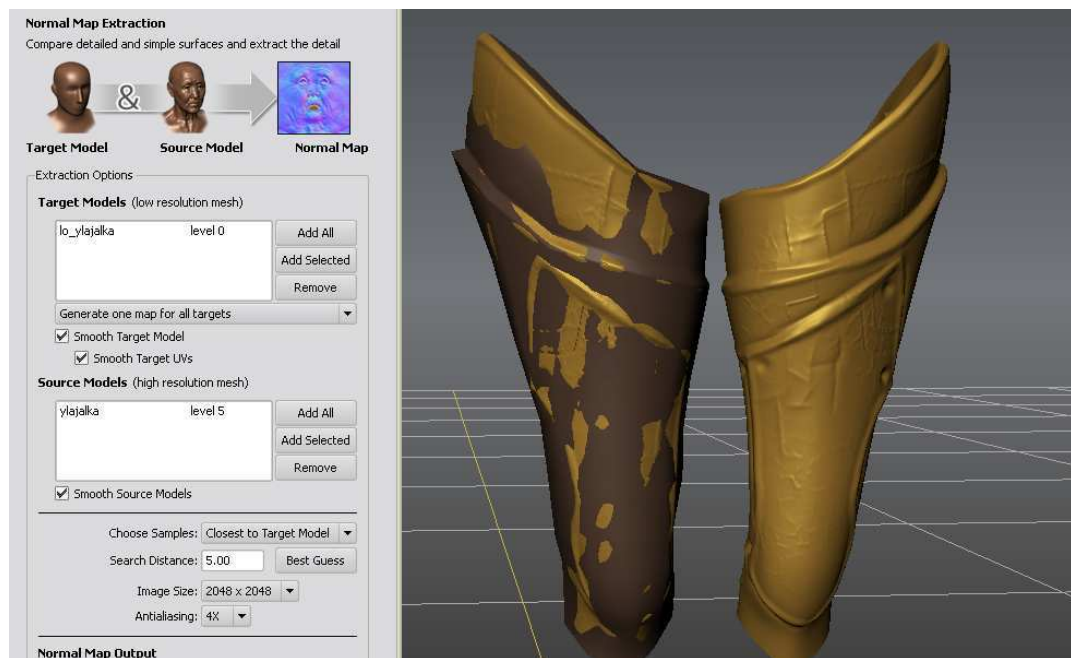
Mudboxissa on ominaisuus, jolla voi tuottaa normaalikarttoja ulos ohjelmasta. Toinen vaihtoehto olisi viedä valmiiksi veistetty malli 3D Studio Maxiin ja luoda normaalikartat

3D Studio Maxin sisällä. Valitsin Mudboxin sisäisen, sillä Mudbox on hyvin älykäs ohjelma tässä suhteessa.

Prosessi ei kuitenkaan ole täysin kivuton tai helppo, ja oikeanlaisen siistin normaalikartan ulostuonti vaatii asetusten säätämistä oikeanlaisiksi.

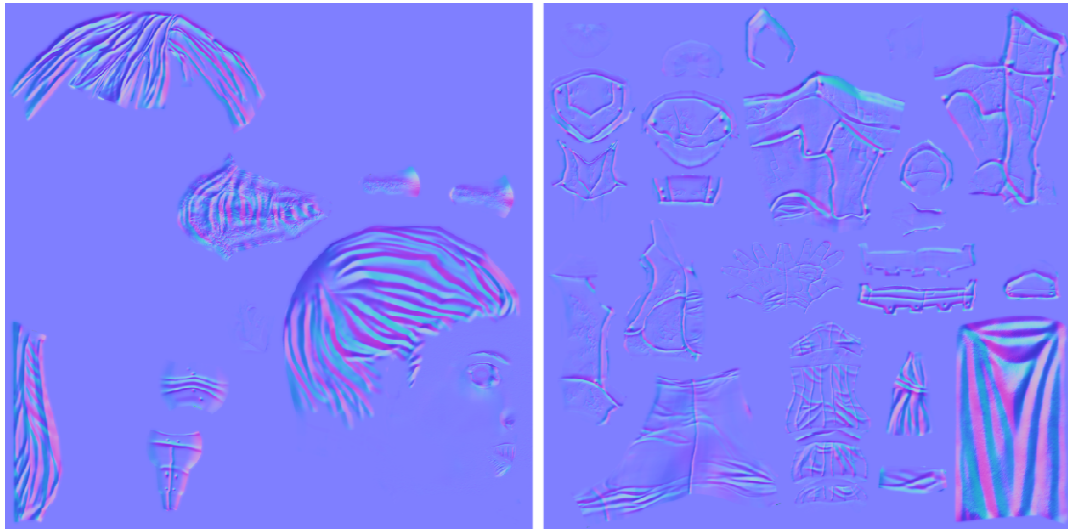
Ongelmakseni koituivat tekemäni UV-kartat. Koska kyseessä on pelihahmo, on silloin säästävää peilata tekstuurit asettaen puolikkaat päällekkäin. UV-kartoissani on näkyvillä vain puolet hahmosta, koska toinen puoli UV-kartoista on peilattu ja aseteltu toisen päälle. Mudboxissa halusin työskennellä nähden kummatkin puolet hahmosta samaan aikaan, mutta normaalikartan luonti vaiheessa kuvaan ilmestyi omituisia kuvioita. Ymmärsin tämän johtuvan siitä, ettei Mudbox ymmärrä kumman puolen hahmosta muuntaa normaalikartaksi yhden puoliskon UV-kartalle, ja tuloksena oli keskellä normaalikarttaa olevia väärinpäin kääntyneitä osia.

Koska normaalikartan luomiseksi Mudbox tarvitsee high-poly- sekä low-poly-mallit, ratkaisin ongelman tuomalla lopulta ohjelmaan vain puolikkaan hahmon low-poly-mallin. Työskentelin siis Mudboxissa kokonaisen High-poly-mallin kanssa, mutta toin ohjelmaan puolikkaan low-poly-mallin normaalikarttojen tekoa varten (kuva 13).



Kuva 13: Normaalikartan tekeminen Mudboxilla. Tummanruskea osa on puolikas Low-poly-malli ja kultainen kokonainen High-poly-malli.

Käyttämällä eri pohjamalleja sain laskettua normaalikartat vain toiselta puolen high-poly-mallia. Koska veistin high-poly-version monesta eri palasta, tein lopulliset kartat luoden jokaiselle objektille normaalikartan erikseen ja yhdistämällä ne lopulta Photoshopilla yhtenäiseksi (kuva 14). Siistin normaalikarttoja maalaamalla pois vääristyneitä reunoja joita jäi muutama.



Kuva 14: Hahmon normaalikartat

## 4.5 UV-kartat ja tekstuurit

Tässä luvussa käyn läpi UV-karttojen tekoa sekä teksturointia ja sitä, kuinka nämä vaiheet vaikuttavat lopputulokseen. UV-kartat tein hahmolle jo ennen high-poly-version luontia, tekstuurit vasta normaalikarttojen luonnin jälkeen.

### 4.5.1 UV-kartat

Teen hahmolle pääasiassa kaksi UV-karttaa: päälle ja vartalolle erikseen. Myös hahmon käyttämälle aseelle tein UV-kartan.

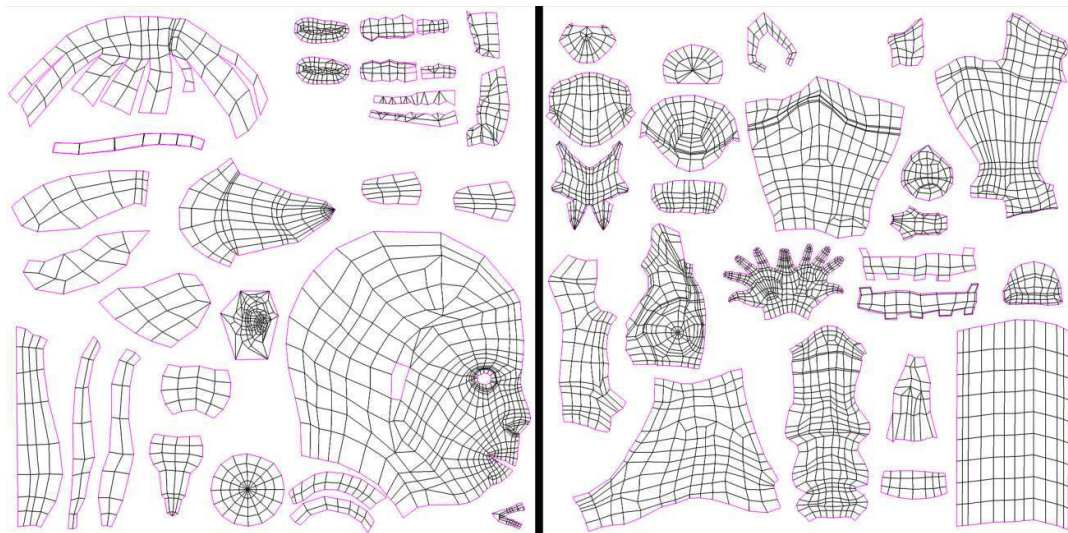
Kasvot ovat alue, johon katsoja kiinnittää huomionsa, siksi minusta on tärkeää, että kasvoille voi varata tarpeeksi suuren alueen UV-kartalta. Lisäksi vain hahmon hiuksissa ja silmäripsissä on alpha-kartan vaativia osuuksia ja olisi tuhlausta yhdellä isolla kartalla olla vain pieni alue, jossa läpinäkyvyyttä esiintyy. Pidin siksi tarpeellisena jakaa hahmo



kahteen erilliseen UV-karttaan, jossa toinen osa on päälle ja toinen vartalolle. Kolmas UV-kartta on hahmon asetta varten.

Tavoitteena oli jakaa UV:t kartalle mahdollisimman tasaisesti ja tehokkaasti. Tasaisesti levittäen niin, että laitettaessa hahmolle shakkikuvion tekstuuriksi hahmo näyttäisi siltä, että kaikki shakkiruudut olisivat suurin piirtein samankokoisia. Poikkeuksena enemmän yksityiskohtaa vaativat alueet, kuten kasvot. UV-kartat asetettiin paikoilleen yrittäen olla säästeliäs ja samalla jakaen erityyppiset osat omille alueilleen, esimerkiksi pansarit yläkulmaan ja vaatteet alakulmaan. Tämä nopeuttaa teksturointiprosessia, kun kaikki samankaltaiset ja -väriset objektit ovat lähellä toisiaan.

Peilaan kaikki mahdolliset peilattavat tekstuurit mitä hahmolle tulee. Tämä tarkoittaa sitä, että asetan UV-kartat päällekkäin peilattavista kohdista (kuva 15). Haluan hahmon mahtuvan kahteen 1024 x 1024 pikselin suuruiseen karttaan, joten peilaaminen oli erinomainen ratkaisu. Hahmo on myös suurimmaksi osin symmetrinen lukuun ottamatta osaa hiuksista. Unreal 3 -pelimoottori sallii 2048 x 2048 suuruisia karttojakin, mutta testasin hahmoani molemmilla resoluutioilla enkä huomannut suurta eroa laadussa, joten valitsin pienemmän koon. Normaalikartat jätin kokoon 2048 x 2048, koska niissä huomasin hieman yksityiskohtien katoamista.



Kuva 15: UV-kartat

Ensisijaisena ongelmana UV-karttojen kanssa oli osaamattomuus ja tietämättömyys siitä, kuinka monesta eri osasta rakennettu malli saadaan UV-kartoitettua helposti

yhdeksi isoksi kartaksi niin, kuin useissa pelihahmojen tekstuurien mallikuvissa on näkynyt. Vaihtoehtoina oli hahmon yhdistäminen yhdeksi objektiksi, mutta tämä hankaloittaisi monimutkaisessa mallissa UV-karttojen tekoa. Aiemmin onnistuin vain rikkomaan UV-kartat hyvin helposti, joten olen ollut niiden kanssa hyvin varovainen. Ratkaisu löytyi kuitenkin sattumalta mallintaessa vapaa-ajan projektia. Opin, että käyttämällä 'Collapse to' komentoa, saa romautettua UV-kartoitus informaation polygonimalliin rikkomatta sitä. Kartoitin joka osan erikseen ja romautettua pinot yhdistin osat yhtenäiseksi kokonaisuudeksi. UV-kartoitus informaatio kantautuu tällöin monista pienistä objekteista yhteen isoon, jolloin voi asetella valmiiksi eritellyt ja levitettyt kartat yhdelle isolle kartalla (kuva 15).

UV-kartoitusprosessi kulki niin, että valitsin saumakohtat 3D Studio Maxin UV-työkaluilla, jonka jälkeen vein objektin RoadKill-ohjelmaan. Mielestäni 3D Studio Maxilla saumojen valinta on helpompaa ja nopeampaa, kuin sen tekeminen Roadkill-ohjelmassa. Roadkill osaa levittää UV-kartat nätisti ja tasaisesti, paremmin kuin 3D Studio Max, mutta tarkempaa huomiota vaativiin kohteisiin, kuten päähän, käytin paljon myös 3D Studio Maxin Pelt ja Relax -toimintoja saadakseni haluamani lopputuloksen. Kaikkein yksinkertaisimmat objektit eivät tarvitse Roadkill-ohjelman toimintoja lainkaan.

UV-karttojen ja hahmon kanssa tuli tehtyä paljon edestakaista säätöä UV-kartoituksen aikana. Olin epävarma miten Mudboxissa normaalikarttojen teko onnistuu ja siksi pelasin varman päälle tehden asioita vaikeimman kautta. UV-kartoitettuani jokaisen hahmon osan erikseen, kokosin ne lopulta yhteen asetellen palaset oikeille kohdin karttaa. Tämän jälkeen purin hahmon takaisin eri palasiin. UV-kartat muistavat missä kohdin karttaa niiden kuulu olla, vaikka ne olisivatkin purettu erilleen. Näin varmistin, etteivät palaset vahingossakaan osu päällekkäin, kun lopulta kokoon tehdyt normaalikartat yhteen. Varman päälle pelaaminen kannatti, sillä palasina Mudboxiin viety hahmo oli myös helpompi ja kevyempi veistää. Hyvin tehokkaan koneen kanssa pitäisin hahmon yhtenäisenä levittäen eri osat erilleen helpompaa veistämistä varten.

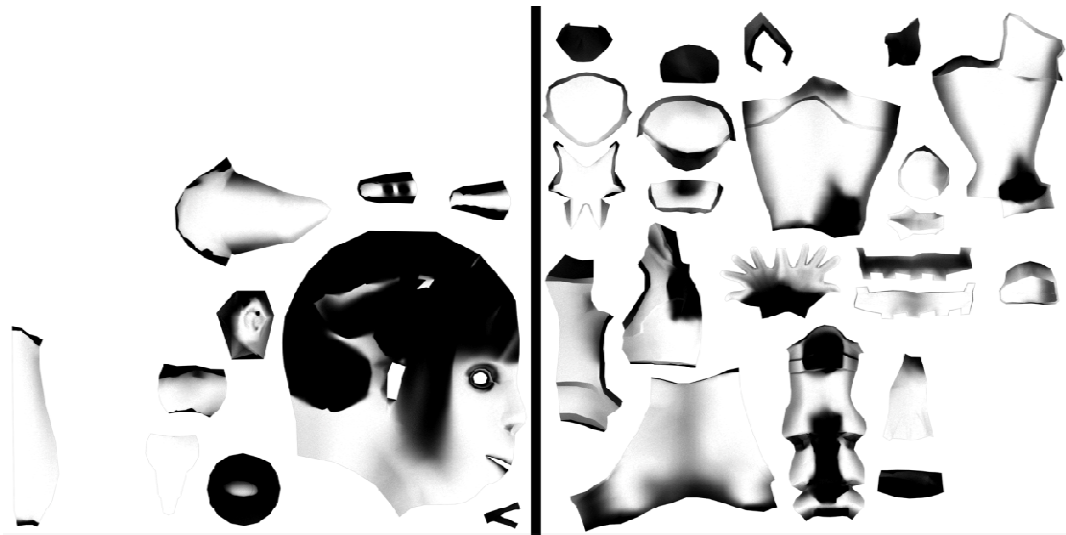
#### 4.5.2 Tekstuurit

Tehdessä tekstuureja peliin, on optimoinnin kannalta hyvä ottaa huomioon tekstuurikarttojen koko. Nämä koot ovat aina kaksi kertaa toisiaan suurempia eli

esimerkiksi 32 x 32, 64 x 64, 128 x 128, 256 x 256, 512 x 512, 1024 x 1024, 2048 x 2048 ja niin edelleen. Tutoriaali KatsBits:n sivustolla kertoo, että tämä on erittäin keskeinen ja välttämätön sääntö, koska pelimoottorit toimivat tietyllä tavalla (KatsBits).

Maalaan hahmon tekstuurit käsin Adobe Photoshop -ohjelmalla ja käytän myös hyväkseni valmiita normaalikarttoja (kuva 14) ja 3D Studio Maxilla luotuja Ambient Occlusion tekstuurikarttoja (kuva 16). Tein kaikki kartat kokoon 2048 x 2048 pikseliä, mutta pienensin osan 1024 x 1024 suuruisiksi, koska yksityiskohtia ei näyttänyt katoavan. Pienensin specular-kartat myöhemmin kokoon 512 x 512, koska ne eivät kärsineet pienennyksestä.

Normaali karttojen luonti ennen tekstuurien tekoa helpottaa teksturointiprosessia. Normaalikartan voi laittaa tekstuurin pohjaksi Photoshopissa muuttamalla sen ensin harmaaksi, vähentämällä kirkkautta ja laittamalla tason arvoksi "Soft Light" (Swanston 2010). Tällöin valmiit yksityiskohdat tulevat esiin oikeille paikoilleen ja niiden päälle on helppo maalata. Ambient Occlusion -kartat auttavat luomaan varjoja paikkoihin, jonne valo ei pääse. Ambient Occlusion- ja normaalikarttojen avustuksella maalasin hahmolle Diffuusi-kartan, eli värit hahmomallin päälle (kuva 17). Ambient Occlusion -karttoja käytin luomaan hieman varjostusta hahmon pinnalle, mutta ei liikaa, sillä Unreal 3 -pelimoottori hoitaa valaistuksen ja varjot.



Kuva 16: Ambient Occlusion -kartat

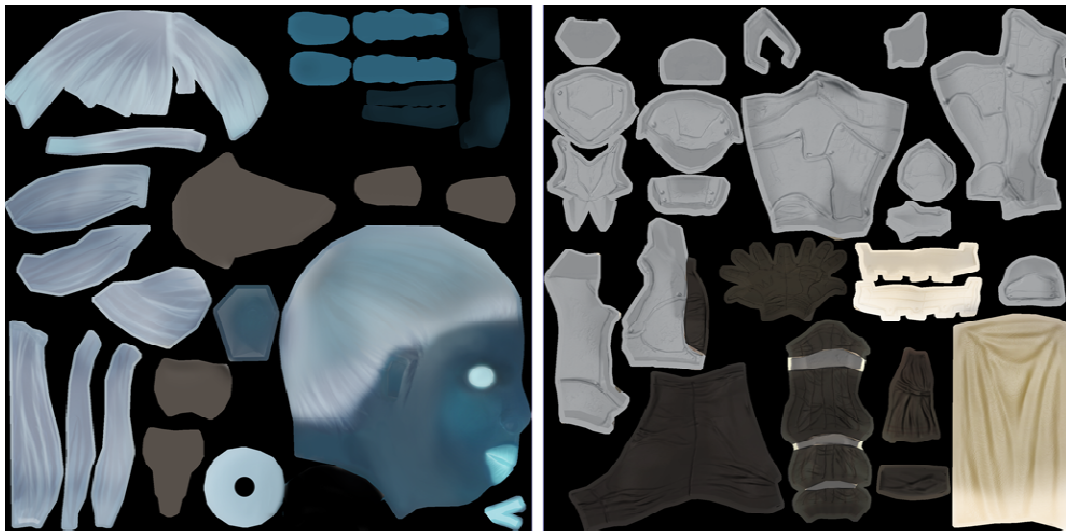
Diffuusi-kartan pohjalta tein hahmolle Specular-, eli kiiltokartan. Tämä tarkoittaa, että mitä vaaleampi sävy kartassa, sitä enemmän pinta tulee kiiltämään ja jos pinnalla on

väriä, sen värisenä pinta kiiltää(Kuva 18). Hiukset ja ihon värjäsin siniseen sävyyn, sillä Wilsonin(Wilson 2009) mukaan sininen antaa iholle puhtaamman värisen kiillon kuin harmaasävy. Specular-kartan iholle tein kääntämällä ihonvärin ensin vastaväriksi ja maalamalla päälle. Muille specular-karttojen osille löysin sopivat värit ja kirkkaudet testaamalla niitä UDK editorissa, kunnes sain haluamani lopputuloksen.

Viimeisenä tein myös hiuksille ja silmäripsille Alpha-, eli läpinäkyvyyskartan (kuva 19). Kartassa musta on läpinäkyvää ja valkoinen kiinteää pintaa. Harmaasävyt ovat taas osittaisia läpinäkyvyyksiä.



Kuva 17: Diffuusi-kartat



Kuva 18: Specular-kartat



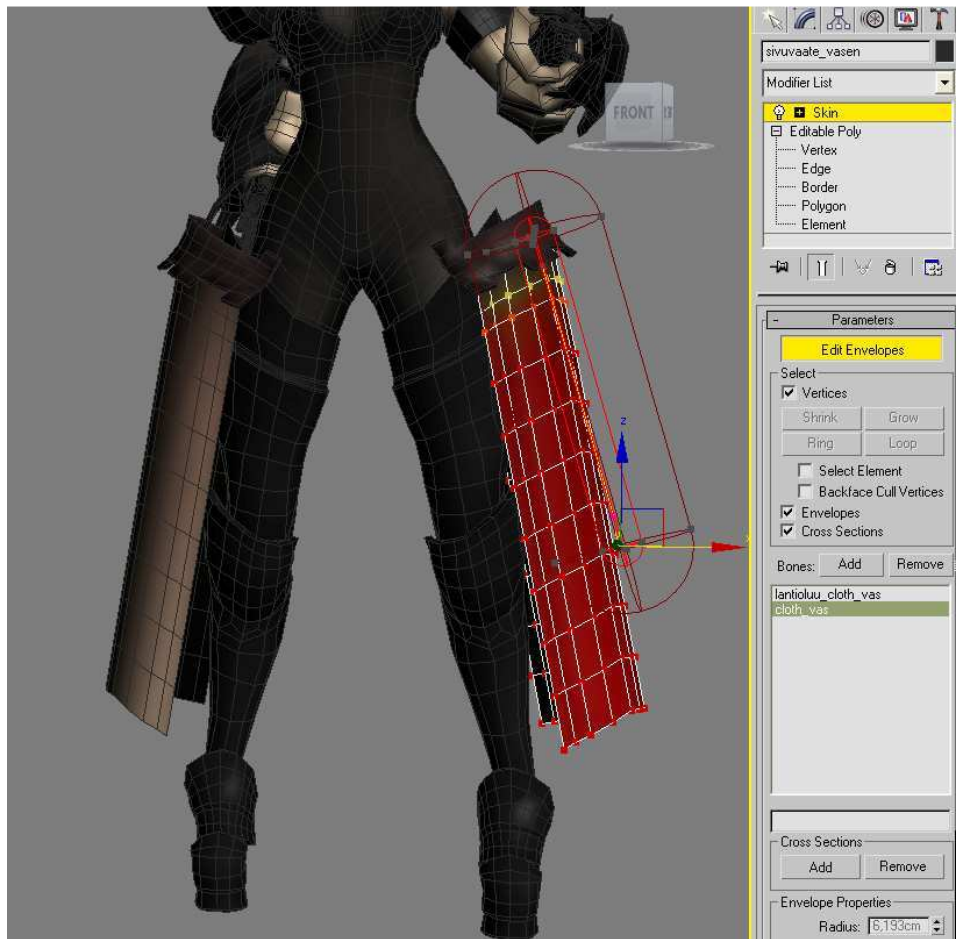


Kuva 19: Alpha-kartta

## 4.6 Riggaus

Tässä luvussa käyn lyhyesti läpi valintojani hahmon luurakenteelle. Tulen käyttämään 3D Studio Maxin sisäisiä Biped-luita sekä yksittäisiä luita näiden lisäksi tarvittaviin kohtiin. UDK:n editorissa on mahdollisuus käyttää Unreal Tournament hahmojen valmiita luita, joille on UDK editorissa paljon monenlaisia animaatioita valmiina. Valmiiden luiden rajoituksena on, että niitä ei voi skaalata joten hahmomalli täytyisi rakentaa luiden päälle niitä mukaillen, lukee nettisivulla UT3CustomCharacters (Jones, Mielke, Herzog). Hahmomallini sisältää myös paljon sellaisia osia, jotka tarvitsevat omat erikoisluit, joten päätin käyttää 3D Studio Maxin Biped- sekä yksittäisiä luita halutun lopputuloksen saavuttamiseen.

Hahmon luusto helmoja ja kasvoja lukuun ottamatta on tehty 3D Studio Maxin omalla Biped-luurangolla. Helmojen animointiin tulen käyttämään UDK:n sisäistä vaatesimulaatiota ja tätä varten tulee määrittää helmalle luu, johon lisätään verteksit jotka tulevat simuloituiksi (kuva 20). Lisäksi tarvitaan toinen luu, joka määrää vaatteen kiinnityspisteen, eli tässä tapauksessa helman yläosa. Kiinnitin tämän luun myös jalkaluuhun, jotta helma seuraa jalkojen liikettä hahmon liikkuessa.



Kuva 20: Valitut verteksit UDK:n vaatesimulaation käytettäväksi.

Kasvojen luiden säätimien tekoon käytin Athey Moravetsin Face Rigging tutoriaalia (Moravets 2007) ja sen yhteydessä mainittua Phatdady's Rig Tool Scriptaa 3D Studio Maxille, joka luo automaattisesti halutun säädinvalikon, johon voi säätää 3D Studio Maxin Reaction Managerilla halutut arvot kasvoluiden hallitsemiseksi (kuva 21). Säätimet toimivat liikuttamalla rajatun alueen sisäpuolella olevaa palloa Y ja X akseleilla. Reaction Manageriin syötetään arvot missä mikäkin luu on, kun tuota palloa liikutetaan eri akseleilla. Esimerkiksi, kun leukaluun säädin on ylhäällä, on suu kiinni ja kun säädintä vedetään alaspäin, suu aukeaa.

Tämä oli ensimmäinen kerta kun tein hahmolle luut kasvoihin sen ilmeiden säätämiseksi; aluksi olisin käyttänyt morph -kontrolleja sillä UDK tukee niitä, mutta ei tarpeeksi hyvin tai haluamallani tavalla, joten kasvoluiden teko oli paras ratkaisu. Olen oikeastaan tyytyväinen morphien toimimattomuuteen, koska sain hyvän syyn ja

tilaisuuden kokeilla kasvoluiden tekoa. Luiden asetteluun en käyttänyt tutoriaaleja, vaan asettelin välttämättömimmät paikkoihin joita uskoin eniten tarvitsevani. Luut laitoin silmiin, ylä- ja alaluomiin, kulmakarvoihin, leukaan, suupieliin, ylähuuleen ja nenään, näillä uskoin voivani saada hahmolle tarvitsemani ilmeet.

Silmiä ja luomia kontrolloi yksi ainut säädin. Silmä liikkuu oikealle ja vasemmalle liikuttamalla säädintä vaakatasossa, hahmon katsetta nostamalla ylä- ja alaluomet seuraavat silmän liikkeitä ja kun kontrollerin vie tarpeeksi alas hahmon silmät menevät kiinni. Mielestäni tämä luo helposti elävyyttä silmäanimaatioon hahmoni tarvitsemalla animaatiotasolla.



Kuva 21: kasvojen luut ja hallintapaneeli

## 4.7 Reaaliaikainen hahmomalli

Tässä luvussa saatetaan valmiiksi reaaliajassa toimiva hahmomalli. Käyn läpi miten kokosin hahmon Unreal-pelimoottorin editorissa ja minkälaisiin ongelmiin törmäsin matkan varrella. Lopullinen tuotos on reaaliaikaisessa ympäristössä tutkailtava hahmomalli. Käytin projektissa joulukuun 2009 beta versiota UDK:sta ja myöhemmin päivitin sen maaliskuun 2010 beta versioksi.

### 4.7.1 UDK ja mallinnetun hahmon siirtäminen Unreal Editoriin

Mallinnettu hahmo on helppo siirtää 3D Studio Maxista UDK-editointiohjelmaan, sillä 3D Studio Maxiin on siihen tarkoitukseen tehty ActorX niminen plugin. Vein hahmon neljässä eri osassa editoriin, hahmon sivuliepeet erillisinä kappaleinaan sekä ase omana objektinaan. Liepeet ovat siksi erikseen, koska halusin niille vaatesimulaation, josta selitän luvussa 4.7.2. Ase on erillisenä, koska se on objekti, joka todennäköisesti voi vaihtua tai sen voi poistaa kokonaan. Tekstuurit täytyy viedä editoriin erikseen ja materiaalit kootaan editorin sisällä, näistä lisää luvussa 4.7.3. Animaatioita varten käytetään ActorX–pluginia, jolla määritellään mitkä keyframeet kuuluvat mihinkin animaatioon. Animaatiosta kerron lisää luvussa 4.7.4.

Kun hahmo on tuotu UDK:n editointiohjelmaan, täytyy hahmo ikään kuin rakentaa uudestaan. Siirrettäessä hahmomalli 3D Studio Maxista UDK editoriin on hahmo staattinen ja tekstuuriton muotti polygoneja: materiaalit, animaatiot ja muut asetukset täytyy asettaa ohjelman sisällä hahmolle uudelleen. Koska toin hahmon neljänä eri palasena editoriin, on ne yhdistettävä taas pelimaailmassa ja kiinnitettävä toisiinsa.

UDK editorin sisällä on paljon valikoita ja asetuksia ja onneksi internet tarjoaa paljon tutoriaaleja siitä mitä asetuksia pitää asettaa päälle, jos haluaa jonkin asian toimivan. Tässä osassa opinnäytetyötä käytin hyvin paljon tutoriaaleja päästäkseni eteenpäin, mutta löysin myös ratkaisuja itsenäisesti.

#### 4.7.2 Vaatesimulaatio UDK:ssa

UDK tukee vaatesimulaatiota, joka saa minkä tahansa objektin elävöitymään kuin oikea vaate liikkeen ja tuulen mukaan. Erikan sivuliepeet olivat täydellinen kohde käyttää UDK:n vaatesimulaatiota. Toin molemmat liepeet erillisinä objekteina UDK-editoriin, ja osoitin niille kaksi eri luuta kummallekin. Toisen luun tehtävänä on osoittaa liepeestä ne kohdat, joista vaate roikkuu. Tämä estää vaatteen putoamista simulaatiossa maahan. Toisen luun tehtävänä on osoittaa mitkä osat liepeessä tulee toimimaan vaatesimulaatiossa(kuva 20). Simulaatiota varten syötetään vaateobjektille arvoja UDK:n sisällä ja jotta vaate törmäisi hahmon kanssa, on hahmolle osoitettava Physical Asset, jolla luodaan hahmon ympärille yksinkertainen kehikko, joka simuloi törmäyskohtia. Sain loppujenlopuksi vaatesimulaation toimimaan nätisti ja eron huomaa parhaiten liikkeessä(kuva 22). Liepeet kannattaa myös kiinnittää hahmoon ja tätä varten toin liepeet kahtena kappaleena, koska objekti täytyy kiinnittää yhteen isäntäobjektin luuhun. Kiinnitin liepeet hahmon kumpaankin reisiluuhun, jotta ne myötäilisivät jalkojen liikettä.



Kuva 22: Vaatesimulaatio pois päältä ja käytössä

#### 4.7.3 Normaalikartat ja materiaalit UDK:ssa

Unreal 3 -pelimoottorissa on vielä tämänhetkisessä (maaliskuu 2010 beta) versiossa ongelma, jossa hahmoon tekstuuriin tulee sauma peilattuihin kohtiin(Texture Seam Problem, 2009). Ratkaisuna tähän löysin sattumanvaraisesti keskusteluketjusta ratkaisun, jossa ratkaisuna maalattiin normaalikartan saumakohtiin puhdasta sinistä RGB väriarvoa 0,0,255 (Normalmap Seams in Zmapper 2009). Testasin tätä omaan normaalikarttaan ja ero oli huomattava (kuva 23). Sauma ei poistu kokonaan, mutta himmenee huomattavasti. Lopullisen ratkaisun löysin kuitenkin UDK:n import-valikon asetuksista. Jos tästä valikosta valitsee normaalikartan sisään tuonnin yhteydessä oikean pakkausmuodon, muuttaa UDK normaalikartat oikeanlaisiksi eikä minkäänlaista saumaa jää jäljelle ja normaalikarttoihin ei tarvitse maalata reunoihin käsin RGB väriarvoa 0,0,255. Jälkeenpäin tekstuurieditorista pakkausmuodon muuttaminen ei tehnyt mitään, vaan kartat täytyi muuttaa siinä vaiheessa kun ne ohjelmaan alun perin toi ja koska jätin tämän vaiheen huomiotta karttojen alkuperäisen tuonnin yhteydessä, aiheutti se minulle paljon päänvaivaa jälkikäteen.

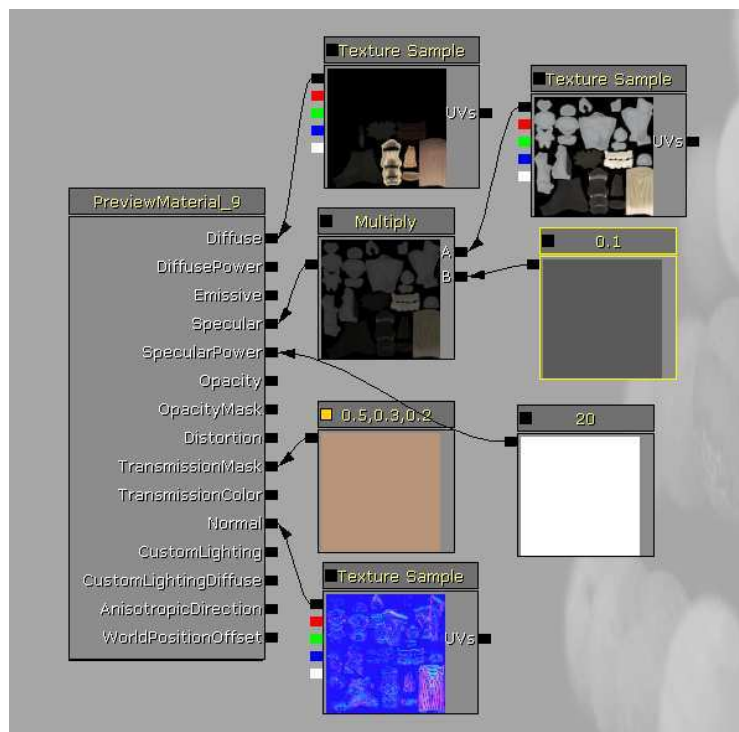


kuva 23: Unreal 3:n normaalikartan sauman ongelma ennen ja jälkeen



Koska tein hahmolle kaksi eri tekstuurikarttaa ja tulen tekemään hahmolle erilaisia materiaaleja, on muistettava asettaa 3D Studio Maxissa jokaiselle polygonille Material ID. UDK:n editori käyttää Material ID:itä määrittelemään sitä, mikä materiaali sijoitetaan minnekin osaan hahmoa. Tein hahmolle 6 eri Material ID määritelmää eri materiaaleja varten. Kaksi eri ID:tä vartalolle, haarniska ja kangasosat erikseen ja kolme eri ID:tä päälle: normaaleille kiinteille osille yhden, toisen kaikille pään osille joissa käytetään alpha-karttaa ja kolmannen materiaali silmän pupillille, jonka tarkoituksena on olla läpinäkyvä, mutta kiiltoa tuova osa. Viimeinen ID oli tarkoitettu aseelle. UDK:n Joulukuun 2009 beta versio ei vielä osannut tehdä pehmeää alphanäkymää, ja alpha piti hoitaa toisella tavalla, tavalla joka olisi saattanut kaikki kiinteätkin osat läpinäkyviksi toistensa kanssa. Siksi asetin eri osille eri materiaali ID:t ja tein niille eri materiaalit UDK:n sisällä.

Materiaalien tekeminen UDK:n sisällä toimii node-periaatteella, jossa yhdistetään UDK-editoriin tuodut tekstuurit ja normaalikartat yhteneväksi materiaaliksi (kuva 24). Kyseinen materiaali on hahmon kaikkien vaateosien materiaali. Tekstuurikartoissa näkyy myös haarniskan osia. Tämä siksi koska olin laittanut koko hahmon kaulasta alaspäin yhdelle UV-kartalle, mutta material ID määrittelyjen ansiosta voin määrittellä eri osille eri materiaalit.



Kuva 24: UDK:n materiaalieditori. Esimerkki materiaalin kokoonpanosta

#### 4.7.4 Animaatio

Unreal 3 -pelimoottori tukee morph-kohteita, joten alkuperäinen idea oli tehdä hahmolle muutama morph-kohde kasvoista. Tämä kuitenkin koitui haasteeksi, koska morphit oli vaikea saada toimimaan, ja kun ne viimein toimi, olin tehnyt tehdä liikaa uhrauksia hahmon ulkonäköön ja animaatioon liittyen. Unohdin siis morphit ja päädyin tekemään elämäni ensimmäisen facerigin eli luut kasvoille. Facerigillä tehtyjä kasvoanimaatioita käytetään peleissä yleisemmin kuin morpheilla tehtyjä, uskoisin siksi, että kovin moni pelimoottori ei ole tukenut morpheja. Oppimisen ja lopputuloksen kannalta pidin facerigin teon opettelua parempana ratkaisuna kuin morpheissa pysymistä.

Koska opinnäytetyöni rajaa lopullisen tuotoksen vain pelimoottorissa tarkasteluun, tein hahmolle vain yhden animaation, jossa hahmo siirtyy taisteluasennosta toiseen. Pyrin pysymään animaatioissa mahdollisimman lähellä hahmon persoonaa eli eleganttia mutta uhkaavaa persoonaa. Pelattavaa pelihahmoa varten täytyisi tehdä monta pientä animaatiota kävelyille, juoksulle ja vaikka käsien liikkeille ellei kyseessä ole ensimmäisestä persoonasta kuvattu peli, jossa pelihahmoa ei näy, vaan korkeintaan kädet tai esineet mitä niissä on. Jos jossain myöhemmässä vaiheessa tulen käyttämään Erikaa johonkin peliprojektiin, teen enemmän animaatioita, jotka kytkeisin näppäinkomentoihin ja tekisin Erikasta pelattavan versio.

#### 4.7.5 Lopullinen reaaliaikainen malli

Saatuani hahmon toimimaan kuten halusin reaaliaikaisessa ympäristössään, rakensin hahmon ympärille yksinkertaisen maailman ja laitoin sinne valonlähteet (kuva 25). Ympäröivän maailman rakentamiseen käytin UDK:n valmiita rakennelmia, materiaaleja ja efektejä kuten sumu. Yritin rakennelmilla luoda tunnelmaa, joka sopisi hahmon luonteeseen, mutta pidin rakennelmat kuitenkin yksinkertaisina, jotta ne eivät veisi liikaa huomiota pois hahmosta.

Tein lopullisesta kohtauksesta asennusohjelman, joka toimii opinnäytetyöni esittelykappaleena (LIITE 1). Sijoitin hahmon kohtaukseen, joka tapahtuu pyöreässä rakennuksessa ja hahmo on animoitu liikkumaan itseksensä.





Kuva 25: Reaaliajassa toimiva lopullinen hahmomalli

## 5 POHDINTAA

Opinnäytetyön tekemisen sujui tasaiseen tahtiin ja varsinkin edetessä UDK:n kanssa edistyminen tapahtui mennessä ongelmasta toiseen, mutta kaikkiin ongelmiin löytyi kuitenkin lopulta ratkaisu. Nyt minulla on parempi käsitys pelihahmon mallintamisen

kokonaiskuvasta. Vaikka moni vaihe oli minulle jo tuttu entuudestaan, onnistuin vahvistamaan osaamistani näillä alueilla ja oppimaan paljon uutta.

Pelihakmon mallintaminen on tiettyine rajoituksineen samanlainen prosessi kuin minkä tahansa animaatiohakmon mallintaminen. Pelihakmojen kohdalla on tärkeää olla tehokas ja optimoida mallit mahdollisimman pitkälle.

Alkuperäinen idea suunnittelun osalta oli, että suunnittelisin koko hakmon topologian suurimmaksi osin paperilla. Idea osoittautui kuitenkin käytännön kannalta liian aikaa vieväksi ja päädyin suunnittelemaan vain tärkeimmät ja vaikeimmat alueet hakmosta valmiiksi, loput topologiasta syntyi hakmoa tehdessä ilman suurempia ongelmia.

Mikä näyttää hyvältä paperilla, ei välttämättä näytä hyvältä kolmiulotteisena. Etukäteen suunnittelusta on kuitenkin paljon hyötyä ja se nopeuttaa mallintamista. Paperille piirretyt hakmon mittasuhteet eivät kuitenkaan näyttäneet tasapainoisilta kolmiulotteisena, siksi päätin muuttaa hakmon mittasuhteita leventämällä vyötäröä ja pienentämällä rintakehää. Visuaalisesti miellyttävät hakmon tai objektin muodot ja mittasuhteet ovat aina olleet minulle tärkeitä hakmoja tehdessä ja siksi käytin ehkä turhankin paljon aikaa muuttaen hakmoa sen jälkeen, kun hakmo oli jo mallinnettu valmiiksi. Osasin kuitenkin ennakoida tämän ja pidin hakmon monessa eri osassa, jos jonkin osan koko tai muoto vaatiikin muokkaamista.

Jotkut puutteet huomasin vasta high-poly veistovaiheessa Mudboxissa, jolloin päätin paksuntaa hakmon käsivarsia, niiden näyttäessä liian kapeilta. Muutettuani käsivarsien paksuutta, se vaikutti myös haarniskaan ja moneen muuhun osaan ja lopuksi minun täytyi veistää nämä osat Mudboxissa uudestaan. Ensi kerralla yritän varmistautua hakmon mittasuhteista jo mahdollisimman aikaisessa vaiheessa, jotta välttyisin turhalta työltä.

Opinnäytetyössäni tuli vastaan muutama osa-alue jonka olisin halunnut tehdä paremmin jos aikaa olisi ollut enemmän, olen kuitenkin erittäin tyytyväinen lopputulokseen varsinkin kun kyseessä on ensimmäinen projekti Unreal Development Kitin kanssa.

Osa-alueita joita olisin halunnut tehdä paremmin, olivat muun muassa hakmon rigi. Asetin selkäluut liian taakse ja animaatio nojatessa eteenpäin tai taaksepäin ei mene aivan kuin olisin halunnut, tässä vaiheessa olin kuitenkin jo tehnyt fagerigin ja

skinnannut koko hahmon, joten rigin uudelleen järjesteleminen olisi aiheuttanut liian suuren projektin, johon minulla ei tämän opinnäytetyön puitteissa ollut aikaa.

Facerigi ei myöskään ollut täysin sellainen kuin olisin sen halunnut olevan ja kontroleissa olisi ollut paranemisen varaa, kuten silmien räpytyksessä, silmät eivät sulkeudu aivan niin, kuin olisin halunnut. Myöskään UV-kartat eivät olleet niin optimoituja kuin ne olisivat voineet olla. UV-kartat kannattaa asetella siten, että tyhjää tilaa jää mahdollisimman vähän.

Opinnäytetyö tarjosi paljon uutta opittavaa ja vanhojen oppien vahvistamista. Aikataulullisesti sain opinnäytetyön projektiosuuden valmiiksi paljon nopeammin kuin olin suunnitellut. Olen myös erittäin tyytyväinen oppimaani ja uskon, että seuraavalla kerralla osaan tehdä pelihahmon entistä paremmin.

## LÄHTEET

- Blender Artists Forum* 20.9.2008. [verkkodokumentti].  
<<http://blenderartists.org/forum/showthread.php?p=1213654>> (10.1.2010)
- Jones, Paul ; Mielke, Chris; Herzog, Aaron. *UT3CustomCharacters*. [verkkodokumentti].  
UDN: <<http://udn.epicgames.com/Three/UT3CustomCharacters.html>> (12.4.2010)
- KatsBits. *Make better textures, optimising, 'power of two' and proper image dimensions*. [verkkodokumentti]. KatsBits: <<http://www.katsbits.com/htm/tutorials/textures-correct-power-of-two-size.htm>> (20.4.2010)
- Lanning, Kevin; Petroc, Zack; Baysal, Timur 2007. *d'artiste Character Modeling 2*. Australia: Ballistic Publishings.
- Moravets, Athey 11.12.2007. *Face Rigging*. [verkkodokumentti].  
<<http://www.bakaneko.com/howto/computer/3d/rigging/>> (10.3.2010)
- Moravets, Athey 22.2.2010. *Hand Modeling and Unwrapping Tutorial for 3dsmax*. [verkkodokumentti]. <<http://athey.comyr.com/tutdump/hand/>> (24.2.2010)
- Normalmap Seams in Zmapper* 15.12.2009. [verkkodokumentti]. Epic Games Forum.  
<<http://forums.epicgames.com/archive/index.php/t-712099.html>> (24.3.2010)
- Omernick, Matthew 2004. *Creating the Art of the Game*. Yhdysvallat: New Riders.
- Roger, Michel. *Modeling Joan of Arc*. [verkkodokumentti].  
<<http://www.3dtotal.com/ffa/tutorials/max/joanofarc/joanmenu.php>> (24.2.2010)
- Slipgatecentral 23.7.2009. *Making of NOX-2292 (part 1) & (part 2)*. [verkkodokumentti].  
Vimeo: <<http://vimeo.com/5738519>> (22.1.2010)
- SomeArtist 20.2.2007. *The Pole*. [verkkodokumentti]. Subdivision Modeling:  
<<http://www.subdivisionmodeling.com/forums/showthread.php?t=907>> (14.4.2010)
- Swanston, Kenny 26.2.2010. *Texturing a Face*. [verkkodokumentti]. deviantArt:  
<<http://kingofbavaria.deviantart.com/>> (15.3.2010)
- Texture Seam problem* 3.11.2009. [verkkodokumentti]. Epic Games Forums:  
<<http://forums.epicgames.com/showthread.php?t=665046>> (24.3.2010)
- Unit Measurements?* 7.11.2009. [verkkodokumentti]. Epic Games Forum:  
<<http://forums.epicgames.com/archive/index.php/t-706271.html>> (24.2.2010)
- Wilson, Joe 17.6.2009. *Why so blue?* [verkkodokumentti]. Polycount:  
<<http://boards.polycount.net/showthread.php?t=64286&highlight=spec>> (10.4.2010)

## LIITTEET

### LIITE 1

CD-levy sisältää tiedostot ja tarkemmat ohjeet niiden käytöstä:

-Asennettava .exe tiedosto:

UDKInstall-ErikaDome.exe

-UDK:n työtiedostot:

erika\_package.upk

erika\_dome3.udk

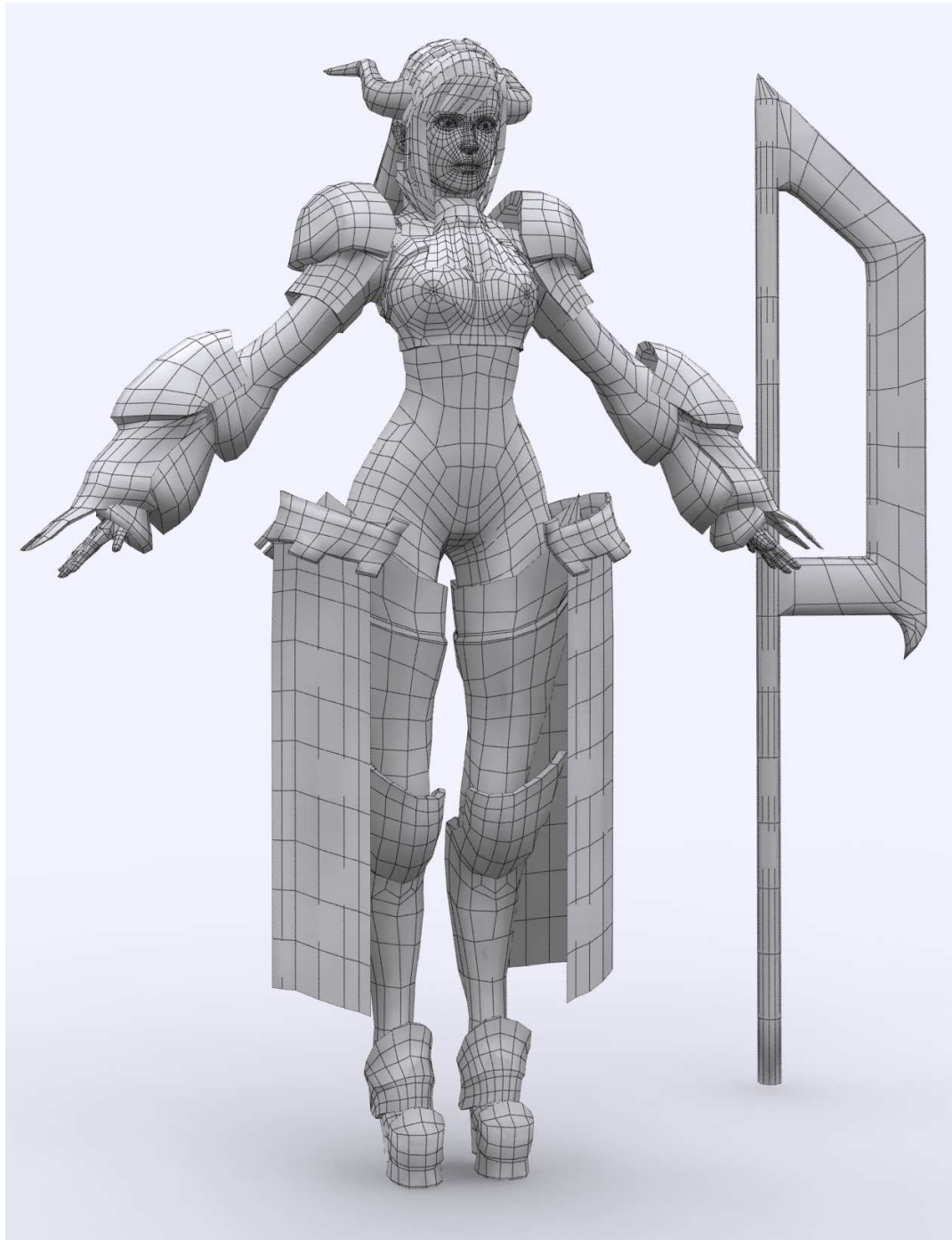
Videokaapaukset valmiista työstä:

erika\_dome.mov

erika\_turntable.mov

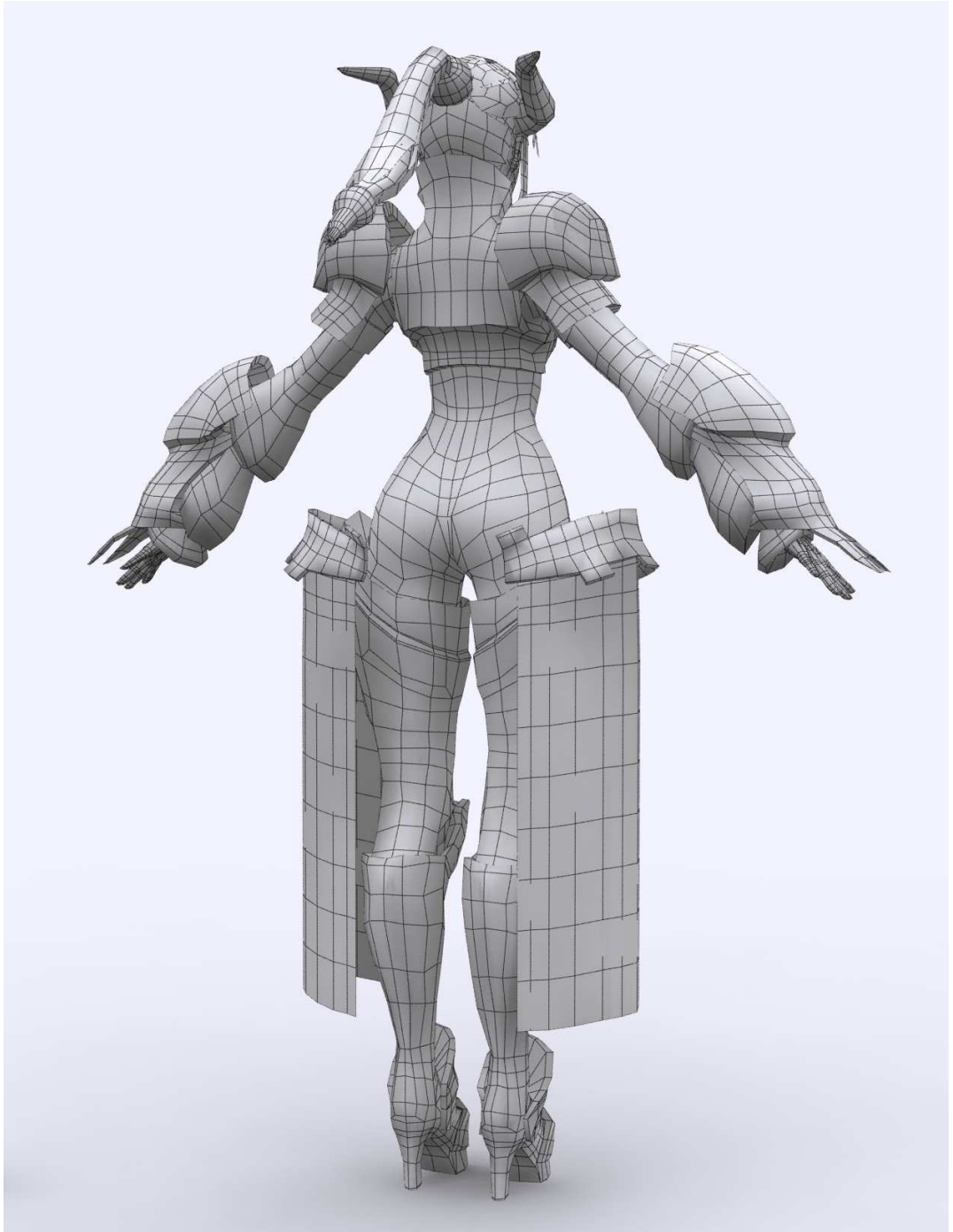
## LIITE 2

Erikan noin 9 500 polygonin versio – edestä



## LIITE 3

Erikan noin 9 500 polygonin versio – takaa





## LIITE 4

Erika Unreal-pelimoottorissa





## LIITE 5

Erika Unreal-pelimoottorissa

