

Teemu Tapanainen

Reaaliaikaisen pelihahmon toteutus

Metropolia Ammattikorkeakoulu

3D-visualisoinnin ja animaation suuntautumisvaihtoehto

Viestinnän koulutusohjelma

Opinnäytetyö

20.11.2013

Tekijä(t) Otsikko	Teemu Tapanainen Reaaliaikaisen pelihahmon toteutus
Sivumäärä Aika	31 sivua + 2 liitettä 29.11.2013
Tutkinto	Medianomi (AMK)
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	3D-visualisoinnin ja animaation suuntautumisvaihtoehto
Ohjaaja(t)	Kristian Simolin
<p>Tässä opinnäytetyössä toteutettiin reaaliaikainen pelihahmo jota pelaajan on mahdollista liikuttaa yksinkertaisessa peliympäristössä. Tavoitteena oli ottaa selville mitkä ovat keskeisiä osa-alueita pelihahmon toteutuksen kannalta ja minkä laista työkulkua tulisi käyttää.</p> <p>Pelihahmon toteutus aloitettiin luovalla suunnitteluprosessilla ja luonnosten teolla. Pelihahmosta lähdettiin tekemään realistista naishahmoa jolla on pieniä karikatyyrisiä piirteitä. Hahmo mallinnettiin ja animoitiin Autodeskin 3ds Maxilla. Opinnäytetyössä käsitellään low-poly ja high-poly -mallien ominaisuudet ja puhutaan polygonimäärien merkityksestä reaaliaikaisissa 3D-malleissa. Mallinnus luvussa käydään läpi topologian ja edge loopien merkitys. UV-kartoituksen yhteydessä tutustutaan TexTools liitännäiseen ja käydään läpi miten teksteli tiheys vaikuttaa tekstuurien resoluutioon.</p> <p>3ds Maxin sisällä mallinnettu low-poly vietiin Autodeskin Mudbox ohjelmaan jota käytettiin digitaaliseen veistämiseen. Mudboxin sisällä low-poly mallista tehtiin high-poly ja siitä voitiin luoda normaalikartta.</p> <p>Adoben Photoshopilla tekstuurit viimeisteltiin Unreal Engine 3 pelimoottoria varten. Tekstuureista materiaaliksi luvussa käsitellään gammatilaa ja sen merkitystä näyttölaitteisiin. Luvussa myös kerrotaan mitä väriprofiilia tulee käyttää kun tekstuuri viedään Unreal Engine 3 pelimoottorin sisään. Tämän jälkeen käydään läpi miten alpha kanavaa hyväksi käyttäen saadaan pelihahmolle läpinäkyviä pintoja. Kerron myös miten kaksi normaalikarttaa voidaan yhdistää Photoshopin kuvatasoja käyttäen. Tutkin myös mitä tulee ottaa huomioon kun toteutetaan diffuse- ja specularkarttoja. Tekstuurit yhdistettiin lopuksi Unreal Development Kit editorissa materiaaliksi.</p> <p>Kerron myös mitä rajoitteita on otettava huomioon kun rigataan ja skinnataan. Toteutin pelihahmolle neljä juoksuanimaatiota jotka yhdistin Unreal Development Kit editorin sisällä ja lopulta lähdekoodin avulla sain reaaliaikaisen pelihahmon toiminta valmiiksi.</p> <p>Liitteenä on kuvia ja video mistä valmista pelihahmoa voi katsoa.</p>	
Avainsanat	3D, pelihahmo, UDK, gamma tila

Author(s) Title	Teemu Tapanainen Making of a Real-Time Game Character
Number of Pages Date	31 pages + 2 appendices 29 Nov 2013
Degree	Bachelor of Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Kristian Simolin
<p>The aim of this thesis was to create a real-time game character, which the player can move around in a simple game environment. The aim was to find out essential aspects of the implementation in the making of a game character and what kind of work flow should be used.</p> <p>The Game character implementation started in the creative design process and sketching a couple of drafts. A realistic female character was made with small caricature features. The character was modeled and animated with Autodesk 3ds Max. This thesis deals with the features of the low-poly and high-poly models. The thesis also explains the importance of the polygon count in real-time 3D models. The chapter discussing modeling depicts the importance of the topology and edge loops. In the chapter, which deals with unwrapping, the TexTools plugin is introduced and the way texel density affects the texture resolution is discussed. From the 3ds Max low-poly model, the reader is exported to Autodesk Mudbox to use it for a digital sculpting. Inside the Mudbox low-poly model was made to a high-poly and it was used to create a normal map .</p> <p>The Adobe Photoshop was used to finalize the textures for the Unreal Engine 3 game engine. In the texture chapter we deal with the gamma space and its importance with respect to the display devices. The chapter also depicts which color profile should be used when the texture is exported in the Unreal Engine 3. After this, we will go through how the alpha channel can be used to create transparent surfaces for the game character. I'll tell you also how the two normal map can be combined with the Photoshop using layers. I also examine what should be considered when creating diffuse and specular maps. In the end, textures will be combined inside the Unreal Development Kit editor to create a new material.</p> <p>Also the limitations that must be taken into account are discussed when rigging and skinning a model. I created four running-animations for the character and joined them together in the Unreal Development Kit and in the end I used source code to make a real-time game character operational. Appendices contain photos and a video of the final game character.</p>	
Keywords	3D, game character, UDK, gamma space

Sisällys

1	Johdanto	1
2	Käsitteet	3
3	Suunnittelu	5
4	Toteutus	6
4.1	Mallinnus	7
4.1.1	Low-poly mallinnus	7
4.1.2	Topologia	9
4.1.3	UV-kartta	10
4.1.4	Digitaalinen veistäminen	13
4.2	Tekstuureista materiaaliksi	14
4.2.1	Gamma tila ja lineaarinen tila	14
4.2.2	Alphakanava	16
4.2.3	Normaalikartta	17
4.2.4	Diffuusi- ja specularkartat	20
4.2.5	UDK materiaali	22
4.3	Skinnaus	24
4.4	Animointi	25
5	Valmis reaaliaikainen pelihahmo	27
6	Yhteenveto	28
	Lähteet	31
	Liitteet	
	Liite 1. Valmis hahmo 3ds Maxin sisällä	
	Liite 2. UDK-tiedostot	
	Liite 3. Video valmiista pelihahmosta pelimoottorin sisällä	

1 Johdanto

Tämän opinnäytetyön tarkoituksena on toteuttaa reaaliaikainen pelattava pelihahmo. Koska aikaisempaa kokemusta pelihahmon toteutuksesta ei ollut, oli tämä aihe hyvä peruste minulle joka toivoo tulevaisuudessa uraa pelialalta. Videopelit ja urahaave toimivat myös motivaationa saada hahmo valmiiksi. Reaaliaikaiset 3D-mallit ja materiaalit ovat osa-alue josta koen olevan paljon uutta opittavaa. Lähtökohtana käytettävät ohjelmat Autodeskin Mudboxia lukuun ottamatta ovat minulle ennestään tuttuja. Miten ohjelmat toimivat keskenään tulee olemaan uutta oppimisen kannalta. Kokonaisen ihmishahmon teosta minulla ei ole aikaisempaa kokemusta. Olen aikaisemmin mallintanut ja animoinut mustekalan. Lonkerot omaavan mustekalan riggaaminen, skinnaaminen ja animointi on kuitenkin täysin oma tapauksensa kuin kahdella jalalla kävelevän ihmismallin toteutus. Erilaisten raajojen lisäksi mustekala ei ollut reaaliajassa pyörivä vaan toteutin sen lyhytanimaatiota varten. Olen mallintanut pienen pojan kädet ja ylävartalon skinnausta varten joten sormien mallinnusprosessi on entuudestaan tuttua. En ole aikaisemmin toteuttanut ihmisen rigiä joten päätin käyttää 3ds Maxin valmista biped-luurankoa, jotta riggaamiseen jäisi enemmän aikaa. Lähtökohtaisesti olen mallintanut ja teksturoinut paljon staattisia 3D-malleja. Osaan siis jo mallintaa, mutta tulen selvittämään mitä ihmisen mallintamisessa tulee ottaa huomioon.

Työnkulku ja muut huomioon otettavat asiat pelihahmon tekoon liittyen tullaan käymään läpi. Pääluvusta ainoastaan toteutus on jaoteltu useampaan alalukuun, minkä takia se on opinnäytetyöni kattavin osuus. Johdatan jokaisen pääluvun alussa lyhyesti lukijan sen luvun keskeisiin osa-alueisiin. Alaluvuissa selostan käytännön työvaiheiden lähtökohdat ja miten valmiiseen lopputulokseen päädytään.

Tulen käyttämään Autodeskin 3ds Maxia pelihahmon mallintamiseen, UV-kartoitukseen, skinnaamiseen, riggaamiseen sekä animointiin. UV-kartoituksessa käytän hyväkseni ilmaista TexTools liitännäistä. Autodeskin digitaaliseen kuvanveistoon tarkoitettua Mudboxia käytän high-poly mallinnukseen ja tekstuurien tekemiseen. Adoben kuvankäsittelyohjelmaa Photoshoppia tulen käyttämään myös tekstuurien parissa. Pelihahmon vien 3ds Maxin kautta Unreal Development Kit (UDK) -pelieditoriin joka käyttää hyväkseen Unreal Engine 3 -pelimoottoria. UDK:n sisällä toteutan materiaalit ja kokoon lopullisen reaaliaikaisen pelattavan pelihahmon. Kaikkiaan käytettäviä ohjelmia

on siis neljä jota tarvitaan tämän pelihahmon tekemiseen. Ajallisesti pitkä kestoisimmat työskentelyvaiheet ovat: mallinnus, tekstuurien toteutus ja animointi.

Adoben Photoshop on tuttu kuvankäsittelyohjelma jonka ominaisuudet hallitsen entuudestaan eikä sen toimintoihin paneuduta samalla painoarvolla kuin muihin opinnäytteentyön kannalta oleellisiin ohjelmiin. 3ds Max ja UDK toimivat ainoastaan Windows-käyttöjärjestelmän sisällä joten työskentely toteutetaan kokonaan sen sisällä. Lopullista työtä on tarkoitus tarkastella näyttöpäätteeltä joten kuvatiedostoja käsitellään aina RGB-väritilassa.

Määritän heti alussa tavoitteet ja halutun lopputuloksen pelihahmolleni. Ennen toteutusvaihetta aloitan luovalla suunnitteluprosessilla. Luon hahmolle taustatarinan ja luonnostelen hänen ulkoisen olemuksen. Luonnosten pohjalta siirryn seuraavaksi toteuttamaan hahmoa. Mallinnus osiossa selvitän mitä teknisiä rajoitteita pelihahmon toteutus sisältää ja miten ne tulisi toteuttaa. Kerron miten reaaliaikainen 3D-malli vaikuttaa mallin polygonimäärään ja mikä on low-polyn ja high-polyn ero. Mallinnusprosessissa myös selitän tarkemmin mitä käsitteet topologia ja tekseli tarkoittavat käytännössä. Tekstuurien ja materiaalien osalta selvitän niiden rajoitteet reaaliaikaisen pelimoottorin sisällä ja miten rajoitteet voidaan kiertää. Valotan myös mitä gammakorjauksella tarkoitetaan ja miten se vaikuttaa siihen miten tekstuurit toistuvat näyttöpäätteellä. Käsittelem myös mitä haasteita skinnaaminen ja animointi aiheutti.

Tavoitteena on tieto ja taito toteuttaa pelihahmo vastaisuudessa ilman että tarvitsee turvautua alan oppimateriaaliin koko ajan. Opinnäytteentyön ollessa valmis osaan varautua vastaavissa projekteissa paremmin miten paljon tulevat työvaiheet vievät aikaa. Mistä ajankohtainen tieto on saatavilla tulisi myös olla selvä kun opinnäytetyö on tuotu päätökseen. Työssä lähden oppimaan uutta ja kehittämään jo olemassa olevaa osaamistani. Koen opinnäytetyön olevan oppimisprosessi jota teen ennen kaikkea itseäni varten. Päämääränä on että lopullinen pelihahmo on visuaalisesti näyttävä kokonaisuus ja voin sen ylpeänä lisätä osaksi portfoliotani.

Lopullinen työ tulee olemaan Unreal Engine 3 -pelimoottorin sisällä missä pelaaja pystyy liikuttamaan ihmisen näköistä pelihahmoa. Kokonaista peliä ei tulla toteuttamaan eikä pelaajalle aseteta pelillisiä tavoitteita.

2 Käsitteet

High-poly on suuren määrän polygoneja sisältävä malli ja sisältää täten tarkempia yksityiskohtia.

Keyframen avulla aikajanalle voidaan tallentaa elementtien kuten luiden sijainti ja kiertoliike. Kahden keyframen välille syntyy automaattisesti animaatio kun ohjelma laskee näiden kahden keyframen väliltä puuttuvan siirtymän.

Low-poly on pienen määrän polygoneja sisältävä malli joka vaatii vähän suorituskehoja tietokoneelta.

Materiaali sisältää tekstuureja ja ominaisuuksia jotka määräävät millä lopullinen pinta näyttää.

3D-malli tai malli on verteksien, reunojen ja tasojen muodostama verkko.

Morphaaminen on animointia verteksi tasolla.

Normaali on vektori jonka suunta määrää mihin suuntaan pinnan tai verteksin etupuoli osoittaa.

Normaalikartta on RGB-kuvatiedosto joka sisältää X, Y ja Z-koordinaattien informaation RGB-värikanavilla. Pelimoottori tavallisesti laskee valon heijastuksen 3D-mallin pinnalta sen normaalien suunnan perusteella. Normaalikartan omaava 3D-mallin pinta heijastaa valon normaalikartan tiedon perusteella. Normaalikartan avulla 3D-malliin saadaan lisää yksityiskohtia lisäämättä polygonien määrää.

Outo laakso (engl. Uncanny Valley) liittyy ihmisenkaltaisuuteen ja tutunomaisuuteen. Termin on keksinyt Masahiro Mori ja sillä tarkoitetaan sitä epäluonnollisuutta mikä kohdataan kun jäljitellään realistisesti ihmistä saavuttamatta kuitenkaan täydellistä kopiota ihmisestä (Mori 1970).

Polygoni on kolmen tai useamman reunan muodostama taso. Neljän reunan polygoni muodostuu kahdesta kolmikulmaisesta tasosta.

RGB-väritila muodostuu punaisesta, vihreästä ja sinisestä värikanavasta. Näyttölaitteet tuottavat värit näistä kolmesta valonlähteestä.

Riggaus on rigin eli luurangon toteutus 3D-mallille. Animointi toteutetaan rigin luita liikuttamalla.

Skeletal mesh tarkoittaa toimivan rigin omaavaa 3D-mallia.

Skinnaus tarkoittaa rigin yhdistämistä 3D-malliin jotta esimerkiksi mallin sormet liikkuvat kun sormiluita animoidaan.

Tekseli (engl. *Texel*. Lyhenne sanoista *Texture Element*). 3D-mallin pinta voidaan visualisoida ruudukon kaltaisena ryhminä tekseleitä, kuten bittikarttakuvat esitetään ryhminä pikseleitä. Pikselit ovat kuvadataa, mutta tekselit eivät aina sitä ole (TV Tropes). Tekseli on 3D-mallin pinnan perusyksikkö.

Tekselitiheys (engl. *Texel Density*). Kuvaa miten tiheään pikselit mapataan 3D-mallin pinnalle. Erikokoiset tekselit omaavat eri määrät pikseleitä ja täten eri resoluutiot.

Tekstuuri on kaksiulotteinen bittikarttakuva.

Topologia termi viittaa tasojen ja verteksien rakenteeseen ja niiden numeeriseen järjestykseen.

UDK on pelieditori. Lyhenne nimestä Unreal Development Kit. Kehittäjä: Epic Games.

Unreal Engine 3 on pelimoottori. Kehittäjä: Epic Games.

UV-kartoitus (engl. *Unwrap*) UV-koordinaattien määrittely prosessi, missä 3D-malli avataan kaksiulotteiseksi UV-kartaksi.

3 Suunnittelu

Suunnittelu vaihe on opinnäytetyöni luovim osuus sillä haluan valmiissa työssä heijastuvan oman käden jälkeni. Lähtökohtana oli suunnitella ihmishahmo joten lähdin pohtimaan mitkä piirteet tekisivät hahmosta uniikin. Koska lopullinen työ tulisi toimimaan näyttönä osaamisestani, päädyin suunnittelemaan hahmoa joka olisi helposti lähestyttävä. En täten halunnut lähteä toteuttamaan tietyn peligenren hahmoa kuten fantasia- tai sotapelin sankaria koska en halunnut että osaamistani yhdistetään yhteen peligenreen. Pelihahmon tulisi siis olla neutraali, mutta erottua edukseen. Lähdin hakemaan inspiraatiota valokuvamalleista. Löydettyäni yhden mielenkiintoiset kasvot, oli helppoa etsiä mallin nimen avulla lisää referenssikuvia samasta mallista.

Alkuperäinen idea opinnäytetyön kannalta oli toteuttaa vanha 50-luvun huoltoasema aavikolle. Tämän idean siirsin taustatarinan muodossa pelihahmolleni. Lopullinen luonnokseni oli 50- ja 60-luvun vaihteella elänyt 30-vuotias naishenkilö jota lähdin toteuttamaan eteen päin. Vaikka en opinnäytetyössäni tee kokonaista peliä, halusin pelihahmoni kuitenkin olevan kuvitteellisen tarina painotteisen seikkailu/mysteeri -pelin henkilöahmo. Lähtökohtaisesti kuvittelin pelin olevan synkkä ja pelottava. Halusin heijastaa niitä elementtejä hahmoon. Suunnittelin toteuttavani hahmolleni melankolisen läsnäolon ja ehkä hieman surulliset kasvot. Halusin hahmolle myös karikatyyrisiä piirteitä kuten solakan vyötärön ja pitkät jalat.

Lähdin hakemaan hahmolle sitä jotain ”omaa juttua” joka toisi hahmolle persoonallisuutta ja myös pientä haastetta toteuttamiseen. Päädyin ensin pitkään huiviin joka liehuisi käsivarressa tuulen mukana (kuvio 1). Jätin idean kuitenkin hyvin aikaisessa vaiheessa pois sillä pelkäsin sen tuovan hallitsemattomia ongelmia reaaliaikaisessa ympäristössä, kuten huivin päätyminen kasvojen eteen kömpelästi. Samoista syistä jätin toteuttamatta pitkät hiukset joiden pelkäsin läpäisevän hahmon hartiat ja lisäävän turhia työtunteja ongelman ratkaisemiseen.



Kuvio 1. Luonnosprosessi.

Lopullinen 50-lukumainen hiusmalli sai inspiraationsa Rachael nimiseltä henkilöahmolta Ridley Scottin elokuvasta Blade Runner. Hahmon lopullinen ominaisuus löytyi sille ihon myötäisestä hameesta, joka toisi samalla ongelman ratkontaa sen toteutuksen kannalta.

Ideoin jo suunnitteluvaiheessa hahmon väritystä. Karikatyyriselle mallille sopisi tyyllitellyt tekstuurit, mutta halusin pyrkiä tekemään niistä realistiset koska se lisäisi lisää haastetta ja tarkastelua varsinkin kasvojen osalta. Luonnollisen iho-tekstuurin luominen tulisi olemaan varmasti yksi haaste.

4 Toteutus

Luonnosten jälkeen on toteutuksen vuoro. Käytän koko toteutuksen ajan referenssinä omia luonnoksia sekä valitsemani valokuvamallin kuvia. Pelihahmon toteutus on pitkäkestoinen prosessi, joten tulen käyttämään ohjelmien aputyökaluja ja työtapoja jotka nopeuttavat hahmon valmistumista. Aloitan hahmon mallintamisella minkä jälkeen teen

tekstuurit ja niistä materiaalit. 3D-mallin skinnaan valmiin rigin ympärille minkä jälkeen animoin kävely-animaatiot. Skeletal mesh, tekstuurit ja animaatiot viedään UDK:n sisään missä ne yhdistetään toimivaksi reaaliaikaiseksi pelihahmoksi.

4.1 Mallinnus

Tässä osiossa käyn läpi low-poly ja high-poly mallinnuksen eron toteutuksen kannalta. Kerron myös mitä on topologia ja miksi se on tärkeä hallita. Mallini tulee muodostumaan kahdesta objektista, päästä sekä vartalosta. Objektit on jaoteltu osioihin jotka eivät ole samaa polygonipintaa keskenään. Hahmoni koostuu ylävartalosta, jaloista, päästä, silmistä, korvakoruista, hiussoljesta ja kauluskorusta (kuvio 2). Turvauduin mallinnuksen aikana pääasiassa Joan of Arc hahmomallinnustutoriaaliin jossa hahmoa lähdetään mallintamaan yhdestä polygonista (Roger).

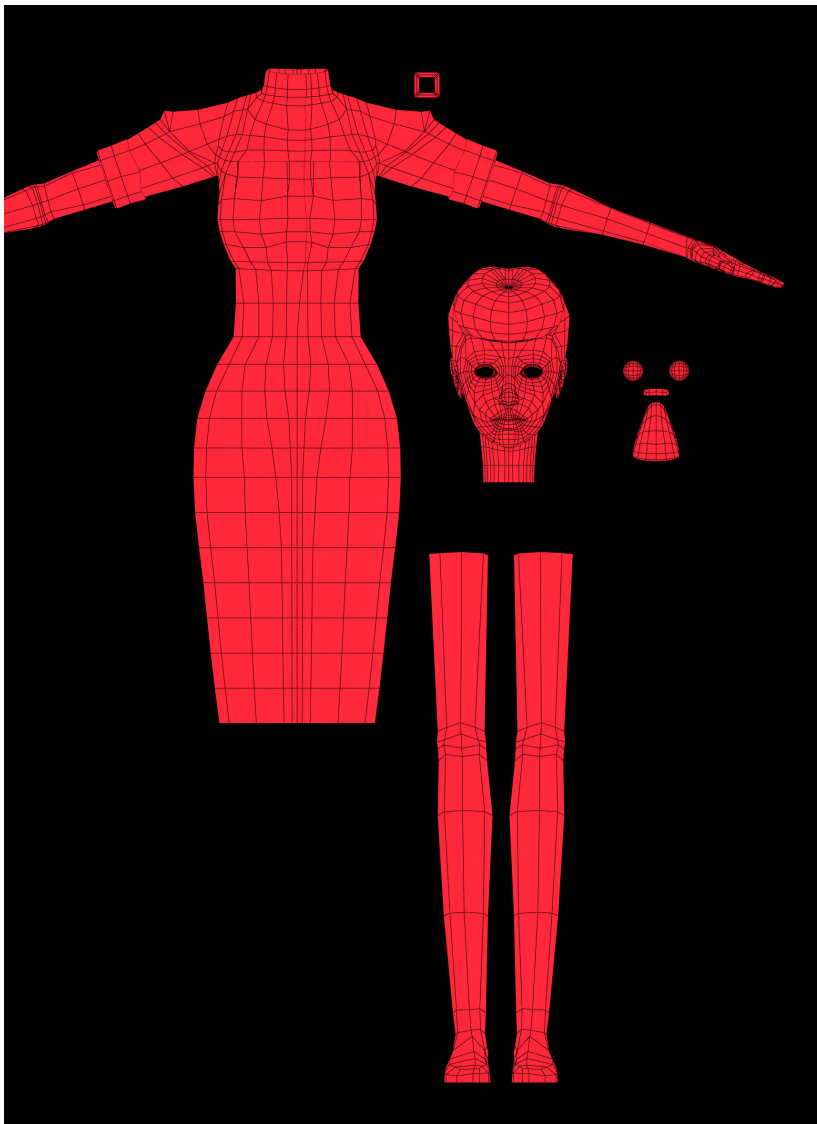
4.1.1 Low-poly mallinnus

Riippumatta siitä toteutetaanko 3D-malli millaiselle alustalle, pyritään reaaliaikaisista malleista karsimaan ylimääräiset polygonit. Mitä vähemmän yksi malli sisältää polygoneja, sitä enemmän vapautuu muille tietokoneennäytöllä samaan aikaan näkyville malleille polygoneja käytettäväksi. Aloitan mallinnus prosessin tekemällä 3D-mallin joka sisältää pienen määrän polygoneja, mistä syystä tätä mallia kutsutaan nimellä low-poly. Pienen polygoni määrän ansiosta pelimoottori jaksaa pyörittää sitä reaaliajassa. Low-poly mallin toteutan kokonaisuudessaan 3ds Maxin sisällä.

Aloitan pään mallintamisella. Käytän referenssinä valokuvamallin kasvokuvia edestä ja sivusta kuvattuina jotka tuon sisälle Maxiin. Mallinnus prosessi lähtee käyntiin silmäluomien ja huulten muotojen löytämisellä. Minun tarvitsee mallintaa vain puolet päästä sillä Maxin Symmetry -aputoiminnolla pään toinen puoli valmistuu peilattuna kopiona reaaliajassa. Pää sisältää paljon vaikeita muotoja minkä takia pyöreät linjat vaativat enemmän polygoneja kuin tasaiset ja kulmikkaat pinnat. Suun mallintamisessa turvauduin Osipan (2003) esimerkkiin toteuttaa suulle suupussi (Osipa 2003, 94-105). Suupussi sisältää kitalaen, kielen sekä ylä- ja alahammasrivin. Suupussi voisi olla oma irrallinen objektinsa, mutta päätin liittää sen pään kanssa samaksi objektiksi.

Koska päädyin päänmyötäiseen kampaukseen missä hiusrajan voi nähdä päätin mallintaa hiukset yhteen pään kanssa. Hiusrajaa selvennän myöhemmin Mudboxissa. Hahmon kasvot kannattaa mallintaa heti alkuvaiheessa sellaiseen ilmeeseen mikä on hahmon luonnollinen perusilme. Näin vältyn ylimääräiseltä työltä myöhemmin kun minun ei tarvitse morphata silmiä, suuta ja kasvojen lihaksia erikseen. Hyvä kuvareferenssi on tässä erittäin tärkeä.

Päätin katkaista jalat omaksi osakseen muusta vartalosta koska hame peitti suuren osan hahmon alavartalosta (kuvio 2). On tarpeetonta mallintaa elementtejä jotka eivät koskaan tule näkymään pelaajalle, koska tällaisten elementtien mallintaminen vie turhaa aikaa ja lisää polygonien määrää.

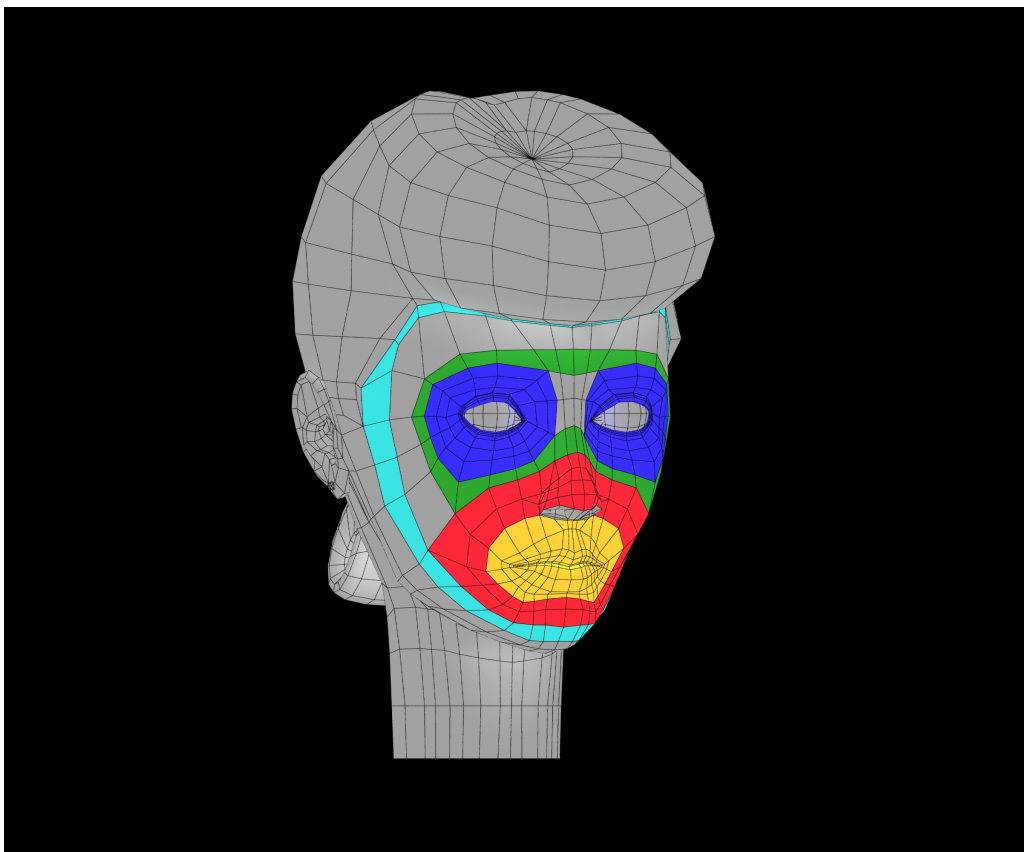


Kuvio 2. Osiin rikottu low-poly -malli

4.1.2 Topologia

Topologian tulee olla siistitty ja huolellisesti suunniteltu. Topologiaan kannattaa kiinnittää huomiota kohdissa missä animaatio vaiheessa syntyy venymistä tai supistumista, kuten kyynärtaiteissa. Edge loopit ovat tärkeitä toimivan topologian toteutuksessa. Edge loopit niiden suppeammassa merkityksessä määrittellään sarjaksi edgejä joiden silmukka kulkee aina neljän edgen leikkauspisteen kautta (The CGSociety 2007). Edge loopit kannattaa toteuttaa venyvien pintojen suuntaisesti.

Etsin netistä kasvojen mallinnusohjeita ja havaitsin että ei ollut yhtä oikeaa tapaa tehdä kasvojen topologiaa. Kuitenkin tietyt piirteet toistuivat aina, kuten silmien ja suun ympäri kulkevat edge loopit sekä nenän päältä suun alapuolelle kulkeva edge loop (kuvio 3). Kasvojen animoinnista saadaan luontevampaa kun edge loopit seuraavat pintaa muokailien lihasten käyttäytymistä. Edge loop joka kulkee nenän päältä leualle on tärkeä siinä mielessä että nenän nyrpistäminen liikuttaa lihaksia suun ja leuan alueella.



Kuvio 3. Kasvojen tärkeimmät edge loopit visualisoituna värialueittain.

Sormien kohdalla otin mallia esimerkistä jonka Archanex-nimerkin omaava henkilö oli lähettänyt CGSociety foorumille (kuva 4). Olin toteuttanut saman esimerkin mukaisen sormen topologian aikaisemmin ja havainnut sen toimivaksi ja nerokkaaksi. Putkimaisissa muodoissa kuten raajat ja sormet, edge loopien on hyvä seurata putken ympäri.



Kuvio 4. Esimerkki toimivasta sormen topologiasta. (Archanex 2009)

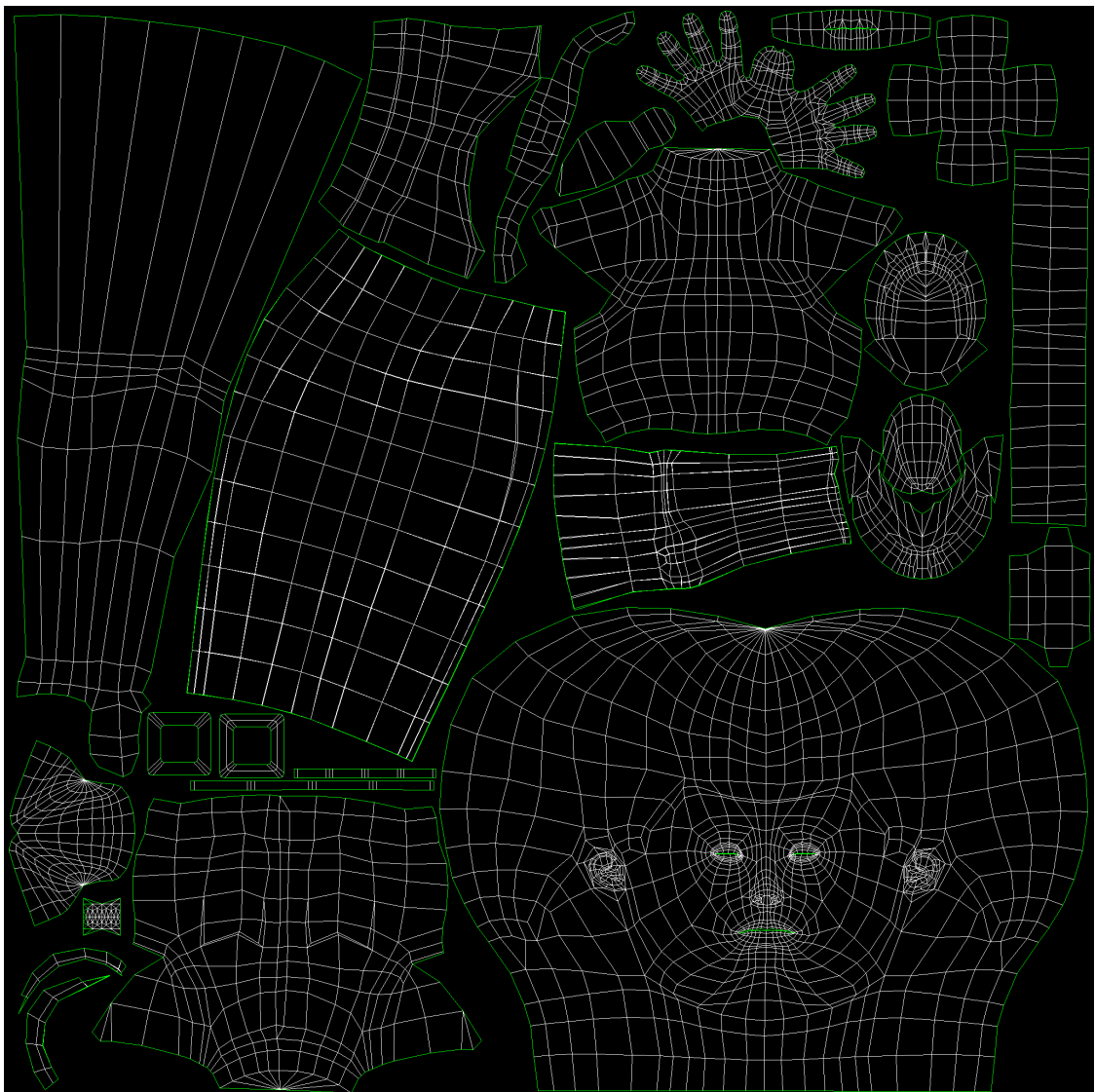
Hyvä topologia helpottaa hahmon skinnausta ja animointia myöhemmässä vaiheessa. Huonossa topologiassa edge loopit eivät myötäile lihasryhmiä ja eivät ota huomioon mallin muotoja. Staattisilla 3D-malleilla topologialla on vähemmän merkitystä, mutta skeletal mesheillä huonon topologian kanssa tulee luonnottoman näköistä animaatiojälkeä. Huono topologia saattaa ilmetä kulmikkaina polygoneina animaatiovaiheessa jos topologiassa on liian vähän edge loopeja siellä missä niitä pitäisi olla enemmän. Edge loopit saattavat olla myös liian tiiviisti siellä missä niiden ei pitäisi jolloin mallin topologia menee itsensä sisään animoidessa.

4.1.3 UV-kartta

Low-polyn ollessa valmis pystyin aloittamaan UV-koordinaattien määrittelyn jotta saisin luotua tekstuuri-kartat. UV-kartoitus prosessina muistuttaa eläimen nylkemistä. Hahmolle luodaan saumakohtia joista se avataan auki ja paloitellaan pienempiin osiin. Hahmo avataan osiin jotta tekstuurit näkyisivät tasaisena ja ilman vääristymiä mallin pinnalla. Koska hahmo on kokonaisuudessaan yhtä objektia tarvitsi minun tehdä vain yksi UV-kartta sille. Symmetrisistä kohdista tarvitsi vain puolet avata auki mikä säästi aikaa. Toisen puolen aina peilasin ja asetin pinnat UV-kartassa päällekkäin. Näin säästin myös tilaa UV-kartassa. Päällekkäiset elementit tulevat saamaan yhtäläisen materiaalipinnan mikä säästää myös teksturointi vaiheessa aikaa, mutta se rajoittaa elementtejä saamasta epäsymmetrisiä pintoja. Ihmisen kasvot eivät koskaan ole keskeltä

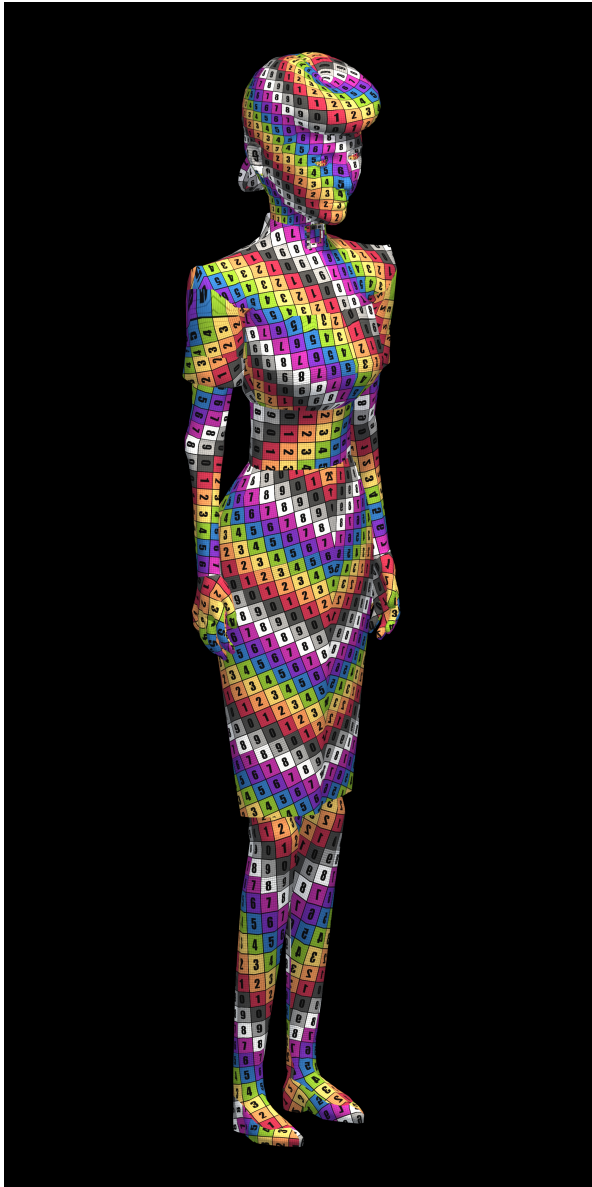
symmetriset, joten en halunnut peilata puolta päätä. Pään peilaaminen UV-kartalle olisi ollut mahdollista sillä olinhan mallintanut sen symmetriseksi. Sen sijaan että olisin mallinnus vaiheessa tehnyt kasvoista epäsymmetriset halusin tekstuureilla luoda kasvojen epäsymmetrian.

Elementtien sijainti UV-kartalla määräsi niiden auki avattu muoto. Jouduin kääntelemään objekteja jotta tyhjä tila elementtien välillä olisi minimaalinen. Elementtien väliin jäävälle tyhjälle tilalle ei ole minkään laista käyttöä. On tärkeää hyödyntää kaikki tila UV-kartassa sillä näin tekstuurit saavat käyttöönsä enemmän pikseleitä ja täten tarkkuutta (kuvio 5).



Kuvio 5. Valmis UV-kartta.

Käytin UV-kartoituksessa TexTools liitännäistä, mikä sisältää monia aputoimintoja. Tekselitiheyden saa TexToolsin avulla napin painalluksella normalisoitua samaan tiheyteen. TexTools mahdollistaa erinäköisten shakkiruutukuvioiden visualisoinnin 3D-mallille mikä visualisoi hyvin tekselitiheyttä ja tekseleiden sijaintia UV-kartassa (kuvio 6). Yhtenäinen tekselitiheys takaa UV-kartan eri elementeille saman resoluution.



Kuvio 6. Tekselitiheys visualisoituna numeroidulla ruutukuviolla.

Kun olin tyytyväinen tekseleiden kokoon ja sijaintiin, tarkistin lopuksi vielä että kaikki osat olivat UV-kartan sisäpuolella ja peilatut objektit varmasti päällekkäin. Koska mahdutin kaiken yhdelle UV-kartalle halusin tehdä kartoista 2048 x 2048 pikselin kokoiset. Suuri kokoinen kartta mahdollistaa korkea resoluutioiset tekstuurit.

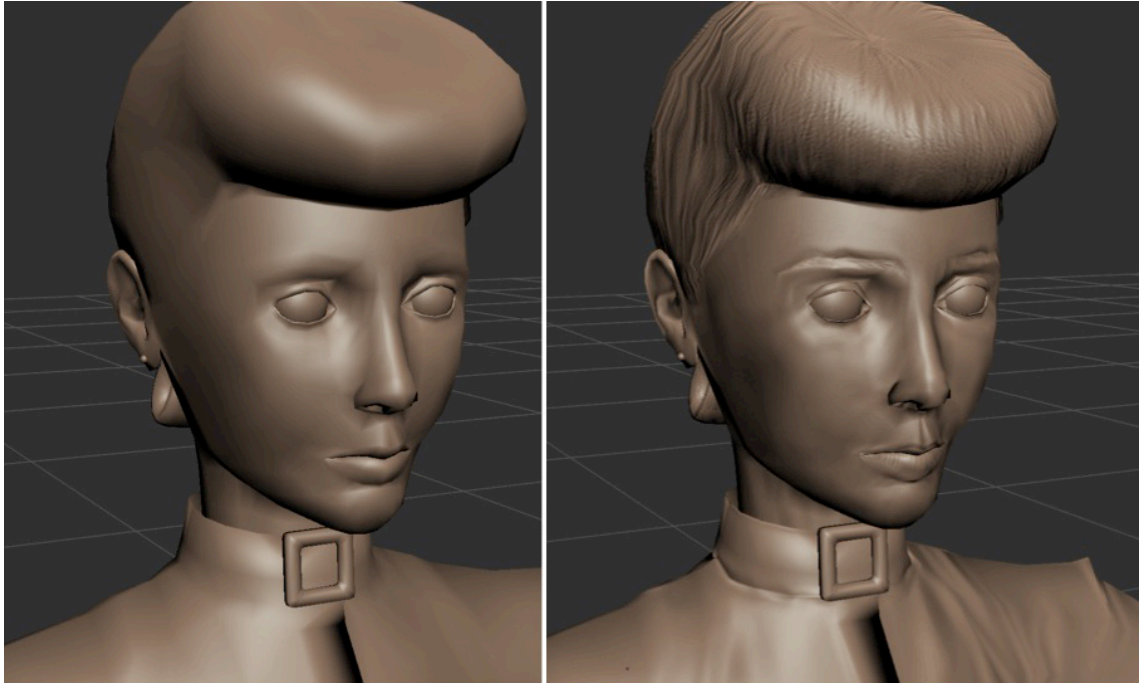
4.1.4 Digitaalinen veistäminen

Low-poly -malli ei itsessään sisällä yksityiskohtia, mutta niitä voidaan luoda digitaalisella veisto-ohjelmalla kuten Mudbox jonka pariin tässä osiossa syvennytään. Valmiin UV-kartan omaava malli viedään Mudboxiin OBJ-tiedoston avulla. Mudboxin sisällä low-poly -malliin saadaan lisää yksityiskohtia lisäämällä sen polygoni määrää. Suuren polygonimäärän omaavaa mallia kutsutaan nimellä hi-poly.

Mudbox suosittelee että 3D-mallit muodostuvat nelisivuisista polygoneista. Mudbox kyllä tukee kolmisivuisia ja nelisivuisia suurempia polygoneja (N-gon), mutta ne saattavat jossain vaiheessa tuottaa ei-toivottuja artefakteja.

Maxi antaa tehdä mallille tallennettuja polygoni valintoja joita kutsutaan nimellä Selection sets. Valinnat voidaan viedä OBJ-tiedoston mukana Mudboxiin mikä helpottaa työkentelyä kun halutaan valita tietty alue kerrallaan. Tein esimerkiksi irtonaisille objekteille kuten silmille ja koruille omat Selection setit. Tallennettujen valintojen avulla pystyin halutessani myös nopeasti piilottamaan tietyt objektit. Hampaille oli järkevintä tehdä oma Selection set sillä ne sijaitsivat pään sisäpuolella mihin oli ulkopuolelta vaikea päästä käsiksi.

Mudboxissa käytetään veistotasoja samaan tapaan kuin Photoshopissa käytetään kuvatasoja. Jokaisen veistotason sisälle voidaan tallentaa omat veistojäljet. Pystyin esimerkiksi yhdellä tasolla veistämään hiuksia ja toisella tasolla kasvojen lihaksia. Mitä enemmän lisäsin polygonien määrää sitä yksityiskohtaisempaa jälkeä pystyin veistämään. Suuri polygonimäärä kuitenkin käyttää enemmän tietokoneen muistia ja hidastaa ohjelmaa. Jotta sain vapautettua tietokoneen muistia päätin toteuttaa pään ja vartalon omien tiedostojen sisällä. Valmiin high-polyn avulla Mudboxilla rendataan normaalkartta jolla saadaan low-polyyn lisää yksityiskohtia (kuvio 7). Normaalikartan ominaisuuksista ja toteutuksesta jatkan lisää luvussa 4.2.3.



Kuvio 7. Low-poly -versio ilman normaalikarttaa ja normaalikartan kanssa.

4.2 Tekstuureista materiaaliksi

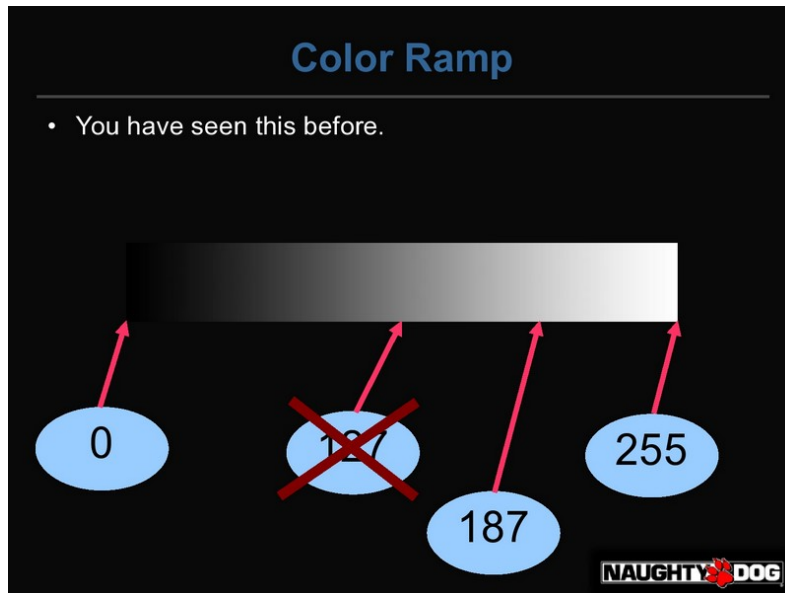
Käsittelen tässä luvussa mitä tulee ottaa huomioon kun tekstuureita aletaan tekemään. Aluksi kerron mitä on gammakorjaus ja mitä väriprofiilia tulee käyttää. Seuraavaksi kerron miten saan pelihahmolleni läpinäkyviä pintoja läpinäkyvyyskartalla. Jokaisella kartalla on oma tarkoituksensa joiden ominaisuuksia avaan. Luvun aikana käy myös ilmi tekstuurien ja materiaalien ero.

4.2.1 Gamma tila ja lineaarinen tila

Ennen kuin tekstuurit tuodaan pelimoottoriin sisään niille suoritetaan gammakorjaus. Gammakorjauksella voidaan tasapainottaa eri monitorien epälineaarista tapaa toistaa värit ja sen avulla saadaan tarkemmat arvot tummempiin väreihin (Epic Games, Inc. d). Apple kuvailee gamman tukisivuillaan seuraavasti: "Gamma on numeerinen arvo, joka kuvaa suhdetta niiden eri kirkkaus- tai luminanssitasojen välillä, jotka näyttö pystyy näyttämään tai digitaalikamera tai videokamera sieppaamaan" (Apple Inc. 2010).

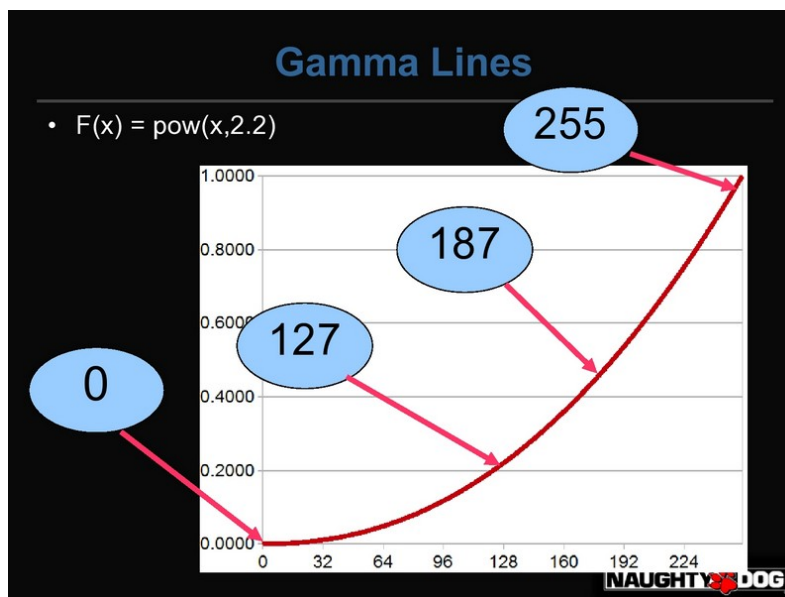
Voidaan lähteä ajattelemaan mikä on näyttölaitteelta tarkasteltavan 50% harmaan numeerinen arvo kun musta on arvoltaan 0 ja valkoinen arvoltaan 255. Oletettavasti 50%

harmaa toistuisi näiden lukujen puolivälissä toisin sanoen arvoltaan 127 tai 128. Näin ei kuitenkaan ole vaan 50% harmaa monitorilta katseltuna on arvoltaan 187 (kuvio 8) (Hable 2010).



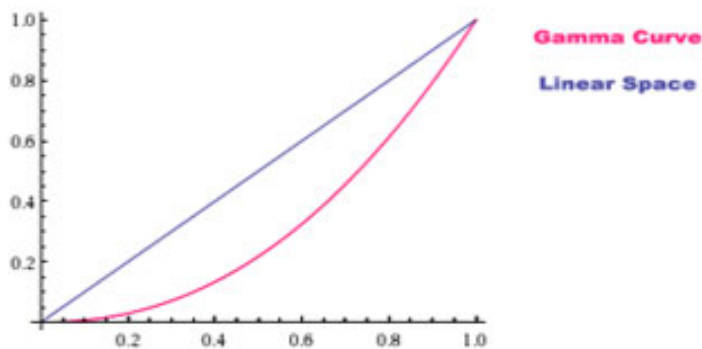
Kuvio 8. Luku 187 on liukuväriässä 50% harmaan sijainti (Hable 2010).

Tämä johtuu gamma arvosta 2.2 jota useimmat näyttölaitteet noudattavat. Gamma arvon 2.2 kuvaajassa luku 187 sijoittuu gammakäyrällä pysty akselin puoliväliin (kuvio 9). Väriarvot eivät siis nouse lineaarisessa vaan epälineaarisisessa muodossa.



Kuvio 9. Gammakäyrä (Hable 2010).

Gamma tilan asettaminen tekstuureille saadaan Photoshopissa määrittelemällä niille väriprofiilin sRGB. Väriprofiili sRGB on hyvin lähellä näyttölaitteiden gamma 2.2 arvoja (Hable 2010b). Pelimoottori automaattisesti konvertoi ennen valojen laskemista kuvat lineaariseen tilaan ja laskee valot tämän jälkeen lineaarisessa tilassa (kuvio 10). Värien yhdistyminen tapahtuu pelimoottorin sisällä myös lineaarisessa tilassa. Jos kuvat tallennettaisiin lineaarisessa tilassa niin värien yhdistymisellä olisi ei haluttu lopputulos. Lopuksi pelimoottori jälkikäsittelee lineaarisen tilan ja gammakorjaa sen vastaamaan epälineaarisisessa tilassa toimivia näyttöjä.



Kuvio 10. Käyrät kuvaavat miten gamma ja lineaarisen tilan arvot jakautuvat. (Epic Games, Inc. b)

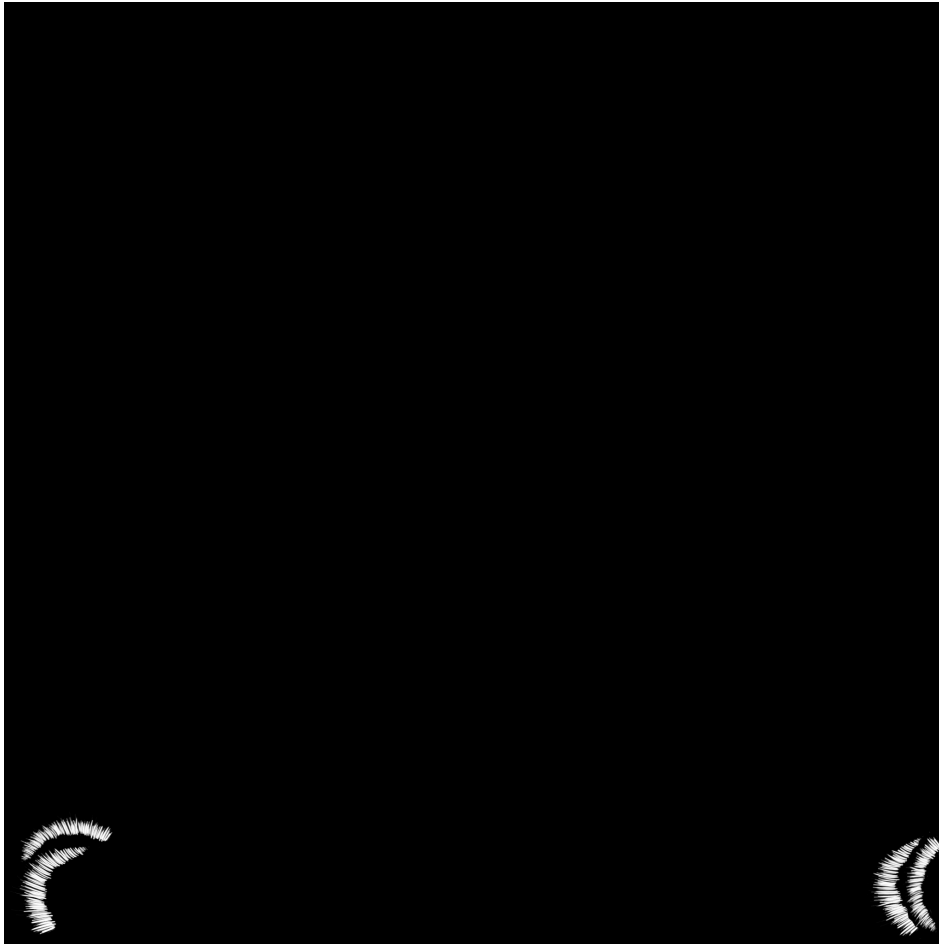
Normaalikartat eivät tarvitse gammakorjausta koska niiden väriarvoilla on erityinen merkitys. Normaalikartan tekstuuriasetuksissa tulisi sRGB-valinta olla pois päältä. (Epic Games, Inc. e)

4.2.2 Alphakanava

Pelihahmoni silmäripset toteutin mallinnus vaiheessa omiksi polygoneiksi jotka kiinnitin silmäluomiin. Pelimoottori kuitenkin renderaa ripset yhtenäisenä polygonina jollei pelimoottorille kerrota niiden tarvitsevan olla osittain läpinäkyvät. Osittain läpinäkyvän niistä saa alphakanavan avulla.

Alphakanava on ylimääräinen värikanava punaisen, vihreän ja sinisen värikanavan lisäksi. Alphakanava on väriltään mustavalkoinen ja sisältää harmaan erisävyjä. Koska läpinäkyvyysaste voidaan luoda omalle värikanavalle ei sitä varten tarvitse luoda omaa tekstuuria vaan se voidaan liittää esimerkiksi diffuusi-tekstuurin mukaan. Valkoiset alu-

eet edustavat näkyvään jäävää tilaa ja musta väri läpinäkyvää tilaa. Piirsin siis valkoiset ripset mustalle pohjalle Photoshopin sisällä (kuvio 11). Jotta Alphakanava tallentuisi on käytettävä targa-kuvaformaattia ja tallentaessa valita kohta 32 bits/pixel.



Kuvio 11. Pelihahmon ripset läpinäkyvyyskartalla.

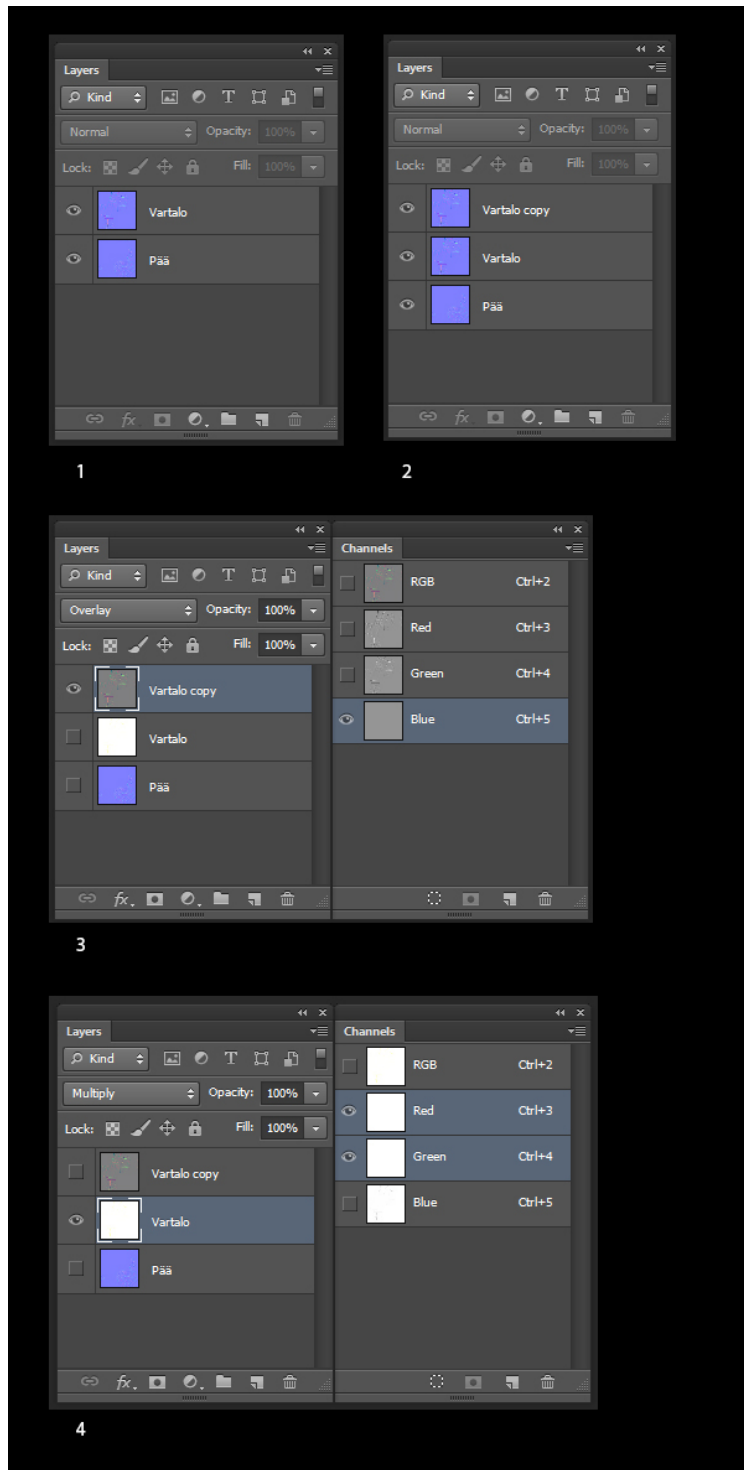
4.2.3 Normaalikartta

Normaalikartalla saan low-poly -malliini lisää yksityiskohtia jolloin sen pinta muistuttaa hi-poly -mallia, mutta ilman hi-poly -mallin suurta polygoni määrää. Pelimoottori taittaa valon suunnan normaalikartan informaation mukaan sen sijaan että se laskisi valon heijastuksen low-poly -mallin normaalien suunnan mukaisesti. Mudboxissa luodut pään ja vartalon normaalikartat minun täytyi yhdistää yhdeksi normaalikartaksi (kuvio 12). Yksi tapa olisi ollut leikata Photoshopissa toisesta normaalikartasta tarvittavat kohdat ja liittää ne toiseen. Pään ja vartalon elementit kuitenkin olivat sen verran erillään toisistaan että halusin kokeilla toista tapaa yhdistää kartat.



Kuvio 12. Pään ja vartalon normaalikartat.

Molemmat normaalikartat kannattaa tuoda samaan Photoshop-tiedostoon omina kuvatasoinaan (kuvio 13: kohta 1). Helpoimmalta vaikuttava tapa on asettaa ylemmän kuvatason sekoitus tilaksi sulauttava (engl. Overlay), mutta tällöin sininen värikanava ei lisää informaatiota vaan poistaa sitä. Covarrubias (2009) kertoo tähän kuitenkin ratkaisun opetusvideollaan *Photoshop - Combine Normal Maps*. Ylempi kuvataso joka halutaan yhdistää alempaan kuvatasoon kopioidaan kahdeksi kuvatasoksi. Kopioin tässä tapauksessa vartalon normaalikartan (kuvio 13: kohta 2). Päälimmäisen kopion sininen värikanava valitaan ja täytetään 50% harmaalla värillä (#808080). Näin ollen punainen ja vihreä kanava sisältää informaatiota, mutta sininen jää 50% harmaaksi. Ylimmän kopion sekoitus tilaksi valitaan sulauttava (kuvio 13: kohta 3). Vartalon alempi kopio valitaan ja täytetään sen punainen ja vihreä kanava valkoisella värillä (#ffffff). Punainen ja vihreä kanava ei nyt sisällä informaatiota vaan ovat valkoisia, mutta sininen kanava edelleen säilyttää informaationsa. Tämän tason sekoitus tilaksi valitaan kertova (engl. Multiply) (kuvio 13: kohta 4). Normaalikarttojen yhdistäminen on nyt toteutettu onnistuneesti kuvatasoja käyttäen.



Kuvio 13. Kahden normaalikartan yhdistäminen Photoshopissa.

Kahden normaalikartan yhdistämiseen on myös helpompi tapa esimerkiksi CrazyBump ohjelman avulla. Normaalikarttojen yhdistäminen käsin Photoshopissa auttoi kuitenkin ymmärtämään mitä kaikkea normaalikarttojen yhdistäminen vaatii prosessina. Normaalikartat eivät myöskään olleet vielä täysin täydelliset, sillä joissain saumakohdissa esiin-

tyi pieniä eroavaisuuksia. Viimeistelin ne saumakohtat molemmilta puolilta samalla värillä missä ongelmakohtia esiintyi. Normaalikartta oli nyt valmis siirrettäväksi UDK:n sisään (kuvio 14). UDK-editorin sisään tuotaessa on suositeltavaa käyttää formaattia TC_NormalmapUncompressed kohdassa Compression Settings. Uncompressed vie enemmän muistia kuin TC_Normalmap, mutta tuottaa korkealaatuisemman lopputuloksen.



Kuvio 14. Valmis normaalikartta.

4.2.4 Diffuusi- ja specularkartat

Diffuusi-tekstuuri määrittää materiaalin värin. Tekstuurin pitäisi kuvastaa sellaista pintaa joka on valaistu 100% kirkkaalla valkoisella valolla (Epic Games, Inc. b). Huomasin ilokseni Mudboxissa olevan mahdollisuus myös maalata 3D-malleja ja luoda tekstuurit ohjelman sisällä. Toteutin diffuusi-kartat lähes kokonaan Mudboxsilla, sillä maalamaan pystyi suoraan kolmiulotteisen mallin pinnalle. Työprosessi mahdollisti reaaliajassa näkemään oman työn lopputuloksen ja oli oivallinen erityisesti orgaanisten pintojen maalaamiseen. Ohjelmassa pystyi siirtymään veistotyötilasta UV-karttanäkymään

ja maalata pintoja suoraan lopulliseen tekstuuriin. Väritin paidan ruutukuvion UV-näkymässä koska tämä vähensi vääristymiä kuviossa. Vaatteet muistuttavat UV-kartassa auki avattuja vaatekaavoja, minkä takia niiden diffuusi-kartta on luonnollista toteuttaa esimerkiksi kuvankäsittelyohjelmassa.

Rendasin Maxilla ambient occlusion -kartan mikä tummentaa kolot ja pinnat jotka sijaitsevat lähellä toisiaan. Tummat pinnat nenän sieraimissa ja hihojen sisäpuolella muistuttavat varjokohtia ja luo täten syvyyttä malliin. Yhdistin Ambient occlusion -kartan photoshopissa diffuusi-kartan päälle sillä en nähnyt syytä miksi tekisin siitä erillisen tekstuurin (kuvio 15).



Kuvio 15. Diffuusi ja ambient occlusion -kartat yhdistettynä.

Speculartekstuurin avulla voidaan hallita kuinka paljon hahmon eri alueet heijastavat valoa ja minkä värisenä ne kiiltää. Mustaväri specular-kartassa ei heijasta yhtään valoa ja vaaleat pinnat taas heijastavat enemmän. Hahmon kiiltävimmät pinnat löytyvät silmistä, huulista ja kiiltävistä koruista. Kasvojen ihosta saa luonnollisemman kun nenä ja otsa kiiltävät enemmän poskiin verrattuna (kuvio 16). UDN ohjeistaa että aikaisemmin

useista syistä johtuen ihon specular-väriksi valittiin sininen tai oranssi koska silloiset renderointiohjelmat eivät tuottaneet muuten haluttua lopputulosta. Iho heijastaa valon kuitenkin takaisin monen muun materiaalin tavoin takaisin valkoisena (Epic Games, Inc. b).

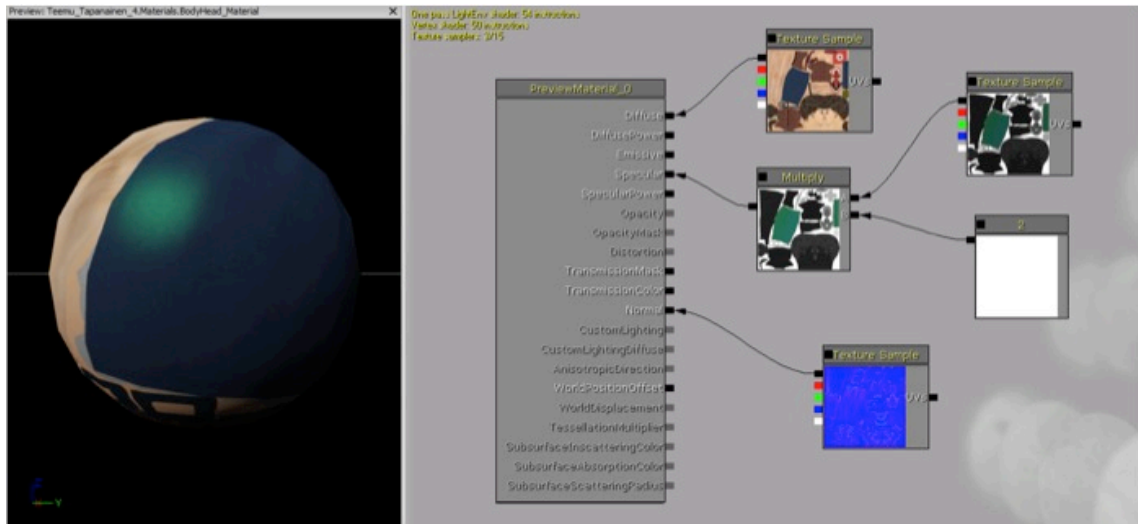
Hameen specular-väriksi valitsin kuitenkin vihreän, koska halusin sen olevan siftaavaa kangasta. Siftaavalla kankaalla (engl. Shot silk) tarkoitetaan kangasta jonka loimi- ja kudelankana on käytetty useampaa kuin yhtä väriä. Kudonta menetelmän ansiosta kankaan pinta saa sille ominaisen metallin kirjavan hohdon. Värillisellä specularvärillä pystyin samalla näkemään miten se käyttäytyisi eri valaistuksessa.



Kuvio 16. Specular-kartta

4.2.5 UDK materiaali

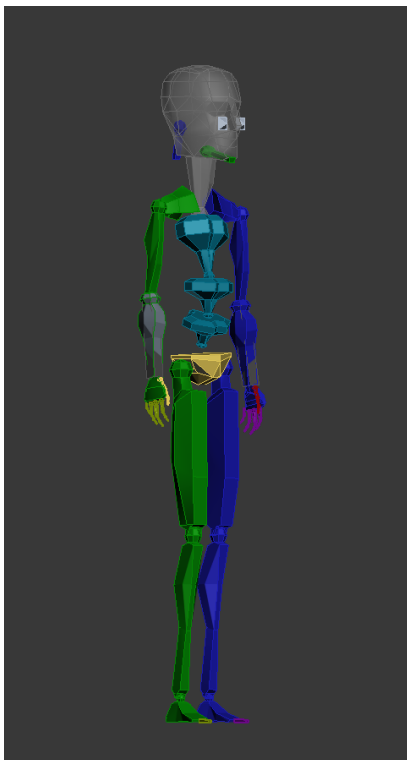
Tekstuureiden ollessa valmis tuodaan ne UDK:n sisään missä ne yhdistetään yhdeksi materiaaliksi. Node pohjaisessa materiaali editorissa pystyin vahvistamaan specular-kartan voimakkuutta multiply-nodella (kuvio 17).



Kuvio 17. Pelihahmon materiaali UDK:n node pohjaisessa materiaali editorissa.

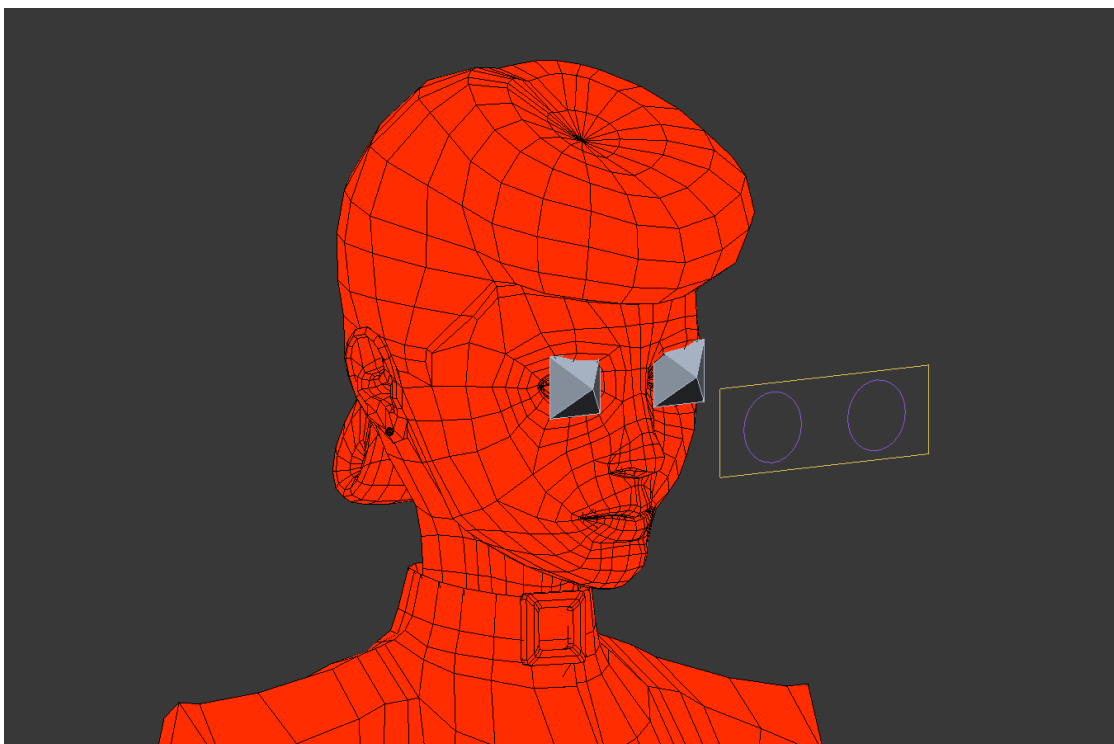
4.3 Riggaus

Käytin 3ds Maxin omaa biped-luurankoa jonka luut skaalasin malliini sopivaksi. Tein lisäksi silmille, leualle ja hiusponnarille omat luut (kuvio 18).



Kuvio 18. Biped ja lisätyt silmä-, leuka- ja hiusponnariluut.

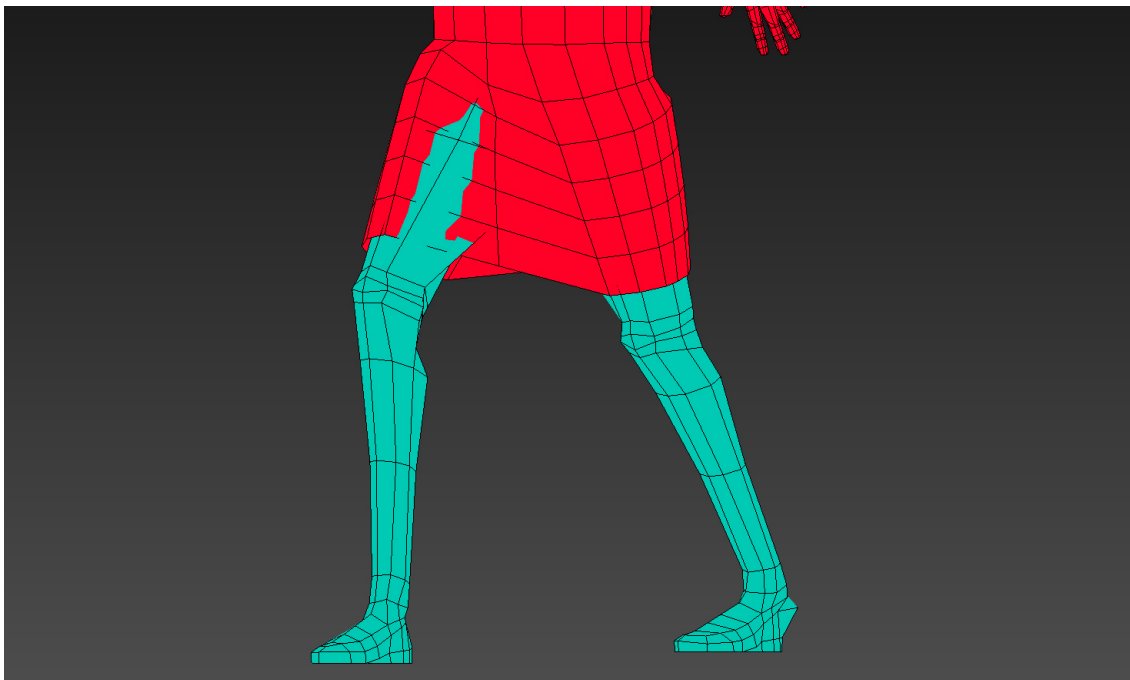
Tein silmille lisäksi hallinta-apurin, mitä silmä-luut seuraavat kun apuria liikutetaan (kuvio 19). Hallinta-apurin avulla silmiä ei tarvitse animoida erikseen ja se helpottaa kun hahmon halutaan katsovan tiettyä kohtaa pelimaailmassa. 3D-malliin vaikuttavien luiden maksimimäärä luurangossa on 75. Luiden totaalinen maksimimäärä luurangossa on 256 (Epic Games, Inc. a). Jos tämä määrä ylittyy, toteutetaan toinen renderointi kerta hahmolle.



Kuvio 19. Silmäluut ja silmien hallinta-apuri.

4.3 Skinnaus

Rigin päälle skinnaaminen oli aikaa vievää puuhaa. Jalat oli helppo skinnata, mutta niiden ympärillä kulkeva hame tuotti suuria ongelmia. Jalat tahtoivat aina lävistää hameen ja se tuotti outoja muotoja jalkojen liikkua ristiin (kuvio 20). Ymmärsin lopulta että olin tehnyt suunnittelu virheen hameen topologiassa. Hameen topologia itsessään oli erittäin siisti, mutta en ollut ottanut huomioon että topologian pitäisi myötäillä jalkojen tulevaa liikettä eikä hameen omaa ulkomuotoa. En lähtenyt korjaamaan topologiaa uudelleen sillä olisin joutunut toteuttamaan UV-kartan uudestaan. Pystyin kuitenkin hieman siirtelemään hameen verteksejä ilman että UV-kartta hajoaisi.



Kuvio 20. Hahmon jalat läpäisevät hameen.

Skinnatessa on otettava huomioon että yksi verteksi saa vaikuttaa enintään neljään luuhun kerrallaan. Pienimmän painoarvon omaavan luun vaikutukset poistetaan verkseiltä joiden painoarvot on laskettu useamman kuin neljän luun kesken (Epic Games, Inc. a).

4.4 Animointi

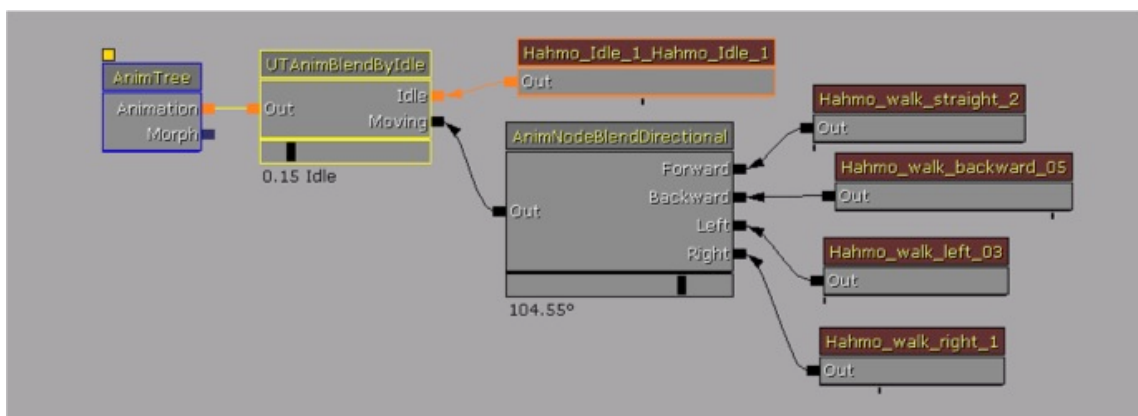
Animaation toteutin bipedin omilla keyframe-työkaluilla. Toimettomana seisova ihminen ei ole eleiltään täysin staattinen. Ihminen tekee aina pieniä liikkeitä kehollaan kuten sormien ja silmien liikuttelua. Tämän takia toteutin pelihahmolleni animaation jossa tämä seisoo joutilaana tehden pieniä kehon liikkeitä. Koska halusin hahmostani pelattavan täytyi sille myös tehdä juoksu-animaatio. Animoin hahmolle eteenpäin ja sivuttain juoksun sekä takaperin kävelyn. Sivuttain juoksu oli ongelmallinen toteuttaa sillä siitä oli vaikea saada luonnollisen näköistä. Koin animaation keinotekoisuuden johtuvan siitä että ihminen ei normaalistikkaan kävele sivuttain. Tarkastelin UDK:n valmiita hahmoille ja niiden kävelyanimaatiota. Huomasin että sivuttain kävellessä hahmojen päät eivät kääntyneet niin jyrkästi kävely suuntaan vaan hahmojen huomio on kiinnittynyt koko kävelyn ajan eteenpäin. Pelihahmojen katse onkin usein suunnattu sinne mihin

pelaaja näkee. Korjasin hahmoni päätä ja koko ryhtiä ja sain animaatiosta tällä tavoin hieman luontevamman.

Animaatioliikkeet toistavat itseään minkä takia hahmon täytyy päätyä aina siihen asentoon mistä animaatio alkaa. Toteutin animaatiot 3ds Maxin sisällä liikutellen rigin luita minkä jälkeen vein animaatiot FBX-tiedoston mukana UDK:n sisälle.

FBX-tiedoston mukana ei tarvitse viedä 3D-mallia vaan pelkän animoidun luurangon vieminen riittää. Ohjelmat kuten Autodeskin Motion Builder pystyy tallentamaan useamman animaation FBX-tiedoston sisään, mutta 3ds Max ei tätä toimintoa tue. Tiedostot saavat sisältää vain yhden animaation yhtä skeletal meshiä kohden, minkä takia animaatiot tuodaan yksitellen AnimSet-editorin sisään. Skeletal mesh voidaan tuoda yhdessä tiedoston mukana jos halutaan käyttää vain yhtä animaatiota. Tällöin pelieditori nimeään AnimSet:n automaattisesti luurangon päälun mukaan. (Epic Games, Inc. c.)

Hahmon liikkeet saadaan toimimaan reaaliajassa kun ne yhdistetään vuokaavio-pohjaisessa AnimTree-editorissa (kuvio 21). AnimTreen käskysarjoja muuttamalla pystytään ohjaamaan animaatioiden keskinäistä käyttäytymistä ja kytkeä ne toisiinsa. Kun pelaaja ei liikuta pelihahmoa työkalu toistaa animaatiota missä hahmo seisoo joutilana. Pelaajan painaessa tietokoneen näppäimistön suuntanäppäimiä AnimTreen käskysarja laukaisee juoksuanimaation. Kun pelaaja haluaa juosta viistosti etu vasemmalle, se sulauttaa eteen ja vasemmalle suuntaavan juoksuanimaation yhteen.



Kuvio 21. Animaatiot UDK:n AnimTree Editorin sisällä.

5 Valmis reaaliaikainen pelihahmo

Hahmon toteutuksen jälkeen pelimoottorin on pystyttävä tunnistamaan sen. Skeletal meshiin, AnimSettiin ja AnimTreehen viitataan lähdekoodissa jotka kirjoitetaan tekstieditorilla (liite 2) (TraptCG.com). Joka kerta kun lähdekoodia muutetaan, UDK täytyy käynnistää uudelleen jotta se voi tarkastaa onko koodissa tapahtunut muutoksia. Virhe koodissa johtaa virheilmoitukseen ja koodi on korjattava ennen kuin UDK:lla voi jatkaa töitä. Tein nopean aavikko ympäristön missä pystyin testaamaan pelihahmoa reaaliajassa. Ensimmäisellä testikerralla hahmo käveli ilmassa maan yläpuolella ja kamera seurasi liian lähellä hahmoa. Ratkaisin leijunta ongelman laskemalla mallin sijaintia alas päin skeletal mesh -asetuksista.

Unreal Kismetin avulla voidaan toteuttaa käskysarjoja jotka vaikuttavat pelattavuuteen ja pelin kulkuun. Kismetillä pystyin korjaamaan kamerakulman ja kameran etäisyyden hahmosta. Pienien ongelmien jälkeen olin onnistunut tavoitteessani ja pystyin liikku-
maan itse toteutetulla pelihahmollani reaaliaikaisessa peliympäristössä (kuvio 22).



Kuvio 22. Valmis reaaliaikainen pelihahmo pelimoottorin sisällä.

6 Yhteenveto

Peliala elää ja kehittyy minkä johdosta tämän työn toteutustapa ja työprosessi voi olla 10 vuoden päästä eri. Huomasin että sinä aikana kun tein omaa hahmoani, olivat toiset työprosessit jo ottaneet vaihtoehdon sijasta normin aseman. Työnkulkuni olisi nyt toinen jos tekisin uuden pelihahmon. Ensimmäinen vaihe olisi high-poly mallin tekeminen Mudboxissa sen sijaan että lähtisin mallintamaan low-polya 3ds Maxissa. Opinnäyte-työtä tehdessä huomasin tämän vakiintuvaksi työtavaksi kun tehdään realistista pelihahmoa. High-poly mallin työstäminen ilman valmista low-poly pohjaa antaa enemmän vapautta mallintamiseen Mudboxin sisällä. High-poly mallista saa myös paremman kuvan halutusta loppu ilmeestä, minkä avulla on selkeämpää todeta milloin työ on valmis. Seuraava työvaihe olisi mallintaa high-polyn pohjalta low-poly malli. Tähän on pääsääntöisesti kaksi tapaa. Lähestyisin helpointa tapaa ratkaista asia ja antaisin ohjelman tehdä high-polysta low-poly version minkä jälkeen siistisin topologian käsin. Jos tämä lähestymistapa tuottaisi liikaa ongelmia topologiaan jo alkutekijöissä, mallintaisin low-polyn kokonaan itse alusta high-polyn ollessa referenssimallina alla.

Hameen topologia tuotti vaikeuksia animointivaiheessa tuottaen kulmikkaita vääristymiä. Toteuttaisin hameen topologian nyt niin että se myötäilisi jalkojen liikettä. Yksi ratkaisu olisi luoda hameelle omat luunsa jotka ottaisivat huomioon jalkaluut. Haasteena oli myös saada sulavan näköistä animaatiota aikaiseksi. Syytän huonosta animaatiosta kuitenkin 3ds Maxin biped-luurankoa jonka logiikka tuotti suurta päänvaivaa. Bipedia animoidaan käyttämällä bipedin omaa käyräeditoria ja animaatiotyökaluja. Työkalut eivät toimineet samalla tavalla mihin olin Maxin sisällä tottunut. Biped esimerkiksi tallentaa automaattisesti keyframen kämmenluulle jos on tarkoitus tallentaa keyframe vain käsivarsiluulle. Bipedin toiminnot olivat mielestäni usein epäkäytännöllisiä ja johti animaation epätoimivuuteen. Tulevaisuudessa opettelen tekemään oman animaatioluurangon jotta vältyn valmiin luurangon tuomilta ongelmilta.

AnimTree myös vaati oman opettelunsa. Ensin animaatiot eivät meinanneet millään sulautua yhteen, mutta löysin lopulta asetuksia jolla animaatiot pystyi ajoittamaan keskenään yhteen. En saanut kuitenkaan animaatioita täysin sulavaksi keskenään ja luovutin koska en ollut varma mikä aiheutti ongelman. AnimTree ja animaatioiden yhteen sulautus vaatisi vielä jatkotutkimuksia.

En ollut täysin tyytyväinen valmiin hahmon visuaaliseen ilmeeseen. Hahmon taustatarinassa ei ollut lopputuloksen kannalta ongelmia, mutta tyyllinen toteutus ei toiminut niin hyvin kuin olisin toivonut. Alkuperäiset ratkaisut yhdistää malliin karikatyyrisiä muotoja kuten pitkät hoikat jalat ja kapean vyötärön, eivät toimineet mielestäni yhdessä materiaalien kanssa jotka tavoittelivat realistista ilmettä. Malli yksinään ja tekstuurit yksinään olivat mielestäni onnistuneita, mutta ongelma syntyy kun nämä yhdistetään. Suunnittelu vaiheessa lupaavalta näyttävä hahmo ei näyttänyt enää valmiina yhtä huolitellulta kun eri palaset olivat asettuneet paikoilleen. Asian voisi korjata joko tekemällä tekstuureista tyylliteltyjä tai mallintaa hahmo realistisia naisen mittoja mukaillen. Luulen että mallinnan vielä joskus lähitulevaisuudessa uuden ihmismallin ja pyrin tällä kertaa enemmän realismiin toteutus 3D-grafiikan avulla kiinnostaa.

Koen että realismia tavoittelevat tekstuurit etenkin kasvoissa tuovat hahmon lähelle outoa laaksoa. Hahmon silmät tuntuvat hiukan tyhjiltä varsinkin silloin kun hahmo nähdään liikkeessä. Silmienräpäytys ja silmien liike tuo hahmoon inhimillisyyttä, mutta niiden animoimiseen voisi kiinnittää vielä enemmän huomiota. Myös silmän pupillin laajeneva ja supistuva liike olisi voinut olla mielenkiintoista animoida morphaamalla. Haluaisin seuraavassa hahmossani toteuttaa rintakehälle sellaisen morph-animaation että vaikuttaa kuin hahmon keuhkot laajenisivat ja hahmo hengittäisi.

Eniten uutta opin Mudbox-ohjelmasta joka ei ollut entuudestaan tuttu. Olin erittäin mielissäni kun huomasin kuinka helppoa Mudboxin käyttäminen oli ja sen työkalujen opetteluun ei kulunut liikaa aikaa. Suurin yllätys oli kun huomasin että ohjelmalla on mahdollisuus myös luoda tekstuureita. Mudboxilla tekstuurien tuottaminen oli kaiken lisäksi erittäin helppoa ja nopeaa verrattuna esimerkiksi jos tekstuurit olisi luonut Photoshopin sisällä. Tulen vastaisuudessa varmasti vielä kokeilemaan digitaalisia veisto-ohjelmia.

Oman arvioni mukaan pelihahmon toteutukseen meni turhan paljon aikaa, mutta luulen että tekisin nyt saman paljon nopeammin kun tiedän mistä aloittaa ja mitä ottaa huomioon. Kasvojen topologian osaisin nyt tehdä ilman että minun tarvitsisi turvautua tutoriaaleihin tai opastaviin kuvamateriaaleihin. Pienet vastoinkäymiset aina hidasti toteutusprosessia, mutta ongelmien ratkointa on aina kuulunut osaksi 3D-mallinnusta ja -animaatiota.

Olin aikaisemmin työskennellyt UDK:n parissa tehden pelikenttää, mutta tämä oli ensimmäinen kerta kun toteutin skeletal meshiä pelimoottorin sisälle. Pelihahmon kohdalla oli tiukemmat polygonimäärä rajoitukset kuin perinteisessä animaatiohahmossa. Normaalikarttojen käyttö oli myös keskeinen aihe kun halutaan tehdä yksityiskohtaisia 3D-malleja peliympäristöön. Pidin UDK:n heikkoutena sitä että lähes kaikki on tehty pelintekijälle valmiiksi. Ongelma on että kaikki valmiiksi pelimoottorissa pyörivä on oleuksena ensimmäisestä tai kolmannesta persoonasta kuvattu räiskintäpeli. Valmiit asetukset toimivat kun halutaan nopeasti testata toimiiko jokin lähdekoodiin muutettu asia UDK:n valmiilla elementeillä. Erityisesti kun halutaan tehdä pelikenttää pääsee pelikenttää testaamaan heti halutessaan. Mukautetun pelihahmon toteutus on kuitenkin turhan vaikeaa sillä UDK:n omat asetukset pitää kumota omalla lähdekoodilla ja en löytänyt ohjeita missä mukautetun hahmon toteutukseen olisi syvennetty paremmin. Lähdekoodista 90% oli sellaista mitä en ymmärtänyt, mutta olisin silti halunnut oppia käyttämään. En esimerkiksi saanut poistettua asesta tähtäintä näytöltäni ja pelimoottori käsitteli pelitilaa räiskintä pelinä. Pystyin myös ampumaan vaikka hahmollani ei tietystikään ollut asetta.

Opinnäytteentyössäni opin uusia käytännön asioita ja osa työvaiheista toimi kertauksena. Vaikka tekisin nyt monet työvaiheet eri tavalla onnistuin kuitenkin tavoitteessani luoda reaaliaikaisen pelattavan pelihahmon.

Lähteet

- Apple Inc. 2010. Mac OS X v10.6: Tietoja gamma-arvosta 2,2. [verkkosivu]
<http://support.apple.com/kb/HT3712?viewlocale=fi_FI> (27.11.2013)
- Archanex 2009. The CGSociety Forum. [verkkosivu]
<<http://forums.cgsociety.org/showthread.php?t=760602>> (10.11.2010).
- Covarrubias, Leo 2009. Photoshop - Combine Normal Maps. Vimeo. [verkkosivu]
<<http://vimeo.com/8025133>> (12.2.2013)
- Epic Games, Inc. a. Creating Animations for the Unreal Engine. UDN. [verkkosivu]
<<http://udn.epicgames.com/Three/CreatingAnimations.html>> (1.4.2013)
- Epic Games, Inc. b. Epic Games Texturing Guidelines. UDN. [verkkosivu]
<<http://udn.epicgames.com/Three/TexturingGuidelines.html>> (1.4.2013)
- Epic Games, Inc. c. FBX Animation Pipeline. UDN. [verkkosivu]
<<http://udn.epicgames.com/Three/FBXAnimationPipeline.html>> (1.4.2013)
- Epic Games, Inc. d. Gamma Correction in UE3. UDN. [verkkosivu]
<<http://udn.epicgames.com/Three/GammaCorrection.html>> (27.11.2013)
- Epic Games, Inc. e. Texture Properties. UDN. [verkkosivu]
<<http://udn.epicgames.com/Three/TextureProperties.html>> (27.11.2013)
- Hable, John 2010a. Linear-Space Lighting (i.e. Gamma). Filmic Games. [verkkosivu]
<<http://filmicgames.com/archives/299>> (27.11.2013)
- Hable, John 2010b. Gamma: 360 vs PS3. Filmic Games. [verkkosivu]
<<http://filmicgames.com/archives/14>> (27.11.2013)
- Mori, Masahiro 1970. The Uncanny Valley. Android Science. [verkkosivu]
<<http://www.androidscience.com/theuncannyvalley/proceedings2005/uncannyvalley.html>> (28.11.2013)
- Osipa, Jason 2003. Stop staring, Facial Modeling and Animation Done Right. Alameda: SYBEX Inc.
- Roger, Michel. Modeling Joan of Arc. 3DTotal. [verkkosivu]
<<http://www.3dtotal.com/ffa/tutorials/max/joanofarc/joanmenu.php>> (15.4.2011)
- The CGSociety 2007. Edge loop. [verkkosivu]
<http://wiki.cgsociety.org/index.php/Edge_Loop> (12.2.2013)
- TraptCG.com. Create a custom game character in UDK Part 01. [verkkosivu]
<<http://www.traptcg.com/createacustomgamecharacterinudkpart01flv>> (21.3.2013)
- TV Tropes. Pixel vs. Texel. [verkkosivu]
<<http://tvtropes.org/pmwiki/pmwiki.php/Main/PixelVsTexel>> (26.11.2013)

Valmis hahmo 3ds Maxin sisällä

Hahmo.max

BodyHead_Diffuse.tga

BodyHead_NormalMap.png

BodyHead_Specular.png

Head_Opacity.png



UDK-tiedostot

Hahmo_Teemu_Tapanainen_4.udk

Teemu_Tapanainen_4.upk

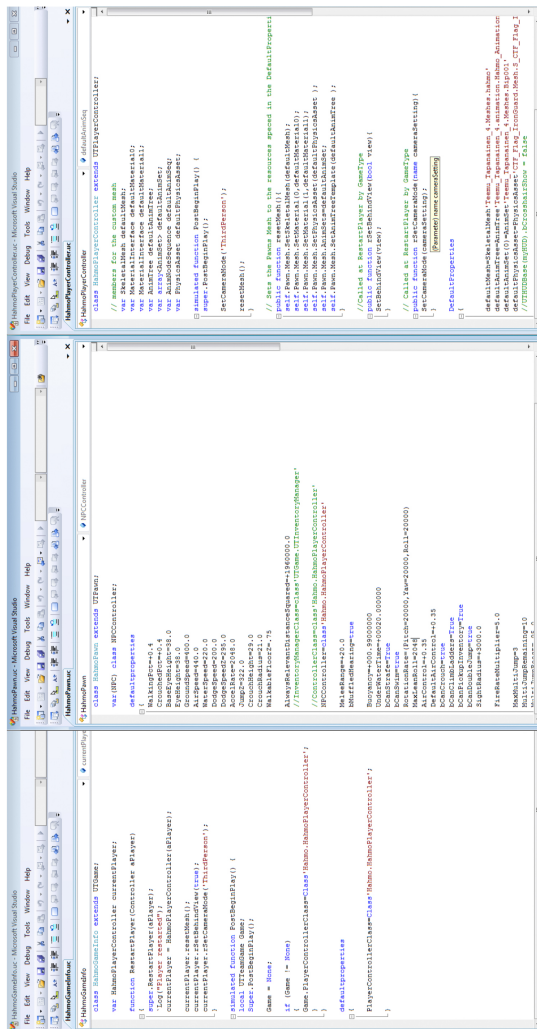
UnrealScript-tiedostot:

HahmoGameInfo.uc

HahmoPawn.uc

HahmoPlayerController.uc

DefaultEngine.ini



Video valmiista pelihahmosta pelimoottorin sisällä

Hahmo.avi