

KARELIA-AMMATTIKORKEAKOULU
Viestinnän koulutusohjelma

Joonas Paakkunainen

PELIHAHMOJEN LUOMINEN PELIMOOTTORIIN

Opinnäytetyö
Marraskuu 2016



OPINNÄYTETYÖ
Marraskuu 2016
Media-alan koulutusohjelma

Länsikatu 15
80110 JOENSUU
013 260 600

Tekijä
Joonas Paakkunainen

Nimeke
Pelihahmojen luominen pelimoottoriin

Tiivistelmä

Opinnäytetyön tavoitteena on toteuttaa kaksi kolmiulotteista pelihahmoa alusta loppuun. Työssä perehdytään myös työskentelymenetelmiin, joita pelihahmojen tekeminen vaatii. Hahmojen luomisprosessi aloitetaan konseptipiirrokselta ja päättyy valmiiseen hahmoon pelimoottorissa.

Opinnäytetyöraportissa esitellään 3D-mallintamisen käsitteet, jotka ovat olennaisia hahmomallintamisen työprosessin ymmärtämiseksi. Päätöksenteon tukena eri työvaiheissa on käytetty pääasiassa 3D-mallintamiseen ja hahmomallintamiseen liittyvää kirjallisuutta, sekä verkkoartikkeleita.

Hahmomallinnusprosessin aikana perehdytään eri ohjelmistoihin, työkaluihin ja työskentelyvaiheisiin joita käytetään kun pyritään kaupalliseen tasoon. Pelihahmojen luominen on tekninen sekä luova prosessi, joka muuttuu nopeasti ajan myötä ja vaihtelee osittain sen mukaan, kuka on tekijä. Hahmomallintajan on tunnettava nämä prosessit onnistuakseen työssään ja pystyäkseen toimimaan mukana pelialan työyhteisöissä.

3D-pelihahmoja voi toteuttaa eri tavoilla, ja eri työvaiheissa tehdyt ratkaisut vaikuttavat siihen, kuinka onnistunut lopputulos on. Tämä opinnäytetyöraportti käy läpi yhden reitin ja esittelee sen lopputulokset sekä voi toimia samalla ohjeena lukijalle, kuinka hahmoja tehdään.

Kieli
suomi

Sivuja 47
Liitteet 4
Liitesivumäärä 4

Asiasanat
3D, Pelihahmo, 3DS Max, ZBrush, 3D-Coat, 3D-mallinnus, 3D-visualisointi



THESIS
November 2016
Degree Programme in Communication
Länsikatu 15
FI 80110 JOENSUU
FINLAND
013 260 600

Author
Joonas Paakkunainen

Title
Creation of Game Characters for a Game Engine

Abstract

The objective of this thesis is to create two 3D-game characters from start to finish and to understand the working methods that are required to create them. The creation process of characters starts from a concept drawing and ends in a finished character in a game engine.

The thesis introduces the concepts of 3D-modeling that are essential for understanding the work process of character modeling. 3D-modeling and character modeling related literature and web articles are used to support the decision making in different work phases.

During the character creation process the aim is to understand different programs, tools and working phases that are used to achieve commercial quality. The creation of game characters is also a technical and creative process, which changes rapidly over time and also varies, depending on the maker. The character artist has to know these processes to succeed in his/her work and to be able to function in game industry work communities.

3D-game characters can be created in many ways, and different choices that are made will affect how good the end result is. This thesis will go through one way and present its results, and can also act as a possible guide for the reader on how to make characters.

Language
Finnish

Pages 47
Appendices 4
Pages of Appendices 4

Keywords
3D, Game character, 3DS Max, ZBrush, 3D-Coat, 3D modeling, 3D visualization

Sisällysluettelo

1	Johdanto.....	7
2	Käsitteet	8
2.1	Polygoni	8
2.2	3D-mallien rakenne	9
2.3	Digitaalinen veistäminen	10
2.4	Tekstuurikartat	11
2.5	UV-Kartoitus.....	12
2.6	Riggaaminen	13
3	Ohjelmistot	14
4	Hahmomallintaminen nykypäivänä.....	15
5	Pelihahmojen toteutus.....	17
5.1	Työprosessin suunnittelu	17
5.2	Hahmojen konseptisuunnittelu	18
5.3	Mallintaminen.....	20
5.3.1	Mallintamisen suunnittelu	20
5.3.2	Korkearesoluutioinen malli.....	21
5.3.3	Matalaresoluutioinen malli	27
5.4	UV-Kartoitus ja teksturointi.....	31
5.4.1	UV-Kartoitus	31
5.4.2	Projektointi	33
5.4.3	Tekstuurikartat	35
5.5	Hahmon riggaus.....	37
5.6	Hahmo pelimoottorissa	39
5.6.1	Importtaus	39
5.6.2	Materiaalit	41
6	Pohdinta	42
	Lähteet	45

Liitteet

Liite 1	Ensimmäinen hahmo valaistuna
Liite 2	Ensimmäinen hahmo ilman valaistusta tai varjoja
Liite 3	Toinen hahmo valaistuna
Liite 4	Toinen hahmo ilman valaistusta tai varjoja

Sanasto

Ambient occlusion	Nimitys tausta- eli ambienttivalon muodostamasta varjostuksesta (Birn 2008).
Bump mapping	Tekniikka jonka avulla lisätään näennäisesti pintaan rakennetta muuttamatta oikeasti pinnan geometriaa (Puhakka 2008, 248).
Dynamesh	ZBrushin työkalu jolla voidaan luoda polygoniverkko uusiksi (Pixologic 2016).
Glossiness-kartta	Tekstuurikartta, joka määrittää kuinka leveitä kiiltokartan muodostamat kiiltopisteet ovat (Autodesk 2016).
High-poly-malli	3D-malli, joka muodostuu suuresta määrästä polygoneja (Flavell 2010, 115).
Kiiltokartta	Tekstuurikartta, joka määrittää kuinka kappaleen pinta heijastaa valoa takaisin (Puhakka 2008, 2016).
Low-poly-malli	3D-malli, joka muodostuu pienestä määrästä polygoneja (Thompson 2007, 144).
Normaalikartta	Osa bump mapping -tekniikkaa. Tekstuurikartta, joka luo illuusion mallin pinnalla olevista yksityiskohdista käyttämällä valaistusta ja varjoja. (Thompson 2007, 144).
Polypaint	Polypaint-työkaluilla maalaamista suoraan 3D-mallin pinnalle ZBrushissa (Pixologic 2016).
Renderöinti	3D-grafiikassa renderöinnillä tarkoitetaan sitä kun kolmiulotteinen esitys muutetaan kaksiulotteiseksi kuvaksi. Renderöinnin voi jakaa esi- ja reaaliaikaiseen renderöintiin. Esi-renderöinnillä tarkoitetaan monimutkaisia visuaalisia esityksiä joissa käytetään paljon polygoneja tai korkearesoluutioisia tekstuureja mikä tekee kuvien luomisen reaaliajassa mahdottomaksi. Reaaliaikaisella renderöinnillä tarkoitetaan visuaalisia esityksiä joiden on tarkoitus pyöriä reaaliaikai-

	sesti ruudulla, mikä yleensä tarkoittaa vähintään yli 20 kuvaa sekunnissa. (Slick 2016.)
Retopology	Retopologialla tarkoitetaan 3D-mallin rakenteen eli polygoniverkon topologian uudelleen tekoa (Theodore 2016).
Riggaus	Luiden luomista animointia varten (Athias 2010).
UV-kartoitus	3D-mallin tekstuurikoordinaattien kartoitusta tekstuurikarttoja varten (Autodesk 2016).
Varjostin	Varjostin tai sävytin (eng. Shader) on tietokonegrafiikassa käytettävä pinnan värin määrittelevä algoritmi. Algoritmi käyttää eri komponenttien yhdistelmiä (mm. Tekstuurikarttoja) pinnan lopullisen värin muodostamiseen. (Puhakka 2008, 385.)
Värikartta	Tekstuurikartta, joka määrittää 3D-mallin pinnan värit yleensä ilman valaistusta tai varjoja (Russel 2014).

1 Johdanto

Opinnäytetyöni käsittelee 3D-pelihahmojen luomista alkaen suunnittelusta ja päättyen valmiiseen animoitavaan hahmoon pelimoottorissa. Käyn läpi hahmojen luomisprosessin eri työvaiheet ja perustelen tekemäni valinnat. Tarkoitukseni on tehdä kaksi erilaista hahmoa, jotka eroavat hieman toisistaan tyyliiltään ja hieman tekniseltä toteutukseltaan. Pyrin selvittämään, millaisia työprosesseja PC- ja konsolipelien pelihahmojen luomiseen kuuluu ja kuinka saavutetaan kaupallinen taso. Selvitän, mitä ohjelmistoja ja plugineja käytetään nykyaikaisissa pelituotannoissa 3D-hahmojen luontiin sekä pyrin löytämään työskentelytapoja joilla saadaan aikaan mahdollisimman hyvä lopputulos.

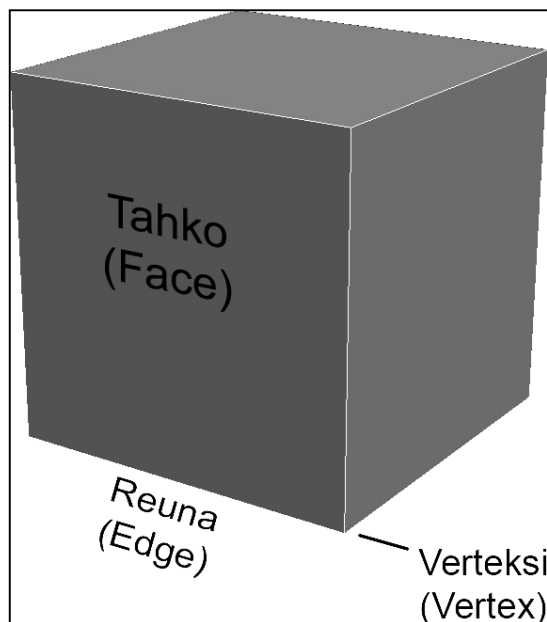
Minulla on entuudestaan kokemusta 3D-mallintamisesta yleisellä tasolla sekä olen mallintanut monenlaisia 3D-malleja, joten voin hyvin keskittyä hahmomallintamiseen liittyviin työvaiheisiin ja ongelmiin. Käytän hahmojen luontiin saatavilla olevia ohjelmia ja plugineja, joista osa on ilmaisia ja osa maksullisia ohjelmia. Näitä samoja ohjelmia käytetään nykyaikaisissa kaupallisissa pelituotannoissa ammattilaisten toimesta.

Työprosessi koostuu hahmojen konseptoinnista, mallintamisesta, teksturoinnista ja riggaamisesta. Lopulta hahmot siirretään Unity-3D pelimoottoriin, missä niiden 3D-mallit, tekstuurikartat ja animaatiotiedostot muodostavat lopullisen kokonaisuuden, joka on reaaliajassa liikkuva pelihahmo.

2 Käsitteet

2.1 Polygoni

Jokainen kolmiulotteinen pelihahmo koostuu näytölle piirretyistä polygoneista. Polygoni koostuu sen kulmapisteistä eli vertekseistä, reunoista ja sen keskiosasta, eli tahkosta (kuva 1). Verteksejä liikuttamalla 3D-avaruudessa (X-, Y-, ja Z-akseleilla) pystytään muokkaamaan polygonin muotoa ja asentoa. Kun polygonit yhdistetään toisiinsa reunoistaan ne voivat muodostaa miljoonista polygoneista muodostuvia 3D-malleja, kuten hahmoja tai esineitä. Useampien polygonien muodostamaa verkostoa sanotaan polygoniverkoksi.



Kuva 1. Kuudesta polygonista muodostuva objekti.

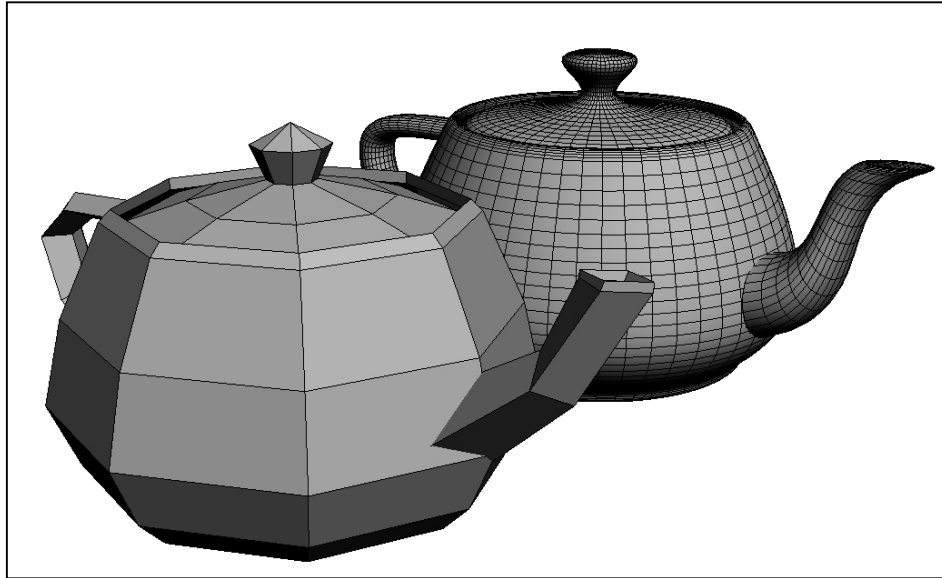
Polygonit voivat koostua mistä tahansa määrästä reunoja ja verteksejä. Verteksejä ja reunoja tarvitaan vähintään kolme, jotta ne muodostaisivat kolmion. Tietokoneen on helppo käsitellä kolmioita, koska ne ovat yksinkertaisin mahdollinen muoto, josta polygoni voi muodostua. Tämä tekee niistä pelimoottorien tarkoituk-

seen sopivan sillä pelimoottorien täytyy pystyä renderöimään monia 3D-malleja ruudulle kerralla. (Chopine 2011, 22.)

2.2 3D-mallien rakenne

Nykyään 3D-mallinnuksessa malleista käytetään termejä high-poly ja low-poly. High-poly-mallin ja low-poly-mallin määritelmät vaihtelevat jatkuvasti tietokoneiden laskentatehon myötä, kun pelimoottorit pystyvät pyörittämään yhä suurempia määriä polygoneja ruudulla.

Yksinkertaisesti ilmaistuna high-poly tarkoittaa 3D-mallia, jolla on suuri määrä polygoneja (kuva 2). Mutta se mitä tarkoittaa ”suuri määrä” on suhteellista, mikä tarkoittaa että myös termi high-poly on suhteellinen. Se mikä on ”suuri määrä” polygoneja riippuu yleensä asiayhteydestä ja sen ajan tietokoneiden laskentatehosta. Nykyajan low-poly olisi ollut vielä muutama vuosi sitten high-polya, ja nykyään elokuvissa käytettävät 3D-mallit ovat high-polya suhteessa pelien käyttämiin malleihin, jne. Pelien 3D-mallien mallinnusprosessin yhteydessä taas high-poly-malli ja low-poly-malli tarkoittavat hieman eri asioita. Low-poly on se lopullinen malli, joka muodostaa kolmiulotteisen pelihahmon joka näkyy lopulta pelissä, kun taas high-poly on malli, jota käytetään vain väliaikaisesti normaalikarttojen luomiseen. Low-poly-mallilla on siis jokin kattoraja kuinka suuri määrä polygoneja sillä saa olla ja jotta se pyörisi pelimoottorissa. High-poly-mallia käytetään ainoastaan normaalikarttatekstuurin luomiseen, joten periaatteessa sen polygonien määrä on rajoittamaton. Luotavan normaalikarttatekstuurin koko asettaa yksityiskohtien tallennuskapasiteetin rajat. (Huhtamaa 2007, 2.)



Kuva 2. Low-poly ja high-poly-mallit.

2.3 Digitaalinen veistäminen

Digitaalinen veistäminen, tai 3D-veistäminen, on tietokoneohjelmistolla tapahtuvaa 3D-mallin muotoilua. Digitaalinen veistäminen on siis periaatteessa kuin saaveen kaivertamista oikeassa elämässä. Ohjelmilla voi 3D-mallia venyttää, nipistää, kaivertaa syvennyksiä tai venyttää ulokkeita. Työskentely tapahtuu yleensä piirtopöydän tai piirtonäytön avulla. Digitaalisessa veistämässä käytetään yleensä jopa miljoonia polygoneja sillä sitä käytetään pääasiassa high-poly-mallien luomiseen, yleensä normaalikarttoja varten.

Digitaalinen veistäminen on suhteellisen uusi tapa mallintaa. Sen avulla voidaan luoda yksityiskohtia, mitkä olisivat melkein mahdottomia tehdä perinteisillä mallinnusohjelmilla. Tämä tekee digitaalisesta veistämisestä suosituksen tavan tehdä realismiin pyrkiviä 3D-malleja. Digitaalista veistämistä käytetäänkin yleensä orgaanisten mallien tekoon.

2.4 Tekstuurikartat

Yleisimmät tekstuurikartat joita käytetään 3D-mallien kanssa peleissä ovat värikiilto- ja normaalikarttatekstuurit. Käytettävät tekstuurityypit riippuvat pelimoottorin rajoituksista ja halutusta visuaalisesta ilmeestä. Kaikkia tekstuureja ei siis tarvitse välttämättä käyttää. Pelimoottorit hyväksyvät monesti vain tekstuurikokoja, jotka ovat aina kaksi kertaa suurempia, kuin edellinen pienempi tekstuurikartan koko, esimerkiksi 32x32, 64x64, 128x128, jne. (KatsBits 2016).

Tekstuurikartoista välttämättömin on värikartta (eng. Color map) jota voidaan kutsua myös diffuusikartaksi (eng. Diffuse map) (Birn 2008). Värikartta määrittelee 3D-mallin pinnan värin ja voi sisältää myös varjoja. Värikartat yksinään voivat kantaa pitkälle, jos käytetään oikean maailman materiaaleja pohjana. (Russel 2014.) Toinen värikartan kaltainen tekstuurikartta on albedokartta. Albedokartoissa kirkkauseroja on tasoitettu poistamalla tekstuurista suuntavalojen vaikutukset eli varjot ja valoheijastukset (Pietiläinen 2016, 25).

Normaalikartta (eng. Normal map) on tekniikka jota käytetään esittämään pienempiä yksityiskohtia 3D-mallien pinnoilla. Normaalikarttatekniikka ei muuta mallin todellista geometriaa, joten mallin siluetti pysyy samana. Ääriviivoja lukuun ottamatta malli saadaan näyttämään lähes samalta kuin alkuperäinen korkearesoluutioinen malli, josta normaalikartta on luotu. Normaalikarttatekstuurissa pikselin RGB-arvot edustavat mallin normaalin X-, Y- ja Z-komponentteja. (Huhtamaa 2007, 2.)

Normaalikartat ovat osa Bump mapping –tekniikkaa, kuten kerrotaan Unity3D-pelimoottorin manuaalissa (Unity Technologies, 2016). Yleisesti puhuttaessa bump mappingilla tarkoitetaan mitä tahansa tekstuurikarttatekniikkaa jonka avulla luodaan yksityiskohtia mallien pinnoille käyttämättä siihen polygoneja, eli hahmon varsinaista geometriaa. Yksityiskohdat syntyvät siis käyttämällä tekstuuri-, eli kuvatiedostoja. Aikaisemmin ennen normaalikarttoja käytössä oli ensimmäinen

bump mapping -tekniikka joka oli periaatteessa vain mustavalkoinen kuva josta pelimoottori tulkitse pinnan eri syvyyksiä ja renderöi varjoja 3D-mallin pinnalle sen mukaan ja tällä tavoin sai mallin pinnan näyttämään yksityiskohtaisemmalta. Normaalikartat taas käyttävät hyväksi värikarttojen kaikkia kolmea RGB-kanavaa mikä antaa huomattavasti paremman lopputuloksen.

Kiiltokartta (eng. Specular map) on yleensä mustavalkoinen tekstuurikartta joka vaikuttaa siihen kuinka paljon 3D-mallin pinta heijastaa valoa takaisin, eli kuinka paljon mallin pinta kiiltää. Kiiltokartta luo monille materiaaleille niille ominaisen kiillon. Täysin valkoinen kiiltokartta heijastaa kaiken valon takaisin, kun taas täysin musta ei heijasta mitään. Kiiltokartoissa voidaan käyttää myös värejä jos halutaan antaa kiilloille jokin väri. Pelimoottorissa pinta kiiltelee sen mukaan missä kulmassa valo osuu 3D-mallin pintaan. Yleensä renderöintimoottoreissa kiiltävyys voidaan määrittellä yleisesti koko kappaleelle, mutta tämä on harvoin paras vaihtoehto, sillä tosielämässä muun muassa pintakäsittely, pöly, lika tai vaikkapa sormenjäljet vaikuttavat pinnan kiiltoon (Pietiläinen 2016, 28). Tästä syystä myös peleissä käytettäville 3D-malleille luodaan kiiltokartat. Ilman kiiltokarttaa voidaan metallisen materiaalin vaikutelma luoda vaikkapa piirtämällä värikarttaan kiilto pisteitä.

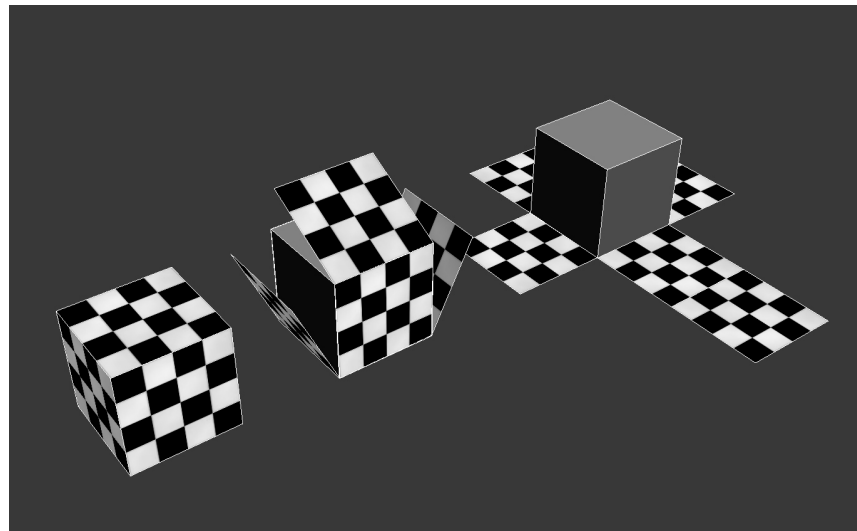
Glossiness-kartta on mustavalkokartta joka toimii yhdessä kiiltokartan kanssa. Glossiness-kartta määrittää kuinka leveitä kiiltokartan muodostamat kiiltopisteet ovat ja vaikuttaa täten oleellisesti siihen kuinka materiaalin pinta kiiltää. Tarkemmin sanottuna glossiness-kartta esittää pinnan epätasaisuutta mikä vaikuttaa siihen millä tavalla valo heijastuu takaisin. Glossiness-kartta on mustavalkokartta joka laitetaan kiiltokarttatekstuurin alpha-kanavaan.

2.5 UV-Kartoitus

Ennen kuin tekstuurit voidaan liittää 3D-mallin pinnalle, täytyy niitä varten 3D-mallit kartoittaa 2D-kuvia varten tekniikalla, jota nimitetään UV-kartoitukseksi.

UV-kartoitus tarkoittaa tekniikkaa, jolla ”kääritään” tai ”paketoidaan” 2D-kuvatekstuuri 3D-mallin pinnalle. ”U” ja ”V” ovat nimiä kuvan pysty- ja vaakatason akselleille, koska ”X”, ”Y” ja ”Z” -akselit ovat jo 3D-avaruuden koordinaattien käytössä.

Jokainen piste UV-kartassa vastaa verteksiä 3D-mallissa. Pisteitä yhdistävät viivat taas vastaavat 3D-mallin polygonin reunoja. Viivojen ja pisteiden väliin jäävät tilat taas vastaavat polygonin pintaa 3D-mallissa. (Blender Foundation 2016.) Periaatteessa UV-kartoitettu 3D-malli avautuu räjäytyskuvan tapaan kaksiulotteiseksi rautalankakuvaksi (kuva 3).



Kuva 3. UV-kartoitus havainnollistettuna.

2.6 Riggaaminen

Jotta 3D-pelihahmoa voidaan animoida, täytyy sille ensin luoda virtuaaliset ”luut”. Hahmon 3D-mallin verteksit kiinnitetään näihin luihin ja kun luita liikutellaan, hahmo liikkuu niiden mukana. Tätä vaihetta nimitetään skinnaukseksi (eng. Skinning). (Masters 2016.) Sen jälkeen hahmolle luodaan kontrollit apuobjektien avulla, jotka mahdollistavat animoinnin. Nämä kontrollit ovat kuin kahvoja, joista animoija liikuttelee hahmon luita (kuva 4). Luut myös linkitetään tietynlaiseen hierar-

kiaan keskenään. Tätä vaihetta nimitetään riggaukseksi (eng. Rigging). Luut ja apuobjektit ovat vain 3D-malleja joiden avulla hahmoja liikutellaan.



Kuva 4. Lantion alueen luut liikkuvat apuobjektien (keltaisten nuolien) mukana.

3 Ohjelmistot

3D-mallinnuksessa käytettäviä ohjelmia on monia, mutta vain muutamia, jotka tarjoavat tarpeeksi monipuoliset työkalut ammattilaiskäyttöön. Monet pienemmistä ohjelmista taas ovat erikoistuneet tiettyyn tarkoitukseen, kuten tekstuurikarttojen luomiseen. Suuremmat ja monipuolisemmat ohjelmat ovat yleensä maksullisia ja päivittyvät vuosittain paremmiksi.

3D Studio MAX –ohjelma on erittäin monipuolinen ja laaja mallinnus-, visualisointi-, ja animointiohjelma. Sen vuoksi ohjelman käyttäjäkunta on hyvin laaja käsittäen eri toimialojen suunnittelijat, muotoilijat, pelien tekijät, graafikot, lavastajat jne. (Illikainen 2002, 5.)

Blender on ainoa ilmainen ammattilaiskäytössä oleva 3D-mallinnus- ja animaatio-ohjelma. Aiemmin Blender oli vaikeakäyttöinen, koska se vaati ohjelman omien pikakomentojen osaamista, jotta mallintaja pystyi toteuttamaan halutut toiminnot. Palautteen perusteella Blender on muokannut ulkoasuaan käyttäjäystävällisemmäksi. (Savander 2014,12.)

Vuonna 2014 suosituimmat digitaalisen veistämisen ohjelmat ovat ZBrush ja Mudbox. Useat elokuvat ja pelit ovat käyttäneet näitä ohjelmistoja, koska niiden avulla voidaan nopeuttaa sekä luoda entistä kehittyneempää taidetta elokuvaan ja peleihin. (Lääveri 2014, 3.) ZBrush sekä Mudbox ovat ohjelmia jotka ovat tarkoitettu digitaaliseen veistämiseen ja ne yhdistävät 3D- ja 2.5D-mallinnusta, sekä teksturointia (Lääveri 2014, 4). Ne eroavat muista perinteisistä mallinnusohjelmista siten että ne ovat painottuneet enemmän digitaaliseen veistämiseen.

xNormal on ohjelma, jota käytetään laajasti pelialalla high-poly-mallien yksityiskohtien projektointiin. Ohjelmalla on tarkoitus saada yksityiskohdat siirrettyä lopulliseen pelissä olevaan low-poly-malliin, siten että se näyttää samalta kuin high-poly-malli. Ohjelma on täysin ilmainen. (Prieto 2014.)

Adobe Photoshop on puhtaaseen teksturointiin käytetyistä ohjelmista ylivoimaisesti paras ja yleisin vaihtoehto pelialalla. Photoshop tarjoaa monipuoliset työkalut kuvankäsittelyyn ja piirtämiseen, joita tarvitaan teksturoinnissa.

4 Hahmomallintaminen nykypäivänä

Peliala on kovassa kasvussa ja alalla käytettävät työprosessit kehittyvät jatkuvasti. Ohjelmistoja tulee jatkuvasti lisää ja niitä kehitetään niin että ne soveltuisivat yhä paremmin artistien tarpeisiin. Esimerkiksi 3D-mallintaminen keskittyy enem-

män kuin koskaan aikaisemmin realismiin pyrkiviin digitaalisen veistämisen tekniikoihin. Teksturointi monesti vaatii maalaajan kädentaitoa sillä tekstuurit voidaan maalata hahmon pinnalle suoraan. (Burns 2015, 4.) Nykyaikaiset työkalut antavat vapaammat kädet luovuudelle eivätkä vaadi tekijältä niin paljoa niiden teknistä tuntemusta kuin ennen.

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. (Heikkinen 2010, 7.) Hahmomallintamisen työprosessit voivat vaihdella paljon riippuen tekijästä. Jokaisen vaiheen tekemiseen on lukuisia eri vaihtoehtoja ja ohjelmia. Sillä, minkä reitin tekijä valitsee, ei ole juurikaan merkitystä kunhan lopputulos on hyvä eikä työhön kulu kohtuuttoman paljon aikaa. Kaupallisissa tuotannoissa on myös otettava huomioon kustannustehokkuus, mikä voi vaikuttaa asioihin. Pelien grafiikka, ja sitä myöten myös työkalut, kehittyvät vauhdilla, joten on pysyttävä ajan hermolla. 3D-mallintajan omat opitut tavat ja ohjelmat voivat olla vanhanaikaisia ja tämä voi hidastaa työskentelyä merkittävästi

Pelihahmot ovat myös usein vaativampia ja työlämpiä luoda kuin ympäristön esineet, ja tämä on hyvä ottaa huomioon jo suunnittelussa. Hahmot ovat tärkeässä roolissa peleissä sillä pelaajat kiinnittävät heihin luonnollisesti enemmän huomiota kuin mitä ympärillä olevaan pelimaailmaan. Tästä johtuen, hahmojen ulkonäön eteen nähdään enemmän aikaa ja vaivaa kuin mitä esineiden ulkonäköön. Pelin päähenkilölle, pelistä riippuen, sallitaan suurempi polygonimäärä kuin sivuhenkilöille (Heikkinen 2010, 7.) Realistista lopputulosta tavoiteltaessa on ihmisahmon myös näytettävä ihmiseltä. Jokainen pelaaja huomaa helposti, jos pelissä esiintyvä hahmo ei näytäkään luonnolliselta. Huolellinen suunnittelu on tärkeää, jotta ongelmilta säästyttäisiin myöhemmin. 3D-mallintamisessa työskentelyprosessi on hyvin lineaarinen, joten mahdollisten virheiden korjaaminen myö-

hemmin voi koitua todella työlääksi. Esimerkiksi jo valmiiksi rigatun hahmon 3D-mallin rakennetta ei voi enää muokata, tuhoamatta rigiä.

Hahmoja mallintaessa täytyy ottaa huomioon, että hahmot tulevat liikkumaan, ja se vaikuttaa myös siihen, kuinka hahmon lopullinen topologia luodaan. Ihmisiä ja eläimiä mallintaessa on myös tunnettava jonkin verran anatomiaa, jotta lopputuloksesta tulee visuaalisesti uskottavan näköinen.

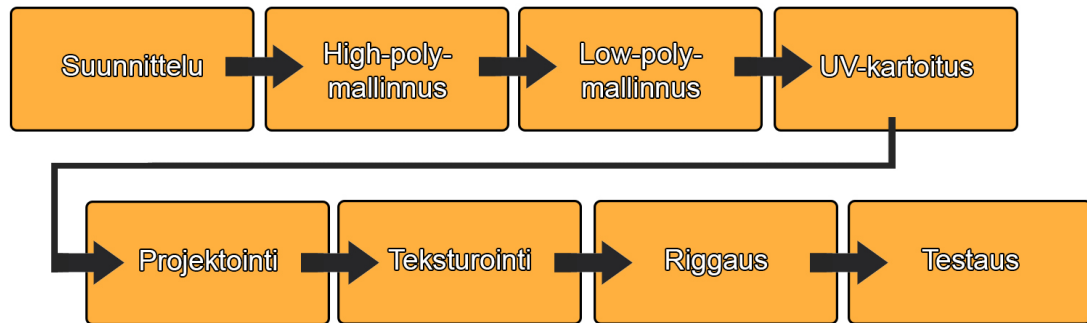
5 Pelihahmojen toteutus

5.1 Työprosessin suunnittelu

Oma prosessini perustuu osaltaan vuosien kokemukseen erilaisten objektien mallintamisesta ja teksturoinnista. Hahmomallintaminen on minulle kuitenkin uutta ja se eroaa monelta osin esimerkiksi esineiden ja ajoneuvojen mallintamisesta, joista minulla on enemmän kokemusta. Pelihahmot ovat myös usein vaativampia ja työläämpiä kuin ympäristön esineet ja tämä on hyvä ottaa huomioon jo suunnittelussa. Hahmot ovat tärkeässä roolissa peleissä, sen vuoksi pelaajat kiinnittävät heihin luonnollisesti enemmän huomiota kuin ympärillä olevaan maailmaan. Tästä johtuen hahmojen ulkonäköön käytetään enemmän aikaa kuin mitä esineiden ulkonäköön.

Ennen työskentelyn aloittamista laadin karkean suunnitelman, kuinka tulen toteuttamaan hahmot (kuva 5). Aluksi on konseptisuunnitteluvaihe, jossa kerään mallikuvia ja teen konseptipiirroksia. Sen jälkeen tulen mallintamaan hahmoista high-poly-versiot mallikuvien ja konseptipiirroksen mukaan. High-poly-malleja hyväksi käyttäen mallinnan hahmojen low-poly-versiot. Jotkin hahmojen pienemmät osat ovat niin pieniä, että ne eivät hyödy normaalikarttojen yksityiskohdista, joten näiden osien kohdalla pystyn jättämään high-poly-mallin tekemisen väliin. Kun

hahmon kaikki osat on mallinnettu, kartoitan mallien UV-koordinaatit, mitkä vaaditaan tekstuureja varten. UV-kartoituksen jälkeen, kasaan kaikki hahmojen eri osat yhteen ja laitan ne paikalleen. Jäljelle jää lopullinen peliin tuleva low-poly-malli. Tämän jälkeen projektoin high-poly-mallien yksityiskohdat normaalikarttoihin ja teen kaikki tarpeelliset tekstuuritiedostot ja lopuksi riggaan hahmon.



Kuva 5. Työvaiheet järjestyksessä.

5.2 Hahmojen konseptisuunnittelu

Yleensä hahmoja suunniteltaessa on niillä valmiina jo jonkinlainen taustatarina tai suunnitelma mitä hahmolta vaaditaan, ja se vaikuttaa myös hahmon ulkonäköön. Pelaajan avatar on usein haastavin suunnittelutehtävä pelintekijälle. Pelattavan hahmon fyysiset ominaisuudet määrittävät osaltaan pelaajan kiinnostusta ja immersiota peliin sekä yhteyttä pelihahmoon. (Pekola 2012, 11.) On siis tärkeää käyttää aikaa konseptointiin, jotta hahmosta saadaan visuaalisesti kiinnostavan näköinen. Peligraafikon näkökulmasta, konseptoinnissa tärkeintä on konseptipiirros tai konseptikuvat, minkä pohjalta pelimaailmoja, esineitä tai hahmoja aletaan työstämään.

Ennen kuin aloin suunnittelemaan hahmojen lopullista ulkonäköä, mietin minkälaiseen peliin tai pelimaailmaan ne sopisivat tai kuuluisivat. En kuitenkaan tässä

raportissa käy sen tarkemmin läpi hahmojen taustatarinoita, sillä tarkoitus on lähinnä perehtyä hahmojen tekniseen toteutukseen.

Hahmojen konseptipiirroksen voi toteuttaa haluamallaan tavalla. Voi piirtää itse, jos on tarpeeksi taitava tai sitten kerätä kuvamateriaalia netistä tai vaikkapa lehdistä ja kirjoista. Käytin hahmojeni konseptisuunnittelussa molempia tapoja. Mikäli hahmon konseptipiirroksen tekee itse (kuva 6), tulee sen olla tarpeeksi tarkkaan tehty, koska mallinnusvaiheessa jokaisen yksityiskohdan tulee olla hahmoteltu vähintäänkin mallintajan omassa päässä. Epäselvä konseptipiirros tai konseptikuvat voivat aiheuttaa ongelmia mallinnusvaiheessa ja voi pakottaa mallintajan tyytymään kompromisseihin, jotka huonontavat lopputulosta.



Kuva 6. Hahmottelemani konseptipiirros

mallintaa low-poly-versiota hahmoista. Eri UV-alueiden kierrättäminen ja tekstuurien peilaaminen on myös kätevää suunnitella tässä vaiheessa.

Character modeling 2 (Lanning, Baysal & Petroc 2007, 22) julkaisussa kerrotaan, kuinka Gears of War (gearsofwar.com) pelin hahmomallintajat jakavat tulevien low-poly-hahmojen eri osat jo konseptisuunnitteluvaiheessa. Konseptikuvan päälle voidaan piirtää karkea suunnitelma kuinka polygoniverkon rakenne tulee muodostumaan. Tämä helpottaa työprosessin suunnittelua ja itse mallintamisen aloittamista, kun hahmottaa mielessään paremmin työn kokonaisuuden. Tällä tavalla välttyy myös mahdollisilta tulevilta ongelmilta, kun

mallintaa low-poly-versiota hahmoista. Eri UV-alueiden kierrättäminen ja tekstuurien peilaaminen on myös kätevää suunnitella tässä vaiheessa.

5.3 Mallintaminen

5.3.1 Mallintamisen suunnittelu

Mallinnusprosessin voi aloittaa monella eri tavalla. Se riippuu tekijästä, minkä lähestymistavan hän haluaa ottaa ja erilaiset visuaaliset tyylit voivat vaikuttaa siihen minkä lähestymistavan valitsee. Onkin siis hyvä hallita useampia. Tavat voivat vaihdella siitä mihin tekijä on tottunut ja todennut toimivaksi.

Yksi yleinen tapa on luoda ensin low-poly-versio, joka toimii myös ns. pohjamallina (eng. base mesh), jota käytetään myöhemmin high-poly-mallin pohjana. Tämän tavan hyviä puolia ovat mm. alusta asti toimiva polygoniverkon rakenne ja helppous saavuttaa tavoiteltu polygonimäärä, sekä mahdollisten ongelmalueiden löytäminen ajoissa. Huono puoli on se, että näin joutuu alusta asti rakentamaan hahmon polygoniverkon rakenteen huolellisesti.

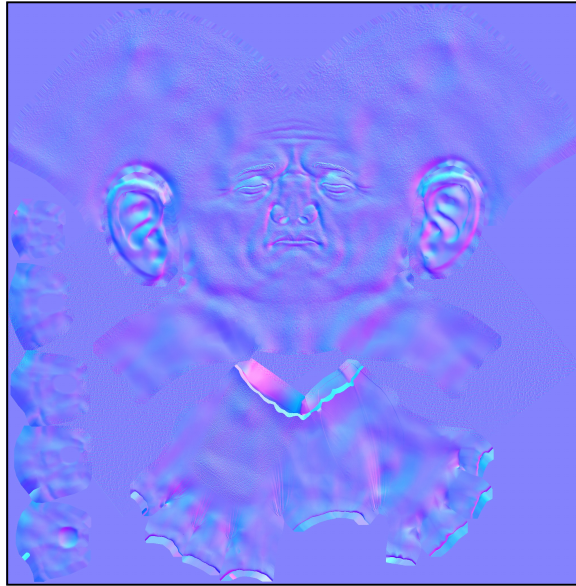
Toinen yleinen tapa on aloittaa suoraan high-poly-version luomisella ja tehdä low-poly-versio sen jälkeen käyttämällä high-poly-mallia pohjana. Tähän tarkoitukseen on kätevää käyttää retopology-työkaluja. Retopology-työkaluilla low-poly-mallin polygonit luodaan high-poly-mallin geometriaa myötäillen, mikä tekee low-poly-mallin luomisesta nopeaa. Tällä tavalla tekijän ei tarvitse miettiä lopullisen mallin polygoniverkon rakennetta alusta alkaen, vaan voi luoda sen vasta kun high-poly-malli on valmis. Hahmon high-poly-malli on kuitenkin se, miltä hahmo tulee näyttämään loppujen lopuksi, kun low-poly-malli on valmis ja normaalikartat ovat paikallaan. On siis helpompaa lopputuloksen onnistumisen kannalta mallintaa low-poly-malli high-poly-mallin päälle eikä päinvastoin. Tämä on nykyään yhä yleisempi tapa lähestyä hahmomallinnusta, sillä monet ohjelmat tarjoavat hyvät työkalut low-poly-mallin mallintamiseen suoraan high-poly-mallin päälle.

En asettanut hahmojeni low-poly-versioiden polygonimäärille minkäänlaista kattoa. Tarkoitukseni oli käyttää niin paljon polygoneja kuin oli tarvittavaa, jotta lopullisesta low-poly-mallista tulisi hyvän näköinen, ja voisin tarvittaessa karsia polygoneja pois. Polygonien määrä on harvemmin ongelmana suorituskyvyn kannalta, joten on varmempaa mallintaa hahmot polygoneja säästämättä. On tärkeää että mallissa ei pistä mikään erityisen pahasti silmään, vain siksi että ei ole käytetty tarpeeksi polygoneja, vaan että malli pyrkii esittämään anatomian mahdollisimman todenmukaisesti, eli pyöreät pinnat näyttävät pyöreiltä eikä kulmikkailta. Nykypäivänä PC- ja konsolipelihahmojen tapauksessa tämä on jo vähimmäisvaatimus.

5.3.2 Korkearesoluutioinen malli

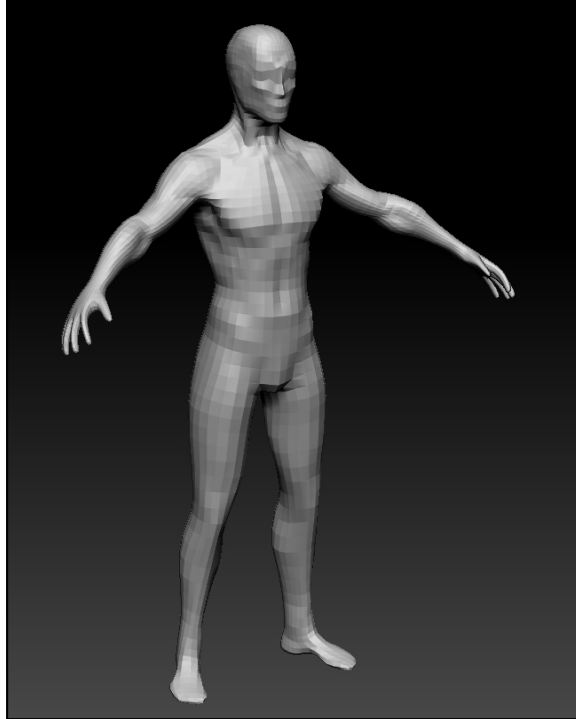
Ennen ja vielä tänäkin päivänä nykyaikaisten tietokoneiden laskentatehot rajoittavat ruudulla näkyvien polygonien määrää reaaliaikaisissa sovelluksissa kuten peleissä. Siksi yksityiskohdat on luotava tekstuurien avulla. Jotta realistinen peligrafiikka yltäisi visuaalisesti lähemmäs esirenderöityä grafiikkaa jota käytetään vaikkapa elokuvissa, käytetään apuna erilaisia tekstuurikarttatekniikoita. Jotta pelien käyttämä low-poly näyttäisi enemmän high-polyilta, käytetään apuna esimerkiksi normaalikarttoja.

Normaalikarttoja varten on hahmoista luotava korkearesoluutioinen malli eli ns. High-poly-malli, josta yksityiskohdat projektoidaan low-poly-malliin. Projektion tuloksena saadaan tekstuurikartta, jota kutsutaan normaalikartaksi (kuva 7).



Kuva 7. Hahmon pään ja käsien normaalikartta.

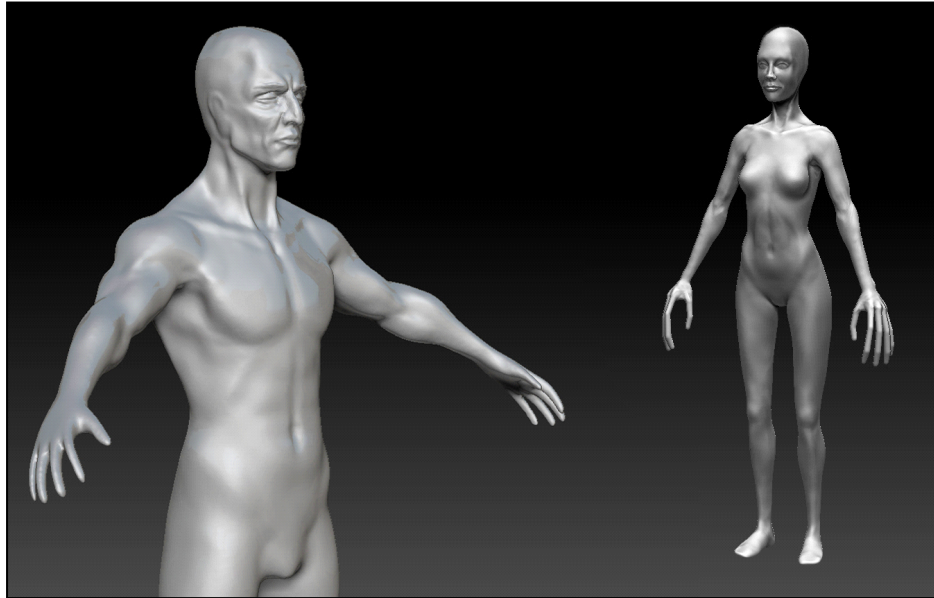
Aloitin molemmat hahmot luomalla ZBrushin ZSphere-työkalulla karkean ihmis-
muodon, josta aloin kaivertamaan ja venyttämään lisää geometriaa (kuva 8).
ZSphere on ZBrushin työkalu, joka on tarkoitettu karkean pohjamallin luomiseen.
Itse suosin tätä tapaa aloittaa mallintaminen, sillä työskentely on nopeaa ja vaatii
vähän minkäänlaista teknistä osaamista tai ohjelman tuntemusta. Työskentely
antaa myös hieman vapaammat kädet luovuudelle koska ei tarvitse juuri keskit-
tyä muuhun, kuin siihen että pohjamalli näyttää hyvältä. Tällä ZSphere-työkalulla
luodulla pohjamallilla, polygoniverkolla ei ole hyvä topologia, mutta sillä ei ole vä-
liä, koska tämä pohjamalli toimii tässä tapauksessa vain alkuna high-poly-mallille.
Huono topologia voi toki aiheuttaa rumia vääristymiä high-poly-mallissakin, jos
polygoniverkko ei ole tarpeeksi tiheä, mutta high-poly-malleilla on yleensä jopa
miljoonia polygoneja. Jos olisin luonut low-poly-mallin ensin ja sen pohjalta alkai-
sin työstämään high-poly-mallia, käyttäisin tätä low-poly-mallia pohjamallina
ZBrushissa. Silloin hahmon polygoniverkon topologialla olisi merkitystä, sillä se
tulisi olemaan lopullinen low-poly-malli, joka päättyy peliin.



Kuva 8. Karkea pohjamalli ZBrush-ohjelmassa.

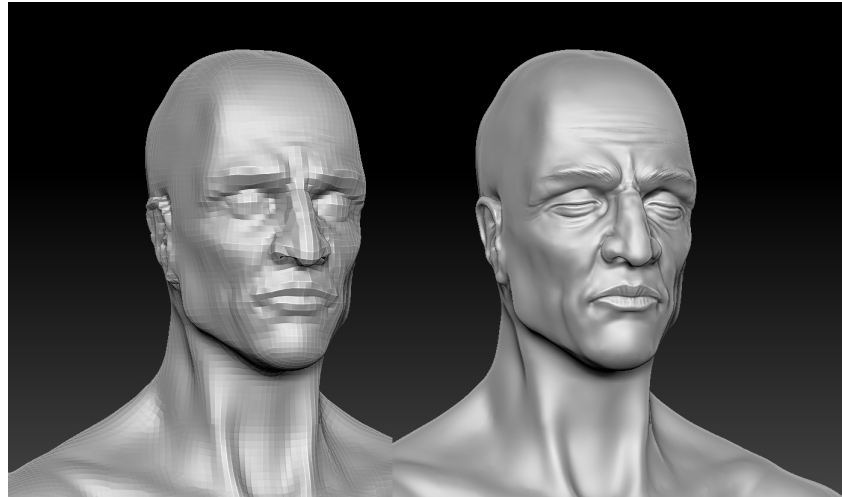
Konseptikuvaa apuna käyttäen muovasin pohjamallin sopivan muotoiseksi. Tarvittaessa käytin apunani ZBrushin dynamesh–työkalua. Hahmon geometriaa muokatessa polygoniverkko vääristyy, kun esimerkiksi hahmon käsiä venyttää, mutta dynamesh–työkalulla voidaan helposti korjata tämä. Työkalu luo valitun alueen polygoniverkoston uudelleen muuttamatta hahmon muotoa ja korjaa automaattisesti vääristyneen polygoniverkon. Dynamesh-työkalun muuttamaa polygoniverkon tiheyttä voi myös säätää, joten käytin sopivan pieniä arvoja, jotta pohjamalli olisi kevyt, mutta ei menettäisi kuitenkaan muotoaan.

Mallinsin molemmat hahmot pystyasentoon (kuva 9), kädet sivuille hieman koukussa, sillä molemmat hahmoni ovat pystyasennossa kulkevia ihmishahmoja. Hahmot on hyvä mallintaa alusta alkaen siihen asentoon missä ne tulevat lopulta olemaan enimmäkseen. Näin vältetään ylimääräiseltä geometrian venymiseltä ja tekstuurin vääristymiseltä kun hahmot animoituna liikuttavat raajojaan ja vartaloon. Vääristymiä tulee tietenkin, kun hahmot liikkuvat, mutta ne pyritään pitämään mahdollisimman huomaamattomina.



Kuva 9. High-poly-mallit alkuvaiheessa.

ZBrushissa hahmon polygoniverkon tiheyttä voi nostaa ja laskea työkalujen avulla. Verkon tiheyttä voi nostaa asteittain, jolloin jokaisella asteella polygonit jaetaan neljään osaan. Näin verkon tiheys kasvaa, jolloin geometriaa voidaan muokata yksityiskohtaisemmin. Suuremmalla tasolla, jolla polygonien määrä on suuri ja polygoniverkko on tiheämpää, voidaan tehdä todella pieniä yksityiskohtia, kuten ihon pintaa ja ryppyjä. Kun taas pienemmällä tasolla, jolla polygonien määrä on pieni, voidaan vaikuttaa isompiin muotoihin, kuten vartalon yleiseen muotoon ja lihaksiin. Aloitin työskentelyn pienimmällä tasolla ja kun olin tehnyt kaiken, nostin polygoniverkon tiheyttä ja siirryin seuraavalle tasolle. Etenin näin, kunnes olin saanut high-poly-mallit täysin valmiiksi. Alimman tason polygonien määrä oli n. 80 tuhatta polygonia ja ylimmän tason n. 5,1 miljoonaa polygonia (kuva 10).



Kuva 10. Polygoniverkon tiheystasojen vertailua.

Kun alastomat ihmismallit olivat valmiit aloin tekemään niille vaatteita. Vaatteet tein valitsemalla tietyn alueen hahmomallista ja kopioimalla sen omaksi erilliseksi objektiksi. Esimerkiksi tehdäkseen housut valitsin jalkojen alueen vyötäröstä alaspäin, kopioin sen ja aloin muokkaamaan sitä suurentamalla sen kokoa ja lisäämällä ryppyjä (kuva 11). Halusin tehdä vaatteet erillisinä kappaleina, jotta niitä olisi helpompi hallita myöhemmin tarvittaessa.



Kuva 11. Keskeneneräiset high-poly-mallit ZBrush -ohjelmassa.

Halusin lisätä toisen hahmon pään alueelle lisää jakoja polygoniverkkoon, jotta saisin muokattua kasvojen yksityiskohtia paremmin. Normaalisti lisäisin koko hahmon polygonien määrää nostamalla koko polygoniverkon tiheyttä, mutta käytössäni olevan työkoneen teho ei tähän riittänyt, joten jouduin rajaamaan kasvojen alueen jonka polygoniverkon tiheyttä kasvatin. Tästä voi muodostua ongelma, sillä kasvojen alueen tiheämmän verkon ja muun alueen harvemman verkon raja-alue voi aiheuttaa vääristymiä, jotka saattavat näkyä valon osuessa oikeassa kulmassa mallin pintaan. Mutta koska kyseessä on vain malli, jota käytetään normaalikarttojen luomiseen, olin luottavainen, että pystyisin helposti korjaamaan nämä mahdolliset ongelmat myöhemmin, kun muokkaisin normaalikarttoja Photoshopissa.

Pienemmät yksityiskohdat kuten ihohuokokset tein Photoshopissa käyttämällä apuna kuvia ihmisen ihosta. Arno Schmitz (2014, 73) kertoo, kuinka kaikista pienimmät yksityiskohdat hahmoon voi tehdä suoraan tekstuurikarttaan eikä niitä tarvitse tehdä high-poly-malliin. Tämä oli minulle täydellinen tapa toteuttaa hahmojen pienimmät yksityiskohdat, sillä työkoneeni ei ollut tarpeeksi tehokas, jotta olisin pystynyt lisäämään polygonien määrää tarpeeksi pienimpiä yksityiskohtia varten. Tätä prosessia varten voi käyttää CrazyBump ohjelmaa tai xNormal pluginia Photoshopiin jota itse päädyin käyttämään.

Saatuani hahmojen vaatteet mallinnettua olivat high-poly-mallit valmiit (kuva 12). Laitoin toiselle hahmomallille pohjavärit valmiiksi, sillä päätin, että kokeilen myöhemmin ennen teksturoinnin aloitusta ZBrushin polypaint-työkaluja, joilla maalaan värejä hahmon pinnalle.



Kuva 12. High-poly-mallit valmiina ZBrush –ohjelmassa.

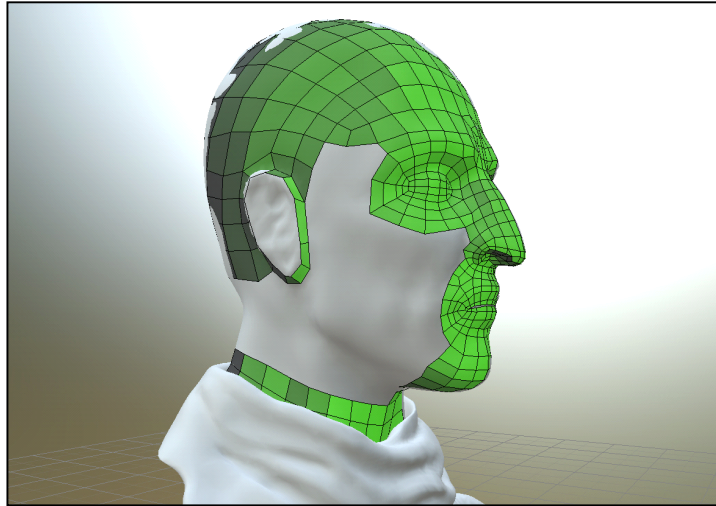
Pelkästään konseptipiirroksen perusteella mallintaminen oli hankalaa paikoitellen ja silloin mallikuvat sekä anatomiakirjat auttoivat paljon. Huomasin tekemäni konseptipiirroksen olevan erittäin hyödyllinen varsinkin mallinnuksen alkuvaiheessa, mutta tarkemmat yksityiskohdat tuottivat ongelmia sillä niitä ei piirroksessa ollut. Tässä taas verkosta löydetyt referenssikuvat osoittautuivat paremmiksi.

5.3.3 Matalaresoluutioinen malli

Low-poly-malli, on lopullinen malli, joka tulee pelimoottoriin ja liikkuu reaaliaikaisena ruudulla. Low-poly-mallissa ei siis voi olla läheskään yhtä paljon polygoneja kuin high-poly-mallissa. Vaikka en asettanut hahmoilleni mitään tiettyä rajaa polygonien suhteen, niin optimoimisen vuoksi on syytä välttää käyttämästä turhia polygoneja siellä, missä niitä ei tarvita. Esimerkiksi alueet jotka eivät liiku, taivu tai käänny ovat oivallisia paikkoja karsia ylimääräiset polygonit pois. Jos polygoni ei myöskään vaikuta hahmon siluettiin tai sitä ei tarvita normaalikarttojen projektointiin, on se silloin turha (Valve Corporation 2012).

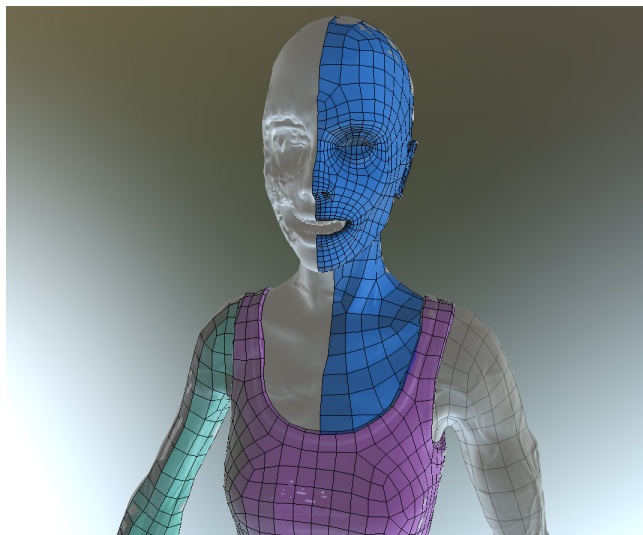
On myös syytä välttää käyttämästä kolmikulmaisia polygoneja jos vain mahdollista, koska ne voivat aiheuttaa ongelmia ja saa valon heijastukset näyttämään sotkuisilta, kuten neuvotaan Dota 2 Character Art Guidessa (Valve Corporation 2012). Peliä varten mallinnettaessa on hyvä tuntea tällaiset pelimoottorin rajoitukset, jotta ongelmilta vältyttäisiin.

Koska loin high-poly-version ensiksi, voin käyttää sitä pohjana low-poly-mallin luomiseen (kuva 13). Hyviä eri ohjelmia tähän tarkoitukseen on muutamia: 3Ds Max, Maya, Retopogun, 3D-Coat ja ZBrush. Itselleni sopivimmat retopology-työkalut olivat ZBrushin ja 3D-Coatin. ZBrushin työkalut tuttuine käyttöliittymineen olivat kätevät ja helppo omaksua, mutta pidin 3D-Coatista enemmän ja päädyin lopulta käyttämään 3D-Coatia low-poly-mallin tekemiseen. 3D-Coatin työkalut olivat helppoja ja käteviä käyttää, ja ohjelma oli vakaa. Koko prosessi hoitui muutamalla työkalulla ja pikanäppäimien opettelulla, mikä teki työskentelystä nopeaa. Pidin 3D-Coatista eniten myös siksi, että sillä oli monipuolisimmat työkalut tarjolla. ZBrushin retopology-työkalut taas tuntuivat kömpelömmiltä ja aiheuttivat jopa paljon ohjelman kaatuilua aivan perusasioiden kanssa, mikä oli turhauttavaa. Kaatuilun syy jäi lopulta epäselväksi. En kuitenkaan tämän kokemuksen perusteella välttelisi tulevaisuudessa ZBrushin retopology-työkaluja, sillä kyseessä saattoi olla vain väliaikainen ongelma työkoneessani tai ZBrushissa, joka korjaantuu päivitysten myötä. Jälleen kerran ohjelman valinta perustui lopulta siihen mikä ohjelmista suoriutui tehtävästä ongelmitta ja minkä työkalut tuntuivat minulle sopivilta. Uusia ohjelmia tulee myös joka vuosi lisää ja vanhat päivittyvät vastaamaan paremmin nykypäivän vaatimuksia. 3D-Coat kuitenkin suoriutui tällä kertaa tehtävästään pienimmällä vaivalla.



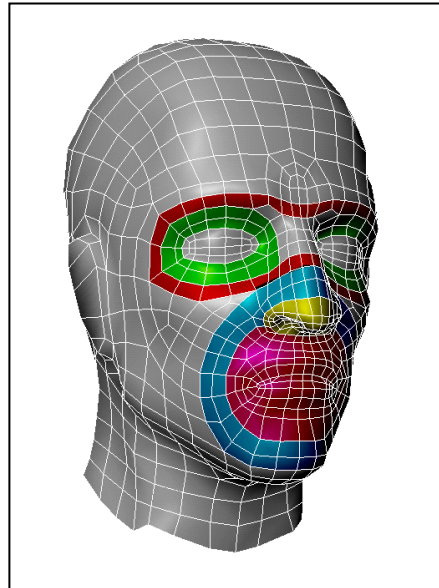
Kuva 13. Retopology-vaihe 3D-Coatissa.

Low-poly-mallia mallintaessa yritin ottaa huomioon geometrian taipumisen, kun hahmo liikkuu ja näin ennakoida mahdolliset ongelmat. Käsi- ja jalkataipeet ovat selkeät paikat, joissa geometria tulee liikkumaan ja taipumaan hahmon liikkuesssa, joten lisäsin näille alueille ylimääräisiä polygoneja välttääkseni pahimmat epämuodostumat. 3D-Coatissa pystyi käyttämään peilaus -ominaisuutta siten, että oli riittävää, kun teki vain toisen puolen hahmosta ja toinen puoli peilaantui automaattisesti. Toisen hahmon kasvojen kanssa en kuitenkaan voinut käyttää tätä ominaisuutta, sillä ne olivat muodoltaan epäsymmetriset (kuva 14).



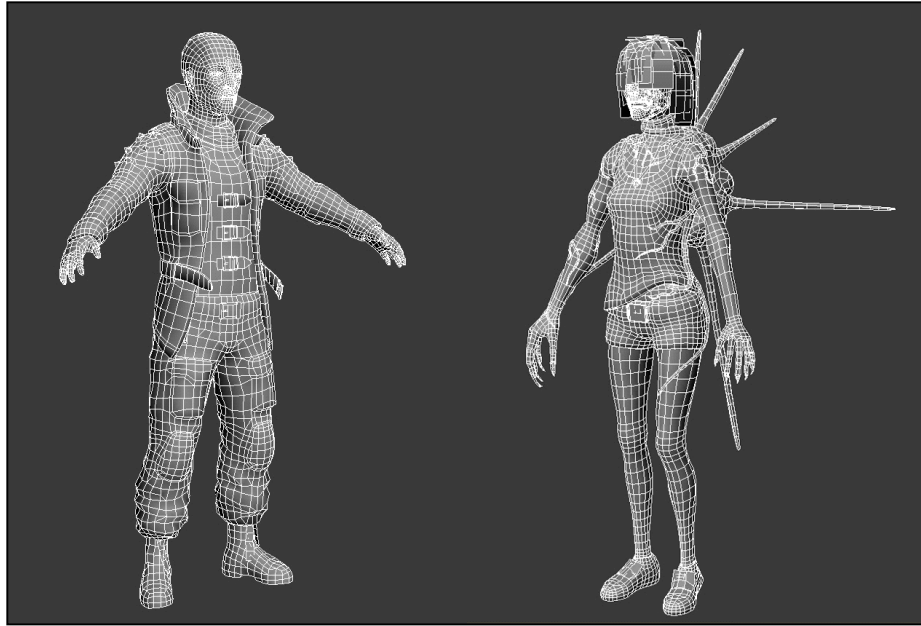
Kuva 14. Kasvojen alue 3D-Coatissa.

Kasvojen alueet vaativat erityisesti huomiota, sillä kasvot ovat täynnä liikkuvia lihaksia. Käytin apuna tekemääni referenssikuvaa (kuva 15), josta näin kuinka hahmon kasvojen polygoniverkon topologia olisi hyvä tehdä, jotta se myös animoituna näyttäisi hyvältä kun polygonit liikkuvat.



Kuva 15. Kasvojen topologiasuunnitelma.

Kun low-poly-mallit oli tehty valmiiksi vein hahmomallit 3DS maxiin (kuva 16). 3DS maxissa kävin hahmot läpi ja korjasin kaikki löytämäni virheet. 3D-Coatissa voi olla joskus vaikeaa huomata pieniä virheitä. Esimerkiksi jos kaksi verteksiä on päällekkäin. Nämä on helpompi hoitaa 3DS maxissa, sillä se tarjoaa paremmat työkalut tällaiseen työskentelyyn.



Kuva 16. Low-poly-mallit rautalankanäkymässä.

5.4 UV-kartoitus ja teksturointi

5.4.1 UV-Kartoitus

Tekstuurikartoilla saadaan hahmojen pinnalle kuvia, värejä ja yksityiskohtia. Tekstuurikartta on ikään kuin 3D-mallin ”ulkokuori”. Jokaisella 3D-mallilla on pelissä yleensä vähintään yksi tekstuurikartta. Tässä kuvatiedostossa on esim. hahmon ihon ja vaatteiden tekstuurit ja värit. Jos käytössä on vain yksi tekstuurikartta per malli, voi tekstuurikartassa olla myös staattisia varjoja ja pinnan kiiltoja, joita voidaan luoda käsin piirtämällä tai renderöimällä mallinnusohjelmassa. 3DS Maxissa voi renderöidä tekstuuritiedoston, jossa on vain 3D-mallin varjot. Tämän voi vaikka kuvankäsittelyohjelmassa yhdistää hahmon väritekstuurikarttaan.

Tekstuurikarttojen luominen alkaa jakamalla 3D-malli eri osiin. Jotta pelimoottorit osaisivat renderöidä kaksiulotteiset kuvatiedostot oikein kolmiulotteisten hahmojen pinnalla, täytyy 3D-mallille luoda tekstuurikoordinaatit. Näiden koordinaattien avulla pelimoottori renderöidessään projektoi kuvatiedoston 3D-mallin pinnalle.

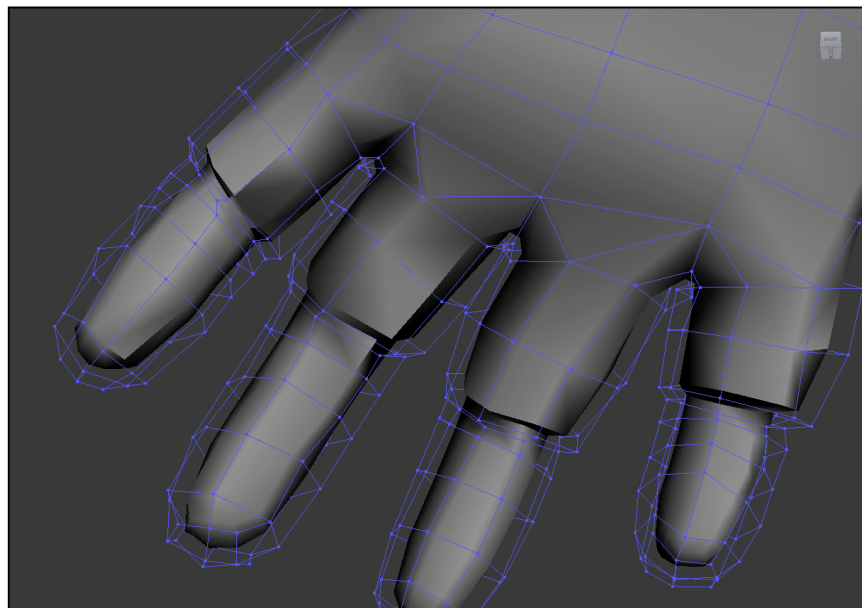
Tekstuurikoordinaattien tekoon löytyy työkalut lähes jokaisesta mallinnusohjelmasta ja on jopa erikseen itsenäisiä ohjelmia pelkästään tähän tarkoitukseen. 3D-mallin reunoja leikellään niin, että polygoneista muodostuu alueita, jotka voidaan tasata kaksiulotteisiksi (Korpilaakso 2010, 23). Esimerkiksi käsivarsi muodostaa oman alueensa, joka leikataan muusta mallista erilleen ja käsivartta pitkin tehdään leikkaus, jotta käsivarsi saadaan levitettyä räjäytyskuvan tavoin auki. Kun kaikki hahmon alueet on leikeltä näin tasaisiksi, täytyy ne asetella neliön muotoiseen UV-tilaan, ja näin jokaisella 3D-mallin verteksillä on nyt UV-tekstuurikoordinaattinsa. Tästä syystä joskus tekstuurikoordinaateista puhutaan myös UV-koordinaatteina, mikä tarkoittaa samaa asiaa.

Säästääkseni aikaa ja tekstuuritilaa tein kuten Heikkinen (2010, 26), joka kertoo opinnäytetyössään kierrättäneensä ja peilanneensa mahdollisimman suuren osan hahmon tekstuureista. Esimerkiksi käsien ja jalkojen UV-alueet laitoin päällekkäin, jotta minun ei tarvinnut teksturoida kuin esimerkiksi toinen oikeanpuolen käsi ja se sama tekstuuri näkyy myös vasemman puolen kädessä. Näin tehdyssä hahmossa täytyy ottaa huomioon, ettei pelaaja/katsoja huomaa, että molemmissa käsissä toistuu sama tekstuuri.

Pyrin antamaan enemmän UV-tilaa semmoisille alueille, joihin pelaaja kiinnittää enemmän huomiota, kuten kasvojen ja pään alueelle joissa on enemmän yksityiskohtia. Kun taas vähemmän semmoisille alueille, kuten kenkiin ja jalkoihin, joihin pelaajan katse harvemmin kiinnittyy. Yritin myös pitää UV-alueiden määrän pienenä ja kaikki osat yhtenä kappaleena, jotta teksturointi olisi helpompaa. Jos UV-alueet jakaa liian moneen osaan, voi teksturointi vaiheessa koitua ongelmaksi UV-alueiden saumakohtat ja tekstuurin jatkuvuus alueelta toiselle, ilman näkyvää saumaa.

5.4.2 Projektointi

Jotta high-poly-mallin yksityiskohdat saataisiin näkymään low-poly-mallissa, täytyy ne projektoida normaalikarttatekstuuriin. Projektointia varten täytyy luoda häkkimalli (eng. Cage model), joka määrittää, kuinka korkealta 3D-mallin pinnasta projektointi alkaa. Häkkimalli luodaan kopioimalla low-poly-malli ja skaalaamalla sitä isommaksi, niin että sekä low-poly- ja high-poly-mallit jäävät häkin sisäpuolelle, jotta projektointi onnistuu. Käytin projektointiin ilmaista ohjelmaa nimeltä xNormal. xNormal hoitaa projektoinnin hieman paremmin kuin vaikkapa 3DS Maxin sisään rakennettu oma projektointityökalu. Jälki on yksityiskohtaisempaa ja tarkempaa, sekä xNormalissa on paremmat säätömahdollisuudet. Häkkimallia tehdessä täytyy kiinnittää huomiota ahtaisiin paikkoihin, kuten sormien väliin (kuva 17), missä verteksit voivat mennä ristiin ja tämä aiheuttaa rumia jälkiä tekstuurikartoissa.



Kuva 17. Häkkimalli käden ympärillä.

Prieto (2014) neuvoo että 3D-mallit kannattaa jakaa osiin ja levittää irti toisistaan projektoinnin ajaksi tai jopa projektoida eri osat erikseen. Joidenkin hahmon osien kohdalla jouduin tekemään näin, mutta yritin välttää mallin jakamista osiin

koska jouduin myöhemmin yhdistämään näiden osien normaalikartat yhteen photoshopissa. Eri osien väliset reunat täytyi korjata photoshopin työkaluilla niin, että ne sulautuivat yhteen, eivätkä aiheuttaneet näkyviä saumoja.

Koska on olemassa niin monia eri 3D-mallinnusohjelmistoja, niin niiden skaalausarvot voivat vaihdella toisistaan. Tästä syystä xNormalissa on syytä muuttaa mallin kokoa isommaksi jos projektointi ei muuten onnistu kunnolla. Liian pienet 3D-mallit xNormalissa aiheuttavat epätarkkaa jälkeä (Prieto 2014.) Tuotuni hahmomallit 3DS maxista jouduin skaalaamaan niiden kokoa, jotta sain tehtyä toimivat normaalikartat projektoinnilla.

Projektoimalla high-poly-mallin yksityiskohtia xNormalilla voi myös renderöidä varjokartan. Tätä varjokarttaa nimitetään Ambient occlusion -kartaksi. Ambient occlusion luo pieniä varjoja semmoisille alueille minne valo ei yleensä pääse, kuten kainaloon ja haaroihin. Myös pienemmät pinnan muodot, kuten vaatteiden rypyt tai ihopoimut, luovat pieniä varjostuksia. Nämä ovat pieniä yksityiskohtia, joita olisi vaikeaa tehdä manuaalisesti, mutta ne vaikuttavat paljon siihen, miten hyvältä ja aidolta lopputulos näyttää. Ambient occlusion -karttaa käyttämällä voidaan halutessa muu valaistus jättää pois, mikäli ei haluta käyttää reaaliaikaista valaistusta. (Birn 2006.) Tämä voisi toimia hyvin vaikka mobiilipelien tapauksessa missä reaaliaikainen valaistus voidaan jättää pois optimoinnin vuoksi.

ZBrushissa voi myös maalata värejä mallin pinnalle käyttämällä polypaint-työkaluja. Projektoimalla voidaan myös tämä polypaintattu pinta siirtää kuvatieläködöön. Tämä toimi minulle hyvänä pohjana, jonka avulla pystyin myöhemmin teksturoimaan hahmon kokonaan. Nykyään peleissä alkaa näkyä yhä enemmän hahmoja, joiden tekstuurit on luotu taitavien artistien toimesta kokonaan maalamalla suoraan hahmon pinnalle, ZBrushissa tai muilla polypainttaukseen erikoistuneilla ohjelmilla.

5.4.3 Tekstuurikartat

Käytin hahmojen tekstuureina värikarttaa, normaalikarttaa, kiiltokarttaa sekä glossiness-karttaa. Näitä tekstuurikarttoja käyttämällä sain mielestäni hyvän ja laadukkaan lopputuloksen. Näiden lisäksi käytin myös värikartan alpha-kanavaa läpinäkyvyyttä varten. Käytin läpinäkyvyyttä vain silmäripsien, hiusten, kengännauhojen ja kaulakorun kohdalla.

Päätin kokeilla tekemällä toisen hahmon tekstuurit kokonaan pelkästään käyttämällä kuvankäsittelyohjelmaa ja toisen hahmon kohdalla käytin ZBrushin polypaint-maalausta apuna. Maalasin toisen hahmon kasvot kokonaan ja muun varhain osittain valmiiksi. Käytin maalauksessa apunani Fullerton yliopiston digitaalisen taiteenalan (2016) julkaisemaa videota. Maalasin videon ohjeiden mukaan ensiksi kasvojen ihon alla olevat värit, joille veri ja rasva antavat oman sävynsä. Tämän päälle maalasin ihon väriä joka peitti alla olevat kirkkaammat värit (kuva 18). Myöhemmin työstäessäni tekstuureja totesin, että käsin maalatut kasvot antoivat aidonmukaisemman lopputuloksen ja hyvän lähtökohdan alkaa teksturoimaan kasvoja Photoshopissa.



Kuva 18. Teksturointia polypaint-työkaluilla.

Tein kaiken lopullisen tekstuurien käsittelyn Photoshopissa. Aloitin renderöimällä 3DS Maxissa UV-kartasta kuvan, jotta näkisin UV-alueiden reunat teksturoidessa. Käytin UV-karttaa pohjana, jonka päälle aloin kasaamaan uusia tasoja. Käytin alkuun apuna generoituja karttoja, kuten polypaintattua värikarttaa, normaalikarttaa ja ambient occlusion -karttaa, joista näin yksityiskohdat, joiden päälle alkaa teksturoimaan. Itse ambient occlusion -karttaa käytin pienellä läpinäkyvyys arvolla, jotta sen vaikutus olisi pieni, sillä varjot syntyvät lopulta pääosin pelimoottorissa reaaliajassa. Pohjatekstuurien avulla aloin ensimmäisenä tekemään värikarttaa (kuva 19), koska kaikkien muiden tekstuurikarttojen toimivuus riippuu värikartan laadusta, sillä ne vain korostavat värikartan informaatiota. Värikartta on siis kaikista tärkein tekstuurikartta. Valmiin värikartan pohjalta on myös helppoa luoda kiilto- ja glossiness-kartat myöhemmin.



Kuva 19. Toisen hahmon pään ja keskivartalon alueen värikartta.

Hain internetistä paljon kuvia, joita käytin teksturoinnissa. Textures.com -kaltaiset sivut ovat loistavia tekstuurien lähteitä, mutta hyviä käytettäviä kuvia löytää myös hakupalveluiden kuvahaun avulla. Käytin teksturoinnissa lähinnä Textures.com -sivun kuvia, koska sieltä löytyi lähes kaikki tarvittava tekstuurimateriaali, jota hahmojeni tapauksessa tarvitsin. Kuvahauilla löydettyjä kuvia käytin lähinnä

maskeina tai tuomaan pieniä yksityiskohtia tai värivivahteita tekstuurin pinnalle. Esimerkiksi veritahrojen kohdalla hain kuvahaulla veritahrojen kuvia, joita käytin maskina ja tämän maskin avulla maalasin tahrat käsin tekstuuriin. Käytin myös apuna paljon eri tasojen sekoitustiloja (eng. Blending mode) joiden avulla lisäsin tekstuureihin likaa ja likatahroja. Likatahrat tulevat erityisen hyvin esille kiilto- ja glossiness-karttojen avulla, koska lika estää valoa heijastumasta takaisin ja tekee näin liat näkyvämmäksi valossa. Tällä tavalla pelkässä värikartassa hädin tuskin näkyvät pienet tahrat tulevat oikeassa valossa näkyviin paremmin, kuten ne tekisivät oikeassakin elämässä.

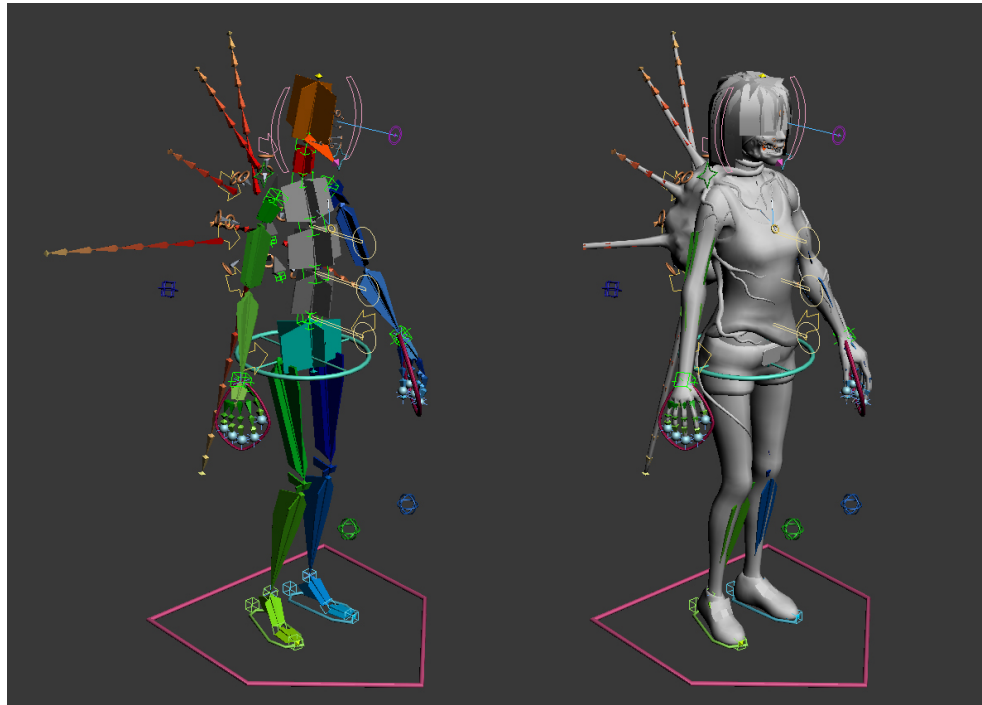
5.5 Hahmon riggaus

Liikkuakseen jokainen 3D-malli ja hahmo tarvitsevat virtuaaliset ”luut”. Nämä luut ovat vain 3D-malleja, joiden avulla hahmoa ja sen raajoja liikutellaan. Ilman luita hahmoa ei voisi animoida. Valmiina ihmishahmolle luotu luusto näyttää karkeasti ihmisluurangolta.

3D-mallin kiinnittäminen luihin tapahtuu verteksipisteiden kautta. Jokaiselle verteksille määritellään, mihin luuhun se on kiinnitetty ja kuinka paljon tuo luu vaikuttaa verteksiin. Tätä vaikutusta voidaan kutsua painoksi ja jokaisella luulla on omat painoarvonsa suhteessa verteksiin. Kaksi tai useampi luu voi vaikuttaa samaan verteksiin yhtä aikaa.

Animoiminen käyttämällä pelkkiä luita voi olla todella työlästä ja epäkäytännöllistä, riippuen rigin monimutkaisuudesta. Siksi yleensä luille luodaan apuobjekteja, jotka selkeyttävät ja helpottavat animoimisprosessia. Luut ovat usein 3D-mallin sisällä, mikä vaikeuttaa niiden animoimista. Apuobjektit taas voivat olla mallin ulkopuolella ja ne linkitetään luiden kanssa niin, että apuobjekteja liikutellessa luut liikkuvat niiden mukana.

Toista hahmoa varten loin apuobjektien avulla mahdollisimman monipuolisen ja helppokäyttöisen rigin (kuva 20) Delan Athiasin (2010) hahmoriggaus tutoriaalin avulla. Toisen hahmon kohdalla taas käytin apunani verkon kautta toimivaa Adoben Mixamo systeemiä (www.mixamo.com). Mixamoon tarvitsee vain ladata oma 3D-hahmo FBX -tiedostomuodossa ja määritellä muutamia kiintopisteitä, kuten polvet, kyynärpäät, leuka, jne. Näiden kiintopisteiden avulla mixamo generoi ihmishahmolle tarkoitetun rigin ja automaattisesti linkittää verteksit luihin sekä määrittää niiden painoarvot. Mixamo tekee hyvää jälkeä ja pienten korjauksien jälkeen rigistä tuli kunnollinen (kuva 21). Kun lisäsin tähän vielä muutaman apuobjektin, niin huomasin riggaamisen olevan paljon nopeampaa Mixamon avulla, sillä se antaa niin hyvän alun, josta lähteä työstämään eteenpäin.



Kuva 20. Monipuolinen ja helppokäyttöinen rigi.



Kuva 21. Yksinkertainen Rigi.

5.6 Hahmo pelimoottorissa

5.6.1 Importtaus

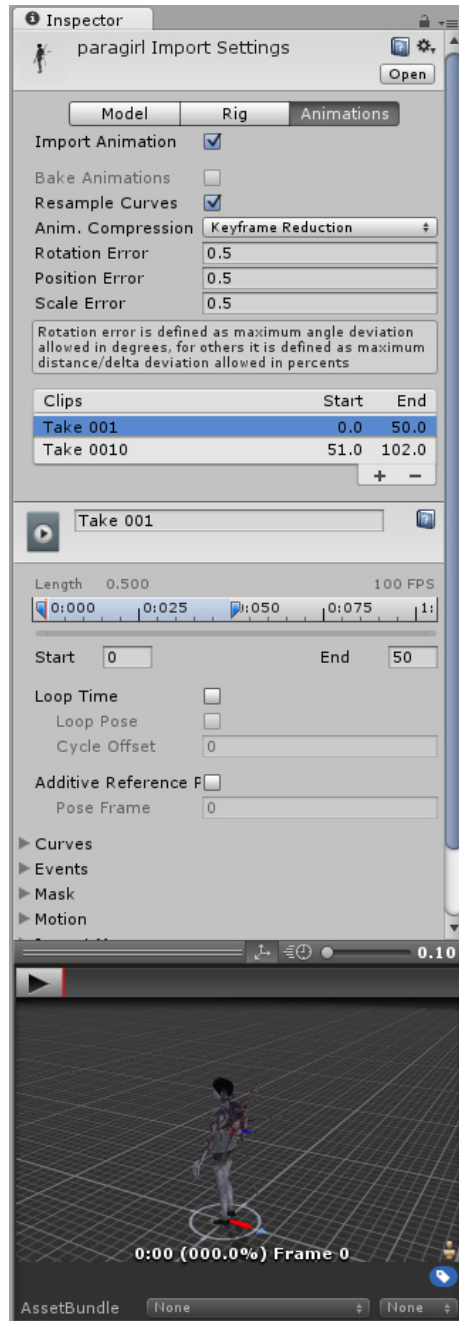
Nähdäkseen että kaikki toimii kuten pitääkin täytyy hahmo viedä lopulta pelimoottoriin. Pelimoottoreita on lukuisia erilaisia, mutta on vain muutamia ammattilaiskäytössä olevia, joita jokainen voi ilmaiseksi kokeilla. Yksi tällä hetkellä suosituimmista on Unity-pelimoottori. Unity tarjoaa mahdollisuudet kehittää niin 2D- kuin 3D-grafiikkaa käyttäviä pelejä. Sen suosio perustuu lähinnä sen monipuolisuuteen ja helppokäyttöisyyteen. Hahmon testaus siis onnistuisi muillakin pelimoottoreilla yhtä hyvin. Monesti eroja on lähinnä vain käytettävyydessä.

Exporttasin hahmot ja rigin Autodesk FBX-tiedostomuodossa 3DS maxista, sillä se on yleisin 3D-mallien käyttämä tiedostomuoto ja myös Unity tukee sitä. Toinen yleinen tiedostomuoto on OBJ, mutta FBX on suositumpi pelialalla sillä se pystyy sisältämään animaatioinformaatiota (Mayden 2016). FBX-tiedostojen exporttaus asetukset ovat oletuksena yleensä jo hyvät. Asetuksissa valitaan mitä eri asioita 3DS maxista tuodaan ulos, kuten esim. valoja, kameroita tai animaatioita. Ehkä tärkein asetukset mikä on hyvä ottaa huomioon on 3D-akselien (X, Y, Z) suunnat, sillä ne voivat erota siitä, mitä pelimoottori käyttää.

Kaikki lopulliset tekstuuritiedostot tallensin TGA-tiedostomuotoon, sillä se tukee alphakanavia ja on yleisin käytetty kuvatiedostojen tiedostomuoto pelialalla, ja näin ollen toimii hyvin myös Unityssa. Toinen hyvä tiedostomuoto on PNG, joka on hyvin samankaltainen kuin TGA. Toisella tekemälläni hahmolla oli kaiken kaikkiaan kuusitoista tekstuuritiedostoa ja toisella kaksitoista.

Käytin Unityn 5.3.4f1 versiota hahmon testaukseen. Loin Unityssa kokonaan uuden projektin ja toin kaikki hahmoon liittyvät tiedostot projektikansioon. Pyrin pitämään koko projektin alusta alkaen järkevän kansiorakenteen, mikä osoittautui tässä vaiheessa hyväksi ratkaisuksi. Selkeä kansiorakenne helpotti tiedostojen viemistä Unityyn sekä helpotti työskentelyä siellä.

Hahmo täytyy "koota" kasaan Unityssa ennen kuin se on valmis. 3D-malleille ja tekstuureille on omat asetuksensa, jotka on hyvä käydä läpi. 3D-mallien asetukset ovat yleensä oletuksena jo toimivat. Ne koskevat mm. Mallin skaalausta, UV-koordinaattien kääntämistä tai mallin optimoimista, mikä voi tulla tarpeeseen peliprojektien loppuvaiheessa, kun peliä optimoidaan kohdelaitteistoille. Animaatioille määritellään lyhyitä pätkiä, joita käytetään pelissä (kuva 22). Esim. kävelyanimaatio voisi koostua ruuduista (eng. frame) 1-10, juoksuanimaatio ruuduista 11-30, jne. Animaatiot tehdään siis putkeen ja se mitä ruutuja käytetään pelissä määritellään Unityssa.



Kuva 22. Animaatioasetukset Unityssa

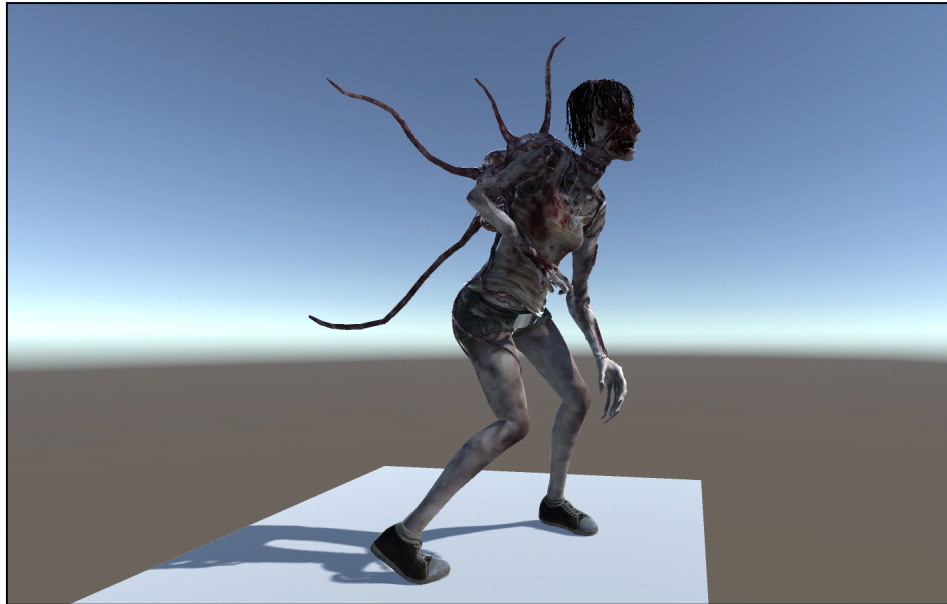
5.6.2 Materiaalit

Unity käyttää tekstuurien näyttämiseen "materiaaleja". Materiaalit ovat periaatteessa sama asia kuin varjostimet (eng. shader) 3D-grafiikassa. Ne määrittävät, kuinka valo käyttäytyy 3D-mallin pinnalla tai kuinka läpinäkyviä ne ovat. Yhdessä tekstuurikartat muodostavat materiaalin, jossa on värikartan värit, normaalkartan yksityiskohdat, sekä kiilto- ja glossiness-karttojen kiillot ja heijastukset, kaikki yhdessä. Yksi materiaali voi olla tarkoitettu pelkästään vaatteille, toinen päälle ja käsille (iholle), jne. Materiaaleilla on omat asetukset Unityssa, joiden avulla voi vielä hienosäätää lopputulosta.

Tekstuureilla ja materiaaleilla on omat asetukset Unityssa mitä pitää huomioida. Normaalkarttojen tekstuurityypiksi täytyy vaihtaa "Normal map", jotta Unity osaisi käsitellä sitä normaalikarttana ja muodostaa sen varjostukset oikein. Tekstuurien kompressoitua ja anisotrooppista suodatusta voi myös vaihtaa tarvittaessa. Nämä vaikuttavat siihen, kuinka hyvältä tekstuurit lopulta näyttävät ja kuinka paljon ne omalta osaltaan vaativat suorituskyvyltä.

Kun materiaalien asetukset on säädetty mieleiseksi ja ne on laitettu paikoilleen oikeisiin hahmon osiin, hahmo on valmis (kuva 23). Tein kokeilun vuoksi yksinkertaisen testianimaation, jonka totesin toimivan Unityssa. Jos kyseessä olisi oi-

kea pelituotanto niin tässä vaiheessa hahmolle tehtäisiin kunnolliset animaatiot ja ohjelmoija voisi käyttää Unityssa tehtyjä animaatiopätkiä pelihahmon liikkumisen luomiseksi.



Kuva 23. Valmis hahmo Unity-pelimootorissa.

6 Pohdinta

Opinnäytetyön tarkoituksena oli perehtyä nykyaikaisten PC- ja konsolipelien hahmojen luomiseen ja tutkia millaisia ovat ne työprosessit joilla saavutetaan kaupallisten pelihahmojen taso. Lopputuloksena oli kaksi hieman tyyliltään erilaista 3D-pelihahmoa, jotka ovat esitetty liitteissä 1 ja 3 valaistuksen kera. Liitteissä 2 ja 4 hahmot ovat ilman valaistusta, jotta tekstuurit näkyisivät selkeämmin ilman valojen tai varjojen muodostamia värieroja. Näitä hahmoja tehdessä pyrin jokaisessa työvaiheessa löytämään tietoa viimeisimmistä työskentelytavoista ja ohjelmistoista mitä alalla käytetään. Huomasin monesti ettei ole mitään tiettyä ta-

paa toteuttaa jotakin tiettyä vaihetta, vaan jokaisella hahmomallintajalla on omat ratkaisunsa ja työskentelytavat.

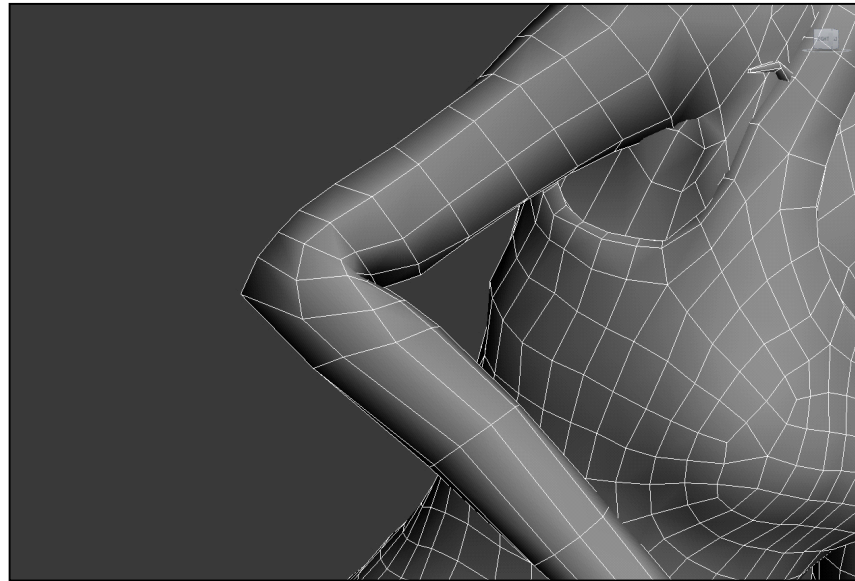
Molempien hahmojen mallintaminen oli lopulta paljon aikaa vievä projekti, sillä moniin ongelmiin ei löytynyt helppoja ratkaisuja vaan jouduin ratkaisemaan ongelmat itse. Ongelmiin saattoi löytyä ratkaisuja jotka vain sulki pois eri vaihtoehtoja ja jouduin itse päättämään loput. Vaikka ohjelmistot ovat muuttuneet käyttäjäystävällisemmiksi, niin ongelmatilanteissa vaaditaan yhä teknistä tietämystä. Silti yhä enemmän hahmomallintajilta vaaditaan nykyään enemmän visuaalista silmää ennen muuta.

Hahmojen suunnittelun olisin voinut tehdä huomattavasti huolellisemmin. Konseptipiirros osoittautui lopulta liian tylsäksi ja jouduin teksturointivaiheessa luomaan enemmän kontrastia hahmon väreihin. Toisen hahmon kohdalla keräämäni mallikuvat taas osoittautuivat alussa hankaliksi, sillä minulla ei ollut mitään selkeää suunnitelmaa mitä noudattaa. Olin hieman epävarma kuinka aloittaa hahmon mallinnus. Työelämälähtöisestä näkökulmasta katsottuna tämä olisi hyvin aikaa vievää.

Aikaisempi kokemukseni teknisestä 3D-mallintamisesta osoittautui monessa kohtaan riittämättömäksi, sillä hahmot ovat orgaanisia ja vaativat hyviä digitaalisen veistämisen taitoja. Hahmojen mittasuhteet ja anatomia olivat yksi asia, jota jouduin muuttelamaan paljon mallinnusvaiheessa, eivätkä ne olleet täydelliset lopullisessakaan versiossa. Huolellisempi perehtyminen ihmisanatomiaan ennen projektin alkua olisi säästänyt paljon päänvaivalta.

Yksi tärkeä pelihahmojen ja esineiden mallintamisen ero tulee siinä, että hahmoja mallintaessa täytyy ottaa huomioon se, että ne tulevat liikkumaan. On tärkeää tietää, kuinka hahmon osat tulevat taipumaan, jotta osaa varautua ja luoda mallintaessa polygoniverkon rakenne niin ettei se aiheuta animointivaiheessa ongelmia. Silloin niitä on vaikeaa käydä enää korjaamaan. Ensimmäisen hahmon kanssa

en kiinnittänyt tarpeeksi huomiota, kuinka paljon kädet voivat liikkua eri suuntiin ja käytin liian vähän polygoneja esimerkiksi kyynärpään alueella (kuva 24). Tuossa vaiheessa ei mallin polygonirakennetta voinut enää muuttaa jos halusi säilyttää hahmon rigin ehjänä. Osasin siis toista hahmoa mallintaessa lisätä ajoissa jakoja. Sain huomata myös, kuinka paljon polygoniverkon rakenne vaikuttaa siihen, miltä mikäkin alue näyttää liikkeessaan.



Kuva 24. Kyynärpään alueen polygonit.

Huono polygoniverkon rakenne voi saada myös tekstuurit näyttämään rumalta kun polygonit liikkuvat. Huomasin tekemistäni hahmoista monia tällaisia kohtia jotka aiheuttivat myöhemmin päänvaivaa. En tiedä kuinka olisin pystynyt välttämään näitä etukäteen muuten kuin tietämällä kokemuksen kautta. Monet muutkin kohtaamani ongelmat olivat tällaisia, jotka johtuivat kokemuksen puutteesta. Pelihahmojen kohdalla optimointi korostuu enemmän verrattaessa vaikkapa animaatiohahmoihin. Lopputulosta tarkastellessa huomasin että yritin turhan paljon optimoida hahmoa polygonimäärien kanssa. Olin tyytyväinen hahmojen kasvojen alueisiin, mutta käsissä olisin voinut käyttää reilusti enemmän polygoneja.

Hahmotan hahmomallintamisen prosessin kokonaisuuden nyt huomattavasti selkeämmin kuin mitä projektin alussa. Tämän toiminnallisen opinnäytetyön jälkeen olen käyttänyt lähes samaa työn kulkua myös työskennellessäni kaupallisen peliprojektin parissa ja todennut sen toimivaksi. Työelämässä Isoimpana erona vastaan on tullut physically based rendering (PBR) –tekstuuritekniikan käyttö, jota en tässä opinnäytetyössä käyttänyt. Mutta odotan PBR–tekstuuritekniikan yleistyvän enemmän konsoli ja PC-pelien kohdalla pelimoottorien ja eri renderöintitekniikoiden kehittymisen myötä, sillä se pyrkii entistä tarkempaan realismiin esittämiseen. Se on minulle selkeä opettelemisen aihe seuraavaksi. 3D- ja peligrafiikkaa kehittävät nopeasti ja se vaatii tekijöiltä jatkuvaa opiskelua ja oman osaamisen kehittämistä. Usein oppimisessa tärkeintä ei ole, kuinka tehdä asiat oikein, vaan oppia kuinka tehdä ne väärin. Näin osaa välttää noita vääriä tapoja ja soveltaa opittua tietoa uusissa tilanteissa.

Lähteet

- Athias, D. 2010. Introduction to Character Rigging in 3ds Max.
<http://www.digitaltutors.com/tutorial/336-Introduction-to-Character-Rigging-in-3ds-Max>. 6.11.2016.
- Autodesk. 2016. Glossiness map.
<http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/index.html?url=files/GUID-76F225CD-982A-4F17-9935-A656E6878BCD.htm,topicNumber=d30e392914>. 15.11.2016.
- Autodesk. 2016. Introduction to UV Mapping.
http://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/UV_mapping_overview_Introduction_to_UV_mapping.htm,topicNumber=d30e218945. 15.11.2016.
- Birn, J. 2006, Digital lighting and rendering, second edition. Berkeley, Yhdysvallat: New Riders.
- Blender Foundation. UV Mapping a Mesh.
https://www.blender.org/manual/editors/uv_image/uv_editing/unwrapping/introduction.html. 25.5.2016.
- Burns, J. 2015. Defining the Modeling Standard For 3D Character Artists. East Tennessee State University. Fine and performing arts. Opinnäytetyö.
<http://dc.etsu.edu/cgi/viewcontent.cgi?article=1307&context=honors>.

- 15.5.2016.
- Chopine, A. 2011. 3D Art Essentials the Fundamentals of 3D-modeling and Animation. Yhdistynyt Kuningaskunta: Elsevier Inc.
- Flavell, L. 2010. Beginning Blender: Open Source 3D Modeling, Animation and Game Design. Yhdysvallat: Apress Media LLC.
- Fullerton College Digital Arts. 2016. DART 153 Intro To ZBrush: Texturing Part 1 Colors And Materials.
<https://www.youtube.com/watch?v=tZwEKqgFuTU>. 5.11.2016.
- Heikkinen, S. 2010. Pelihahmon toteuttaminen, Metropolia Ammattikorkeakoulu. Viestinnän koulutusohjelma. Opinnäytetyö.
<http://urn.fi/URN:NBN:fi:amk-2010061112047>. 1.5.2016.
- Huhtamaa, L. 2007. Normal map –tekniikan käyttö 3D-mallinuksessa. Lahden Ammattikorkeakoulu. Mediatekniikan koulutusohjelma. Opinnäytetyö.
<http://www.theseus.fi/handle/10024/11552>. 5.5.2016.
- Illikainen, K. 2002. 3D Studio Max 4.0. Jyväskylä: Docendo Finland Oy.
- KatsBits. 2016. Make Better Textures for Games, 'Power of Two' & Proper Image Dimensions. <http://www.katsbits.com/tutorials/textures/make-better-textures-correct-size-and-power-of-two.php>. 6.11.2016.
- Korpilaakso, J. 2010. Hahmosuunnittelu Source Engine – pelimoottorille. Metropolia Ammattikorkeakoulu. Viestinnän koulutusohjelma.
<http://urn.fi/URN:NBN:fi:amk-2010061112038>. 9.5.2016.
- Lanning, K., Baysal, T. & Petroc, Z. 2007. D'artiste Character Modeling 2. Australia: Ballistic Publishing.
- Lääveri, I. Digitaalinen veistäminen: ZBrush-veisto-ohjelma pelitaiteen työkaluna. Lahden Ammattikorkeakoulu. Mediatekniikan koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-201405086693>. 15.8.2016.
- Mayden, A. 2016. CG File Formats You Need To Know – Understanding OBJ, FBX, Alembic and More. Pluralsight. <http://blog.digitaltutors.com/cg-file-formats-you-need-to-know-understanding-obj-fbx-alembic-and-more/>. 2.6.2016.
- Masters, M. 2016. Understanding Skinning – The Vital Step for Any Rigging Project. Pluralsight <http://blog.digitaltutors.com/understanding-skinning-vital-step-rigging-project/>. 26.5.2016.
- Pietiläinen, H. 2016. Fotorealismi Blenderissä ja Blend4Webissä: Karelia-amk:n CHP-voimalan 3D-visualistointi. Karelia-ammattikorkeakoulu. Viestinnän Koulutusohjelma. Opinnäytetyö.
<http://urn.fi/URN:NBN:fi:amk-2016100314771>. 6.11.2016.
- Pixologic. 2016. DynaMesh. <http://docs.pixologic.com/user-guide/3d-modeling/modeling-basics/creating-meshes/dynamesh/>. 6.11.2016.
- Pixologic. 2016. Polypaint. <http://docs.pixologic.com/user-guide/3d-modeling/painting-your-model/polypaint/>. 6.11.2016.
- Prieto, C. 2014. Ambient occlusion and Normal map bake using Xnormal. CRYTEK.
<http://docs.cryengine.com/display/SDKDOC2/Ambient+Occlusion+and+Normal+map+bake+using+Xnormal>. 7.11.2016.
- Puhakka, A. 2008. 3D-grafiikka. Helsinki: Talentum.
- Russel, E. 2014. Understanding the Difference Between Texture Maps. Plu-

- ralsight. <http://blog.digitaltutors.com/understanding-difference-texture-maps>. 25.5.2016.
- Savander, S. 3D-Hahmon Valmistaminen Pelimootoria Varten. Turun Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-2014061212692>. 10.8.2016.
- Slick, J. 2016. What is Rendering. About.com <https://www.lifewire.com/what-is-rendering-1954>. 13.9.2016.
- Theodore, S. 2016. What is retopology, and how do you do it with ZBrush. <https://www.quora.com/What-is-retopology-and-how-do-you-do-it-with-ZBrush>. 15.11.2016.
- Thompson, J. 2007. Game Design: Principles, Practice, and Techniques – The Ultimate Guide for the Aspiring Game Designer. Yhdistynyt Kuningaskunta: Quarto Publishing Inc.

Ensimmäinen hahmo valaistuna



Ensimmäinen hahmo ilman valaistusta tai varjoja



Toinen hahmo valaistuna



Toinen hahmo ilman valaistusta tai varjoja

