

Tuomas Hautamäki

3D-PELIHAHMON TUOTANTOLINJA

Työvaiheet konseptista pelimoottoriin

3D-PELIHAHMON TUOTANTOLINJA

Työvaiheet konseptista pelimoottoriin

Tuomas Hautamäki
Opinnäytetyö
Kevät 2019
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma, Internet-palvelut ja digitaalinen media

Tekijä(t): Tuomas Hautamäki
Opinnäytetyön nimi: 3D-pelihahmon tuotantolinja
Työn ohjaaja: Eero Leskinen
Työn valmistumislukukausi ja -vuosi: Kevät 2019 Sivumäärä: 73

Opinnäytetyö käsittelee 3D-hahmon tuotantoa konseptiasteelta aina pelimoottorissa käytettäväksi asti ja kuvaa prosessin eri vaiheet järjestyksessä. Esimerkkinä käytetään kuvatulla menetelmällä tuotettua hahmoa, joka vastaa nykyaikaista, realismiin tähtäävää pc - tai konsolipelihahmoa. Työkaluina on pelialan ammattilaisten käyttämiä ohjelmistoja, joiden käyttötarkoitusta ja suhdetta toisiinsa tarkastellaan työn puitteissa.

3D-hahmon mallinnus konseptista pelimoottorissa pelattavaksi hahmoksi on monimutkainen ja vaativa prosessi, jonka työvaiheista ei ole yhtä oikeaa standardia. Tässä työssä esitetyt menetelmät perustuvat alan ammattilaisten, käytettyjen ohjelmien kokeneiden käyttäjien ja luojien lausuntoihin. Tarkoituksena on koostaa toimiva tuotantolinja, jonka lopputuotteena on korkealuokkainen 3D-hahmo.

Työssä ei käydä läpi jokaista vaihetta hyvin yksityiskohtaisella tasolla, vaan tarkoitus on ensisijaisesti antaa jo hieman alaa tuntevalle tehokas katsaus prosessiin kokonaisuutena. Työhön on kuitenkin pyritty sisällyttämään tiettyjä, usein vaikeasti löydettäviä yksityiskohtia ja huomioita, sekä yleisesti hyviä työtapoja ja käytänteitä.

Hahmomallinnus on haastava, mutta erittäin palkitseva prosessi, ja 3D-alalla hyvin kysytty taito. Yksi tämän työn tavoitteista on olla hyvä referenssimateriaali aloittelevalle, ja kenties kokeneemmallekin hahmoartistille, sillä se pyrkii huomioimaan mahdollisimman monta osa-aluetta ja antaa suuntaa lisätiedonhauulle. Tämän ohella se kenties inspiroi taiteellisesti ja havainnollistaa mihin modernit ohjelmistot ja pelimoottorit teknisesti pystyvät.

Asiasanat:

3D-mallinnus, mallintaminen, pelihahmot, tuotantoketjut, peligrafiikka

ABSTRACT

Oulu University of Applied Sciences
Degree programme of business information systems, Internet services and digital media

Author(s): Tuomas Hautamäki
Title of thesis: 3D character model pipeline
Supervisor(s): Eero Leskinen
Term and year when the thesis was submitted: Spring 2019 Number of pages: 73

This thesis depicts a one possible pipeline option to be used in production to create a 3D character model from concept stage all the way to be used in game engine. As an example, an actual character model that matches modern, realistic pc- and console game standards, is made using this method. The software used here are common tools in the videogame industry and their function and relation to each other is examined here.

3D character modeling from concept to a finished, playable game asset is a rather complex and demanding process, which has no standardized workflow. The methods presented here are based on the statements of industry veterans and experienced users and creators of used software. The goal is to compile one functioning production pipeline and as an end-product, produce a high-quality 3D character.

The thesis does not depict every phase in a highly specific detail, rather the idea is to shed light on the process as a whole for someone with some prior knowledge and experience on the matter. It does however include small details and notes that are sometimes rather hard to come by otherwise, as well as workflows generally considered good practice.

Character modeling is a challenging but very rewarding process and highly sought-after skill in the 3D industry. One of the goals of this thesis is to be a good reference for aspiring, or even for a seasoned character artist, in that it seeks to cover as many aspects of the process as possible and points the way for additional research. On top of that, it inspires artistically perhaps and demonstrates what modern software and game engines are technically capable of.

Keywords:

3D modeling, modeling, game characters, production pipelines, game graphics

SISÄLLYS

1	JOHDANTO	7
2	HAHMO: IDEA JA TAVOITTEET	9
2.1	Idea.....	9
2.2	Tavoitteet.....	10
3	TERMINOLOGIA	15
4	OHJELMISTOT	18
4.1	Photoshop	18
4.2	3D-mallinnus	19
4.2.1	Maya.....	19
4.2.2	Blender	20
4.3	ZBrush	21
4.4	Topogun	22
4.5	xNormal	23
4.6	Substance Painter	24
4.7	Marvelous Designer	25
4.8	Marmoset Toolbag 3.....	26
4.9	Unreal Engine 4	27
5	KONSEPTITAIDE	29
6	3D-MALLINNUS	31
6.1	Lowpoly	31
6.2	Highpoly	33
6.3	Mallin iterointi	40
6.4	Retopologia	41
6.5	UV-kartat	43
6.6	Hiukset	45
7	MATERIAALIT JA TEKSTUROIINTI.....	48
7.1	Fyysisiin ominaisuuksiin perustuva renderointi	48
7.2	Tekstuurikartat ja niiden valmistus.....	48
7.3	Väri ID:t	49
7.4	Materiaalit.....	50
8	SIMULAATIOT	53

8.1	Vaatetuksen luonti	53
8.2	Fysiikkasimulaatio.....	53
9	HAHMON LUUSTON LUONTI.....	55
9.1	Manuaalinen luonti.....	55
9.2	Auto-Rig Pro.....	56
9.3	Painoarvojen muokkaus	56
9.4	Korjaavat muodot.....	57
9.5	Luiden kontrollit.....	59
9.6	Mixamo.....	60
10	ANIMAATIOT	61
11	REAALIAIKAINEN RENDEROINTI	62
12	PELIMOOTTORI.....	63
12.1	Hahmon tuonti pelimoottoriin	63
12.2	Alustus.....	64
12.3	Hahmon kontrollointi.....	66
12.4	Animaatioiden käyttö	66
13	POHDINTA	68
	LÄHTEET.....	69

1 JOHDANTO

Opinnäytetyö käsittelee 3D-hahmon luomisprosessia ja pyrkii luomaan ehjän kokonaiskuvan tuotantolinjasta, joka vastaa ammattimaista tuotantoa. 3D-hahmo, joka on käytettävissä niin peli- kuin animaatiotuotannoissakin, vaatii monivaiheisen valmistusprosessin. Teknologian kehittyessä tämä prosessi elää koko ajan ja tapoja on useita, jolloin hyviä ja huonoja käytänteitä on vaikea havaita. Siinä missä ei ole välttämättä oikeaa ja väärää tapaa tehdä asioita, työtä arvioidaan lopputuloksen perusteella. Siksi työssä on, mahdollisuuksien mukaan, pyritty pohjustamaan esitettyjä tapoja kokeneiden ammattilaisten lausunnoilla.

Tutkimusongelmana ja kehittämiskohteena oli sen myötä eri ohjeiden yhdistäminen ja koonti toimivaksi kokonaisuudeksi. Prosessin eri vaiheista on paljon ohjeita sinällään, mutta ne eivät ole läheskään aina yhteensopivia toistensa kanssa, jolloin kokonaiskuva tuotannosta jää hämärän peittoon. Prosessi vaatii ammattitasolla useita eri ohjelmistoja, monien tiedostomuotojen ymmärrystä ja yhteensopivuutta teknisellä tasolla sekä taiteellista silmää tekijältään. Tämän vuoksi tuotantolinja on yrityksissä jaettu yleensä useamman työntekijän vastuulle, jolloin jokainen osoittaa erityistä ymmärrystä ja ammattitaitoa omalle vastuualueelleen. Tämän vuoksi on helppo ymmärtää, miksi yleisessä jaossa olevat ohjeistukset ovat yleensä niin pienissä palasissa ja hajanaisia. Aloittelevan tekijän on tärkeä ymmärtää koko prosessi ja pienten palasten keskellä on usein vaikea hahmottaa kokonaisuutta.

Työn tarkoituksena ei siis ole opettaa lukijaa käyttämään jokaista ohjelmistoa mitä tuotantolinjaan kuuluu, vaan esitellä työnkulku, joka antaa hyvän yleiskuvan koko prosessista ja mitä vaiheita se sisältää. Työssä esitellään eri työvaiheita luomalla oma hahmo ja käyttämällä sitä käytännön esimerkkinä. Erityispiirteenä prosessissa on hahmon laatuvaatimukset, sillä hyvin yksinkertaisten hahmojen luontiin löytyy kattavia ohjeita, mutta nuo hahmot eivät täyttäisi useimpien pelistudioiden vaatimuksia. Työssä pyrin luomaan mahdollisimman fotorealistisen hahmon, joka vastaa esimerkiksi nykyaikaisten ns. AAA-pelien tasoa. Tämä tarkoittaa huomattavasti monimutkaisempaa prosessia, johon kuuluu mm. vaatetus simulaatioita, kasvoanimaatioita sekä hius- ja iho detaljeja.

Tavoitteena oli, että työ auttaisi erityisesti aloittelevia hahmoartisteja, ja toimisi eräänlaisena reseptinä hahmon luontiin säästäten lukijan aikaa pakkaamalla mahdollisimman paljon hyödyllistä ja hy-

vin jäseneltyä asiaa samoihin kansiin. Haasteena oli aineiston kerääminen hyvin monesta lähteestä, eri lähteiden uskottavuuden ja esitettyjen menetelmien toimivuuden arviointi. Myös vakiintuneen suomenkielisen terminologian puute luo omat haasteensa dokumentointiin. Sen lisäksi itse prosessin läpivienti ja varsinaisen hahmon luonti on haastavaa, mutta tarpeellista prosessin toimivuuden todistamisen kannalta. 3D-hahmon tuotantolinja ei siis usein ole yhden ihmisen vastuulla työelämässä, mutta prosessin ymmärrys on varmasti avuksi osastojen välisessä yhteistyössä, ja tämä tekee osaavasta työntekijästä entistä arvokkaamman osan kokonaisuutta.

2 HAHMO: IDEA JA TAVOITTEET

2.1 Idea

Hahmon luontiprosessi alkaa luonnollisesti ideasta. Millainen hahmo peliin halutaan. Jokaista ideaa tulee sitten arvioida tarkemmin ja tarkastella, mitä vaatimuksia sen tulee täyttää pelin tarinan, pelimekaniikan ja kohdealustan teknisten vaatimusten ja pelin kohdeyleisön kannalta. Esimerkiksi, jos pelin tarina sijoittuu toiseen maailmansotaan, on hahmo mitä luultavimmin ihminen, kun taas eksotisella ulkoavaruuden planeetalla voi nähdä mitä mielikuvituksellisimpia hahmoja. Onko pelin keskeinen pelimekaniikka tasolta toiselle hyppiminen vaiko ahtaiden tunneleiden tutkiminen? Ensin mainittuun soveltuu paremmin pitkä hahmo, kun taas toiseen kenties lyhyempi hahmo on luontaisempi valinta. Jos kohde pelialustana on tehokas pc tai pelikonsoli, on vaihtoehtona luoda näyttävämpi ja realistisempi 3D-hahmo, kun taas käsin piirretty 2D-hahmo voi soveltua paremmin mobiilipelialustoille. Värikäs, sarjakuvamainen hahmo voi toimia paremmin nuorelle yleisölle, kun taas hillitty, realistisempi versio saattaa olla enemmän aikuiseen makuun. Vaihtoehtoja on paljon ja siksi on tärkeä käydä läpi tärkeimmät kriteerit ja tehdä hahmosuunnittelua näihin tietoihin perustuen. Näistä tiedoista on myös hyvä ammentaa hahmon persoonallisuutta ja ilmentää sitä visuaalisesti suunnittelussa.

Tässä työssä esimerkkinä käytettävä hahmo on kuvitteellisen pelin päähenkilö ja suunnitelma perustuu samoihin yllämainittuihin parametreihin. Oletettu peli on 3D toimintaseikkailu pc:lle ja uusimmille konsoleille ja sen tarina sijoittuu lähitulevaisuuden kaupunkiin, science fiction -genressä. Hahmo on naispuolinen, suuryrityksen turvallisuuspäällikkö, joka on luonteeltaan säntillinen ja pikkutarkka introvertti. Näistä lähtökohdista voimme jo rakentaa hahmoa visuaalisesti: realistinen ihmishahmo, nainen, atleettinen, varusteltu, aseistettu, futuristinen, huoliteltu. Näillä hakusanoilla voidaan etsiä sopivia referenssejä internetin kuvapalveluista, kuten esimerkiksi Pinterestistä (www.pinterest.com), joka on oivallinen tähän tarkoitukseen. Referenssikuvien perusteella konseptiartistit voivat aloittaa hahmon kuvittamisen vakaalta pohjalta, jossa kuvakieli on jo selvillä.



KUVIO 1. Esimerkkejä referenssikuvista (Pinterest 2019, viitattu 19.3.2019).

2.2 Tavoitteet

Työn tavoitteena oli toteuttaa hahmo, joka vastaa nykyaikaista, korkealaatuista pc - ja konsolipelien tasoa. Hahmon teknisinä esikuvina toimivat esimerkiksi Lara Croft (Shadow of the Tomb Raider 2019, viitattu 25.3.2019) ja Cassandra (Assassin's Creed: Odyssey 2018, viitattu 25.3.2019). Näille, ja useille muille nykyaikaisille pelihahmoille on ominaista näyttää hyvinkin aidoilta niin ruumiinrakenteen, kuin materiaalien ja animaatioidenkin osalta. Hahmot liikkuvat realistisesti ja kykenevät näyttämään eri tunnetiloja elekielen ja hienovaraisten kasvoanimaatioiden avulla. Jokainen tekninen osa-alue on tärkeä hahmon uskottavuuden kannalta. Ihon tulee näyttää iholta ja hiusten hiuksilta ja fysiikan lakien tulee näkyä hahmon liikkeissä, muutoin kokonaisuus kärsii. Modernit, realistiset pelihahmot koostuvat jopa noin sadastatuhannesta polygonista, joka on verrattavissa kuvaresoluutioon, eli mitä enemmän polygoneja, sitä yksityiskohtaisempi hahmo, mutta vastaavasti se kuormittaa pelimoottorin laskentatehoa (Polycount 2014, viitattu 19.3.2019). Jotkin pelit käyttävät malleina hahmoina esiintyviä näyttelijöitä, jolloin pelihahmon laatua voi vertailla suoraan näyttelijän ulkonäköön ja olemukseen.

TAULUKKO 1. Otanta pelihahmojen polygon määrästä peleissä (Polycount 2014, viitattu 19.3.2019)

Peli	Hahmon tri-polygon määrä
Infamous: Second Son	120,000
The Order 1886	100,000
Final Fantasy XV	100,000
Street Fighter V	80,000



KUVIO 2. Lara Croft, Shadow of the Tomb Raider (Dan Roarty 2015, viitattu 24.3.2019).



KUVIO 3. Cassandra, Assassin's Creed: Odyssey (Sabin Lalancette 2018, viitattu 24.3.2019).



KUVIO 4. Valorie Curry pelissä Detroit: Become Human (Larryn Bell 2018, viitattu 19.5.2019).



KUVIO 5. Minka Kelly pelissä Detroit: Become Human (Larryn Bell 2018, viitattu 19.5.2019).

Työssä käydään läpi jokainen työvaihe ja prosessi, joka johtaa teknisesti valmiiseen hahmoon, mutta viimekädessä työn laatu ja lopputulos on kiinni artistin kyvyistä. Alla on kuvattuna työn lopputuote eli esimerkkinä käytetty, viimeistelty hahmo. Tämä antaa kuvan työn tasosta ja auttaa säästämään eri työvaiheiden tarkoituksen paremmin. Hahmo koostuu 91,507 tri-polygonista, joka on linjassa yllämainittujen esimerkkien kanssa.



KUVIO 6. Valmis, työssä kuvatulla tuotantolinjalla toteutettu hahmo.

3 TERMINOLOGIA

3D-ala on vielä nuori Suomessa, joten vakiintunutta termistöä ei vielä ole ja usein käytetään englanninkielisiä termejä, joita avataan alla. Pitäydyn tässä työssä käyttämään joko alkuperäisiä englanninkielisiä termejä tai alla esitettyjä käännöksiä, koska mahdollinen jatkotiedonhaku tuottaa parempia tuloksia niiden avulla.

Rendering: Renderointi tai renderöinti, tarkoittaa prosessia, jossa tietokone muodostaa kuvan 3D-ohjelmassa määritetystä sisällöstä. Tähän liittyy usein monimutkaisia geometria- ja valolaskentoja, jotka voivat kestää perinteisissä 3D-ohjelmissa hyvin kauan. Pelimoottorit ovat rakennettu alusta asti reaaliaikaista renderointia varten, joten hyvin optimoitu peli voidaan renderoida ruudulle jopa 60 kertaa sekunnissa.

Vertex: Piste 3D-tilassa. Kaikki 3D-mallit muodostuvat näistä pisteistä, esimerkiksi kuutiossa jokainen kulma on vertex-piste, kun taas monimutkaisempaan geometrian muodostamiseen tarvitaan enemmän pisteitä.

Line: Viiva tai reuna, joka yhdistää vertex-pisteet toisiinsa, muodostaen polygoneja.

Polygon: Polygoni on pinta, joka muodostuu, kun kolme tai useampi vertex-piste yhdistetään viivoilla toisiinsa. Hyvänä käytäntönä pidetään neljän pisteen muodostamaa quad-pintaa, mutta myös kolmen pisteen muodostama tri-pinta on toimiva. Sen sijaan viiden tai useamman pisteen muodostamia ngon-pintoja ei kannata käyttää, sillä ne aiheuttavat virheitä mallin pinnassa. 3D-mallit muodostuvat näistä pinnoista, jotka voidaan teksturoida ja valaista.

Plug-in: Ohjelmistoon asennettava lisäosa, joka tuo ohjelmaan jonkin uuden toiminnon tai ominaisuuden. Lisäosia on niin ilmaisia, kuin maksullisiakin ja usein niitä kehitetään myös ohjelmiston käyttäjien toimesta, silloin kuin se ohjelmiston omistajan osalta sallitaan.

Rigging: Rigaus on toimenpide, jossa 3D-mallille asetetaan ”luut”, eli hahmolle määritetään taivepisteet, joiden avulla hahmoa voidaan animoida.

Sculpting: Sculptaus, eli veistäminen, kuvaa perinteisestä "hard surface"-mallinnuksesta poikkeavaa 3D-mallinnustapaa, jossa 3D-objektia veistetään ja muovataan kuin savea.

Lowpoly: 3D-objekti, jonka polygoniluku tai "resoluutio" on alhainen. Käytetään myös tyylikeinona, kun tavoitellaan yksinkertaista, tyyliä ulkoasua esimerkiksi mobiilipelille.

Highpoly: Vastakohta lowpoly-mallille. Korkean polygoniluvun omaava 3D-malli, jonka ensisijainen tarkoitus on visuaalisuudessa. Joskus jopa kymmenistä miljoonista vertex-pisteistä koostuvaan hahmoon voidaan mallintaa pieniäkin yksityiskohtia, kuten esimerkiksi ihohuokosia. Koska highpoly-mallit ovat hyvin raskaita prosessorille, taltioidaan niiden yksityiskohdat kuvatiedostoiksi, joita käyttämällä lowpoly-mallin kanssa saadaan lopputulos, joka näyttää lähes samalta kuin highpoly-malli, mutta on huomattavasti kevyempi prosessorille ja soveltuu siten esimerkiksi pelimoottorissa käytettäväksi.

Texture: Tekstuuri tai tekstuurikartta on kuvatiedosto, jota hyödynnetään 3D-mallien pintamateriaaleissa. Esimerkiksi jos 3D-objektin halutaan näyttävän puupinnalta, tarvitaan siihen vähintään yksi kuva vastaavasta puupinnasta, joka asetetaan näkymään 3D-objektin pinnalla.

Shader: Shader on yksi vaikeimmin selitettävistä termeistä, mutta se kuvaa 3D-objektin materiaalia ja sen käyttäytymistä. Erilaiset shaderit hyödyntävät erilaisia tekstureja ja parametreja, jotka kuvaavat erilaisten pintojen käyttäytymistä valossa. Esimerkiksi iho reagoi valoon eri tavalla kuin metalli ja siksi ne tarvitsevat erilaisen shaderin. Yksinkertaisimmillaan shader on materiaali, mutta osaavissa käsissä se on paljon muutakin.

Topology: Topologia viittaa tapaan, jolla polygonit muodostavat mallin pinnan. Hyvänä topologiana pidetään selkeää, tasaisen kokoisista neliöistä koostuvaa pintaa, kun taas huono topologia on sekavaa, vaihtelevan kokoisia ngon-polygoneista muodostuva pinta.

Retopology: Retopologia on prosessin vaihe, jossa highpoly-malli luodaan uudelleen lowpoly-muotoon. Käytännössä korkean resoluution malli "päällystetään" uudella mallilla, joka pyrkii noudattamaan alkuperäisen highpoly-mallin muotoa, mutta vähemmällä vertex-pisteillä ja hyvän topologian periaatteilla. Lopputuloksena tästä on kevyempi, optimoitu 3D-malli, joka saadaan tekstuurilla näyttämään lähes identtiseltä highpoly-mallin kanssa.

Baking: Beikkaus tarkoittaa prosessia, jossa valmistetaan tekstuurikarttoja. Esimerkiksi, kun halutaan highpoly-mallin yksityiskohdat lowpoly-mallille, niin sitä varten valmistetaan normal-kartta. Beikkaus-ohjelmistoon syötetään molemmat mallit ja ohjelmisto vertaa niitä toisiinsa ja tallentaa eroavaisuudet normal-karttaan kuvatiedostoksi. Erilaisia tekstuurikarttoja voidaan valmistaa useita eri käyttötarkoituksiin ja tarpeisiin.

Blend shape: Alkuperäisestä 3D-objektista muokattu muoto, jota käytetään animaatioiden tukena. Blend shape-metodin avulla voidaan esimerkiksi animoida kasvojen hienovaraisempia liikkeitä ja ilmeitä, mihin pelkkään luustoon pohjautuva animaatiokontrolli ei usein kykene. Tätä metodia käytettäessä tehdään alkuperäisestä objektista kopio, jota muokataan ja on tärkeää, ettei geometriaan lisätä tai siitä poisteta vertex-pisteitä, sillä blend shape käytännössä taltioi kaikkien käytössä olevien pisteiden sijainnin 3D-tilassa, jolloin alkuperäinen objekti pystyy portaattomasti muuntumaan kopion osoittamaan muotoon ja takaisin.

4 OHJELMISTOT

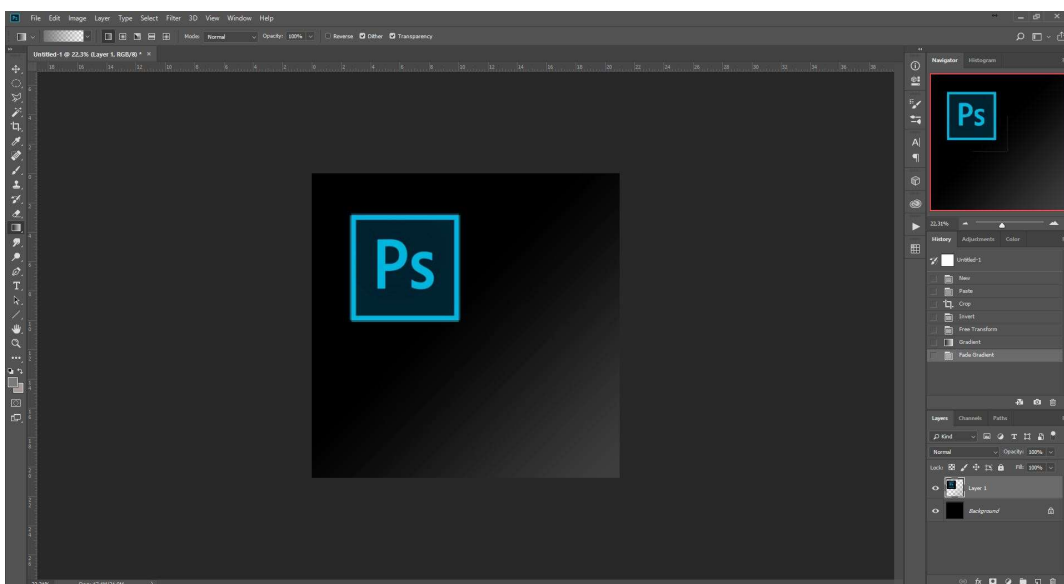
3D-hahmon luomisessa käytetään yleensä useampaan ohjelmistoa, joista jokainen on erikoistunut johonkin tuotannon osa-alueeseen. Tässä kappaleessa esitellään lyhyesti tässä työssä käytetyt ohjelmistot.

Työssä käytetystä kymmenestä ohjelmasta kolme on ilmaisohjelmia, ja useimmista on saatavilla ilmainen kokeiluversio tai opiskelijalisenssi. Ohjelmien kuvauksissa pyritään mainitsemaan vaihtoehtoisia ohjelmia, jos sellaisia on saatavilla.

4.1 Photoshop

Photoshop on Adoben omistama kuvankäsittelyohjelma. Se on suosittu ja yleisesti käytetty ohjelma niin valokuvaajien, graafikoiden kuin pelialan ammattilaistenkin piirissä. Se tarjoaa laadukkaita ja monipuoliset työkalut kuvankäsittelyyn.

Photoshop on nykypäivänä lähes synonyymi kuvankäsittelylle, eikä sille ole suoraa vartenotettavaa kilpailijaa, mutta ilmainen Krita on hyvä vaihtoehto. (Adobe 2019, viitattu 28.3.2019). (Krita 2019, viitattu 28.3.2019).



KUVIO 7. Photoshop 2018 -käyttöliittymä.

4.2 3D-mallinnus

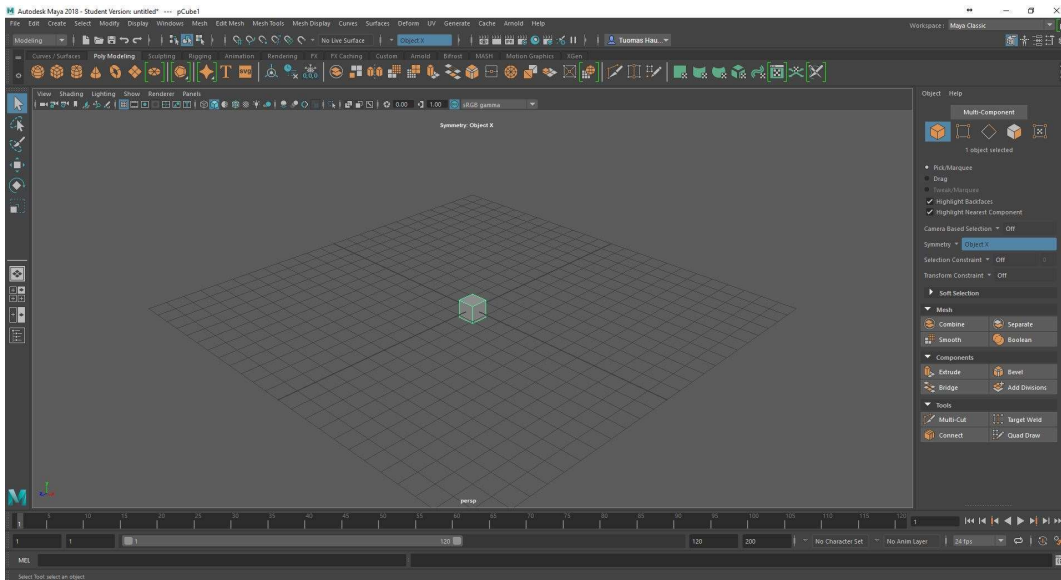
Kolmiulotteisten mallien tekemiseen on saatavilla monta eri ohjelmistoa, niin ilmaisia kuin maksullisiakin. Teoriassa samankaltaisen lopputuloksen voi saavuttaa millä tahansa ohjelmistolla, mutta todellisuudessa ohjelmistojen sisäiset työkalut vaihtelevat, jolloin haluttu lopputulos voidaan tuottaa huomattavasti helpommin tai nopeammin jollain tietyllä ohjelmistolla.

Yleisesti suosittuja 3D-mallinnusohjelmistoja ovat esimerkiksi Autodeskin 3D Studio Max ja Maya sekä Blender. Näistä kolmesta Blender on ilmainen ja siksi suosittu erityisesti harrastelijoiden ja puoliammattilaisten keskuudessa. Maya koetaan pelialalla standardiksi (Medium 2018, viitattu 28.3.2019) ja 3D Studio Max on usein käytössä arkkitehtuurin visualisoinnissa. Tässä työssä tarkastelemme mallinnusta käyttäen Mayaa ja Blenderiä.

4.2.1 Maya

Maya on Autodeskin omistama 3D-mallinnusohjelma ja on erityisesti peli- ja elokuva-alan suosima. Sen työkalut soveltuvat hyvin pelikehitykseen, koska sillä voi muun muassa mallintaa, rigata ja animoida. Maya on 3D:n vahva yleistyökalu.

Maya on kuukausimaksullinen, mutta esimerkiksi Blender on hyvä ilmainen vaihtoehto. Mayasta on saatavilla myös kolmen vuoden ilmainen opiskelijalisenssi (Autodesk 2019, viitattu 28.3.2019).

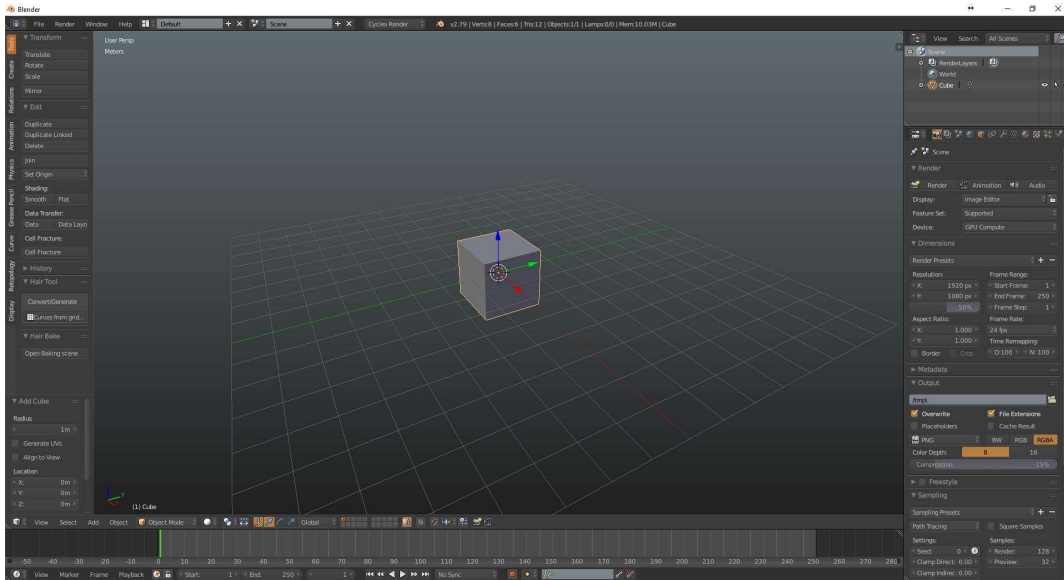


KUVIO 8. Maya 2018 -käyttöliittymä.

4.2.2 Blender

Blender on avoimen lähdekoodin ilmainen 3D-mallinnusohjelma, joka on viime vuosina noussut suosioon ja vakavasti otettavaksi kilpailijaksi Mayalle ja 3D Studio Maxille tehokkaan kehityksensä vuoksi. Se tarjoaa pitkälti samat ominaisuudet kuin Mayakin, mutta se ei ole niin sanottu ”industry standard” - alan standardi, kuten Maya. Blenderillä voi muun muassa mallintaa, teksturoida, rigata, animoida, aivan kuten Mayallakin (Blender 2019, viitattu 28.3.2019).

Valinta näiden kahden välillä on ensisijaisesti kysymys hinnasta ja käyttöliittymän omaksumisesta. Teoriassa Blenderillä voisi tehdä kaikki tässä työssä mainitut vaiheet, mutta kaikki sen työkalut eivät ole tarpeeksi kehittyneitä, jotta lopputulos olisi täysin samaa tasoa.

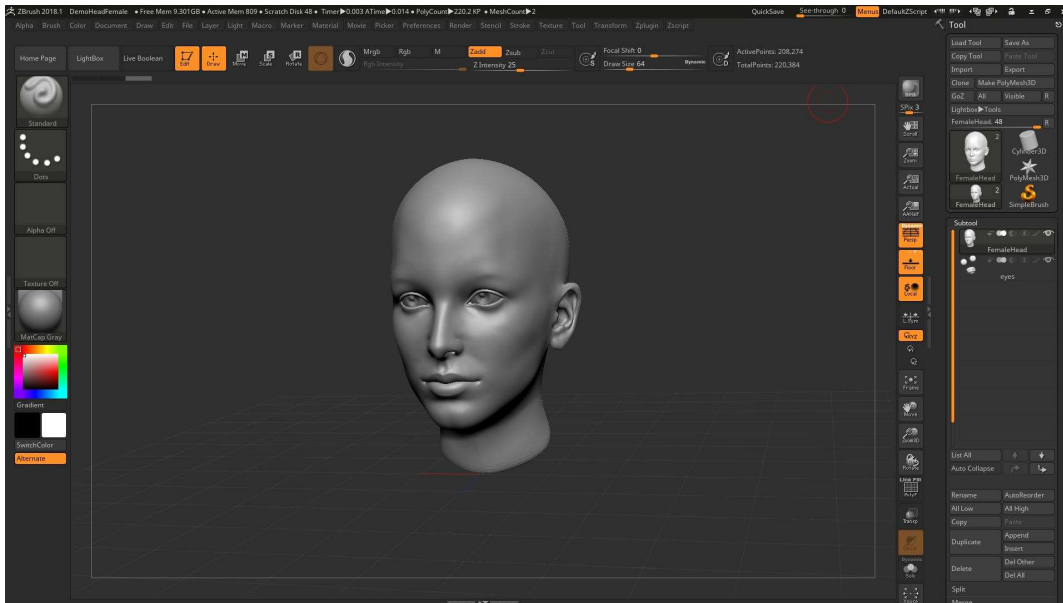


KUVIO 9. Blender 2.79 -käyttöliittymä.

4.3 ZBrush

ZBrush on Pixologicin omistama 3D-mallinnusohjelmisto, mutta lähestymistavaltaan se eroaa huomattavasti Mayasta ja Blenderistä. Se on niin sanottu ”sculptaus”-työkalu, jolla 3D-malleja veistään ikään kuin digitaalisesta savesta. Tämä tuottaa usein hyvin yksityiskohtaisia malleja, joiden polygoniluku tai ”resoluutio” on erittäin korkea. Tällöin puhutaan highpoly-mallista, jonka vertex-pisteiden lukumäärä voi olla kymmenissä miljoonissa.

Highpoly mallit ovat ZBrushin vahvuus, sillä esimerkiksi Mayalla ja Blenderillä on vaikeuksia prosessoida huomattavan suuren resoluution hahmoja. ZBrushin työkalut ovat erittäin monipuoliset, mutta yleisesti koetaan, että käyttöliittymä vaatii totuttelua, sillä se poikkeaa huomattavasti perinteisimmistä mallinnusohjelmista. Tämän lisäksi se on poikkeuksellisen kallis ohjelmisto, mutta siitä huolimatta ZBrush on myös alan standardi ja laajalti käytetty. Vaihtoehto ZBrushille on esimerkiksi Autodeskin Mudbox-ohjelmisto. (Pixologic 2019, viitattu 28.3.2019).

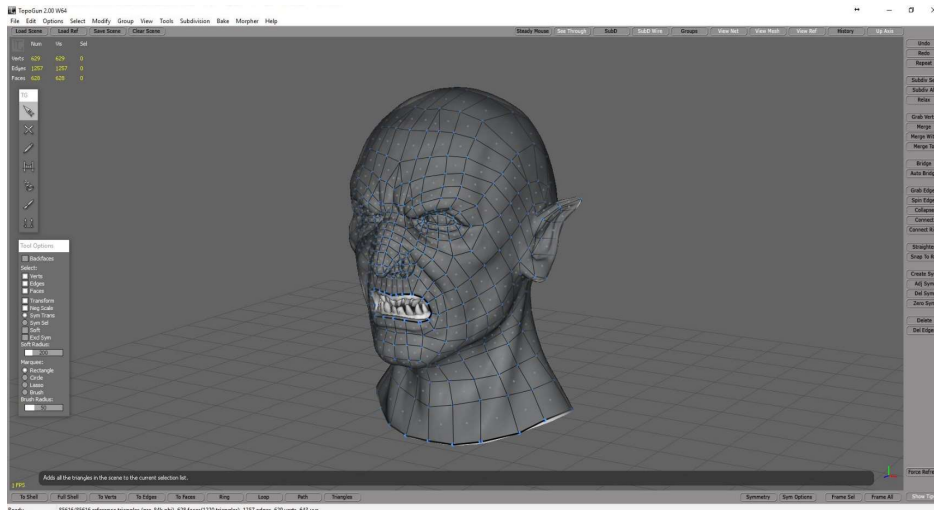


KUVIO 10. ZBrush 2018 -käyttöliittymä.

4.4 Topogun

Topogun on retopologia-ohjelmisto, jossa korkean resoluution mallin pohjalta luodaan optimoitu, matalamman resoluution malli, jota voidaan käyttää animaatioissa ja pelimoottorissa, toisin kuin raskasta highpoly-mallia. Useimmat 3D-mallinnusohjelmat omaavat retopologia-työkaluja ja saman lopputuloksen voi saavuttaa esimerkiksi Mayassa tai Blenderissä, mutta koska Topogun on erikoistunut retopologiaan, on prosessin aloittaminen nopeaa ja helppoa.

Topogun on laajalti käytetty peliyrityksissä ja sen hinta on noin sata dollaria. Ohjelmistosta on saatavilla myös kokeiluversio. (Topogun 2018, viitattu 4.4.2019).

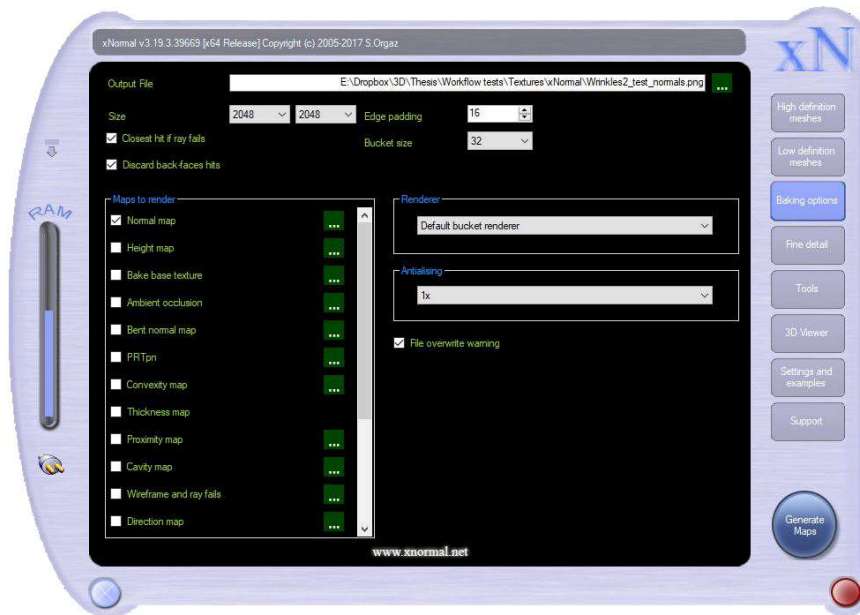


KUVIO 11. Topogun 2 -käyttöliittymä.

4.5 xNormal

xNormal on ilmainen ohjelma, jolla voidaan valmistaa tai "beikata" tekstuurikarttoja. Tekstuureja voidaan valmistaa muillakin ohjelmistoilla, mutta xNormalia suositetaan sen helppokäyttöisyyden ja korkealaatuisten ja monipuolisten karttojen vuoksi.

Ulkoasultaan ohjelmisto on vanhentunut, mutta toiminnaltaan se on yhä yksi lajinsa parhaita. Ohjelmiston historia on hieman hämärän peitossa, eikä esimerkiksi ohjelman alkuperäistä julkaisu-vuotta ole selvillä (xNormal 2019, viitattu 4.4.2019).

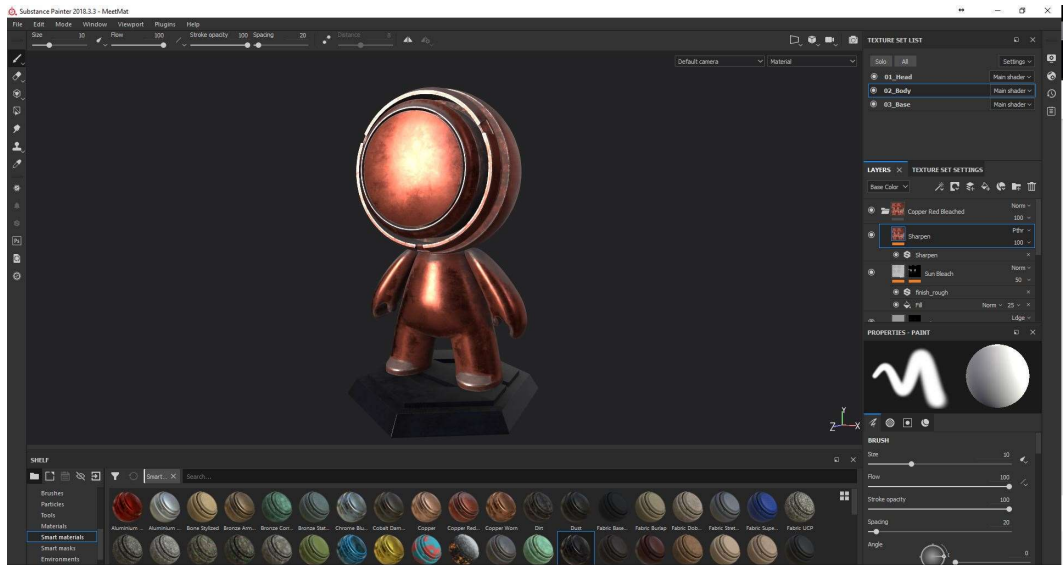


KUVIO 12. xNormal-käyttöliittymä.

4.6 Substance Painter

Substance Painter on Allegorithmicin kehittämä, ja hiljattain Adoben hankkima, teksturointi -ohjelmisto. Sen avulla voidaan 3D-objektin pintaan maalata tai muutoin lisätä erilaisia materiaaleja ja tekstuureja hyvin monipuolisesti ja niitä voidaan hallita sekä muokata kerroksittain. Substance Painterissa on valmiina kymmeniä eri materiaaleja ja lisää voi joko valmistaa itse tai hankkia esimerkiksi Allegorithmicin internet-sivuilta substance share -kirjastosta, johon käyttäjät voivat jakaa materiaalejaan.

Substance Painter on käytössä useissa pelistudioissa, kuten esimerkiksi Ubisoft ja Epic games. Kilpaileva teksturointiohjelmisto sille on Mari (Foundry 2019, viitattu 4.4.2019), jossa on muutamia hieman kehittyneempiä ominaisuuksia, mutta vastaavasti kalliimpi hinta. (Allegorithmic 2019, viitattu 4.4.2019).

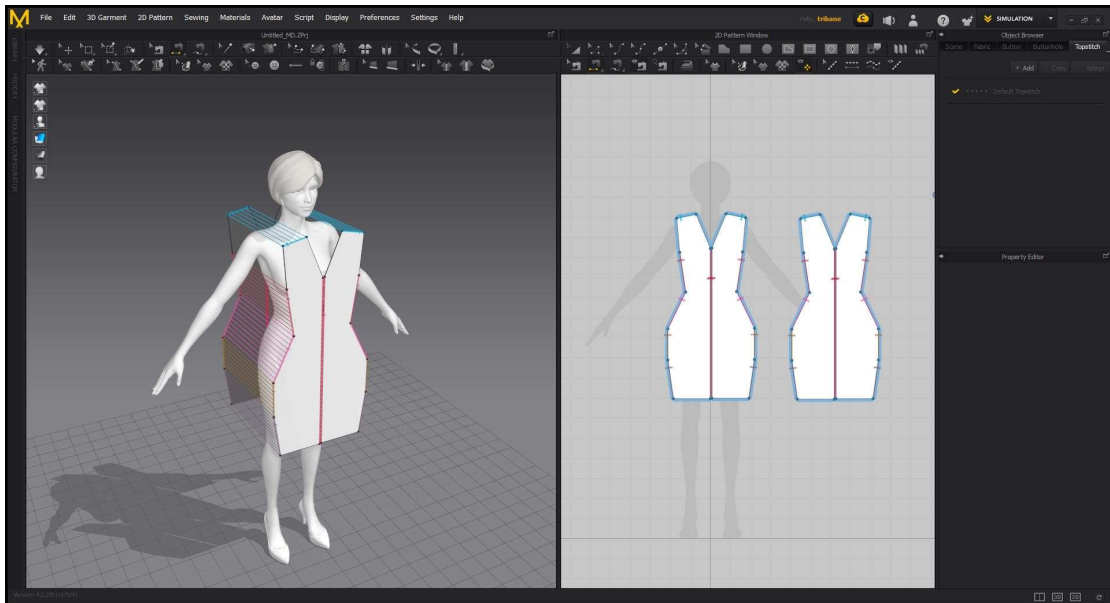


KUVIO 13. Substance Painter 2019 -käyttöliittymä.

4.7 Marvelous Designer

Marvelous Designer on 3D-vaatetusohjelmisto, jota käytetään erilaisten vaatekappaleiden valmistamiseen ja simulointiin. 3D-hahmon vaatetus perinteisin mallinnuskeinoin saattaa olla haastavaa, mutta Marvelous Designer on erikoistunut juuri tähän. Sen avulla vaatteet luodaan piirtämällä 2D-kaavat ja määrittämällä yhteen ommeltavat sivut. Sitten ohjelma simuloi vaatekappaleet hahmon ylle yhdistämällä osat toisiinsa ja asettamalla ne hahmon pinnalle. Vaatetta voidaan muokata ja simuloida yhä uudelleen, kunnes lopputulos on halutunlainen.

Aiemmin pelihahmojen vaatteet luotiin ohjelmilla, kuten ZBrush, mutta nyt yhä useampi pelistudio käyttää tuotannossaan Marvelous Designeria, sillä se tuottaa nopeasti realistisia ja näyttäviä tuloksia. Jatkotoimenpiteenä Marvelous Designerissa luotuja objekteja voidaan toki muokata lisää muissa 3D-ohjelmissa. (Marvelous Designer 2018, viitattu 4.4.2019.)

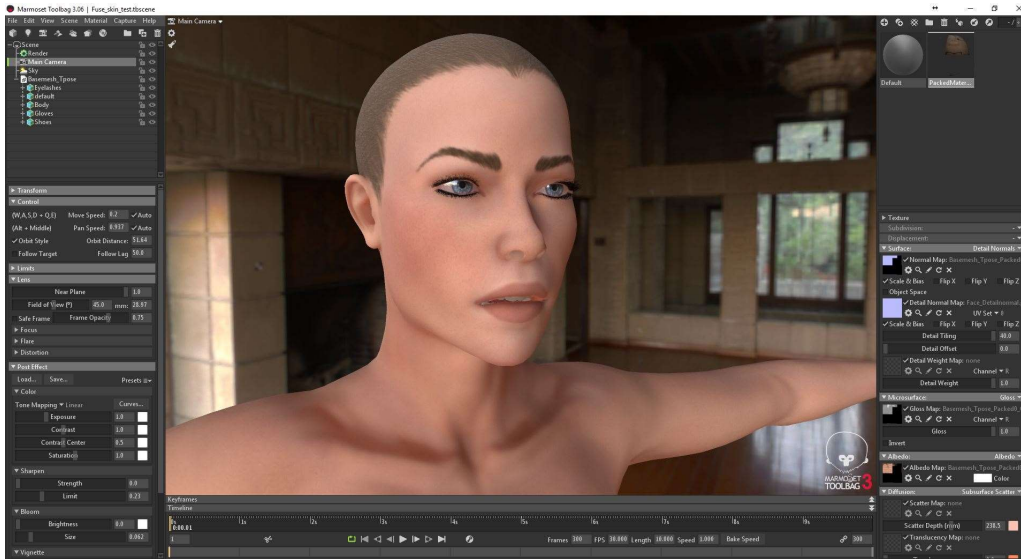


KUVIO 14. Marvelous Designer 8 -käyttöliittymä.

4.8 Marmoset Toolbag 3

Marmoset Toolbag on reaaliaikarenderointi-ohjelmisto, jolla voi myös beikata, eli valmistaa tekstuurikarttoja. Sen avulla 3D-artistit voivat helposti esitellä töitään esimerkiksi asiakkaille, sillä valmiit objektit ja tekstuurit voidaan yhdistää, valaista ja lopputulosta tarkastella korkealaatuisena reaaliajassa.

Valmiit tuotokset voi tallentaa viewer-muotoon, jolloin asiakkaalle voi lähettää html-tiedoston, jonka avulla pystyy helposti tarkastelemaan hahmoa suoraan selaimessa ja halutessaan, mallin voi myös ladata suoraan ohjelmasta Sketchfab-verkkokauppaan myytäväksi. (Marmoset 2018, viitattu 4.4.2019.) (Sketchfab 2019, viitattu 4.4.2019).



KUVIO 15. Marmoset Toolbag 3 -käyttöliittymä.

4.9 Unreal Engine 4

Unreal Engine 4 on Epic Gamesin kehittämä ja julkaisema ilmainen pelimoottori niin harrastus- kuin ammattikäyttöönkin. Se on jo vuosia ollut yksi suosituimmista pelimoottoreista monipuolisten ja tehokkaiden työkalujensa ansiosta ja on tunnettu hyvistä grafiikkaominaisuuksistaan.

Unreal Enginen suora kilpailija on Unity-pelimoottori, joka on tunnettu erityisesti mobiilipeleistään. Unrealia pidetään kuitenkin vahvempana 3D pc- ja konsolipelien kehitysalustana, pitkän historiansa ja monipuolisten 3D-työkalujensa ansiosta. (New Gen Apps 2018, viitattu 10.4.2019.) (Unreal Engine 2019, viitattu 4.4.2019).



KUVIO 16. Unreal Engine 4 -käyttöliittymä.

5 KONSEPTITAIDE

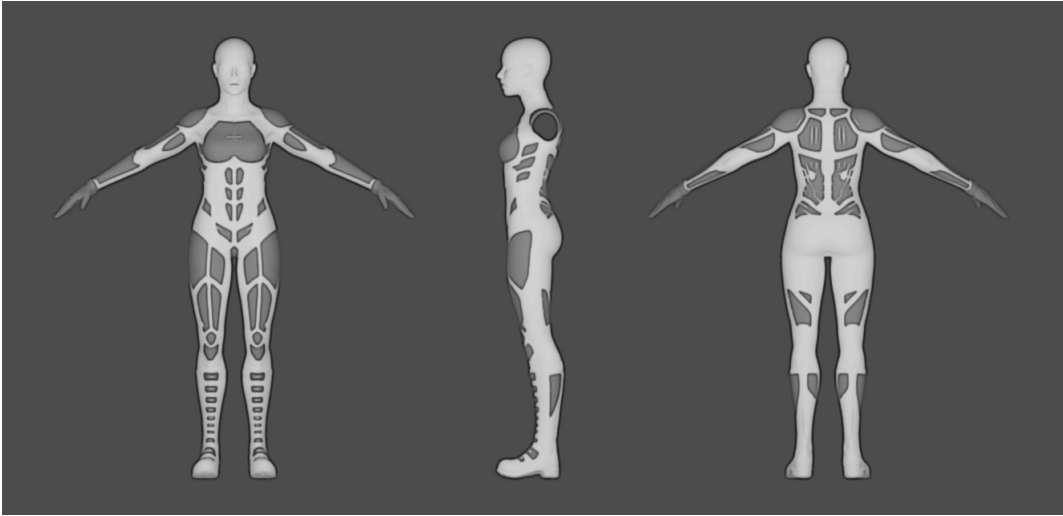
Konseptitaiteen tarkoitus on kartoittaa hahmon ulkonäkö visuaalisesti piirtämällä siitä yksi tai useampi versio, perustuen hankittuihin referenssikuviiin ja pelisuunnittelun linjauksiin. Voidaan suunnitella esimerkiksi useampi variaatio hahmon vaatetuksesta tai hiuksista ja valita niistä parhaimmat jatkoon ja rajata lopulliseen versioon halutut designit. Konseptiartistit tuottavat kuvat haluamallaan tavalla, usein Photoshopissa digitaalisesti piirtäen ja nämä kuvat annetaan eteenpäin 3D-mallintajille, jotka rakentavat hahmon kuviin perustuen. (Concept Art Empire 2019, viitattu 12.5.2019.)

Tässä työssä konseptointiin käytettiin Adobe Photoshop -ohjelmistoa ja Wacom -piirtotablettia. Hahmosta tehtiin tässä tapauksessa vain kaksi variaatiota, takin kanssa ja ilman. Konseptitaide perustui 2.luvussa esitelyihin ideoihin ja referenssikuviiin. Hahmosta haluttiin vaaleahiuksinen, sinisilmäinen, noin 30-vuotias nainen, jonka asu olisi futuristinen, taktinen, vartalonmyötäinen ja väritään vaalea. Konseptitaide itsessään oli digitaalisen piirtämisen ja ”photobashingin” (Concept Art Empire 2019, viitattu 5.4.2019), eli valokuvien manipuloinnin, yhdistelmä.



KUVIO 17. Hahmon konseptivariaatiot.

Siinä missä varsinainen konseptikuva antaa hahmosta yleensä imartelevan ja taiteellisen tunnelmakuvan, on mallinnusta varten hyvä tehdä myös teknisemmät, niin sanotut ortografiset kuvat hahmosta. Tällä tarkoitetaan kohteen kuvaamista kaksiulotteisena ilman perspektiivivääristymiä suoraan edestä, sivulta ja takaa, jotta mallintaja saa kokonaisvaltaisen ja kattavan kuvan hahmosta ja voi mallintaa suoraan kuvien päälle 3D-ohjelmassa.



KUVIO 18. Ortografinen projektio edestä, sivulta ja takaa.

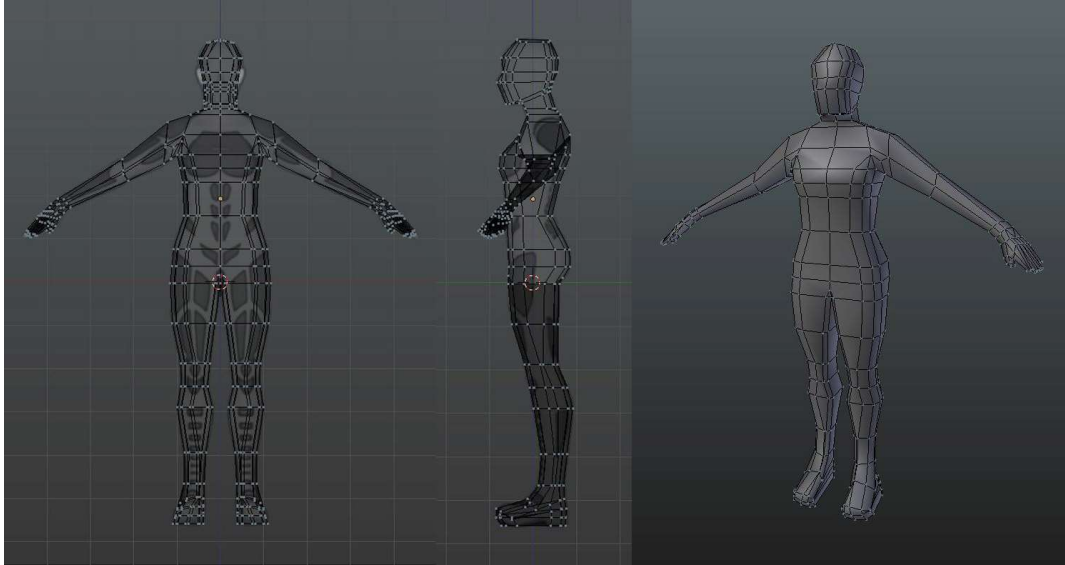
6 3D-MALLINNUS

6.1 Lowpoly

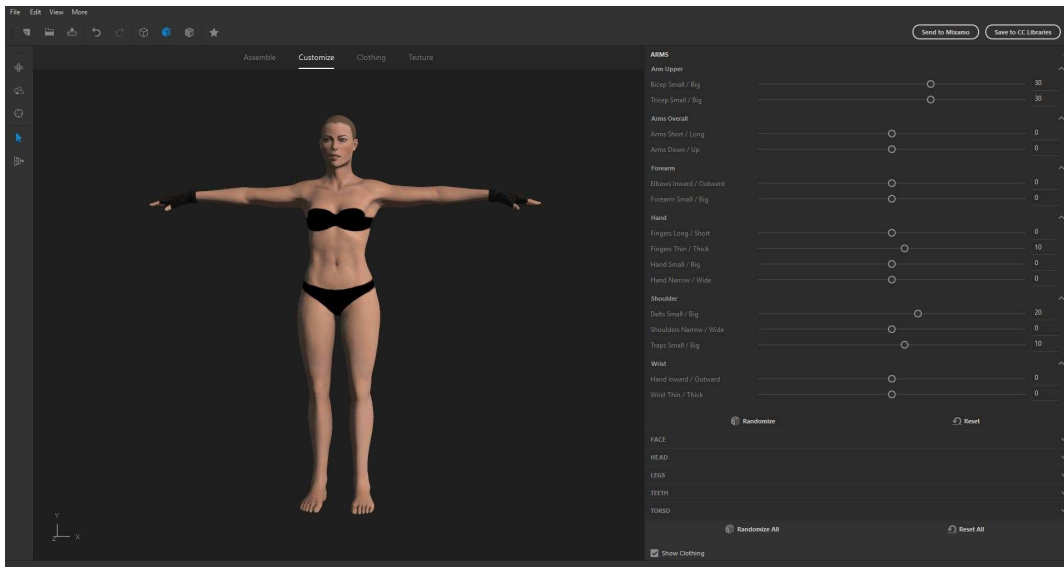
Konseptikuvien pohjalta varsinainen 3D-mallinnus voidaan aloittaa. Tässä vaiheessa voi 3D-artisti päättää tehdäänkö ensin yksinkertainen versio, eli lowpoly-malli Mayaassa tai Blenderissä vai hyödynnetäänkö ohjelmistoja, kuten esimerkiksi Fuse tai MakeHuman, joissa voi nopeasti luoda ihmishahmon ja kustomoida ulkonäköä. Nämä ohjelmistot tarjoavat nopean pohjan hahmon luontiin, mutta ovat usein liian geneerisiä ja heikkolaatuisia ollakseen varteenotettava vaihtoehto lopulliselle pelihahmolle. Studioiden sisäisessä käytössä voi myös olla jokin valmis hahmopohja, jota hyödynnetään projektista toiseen ajan säästämiseksi (Flipped Normals 2018, viitattu 5.5.2019). Artistin niin halutessa, voidaan tämä vaihe ohittaa täysin ja siirtyä suoraan korkean resoluution, eli highpoly-mallin työstöön ZBrushissa. Tämä voi kuitenkin olla haastavampi lähestymistapa ja vaatii kokemusta, koska on helppo eksyä työstämään pieniä yksityiskohtia ennen kuin perusmuoto on valmis, joka saattaa johtaa ongelmiin ja epätasaiseen laatuun.

Hahmo on hyvä mallintaa tiettyyn asentoon myöhemmin tulevaa luuston luontia ajatellen. Tyypillisiä asentoja ovat niin sanottu T-pose, jossa hahmon kädet ovat ojennettuina sivulle, yhdeksänkymmenen asteen kulmassa vartaloon nähden ja A-pose, jossa kädet ovat edelleen sivuilla, mutta alempana, noin neljänsäksenkymmenen asteen kulmassa vartaloon nähden. A-pose asentoa suositetaan nykyään enemmän, sillä siinä hahmo on luonnollisemmassa, rennommassa asennossa, jolloin tekstuurien venymistä tai muodon vääristymistä esiintyy vähemmän. Tähän liittyen myös muita kehon osia on hyvä tarkastella animaation kannalta. Esimerkiksi silmäluomet voidaan mallintaa puoliksi suljetuiksi. Näin silmäluomiin saadaan enemmän geometriaa, jota animaatio hyödyntää ja tekstuurien venymistä ilmenee vähemmän. Vastaavasti suu voidaan mallintaa myös hieman avoimaksi, jotta myös suun sisäpuolelle saadaan hieman lisää geometriaa, joka näkyy, kun hahmo avaa suun animaatioissa. Luonnollisesti myös hampaat, ikenet ja kieli tulee mallintaa. Lowpoly-mallin luodaan ensin yksinkertaisesta geometriasta, kuten kuutioista ja tarkoituksena on saada vain hahmon perusmuodot esille. Mallinnuksessa on hyvä käyttää mirror-modifikaatiota, joka peilaa yhdelle puolelle mallia tehdyt muutokset myös toiselle puolelle. Tällöin tarvitsee mallista tehdä manuaalisesti vain toinen puoli ja ohjelmisto hoitaa automaattisesti toisen puolen. Sen jälkeen muotoja

voi tarkentaa lisäämällä objektiin enemmän vertex-pisteitä ja sen myötä polygoneja, kunnes saavutetaan halutunlainen mallipohjan, josta highpoly-versio voidaan veistää. Mallipohja tallennetaan .obj -tiedostomuotoon ja siirretään ZBrushiin. Lowpoly vaihe on periaatteiltaan samanlainen niin Mayassa kuin Blenderissäkin.



KUVIO 19. A-asennossa oleva hahmo muodostetaan ensin yksinkertaisemmasta geometriasta.



KUVIO 20. T-asennossa oleva Fuse-hahmo ja Fusen käyttöliittymä, jossa monia muokkausmahdollisuuksia.

6.2 Highpoly

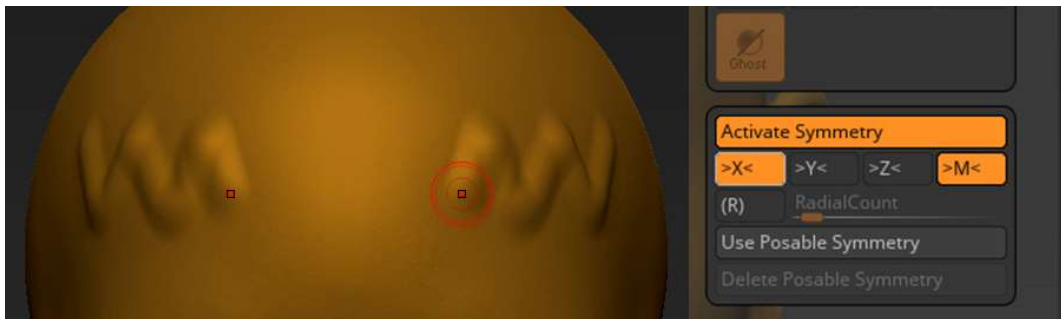
Kun halutunlainen pohjamalli on luotu, tuodaan se ZBrushiin, jossa hahmoa hiotaan yksityiskohtaisemmaksi. Samat ortografiset ohjekuvat on hyvä avata myös ZBrushiin, jotta design pysyy yhtenäisenä. Highpoly-vaiheessa saattaa mallissa olla lopulta kymmeniä miljoonia vertex-pisteitä, verrattuna lowpoly-lähtökohtaan, jossa on todennäköisesti muutamia satoja vastaavia pisteitä. Tämän vuoksi highpoly-malli on raskas tietokoneelle ja tiedostot saattavat kasvaa jopa useamman gigatavun kokoiseksi. Tästä syystä on tärkeää aloittaa veistäminen suurista muodoista ennen kuin hahmoon lisätään erityisen paljon resoluutiota. Hyvä anatomian tuntemus on hyödyksi veistoprosessissa. Tässä työssä luodussa highpoly-mallissa oli korkeimmillaan noin 18,5 miljoonaa vertex-pistettä, jolloin sen korkean resoluution omaavaan pintaan pystyi veistämään ihohuokosten tasoisia yksityiskohtia.

ZBrushissa työskennellään enimmäkseen erilaisilla siveltimillä (brushes), josta myös ohjelma on saanut nimensä. Tässä työssä käytettiin suurimman osan ajasta vain noin viittä eri siveltintä; move, claybuildup, damstandard, smooth ja morph -siveltimiä. Move-siveltimellä voi liikuttaa halutun koosta aluetta mihin suuntaan vain ja sillä on hyvä oikoa muotoja. Claybuildup-sivellin taas kerryttää lisää massaa maalattuun kohtaan, tai alt-näppäintä pohjassa pitämällä se kaivaa vastaavasti koloa pintaan. Damstandard-sivellin on hyödyllinen kaikenlaisten rakojen tekemiseen, ja alt-variaatio siitä vetää vertexejä toisiinsa ja tekee hieman terävämmän reunan ja antaa täten ryhtiä kokonaisuudelle, jottei kaikki ole liian pehmeää ja vailla muotokontrastia. Shift-näppäimellä otetaan käyttöön smooth-sivellin, joka tasoittaa pintaa ja häivyttää haluttuja piirteitä. Morph-sivellin vaatii toimiakseen hahmon silloisen tilan tallentamista morph-kohteeksi, morph target-valikosta ja sitä käytetään muiden siveltimien ohella, kun tietty osa hahmosta halutaan palauttaa aiemmin tallennettuun tilaan. Ihon yksityiskohdat hahmon kasvoihin luotiin käyttämällä ilmaisia, Rafael Souzan luomia alfasiveltimiä (Cg Elves 2016, viitattu 21.3.2019).



KUVIO 21. ZBrushin sivellinvalikko. Tarvittaessa eri siveltimiä voi hankkia lisää esimerkiksi Artstationin verkkokaupasta (Artstation 2019, viitattu 21.3.2019).

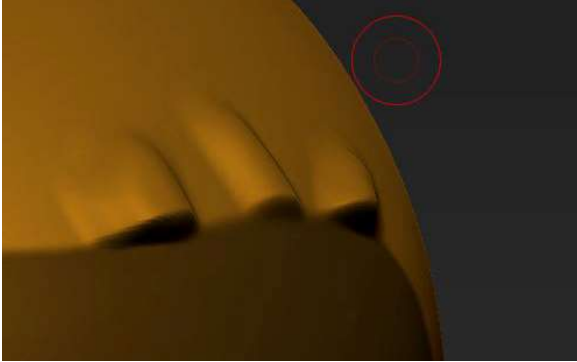
Samoin kuin lowpoly-mallinnuksessa, myös highpoly hahmomallinnuksessa käytetään symmetria työkaluja, jotta molemmat puolet noudattavat samoja muotoja. Tällöin vältetään tekemästä kaksinkertaista työtä, kun ohjelmisto muokkaa toisen puolen hahmosta identtiseksi manuaalisesti työstehtyn puolen kanssa. Myöhemmässä vaiheessa symmetria voidaan kytkeä pois päältä hienovaraisen muutosten tekemiseksi esimerkiksi kasvoihin, sillä ihmisen kasvat eivät ole täysin symmetriset.



KUVIO 22. X-akselin symmetria mahdollistaa hahmon vasemman ja oikean puolen yhtäaikaisen työstämisen.

Zbrushissa voidaan myös luoda maski mallin tiettyihin osiin ja silloin muutokset koskevat vain ei-maskattua aluetta. Maskatut alueet voidaan myös eristää näkymässä tai vaikka erottaa omaksi objektikseen. Tällöin uudesta osasta tulee niin sanottu subtool ja sitä voidaan muokata täysin erillään alkuperäisestä objektista. Objekteja voidaan myös jakaa polygroupeihin, jolloin niiden rajaa-
 mia alueita voidaan muokata itsenäisesti, mutta ne ovat silti yhä edelleen osa alkuperäistä objektia. Polygroupit voidaan myös helposti muuntaa polypaintiksi, josta on apua myöhemmin teksturoin-

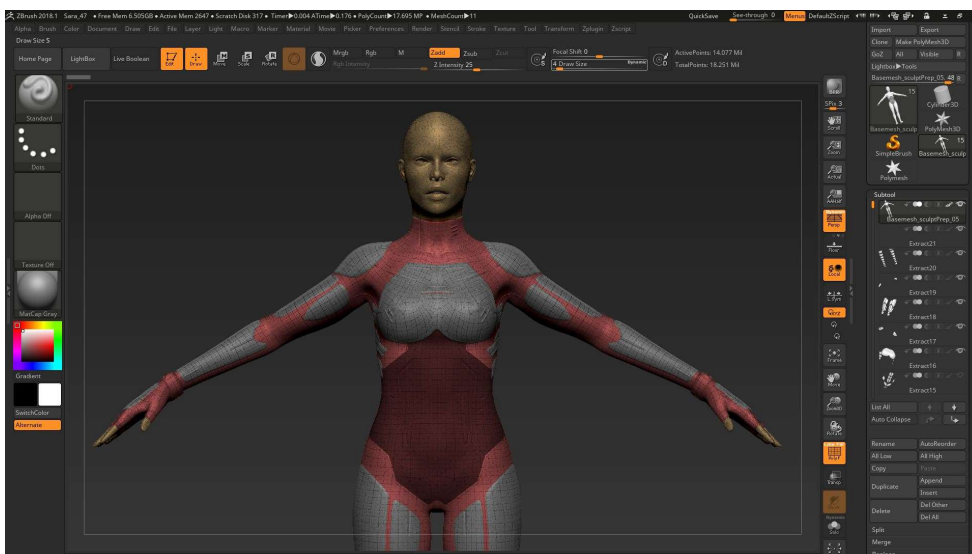
nissa. Polypaintilla, tai "polymaalilla" voidaan hahmon eri osia värittää ja tämä voi toimia joko lopullisena värinä tai sitten polyväreistä voidaan luoda väri ID-kartta, jonka perusteella teksturointiohjelmissä osaa sijoittaa erilaiset materiaalit oikeisiin kohtiin mallia.



KUVIO 23. Objektin pintaan tehty maski näkyy tummana alueena, johon muokkaukset eivät vaikuta.

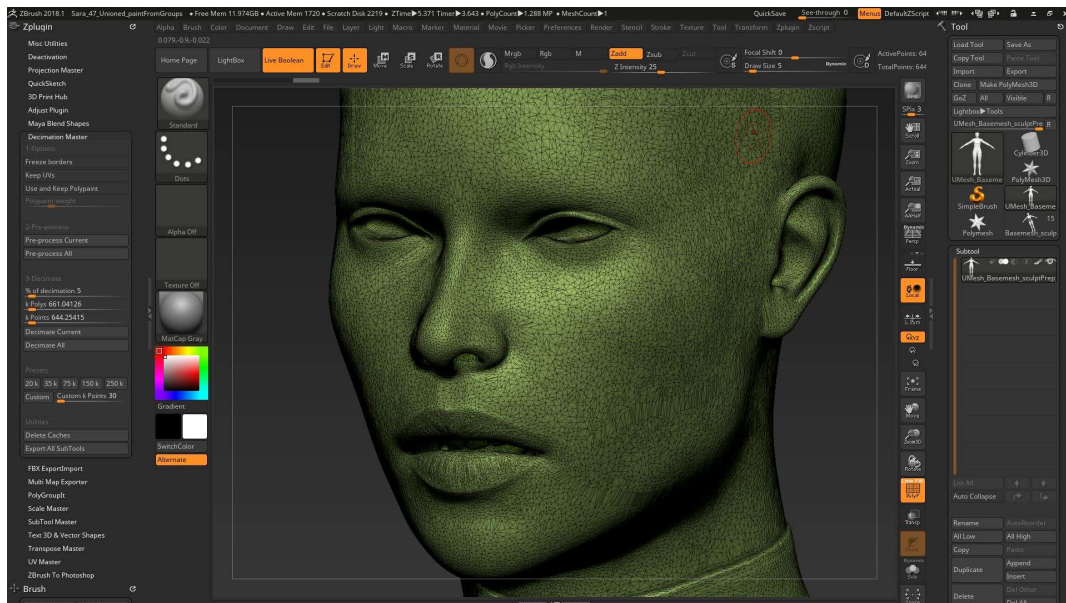


KUVIO 24. Subtool-valikko, jossa nähtävillä hahmon eri osat.



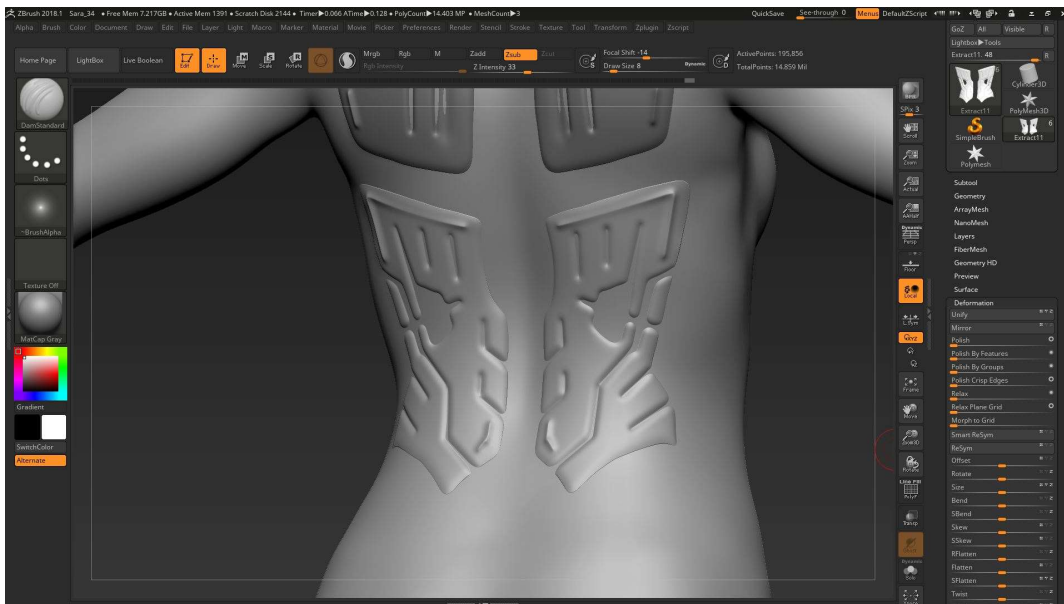
KUVIO 25. Polygroupit näkyvät eri väreinä hahmossa.

Mallien iteroinnissa layerit, eli tasot, on hyödyllinen ominaisuus ZBrushissa. Tasot toimivat pitkälti samalla periaatteella kuin esimerkiksi Photoshopissa, eli niihin voi taltioida eri variaatioita esimerkiksi hahmon yksityiskohdista. Kun tasolle on tallennettu halutut muokkaukset, voi sen tarvittaessa piilottaa näkyvistä ja tehdä uudelle tasolle toinen variaatio samasta kohdasta ja sitten vertailla tasojen muokkauksia toisiinsa. Lopuksi kaikki tasot ja subtool-objektit on hyvä yhdistää yhdeksi objektiksi ja ikään kuin lukita hahmo valmiiseen muotoon. Hahmon voi yhdistää joko merge-työkalulla, joka liittää kaikki osat yhdeksi subtooliksi, mutta sallii silti tallentaa kaikki osat erillisinä tiedostoina, tai sitten boolean-työkalulla, joka vähentää risteävät osat toisistaan ja poistaa piiloon jäävät vertex-pisteet ja tallennettaessa luo vain yhden tiedoston, mutta jonka sisällä kaikki osat ovat vielä erillisinä. Molemmille tavoille voi olla käyttöä eri projekteissa, mutta tässä työssä käytettiin jälkimmäistä vaihtoehtoa. Hahmosta on hyvä tallentaa kaksi eri versiota, joista ensimmäinen on valmis, korkean resoluution malli sellaisenaan ja toinen on decimation master-työkalun läpi ajettu malli. Ensimmäistä mallia käytetään myöhemmin normal-kartan valmistukseen, jonka vuoksi sen kannattaa olla mahdollisimman korkealaatuinen. Decimation master -työkalu analysoi highpoly-mallin ja optimoi, eli vähentää sen vertex-lukua siten, että mallin ulkomuoto säilyy samanlaisena mahdollisimman pitkään. Niin kauan kuin visuaalista eroa ei havaita, kannattaa vertex-lukua laskea mahdollisimman alas. Tätä optimoitua mallia käytetään retopologia-vaiheessa, jossa mallista tehdään vielä optimoidumpi versio.



KUVIO 26. Decimation master -työkalulla optimoitu hahmo. Objekti säilytti paljon yksityiskohtia, mutta topologia jättää toivomisen varaa.

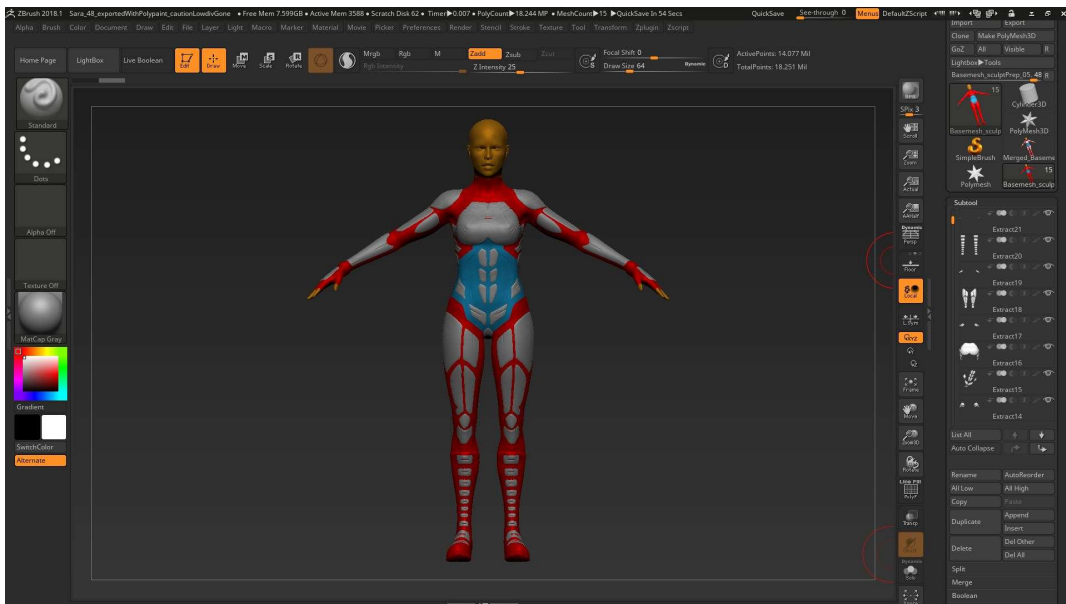
Ajan säästämiseksi tässä työssä käytettiin pohjana Fuse-ohjelmassa luotua mallia. ZBrushissa mallin kehoa muokattiin ensin paremmin konseptiin sopivaksi ja kasvoja muokattiin huomattavasti referenssikuviiin perustuen. Perusmuotojen löytyttyä, muokattiin seuraavaksi hahmon puvun reunat, kuten kaulus, hihansuut ja saappaat. Sen jälkeen kehoon piirrettiin maskit panssariosia varten. Nämä kohdat eristettiin omiksi subtooleikseen extract-toiminnolla, siten että jokaiselle uudelle subtoolille määritettiin automaattisesti uusi oma polygroup. Sitten niiden reunat hiottiin move- ja smooth-siveltimillä ja viimeisteltiin deformation -valikosta löytyvällä polish crisp edges -toiminnolla, joka tuotti huolitellun lopputuloksen. Seuraavaksi alkuperäiseen subtooliin veistettiin lisää yksityiskohtia damstandard-siveltimellä. Viimeiseksi hahmon kasvoihin ja käsiin luotiin yksityiskohtia, kuten ihohuokosia, rypyjä, sormenkynnet ja huulten pinnassa olevat pienet lovet. Kun varsinainen sculptaus oli tehty, siirrettiin polygroup data polypaintiksi väri ID-karttaa varten. Tämän lisäksi vatsan ja vyötärön alueelle maalattiin vielä manuaalisesti oma ID, jos kyseiselle alueelle haluttaisiin eri materiaali teksturointivaiheessa. Sitten eri subtoolit yhdistettiin yhdeksi objektiksi valitsemalla live boolean -tila ja geometry-valikosta boolean-toiminto: make boolean mesh. Tämä loi uuden objektin, joka valittiin sitten aktiiviseksi. Tässä vaiheessa hahmosta tallennettiin korkealaatuinen versio normal-karttaa varten. Sitten hahmo optimoitiin decimation master-työkälulla, jolloin sen vertex-luku putosi 12,9 miljoonasta 644 254:een pisteeseen. Tämä versio tallennettiin retopologiaa varten.



KUVIO 27. Puvun panssaripaneelien muokkausta.



KUVIO 28. Kasvojen ihon yksityiskohdat; huokokset, rypyt ja huulien pinta.



KUVIO 29. Polypaint, jonka pohjalta väri ID-kartta valmistetaan teksturointiohjelmassa.



KUVIO 30. Valmis highpoly-veistos.

6.3 Mallin iterointi

Kun highpoly-malli alkaa olla loppusuoralla, on tässä vaiheessa tuotantoa hyvä tarkistaa, että design on edelleen tarkoituksen mukainen, eikä muutoksia tai iterointia tarvitse tehdä. Jos kuitenkin todetaan, että hahmoon halutaan muutoksia, on niitä vielä tässä vaiheessa helppo tehdä, joustavan veistoprosessin ansiosta. Seuraavassa työvaiheessa hahmosta jäljennetään taas lowpoly-versio,

jonka jälkeen isompien muutosten tekeminen on hankalampaa, koska lowpoly-mallinkin jouduttai-
siin jäljentämään taas uudelleen.

Tässä työssä iteroinnille ei ollut tarvetta projektin ollessa fiktiivinen. Hahmon saappaille harkittiin vaihtoehtoisia designia, mutta se todettiin tarpeettomaksi ja lopulta haluttiin myös säästää aikaa. Todellista peliä kehitettäessä hahmoa on hyvä tarkastella koko ajan niin taiteellisesta, kuin käytännöllisestä näkökulmasta. Hahmon ulkonäköä iteroidessa, tulee hahmon edelleen täyttää pelimeka-
niikankin asettamat vaatimukset.

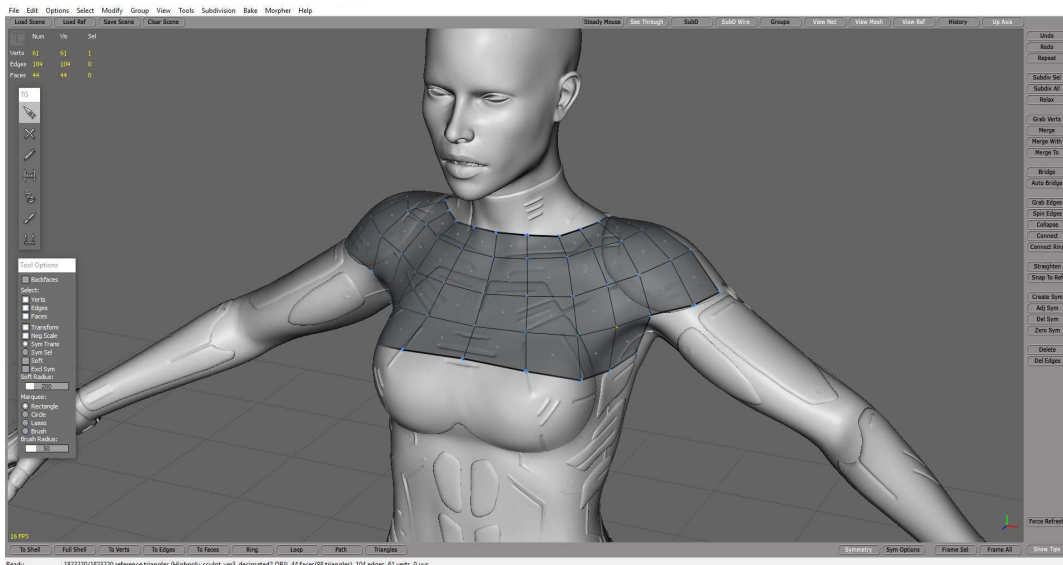
Tässä vaiheessa on myös hyvä käydä läpi hahmon eri osat ja mahdolliset lisävarusteet. Hahmo tarvitsee silmät, hiukset, hampaat, kielen ja tässä tapauksessa myös takin, aseensa ja asekotelon. Kaikki nämä osat tulee valmistaa 3D-ohjelmalla, noudattaen samoja askelia ja periaatteita kuin varsinainen hahmo. Poikkeuksena tästä on hahmon silmät, jotka voidaan ottaa Unreal Enginen digital human-esimerkki projektista. Silmät ovat rakenteeltaan erikoiset ja vaativat siksi hyvin monimutkaisen shaderin. Digital human-projektissa on sekä silmäobjekti, että sitä vastaava materiaali, joita voi vapaasti hyödyntää omassa peliprojektissa. (Unreal Engine 2019, viitattu 5.5.2019.) Toki silmän voi mallintaa myös perinteisellä tavalla laittamalla kaksi pyöreää objektia miltein päällekkäin, joista ulommainen on läpinäkyvä, mutta valoa voimakkaasti heijastava sarveiskalvo ja sisempi itse lasiainen sekä iiris ja pupilli (Saurabh Jethani 2018, viitattu 6.5.2019).

Kaikki hahmon osat luodaan ensin hahmon ylle hyvän yhteensopivuuden varmistamiseksi, mutta voidaan sen jälkeen käsitellä erikseen. Kun koko prosessi on käyty läpi jokaisen objektin osalta, kasataan koko malli lopulta yhteen luuston luontia varten.

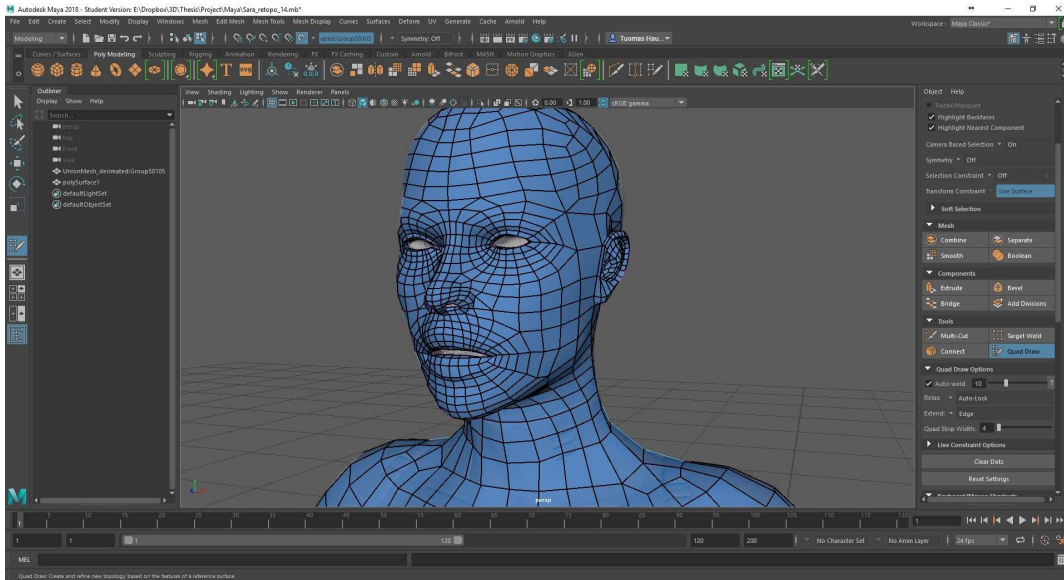
6.4 Retopologia

ZBrushissa optimoitu malli tuodaan seuraavaksi retopologia-ohjelmistoon, kuten esimerkiksi Topoguniin. Retopologioinnin voi periaatteessa tehdä millä tahansa 3D-mallinnusohjelmalla, ensisijaisesti käyttökokemus vaihtelee, ei niinkään lopputulos. Prosessi on sama joka ohjelmalla, mutta työkalut saattavat olla erilaiset. Topogun on retopologiaan erikoistunut ohjelmisto, jolla työn aloittaminen on nopeaa, mutta tässä työssä päädyin käyttämään tähän työvaiheeseen Mayaa, koska tarvittaessa käytössä oli myös kaikki muut 3D-mallinnustyökalut, mitä taas Topogun ei tarjoa.

Retopologian tavoitteena on kohta kohdalta luoda uusi pinta heikon topologian omaavan highpoly-mallin päälle ja pyrkiä käyttämään tasaisen kokoisia neliö polygoneja. Luonnollisesti joihinkin kohtiin tulee enemmän yksityiskohtia ja tiheämpää geometriaa kuin toisiin, esimerkkinä kasvot verrattuna selkään, mutta tarkoituksena on kattaa suuret yleismuodot, joilla on toiminnallisuutta animaatiovaiheessa. Tästä esimerkkinä, silmäluomien pinta täytyy huomioida, koska ne liikkuvat, mutta kynsien yksityiskohdat voidaan sivuuttaa, koska ne eivät liiku. Topologian olisi myös hyvä myötäillä oikeita lihaksia esimerkiksi kasvoissa. Silmien ja suun ympärillä on monia pieniä lihaksia, jotka avaavat ja sulkevat niitä. Siksi topologiassa ne on hyvä ympäröidä polygonien muodostamilla kaa-revilla muodoilla, jotka animaatioissa vastaavat lihasten liikettä ja lopputulos näyttää luonnolliselta. Kun koko hahmo on retopologioitu tallennetaan se ja siirrytään tekstuurikarttojen valmistukseen. (Flipped Normals 2018, viitattu 20.3.2019).



KUVIO 31. Retopologia-työvaiheen alkumetri, kuvattuna Topogun-ohjelmassa.



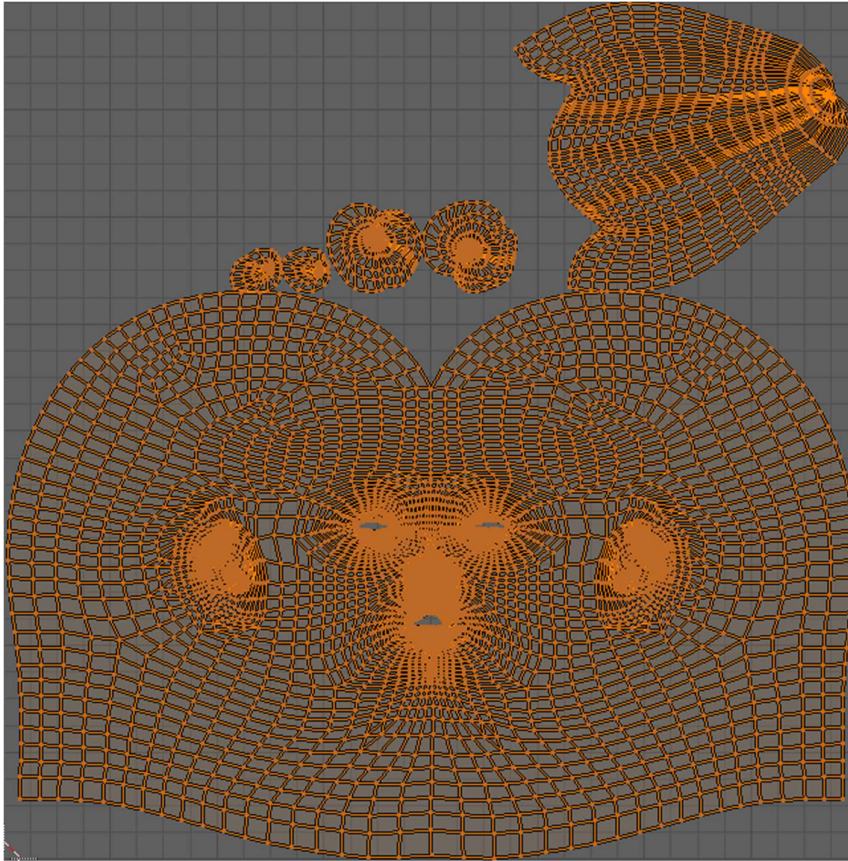
KUVIO 32. Valmis retopologia, kuvattuna Maya-ohjelmassa.

6.5 UV-kartat

Tekstuurikarttojen valmistusta varten hahmo tarvitsee niin sanotut UV-koordinaatit, eli tiedon missä kohti mallia mikäkin tekstuurin kohta esitetään. UV-koordinaatit luodaan 3D-ohjelmassa, tässä tapauksessa Blenderissä. Prosessi tunnetaan nimellä ”UV unwrapping”, koska siinä 3D-mallille määritellään saumat, joiden mukaan malli puretaan 2D-muotoon. Toimenpide vaatii jonkin verran suunnittelua, koska se vaikuttaa oleellisesti hahmon lopulliseen ulkonäköön. Saumojen paikat tulee valita siten, että ne ovat mahdollisimman piilossa, sillä tietyissä tekstuureissa sauma saattaa näkyä virheenä pintakuvion jatkumossa.

Toinen tärkeä aspekti UV-karttojen luonnissa on jälleen resoluutio. Koska 2D-tekstuurit ovat tiedostokokonsa ja määränsä takia potentiaalisesti suuria muistisyöppöjä pelimoottorille, on niiden kokoa ja lukumäärää harkittava. Resoluutio on myös hyvä pitää kahdella jaollisena muistinkäytön optimoimiseksi. Jos koko hahmo määritetään käyttämään vain yhtä UV-karttaa, täytyy sen resoluution olla vastaavasti suurempi, jotta hahmo ei näytä epäselvältä ja suttuiselta, mutta suuri resoluutio tuottaa ison tiedostokoon. Jos taas hahmo puretaan useaan osaan ja UV-karttaan, saa jokainen osa enemmän resoluutiota käytettäväkseen, mutta tuottaa taas enemmän kuvatiedostoja pelimoottorin käsiteltäväksi. Tärkeää on löytää hyvä tasapaino projektikohtaisesti. Esimerkiksi Uncharted 4 -pelin päähahmossa on käytetty vain 1024x1024 resoluution karttoja, mutta taas karttojen määrää ei ole tiedossa (Playstation 2014, viitattu 5.5.2019).

Tässä työssä käytettiin 2048x2048, ns. 2K resoluutiota ja itse hahmo jaettiin kahteen karttaan, jossa pää oli yhdellä kartalla ja vartalo toisella. Tämän lisäksi eri lisävarusteet, kuten takki oli omalla 2K resoluution kartalla ja ase ja kotelo toisella 1K resoluution kartalla. Hiukset, ripset ja kulmakarvat olivat omalla 2K kartallaan. Niin sanottu texel density, eli karttakokojen suhde toisiinsa ja ruudulla näytettäviin pikseleihin, on myös syytä pitää mielessä (Leonardo Iezzi 2017, viitattu 5.5.2019). Ideaalisti, kaikki kartat olisivat kokonsa puolesta suhteessa toisiinsa, eikä missään kohdassa esiintyisi venymistä. Hahmon kasvoissa erityisesti nenä oli haastava kohta, sillä sen ulkonevuuden vuoksi tekstuurissa esiintyi hieman venymistä. Tähän auttoi minimize stretch -toiminto, joka optimoi karttaa venymisen vähentämiseksi.



KUVIO 33. Pään UV-kartta, jossa yllä suun sisäosa ja alla kasvot.

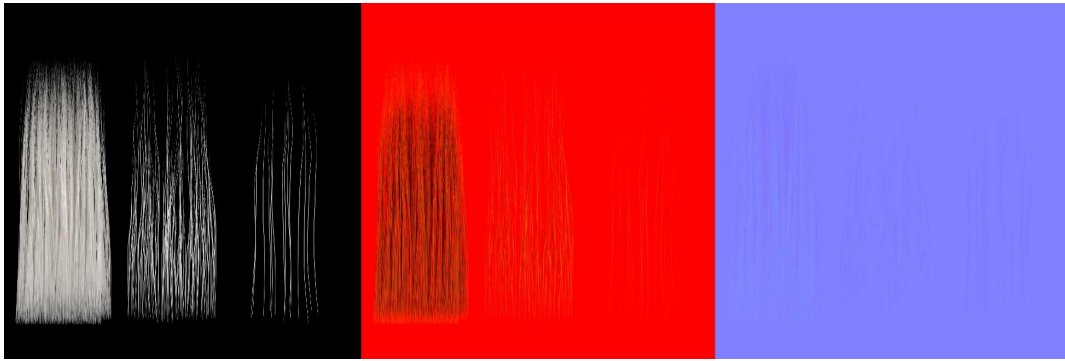
6.6 Hiukset

3D-pelihahmon hiukset ovat usein yksi haastavimmista kohdista tuotantolinjastossa. Joissain tapauksissa hahmon hiukset mallinnetaan samalla tavalla kuin muutkin osat, jolloin lopputulos on enemmän tyylielty, lowpoly lähestymistapa kuin tässä tapauksessa haettu realistinen tyyli. Realistisen tyylin hiukset toteutetaan luomalla ensin hiustekstuurit ja kartoittamalla ne sitten vastaavan kokoiseen "hiuskortteihin" (hair cards). Nämä kortit asetellaan hahmon päähän kerroksittain, halutun hiustyylin mukaiseen muotoon, ja koska hiustekstuureissa on läpinäkyvä alfa -kanava, saadaan renderoinnissa luotua illuusio yksittäisistä hiussuortuvista. Täten lopputulos näyttää aidommalta kuin jos hiukset olisivat yksi mesh-objekti.

Hiuskorttien asettelu on erittäin aikaa vievä vaihe. On olemassa joitain aputyökaluja, kuten esimerkiksi Blenderin Hair tool-lisäosa, jotka automatisoivat prosessia, mutta usein koetaan, että parhaimman tuloksen saa asettelemalla kortit manuaalisesti yksi kerrallaan (Bartosz Styperek 2018, viitattu 5.5.2019).

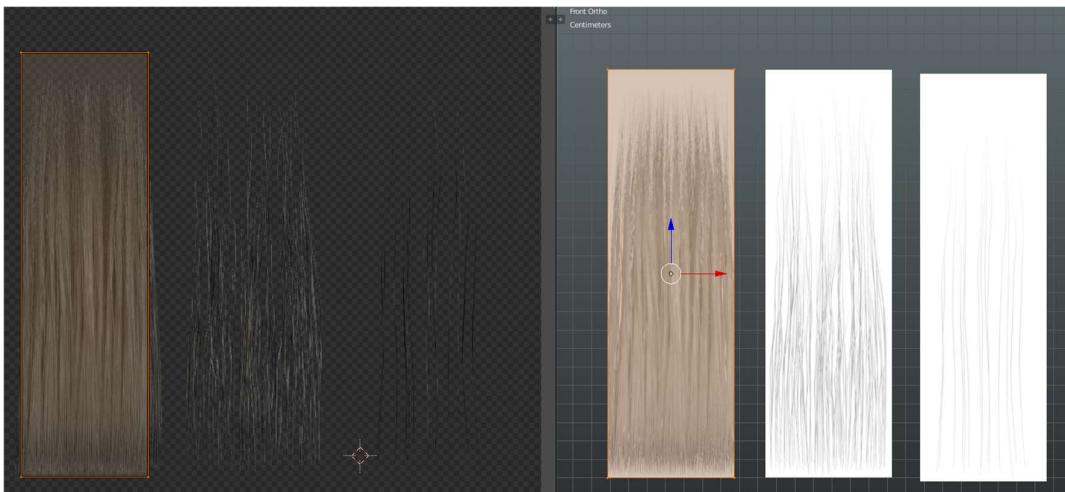
Hiustekstuurit voidaan luoda esimerkiksi piirtämällä ne Photoshopissa, mutta yleensä käytetään Mayan XGen-lisäosaa, jolla luodaan 3D-hiukset halutuin parametrein ja valmistetaan niistä kaikki tarvittavat tekstuurikartat asettamalla kamera kuvaamaan hiuksia kohtisuoraan ortografisessa-tilassa. Hiuksia varten tarvitaan vähintään yksi alfa -kanavan omaava kartta, mutta hyvän lopputuloksen saamiseksi käytetään viittä, tai useampaa karttaa. Tarvitaan diffuse-kartta, joka määrittää hiuksen värin, alfa-kartta, joka määrittää hiusten läpinäkyvyyden, ambient occlusion-kartta määrittää hiusten väliin jäävät varjot, specular-kartta määrittää millaisena valo heijastuu hiusten pinnasta ja normal-kartta luo taas illuusion syvyydestä. (Vadim Sorici 2018, viitattu 5.5.2019.) Unreal engineen dokumentaatioissa käsitellään vielä lisäkartoja, joita voi halutessaan hyödyntää realistisen tuloksen saamiseksi (Unreal Engine 2019, viitattu 5.5.2019).

Työtä varten luotiin XGenillä yllämainitut viisi hiustekstuuria ja alfa-kartta lisättiin diffuse-kartan alfa-kanavaksi ja specular- ja ambient occlusion-kartat yhdistettiin saman kuvatiedoston vihreäksi ja punaiseksi kanavaksi. Kartat, joissa on vain harmaasävytietoa, voidaan yhdistää yhdeksi tiedostoksi hyödyntäen kuvatiedoston punaista-, vihreää-, sinistä- ja alfa-kanavaa. Tällöin vähennetään pelimoottorin taakkaa, kun voidaan lukea muistista vain yksi tiedosto, kahden, kolmen tai jopa neljän tiedoston sijaan. (Polycount 2018, viitattu 5.5.2019). Kaikki kartat tallennettiin png -formaattiin.



KUVIO 34. Diffuse-värikartta, jossa mukana alfanava. Yhdistetty specular- ja ambient occlusion-kartta sekä normal-kartta.

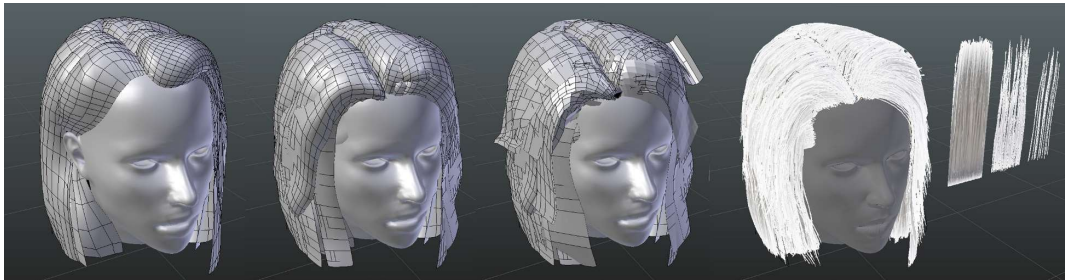
Itse hiuskortit luodaan 3D-ohjelmassa aluksi mahdollisimman yksikertaisena geometriana, jota voidaan prosessin aikana optimoida. Tämä tarkoittaa sitä, että hiuskorttiin lisätään aluksi vain niin monta polygonia kuin mitä sen asettelu haluttuun muotoon vaatii. Myöhemmin kortteihin voi lisätä polygoneja, jos ne näyttävät liian kulmikkailta. Hyvä tekniikka on myös käyttää smooth-työkalua, joka tasoittaa vertexien etäisyyttä toisiinsa ja luo sulavamman muodon objektille. Ennen asettelua hiuskortti UV-kartoitetaan niin, että XGenissä luotu tekstuuri täsmää kortin kokoon.



KUVIO 35. Vasemmalla hiustekstuuri ja UV-kartta. Oikealla vastaava hiuskortti polygoni.

Kortti, jossa tekstuuri on tihein, laitetaan ensin alimmaiseksi kerrokseksi, joka peittää hahmon kal- lon. Tämä kerros määrittää hiusten muodon, tai "volyymin". Seuraavan kerroksen tekstuuri ei ole niin tiheä, eli sen lävitse voi nähdä hieman ensimmäistä kerrosta. Tämän kerroksen tarkoitus on elävöittää hiusten muotoa ja tehdä hiuksista luonnollisemman näköiset. Kolmas kerros on hyvin

ohut, ns. flyaways-kerros, joka luo hiuksiin luonnollista sekaisuutta ja irtohiuksia. Lopuksi kortit yhdistetään kerroksittain ja tuodaan osaksi koko mallia.



KUVIO 36. Hiuskortit kerroksittain.

7 MATERIAALIT JA TEKSTUROIINTI

7.1 Fyysisiin ominaisuuksiin perustuva renderointi

Valon ja eri materiaalien fyysisiin ominaisuuksiin perustuva renderointi eli PBR (physically based rendering) on nykyään standardi 3D-tuotannoissa, sillä se tuottaa tuloksia, jotka vastaavat tosielämän objektien pintoja ja näyttää siten erittäin realistiselta. Se on luotettava metodi, joka tuottaa varmoja tuloksia valaistusolosuhteista riippumatta, sillä sen algoritmit perustuvat fysiikan lakeihin. (McDermott 2019, 45.)

Myös tässä työssä käytettiin PBR -lähestymistapaa yhdessä Unreal Enginen käyttämän ns. metallic/roughness -työtavan kanssa. Metallic ja roughness-arvot määrittävät onko materiaali metallia vai ei ja kuinka kiiltävä tai matta sen pinta on. Näiden lisäksi tarvitaan specular- ja väriarvot, jotta voidaan luoda realistinen PBR -materiaali. Väriarvo luonnollisesti määrittää materiaalin värin, kun taas specular määrittää valon heijastumista materiaalin pinnasta. (Unreal Engine 2019, viitattu 5.5.2019.)

7.2 Tekstuurikartat ja niiden valmistus

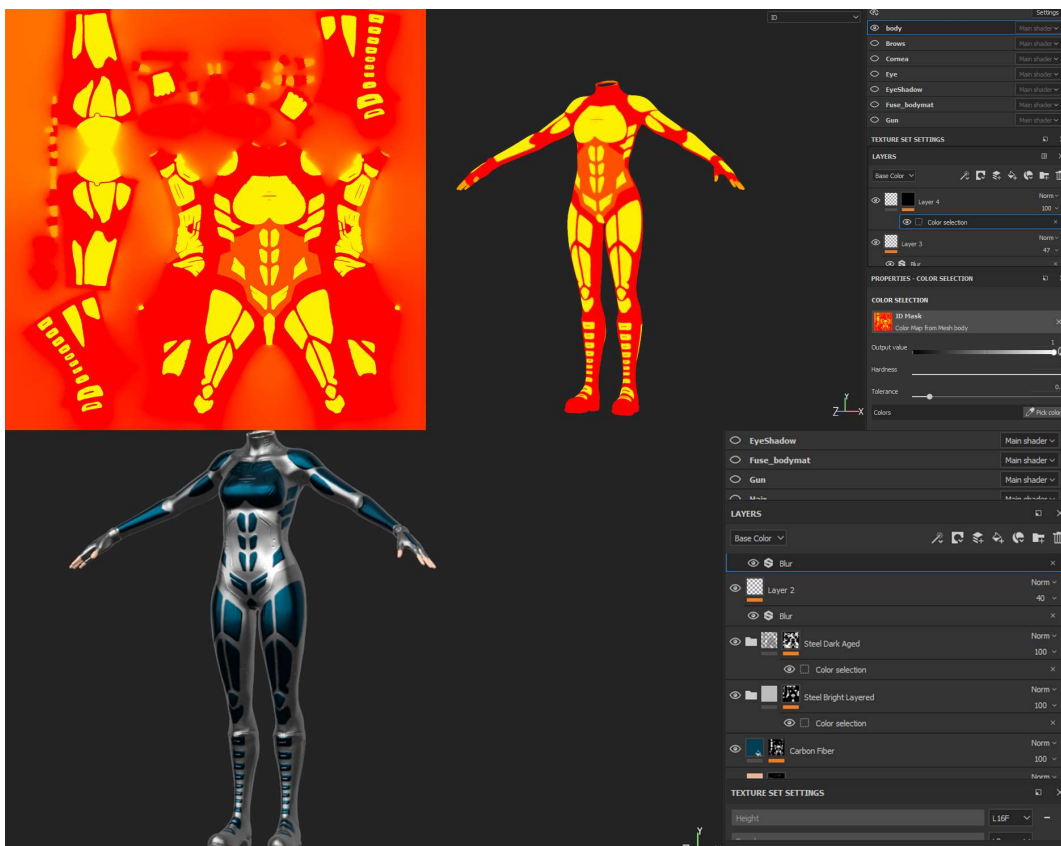
Ensimmäinen kartta, josta kannattaa aloittaa on normal-kartta, joka luo illuusion suuremmasta resoluutiosta kuin mitä objekti todellisuudessa on. Huomaa, että vaikka normal-kartan voi luoda usealla eri ohjelmalla, voi kartta tuottaa eri tuloksia riippuen onko se luotu Direct X vai OpenGL formaattilla. Näiden kahden ero on yksinkertaisesti vihreän kanavan käänteisyys, jonka tarpeen vaatiessa voi usein kääntää renderointi-ohjelmassa. Lähtökohtaisesti Unreal Engine ja työssä käytetty teksturointi -ohjelma Substance Painter käyttävät Direct X -formaattia, kun taas esimerkiksi Marmoset Toolbag ja Unity käyttävät OpenGL-formaattia. (Vincent Gault 2016, viitattu 5.5.2019.)

Normal-kartta luodaan antamalla ohjelmalle sekä highpoly-veistos, että retopologioitu lowpoly -malli ja ohjelma luo kartan perustuen näiden mallien eroavaisuuksiin ja asettaa ne sitten lowpoly-mallille luotuun UV-karttaan. Kokeilin aluksi xNormal-ohjelmaa tähän, mutta se ei tuottanut tällä kertaa hyviä tuloksia, vaan aiheutti mallin saumoihin virheitä. Kokeilin myös valmistaa normal-kar-

tan Marmoset Toolbag:ssa, joka tuotti virheettömän kartan. Lopulta päädyin kuitenkin valmistamaan kaikki kartat Substance Painter -ohjelmassa, koska se loi automaattisesti Direct X -formaatin normal-kartan ja pakkasi myös muut kartat Unreal Engineelle sopivaan muotoon, eli yhdisti ambient occlusion, roughness ja metallic -arvot yhden kuvatiedoston punaiselle, vihreälle ja siniselle kanavalle.

7.3 Väri ID:t

Väri ID -kartta valmistetaan highpoly-malliin määritetyistä polypaint-arvoista ja sitä hyödynnetään teksturoinnissa osoittamaan mihin kohti mallia, mikäin materiaali kuuluu. Substance Painter osaa lukea ZBrushista tulevat arvot ja luoda väri ID-kartan. Itse kartta on siis vain mallin UV-kartta, jonka eri osat ovat jaettu satunnaisiin väreihin.

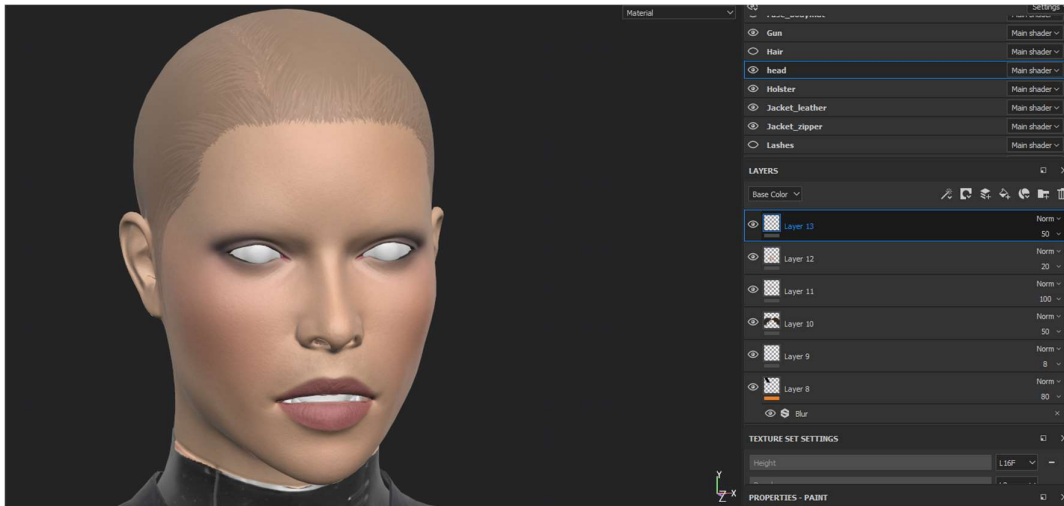


KUVIO 37. Väri ID-kartan avulla voi teksturoida yksityiskohtia, joita lowpoly mallissa ei ilman normal-karttaa edes ole.

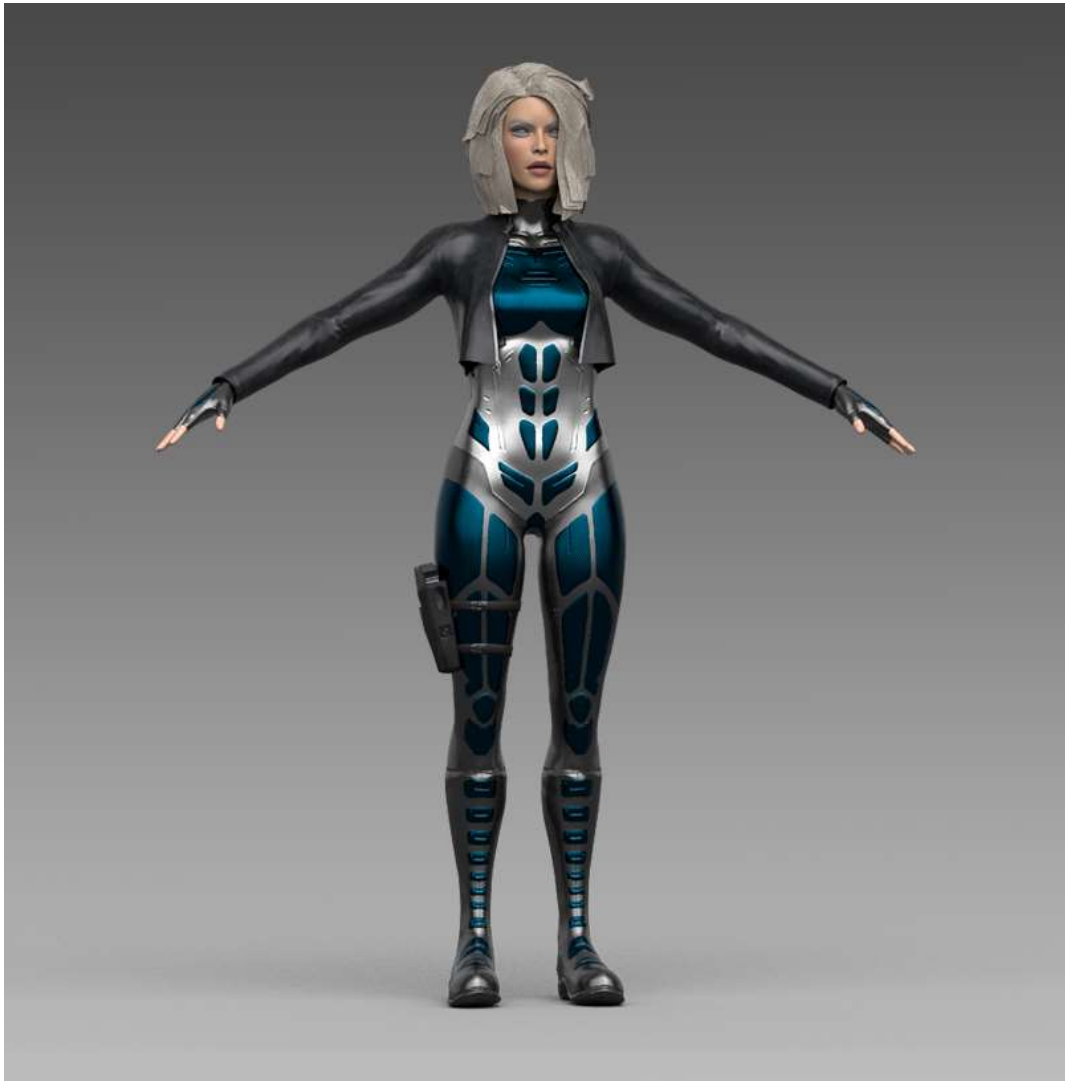
Jokainen väri ID on oma yksilöllinen värinsä ja jokainen väri vastaa omaa materiaaliensa mallissa. Kuviossa 37. punainen väri vastaa mallin pohjamateriaalia, keltainen panssarimateriaalia, oranssi on vaaleampi variaatio pohjamateriaalista ja ruskea ID on sormien ihoa.

7.4 Materiaalit

Substance Painter -ohjelmassa luodaan hahmolle tekstuurit ja materiaalit ja siinä on valmiina pieni perusmateriaalien kirjasto, joka toimii hyvänä pohjana. Esimerkiksi, hahmon kasvojen pohjana on valmis iho -materiaali, jonka päälle on maalattu useita kerroksia hienovaraisia värimuutoksia ja lopuksi myös hiusraja ja meikki. Tässä vaiheessa voisi halutessaan käyttää esimerkiksi TexturingXYZ-sivuston korkealaatuisia iho -tekstuureja, joita voi heijastaa mallin päälle projection-työkalulla (Texturingxyz 2019, viitattu 5.5.2019). Tässä työssä ei käytetty TexturingXYZ-tekstuureja, koska yksi kasvosetti maksaa 39,90 dollaria, joka todettiin liian kalliiksi.

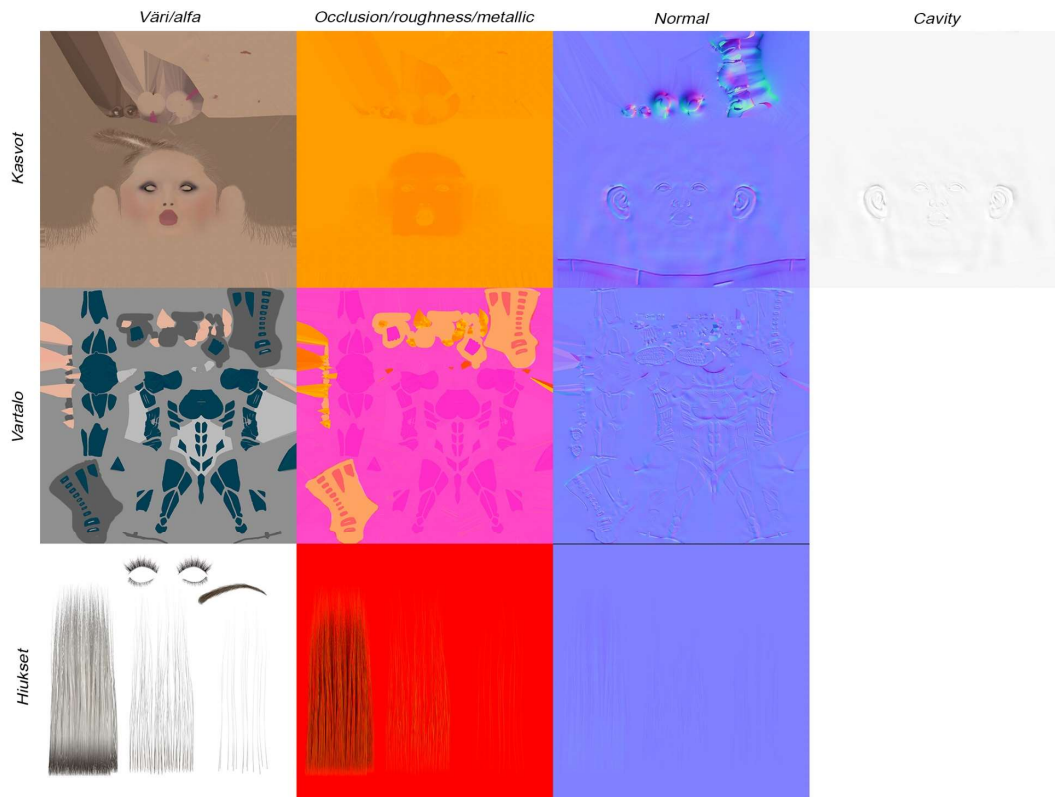


KUVIO 38. Kasvojen materiaali koostuu pohjamateriaalista ja kolmestatoista eri kerroksesta.



KUVIO 39. Lopulliset materiaalit Substance Painterista. Hiustekstuurit tehtiin XGenillä ja kulmakarvat sekä ripset piirrettiin Photoshopissa.

Substancen työkaluilla voi luoda hyvinkin monimutkaisia materiaaleja, mutta koska hahmon luonteeseen sopi tietty selkeys ja puhtaus, huomioitiin se myös materiaaleissa. Erilaiset materiaalit luovat lopulta erilaisia osia kuhunkin karttaan. Vaikka vartalon UV -karttaan kuului metallisten osien lisäksi myös ihoa, hoitavat metalness- ja roughness-kartat kunkin materiaalin ominaisuuksista, jotta ne näyttävät lopulta oikeilta. Kun kaikki halutut materiaalit olivat paikallaan, tallennettiin ne Unreal Engine 4 -esiasetusta käyttäen, jolloin ne pakattiin kaikkia värikanavia hyödyntäen ja optimoitiin pelimoottoria varten.

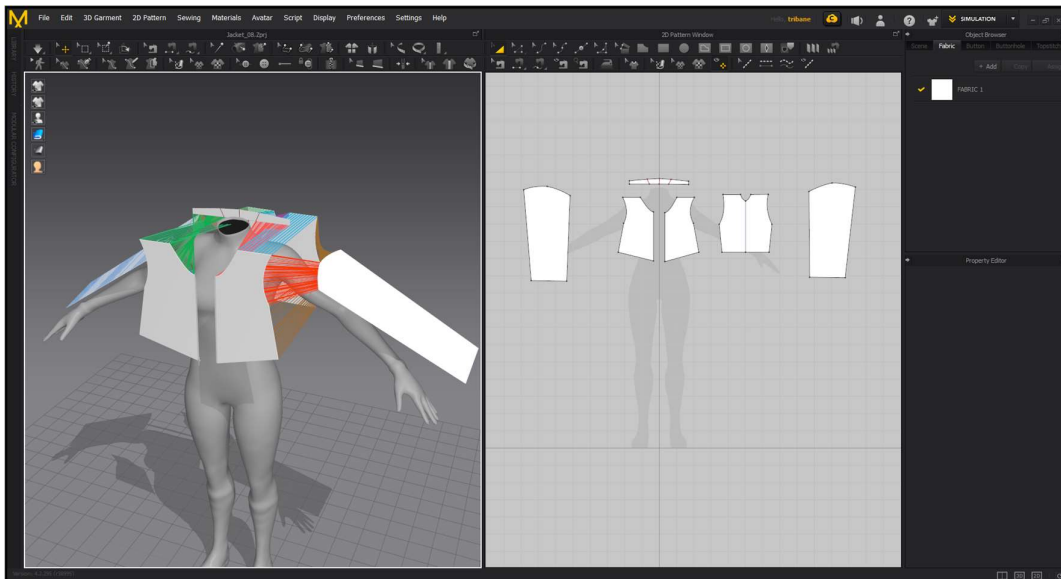


KUVIO 40. Valmiit tekstuurikartat. Väri ja alfa-kartat pakattuna yhteen, kolme harmaasävy-karttaa pakattuna eri värikanaville, normal-kartta ja kasvojen kohdalla vielä cavity-kartta, jonka voi luoda esimerkiksi xNormal-ohjelmalla.

8 SIMULAATIOT

8.1 Vaatetuksen luonti

Vaatteet ja kankaat ovat haastava osa 3D-mallinnusta laskosten ja ryppyjen vuoksi, mutta tähän erikoistuneen Marvelous Designer -ohjelman avulla on helpompi tehdä uskottava vaatetus. Oma hahmo tuodaan ohjelmaan, jonka jälkeen halutut vaatekappaleet piirretään ensin 2D-kaavoina, aivan kuten oikeatkin vaatteet. Sen jälkeen määritetään mistä kohtaa osat ommellaan yhteen ja halutaanko esimerkiksi vetoketjuja tai nappeja.



KUVIO 41. Takin kaaviot valmiina simulaatioon.

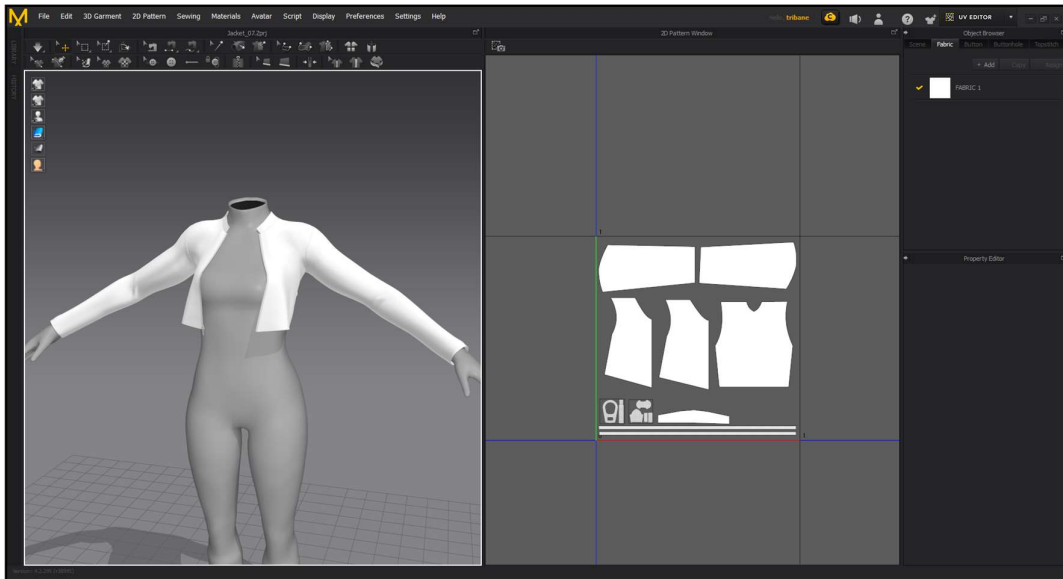
Kaavioita ja ompeleita voi muokata missä tahansa vaiheessa prosessia ja usein ensimmäiset simulaatiot paljastavat ongelmakohtia. Jokin kohta voi olla liian kireä ja lyhyt, jolloin sitä pitää suurentaa kaavassa. Lopulta kaavan osat muodostavat UV-kartan vaateobjektille.

8.2 Fysiikkasimulaatio

Kun vaatekappaleen kaavat ovat valmiina ja ompeleet määritetty, voidaan ajaa simulaatio, joka liittää osat toisiinsa ja ripustaa vaatteen hahmon ylle. Tämän jälkeen vaatetta voi vetää halutusta kohdasta, jos jostain tulos ei miellytä ja simulaatio päivittyy reaaliajassa muutokseen. Vaatteille voi

myös antaa erilaisia fyysisiä ominaisuuksia, jotka vaikuttavat muun muassa sen painoon ja rypistymiseen. Lopullista simulaatiota varten on myös hyvä nostaa simulaation laatua, joka tapahtuu laskemalla particle distance -arvoa.

Tätä hahmoa varten luotiin yksinkertainen ja selkeälinjainen nahkatakki, mutta prosessi on sama muunlaisillekin vaatteille. Kun vaate oli simuloitu hahmon ylle siten, että se istui hyvin ja näytti hyvältä, varmistettiin että jokaisella osalla oli sama particle distance, joka vastaa objektin resoluutiota, kuten high- ja lowpoly -malleissa. Koska simulaatio luo sotkuisen topologian, myös tämä objekti pitäisi perinteisesti retopologoida, mutta tässä työssä hyödynsin designerin omaa, uutta ominaisuutta nimeltä remesher, joka konvertoi meshin siistiin quad -muotoon automaattisesti ja toimi tässä tapauksessa riittävän hyvin. Lopuksi kaavion osat järjestettiin UV-kartalle ja objekti tallennettiin obj-muotoon, jonka jälkeen se siirrettiin ZBrushiin. Veisto-ohjelmassa takin pintaan tehtiin lisää yksityiskohtia ja laskoksia, jotka sitten muunnettiin normal-kartaksi Substance Painterissa, noudattaen samaa tekniikkaa kuin varsinaisen hahmonkin kohdalla.

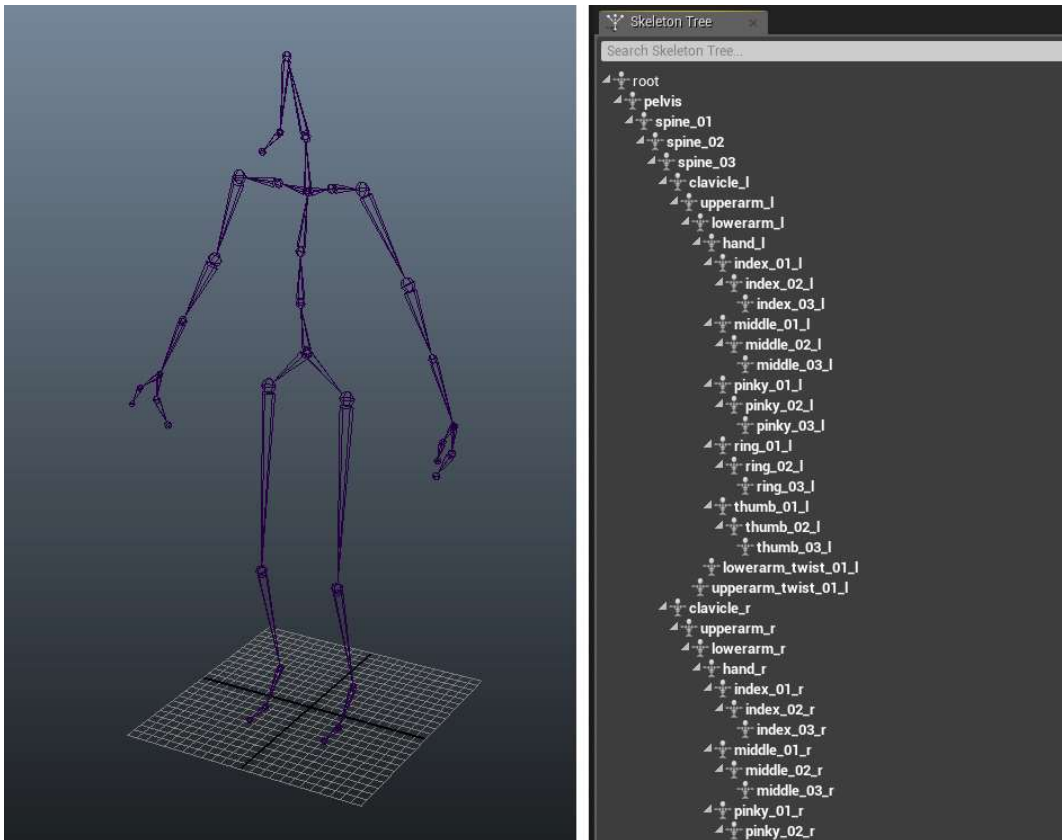


KUVIO 42. Valmis takki ja kaaviot UV-tilaan järjestettynä.

9 HAHMON LUUSTON LUONTI

9.1 Manuaalinen luonti

Luuston luonti, eli "rigaus" (rigging), tehdään 3D-ohjelmassa, kuten Mayassa tai Blenderissä. Tätä varten tuodaan kaikki hahmon erilliset osat ohjelmaan ja asetetaan hahmon ylle ja tarpeen vaatiessa tehdään vielä viimeiset muokkaukset ennen meshien sitomista luustoon. Luuston luonti aloitetaan ortografisessa etunäkymässä, jossa luut on helppo asetella suunnilleen oikeisiin kohtiin meshiä ja sitten viimeistellä perspektiivinäkymässä. Kun hahmon kohteena on pelimoottori, kuten Unreal Engine, on hyvä selvittää paras yhteensopivuus luuston osalta. Hahmon käyttö pelimoottorissa on sujuvinta silloin, kun se vastaa moottorin odotuksia, eli sillä on vähintään tietty määrä luita ja ne on nimetty oikein. Toki erilaisiakin luurankoja voi käyttää, mutta ne vaativat aina lisätyötä toimiakseen.



KUVIO 43. Vasemmalla Unreal Enginen standardilla Mayassa luotu luusto ja oikealla Unreal Enginessä luiden nimeämiskäytäntö (Unreal Engine 2019, viitattu 5.5.2019).

Jotkin hahmot vaativat enemmän luita kuin mitä Unreal engine suosittelee. Esimerkiksi, tässä työssä tehtävä realistinen hahmo tarvitsi kasvon luut, kasvoanimaatioita varten ja niiden lisääminen ei tuota ongelmia pelimoottorille. Myöskin, jos hahmolla olisi esimerkiksi pitkä poninhäntä kampaus, voitaisiin hiuksiinkin lisätä luita, joiden avulla sitä voisi animoida.

9.2 Auto-Rig Pro

Tämän työn hahmo rigattiin Blenderin Auto-Rig Pro -lisäosalla, joka on maksullinen, mutta nopeuttaa prosessia huomattavasti, sillä se luo luuston automaattisesti, kunhan sille määritetään missä hahmon kaula, leuka, häntäluu, olkapäät, ranteet ja nilkat ovat. Samalla periaatteella se pystyy nopeasti luomaan myös kasvojen luut ja se automaattisesti nimeää kaikki luut halutun, tässä tapauksessa Unreal Enginen, standardin mukaisesti (Blender market 2017, viitattu 5.5.2019).

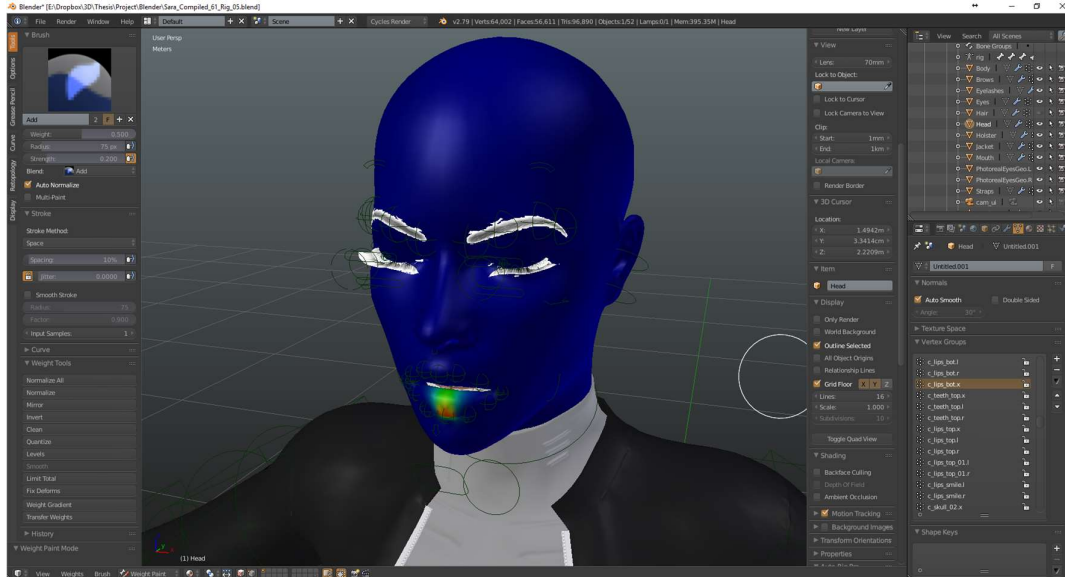
Halutessaan voi myös tähän luurankoon lisätä uusia luita, kuten esimerkiksi aiemmin mainitut hiusluut, mutta tämän hahmon tapauksessa sitä ei katsottu tarpeelliseksi. Kun kaikki luut ovat paikallaan, valitaan kaikki hahmon osat ja lopuksi luuranko ja sidotaan kaikki osat kiinni luurankoon (skinning).

9.3 Painoarvojen muokkaus

Jokaisella luulla on tietty vaikutusalue, painoarvo vertex-pisteisiin nähden. Nämä arvot voidaan luoda automaattisesti tai manuaalisesti, mutta usein niitä joudutaan hienosäätämään käsin. Tätä prosessia kutsutaan weight paintingiksi. Jos animaattori liikuttaa esimerkiksi hahmon kättä, tulee kyseisen luun vaikutusalue ylettyä vain meshin käteen, eikä esimerkiksi rintaan asti. Tarvittaessa meshiin maalataan uusia, tai korjattuja vaikutusalueita. Haastavan tästä tekee se, että moni kohta meshistä voi kuulua useamman kuin yhden luun vaikutusalueeseen ja niiden välillä täytyy löytää tasapaino, jotta hahmo liikkuisi luonnollisesti.

Luuston sidonta loi hahmon jokaiselle objektille vertex-ryhmät, jotka vastaavat jokaista luuta. Näitä ryhmiä on syytä käydä läpi ja tarkistaa, että painoarvot ovat oikein ja korjata tarvittaessa. Halutessaan voi tarpeettomia ryhmiä jopa poistaa, kuten esimerkiksi pään objekti ei kaipaa jalkojen vertex-

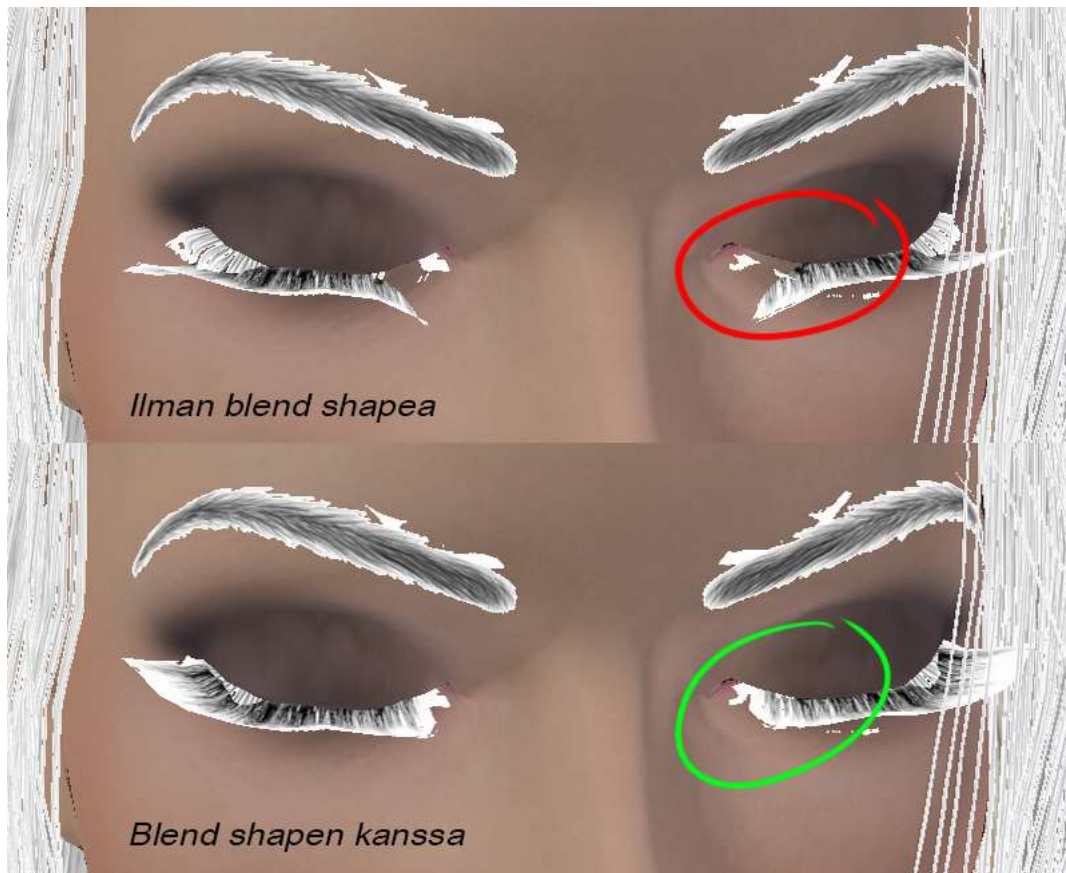
ryhmiä, mutta on mahdollista, että automaattinen painotus on arvioinut jotain väärin ja antanut jollekin jalan luulle painoarvoa kasvoihin. Tämä ei tietenkään ole haluttua, joten turhien ryhmien poistaminen saattaa ratkaista useita ongelmakohtia.



KUVIO 44. Weight painting -prosessi. Alahuulen yksi luu ja sen vaikutusalue. Sininen alue on neutraali, kun taas punainen on maksimi painoarvo.

9.4 Korjaavat muodot

Korjaavat muodot tunnetaan yleensä nimellä blend shapes ja Unreal Engine:ssä taas nimellä morph targets. Näillä tarkoitetaan vaihtoehtoista tai luustoa tukevaa animointitekniikkaa, jossa meshiä animoidaan portaattomasti kahden tai useamman, ennalta asetetun muodon välillä. Alkuperäisestä objektista voidaan tehdä kopio, joka sitten muovataan uuteen, haluttuun muotoon ja tämä tieto syötetään alkuperäiselle objektille, joka pystyy sen myötä muuntumaan uuteen muotoon. On tärkeää, ettei tässä vaiheessa meshiin enää lisätä, tai siitä poisteta vertex-pisteitä, sillä pisteillä on tietty järjestys ja omat koordinaattinsa ja jos tämä rakenne rikkoutuu, eivät blend shapet enää toimi. Blend shapet ovat erityisesti hyödyllisiä esimerkiksi kasvoanimaatioissa, jossa luilla on vaikea muodostaa useaa, luonnollista ilmettä. Blend shapet voivat toimia itsenäisesti erillään luurangosta tai myös luiden ohjaamana.



KUVIO 45. Hahmon ripset muovautuivat blend shapen avulla, silmäluomen luun ohjaamana.



KUVIO 46. Hahmon hymy on itsenäinen blend shape.

Unreal Engine tukee blend shapeja, mutta kutsuu niitä sisäisesti morph targeteiksi, mutta kyseessä on yksi ja sama asia. Kun hahmo tuodaan pelimoottoriin, täytyy valita morph target tuki hyväksytyksi asetuksissa. Myös materiaalit vaativat saman tuki hyväksynnän, mutta sitten nämä muodot toimivat Unrealissa täysin samoin kuin Blenderissä tai Mayassa.

9.5 Luiden kontrollit

Kun luita on useita ja joissain kohdissa, kuten kasvoissa, ne ovat hyvin lähekkäin, on niiden käyttäminen ja kontrollointi animoidessa hankalaa. Siksi luille luodaan yleensä omat kontrollit, jotka ohjaavat yhtä tai useampaa luuta.



KUVIO 47. Luiden kontrollimuotoja.

Luille voidaan myös asettaa rajoituksia, kuten mihin suuntaa ne voivat kääntyä tai liikkua, ja kuinka paljon. Tämä helpottaa animointia, sillä luut voidaan rajata käyttäytymään kuten oikeat luut, joiden luonnollinen liikerata on rajallinen. Esimerkiksi, hahmon silmäluomille asetettiin rajoitukset, kuinka paljon ne voivat aueta, tai kuinka tiukkaan ne voi sulkea.

9.6 Mixamo

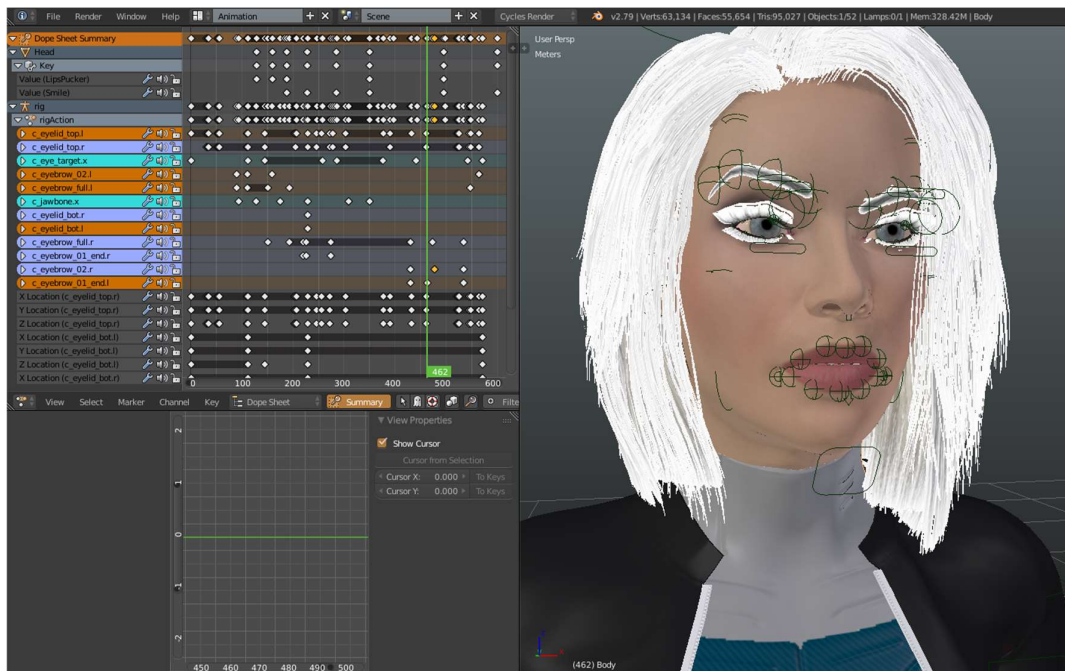
Adobe Mixamo-palvelu on automaattinen luuston luonti -sivusto internetissä, joka ei aina tuota korkealaatuisia tuloksia, mutta on oivallinen työkalu esimerkiksi nopeaan prototyypitestaukseen tai aloittelevalle animaattorille. Hahmo on helppo ladata palveluun ja luoda hyvin nopeasti perusluusto hahmolle, jonka jälkeen sen voi jälleen ladata omalle koneelle ja tuoda pelimoottoriin (Mixamo 2019, viitattu 5.5.2019). Mixamon luoma luuranko ei tosin vastaa esimerkiksi Unreal Enginen suosittelemaa mallia tai nimeämiskäytäntöä, joten se vaatii hieman ylimääräistä työtä ennen kuin sen voi laittaa peliin. Uusi luuranko tulee kohdentaa (retarget) Unrealin käyttämään luurankoon, jolloin Unreal luo rigin, joka toimii eräänlaisena tulkkina kahden luurangon välillä (Unreal Engine 2019, viitattu 5.5.2019).

Toinen Mixamo -palvelun hyöty, on sen suuri animaatiokirjasto. Palvelusta löytyy lukuisia hyödyllisiä perusanimaatioita, joita voi hyödyntää esimerkiksi testaus- ja prototyypivaiheissa. Tässä työssä käytettiin kahta Mixamo-animaatiota; kävely- ja juoksuanimaatioita. Oikeassa projektissa tärkeät hahmot saivat omat kustomoidut animaationsa joko motion capture -tekniikalla, animaatioartistin käsityönä tai näiden yhdistelmänä.

10 ANIMAATIOT

Animaatiot ovat erittäin tärkeä osa mitä tahansa hahmoa. Vaikka hahmo näyttäisi kuinka hyvältä, animaatiot antavat sille eloa ja luonnetta. Kun hahmo on rigattu ja luiden vaikutusalueet hiottu, voidaan hahmoa alkaa animoida. Tämä voidaan taas tehdä useammassakin 3D-ohjelmassa, mutta yksi siihen erikoistunut ohjelmisto on Mayan animaatiotyökaluista alun perin kehitetty Motionbuilder (Autodesk 2019, viitattu 6.5.2019).

Tämän työn puitteissa ei animaatioihin perehtymiseen ollut aikaa, mutta ne kuuluvat silti osaksi tuotantolinjaa ja kun hahmo on rigattu, voidaan animaatioita alkaa työstää ja tehdä niitä niin kauan kuin tarve vaatii. Hahmolle tehtiin yksi oma, kustomoitu animaatio. Niin sanottu idle-, eli toimeton animaatio, aktivoituu, kun pelaaja ei liikuta hahmoa ja tämä animaatio tehtiin Blenderissä. Animaatiot luodaan asettamalla aikajanelle niin sanottuja keyframeja, joihin objektien, kuten luiden sijainti ja asento voidaan tallentaa. Sitten siirrytään aikajanalla eteenpäin, muutetaan luun asentoa ja tallennetaan uusi asento keyframeksi, jolloin ohjelma animoi luun, ja sen myötä hahmon, liikkumaan ensimmäisestä asennosta toiseen.



KUVIO 48. Kasvoanimaation luomiseen tarvitaan useita keyframeja, jotka näkyvät pisteinä vasemmassa yläkulmassa.

11 REAALIAIKAINEN RENDEROINTI

Marmoset Toolbag on pelimoottorin tavoin reaaliaikarenderointi-ohjelma, johon oman 3D-mallin voi tuoda, sen ulkonäköä hienosäätää ja lopulta asettaa esille. Marmosetissä ei voi muokata hahmon geometriaa tai topologiaa, mutta sen vahvuus on näyttävissä materiaaleissa, joita on helppo säätää sekä valaistuksessa. Kun ohjelmaan syötetään PBR-tekstuurit, saadaan luotua hyvin realistisia ja uskottavia materiaaleja ja pintoja, kuten iho tai hius. Mallin voi valaista hdr (high dynamic range) kuvalla ja tavallisilla 3D-valoilla ja tehdä esimerkiksi tyylikäs turntable-animaatio, jonka avulla mallia on helppo tarkastella joka suunnasta.

Marmosetia käytettiin tässä työssä niin sanottuun look dev -prosessiin, jossa hioin tekstuurikarttoja Photoshopissa ja palasin Marmosetiin tarkistamaan tulokset suoraan, reaaliajassa hahmon päältä. Tämän lisäksi Unreal Engineen on saatavilla ilmainen lisäosa, joka mahdollistaa Marmoset-projektin tuonnin suoraan Unreal-projektiin, jolloin kaikki Marmosetissa luodut materiaalit saadaan helposti myös Unrealiin. Tosin lisäosa ei tue vielä kaikkia materiaaliominaisuuksia, joten materiaaleista riippuen, joitain korjauksia tulee tehdä vielä Unrealin materiaalieditorissa. Marmosetissa on jopa valmis Unreal Engine-materiaalipohja, joka hyödyntää PBR metallic/roughness-työtappaa.

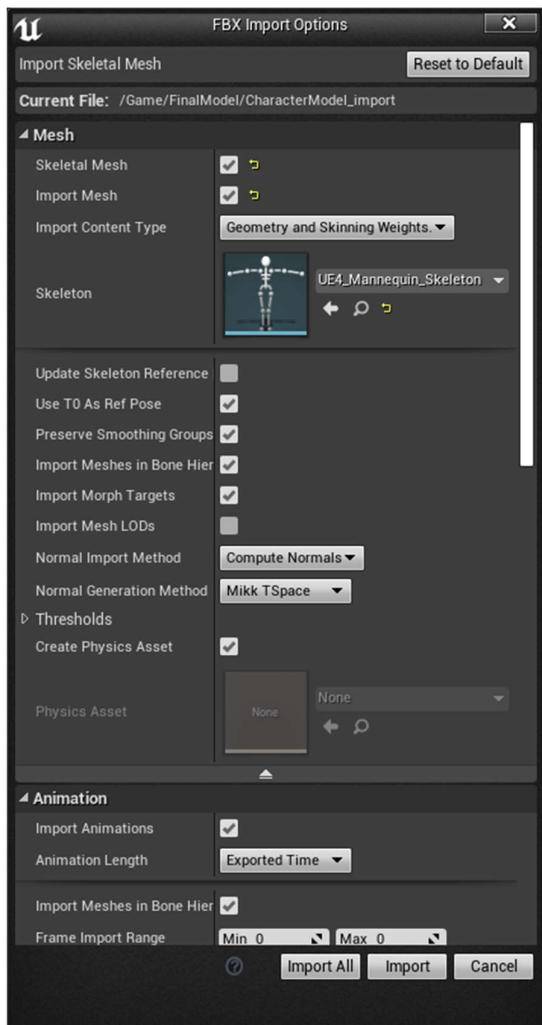


KUVIO 49. Marmosetissa renderoitu hahmo, joka käyttää Marmosetin Unreal-materiaalipohjia.

12 PELIMOOTTORI

12.1 Hahmon tuonti pelimoottoriin

Hahmo voidaan tuoda Unreal Engineen kahdella tavalla; joko suoraan rigauksesta FBX-tiedosto-
muodossa tai Marmoset Toolbag-projektina. Koska Marmoset-lisäosa on vielä kehityksessä, eikä
tue kaikkia ominaisuuksia, on parempi tuoda varsinainen hahmo omana tiedostona pelimoottoriin
ja ottaa Marmosetistä vain materiaalit. Jos hahmon luut on tehty Unrealin suosittelemalla mallilla,
valitaan Import content type-valikosta UE4_Mannequin_Skeleton ja varmistetaan, että import
morph targets on valittu.



KUVIO 50. Skeletal meshin tuonti Unreal Engineen.

Jos hahmon luut eivät kuitenkaan vastaa Unrealin esimerkkiä, jätetään import content type-valikko tyhjäksi, jolloin Unreal luo hahmolle oman luurangon eli skeleton assetin. Tällöin hahmo ei voi suoraan käyttää Unrealin valmiita animaatioita.

12.2 Alustus

Projektipohjaksi tässä tapauksessa toimii hyvin Third person-template, joka luo automaattisesti kolmannen persoonan hahmokontrollerin projektiin, jota voidaan käyttää myös oman hahmon kanssa. Projektissa on myös pieni testitaso, jossa voi juoksennella ympäriinsä, kiivetä portaita ja hypätä alas.

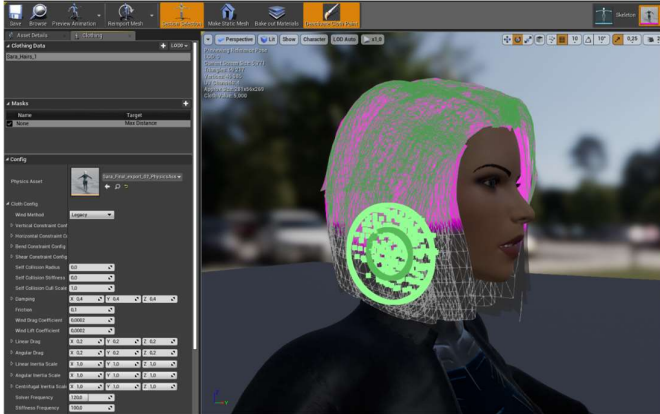
Kun hahmo on tuotu projektiin, pitää se alustaa pelikäyttöön. Jos hahmo ei käytä UE4_Mannequin_Skeletonia pitää sille kenties tehdä retargetin, jossa hahmon luut kohdennetaan vastaamaan oletusarvoisia UE4_Mannequin_Skeletonin luuta. Myös animaatiot pitää retargetoida uudelle luurangolle. Luihin voi myös luoda socketteja, joihin voi kiinnittää hahmon varusteita, kuten tässä tapauksessa hahmon aseeseen. Hahmolle pitää myös antaa oikeat materiaalit, jotka voi joko kasata itse, valmistetuista tekstuurikartoista tai sitten hyödyntää Marmosetista tulleita materiaaleja. Käytin tässä työssä molempia lähestymistapoja, sillä osa materiaaleista näytti paremmalta itse tehtynä ja osa näytti paremmalta Marmosetista tuotuna. Käytettäessä värikanaville pakattuja karttoja, täytyy tekstuurin sRGB ominaisuus kytkeä pois päältä ja asettaa se materiaalieditorissa linear color -tilaan. Materiaalien morph target -ominaisuus pitää myös kytkeä päälle.



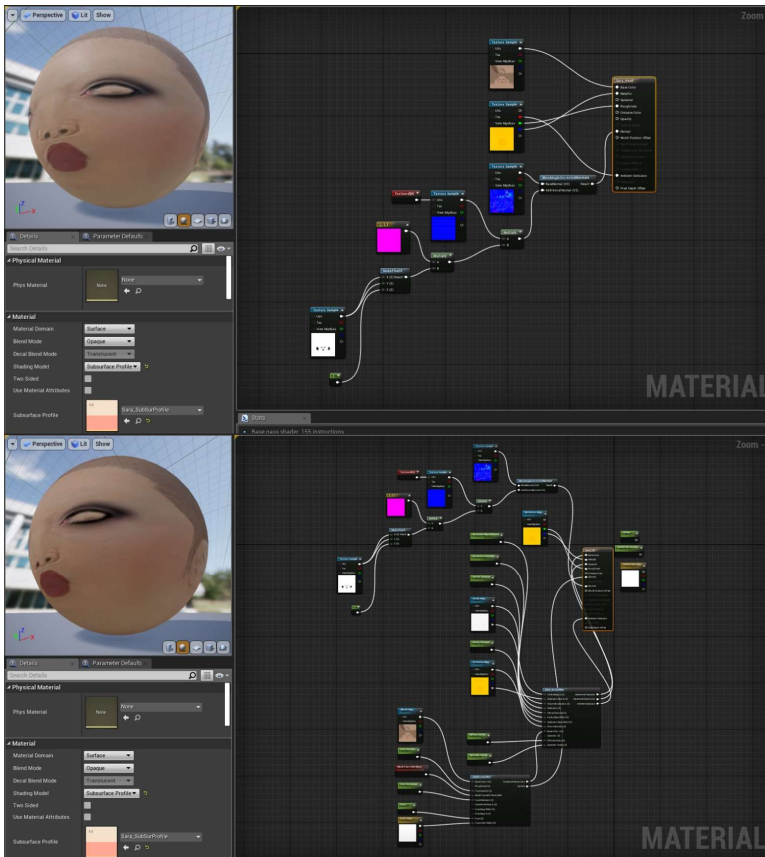
KUVIO 51. Materiaalit on hyvä tarkistaa Unrealin Advanced lighting -kalibrointitasossa.

Hahmon hiuksille tai isommille vaatekappaleille voi asettaa cloth-fysiikat, jolloin ne liikkuvat hahmon liikkuesssa, tai jopa tuulessa. Cloth painting -prosessia varten avataan skeletal mesh -tiedosto

ja valitaan activate cloth paint. Sitten luodaan uusi clothing asset ja aktivoidaan se objektin käyttöön ja sitten haluttu osa objektista maalataan, jolloin maalattu osa alkaa käyttäytyä niin kuin kangas ja asetuksia säätämällä saadaan aikaan erilaisia tuloksia.



KUVIO 52. Cloth paint Unreal Enginessä.



KUVIO 53. Ylhäällä itse tehty ihomateriaali ja alla Marmosetista tuotu ihomateriaali.

12.3 Hahmon kontrollointi

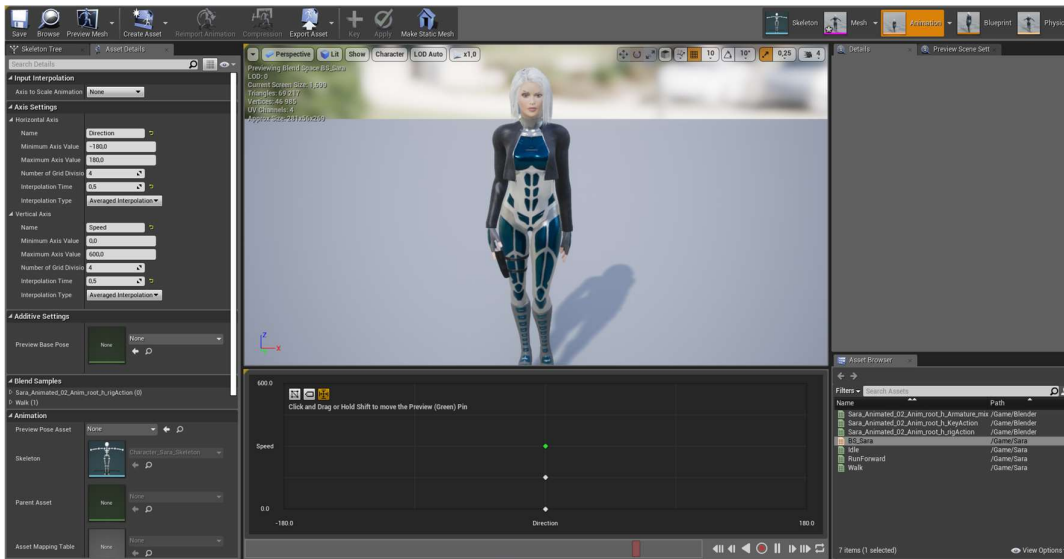
Jos hahmo käyttää UE4_Mannequin_Skeletonia, voidaan avata third person blueprint (ThirdPersonCharacter), mennä viewport välilehdellä kohtaan mesh (inherited) ja vaihtaa Skeletal mesh kohtaan oma hahmo. Kun muutos vahvistetaan compile-napista, pitäisi oman hahmon olla pelattavissa, kun painaa editorin play-nappia.

Hahmolle voi tehdä myös oman animaatio blueprintin, johon voi määritellä omia ominaisuuksia, kuten animaatio blend space-assetin. Tämä blueprint hyödyntää third person character-blueprintiä, josta se saa hahmon nopeuden ja kääntymissuunnan. Se otetaan käyttöön valitsemalla se third person character-blueprintin mesh (inherited) objektin Anim class-valikosta.

12.4 Animaatioiden käyttö

Omia, ja esimerkiksi Mixamosta tuotuja, animaatioita voidaan käyttää esimerkiksi yllämainitussa blend space-assetissa. Blend space vaihtaa sulavasti animaatiosta toiseen, lukien samalla third person character-kontrollerista tulevia hahmon nopeus ja suunta muuttujia.

Blend spaceen voidaan asettaa idle animaatio toistumaan silloin kun hahmon nopeus on nolla ja vaihtaa siitä sujuvasti kävelyanimaatioon ja lopulta juoksuun, hahmon nopeuden noustessa. Suunta-muuttuja voi lisätä animaatioon esimerkiksi käännösaskelia hahmon suunnan muuttuessa.



KUVIO 54. Blend space, jossa Blenderissä tehty idle-animaatio ja Mixamosta tuotu kävelyanimaatio.

Lopulta pelimoottorissa on pelattavissa hahmo, joka on korkealaatuinen ja alusta loppuun asti itse tehty. Se noudattaa PBR-materiaali standardia, omaa täydellisen luuston ja on jo tässä vaiheessa hyvin optimoitu.



KUVIO 55. Valmis hahmo peliympäristössä. Taustan taso on Scifi Hallway Unreal Enginen opetusmateriaaleista.

13 POHDINTA

Työ oli hyvin haastava, koska se käsittää niin monta eri vaihetta ja joissain kohdissa ero onnistumisen ja epäonnistumisen välillä saattoi olla hyvin pienestä muutoksesta kiinni. Usein jokin pieni, mutta oleellinen tiedonjyvänen löytyi sattumalta, kun etsi tietoa jostain toisesta asiasta ja juuri tällaisia yksityiskohtia yritin sisällyttää tähän dokumenttiin. Paljon informaatiota oli myös ripoteltu pitkiin videoihin, josta sitä oli hidasta etsiä. Myös sanasto ja sen käyttö oli oma haasteensa, sillä esimerkiksi työn ohjausseminaarissa käyty keskustelu osoitti sen, että vähääkään 3D-mallinnusta tunteva yleisö ymmärtäisi puhtaasti englanninkieliset termit ja ilmaisut paremmin, kun taas opinäytetyön formaattiin soveltuisi paremmin suomennokset. Lopulta pyrin löytämään tässä kultaisen keskitien ja käyttämään molempien kielten termejä, siinä missä ne parhaiten sopivat kontekstiin.

Koen, että itse työ onnistui lopulta erittäin hyvin ja vastaa sille asetettuja tavoitteita. Hahmo on pelattavissa Unreal Engine-projektissa ja näyttää ammattimaiselta ja teknisesti laadukkaalta. Se täyttää konseptin ja pelimekaniikan asettamat kriteerit ja noudattaa PBR-materiaalistandardeja. Työn edetessä toki huomasin joitain asioita, joita voisi tehdä toisin, tai ehkä paremmin. Veistosvaiheessa epäsymmetriaa ja ihon tekstuureja voisi korostaa vielä enemmän. Lowpoly-mallin topologia voisi olla parempikin. Hiustekstuurit voisivat myös olla vähän "rohkeampia" ja voimakkaampia muotoja ja niistä voisi olla enemmän variaatioita. Samoin itse hiuskortit voisivat olla kapeampia ja niitä voisi olla enemmän, jotta hiukset näyttäisivät erityisesti liikkeessä paremmalta. Toki tässä myös cloth-asetuksia säätämällä voitaisiin päästä parempiin tuloksiin. Hahmon kasvoin käyttäisin kuitenkin ehkä TexturingXYZ:n ihokarttoja, jotta kasvot eivät näyttäisi ihan niin pehmeiltä, kuin mitä ne nyt näyttävät. Weight painting on aina haastava osio, joka voisi aina olla vielä vähän parempi. Mutta opin projektista erittäin paljon ja tiedän, että seuraava hahmo on paljon helpompi ja nopeampi luoda.

Jatkokehityskohteena tutkisin animaatioiden ohjaamia normal-karttoja, jossa hahmon eri ilmeitä varten luotaisiin uusia normal-karttoja, jotka korostaisivat kutakin ilmettä esimerkiksi otsarypyin tai hymyjuovin ja niiden välillä tapahtuisi sulava muutos. Toinen kehityskohde olisi niin sanotut transitional meshes, eli eräänlaiset välimateriaalit, jotka tekisivät esimerkiksi silmäluomen ja silmämunan tai hampaan ja ikenen rajaa pehmeämmän ja luonnollisemman näköiseksi. Silmiin voisi myös tehdä varjo-objektin, joka mukailisi ripsien silmään aiheuttamaa varjoa, ja saisi silmän näyttämään vielä realistisemmalta.

LÄHTEET

Adobe 2019. Todellisuuden uudelleensuunnittelu. Viitattu 28.3.2019,
<https://www.adobe.com/fi/products/photoshop.html>.

Artstation 2019. Marketplace: resources for artists. Viitattu 21.3.2019,
www.artstation.com/marketplace.

Assassin's Creed: Odyssey 2018. Assassin's Creed: Odyssey. Viitattu 25.3.2019,
<https://assassinscreed.ubisoft.com/game/en-us/home>.

Autodesk 2019. Maya. Viitattu 28.3.2019,
<https://www.autodesk.fi/products/maya/overview>.

Autodesk 2019. Motionbuilder. Viitattu 6.5.2019,
<https://www.autodesk.com/products/motionbuilder/overview>.

Bartosz Styperek 2018. Hair Tool for Blender. Viitattu 5.5.2019,
<https://www.artstation.com/artwork/qbOqP>.

Blender 2019. Free & open source 3D creation. Free to use for any purpose, forever. Viitattu 28.3.2019,
<https://www.blender.org/>.

Blender market 2017. Auto-Rig Pro. Viitattu 5.5.2019,
<https://blendermarket.com/products/auto-rig-pro>.

Cg Elves 2016. Download free ZBrush skin alpha brushes pack. Viitattu 21.3.2019,
<https://cgelves.com/download-free-zbrush-skin-alpha-brushes-pack/>.

Claire Heginbotham 2019. What is photobashing. Viitattu 5.4.2019,
<https://conceptartempire.com/photobashing/>.

Concept Art Empire 2019. What is a concept artist? Viitattu 12.5.2019,
<https://conceptartempire.com/what-is-concept-artist/>.

Dan Roarty 2015. Lara – Rise of the Tomb Raider. Viitattu 24.3.2019,
<https://www.artstation.com/artwork/lara-rise-of-the-tomb-raider>.

Daniel Rose 2018. The game art software landscape of 2018 part 1. Viitattu 28.3.2019,
<https://medium.com/gametextures/the-game-art-software-landscape-of-2018-part-1-d677cd906420>.

Flipped Normals 2018. How to retopologize the body in Maya. Viitattu 5.5.2019,
https://www.youtube.com/watch?v=_TYOgl9kJtU&t=1446s.

Flipped Normals 2018. Retopology for beginners in Maya. Viitattu 20.3.2019,
<https://www.youtube.com/watch?v=xpDWta5O3n8>.

Foundry 2019. Mari: High-resolution digital 3D painting and texturing. Viitattu 4.4.2019,
<https://www.foundry.com/products/mari>.

Krita 2019. Krita. Viitattu 28.3.2019,
<https://krita.org/en/#>.

Larryn Bell 2018. Detroit: Become Human – Voice actors and cast. Viitattu 19.5.2019,
<https://www.primagames.com/games/detroit-become-human/feature/detroit-become-human-voice-actors-and-cast>.

Leonardo lezzi 2017. All you need to know about texel density. Viitattu 5.5.2019,
<https://www.artstation.com/artwork/qbOqP>.

Marmoset 2018. Marmoset Toolbag 3. Viitattu 4.4.2019,
<https://marmoset.co/>.

Marvelous Designer 2018. Marvelous Designer. Viitattu 4.4.2019,

<https://www.marvelousdesigner.com/>.

McDermott, W. 2018. The PBR guide: A handbook for physically based rendering. Yhdysvallat: Allegorithmic.

Mixamo 2019. Mixamo. Viitattu 5.5.2019,
<https://www.mixamo.com/>.

New Gen Apps 2018. Unreal Engine review: pros, cons and suitability. Viitattu 10.4,
<https://www.newgenapps.com/blog/unreal-engine-review-pros-cons-suitability>.

Pinterest 2019. Character refs. Vain sisäiseen käyttöön. Viitattu 19.3.2019,
<https://fi.pinterest.com/tribane86/character-refs>.

Pixologic 2019. ZBrush. Viitattu 28.3.2019,
<http://pixologic.com/>.

Playstation 2014. Playstation Experience, modeling Nathan Drake: bringing an iconic character to PS4 panel. Viitattu 5.5.2019,
https://www.youtube.com/watch?v=70jVUBnp6lQ&t=89s&has_verified=1.

Polycount 2014. Polycounts in next gen games. Viitattu 19.3.2019,
<https://polycount.com/discussion/141061/polycounts-in-next-gen-games-thread>.

Polycount 2018. Channel packing. Viitattu 5.5.2019,
<http://wiki.polycount.com/wiki/ChannelPacking>.

Sabin Lalancette 2018. Alexios/Kassandra Outfit – Mercenary. Viitattu 24.3.2019,
<https://www.artstation.com/artwork/EVqrBN>.

Saurabh Jethani 2018. Creating realistic skin in Toolbag with Saurabh Jethani. Viitattu 6.5.2019,
<https://marmoset.co/posts/creating-realistic-skin-toolbag-saurabh-jethani/>.

Shadow of the Tomb Raider 2019. Shadow of the Tomb Raider. Viitattu 25.3.2019,

<https://tomraider.square-enix-games.com/en-us>.

Sketchfab 2019. Buy royalty free 3D models. Viitattu 4.4.2019,
<https://sketchfab.com/store>.

Substance by Adobe 2019. Substance Painter: 3D painting. Viitattu 4.4.2019,
<https://www.allegorithmic.com/products/substance-painter>.

Texturingxyz 2019. The finest art of detail. Viitattu 5.5.2019,
<https://texturing.xyz/>.

Topogun 2018. Topogun 2. Viitattu 4.4.2019,
<http://www.topogun.com/>.

Unreal Engine 2019. Animation retargeting. Viitattu 5.5.2019,
<https://docs.unrealengine.com/en-us/Engine/Animation/RetargetingDifferentSkeletons>.

Unreal Engine 2019. Digital humans. Viitattu 5.5.2019,
<https://docs.unrealengine.com/en-us/Resources/Showcases/DigitalHumans>.

Unreal Engine 2019. Photorealistic character. Viitattu 23.3.2019 & 5.5.2019,
<https://docs.unrealengine.com/en-us/Resources/Showcases/PhotorealisticCharacter>.

Unreal Engine 2019. Physically based materials. Viitattu 5.5.2019,
<https://docs.unrealengine.com/en-us/Engine/Rendering/Materials/PhysicallyBased>.

Unreal Engine 2019. Skeleton assets. Viitattu 5.5.2019,
<https://docs.unrealengine.com/en-us/Engine/Animation/Skeleton>.

Unreal Engine 2019. Unreal Engine. Viitattu 4.4.2019,
<https://www.unrealengine.com/en-US/what-is-unreal-engine-4>.

Vincent Gault 2016. OpenGL and Direct X. Viitattu 5.5.2019,
<https://forum.allegorithmic.com/index.php?topic=8680.0>.

xNormal 2019. xNormal. Viitattu 4.4.2019,
<https://xnormal.net/>.