



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Joona Pääkkönen

Gender Blending

Kahden animoitavan hahmon sekoittaminen Unityssä

Metropolia Ammattikorkeakoulu

Medianomi

Viestinnän koulutusohjelma

Opinnäytetyö

20.4.2018

Tekijä(t) Otsikko	Joona Pääkkönen Gender Blending
Sivumäärä Aika	46 sivua + 2 liitettä 20.4.2018
Tutkinto	Medianomi
Tutkinto-ohjelma	Viestinnän tutkinto-ohjelma
Suuntautumisvaihtoehto	3D-animointi ja -visualisointi
Ohjaaja(t)	Lehtori Jaro Lehtonen
<p>Tämän opinnäytetyön tarkoituksena on kuvailla ja analysoida kirjoittajan kehittämää järjestelmää, joka mahdollistaa kahden animoitavan 3D-hahmon sekoittamisen keskenään ja hahmojen välisen animaatioiden retargetoimisen. Työn tavoitteena on tarjota lukijalle tarvittavat tiedot järjestelmän replikointiin. Lopussa pohditaan järjestelmän käyttökelpoisuutta ja mahdollisia käyttötarkoituksia.</p> <p>Prosessi on jaettu artistiosuuteen ja ohjelmointiosuuteen. Artistiosuudessa käydään läpi kahden hahmon mallinnus- riggaus- ja teksturointiprosessi, ja kerrotaan niissä järjestelmän kannalta huomioonotettavista seikoista. Ohjelmointiosuudessa kuvataan järjestelmän varsinainen toiminta. Järjestelmä toteutettiin Unity-pelimoottorissa, mutta toimintaperiaatteet pyritään selittämään mahdollisimman alustasta ja ohjelmointikielestä riippumattomasti.</p>	
Avainsanat	Unity, riggaus, skriptaus, animaatio, 3D, blendaus

Author(s) Title	Joona Pääkkönen Gender Blending
Number of Pages Date	46 pages + 2 appendices 20 April 2018
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Jaro Lehtonen, Senior Lecturer
<p>This thesis aims to describe and analyse a system developed by the author, which enables the blending of two animatable 3D-characters and animation retargeting between the characters. The goal of this thesis is to provide the reader with the knowledge required to replicate the system. Lastly, the usefulness and possible applications of the system are examined.</p> <p>The process has been divided into two sections: art and programming. The art section walks through the modelling, rigging and texturing of the two characters and the requirements for blending them. The programming part describes the way the blending system works. The system was implemented in the Unity game engine; however, this thesis aims to explain the principles as independently from platform or programming language as possible.</p>	
Keywords	Unity, rigging, scripting, animation, 3D, blending

Sisällys

1	Johdanto	1
2	Pohjatietoa	2
2.1	Luuanimaatio	2
2.2	Blend shapet	5
2.3	Topologia ja verteksijärjestys	6
2.3.1	Topologia	6
2.3.2	Verteksijärjestys	7
2.4	Verteksin todellinen luonne	9
3	Hahmojen toteutus	9
3.1	Suunnittelu	10
3.2	Mallintaminen	11
3.2.1	Päiden mallinnus	12
3.2.2	UV-koordinaatit	16
3.3	Riggaaminen	18
3.3.1	Luuranko	18
3.3.2	Skinnaaminen	20
3.3.3	Animaatio-ohjaimet	21
3.4	Teksturoidi	23
3.5	Animointi	24
4	Blendauksen toteutus	25
4.1	Yleiskatsaus	25
4.2	Hahmojen tuominen Unityyn	27
4.3	Geometrioiden blendaus	30
4.4	Luurankojen blendaus	31
4.5	Bind pose	33
4.6	Tekstuureiden blendaus	34
4.7	Lopputuloksen valmistelu	35
4.8	Animaation retargetointi	35
5	Jatkokehitysideoita	39
5.1	Wrap	39
5.2	Moniulotteisuus	41

6	Pohdintaa	41
	Lähteet	45
	Liitteet	
	Liite 1. Video: Blendaus toiminnassa	
	Liite 2. Video: Ilmeet eri blendauksen painoarvoilla	

1 Johdanto

Tässä opinnäytetyössä kerron kehittämästäni kokeellisesta järjestelmästä, joka sallii kahden animoitavan 3D-hahmon keskinäisen sekoittamisen eli blendaamisen, jonka tuloksena syntyy uusia hahmoja, jotka ovat alkuperäisten hahmojen välimuotoja. Järjestelmäni kattaa myös animaatioiden siirtämisen eli retargetoinnin hahmolta toiselle. Opinnäytetyön tavoitteena on antaa lukijalle tarvittavat tiedot järjestelmän replikointiin, ja analysoida sen tuottamia tuloksia.

Järjestelmän toiminnan ymmärtäminen vaatii hiukan sellaista pohjatietoa 3D-grafiikan toiminnasta, jota tavallisella 3D-graafikolla ei välttämättä ole. Siksi kerronkin ensin luvussa 2 hieman, mitä 3D-grafiikan konepellin alla tapahtuu. Tätä voi pitää opinnäytetyön teoriaosuutena.

Opinnäytetyön käytännön osuus käsittelee järjestelmän testausta varten tekemääni projektia, jossa toteutin kaksi hahmoa ja implementoin kehittämäni järjestelmän Unity-pelimoottorissa. Käytännön osuus on jaettu artisti- ja ohjelmointiosuuksiin. Artistiosuudessa kerron prosessistani hahmojen luontiin, ja siinä blendaamisen kannalta huomioonotettavista seikoista. Ohjelmointiosuus käsittelee varsinaisen blendausjärjestelmän toteutusta Unityssä, ja kerron siinä järjestelmän toimintaperiaatteista mahdollisimman alustasta ja ohjelmointikielestä riippumattomalla tavalla.

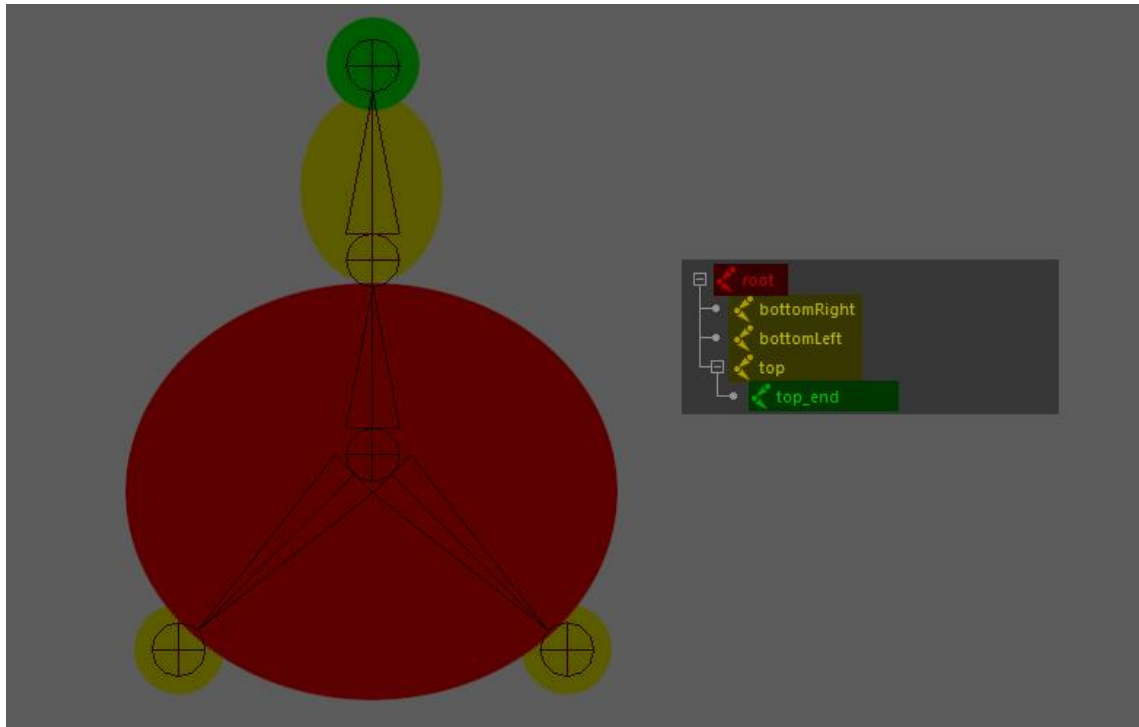
Vaikka olen yksinkertaistuksen vuoksi kuvannut prosessin vaiheittaisena, todellisuudessa tein iteratiivista testailua eri vaiheiden välillä. Ensimmäiset testit tein pelkkää geometriaa blendaamalla, kun hahmoilla ei ollut vielä kulmakarvoja tai hampaita. Tämä siksi, että halusin päästä tekemään testejä mahdollisimman aikaisessa vaiheessa projektia. Nyt kun olen kerran tehnyt sen, tekisin sen tulevaisuudessa kuitenkin tämän opinnäytetyön kuvaamassa järjestyksessä.

2 Pohjatietoa

Avaan tässä luvussa 3D-grafiikan toimintaa hieman syvemmin. Kehittämäni järjestelmän toiminnan ymmärtäminen edellyttää jonkin verran pohjatietoa siitä, mitä 3D-grafiikan konepellin alla tapahtuu. Oletan, että lukija omaa jo valmiiksi perustiedot 3D-grafiikasta. Osa käsittelemistäni asioista on kuitenkin sellaisia, joita 3D-graafikon ei ole tavallisesti tarpeen tietää, koska mallinnusohjelmat piilottavat tyypillisesti ylimääräisen monimutkaisuuden käyttöliittymänsä taakse. 3D-grafiikan pinnan alla on jonkin verran yliopistotason matematiikkaa, mutta pyrin selittämään sen mahdollisimman suurpiirteisesti ja ymmärrettävästi niin, että 3D-graafikko kykenee ymmärtämään mistä on kyse. Koska 3D-grafiikassa ei ole kaikelle vain yhtä yleisesti käytettyä termiä, vaan asioiden nimet riippuvat usein käytetystä mallinnusohjelmasta, määrittelen samalla myös käyttämäni termit.

2.1 Luuanimaatio

Luuanimaatio on tekniikka, jossa animoitavalle 3D-mallille, usein ihmismäiselle tai muulle orgaaniselle hahmolle, luodaan luuranko (skeleton tai armature), joka ohjailee sen liikkeitä. Luuranko on hierarkia, joka koostuu objekteista, joita kutsutaan joko luiksi (bone) tai niveliksi (joint). Aion jatkossa käyttää termiä luu, koska se on luurangon kontekstissa selkeämpi. Luuobjekteja ei kuitenkaan pidä ajatella anatomisina luina, koska se voi olla niiden toiminnan kannalta harhaanjohtavaa.



Kuvio 1. Yksinkertainen luuhierarkia Autodesk Mayassa.

Kuviossa 1 olen havainnollistanut luuhierarkiaa värittämällä sen eri tasot eri väreillä. Punaisella merkitty hierarkian juuriluu (root) on yksittäinen luuobjekti, vaikka se näyttää koostuvan kolmesta luusta. Juuriluun alapuolella sijaitsevilla kahdella keltaisella luulla ja ylimmällä, vihreällä luulla taas ei ole pituutta lainkaan. Todellisuudessa "nivel" kuvaa luuobjektien toimintaa paljon paremmin, koska niillä ei oikeasti ole pituutta, vaan niiden väliset viivat on piirretty vain havainnollistamaan niiden parent – child -suhteita hierarkiassa. Luun "pituus" saadaan laskemalla välimatka kahden luuobjektin välillä.

Jokaisella luuobjektilla on oma transformaatio, joka muodostuu translaatiosta, rotaatiosta ja skaalasta. Translaatio määrittää luun paikan avaruudessa, rotaatio luun kierron keskipisteensä ympäri. Skaala määrittää luun koon kertoimet kolmella akselilla. Koska luut ovat hierarkiassa, jokainen luu perii lisäksi isäntäluunsa transformaatiot. Luun transformaatiota isäntäluunsa suhteen kutsun tästedes lokaaliksi transformaatioksi, kun taas transformaatiota suhteessa maailmaan, johon lasketaan mukaan koko aikaisemman hierarkian transformaatiot, sanon maailma-transformaatioksi. Nämä muodostavaa translaatiota, rotaatiota ja skaalaa kutsun vastaavasti lokaaliksi translaatioksi, maailma-translaatioksi, lokaaliksi rotaatioksi jne.

On tyypillistä, että luun x-akseli osoittaa kohti hierarkiassa seuraavaa luuta, ja tällöin kahden luun välisen viivan asennosta voi päätellä isäntäluun rotaation. Tällaisessa tapauksessa luiden välisen etäisyyden voi myös helposti selvittää katsomalla lapsiluun lokaalin x-akselin suuntaista translaatiota. Joskus hierarkian "oksien" päihin lisätään ylimääräiset end-luut, jotka ovat luurangon toiminnan kannalta turhia, mutta auttavat havainnollistamaan niiden isäntäluiden rotaatioita. Kuviossa 1 top -luun x-akselin voi olettaa osoittavan kohti top_end -luuta. Kyseessä on kuitenkin vain yleinen käytäntö, ja tämä sääntö ei kaikissa tapauksissa implisiittisesti päde. Lisäksi esimerkiksi root -luun rotaatiota ei ole mahdollista päätellä, koska sillä on kolme lasta ja x-akseli voi osoittaa vain yhteen suuntaan.

Kun 3D-hahmolle on luotu luuranko, se kytketään hahmoon niin sanotun skinin avulla. Skin on kokoelma verteksikohtaisia painoarvoja, jotka määrittävät, minkä verran mikäkin luu vaikuttaa mihinkin verteksiin. Prosessia, jossa 3D-hahmolle luodaan skin, kutsutaan skinnaamiseksi. Skiniin voi vaikuttaa koko luuranko, tai vain osa siitä, ja siihen voi myöhemmin lisätä tai siitä voi poistaa vaikuttavia luita. Kun luita liikutellaan, käännellään tai skaalataan, niihin skinnatun mallin verteksit perivät luiden transformaatiot niiden painoarvojen mukaisesti joko kokonaan, osittain tai ei lainkaan. Transformaatiot tapahtuvat suhteessa luihin, joten halutunlaisen deformaation aikaansaamiseksi luiden oikea sijoittelu on tärkeää. Yhteen verteksiin voi vaikuttaa rajallinen määrä luita. Määrä on yleensä riggaajan päätettävissä, tosin pelimoottoreilla on omat rajoitteensa. Esimerkiksi Unity rajoittaa yhteen verteksiin vaikuttavien luiden määrän korkeintaan neljään (Unity 2018).

Luurangon perusasentoa, jossa hahmo ei ole deformoitunut, kutsutaan bind poseksi. Nimi tulee siitä, että se on asento jossa hahmo on sillä hetkellä kun siihen kytketään skin. Bind pose kertoo skinnatulle mallille, mitkä ovat ne luiden alkuperäiset transformaatiot, joiden suhteen sen tulee deformoitua. 3D-mallin verteksit liikkuvat luiden bind posen ja nykyisten transformaatioiden välisen muutoksen verran. Kun luuranko on bind poseissa, 3D-malli ei siis deformoidu. Koska luuranko voi olla kiinni useammassa skinissä, sillä voi myös olla useampia bind poseja. Vaikka bind posen tarkoituksena on tallentaa luurangon asento skinnaushetkellä, sitä voi kuitenkin myöhemmin muuttaa. Koska bind pose määrää miten malli deformoituu, sen muuttaminen deformoi mallia samaan tapaan kuin luurangon liikuttelu (Autodesk 2017).

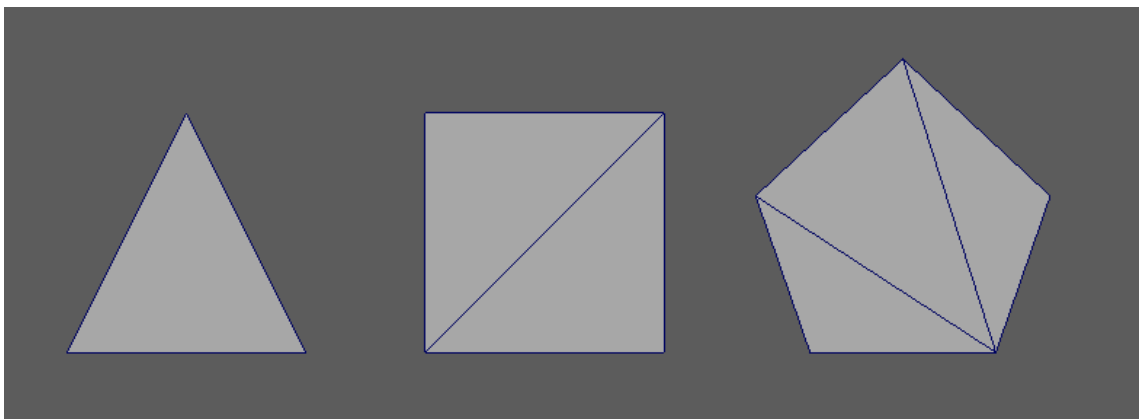
2.3 Topologia ja verteksijärjestys

Jotta 3D-malli voisi toimia blend shapena toiselle, molemmilla on oltava samanlainen topologia ja sama verteksien järjestys. Tämän vuoksi ne tehdäänkin yleensä kopioimalla alkuperäinen malli ja tekemällä siihen halutut muutokset. Blendausjärjestelmäni toimii hieman samaan tapaan kuin blend shapet, ja sen ymmärtämisen kannalta on tärkeää ymmärtää näiden kahden käsitteen määritelmät.

2.3.1 Topologia

3D-grafiikassa topologialla tarkoitetaan 3D-mallin pinnan muodostavan polygoniverkon (mesh) rakennetta. Verkko koostuu verteksistä jotka muodostavat muiden verteksien kanssa polygoneja tai ns. "tahoja" (face). Kahdella polygoniverkolla voidaan sanoa olevan sama topologia, kun molemmissa on sama määrä pisteitä, jotka liittyvät toisiinsa samalla tavalla.

Yhtenäinen topologia ei edellytä, että verkot olisivat saman muotoisia, mutta se asettaa rajoitteita sille, kuinka pitkälle muotoa voi muuttaa. Esimerkiksi pallosta ei saa tehtyä donitsia muuttamatta sen topologiaa, koska donitsissa on keskellä reikä. Toinen rajoittava tekijä on verkon muodostavien pisteiden ja siten polygonien määrä, jota kutsutaan usein myös mallin resoluutioksi. Toisin kuin matematiikassa, jossa kahvikuppi ja donitsi omaavat saman topologian, 3D-grafiikassa donitsista on mahdotonta saada kahvikuppia, jos sen resoluutio on liian pieni. Käytettävissä olevat polygonit eivät välttämättä riitä kuvaamaan haluttua muotoa tai detaljin määrää.



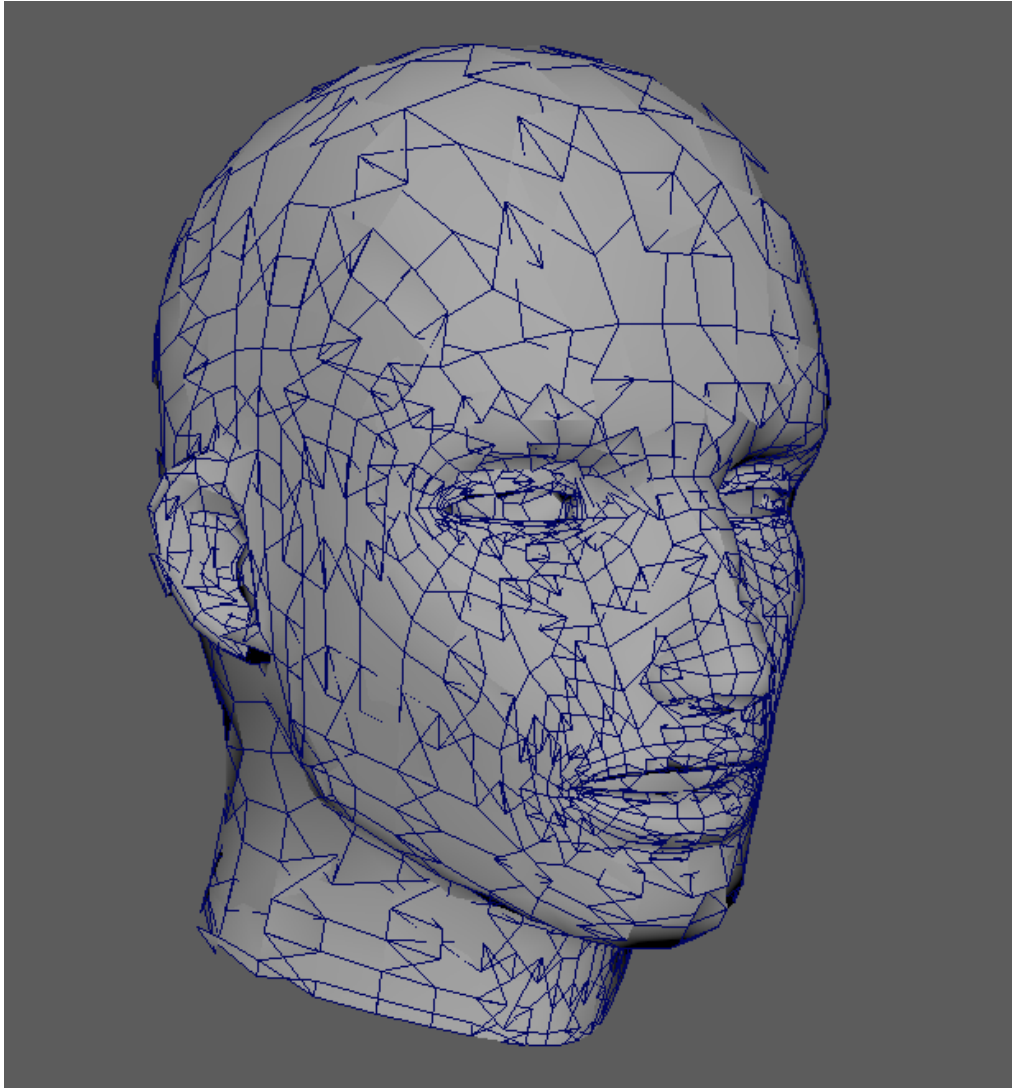
Kuvio 3. Kaikki polygonit voi jakaa kolmioiksi.

Oleellinen osa topologiaa on sen muodostavien polygonien muoto. Yksi taho voi koostua vähintään kolmesta pisteestä, jolloin kyseessä on kolmio. Kun 3D-malli piirretään näytölle, kaikki polygonit jaetaan kolmioiksi. Kolmiota voidaankin ajatella pienimpänä yksikkönä polygonien maailmassa. Nelikulmio koostuu kahdesta kolmiosta, viisikulmio kolmesta, ja niin edelleen (kuvio 3). Mallinnusvaiheessa on tyypillistä käyttää nelikulmaisia polygoneja ("quad", lyhennys sanasta quadrilateral), koska se helpottaa mallinnusta, auttaa ymmärtämään mallin rakennetta ja ennustamaan deformaatioiden toimivuutta (Anatomy for 3D Artists 2015, 256).

2.3.2 Verteksijärjestys

Verteksit ovat tietokoneen muistissa vain lista numeroita kuten kaikki muukin, ja ne sijaitsevat siellä jossakin järjestyksessä. Tätä kutsutaan 3D-mallin verteksijärjestykseksi (vertex order). Verteksien järjestys määrittyy usein sen mukaan, missä järjestyksessä verteksit on luotu. Viimeisin luotu verteksi lisätään aina listan perälle. Verteksijärjestys ei täten ole suoranaisesti kytketty verteksien keskinäisiin topologisiin suhteisiin. Verteksien järjestystä voi halutessaan muuttaa (Stack Exchange 2015).

Jos yhtä 3D-mallia käytetään blend shapena toiselle, tietokoneen on jotenkin tiedettävä, mitkä kahden mallin verteksit vastaavat toisiaan. Tämän vuoksi mallien verteksien on oltava muistissa samassa järjestyksessä. Näin ohjelma voi olettaa, että vertekseillä jotka vastaavat toisiaan on sama indeksi. Jos järjestys on malleissa eri, blend shape ei toimi halutulla tavalla, koska ohjelma interpoloi verteksejä, jotka eivät vastaa malleissa toisiaan (kuvio 4).



Kuvio 4. Epäyhtenäisestä verteksijärjestyksestä johtuva blend shapen rikkoutuminen.

Tämä voi tuottaa yllättäviä ongelmia. Koska 3D-mallin data on mahdollista järjestellä eri tavoilla ja koska verteksijärjestys on lopulta mielivaltainen, verteksien järjestys voi muuttua tallennettaessa malli eri ohjelmassa tai eri formaatissa, kuin missä se alun perin mallinnettiin. Jotkin malleille tehtävät operaatiot voivat myös muuttaa sen verteksien järjestystä. Käytettäessä useita eri ohjelmia mallin käsittelyyn, verteksijärjestys hajoaa kokemukseni mukaan käytännössä aina. Verteksijärjestyksen voi onneksi yhtenäistää jälkikäteen, kunhan malleilla on jotain yhteistä, kuten topologia tai UV-koordinaatit. Tämän voi tehdä esimerkiksi Mayan Transfer Vertex Order -työkalulla.

2.4 Verteksin todellinen luonne

Vertekseistä puhuttaessa tarkoitetaan yleensä pisteitä kolmiulotteisessa avaruudessa, joista polygoniverkko koostuu. Tämä on suuressa osassa tapauksista hyvä yksinkertaistus, ja useimmat mallinnusohjelmat piilottavat tarpeettoman monimutkaisuuden käyttöliittymänsä taakse käsittelemällä verteksejä pisteinä.

Verteksi on todellisuudessa kokoelma attribuutteja, joista piste avaruudessa on vain yksi. Muita ovat esimerkiksi normaalivektori ja UV-koordinaatti. Teknisemmät ohjelmistot kuten Houdini tekevät selkeän eron näiden välillä, käyttäen kahta termiä "point" ja "vertex". Point tarkoittaa pistettä avaruudessa, ja yhden pisteen voi jakaa useampi verteksi. Jos 3D-mallissa on terävä kulma, jossa polygonien kulmien normaalivektorit osoittavat eri suuntiin, siinä pisteessä on todellisuudessa kaksi tai useampia verteksejä. Verteksit jakavat saman pisteen avaruudessa, mutta niillä on eri normaalivektorit. Sama pätee pisteisiin, joilla on useampi UV-koordinaatti. Kahdessa saman topologian jakavassa polygoniverkossa voi todellisuudessa olla eri määrä verteksejä, riippumatta siitä, mitä mallinnusohjelma saattaa käyttäjälleen väittää. Aion jatkossa puhuessani vertekseistä käyttää tätä määritelmää (Cgwiki 2017).

Verteksien esittäminen pisteinä ei yleensä vaikuta mitenkään itse mallinnusprosessiin, mutta voi aiheuttaa ongelmia, jos mallit, joissa halutaan olevan sama määrä verteksejä, viedään myöhemmin sellaiseen ohjelmaan, joka ei abstrahoi verteksejä, kuten esimerkiksi Unityyn. Mahdolliset tästä aiheutuvat ongelmat voi välttää varmistamalla, että mallit vastaavat toisiaan myös terävien reunojen ja UV-pisteiden osalta.

3 Hahmojen toteutus

Kerron tässä luvussa projektin "artistiosuudesta" eli hahmojen toteutuksesta ja seikoista, jotka siinä tuli ottaa huomioon, jotta hahmojen blendaaminen oli mahdollista. Käytin hahmojen mallinnukseen Autodesk Maya ja Pixologic Zbrush -ohjelmia ja karttojen leipomiseen Blenderiä. Mikä tahansa mallinnusohjelma kuitenkin käy, jos siitä vain löytyy vastaava toiminnallisuus. Käytin itse eri ohjelmia eri työvaiheisiin, koska kaikilla ohjelmilla on omat vahvuutensa joissa ne ovat ylitse muiden. Kaikki vaiheet olisi kuitenkin varmasti mahdollista tehdä esimerkiksi Blenderissä, joka on ilmainen ja erittäin yleispätevä mallinnusohjelma.

3.1 Suunnittelu

Halusin keskittyä projektissani enemmän järjestelmän tekniseen toteutukseen, joten käytin hahmojen suunnitteluun varsin vähän aikaa. Jonkin verran suunnittelua oli kuitenkin tarpeen, koska blendausjärjestelmä asettaa hahmoille tiettyjä rajoitteita. Jotta hahmojen blendaaminen olisi mahdollista, niillä on oltava sama topologia. Koska animoin hahmoja, niillä piti myös olla samanlainen luuhierarkia. Hahmot eivät voineet siis erota toisistaan liikaa. Ajan säästämiseksi päätin toteuttaa hahmoista pelkät päät. Alkuperäinen ideani oli käyttää 3D-skanneja kahdesta oikeasta ihmisestä, mutta se osoittautui ajan ja resurssien puutteen vuoksi turhan kunnianhimoiseksi. Päädyin lopulta tyylieltyihin Disney-inspiroituneisiin ihmishahmoihin.

Pidin ideasta, että hahmot olisivat maskuliininen ja feminiininen versio toisistaan. Näin ne kontrastoisivat toisiaan pysyen silti hyvin samantyyllisinä. Hahmoja suunnitellessani kuvittelin mielessäni pelin, jossa pelihahmon sukupuoli olisi pelaajan päätettävissä, mutta hahmo säilyttäisi silti saman persoonallisuuden ja tyylin. Tämä oli mielestäni hyvä lähtökohta, koska molemmat hahmot ja niiden välimuodot tulisivat käyttämään samoja animaatioita. Tyylieltyessä sukupuolierot viedään usein äärimilleen (kuvio 5). Halusin mieshahmon olevan korostetun maskuliininen ja naisen vastaavasti mahdollisimman feminiininen. Naisilla on tyypillisesti pyöreämmät posket ja pienempi nenä. Miehekkäitä piirteitä taas ovat suuri nenä, matala otsa ja vahvat kulmakarvat (Burton *et al* 1993). Miesten kasvot ovat myös yleisesti suuremmat ja jyrkemmät. Tämä saa naisten silmät vaikuttamaan suuremmilta suhteessa kasvojen kokoon. Korostetun feminiiniset kasvot ovat miltei lapsenomaiset: suuret silmät, pieni nenä ja siro leuka. Nuorekkaita piirteitä pidetäänkin naisilla yleisesti viehättävinä (Schacht 2005).



Kuvio 5. Aunt Cass ja Callaghan elokuvasta Big Hero 6 (Julius 2014).

Tiesin jo etukäteen, että jos hahmoilla olisi erilaiset hiustyylit, niiden mallinnus tulisi olemaan hyvin haastavaa, koska mallinnetuilla hiuksilla on hyvin spesifinen topologia. Jokaisen välimuodon tulisi myös toimia itsenäisenä hiustyylinään. Hiusongelman voisi mahdollisesti välttää mallintamalla hiukset ns. hiuskorteilla, jotka jäljittelevät paremmin oikeiden hiusten toimintaa, mutta tämä lähestymistapa toimii paremmin realistisemmilla hahmoilla. Harkitsin antavani molemmille hahmoille saman hiustyylin, mutta jätin lopulta hiukset kokonaan pois.

3.2 Mallintaminen

Projektin kokeiluluontoisuuden vuoksi mallinnusprosessi oli hyvin iteratiivinen. Halusin päästä kokeilemaan rakentamaani järjestelmää mahdollisimman aikaisessa vaiheessa, ja siksi mallinsin hahmojen eri osat testailun eri vaiheissa. Lopulliset hahmomallit koostuvat viidestä eri osasta eli meshistä: Päästä, silmistä, kulmakarvoista, ripsistä ja suun sisuksista (kieli ja hampaat). Nämä kaikki olisi voinut yhdistää, ja pelimoottorissa se olisikin performanssin kannalta optimaalista. Koin kuitenkin osien pitämisen erillisinä hyväksi ratkaisuksi testailutarkoituksiini, koska se helpotti niiden muokkaamista erillään muusta hahmosta.

Omat uniikit haasteensa mallinnusprosessiin toi se, että hahmojen tuli täyttää blend shapejen vaatimukset eli jakaa sama topologia ja verteksien järjestys. Koska kehittämäni järjestelmä tulisi käsittelemään mallien raakaa dataa, niiden tuli myös jakaa samat UV-pisteet ja terävät kulmat. Tämän vuoksi mallinsin jokaisen hahmon osan ensin naiselle, ja käytin sitä sitten pohjana miesversiolle. Tämän jälkeen työstin molempia versioita rinnakkaisesti.

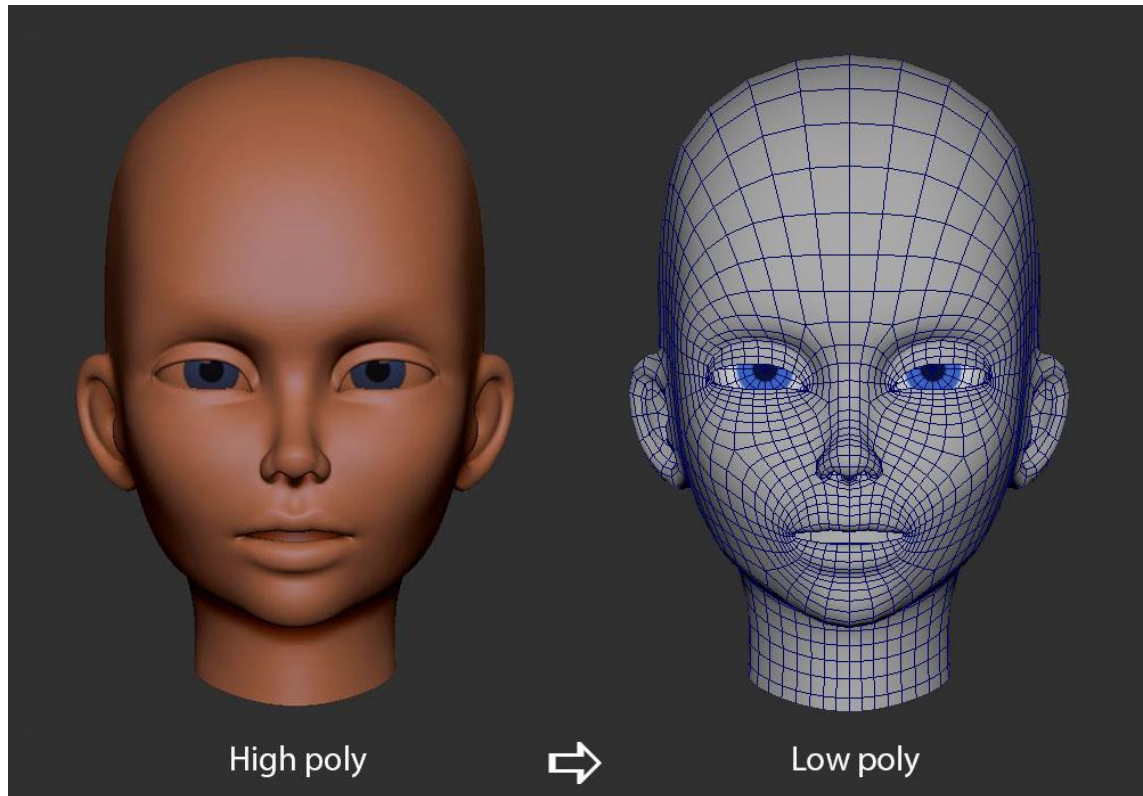
Pyrin hyvällä suunnittelulla välttämään topologiaan tai verteksien määrään vaikuttavien muutosten tekemisen, koska ne olisi pitänyt tehdä molemmille hahmoille samalla tavalla, mikä on usein vaivalloista ja aikaa vievää. Pyrin käytännössä varmistamaan, että eri osilla oli toimiva topologia jo ennen miesversion tekemistä. Naishahmossa tuli myös olla tarpeeksi resoluutiota, jotta siitä sai muokattua miehen ilman, että lopputulos oli kulmikas. En toisaalta halunnut resoluution myöskään olevan liian korkea. Liiallinen polygonien määrä voi pelimootoreissa aiheuttaa performanssiongelmiä ja vaikeuttaa myös itse mallinnus- ja riggausprosessia. Onnistuin mielestäni suurimmaksi osaksi hyvin. Kummaltakin hahmolta löytyy samat piirteet, vain eri kokoisina, mikä teki tapauksesta helpon. Jos olisin halunnut antaa toiselle hahmoista jonkin ruumiinosan jota toisella ei ole, esimerkiksi naishahmon rinnat jos olisin päätenyt mallintamaan koko kehon, tasapainoilu liian vähäisen ja liian korkean resoluution välillä olisi ollut haastavampaa.

Jouduin muokkaamaan topologiaa jälkikäteen vain kerran. Mallinsin hampaista alun perin vain etupuolen, ja päätin myöhemmin, että ne onkin mallinnettava kokonaan. Koska miehen hampaat olivat vain skaalattu versio naisen hampaista, koin, että tässä tilanteessa oli järkevintä vain poistaa miehen hampaat, muokata naisen hampaista ja kopioida ne uudestaan miehelle. Tilannetta helpotti se, että suun sisäosat olivat oma, erillinen osansa.

3.2.1 Päiden mallinnus

Päiden geometriat olivat hahmojen suurimmat ja tärkeimmät rakennuspalikat, ja käytinkin niiden mallintamiseen kaikista eniten aikaa. Loin ensin naisen kasvot korkearesoluutioisina digitaalisesti veistämällä Zbrush-ohjelmassa. Korkearesoluutioisen high poly -mallin halusin tehdä siksi, että sen yksityiskohdat voi leipoa myöhemmin hahmon tekstuuriin. Rakensin veistämieni kasvojen päälle lopullisen topologian käyttämällä Mayan quad draw -työkalua (kuvio 6). Tämä oli mielestäni hyvä työvaiheiden järjestys siksi, että se mahdollisti hyvän topologian miettimisen erillään itse

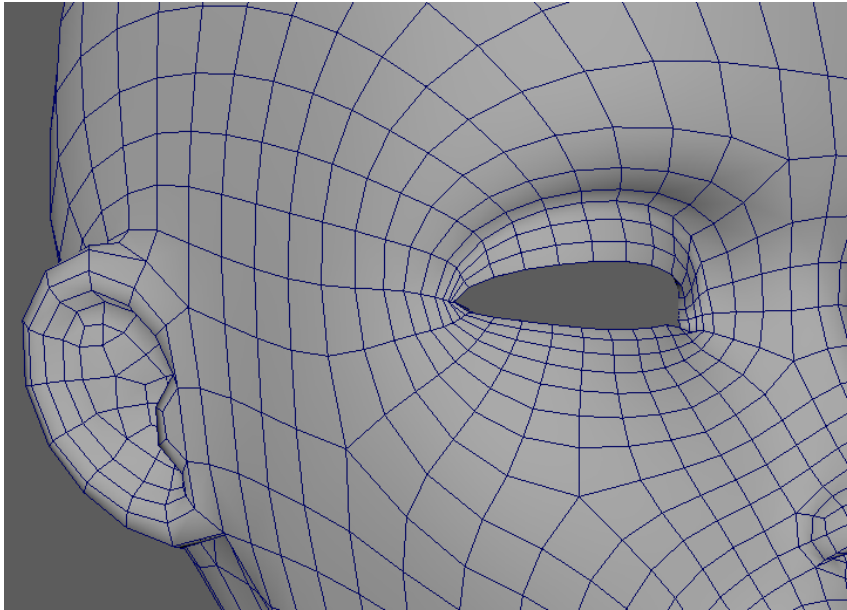
hahmon muodosta. Low poly -mallin voisi tehdä myös ennen veistämisvaihetta, kuten myöhemmin mieshahmon kohdalla teinkin. Toisin kuin Zbrush, Maya pyörittää varsin huonosti todella korkearesoluutioisia malleja. Siksi ennen Mayaan viemistä vähensin sen polygonimäärää käyttämällä Zbrushin decimate-työkalua.



Kuvio 6. Naishahmon pään high poly- ja low poly -mallit.

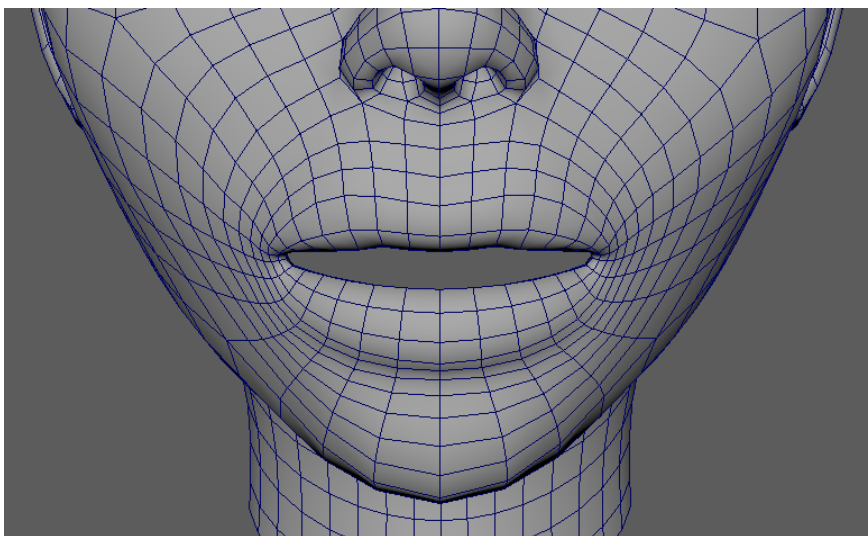
One of the most important parts of modeling a character is not where your points live on your mesh but why they live there. All points must have a logical reason for existing. Not just to hold the shape but to retain that shape while in motion. (Tindall 2009)

Hyvä topologia, joka deformatuu halutulla tavalla, on oleellinen osa animoitavan hahmon mallinnusta ja on erityisen tärkeää varsinkin kasvoissa, koska kasvojen on kyettävä hienovaraisiin ilmeisiin ja tunteiden ilmaisuun. Katsojan huomio on myös usein keskittynyt kasvoihin, ja lähikuvat kasvoista ovat yleisiä. Tämän vuoksi on tärkeää, että topologia tukee kasvojen liikkeitä. Erityistä huomiota vaativat silmien ja suun alue. Topologian on sallittava silmien ja suun sulava avautuminen ja sulkeutuminen. Tämä toteutetaan tyypillisesti suuta ja silmiä kiertävillä polygoninauhoilla, jotka vastaavat lihasten anatomiaa (Anatomy for 3D Artists 2015, 256–257).



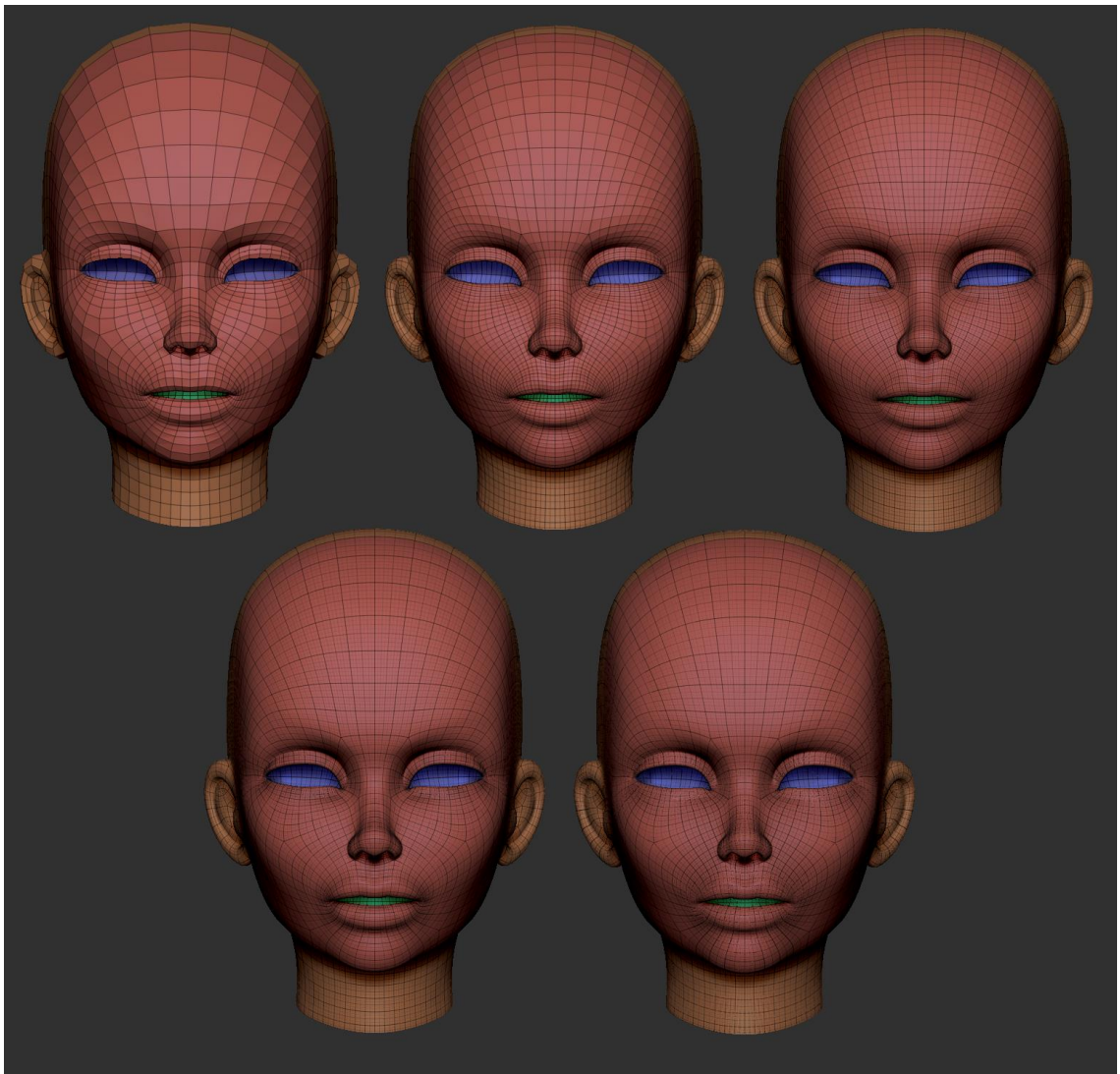
Kuvio 7. Silmiä ympäröivät polygoninauhut peilaavat silmälihaksia.

Polygoniverkko kuvaa kasvojen ulkopintaa, joten sillä ei kuitenkaan tule liiallisesti pyrkiä mallintamaan lihaksistoa, vaan paremminkin ihoa ja sen deformaatioita. Alueet, jotka deformoituvat enemmän, vaativat enemmän geometriaa. Kasvojen rypyt syntyvät kaikilla ihmisillä samoihin paikkoihin, kuten nasolabiaalipoimun alueelle. Topologian ei ole tarpeen kyetä kuvaamaan varsinaisia rypyjä, koska tämän voi tehdä normal tai displacement mapilla, mutta polygoninauhojen on hyvä mukaila muodostuvien ryppyjen suuntaa ja muotoa (Ritchie *et al* 2006).



Kuvio 8. Suuta ympäröivät polygoninauhut sallivat hymyryppyjen muodostumisen.

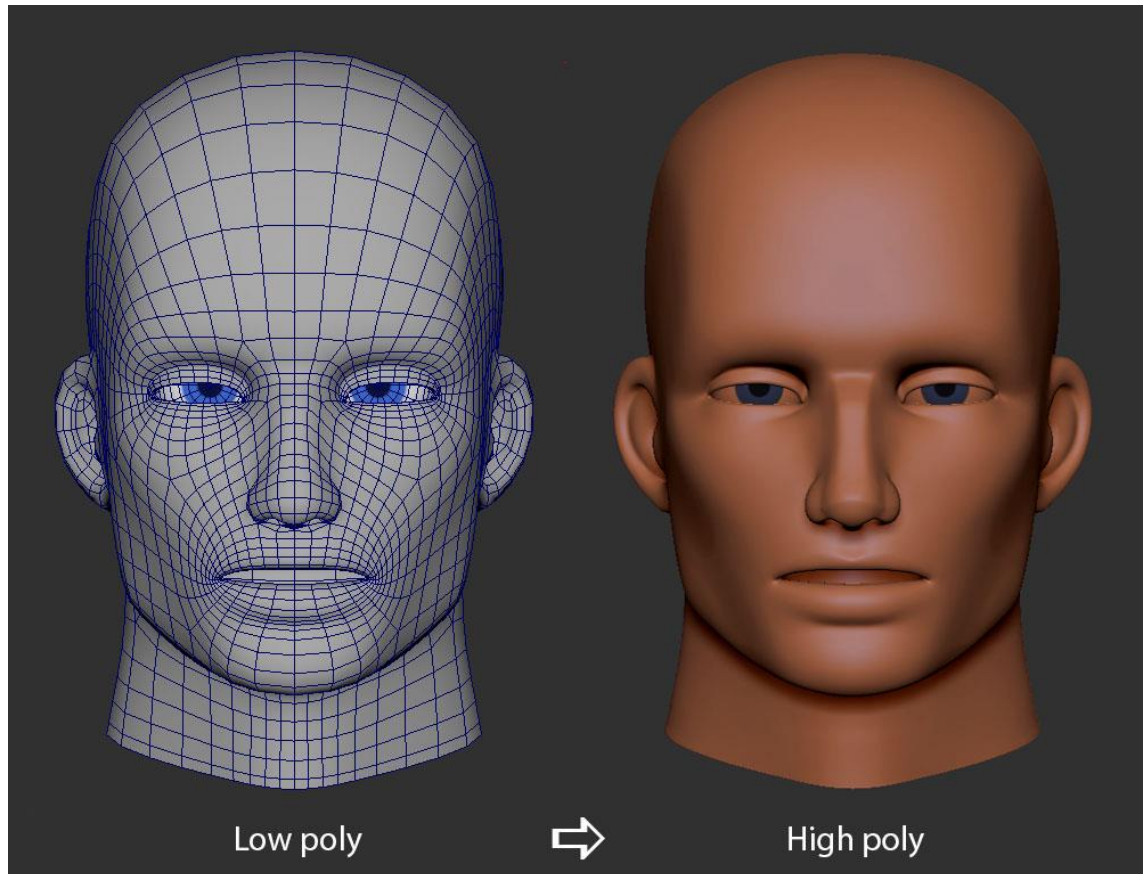
Kun olin lyönyt topologian lukkoon, vein naisen kasvot takaisin Zbrushiin. Jakamalla (divide) mallia kasvatin sen polygonimäärää suurin piirtein vastaamaan alkuperäistä veistämällä tehtyä high poly -mallia. Jokaisen yksittäisen jako-operaation välissä projisoin sen pisteet high poly -mallin pinnalle käyttäen project-työkalua (kuvio 9). Näin sain siirrettyä high poly -mallin yksityiskohdat uuteen topologiaan. Mallin jakaminen ja projisointi pehmensivät hiukan alimman subdivisiotason kulmia. Koska halusin silmäluomien reunojen pysyvän terävinä, jouduin tekemään jonkin verran ylimääräistä käsityötä niiden palauttamiseksi.



Kuvio 9. Korkearesoluutioisen mallin pinnalle projisoidun geometrian subdivisiotasot.

Tämän jälkeen palasin takaisin alimpaan subdivisiotasoon ja vääntelin kasvot miehiseen muotoon käyttäen move-työkalua. Zbrushissa alempiin subdivisiotasoihin tehdyt muutokset siirtyvät korkeammille tasoille, mutta korkeammassa resoluutiassa veistetyt

yksityiskohdat säilyvät ennallaan. Koska naishahmon kasvoihin veistämäni yksityiskohdat säilyivät näin mieshahmolla, korkearesoluutioisen mallin tekeminen miehelle vaati varsin vähän työtä.



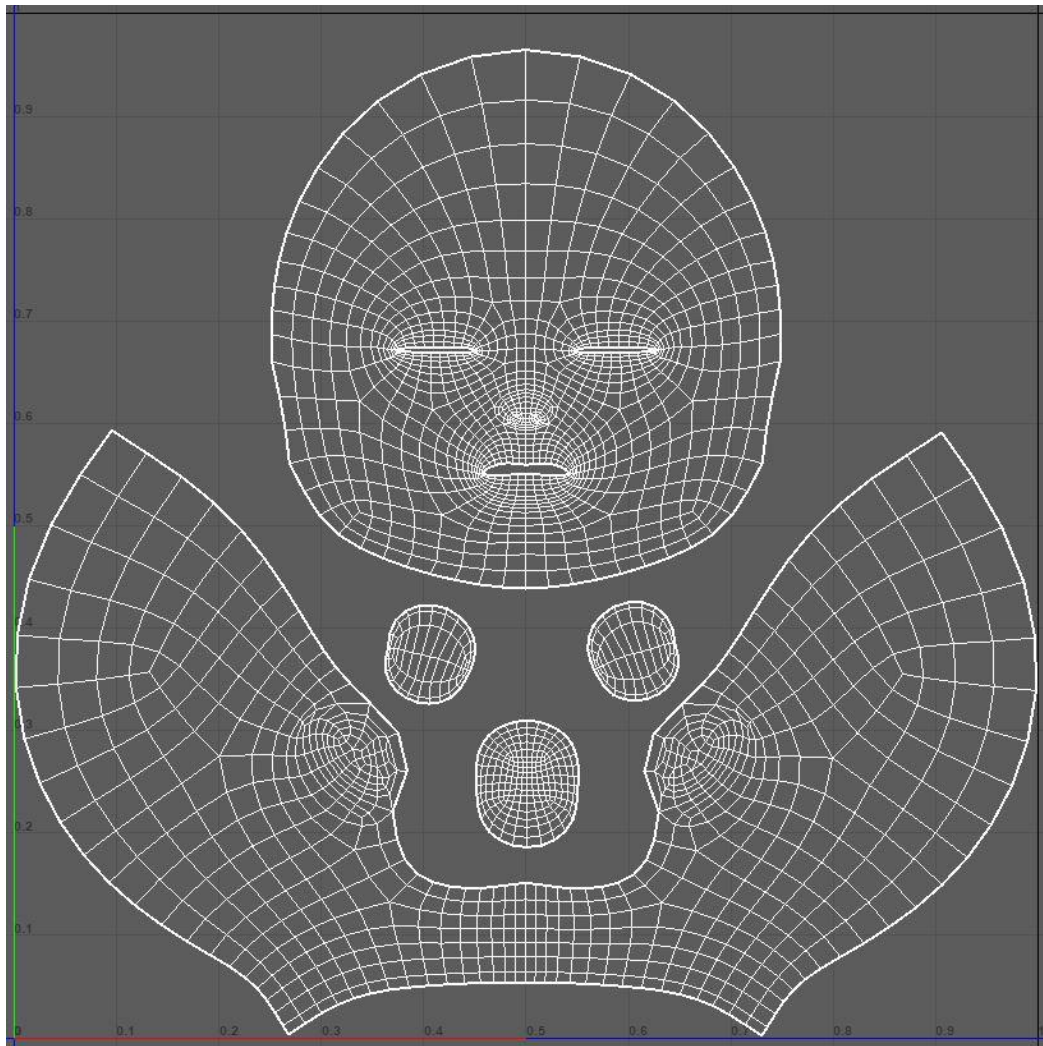
Kuvio 10. Miehen kasvojen low poly- ja high poly -mallit.

Koska hahmot tulivat käyttämään samoja animaatioita, oli tärkeää, että molemmilla hahmoilla oli täysin sama perusilme. Jos ilmeet eroavat toisistaan, hahmot eivät animoidu täysin samalla tavalla. Animaatiovaiheessa mieshahmon yläluomet käyttäytyivät odottamattomasti, ja huomasin, että yläluomet olivat miehellä hiukan enemmän kiinni kuin naisella. Pidin miehen silmäluomien asentoa parempana, joten korjasin tilanteen sulkemalla naishahmon silmiä hiukan jälkikäteen.

3.2.2 UV-koordinaatit

Jotta materiaalit toimisivat blendattaessa oikein, molemmilla hahmoilla on oltava samat UV-koordinaatit kaikissa niissä osissa, jotka käyttävät tekstuureja. Tämä tarkoitti omassa projektissani kasvoja, silmiä ja suun sisäosia. Optimoin UV-koordinaatit naishahmon geometrialle (kuvio 11). Silmien ja suun sisäosien mittasuhteet olivat molemmilla

hahmoilla samat, joten niiden UV-koordinaatit pysyvät optimaalisina myös miehellä, mutta tiesin että kasvojen tekstuuri tulisi miehellä erilaisten mittasuhteiden vuoksi hiukan venymään. Olin kuitenkin päättänyt käyttää sen verran suurikokoisia tekstuureja (4096 x 4096), etten kokenut tämän olevan ongelma. Jos olisin joutunut käyttämään pienempiä tekstuureja, olisin ehkä ottanut venymisen huomioon esimerkiksi nenän alueella, jonka koko muuttuu varsin radikaalisti, ja suurentanut sen osuutta UV-kartasta. UV-koordinaatit voi tehdä missä mallinnuksen vaiheessa tahansa, sillä ne on mahdollista kopioida hahmolta toiselle käyttämällä esimerkiksi Mayan Transfer Attributes -työkalua.



Kuvio 11. Kasvojen jaetut UV-koordinaatit.

Käytin UV-kartan levitykseen Mayan unfold-työkalua. Käsittelin sitä manuaalisesti sen verran, että suljin silmäluomet. Tein tämän, jotta tekstuuri venyisi mahdollisimman vähän hahmon sulkiessa silmänsä. UV-saarekkeet on skaalattu kartalla niin, että ne ovat

oikeassa mittakaavassa suhteessa toisiinsa, tosin suun sisäpuoli eli ns. "mouth bag" on kartalla todellista pienempi. Tämä johtuu siitä, että päädyin suurentamaan sitä jälkikäteen, koska vasta myöhemmässä vaiheessa mallintamani hampaat ja kieli eivät aluksi mahtuneet sen sisään. Koko UV-kartassa jäi samaksi, koska olin tässä vaiheessa jo ehtinyt teksturoimaan kasvot. Tämä ei kuitenkaan ollut suuri ongelma, koska suun sisäpuoli näkyy ulospäin hyvin vähän ja onkin suureksi osaksi hampaiden ja kielen peittämä.

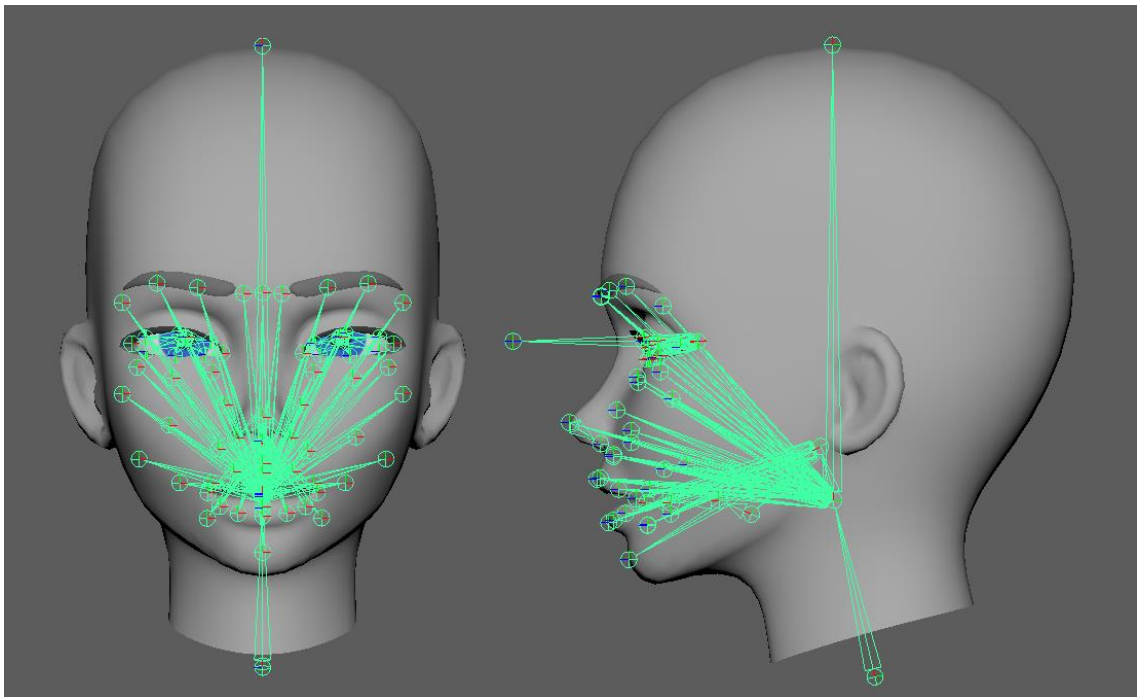
3.3 Riggaaminen

3D-grafiikassa termillä rig voidaan viitata joko pelkästään hahmon skiniin ja sitä ohjailevaan luurankoon, mutta tyypillisesti määritelmään luetaan myös animaatio-ohjaimet, joilla luurankoa animoidaan mallinnohjelmissa. 3D-hahmon riggausprosessi koostuu tavallisesti kolmesta vaiheesta: Luurangon luomisesta, geometrian skinnaamisesta luihin ja animaatio-ohjainten tekemisestä. Prosessi piti projektissani suorittaa kokonaisuudessaan vain blendauksen pohjahahmolle, tässä tapauksessa naiselle, ja mieshahmolle oli tarpeen tehdä vain oma luuranko. Tämä oli kätevää, koska riggaus on usein vaivalloinen ja aikaa vievä prosessi. Kerron tässä luvussa hieman riggauksen kulusta ja siinä blendausjärjestelmän kannalta huomioonotettavista seikoista.

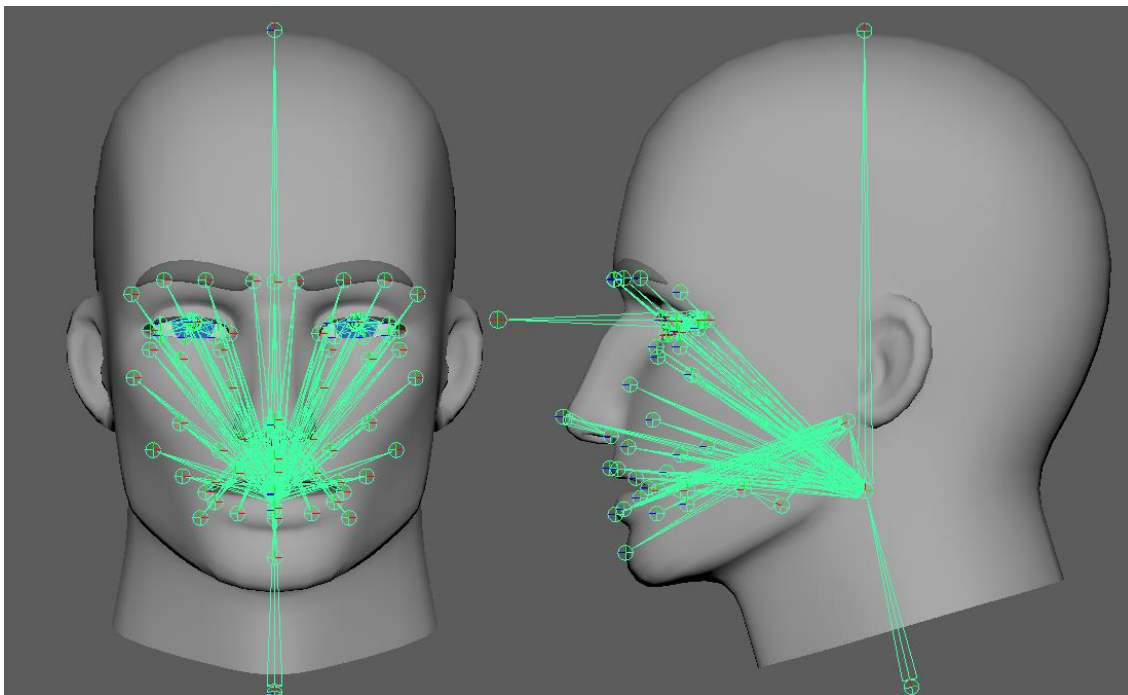
3.3.1 Luuranko

Kasvoja animoidaan peleissä tyypillisesti joko pelkillä luilla tai luilla ja blend shapeilla. Päätin omassa projektissani animoida kasvot kokonaan luilla. Siinä missä pään geometrian voi ajatella kuvaavan ihoa ja sen deformatumista, pään ja kasvojen luuobjektit jäljittelevät vastaavasti luita ja lihaksia. Tyylitellyn hahmon kasvojen voidaan kuitenkin haluta venyvän liioiteltuihin ilmeisiin, jotka eivät ole mahdollisia tosielämässä, jolloin ne vaativat myös sellaisia luita, jotka eivät ole yhtä sidottuja todelliseen anatomiaan. Toisaalta ollakseen uskottava tyylitellyn hahmon ei tarvitse kyetä aivan yhtä hienovaraisiin ilmeisiin kuin realistisen hahmon, ja se vaatii siksi vähemmän anatomiaan sidottuja luita. Riggaajan onkin hyvä tietää etukäteen, millaisiin ilmeisiin kasvojen tulee kyetä, ja suunnitella luurangon rakenne sen mukaan. Omat hahmoni olivat tyyliteltyjä, ja tiesin, että ilmeet tulisivat olemaan suhteellisen realistisia ja hillittyjä (ei äärimmäistä deformatumista), joten päädyin suhteellisen yksinkertaiseen kasvorigiin, johon otin vahvasti mallia Overwatch-pelin kasvorigeistä.

Voidakseen blendautua keskenään molemmilla hahmoilla tulee olla samanlainen luuhierarkia. Molemmista luurangoista pitää löytyä samat luut, ja niiden on oltavassa kummassakin hierarkiassa samassa järjestyksessä. Samaan tapaan kuin mallinnusvaiheessa, tein luurangon ensin naishahmolle ja kopioin sen tämän jälkeen miehelle. Asettaessani kasvoja ohjailevia luita naishahmolle, kohdistin ne kasvogeometrian vertekseihin. Tällä tekniikalla luut oli helppo sijoittaa mieshahmolla oikeisiin, vastaaviin paikkoihin. Lopullisessa luurangossa on kaula, pää ja leukaluiden lisäksi 65 kasvonilmeitä ohjailevaa luuta. Ylä- ja alahampaita voi liikuttaa erikseen omilla luillaan. Tein lisäksi kolme luuta kielen liikutteluun ja luut ohjailemaan silmiä sekä pupillien kokoa.



Kuvio 12. Naishahmon luuranko.



Kuvio 13. Mieshahmon luuranko.

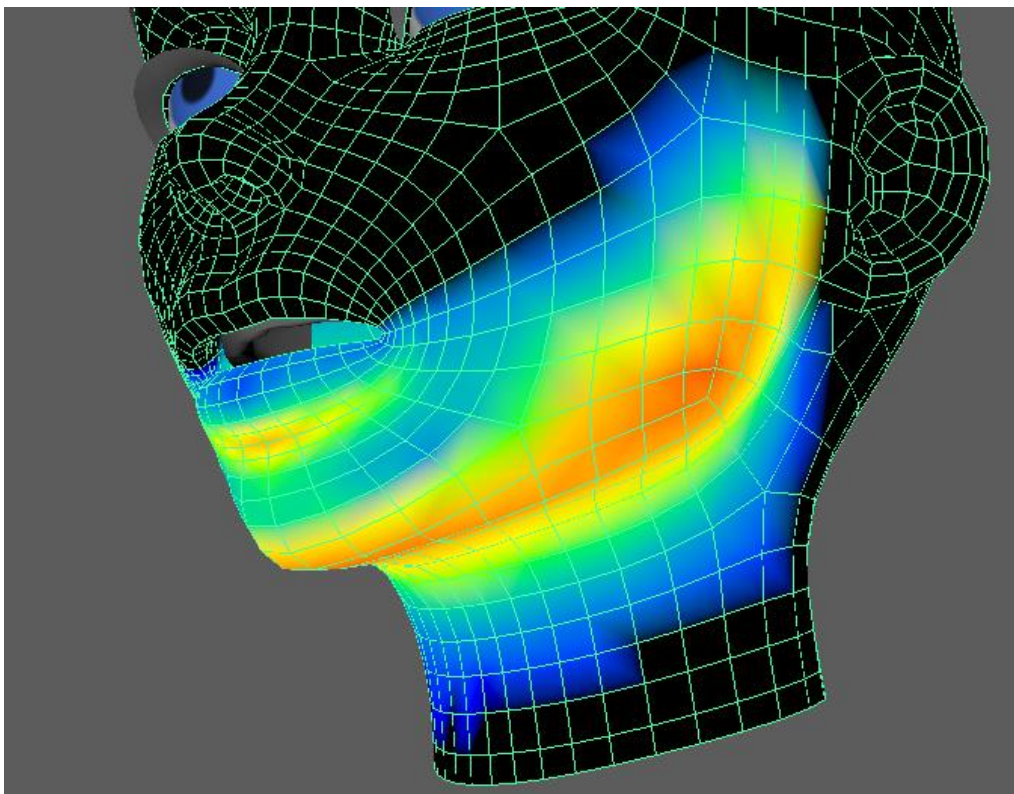
Lisäsin kieltä ja pupillien kokoa ohjaavat luut hierarkiaan vasta myöhemmässä vaiheessa. Tehtäessä muutoksia hierarkiaan oli huomioitava, että ne tehdään samalla tavalla kumpaankin luurankoon ja että luiden järjestys pysyy samana. Luuhierarkian muuttaminen on suoraviivaisempaa kuin geometrian, koska siinä ei ole samanlaista piilotettua monimutkaisuutta ja luut voi liikutella hierarkiassa oikeaan järjestykseen käsin.

Luuobjektit on hyvä nimetä johdonmukaisella tavalla, ja kahdessa luurangossa toisiaan vastaavat luut on myös hyvä pitää samannimisinä. Tämä helpottaa riggauksen myöhempiä vaiheita ja animointia. Unityä käytettäessä se on myös blendauksen toiminnan kannalta välttämätöntä, koska Unity järjestää luuhierarkian uudelleen aakkosjärjestykseen. Luiden suhteet toisiinsa hierarkiassa pysyvät samana, mutta samalla hierarkiatasolla olevien luiden keskinäinen järjestys voi muuttua. Jos luita ei ole nimetty samalla tavalla, niiden järjestys voi päätyä olemaan erilainen luurankojen välillä, joka on ei-haluttu lopputulos.

3.3.2 Skinnaaminen

Skinnaamisella tarkoitetaan prosessia, jossa 3D-mallin geometria kytetään luurankoon skinin avulla. Tärkeä osa skinnausta on vaikuttavien luiden painoarvojen määrittäminen mallin verteksille. Painot voi määrittää jokaiselle verteksille erikseen, ja tästä voi olla

hyötyä, jos painoarvojen pitää olla jotakin hyvin spesifistä. Tyypillisempää on kuitenkin ”maalata” painoarvot geometrian pinnalle (weight painting). Jos luurangossa on paljon luita, painojen maalaaminen voi olla sekavaa ja kontrollin säilyttäminen vaikeaa. Siksi prosessi on hyvä jakaa osiin skinnaamalla malli ensin vain muutama luuhun ja lisäämällä skiniin vaikuttavia luita työn edetessä (Rigging Dojo 2016). Symmetrisellä hahmolla skinnausta helpottaa myös se, että skinin painot voi peilata puolelta toiselle, mikä käytännössä puolittaa työmäärän.



Kuvio 14. Leukaluun vaikutuksen painoarvot havainnollistettuna verteksin väreillä Mayassa.

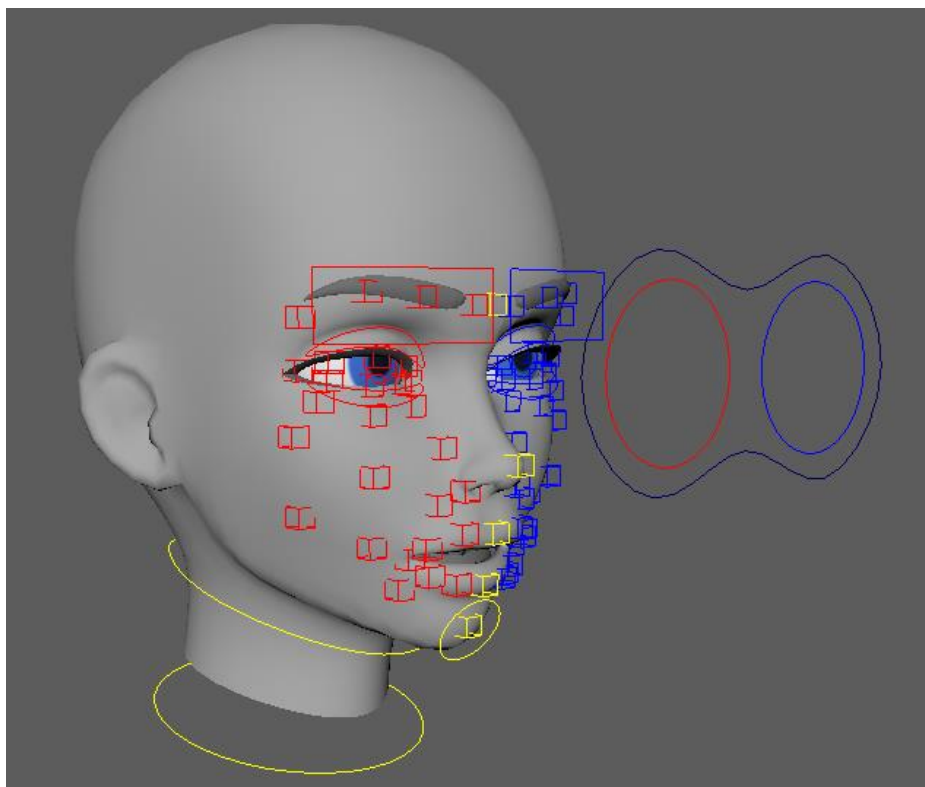
Blendausjärjestelmä vaati toimiakseen vain pohja- eli naishahmon skinnaamisen, eikä mieshahmoa ollut tarpeen skinnata lainkaan. Koska molemmilla hahmoilla oli sama topologia ja UV-koordinaatit, skinin painot olisi kuitenkin myös haluttaessa ollut helppo siirtää hahmolta toiselle esimerkiksi Mayan Import/Export Skin Weights -työkalulla. Tämä olisi kätevää, jos haluaisi myöhemmin animoida naishahmon sijasta miestä.

3.3.3 Animaatio-ohjaimet

Animaatio-ohjaimet ovat objekteja, jotka toimivat ylimääräisinä abstraktitasoina luurangon päällä piilottaen luiden omat transformaatiot ja ylimääräisen monimutkaisuuden, ja ne mahdollistavat rigin animoinnin koskematta suoraan itse

luurankoon. Ne eivät ole välttämätön osa rigiä, mutta niiden tarkoituksena on tehdä animoinnista helpompaa ja mukavampaa, ja mahdollistavat esimerkiksi inverse kinematics -animoinnin, ja useamman luun tai kokonaisten luuketjujen liikuttelun samanaikaisesti. Monimutkaisemmissa rigeissä ohjainten käyttö estää myös animaattoria vahingossa rikkomasta rigiä kajoamalla johonkin sellaiseen osaan rigiä, jota ei ole tarkoitus animoida (Pluralsight 2014).

Oma rigini oli sen verran yksinkertainen, että sitä olisi voinut hyvin animoida liikuttelemalla pelkkiä luita. Päätin kuitenkin tehdä ohjaimet, jotta saisin helposti palautettua rigin takaisin bind poseen nollaamalla ohjainten transformaatiot. Ohjaimet mahdollistivat myös kulmia ja silmäluomia ohjaavien luiden ryhmittelyn ja liikuttelun yhdessä ja helpottivat silmien animointia. Silmät osoittavat kohti silmäohjaimia, jolloin hahmo on helppo saada katsomaan haluamaansa suuntaan ohjaimia liikuttelemalla.



Kuvio 15. Naishahmon rigin ohjaimet.

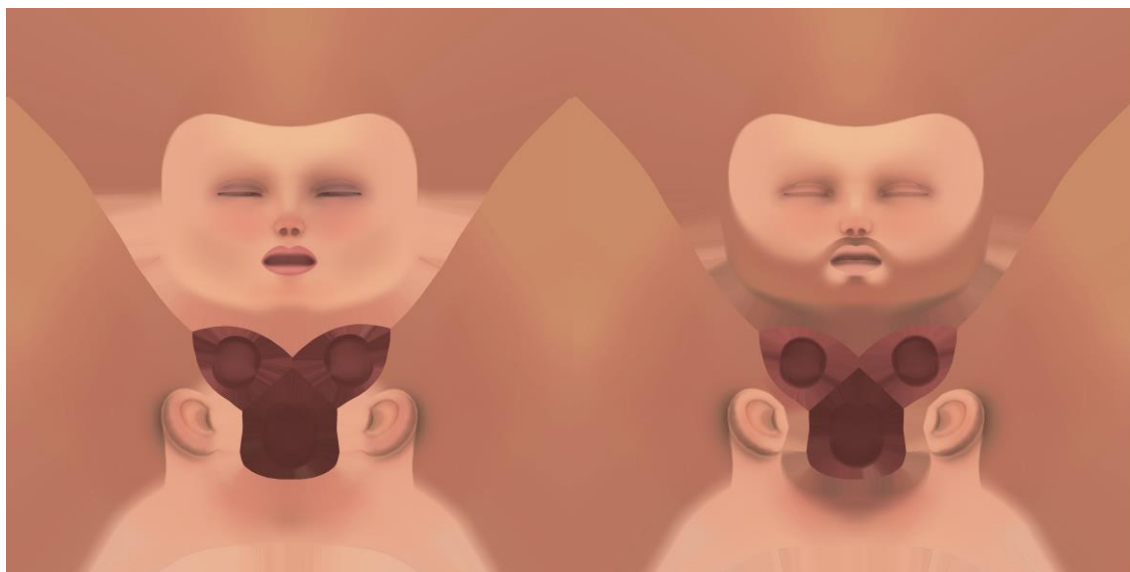
Koska animoin vain naishahmoa, tein ohjaimet vain naiselle (kuvio 15). Jos olisin halunnut animoida myös miestä, olisin joutunut tekemään mieshahmolle omat ohjaimet. Toisin kuin UV-koordinaatit tai skinin painot, ohjaimet eivät ole siirrettävissä hahmolta

toiselle yhdellä napinpainautuksella. Ohjaintentekoprosessi olisi kuitenkin mahdollista halutessaan automatisoida skriptin avulla. Muussa tapauksessa se olisi tehtävä käsin.

3.4 Teksturointi

Hahmoilla oli kolme teksturoitua osaa: iho, silmät ja suun sisäosat. Silmät teksturoin yksinkertaisesti maalaamalla. Iholle ja suun sisäosille leivoin lisäksi normal ja ambient occlusion mapit. Vain ihotekstuureille leivoin myös ns. point light mapin. Asetin päägeometrian ympärille Blenderissä muutaman pistemäisen valonlähteen, ja leivoin niiden vaikutukset tekstuuriin. Seurasin ihon teksturoinnissa Dota 2 Workshopin tutoriaalia (Valve 2017).

Koska blendattavilla hahmoilla on samat UV-koordinaatit, niiden on mahdollista käyttää samoja tekstuureita. Korostaakseni naisen feminiinisyyttä ja miehen maskuliinisuutta tein hahmoille kuitenkin omat kasvotekstuurinsa (kuvio 16). Annoin naiselle meikkiä ja miehelle parransängen. Maalasin ensin molemmille yhteisen pohjaihotekstuurin, jonka päälle lisäsin sitten hahmojen yksilölliset piirteet. Leivoin mapit myös hahmoille erikseen.



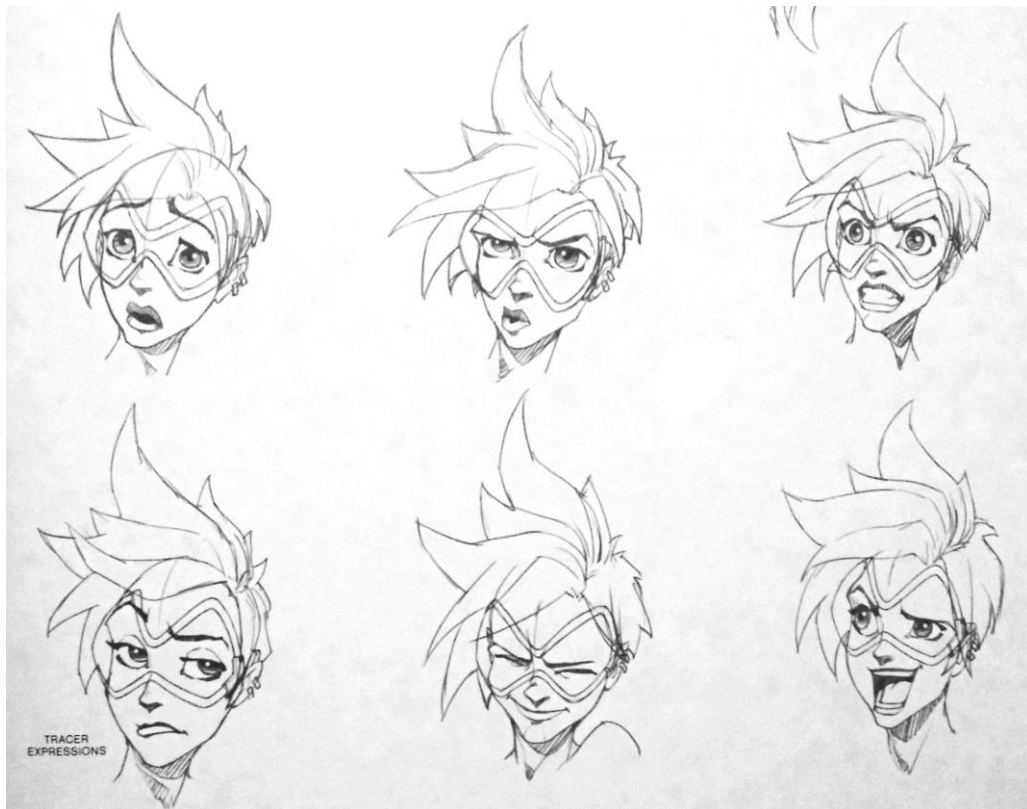
Kuvio 16. Naisen ja miehen kasvotekstuurit, joissa on mukana ambient occlusion ja point light mapit.

Annoin hahmoille myös eriväriset silmät; naiselle pinkit ja miehelle siniset. Silmien väri havainnollistaa blendauksen senhetkistä painoarvoa. Painoarvon ollessa 0.5, silmät ovat violetit. Mitä sinisemmät, sitä miehempi, ja mitä pinkimmät, sitä naisempi. Maalasin

silmät ensin sinisinä ja tein sitten pinkin variantin säätämällä silmätekstuurin sävyä, kylläisyyttä ja kirkkautta.

3.5 Animointi

Animaatio ei ollut projektin fokus, mutta se oli tärkeä osa sitä, koska halusin tutkia, miten blendausjärjestelmä sallii tunnetilojen ja ilmeiden välittymisen hahmolta toiselle. Pyrin siksi tekemään hyvin yksinkertaisen ja nopeasti toteutettavan animaation, josta välittyisi kuitenkin useita eri ilmeitä ja tunnetiloja. Tekemäni animaatio on kokoelma vaihtuvia ilmeitä ja koostuu kahdestatoista eri ilmeestä. Käytin referenssinä Disneyn hahmoluonnoksia ja ilmeluonnoksia kirjasta *The Art of Overwatch*, jotka vaikuttivat myös olleen hyvin Disney-inspiroituneita (kuvio 17). Animoin pelkästään naishahmoa, mutta koska ilmeiden tuli toimia molemmilla hahmoilla, otin mallia sekä mies- että naishahmojen ilmeistä. Järjestelmän luonne ei sinänsä estä molempien hahmojen animoimista, mutta päätin toimia näin, koska se säästi työtä ja oli kokeilun kannalta tehokkaampaa.



Kuvio 17. Tracerin ilmeitä (*The Art of Overwatch* 2017, 291).

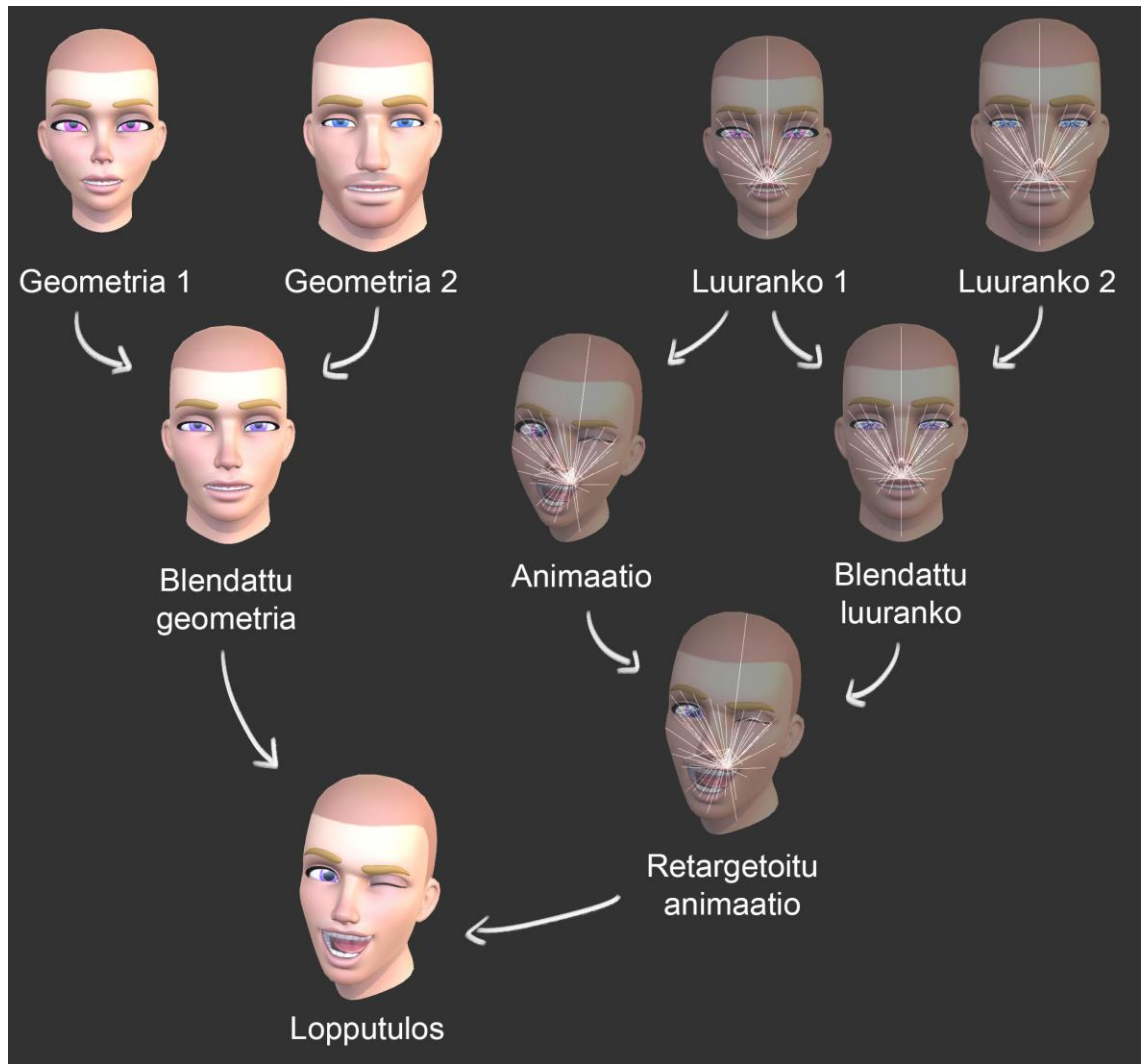
4 Blendauksen toteutus

Toteutin kahden hahmon välisen blendauksen Unity-pelimoottorissa. Unity on laajasti käytetty, hyvin dokumentoitu ja sen ympärille on kertynyt laaja yhteisö, joka tekee ongelmanratkaisusta helppoa. Minulla oli myös Unitystä jonkin verran aikaisempaa kokemusta. Unityssa on kuitenkin omat ongelmansa, jotka toimivat kompastuskivinä. Sain ongelmat ohitettua tavoilla, jotka soveltuivat omiin testitarkoituksiini, mutta eivät välttämättä ole optimaalisia.

Kerron tässä luvussa systeemin toiminnasta ja toteutuksesta eli projektin ”ohjelmointiosuudesta”. Selitän ensin systeemin toiminnan periaatteet yksinkertaistetusti ja syvennyn sen jälkeen tarkemmin järjestelmän yksityiskohtiin. Pysin kertomaan järjestelmän toiminnasta tavalla, joka on pelimoottorista ja ohjelmointikielestä riippumaton. Joistakin työvaiheista on kuitenkin mahdotonta kertoa viittaamatta käyttämäni pelimoottorin toimintaan ja tapaan säilöä dataa, sillä se määrittä osaltaan sen, miten järjestelmä tuli toteuttaa.

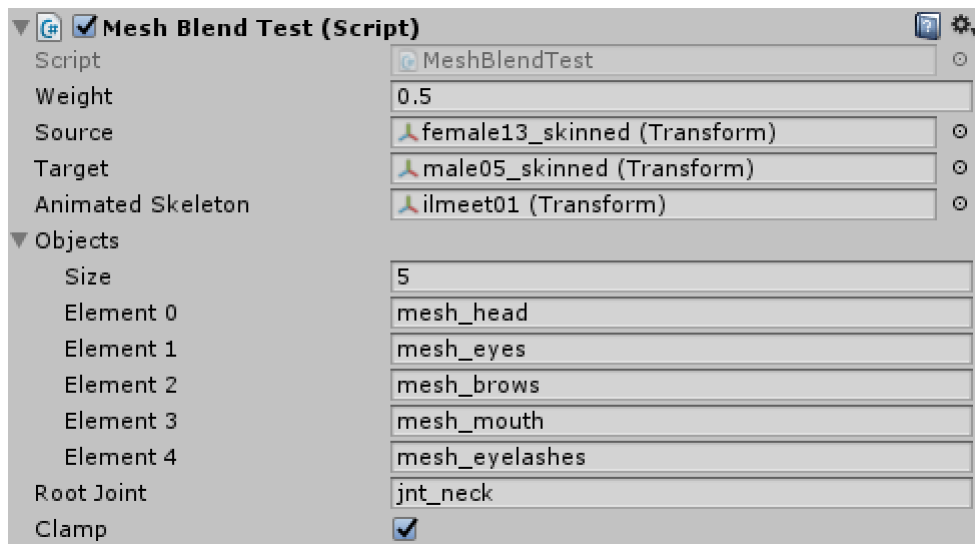
4.1 Yleiskatsaus

Järjestelmän toiminta on kaksivaiheinen: Se blendaa ensin lähde- ja kohdehahmojen geometriaa ja luurankoja perusasunnoissaan annetulla painoarvolla. Blendauksen lopputuloksena syntyvälle hahmole siirretään eli retargetoidaan sen jälkeen lähdehahmolle tehty animaatio, ja tuloksena on animoitu blendattu hahmo (kuvio 18). Kuten olen aiemmin maininnut, käytin projektissani lähdehahmona naista ja kohdehahmona miestä.



Kuvio 18. Järjestelmän suurpiirteinen toimintaperiaate.

Geometrian ja luurankojen blendaus tapahtuu lineaarisesti interpoloimalla verteksejä sekä luiden transformaatioita. Blendatun luurangon asentoa käytetään sitten blendatun hahmon bind posena. Blendauksen suorittaa kirjoittamani skripti, jolle syötetään haluttu blendauksen painoarvo (weight), jota on mahdollista muuttaa reaaliajassa (kuvio 19). Painoarvo on oletuksena rajoitettu välille $\{0, 1\}$, jossa 0 on täysin nainen ja 1 täysin mies. Rajoittimen (clamp) voi kuitenkin poistaa, jolloin painoarvon on mahdollista olla alle 0 tai yli 1, jolloin blendaus toimii interpoloinnin sijaan ekstrapoloinnilla.

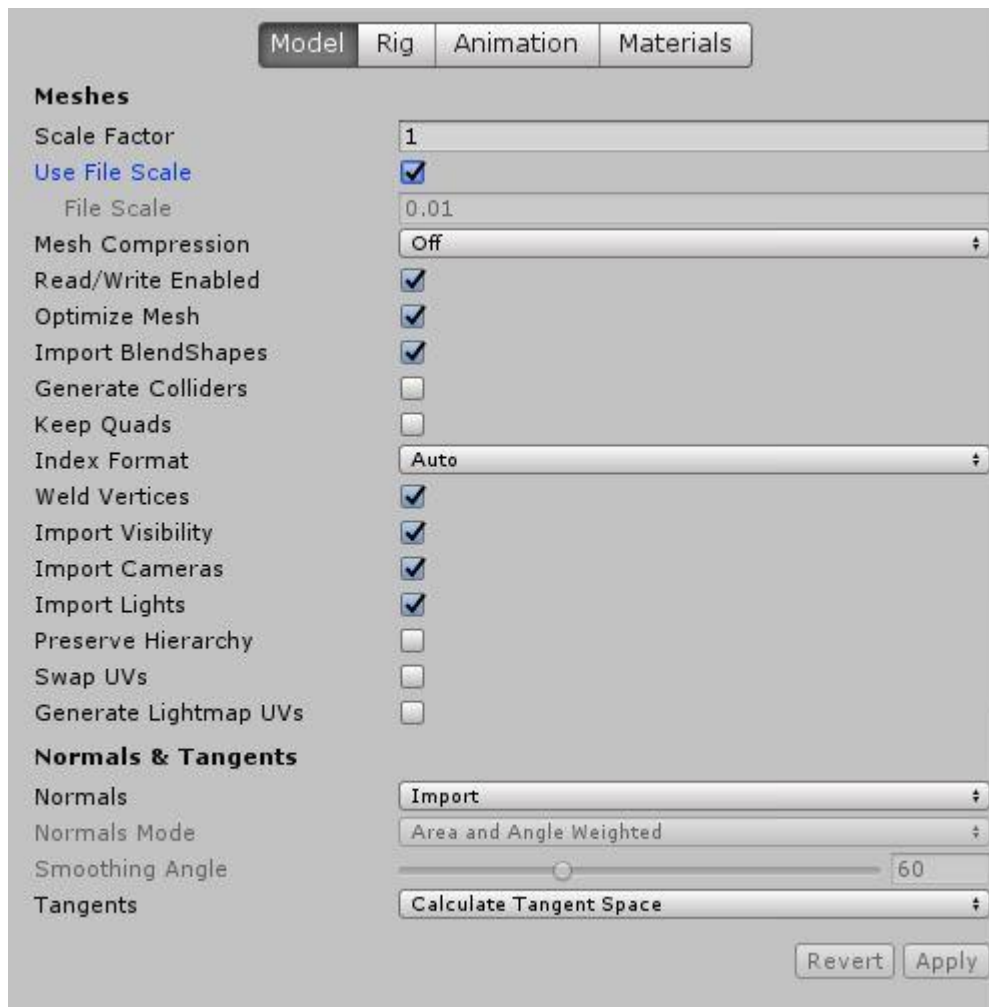


Kuvio 19. Kirjoittamani skriptin asetukset Unity-editorissa.

Skripti ottaa painoarvon lisäksi syötteinä lähde- ja kohdehahmot prefabeina (source ja target), animoidun luurangon prefabina (animated skeleton), listan blendattavien geometrioiden nimiä (objects) sekä luurangon juuriluun nimen (root joint). Jotta skripti toimisi oikein, geometrioiden ja juuriluun tulee olla samannimisiä molemmilla hahmoilla.

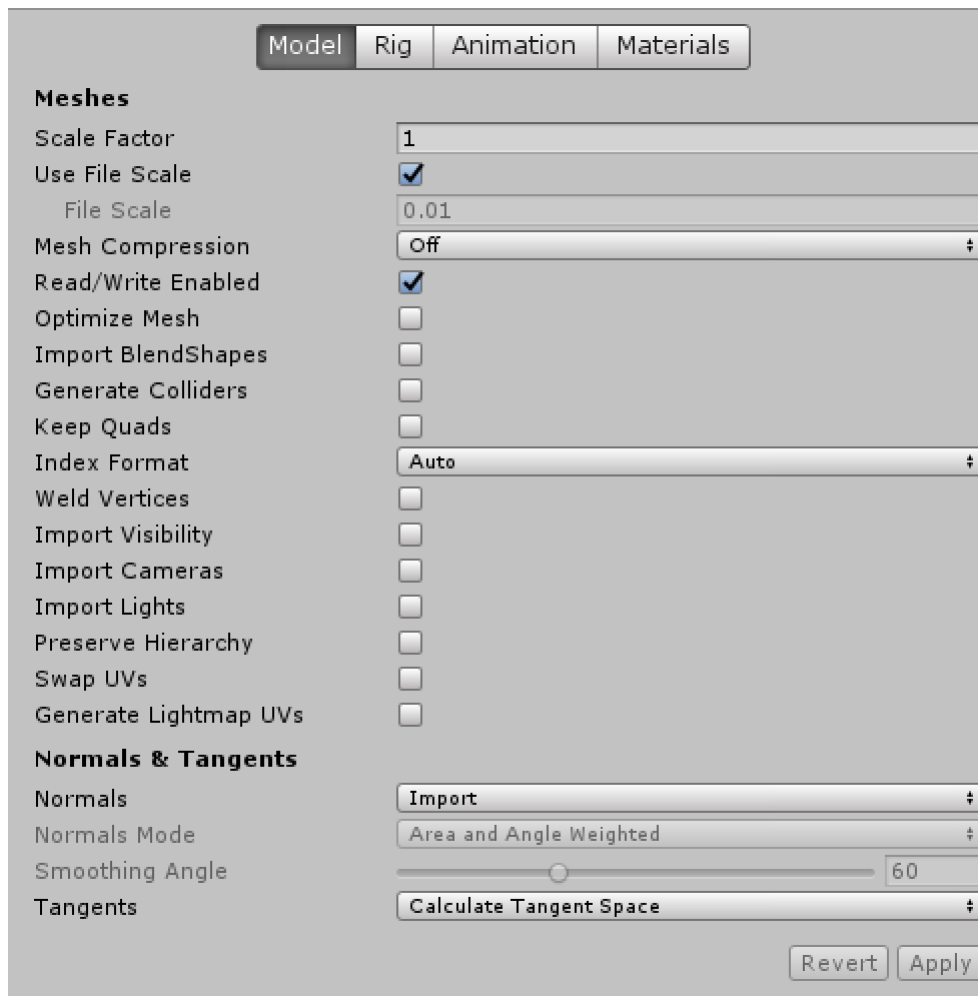
4.2 Hahmojen tuominen Unityyn

Unityssä on 3D-malleille paljon tuontiasetuksia, joista osa voi vaikuttaa polygoniverkon rakenteeseen (kuvio 20). Jotta lähde- ja kohdemallien rakenteet säilyisivät varmasti yhteneväisinä, tällaiset asetukset on otettava pois käytöstä. Oletusasetuksista tämä tarkoittaa Optimize Mesh ja Weld Vertices-asetuksien poistamista. Optimize Mesh järjestää mallin kolmiot uudelleen ja saattaa tehdä sen hahmoille eri tavalla. Weld Vertices yhdistää sellaiset pisteet, jotka jakavat saman position, ja vaikuttaa siten verteksin määrään. Omissa malleissani ei tällaisia pisteitä ollut, mutta poistin asetuksen silti varmuuden vuoksi.



Kuvio 20. 3D-mallin oletustuontiasetukset Unityssä.

Koska hahmoillani ei ollut blend shapeja, otin Import BlendShapes -asetuksen pois käytöstä. Poistin myös asetukset Import Visibility, Import Cameras, ja Import Lights. Verteksien normaalit tuodaan oletuksena Unityyn tiedostosta, ja pidin tämän asetuksen. Myös pinnansuuntaiset tangenttivektorit voi halutessaan tuoda tiedostosta, mutta pidin tässäkin oletusasetuksen "Calculate Tangent Space", jolloin tangenttivektorit laskettiin Unityn päässä normaalivektoreiden ja UV-koordinaattien perusteella. Koska malleissa oli samat UV-koordinaatit, lasketut tangenttivektorit olivat molemmissa johdonmukaiset. Read/Write Enabled tulee pitää päällä, koska järjestelmän pitää pystyä lukemaan mallien dataa.



Kuvio 21. Mallien lopulliset tuontiasetukset.

Kun 3D-malli tuodaan Unityyn, siitä luodaan automaattisesti ns. prefab (Unity 2017). Prefab on ikään kuin sapluuna valmiiksi määritellylle peliobjektille, jonka voi instansoida skeneen. Tällöin jokaista peliobjektia ei ole tarpeen rakentaa alusta asti uudelleen. Luut ovat prefabissa hierarkia peliobjekteja, joilla on pelkästään pakollinen transform -komponentti. Jokainen 3D-mallin erillinen osa tai "mesh" saa prefabissa oman lapsiobjektin, jolla on joko Mesh Renderer tai Skinned Mesh Renderer -komponentti, riippuen siitä onko meshiä skinnattu vai ei. Tämä komponentti on vastuussa kyseisen 3D-mallin osan piirtämisestä näytölle. Kunkin skinnatun meshin Skinned Mesh Renderer -komponentti pitää kirjaa niistä luista, joihin se on skinnattu, ja näiden luiden bind pose-aseennoista. Nämä on listattu muistissa samassa järjestyksessä.

4.3 Geometrioiden blendaus

Geometrioiden blendaus suoritetaan lineaarisesti interpoloimalla lähde- ja kohdegeometrioiden verteksejä. Interpoloitavat attribuutit ovat paikkavektori, normaalivektori ja tangenttivektori. UV-koordinaatteja ei ollut tarpeen interpoloida, koska ne olivat molemmilla hahmoilla samat. Lineaarinen interpolaatio, jossa painoarvo ei ole rajattu välille $\{0, 1\}$, lasketaan kaavalla $\mathbf{a} + w(\mathbf{b} - \mathbf{a})$, jossa \mathbf{a} on jokin lähdegeometrian verteksin attribuutti, \mathbf{b} on vastaava kohdegeometrian verteksin attribuutti ja w blendauksen painoarvo. $\mathbf{b} - \mathbf{a}$ on attribuuttien välinen muutos Δ_{ab} . Olkoon Δ_M vastaavasti lähde- ja kohdegeometrioiden välinen muutos kokonaisuudessaan (kuvio 22).



Kuvio 22. Geometrioiden välinen muutos lasketaan kohde- ja lähdegeometrioiden erotuksena.

Skripti tekee ensin kopion lähdegeometriasta. Varoin tekemästä mitään muutoksia alkuperäiseen lähdegeometriaan, koska se rikkoisi järjestelmän. Tämän jälkeen skripti käy lähde- ja kohdehahmon meshien verteksit läpi järjestyksessä ja laskee toisiaan vastaavien verteksin väliset attribuuttien muutokset.

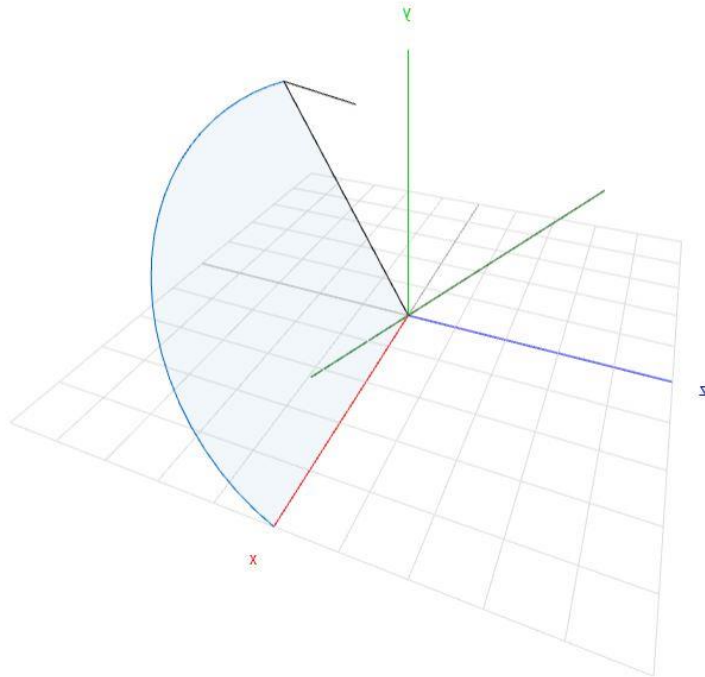


Kuvio 23. Blendattu lopputulos saadaan lisäämällä kopioituun lähdegeometriaan lähde- ja kohdegeometrioiden välinen muutos kerrottuna blendauksen painoarvolla.

Lasketut muutokset lisätään kopioidun lähdegeometrian vertekseihin kerrottuna blendauksen painoarvolla, ja tuloksena on blendattu geometria (kuvio 23). Operaatio suoritetaan uudelleen joka kerta, kun blendauksen painoarvoa muutetaan.

4.4 Luurankojen blendaus

Luurankojen blendaaminen toimii hyvin samaan tapaan kuin geometrioiden. Perusperiaate on sama: Tehdään kopio lähdehahmon luurangosta, ja lisätään siihen luiden transformaatioiden väliset muutokset. Kuten vertekseillä, luilla on myös kolme attribuuttia, joille muutokset tulee laskea: Translaatio, rotaatio ja skaala. Translaatio ja skaala ovat vektoreita, joten niiden väliset muutokset lasketaan vektoreiden vähennyslaskulla samaan tapaan kuin verteksien attribuuteille.



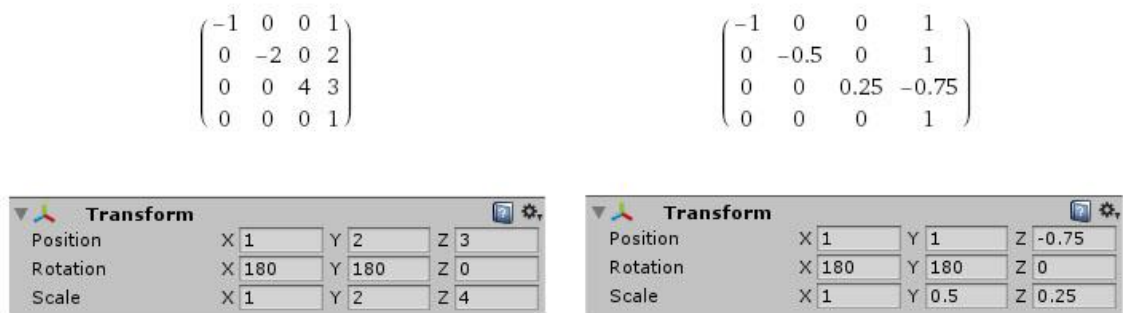
Kuvio 24. Visualisointi kvaterniosta, rotaatio tummanvihreän akselin ympäri (quaternions.online).

Rotaatio on oma erityistapauksensa. Se on Unityssä vektorin sijaan kuvattu kvaterniona. Kvaterniot ovat neliulotteinen versio kompleksiluvuista, mutta niitä voi käsitellä skalaarin ja vektorin yhdistelmänä (Shoemake 1993). Kvaternioita, joiden normi ("pituus") on yksi, sanotaan yksikkökvaternioiksi, ja niitä käytetään kuvaamaan rotaatioita kolmessa ulottuvuudessa mielivaltaisen akselin ympäri (kuvio 24). Kvaternioiden käytöstä on hyötyä mm. siinä, että sillä välttää 3D-animaattorin arkkivihollisen, Gimbal Lockin. Kahden kvaternion välistä rotaation muutosta ei voi laskea vähennyslaskulla samaan tapaan kuin vektoreilla. Sen voi kuitenkin tehdä kertomalla kohderotaatio lähderotaation käänteiskvaterniolla:

To measure the distance from rotation Q to Q' , look at $Q' \cdot Q^{-1}$.
 Since $Q' = (Q' \cdot Q^{-1}) \cdot Q$, the composition $Q' \cdot Q^{-1}$ is the rotation connecting them.
 (Shoemake 1993, 9)

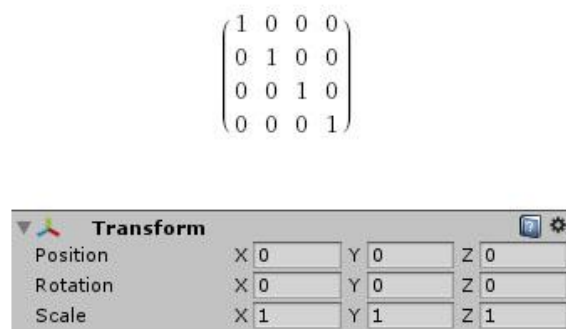
4.5 Bind pose

Jotta blendattu hahmo deformoituisi oikein, sen luurangolle on määritettävä bind pose. Luiden bind poseja käsitellään Unityssä transformaatiomatriiseina. Transformaation esittäminen matriiseina on yleinen tapa säilöä ja käsitellä translaatiota, rotaatiota ja skaalaa yhtenä yksikkönä. Bind posen transformaatiomatriisit ovat luiden maailma-transformaatioista johdettujen matriisien inverssejä. Transformaatiomatriisin inverssi eli käänteismatriisi kuvaa vastakkaista transformaatiota (kuvio 25).



Kuvio 25. Eräs transformaatiomatriisi (vas.) ja sen käänteismatriisi (oik.)

Kertomalla luun maailma-transformaatiosta johdettu transformaatiomatriisi ja bind posen invertoitu matriisi keskenään saadaan tuloksena luun nykyisen asennon ja bind posen välinen muutos. Skinnattu geometria deformoituu tämän muutoksen verran. Luun ollessa valmiiksi bind posessa tuloksena on yksikkömatriisi, joka kuvaa nollatransformaatiota (kuvio 26). Tällöin geometria ei deformoidu.

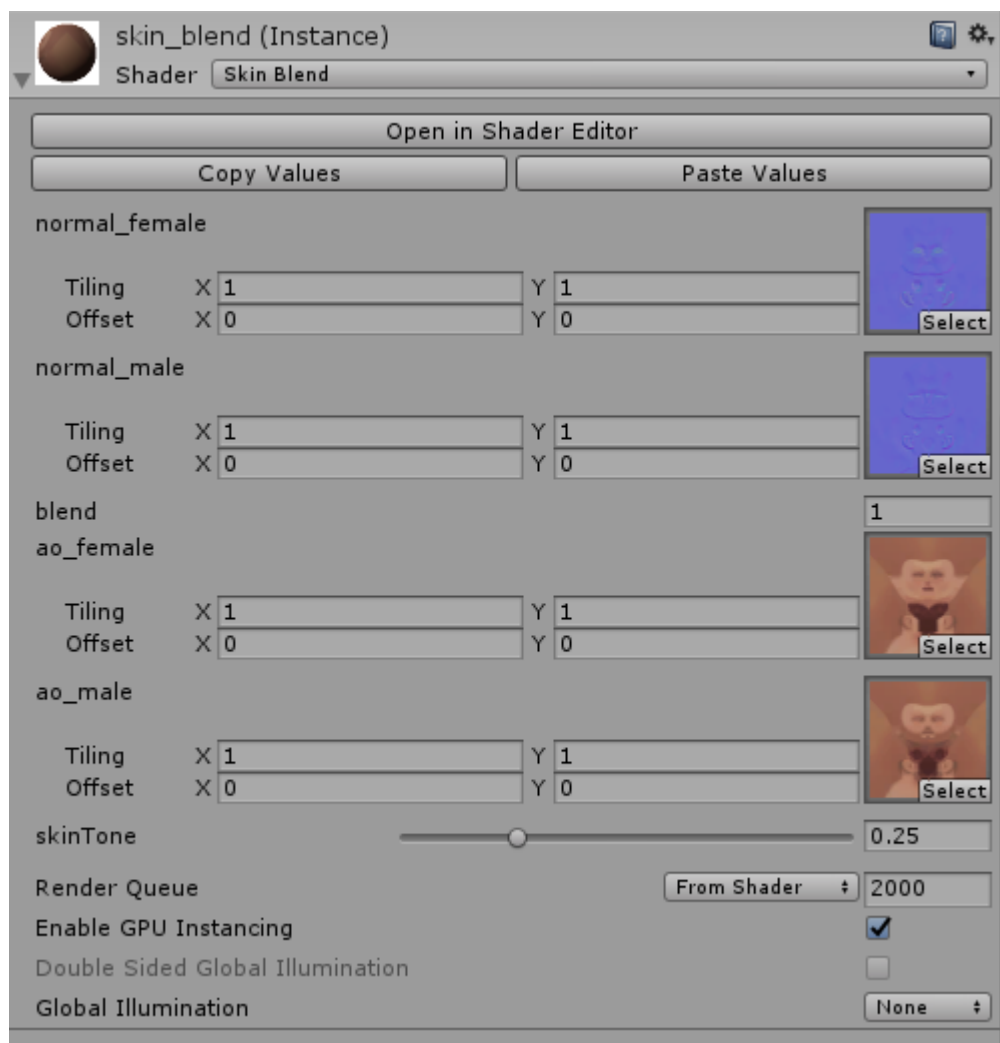


Kuvio 26. Yksikkö- eli identiteettimatriisi.

Tämän tiedon varjolla bind pose -matriisien laskeminen blendatun luurangon luille oli helppoa. Blendattu luuranko oli valmiiksi bind pose -asennossaan, joten riitti, että luiden maailma-transformaatioille muodostettiin käänteismatriisit.

4.6 Tekstuureiden blendaus

Järjestelmä blendaa lähde- ja kohdehahmojen ihotekstuureja ja normal mappeja interpoloimalla pikseleiden värejä. Tein tätä tarkoitusta varten ihomateriaalille oman shaderin eli varjostinohjelman, joka ottaa syötteenä molempien hahmojen tekstuurit ja normal mapit sekä blendauksen painoarvon (kuvio 27). Kun blendausskriptin painoarvoa muutetaan, se päivittää automaattisesti ihomateriaalin painoarvon.



Kuvio 27. Ihomateriaalin asetukset Unityssä. SkinTone säätää ihon tummuusastetta.

Halusin mies- ja naishahmon keskiarvon näyttävän mahdollisimman androgyyniltä. Ongelmia tuotti miehen ihotekstuuriin maalaamani parransänki, joka näkyi painoarvolla 0.5 jo sen verran, että hahmo näytti varsin miehiseltä. Harkitsin kolmannen tekstuurin käyttämistä puoliväli tilanteelle, mutta päädyin lopulta yksinkertaisempaan ratkaisuun: korotin blendauksen painoarvon shaderin sisällä toiseen potenssiinsa. Tällöin todellisten painoarvojen ollessa 0.25, 0.5 ja 0.75 tekstuuri interpoloitui painoarvoilla 0.0625, 0.25 ja 0.5625 (ja niin edelleen). Tämä sai tekstuurin pysymään naismaisena pidempään ja sängön ilmestymään vasta korkeammilla painoarvoilla.

4.7 Lopputuloksen valmistelu

Jotta blendatun hahmomallin voisi animoida ja piirtää näytölle, jokainen sen osa tarvitsee oman peliobjektinsa, jossa on Skinned Mesh Renderer -komponentti. Skinned Mesh Renderer -komponentille on syötettävä luut, joihin mesh on skinnattu sekä niiden bind poset ja skinin painot. Animointia varten on myös hyvä tehdä luurangosta uusi kopio. Helpoimmalla pääsee, kun instansoi naishahmon Prefabin, jossa kaikki tarvittava on jo valmiina. Tämän jälkeen instanssin Skinned Mesh Renderer -komponenttien mesh-datan voi korvata blendatulla geometrialla ja luiden transformaatiot blendatun luurangon luiden transformaatioilla. Lopuksi vanhat bind poset on korvattava blendatun luurangon luiden bind poseilla. Tässä kohtaa on huomioitava, että Skinned Mesh Renderer -komponenttiin tallennetut luut ja bind poset ovat todennäköisesti eri järjestyksessä kuin luuhierarkia. Jos meshiä ei ole skinnattu koko luurankoon, välistä myös puuttuu luita.

4.8 Animaation retargetointi

Yhdelle hahmolle tehdyt animaatiot toimivat harvoin täydellisesti toisella hahmolla. Mitä erilaisempia hahmot ovat, sitä huonommin toisen animaatiot toimivat toisella. Tämä johtuu siitä, että animaatioita käsitellään absoluuttisina transformaatioina, jolloin luurankojen välisiä mittasuhte-eroja ei oteta huomioon. Animoitavan hahmon luuranko pakotetaan sen hahmon muotoon, jolle animaatio on tehty. Tämä aiheuttaa ei-haluttuja deformaatioita (kuvio 28). Jos hahmoilla on erilaiset luuhierarkiat, toisen animaatiot eivät toimi toisella lainkaan, koska luille ei löydy suoria vastineita.



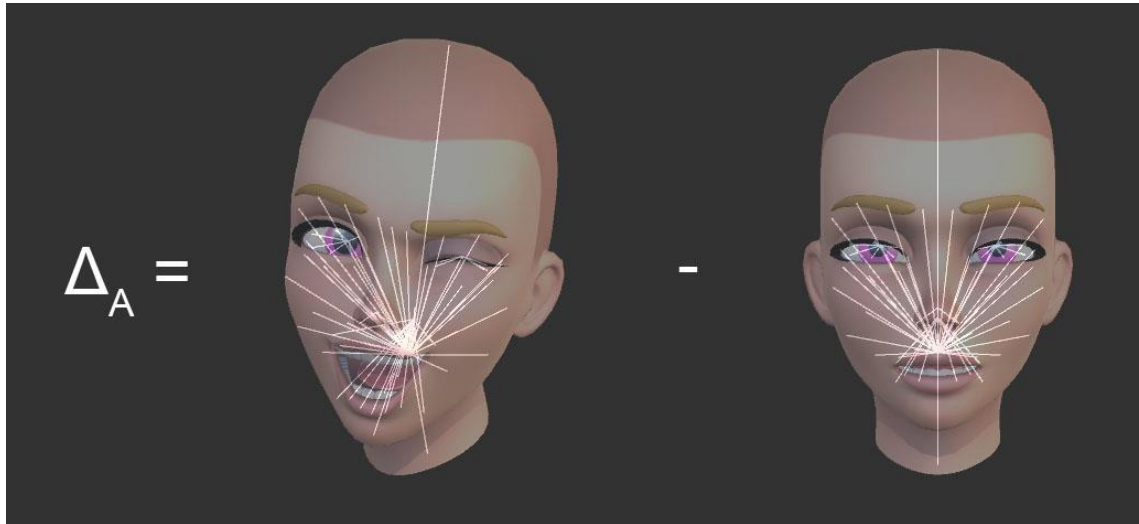
Kuvio 28. Retargetoimaton animaatio tuottaa muilla kuin naishahmolla ei-halutun lopputuloksen.

Koska tein animaatiot naishahmolle, ne toimivat sellaisenaan vain naishahmolla. Siksi animaatiot oli retargetoitava blendatulle hahmolle. Retargetoinnilla eli uudelleenkohdistuksella tarkoitetaan animaation siirtämistä hahmolta toiselle niin, että luurankojen väliset erot otetaan huomioon. Koska blendaus tapahtuu reaaliajassa, myös retargetoinnin oli tapahduttava reaaliajassa. Kirjoitin tätä varten oman skriptin.

Animaatioiden retargetointiin on olemassa useita eri metodeja. Yksinkertaisimmillaan retargetointi voi olla sitä, että luiden translaatiot vain sivuutetaan animaatiossa täysin, jolloin pelkät rotaatiot animoituvat. Tämä säilyttää luurangon mittasuhteet eri hahmoilla, mutta toimii vain, jos luuranko on animoitu pelkillä rotaatioilla (Epic Games 2018). Monimutkaisimmillaan retargetointi voi siirtää animaatiot kokonaan erilaiselle luurangolle, jolla on samankaltainen rakenne. Tähän kykenee esimerkiksi Autodeskin HumanIK, joka mahdollistaa animaatioiden retargetoinnin ihmismäisten hahmojen välillä riippumatta luuhierarkiasta. Koska retargetoinnin tuli tapahtua reaaliajassa, oma retargetointijärjestelmäni ei voinut olla liian laskennallisesti monimutkainen. Pelkkä translaatioiden sivuuttaminen ei myöskään tullut kysymykseen, koska hahmojen kasvot vaativat luiden translaatioiden animoimista.

Päädyn ratkaisuun, jossa retargetointi tapahtuu muuttamalla animaatioidatan absoluuttiset transformaatiot bind poseen nähden suhteellisiksi. Naishahmon animoitu luuranko instansoidaan alussa skeneen ohjaamaan animaatiota. Ohjelma laskee sitten joka päivityssyklillä naishahmon luurangon bind posen ja senhetkisen asennon välisen

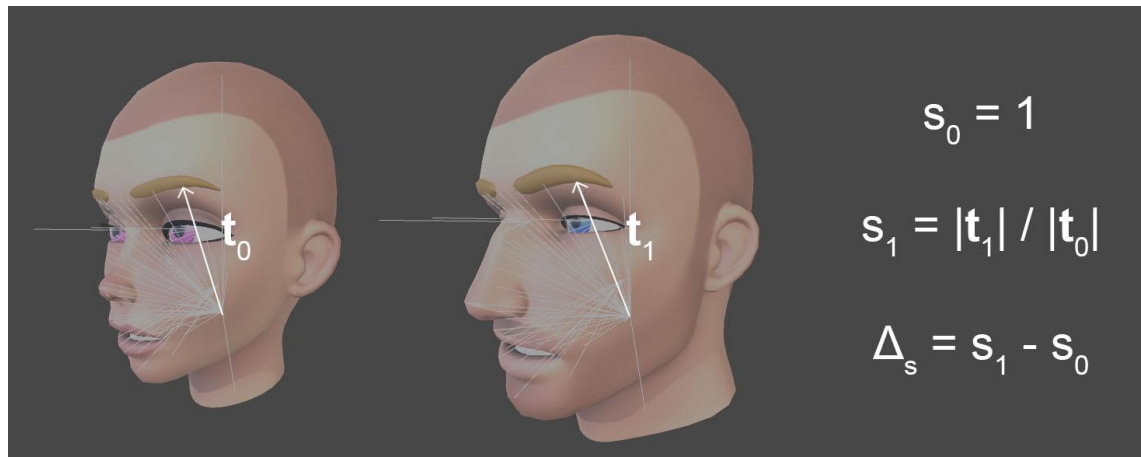
muutoksen Δ_A (kuvio 29). Animaatio retargetoituu, kun tämän muutoksen lisää blendatun luurangon bind pose -transformaatioihin. Alkuperäisen blendatun luurangon halutaan pysyvän bind posessaan, joten retargetointi suoritetaan lopputuloksen luurangolle.



Kuvio 29. Bind posen ja nykyisen asennon välinen muutos lasketaan joka päivityssyklillä.

Muutoksen lisääminen lopputuloksen luurankoon skaalaamattomana tuottaa jo varsin hyvän tuloksen. Ilmeissä on kuitenkin nais- ja mieshahmon välillä hienoja eroja johtuen hahmojen erilaisista mittasuhteista. Koska miehen kasvot ovat suuremmat, kasvoluiden liikkumat välimatkat ovat lyhyempiä suhteessa miehen kasvojen kokoon. Erityisen räikeä esimerkki on miehen suu, joka on paljon naishahmon suuta leveämpi eikä siksi mene skaalaamattomilla translaatioilla yhtä suppuun kuin naisella. Tämä aiheuttaa sen, että hienovaraisemmat ilmeet eivät välity oikein.

En saanut ongelmaa täysin korjattua, mutta paransin tilannetta ottamalla laskuihin mukaan luiden väliset pituuserot. Tämä ei ota huomioon tarkkoja mittasuhte-eroja, mutta on mielestäni hyvä yleispätevä ratkaisu. Laskin jokaiselle luulle mies- ja naishahmon välisen skaalakertoimen s_1 , joka saadaan laskemalla luiden lokaalien translaatiovektoreiden \mathbf{t}_1 ja \mathbf{t}_0 pituuksien välinen suhde. Luiden välinen skaalan muutos Δ_s lasketaan vähentämällä s_1 alkuperäisestä skaalasta s_0 . Koska skaalat lasketaan suhteessa naishahmon luurankoon, alkuperäinen skaala s_0 on kaikilla luilla 1 (kuvio 30).



Kuvio 30. Skaalakertoimet saadaan luiden lokaalien translaatiovektoreiden pituuksien suhteena.



Kuvio 31. Mieshahmoon retargetoitu ilme ilman skaalausta (kesk.) ja skaalauksen kanssa (oik.).

Skaaloja interpoloidaan blendauksen painoarvon mukaisesti kaavalla $s_0 + w\Delta_s$. Aiemmin lasketun luurangon asennon muutoksen Δ_A luiden translaatiot kerrotaan sitten interpoloiduilla skaaloilla. Kun skaalatut muutokset lisää lopputuloksen luurankoon, tuloksena on retargetoitu animaatio, jossa luiden skaalat on otettu huomioon ja ilmeet välittyvät paremmin hahmolta toiselle (kuvio 31).

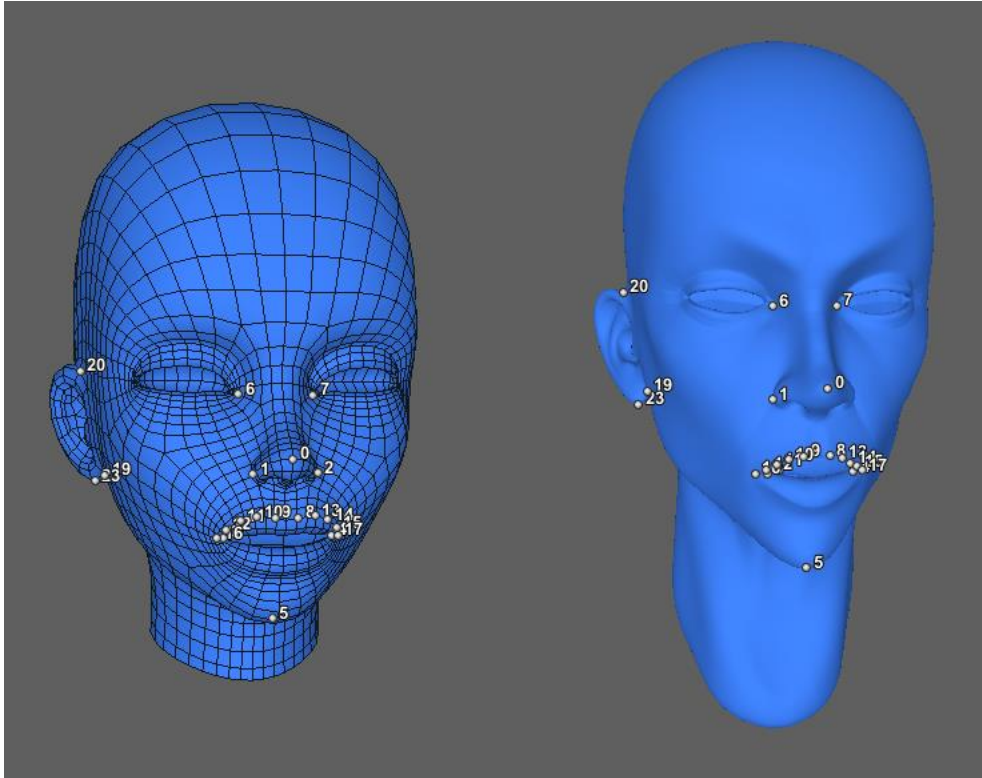
5 Jatkokehitysideoita

Koen, että kehittämälläni järjestelmällä on potentiaalia, jota en tämän projektin puitteissa saanut vielä täysin hyödynnettyä. Minulla onkin jo ideoita sille, miten järjestelmää voisi kehittää pidemmälle, ja siten laajentaa sen käyttömahdollisuuksia.

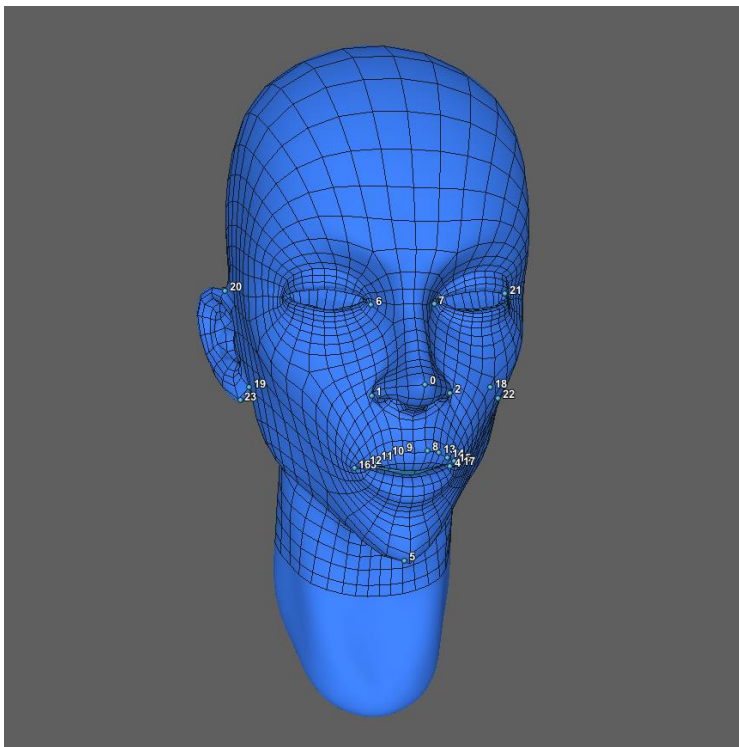
5.1 Wrap

Sain tätä opinnäytetyötä kirjoittaessani tietää ohjelmasta nimeltä R3DS Wrap. Wrap on erityisesti 3D-skannien kanssa työskentelyyn tarkoitettu ohjelma, joka mahdollistaa korkearesoluutioisen mallin ”paketoimisen” aiemmin mallinnettuun topologiaan. Tämä mahdollistaa työnkulun, jossa blendattavat hahmot mallinnetaan tai digitaalisesti veistetään täysin toisistaan erillään, ja haluttu topologia siirretään niille myöhemmin. Tämä helpottaisi mallinnusprosessia, varsinkin monen ihmisen projekteissa. Topologiaa olisi myös erittäin helppo jälkeenpäin muuttaa, koska sen voisi aina siirtää kaikille hahmoille uudelleen.

Wrapista on saatavilla 30 päivän kokeiluversio. Latasin sen ja päätin kokeilla tekemäni kasvotopologian siirtämistä eräälle vanhalle, keskeneräiselle hahmolle, josta oli olemassa vain korkearesoluutioinen versio. Ohjelma teki yllättävän hyvää työtä jo kylmiltään, ilman mitään ohjeistusta, kunhan mallit asetti suurin piirtein päällekkäin. Sillä oli kuitenkin vaikeuksia joillain kasvojen alueilla, kuten ylähuulen ja silmien alueilla. Lisäksi uuden topologian suun sisuksilla oli vaikeuksia pysyä suun sisäpuolella, oletettavasti koska vanhalla hahmollani suun sisäosat olivat paljon pienemmät.



Kuvio 32. Tähän projektiin tekemäni kasvotopologia, ja vanha hahmo, jolla ei ollut kunnollista topologiaa.



Kuvio 33. Kasvotopologia siirrettynä vanhalle hahmolle.

Käyttämällä Select Points -työkalua mallien pintaan voi merkitä pisteitä, joita Wrap käyttää apuna topologian siirtämisessä pyrkimällä kohdistamaan toisiaan vastaavat pisteet (kuvio 32). Ohjelmassa on myös mahdollista "jäädyttää" sellaiset polygonit, joiden ei halua projisoituvan ollenkaan. Tein tämän suun sisäosille. Tämän jälkeen lopputulos oli lähes täydellinen (kuvio 33). Pienemmät virheet on helppo korjata myöhemmin käsin.

5.2 Moniulotteisuus

Käsittelin projektissani blendaamista vain kahden hahmon välillä. Jo tämä tuottaa teoriassa äärettömän määrän erilaisia hahmoja, mutta ne kaikki voivat olla vain joko A tai B tai jotain niiden väliltä. Jos hahmo A olisi esimerkiksi luiseva mies ja hahmo B lihaksikas nainen, lihaksikkaan miehen tuottaminen olisi mahdotonta. Tämä vaatisi kahden eri "ulottuvuuden" - sukupuolen ja kehonrakenteen - blendaamisen erikseen. Järjestelmän voisi pienillä muutoksilla saada tukemaan moniulotteista blendausta. Tämä toimisi samaan tapaan kuin päällekkäiset blend shapet: ulottuvuuksilla olisi omat painoarvonsa, joilla niitä interpoloidaan, ja muutokset lisättäisiin lähdegeometriaan.

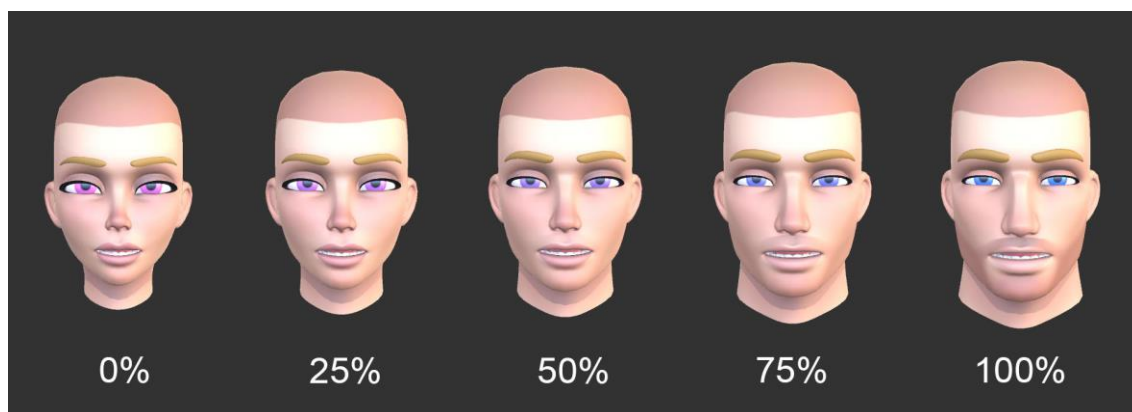
Blendauksen voisi samaan tapaan suorittaa erikseen myös eri ruumiinosille. Ulottuvuuksien määrän kasvattaminen moninkertaistaisi syntyvien erilaisten hahmojen määrän. Moniulotteisessa blendauksessa olisi kuitenkin otettava huomioon ulottuvuuksien kasvavan määrän vaikutus suorituskykyyn. Se myös monimutkaistaisi mallinnusprosessia, koska kaikki blendattavat piirteet olisi eristettävä toisistaan.

6 Pohdintaa

Kahden hahmon välinen blendaaminen toimii juuri niin kuten olin odottanutkin. Geometrioiden blendaus toimii kuten blend shapet, joten sen toiminnasta minulla ei ollut juurikaan epäilyksiä. Luurankojen blendauksesta olin hieman epävarmempi, mutta sekin toimii juuri kuten oli tarkoituskin. Luurangot ja geometriat blendautuvat myös täysin synkronisesti niin, että lopputuloksen luut ovat aina oikeissa paikoissa suhteessa geometriaan. Esimerkiksi silmäluu on aina täysin silmägeometrian keskellä. Tämä on tärkeää, jotta blendauksen lopputuloksena syntyvät hahmot deformoituvat oikein.

Projektia varten tekemäni hahmot ovat varsin samankaltaisia. Tämän seurauksena blendauksen lopputuloksena syntyvät hahmot ovat hyvin uskottavia (kuvio 34). Uskottavalla tarkoitan sitä, että ne voisivat toimia omina erillisinä hahmoinaan kahden

alkuperäisen hahmon maailmassa. En kuitenkaan näe, miksei järjestelmä voisi toimia myös erilaisemmillä hahmoilla. Uskon, että suuri vaikutus tulosten uskottavuuteen oli omassa projektissani sillä, että alkuperäisillä hahmoilla ei ollut juurikaan yksilöllisiä piirteitä, kuten erilaisia hiustyyliä. Tällaiset piirteet tekisivät ehkä blendauksen tuloksista vähemmän yksilöllisiä suhteessa pohjahahmoihin, ja selkeämmin sen näköisiä, että ne on luotu sekoittamalla kahta hahmoa. Koenkin, että yksilöllisten piirteiden jättäminen pois pohjahahmoista voisi olla yleisesti hyvä ratkaisu, ja niiden lisääminen vasta blendauksen jälkeen tekisi myös tuloksena syntyvistä hahmoista yksilöllisempiä ja siten mahdollisesti käyttökelpoisempia joihinkin tarkoituksiin.



Kuvio 34. Blendauksen vaihteita eri painoarvoilla.

Mainitsin aiemmin, että blendauksen painoarvo on mahdollista viedä myös $\{0, 1\}$ -välin ulkopuolelle, jolloin nais- ja mieshahmon väliset erot voi viedä äärimmilleen. Tällä on kuitenkin lähinnä huumoriarvoa, ja tarpeeksi pitkälle viety painoarvo rikkoo hahmot täysin. Painoarvolla -2 kasvot surkastuvat ja kääntyvät jo osittain sisäpuoli ulospäin. Blendauksen painoarvon ollessa 2 syntyy "luolamies", joka muistuttaa vielä yllättävän paljon ihmistä (kuvio 35).



Kuvio 35. Blendauksen lopputulokset painoarvoilla -2 (vas.) ja 2 (oik.)

Ilmeiden ja tunnetilojen välittyminen oikein hahmolta toiselle on kriteeri blendausjärjestelmän käytettävyydelle. Mikäli tämä kriteeri ei täyty, on vaikea kuvitella, että järjestelmästä olisi juurikaan hyötyä. Retargetointisysteemi ei ota täydellisesti huomioon hahmojen välisiä mittasuhte-eroja, joten on mahdollista, että hienovaraisemmat ilmeet eivät siirry täysin oikein. Eroja voi aiheuttaa myös se, jos molempien hahmojen perusilme ei ole täysin sama.

Ilmeet siirtyivät mielestäni varsin hyvin. Ilmeissä on kuitenkin huomattavissa eroja hahmojen välillä. Nämä tuntuvat johtuvan suurimmaksi osaksi siitä, että mieshahmon suu on leveämpi kuin naisen. Retargetointi näyttäisi saavan suupielet liikkumaan miehellä oikein suhteessa suun kokoon, joten esimerkiksi hymy ja murjotus näyttävät varsin samoilta molemmilla hahmoilla. Suun ollessa auki se tosin näyttää naishahmolla olevan enemmän auki kuin miehellä, vaikka leukaluun kulma on molemmilla sama. Tämä saa naiselle tehdyt liioitellut ilmeet näyttämään mieshahmolla vähemmän liioitelluilta (kuvio xx). Ilmeiden väliset erot ovat kuitenkin mielestäni niin pieniä, että järjestelmän käytön hyödyt voittavat ylivoimaisesti sen haitat.

Mahdollisia käyttötarkoituksia blendausjärjestelmälle voisivat olla esimerkiksi proseduraalinen hahmogenerointi, hahmonluonti ja -kustomointi peleissä. Pienet indie-pelifirmat, joilla ei muuten olisi siihen tarvittavia resursseja, voisivat käyttää sitä projekteissa, jotka vaativat suuren määrän erilaisia hahmoja. Suuremman budjetin AAA-peleissä systeemin voisi viedä vielä seuraavalle tasolle hyödyntämällä moniulotteisuutta. Tätä opinnäytetyötä kirjoittaessani sain tietää samankaltaisesta tekniikasta, joka on kehitteillä Beyond Good and Evil 2 -peliin, joka mahdollistaa blendaamisen erilaisten antropomorfisten eläinhahmojen välillä (kuvio 36).



Kuvio 36. Blendaamalla luotu hybridi pelistä Beyond Good and Evil 2 (Ubisoft 2017).

Opin projektin kuluessa ja tätä opinnäytetyötä kirjoittaessa paljon aiheista, joista olin halunnut tietää lisää jo pidempään. Erityisen hyödylliseksi koin kvaternioiden toiminnasta lukemisen. Kirjoittaminen myös auttoi selkeyttämään ajatuksiani, ja oppimaan uudelleen asioita, jotka luulin jo osaavani. Jouduin useampaan otteeseen kyseenalaistamaan omat pohjatietoni, ja päädyin prosessissa oppimaan jotakin uutta. Projekti lisäsi valmiuksiani työskennellä vastaavanlaisten projektien parissa tulevaisuudessa, sekä artistina että ohjelmistokehittäjänä.

Lähteet

Anatomy for 3D Artists. 2015. United Kingdom: 3dtotal Publishing.

Autodesk 2017. Bind pose. Autodesk Knowledge Network, Maya User Guide.

<https://knowledge.autodesk.com/support/maya/troubleshooting/caas/CloudHelp/cloudhelp/2017/ENU/Maya/files/GUID-36808BCC-ACF9-4A9E-B0D8-B8F509FEC0D5-htm.html> (15.04.2018)

Autodesk 2018. Nodes: blendShape. Autodesk Knowledge Network, Maya User Guide.

https://help.autodesk.com/view/MAYAUL/2018/ENU/?guid=Nodes_blendShape_html (25.03.2018)

Burton, A. Mike, Bruce, Vicki, Dench, Neal 1993. What's the difference between Men and Women? Evidence from Facial Measurement

Cgwiki 2017. Points and Verts and Prims.

http://www.tokeru.com/cgwiki/index.php?title=Points_and_Verts_and_Prims (22.03.2018)

Epic Games 2018. Animation Retargeting, How Does Retargeting Work? Unreal Engine 4 Documentation.

<https://docs.unrealengine.com/en-us/Engine/Animation/AnimationRetargeting#how-doesretargetingwork?> (24.03.2018)

Julius, Jessica 2014. The Art of Big Hero 6. San Francisco, California: Chronicle Books LLC.

Liu, Chen 2006. An analysis of the current and future state of 3D facial animation techniques and systems.

Pluralsight 2014. 5 Tips for Character Rigging. Pluralsight.com

<https://www.pluralsight.com/blog/film-games/5-tips-character-rigging> (16.04.2018)

Rigging Dojo 2016. Disney Character Artist Shares His Story: Interview with Character TD Sergi Caballer. <http://www.riggingdojo.com/2016/02/13/disney-character-artist-shares-his-story/> (24.03.2018)

Ritchie, Kieran, Alexander, Oleg, Biri, Karim 2006. The Art of Rigging Volume 3. CG Toolkit.

Schacht, Ryan 2005. Female and Male Perceptions of Attractiveness: What is Attractive and Why?

Shoemake, Ken 1993. Quaternions.

Stack Exchange 2015. How are vertex indices determined? Blender Stack Exchange. <https://blender.stackexchange.com/questions/36577/how-are-vertex-indices-determined> (15.04.2018)

The Art of Overwatch. 2017. Milwaukie, Oregon: Dark Horse Books.

Tindall, Brian 2009. Articulation: The movement of points in space. Hippydrome.com <http://hippydrome.com/Skeleton.html> (24.03.2018)

Ubisoft 2017. Beyond Good and Evil 2 – Space Monkey Report #1 Live Stream. youtube.com <https://www.youtube.com/watch?v=cYWPyyqsLnM> (19.04.2018)

Unity 2017. Model Importer: Model. Unity Documentation. <https://docs.unity3d.com/Manual/FBXImporter-Model.html>

Unity 2018. Skinned Mesh Renderer. Unity Documentation. <https://docs.unity3d.com/Manual/class-SkinnedMeshRenderer.html> (15.04.2018)

Valve 2017. Dota 2 Workshop – Color Texture Light Baking. Steam Support. <https://support.steampowered.com/kb/8700-SJKN-4322/dota-2-character-texture-guide> (16.04.2018)

Video: Blendaus toiminnassa

<https://youtu.be/dIBAFsIAvpQ>

Video: Ilmeet eri blendauksen painoarvoilla

<https://youtu.be/jJhTSdx86ik>