



Kalevi Toivanen

## Realiaikaisen ihmishahmon tuotanto Blender ja Unity sovelluksilla

Case: Merivoimien VR koulutusjärjestelmän koulutaja-avatarin tuotanto

Metropolia Ammattikorkeakoulu

Muotoilija AMK

3D-animointi ja -visualisointi

Opinnäytetyö

Päivämäärä

## Tiivistelmä

Tekijä(t):	Kalevi Toivanen
Otsikko:	Realiaikaisen ihmishahmon tuotanto Blender ja Unity sovelluksilla
Sivumäärä:	39 sivua + 2 liitettä
Aika:	01.5.2025
Tutkinto:	Muotoilija AMK
Tutkinto-ohjelma:	Muotoilun tutkinto-ohjelma
Suuntautumisvaihtoehto:	3D-animointi ja -visualisointi
Ohjaaja(t):	Lehtori Kristian Simolin

---

Opinnäytetyö on toimeksianto Puolustusvoimilta tuottaa toiminnallisiin VR harjoitteisiin kouluttaja-hahmo, joka toimii avatarin lisäksi tulevan palautteenantojärjestelmän visuaalina. Työ koostuu enimmäkseen projektiosuudesta, joka on prosessikuvaus hahmon tuotannosta ja sen viemisestä realiaikaiseen 3D moottoriin.

Hahmon tuotantoon sisältyy referenssikuvaukset, kasvojen digitaalinen veistäminen, ihomateriaalin teksturointi sekä splinihiuksien mallintaminen. Tuotannon jälkeen hahmo siirretään realiaikaiseen 3D moottoriin.

Puolustusvoimat tulee hyötymään dokumentoidusta hahmon tuotannon prosessikuvauksesta, kun tulevaisuuden toiminnalliset VR harjoitteet vaativat useampia ja realistisempi ihmishahmoja. Tämän opinnäytetyön hahmo tullaan liittämään myös olemassa oleviin VR harjoitteisiin.

Avainsanat: Digi-ihminen, Blender, Unity, VR

---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author(s): Kalevi Toivanen  
Title: The production of a real time human character with Blender and Unity  
Number of Pages: 39 pages + 2 appendices  
Date: 1 May 2025

Degree: Bachelor of Culture and Arts  
Degree Programme: Design  
Specialisation option: 3D Animation and Visualization  
Instructor(s): Kristian Simolin, Principal Lecturer

---

This thesis is the result of an assignment from the Finnish Navy to produce a virtual realtime instructor character for VR applications. In addition to working as an avatar for a user in the role of an instructor, the character is planned to work as the visual component of a new kind automatic feedback system. This work consists mostly of the process description of a practical project. The project phase first focuses on producing the character assets, and then on setting them up in a real time 3D engine.

The Finnish Defence Forces will take advantage of this documentation-style work in the future, when future VR training applications require new realistic real-time human characters. The character produced for this thesis will be appended to existing training applications.

Keywords: Digital human, Blender, Unity, VR

## Sisällys

1	Johdanto	6
2	Käsitteitä	6
3	Käyttötarkoitus	7
4	Realistiset digi-ihmiset nykyviihdeteollisuudessa	8
4.1	Haasteena Fotorealistiset Ihmiskasvot	8
4.2	Esimerkkejä reaaliaikaisista digi-ihmis toteutuksista	10
5	Projektiosuus 1: Realistiset kasvot Blenderissä	12
5.1	Referenssin kerääminen	12
5.2	Veistäminen	13
5.2.1	Dyntopo	14
5.2.2	Referenssin käyttäminen	15
5.2.3	Remeshaus ja Multires veistäminen	18
5.3	Ihon Teksturointi	20
5.3.1	Ihomateriaalin luominen dyntopoveistokselle	20
5.3.2	Ihomateriaalin leipominen	22
5.4	Hiukset ja karvat	23
5.4.1	Hiuspinta	24
5.4.2	Kontrollihiukset	25
5.4.3	Hiusmodifierit	26
5.4.4	Viimeistely	28
5.5	Hahmon vartalo ja rigaus	29
6	Projektiosuus 2: Hahmon vienti Unity-pelimoottoriin	29
6.1	Hahmon vieminen Blenderistä	30
6.2	Digital Human Package:n asentaminen	31
6.3	Pään mallin valmistelu pelimoottorin puolella	32
6.4	Digital Human silmät	33
6.5	Hiukset ja karvat Unityssa	34
7	Tulos	36
8	Loppusanat	37

Lähteet	38
Kuvalähteet	39
Liitteet	41
Automaattinen tekstuurikarttojen leivontaan tarkoitettu python skripti	41
Realiaikainen demoaplikaatio, jossa kouluttaja-avatar	42

## 1 Johdanto

Opinnäytetyö on tulos toimeksiannosta luoda reaaliaikainen kouluttaja-hahmo VR sovellukseen. Tämän lisäksi tuotannosta piti laatia prosessikuvauksellinen dokumentaatio, jota voisi tulevaisuudessa soveltaa. Hahmo toteutetaan kohtuullisilla resursseilla (järjestelmäkamera, tehokas graafinen työasema). Tämän hahmoprojektin assetit eivät voineet pohjautua kalliiseen datasettiin, kuten LightStage studiokaappaukseen. Tulevaisuudessa samankaltaisissa hahmoprojekteissa on tärkeää että tekijällä on kokemusta ihmispiirteiden veistämisestä sekä referenssi-ihmisen piirteiden ja näköisyyden toisintamisesta. On mahdollista että toisintaminen tapahtuu lähes kokonaan käsin kuvareferenssin pohjalta. Opinnäytetyön projektiosuus jakautuu kahteen osaan. Ensimmäisessä osassa hahmon assetit luodaan Blender-ohjelmalla. Projektiosuuden toinen osa keskittyy DigitalHuman-lisäosan avulla hahmon viemiseen Unity-pelimoottoriin.

## 2 Käsitteitä

VR – virtuaalitodellisuus.

Shaderi (shader) – Ohjelma, joka suoritetaan tietokoneen näytönohjaimella. Yleensä käytetty grafiikan piirtämistä varten.

Modifieri (modifier) – Blender mallinnusohjelman ominaisuus, jossa 3D objektille voidaan lisätä erilaisia ohjelmapalikoita, jotka muokkaavat objektin dataa.

Retopo (retopology) – Mallintamistekniikka, jossa 3D mallin meshiverkko rakennetaan uusiksi. Tavoitteena on yleensä siistimpi meshiverkko.

Splinihiukset (spline hairs) – 3D grafiikassa käytetty tapa piirtää hiuksia ja karvoja käyttäen hyväksi 3D kurvi-splinejä.

Unity GameObjekti (GameObject) – Nimitys objekteille, jotka on aseteltu Unity pelimoottorissa pelimaailmaan.

Unity Assetti (Asset) – Nimitys objektille, joka sijaitsee Unity projektin kansion rakenteessa.

Unity komponentti (component) – Unity GameObjektiin liitettävä C# kielellä kirjoitettu ominaisuuspalikka.

### **3 Käyttötarkoitus**

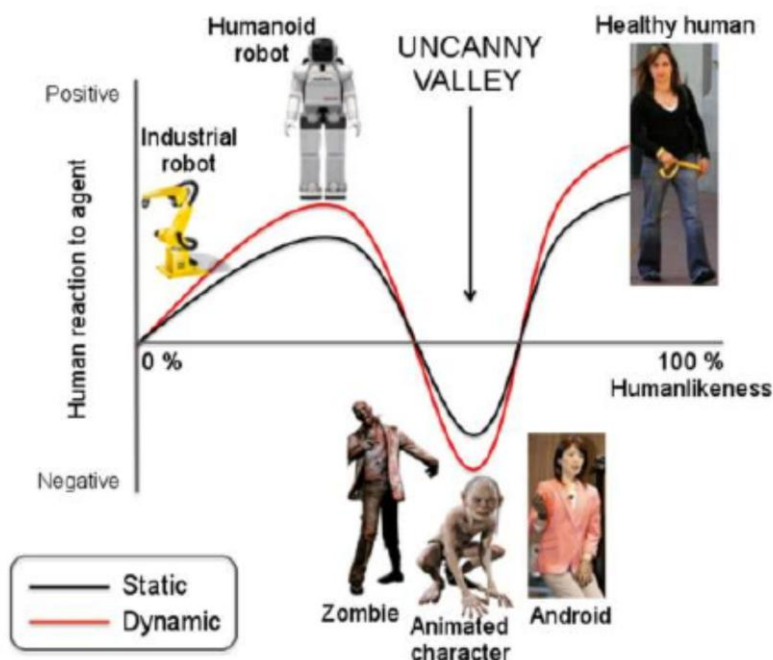
Idea opinnäytetyön aiheesta tuli Merivoimien oppimisympäristöjen kehittämispäälliköltä Komentajakapteeni Pasi Leskiseltä. Tämän teoksen kirjoitushetkenä Merivoimilla on parhaillaan käynnissä POC (proof of concept) tutkimusta VR koulutuksesta sekä VR harjoitteiden mahdollisesta siirtovaikutuksesta oikean maailman toimenpiteisiin. Unity pohjaisen kouluttaja-avataarin tarve on peräisin Leskisen ideasta luoda kaikille VR harjoitteen käyttäjille henkilökohtainen palautetuokio harjoituksen lopuksi. Palautetilaisuudessa kouluttaja-avataari ja sitä ajava ohjelma antaa syntetisoidun puheen avulla mahdollisimman luonnollisen kuuluisen palautteen harjoitteen suorittaneelle henkilölle. Tämän hahmoprojektin referenssihenkilönä toimii eräs Merivoimien ammattilaishenkilö, joka esiintyy moniaillaisen asiantuntijuutensa takia monessa Puolustusvoimien koulutusvideossa. Toiveena on nostaa eri opetusmateriaalien yhtenäisyyttä antamalla VR harjoitteiden kouluttaja-hahmolle kyseisen henkilön näköisyys.

Opinnäytetyö toimii dokumentaationa Merivoimille, sekä tuleville VR harjoitteiden kehittäjille, kuten virtuaalisen koulutusympäristön varusmiehille.

## 4 Realistiset digi-ihmiset nykyviihdeteollisuudessa

### 4.1 Haasteena Fotorealistiset Ihmiskasvot

Fotorealististen kasvojen saavuttaminen on 3D-artistille haastava, mutta se on myös erittäin hyödyllinen taito. Fotorealistiset maailmat ja sitä myötä fotorealistiset ihmishahmot ovat suuressa kysynnässä nykyisessä viihdeteollisuudessa ja niiden tuottaminen ei ole ikinä ollut helpompaa kuin nyt. Muun muassa monet korkean budjetin videopelit ovat riippuvaisia fotorealismista, koska se houkuttaa monia pelaajia, jotka haluavat immerstiivisen pelikokemuksen (Price 2016). Vaikka hahmoartistin tavoitteena olisi enemmänkin tyylieltyjen hahmojen luonti, perustana pitää kuitenkin ymmärtää realistiset ihmispiirteet, jotta tyylielty lopputulos olisi visuaalisesti miellyttävämpi. Artistin on usein tiedettävä realismiin perustuvat ominaisuudet, ennen kuin hän voi soveltaa niitä omassa työssään. (Price 2016).



Kuva 1: Uncanny valley diagrammi. (Manaf, Ismail, Arshad & Lee 2022)

Digitaalisten ihmisten kasvoja on yritetty vuosikymmeniä saada realistisemmiksi, uskottavammiksi, välttämällä "uncanny valley" ilmiötä. Hypoteesi uncanny vallestä syntyi vuonna 1970 kun Masahiro Mori esitti konseptin tunteesta, joka

ihmisille syntyy realismia tavoittelevasta robotiikasta ja proteeseista. Hypoteesin mukaan, kun robottien ja humanoidien hahmojen ihmisyyks kasvaa, hahmon miellyttävyyks kasvaa vain tiettyyn pisteeseen asti. Kun ihmisyyttä yritetään alkaa lisäämään, tulos voi siinä vaiheessa alkaa näyttää katsojasta vastenmieliseltä tai pelottavalta. Kun ihmisyyttä lopulta lisätään riittävästi, alkaa hahmon miellyttävyyks vähitellen taas nousemaan ja se alkaa tuntua uskottavalta. Ihmisten kokemaa tunnetta miellyttävyyden hetkellisestä laskusta ja epämukavuuden tunteesta kutsutaan nimellä “Uncanny Valley” (Ratajczyk 2019, 137). Tämä ilmiö pätee myös digitaalisiin ihmiskasvoihin. Huomioitavaa on myös se, että Uncanny Valley -ilmiön vaikutus moninkertaistuu liikkuvien ihmishahmojen kohdalla. Tämä johtuu siitä että ihmisten elekieltä on vaikea jäljentää niin, että se näyttäisi luonnolliselta. Tämän takia uskottavien, liikkuvien ihmiskasvojen luonti on vieläkin haastavampaa kuin paikallaan olevien veistosten. Ilmiöstä eroon pääseminen vaati paljon harjoittelua ja kokemusta.

Virtuaalisessa ympäristössä on haastavaa saada ihmisen iho näyttämään uskottavalta. Yksi todella tärkeä ihon ominaisuuksista on “Subsurface Scattering” (SSS), eli suomeksi valon pinnanalainen syvähajonta. Tällä tarkoitetaan sitä efektiä, mikä syntyy kun valo heijastuu ihon alle ja ihon rakenteen takia hajautuu eri suuntiin. Hajautumisen jälkeen valo osittain imeytyy ja osittain poistuu ihosta (d'Eon, Luebke 2007). Subsurface scattering:in renderöiminen materiaalissa vaatii huomattavasti enemmän laskentatehoa, verrattuna kiinteään pintaan, jossa ei ole tätä efektiä. Realiaikaisissa sovelluksissa haasteena on huomattu laskentatehon ja fideliteetin tasapainottaminen (d'Eon, Luebke 2007). On myös tärkeää saada ihon pinnan muodon, sekä sen monikerroksisen heijastavuuden näyttämään aidolta.

## 4.2 Esimerkkejä reaaliaikaisista digi-ihmis toteutuksista



Kuva 2: "Ira" reaaliaikainen teknologiademo (Nvidia 2013)

Nykyään elokuvien digi-ihmisiä luodaan uskottavammiksi "Light Stage" teknologian avulla. Light Stage:n toimintaperiaate on luoda tila, jossa näyttelijä pystytään valoittamaan ja kuvaamaan jokaisesta mahdollisesta kulmasta samanaikaisesti. Light Stage studioon tarvitaan lähes kaksisataa järjestelmäkameraa, sekä satoja ledivaloja. Kameran ja valon monesti suodatetaan esimerkiksi polarisoivilla suodattimilla. Esimerkiksi ihmisten kasvoista voidaan kerätä kattava datasetti tällä menetelmällä. Datasetti voi sisältää kasvojen mirkomuotojen lisäksi tietoa kohteen ihon heijastavuudesta sekä sen läpikuultavuudesta. Tällaista dataa ei pysty saamaan esimerkiksi henkilön kasvojen kipsivalun laserskannauksesta (Debevec, 2012), joita on käytetty realististen ihmishahmojen tuottamiseen aikaisemmin. Kuvan 2 "Ira" teknologiademo on tuotettu Light Stage menetelmän avulla. Demon alkuperäinen Light Stage kaappaus oli kooltaan lähes 32 gigatavua, joten datan muuttaminen reaaliaikaiseksi vaati Nvidialta uudenlaisen työkalun, nimeltä FaceWorks. Väitetysti työkalu pystyi lopulta kompressoimaan ja optimoimaan Light Stage datan 400 megatavuun (Nvidia 2013). FaceWorks -työkalua varten kehitettiin uudenlaisia subsurface scattering shadereita, jotka mahdollistivat ihon reaaliaikaisen valon hajottamisen. Shaderohjelma mahdollisti myös reaaliaikaisen ihon läpikuultavuuden ohuissa kohdissa, kuten korvissa ja nenässä (Reed 2014).

Activion-yhtiön tutkimusosasto tuotti kyseisen “Ira” hahmon kasvojen Light Stage datasta kevyemmän reaaliaikaisen version. Blendshapejen sijaan heidän versio käytti kasvojen animoimista varten yhtä meshiä ja 70:n luun kasvorigiä (Activision R&D 2013).

Vuonna 2021 Unreal Engine 5 uusioversion esittelyn ohella Epic Games julkaisi Metahuman Creator työkalun, joka mahdollistaa realististen digi-ihmisten nopean luomisen Unreal Engine 5:lle. MetaHuman Creator on täysin selaimen kautta toimiva pilvipalvelu. Käyttäjät voivat valita hahmonsa pohjaksi listalta valmiin ihmishahmon. Hahmon muokkaaminen onnistuu sekoittamalla valmiiden ihmishahmojen piirteitä keskenään tai veistopisteiden avulla. Korkeatasoisten muokkaustyökalujen lisäksi Metahumanit tukevat automaattisesti monia liikkeenkaappaus järjestelmiä, sekä kasvojen liikkeiden kaappausta Live Link Face-älypuhelinsovelluksen avulla (Epic Games 2021).

Unity julkaisi vuosina 2020 ja 2022 demovideot “The Heretic” ja “Enemies”, jotka molemmat esittelivät Unity demotiimin kehittämiä työkaluja realististen ja reaaliaikaisten ihmishahmojen luomiseen. Kehittyneempi “Enemies” demo esitteli uutta reaaliaikaista hiussimulaatiotyökalua, sekä “4D” datan prosessointia, jolla tarkoitetaan 3D meshisekvenssien toistoa reaaliaikaisesti (Unity Demoteam 2023). Reaaliaikainen elokuva oli tuotettu Light Stage teknologian avulla (Schoennagel 2023). Näitä demoja varten Unity pelimoottorille kehitettiin “DigitalHuman” lisäosa. Lisäosa tuo mukanaan ihmishahmoille tarkoitetun Ihomateriaalikokonaisuuden, silmien renderöintisysteemin, Ihon pintakiinnityssysteemin, ihon venytys- ja jännitesysteemin reaaliaikaista rypytystä varten, sekä aikaisemmin mainitun 3D sekvenssidatan toistomahdollisuuden (Schoennagel 2023). Tämän opinnäytetyön projektiosuuden ihmishahmon realisointia varten käytin monia näistä DigitalHuman paketin ominaisuuksista.

## 5 Projektiosuus 1: Realistiset kasvot Blenderissä

Ensimmäinen projektiosuus keskittyy kouluttaja-hahmon asettejen tuottamiseen Blender-sovelluksella. Hahmoa varten tuotan vain referenssihenkilön pään, koska saatavilla on valmis korkealaatuinen vartalon malli.

### 5.1 Referenssin kerääminen

Tämän opinnäytetyön kouluttaja-avatar tullaan luomaan yhdestä ihmisreferenssistä. Referenssi kerätään valokuvaamalla ja videokuvaamalla referenssihenkilöä järjestelmäkameralla.



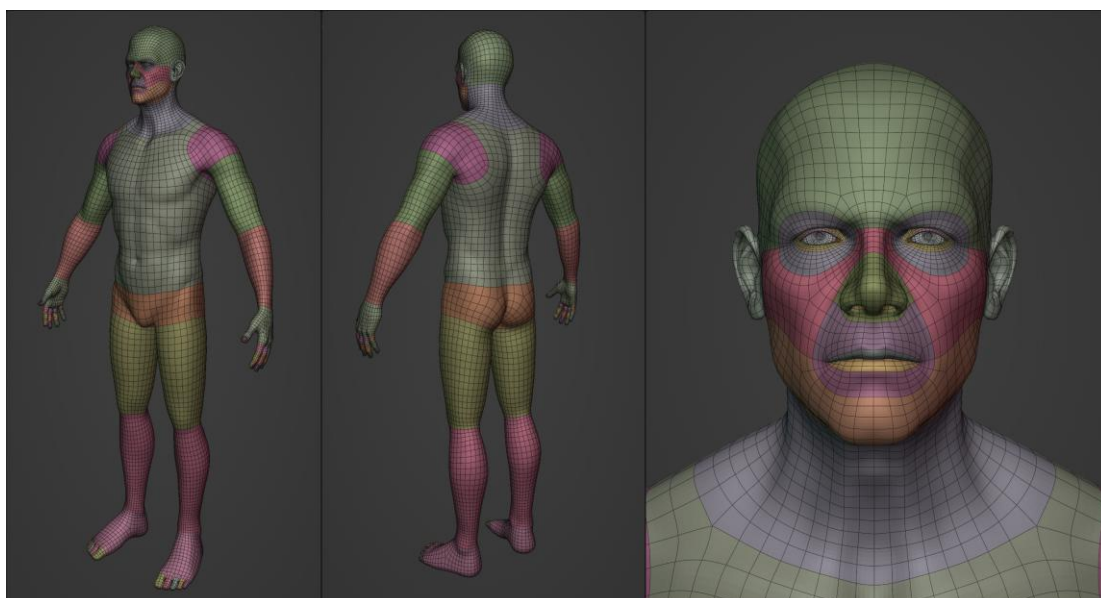
Kuva 3: kollaasi joistakin otetuista referenssikuvista.

Referenssikuvia tuli otettua kuvauspäivänä lähes 70. Kuvasin henkilön päätä ja kasvoja aluksi kahdeksasta eri leveysasteesta. Jokaisen leveysasteen kohdalla otin kuvan myös kolmesta eri pituusasteesta: ylhäältä, keskeltä ja alhaalta. Otin kuvia etenkin kasvoista ja pään yksityiskohdista, kuten korvista, silmistä ja suusta. Lopulta otin myös videoreferenssiä mm. siitä kun henkilö puhui ja tämä video toimi referenssinä kasvojen animoimista varten.

Referenssikuvan valoituksella voi vaikuttaa referenssin käyttötarkoitukseen. On olemassa erilaisia valoitustekniikoita, joilla voidaan saada referenssikuvia, joissa korostuu erilaiset kasvopiirteet, sekä ihon pienet yksityiskohdat. Tällai-

set kuvat ovat hyödyllisiä varsinkin veistosvaiheessa, kun muotojen tunnistaminen referenssistä on tärkeää. Toinen hyödyllinen referenssikuvamuoto on ”Cross Polarization”, tai ristipolarisoitu kuva. Kuvaustekniikka viittaa tapaan asettaa valonlähteeseen sekä kameran linssiin polarisointisuotimet siten, että kuvatun kappaleen heijastukset ja kiillot katoavat kokonaan. (Zmejevskis 2022). Ristipolarisointivalokuvausta sovelletaan monesti fotogrammetriamallinnuksessa, sillä kiiltävistä pinnoista aiheutuneet heijastukset voivat heikentää fotogrammetrian lopputulosta. Kiiltojen aiheuttamien vääristymien puuttumisen vuoksi ristipolarisoidusta valokuvasta on myös helpompaa luoda esimerkiksi ihon tai silmien väritekstuurikartan.

## 5.2 Veistäminen



Kuva 4: Ilmainen blender studion tuottama ihmisen pohjameshi. Mallilla on valmiiksi erittäin siisti topologia, jota voi myös hyödyntää myöhemmin retopo vaiheessa. (Blender Studio 2022)

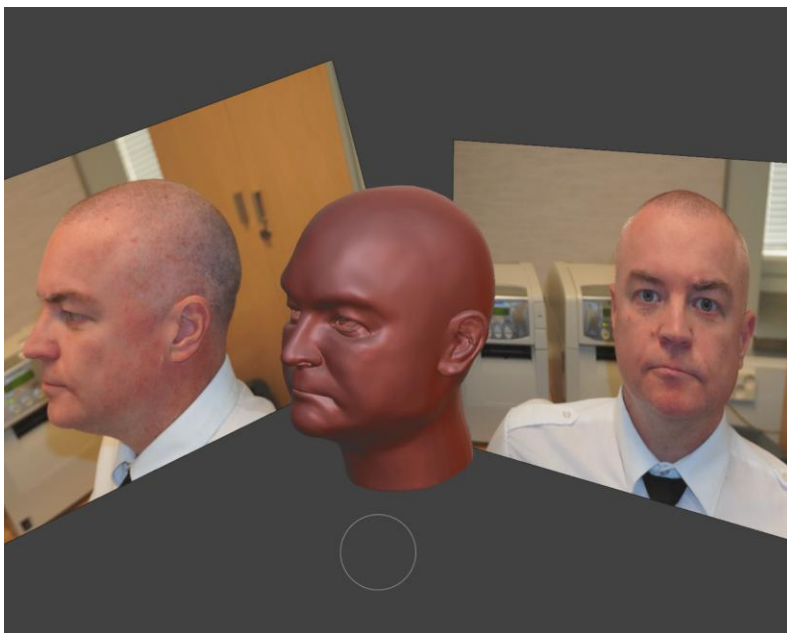
Hahmon varsinainen tuotanto alkaa veistovaiheesta. Jotta aikaa säästyisi, pään veistäminen kannattaa laatia valmiin pohjamallin päälle, sen sijaan että veistäminen aloitettaisiin primitiivistä, kuten pallosta. Pohjamallina voi käyttää melkein mitä vaan internetistä löytyvää pään mallia. Esimerkiksi kuvan 4 pohjamalli on saatavilla ilmaiseksi Blender studion sivuilta ja on erittäin hyvä lähtökohta hahmon pään, sekä mahdollisen vartalon tuottamiselle.

### 5.2.1 Dyntopo

Kun hakee pään muotoa, on suotavaa käyttää aluksi Blenderin veistotilan dyntopo järjestelmää. Dyntopo mahdollistaa meshiverkon dynaamista lisäämistä sekä poistamista veistämisen aikana. Dyntopo asetuksissa, veistonäkymän ylänurkassa, voi muuttaa resoluutiota sekä "Detailing" toimitaperiaatetta. "Detailing" kannattaa muuttaa "Constant Detail" asetukseen. Constant Detail -asetuksen ansiosta generoidun meshin tiheys pysyy samana, riippumatta siitä kuinka kaukana kamera on mallin pinnasta. Resoluutio on hyvä pitää aluksi matalalla kun muodostetaan kasvon suurempia piirteitä. Vasta pienempien yksityiskoh- tien, kuten silmäluomien, huulien ja korvien kohdalla, kannattaa dyntopon reso- luutiota nostaa. Korkeimpia resoluutioarvoja tulee todennäköisesti tarvitsemaan silmäluomissa, jolloin resoluutio kenttään voi syöttää jopa numeron 3000.

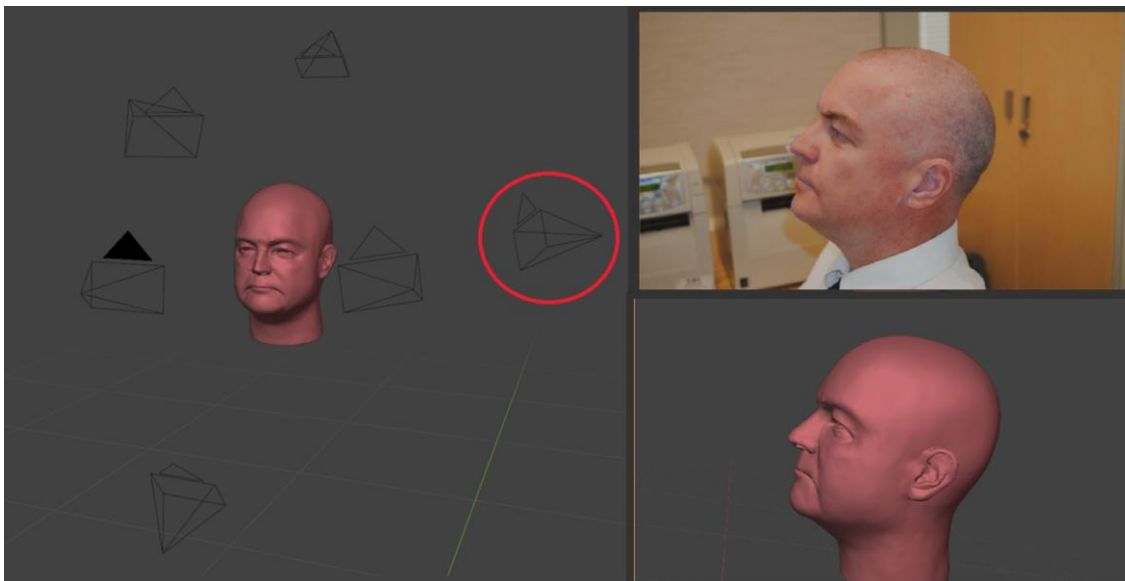
Välillä dyntopo-veistämisen aikana voi malli muuttua liian raskaaksi. Tällöin veistosmalli voidaan viedä "Decimate" modifierin läpi. Modifierin "Collapse" ase- tuksella pienennetään veistosten kolmiomäärää siten että yksityiskohdat säily- vät. Modifierin "Ratio" liikusäätöä voi viedä hiljalleen alaspäin, kunnes sopiva kolmioluvun ja yksityiskohtaisuuden suhde löytyy.

## 5.2.2 Referenssin käyttäminen



Kuva 5: karkean päänmuodon hahmottamiseen tarvitsee vain muutaman referenssikuvan.

Kun aloittaa veistämisen oikean maailman ihmisreferenssin pohjalta, riittää kuvan 5 mukaisesti etu- ja sivuprofiilikuvat muodon löytämiselle. Kuvat voi asetella veistoksen ympärille Blenderin "Reference Image" objektien avulla. Reference Image objektin asetuksissa voi säätää mm. kuvan läpinäkyvyyttä sekä piirtojärjestystä (Default, Front, Back). Referenssikuvat eivät myöskään näy renderissä, koska ne piirretään varsinaisen 3D näkymän päälle erillisenä kerroksena, samalla tavalla kuten lamput ja luut.



Kuva 6: Blenderin kameroiden avulla referenssikuvat asetettu mallin päälle oikeista kulmista. Kuvakollaasin oikeanpuolinen kuvapari on peräisin punaisella ympyröidyn kameran näkymästä.

Referenssikuvia voi käyttää myös asettamalla ne 3D mallin päälle (Englanniksi "Superimposing"). Tämä onnistuu helpoiten lisäämällä kuvat Blenderin kameran taustakuvaksi ja laskemalla alkuperäisen kuvan karkean polttovälin (Courtois 2024). Jos referenssikuvauksissa käytetty kamera tallentaa kuviin metadatan, voi polttovälin löytää sieltäkin. Esimerkiksi Gimpin kuvanmuokkausohjelmalla voi helposti selata kuvan metadatan menemällä päänäkökuvassa "Image -> Metadata -> View Metadata" osioon. Kuvien asettaminen Blenderin kameroiden taustalle tapahtuu kameraobjektin "Data" paneelista. Data-paneelin valikosta löytyy päälle kytkettävä asetus nimeltä "Background Images". Kuvan voi lisätä painamalla asetuksen sisällä olevaa "Add Images" -nappia. Jos referenssikuvan kuvasuhde ei ole sama kuin kameran, voi painaa "Frame Method" -aliasetuksesta "Fit" -vaihtoehdon voimaan. Tällöin kuva säilyttää alkuperäisen kuvasuhteen kamerasta riippumatta. "Depth" -asetus asetetaan "Front" -asentoon, jotta kuva on jatkuvasti 3D näkymän päällä. Jotta alla oleva veistos kuitenkin näkyisi, voi "Opacity" -kohdasta säätää kuvan läpinäkyvyyttä suuremmaksi.

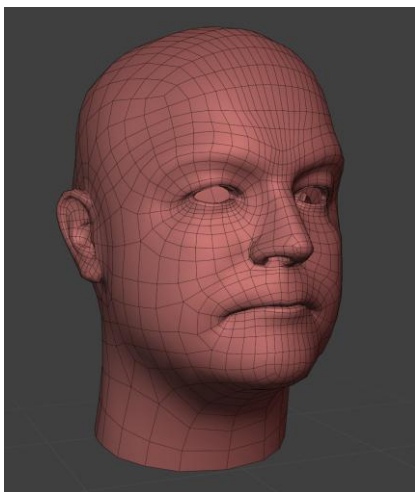


Kuva 7: Vasemmalla alkuperäinen referenssikuva. Oikealla muokattu referenssikuva, jossa naaman maamerkit kuten silmien ja suun muodot on korostettu sinisillä viivoilla.

Kameran asettelu ei välttämättä onnistu ensimmäisellä yrityksellä, varsinkin jos näköisyys ei ole vielä täysin kohdillaan. Kannattaa siis hienosäätää kameroiden asentoa ja kulmaa jatkuvasti paremmaksi veistämisen ohella. On myös mahdollista, että kameranäkymässä on vaikeata erottaa mikä muoto kuuluu veistokselle ja mikä referenssikuvalle. Tällöin voi muokata referenssikuvaa korostamalla tärkeitä muotoja ja piirteitä viivojen avulla, kuten kuvassa 7.

Kun tavoitteena on saavuttaa lopullista pään muotoa ja näköisyyttä, on erityisen tärkeitä luopua symmetrisyydestä hyvissä ajoin.

### 5.2.3 Remeshaus ja Multires veistäminen

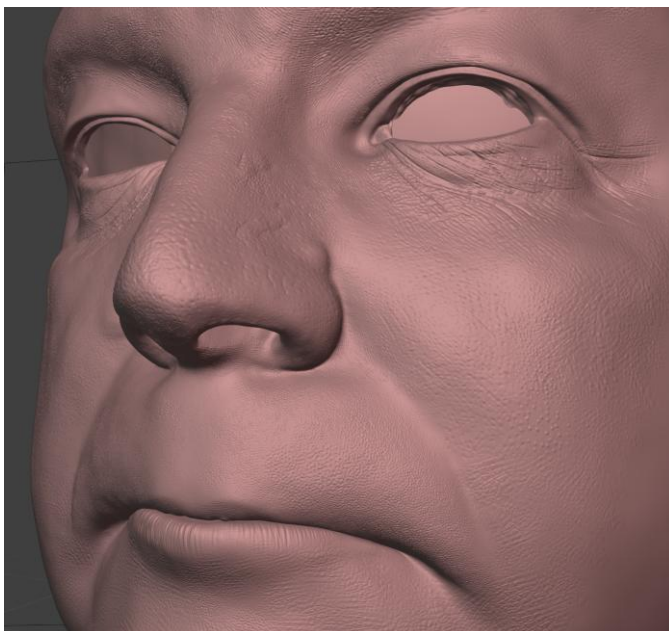


Kuva 8: käsin retopottu kouluttaja-hahmon pään 3D malli.

Kun Dyntopolla veistetty pään veistos on saavuttanut riittävän näköisyyden, on aika luoda sen pohjalta siisti ja optimoitu “Retopo” malli. Veistoksen retopologointi mahdollistaa monta asiaa, kuten hahmon animoinnin ja reaaliaikaisen toiston pelimoottorissa. Tämän takia se on todella tärkeä vaihe tämän projektin lopputuloksen saavuttamiselle. Retopo-mallia tullaan käyttämään myös “Multires” veistämiseen seuraavassa vaiheessa. Retopo-mallin tuottaminen onnistuu esimerkiksi uudellenkäyttämällä kuvan 4 pohjameshiä ja muokkaamalla sitä dyntopo-veistoksen mukaiseksi, tai luomalla meshin alusta työkalulla kuten Retopoflow.

Kun remeshattu malli on valmis, siitä voi tehdä kopion multires veistämistä varten. Kopioon pitää lisätä kaksi modifieria: “Multires” ja “Shrinkwrap”. Multires toimii hiukan samalla tavalla kuten “Subdivide Surface” modifieri, koska se pystyy myös jakamaan meshin polygoneja pienemmiksi ja sen seurauksena siloittamaan meshiä. Mutta toisin kuin Subdivision surface modifieri, Multiresin eri subdivision kerroksien luomaa uutta verteksidataa pystyy muokkaamaan Blenderin veistotyökaluilla myös silloin kun modifieri on vielä liitettyä objektiin. Multires modifieri on Blenderin ainoa modifieri, jota ei voi siirtää modifierilistalla toisen modifierin alle, joka luo meshille uutta verteksidataa, kuten esimerkiksi “Bevel” modifieri (Blender Manual, 2025). Shrinkwrap modifieri pystyy projisoimaan yhden 3D mallin verteksipisteitä toisen 3D mallin pintaan. Modifieryhdistelmän

avulla Multires modifierin subdivoima tiheämpi versio retopomallista projisoidaan Shrinkwrap modifierilla dyntopo veistoksen pintaan. Shrinkwrap modifierissa kannattaa käyttää parhaita tuloksia varten "Wrap Method" asetuksessa "Project" tilaa. Lisäksi "Negative" ruksiruutu pitää laittaa päälle, jotta projisointi tapahtuisi meshin pintaan nähden sisältä sekä ulkootapäin. Modifier listassa näiden kahden modifierin järjestyksellä on väliä, koska jos tässä vaiheessa Shrinkwrap modifieri on listan huipulla, ja sen "Apply" nappia painetaan, dyntopo veistoksen muodot eivät siirry Multires modifieriin. Jos painaa "Apply" nappi kun Shrinkwrap modifieri on Multiresin alapuolella, Shrinkwrapin projisointidata siirtyy Multires modifieriin. Tällöin dyntopoveistoksen tarkempi muoto siirtyy Multires-veistokseen.



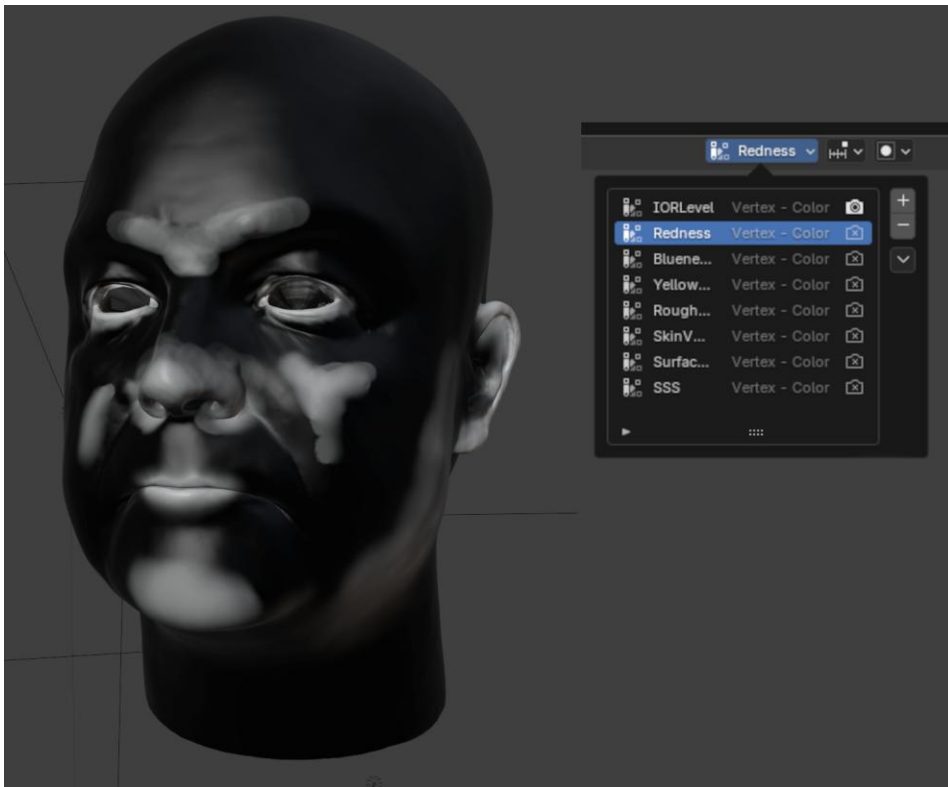
Kuva 9: lähikuva multires veistoksesta, johon on mallinnettu ihohuokokset ja pienoisrypyt.

Kun Multires malli on valmisteltu veistämistä varten, alkaa mikroyksityiskohtien, kuten ihonhuokosten ja rypyjen veistäminen. Huokosten veistämistä varten käytän valmista kaupallista huokosbrushipakettia. Pään multires veistoksen suositeltu resoluutio on 10-20 miljoonaa kolmiota. Multires modifieria ei tulla leipomaan malliin missään kohtaan. Muuten modifierin sisältämä data leipoutuu meshiin ja mallista tulee käyttökelvottoman raskas.

Veistovaiheen jälkeen hahmon päästä on luotu kolme erilaista versiota: Multires veistos, jossa on ihon mikroyksityiskohdat mallinnettuna, Dyntopo veistos, jossa on pään ja kasvojen piirteet tarkasti mallinnettuna ja lopulta remeshatu pään malli, joka on vähiten tarkka näistä kolmesta, mutta siisteimmällä topologialla varustettu pään malli. Retopo-malli on myös keveytensä takia se versio päästä, joka päättyy lopulta pelimoottoriin.

## 5.3 Ihon Teksturointi

### 5.3.1 Ihomateriaalin luominen dyntopoveistokselle



Kuva 10: veistonäkymässä hahmon dyntopo mallille lisättyinä monta erillistä väriattribuuta, joihin on maalattu arvoja.

Blenderin 3.2-versiossa saapui uutena ominaisuutena veistosten maalaustyökalu, joka perustuu veistoksen meshin verteksivärien muokkaamiseen ohjelman veistosnäkömön siveltimillä (Blender Developer Documentation 2022). Ominaisuus toimii lisäämällä meshin dataan uusia väriattribuuteja, joita Blenderin materiaalijärjestelmä voi tulkita oman noodin avulla. Teksturoidessa tulen käyttämään

eri väriattributeja maskeina ihomateriaalissa, jotta ihon väri saadaan luotua uskottavaksi. Kuvassa 10 nähdään hahmon pään dyntopomalliin lisättynä monta eri väriattributea, jotka kukin ajavat jotain ihomateriaalin ominaisuutta. Aikaa säästää se, ettei Sculpt Paint ominaisuuden käyttöä varten tarvitse UV kartoittaa veistosta tai myöskään luoda mallille tekstuurikarttoja. Kaikki data säilyy mallin verteksipisteissä.



Kuva 11: tyyllitetyllä ihmishahmolla perusihonväri luotu sekoittamalla ihon subdermaalisia värejä yksinkertaisen ihonvärin kanssa (Guldbæk 2020)

Ensimmäinen työstämisen kohde on itse ihon värin rakentaminen. Ihmisiho on erittäin moniväristä, joten yksittäinen RGB arvo ihomateriaalin väriulostulossa ei riitä luomaan realistista lopputulosta. Blenderin materiaalieditorissa kytken "Attribute" noodilla veistotilassa maalatut väriattributet Ihomateriaaliin. Kuvan 11 tyyllisen ihon pohjavärisekoitelman luon määrittelemällä ensin nodeverkostossa ihon pohjavärin "RGB" noodilla. Seuraavaksi lisään kolme Attribute –noodia, joista saan punaisille, sinisille ja keltaisille "Subdermal" sävyille mustavalkoiset sekoitusarvot. Näitä kolmea mustavalkomaskia työstän veistotilassa. Huomioitavaa Attribute noden käytössä on se että veistotilassa maalatuille väreille tulee gamma korjaus. Se tarkoittaa sitä, että veistotilassa maalattu mustavalkoinen 0.5 arvo tulkitaan Attribute nodessa arvona  $\sim 0.22$ . Jotta Attribute noden arvot olisivat lineaarisia, pitää arvot pistää potenssilaskun läpi, jonka eksponenttina on käänteinen sRGB gamman arvo, eli  $1 / 2.2$  (Wikipedia, Gamma correction).



Kuva 12: Kouluttajahahmon ihomateriaali.

Lopullisessa Ihomateriaalissa on subdermal-värien lisäksi lisätty pientä värikohinaa ihon väriin. Erillisillä attribute kanavilla lisäsin ihoon purkautuneiden pinta-suonien punaisuutta sekä vaaleutta ja tummuutta. Toisilla attribute kanavoilla säädin ihomateriaalin Principled BSDF shaderin karkeutta (Roughness), kiiltoa (IOR Level/Specular level) ja Subsurface scattering:in voimakkuutta.

### 5.3.2 Ihomateriaalin leipominen

Seuraavaksi dyntopo veistoksessa oleva ihomateriaali leivotaan retopomalliin tekstuurikartoiksi. Retopologisoidussa mallissa pitää olla UV kartta. Ihon väri-tekstuurin (basecolor), kiiltokartan (specular) ja karkeuskartan (roughness) asetin resoluutioksi 1024 pikseliä, eli "1K" kokoisiksi. Tekstuurit ovat kaikki täsmälleen neliön muotoisia. Normaalikartan resoluutioksi asetin 1K-tarkkuden sijaan 4096 pikseliä, eli 4K-tarkkuden, jotta ihonhuokosten muodot näkyisivät. Normaalikartta leivotaan multires veistoksesta, koska se sisältää yksityiskohtaiset ihonhuokosmallinnukset. Muut tekstuurikartat leivotaan dyntopo veistoksesta, jossa on ihomateriaali.

```

def bake_diffuse():
    deselect_nodes()
    source_mat.node_tree.links.remove(src_mat_output_socket.links[0])
    source_mat.node_tree.links.new(src_main_socket, src_mat_output_socket)

    node = target_mat.node_tree.nodes["diffuse"]
    node.select = True
    target_mat.node_tree.nodes.active = node

    set_bake_settings()
    bpy.ops.object.bake(type='DIFFUSE')

```

Teksti 1: Blender python koodinpätkä liitteestä 1, joka automaattisesti leipoo pään diffuse-tekstuurikartan.

Hahmon pään tekstuurikarttojen leipomista varten kehitin liitteen 1 yksinkertaisen python skriptin. Teksti 1 on pieni osa kyseisestä skriptistä. Tarve skriptille syntyi siitä, kun piti jatkuvasti iteroida ihon materiaalia. Blenderin tekstuuri-leivontajärjestelmä vaatii että leivonnan kohdemateriaalissa on aktiiviseksi valittuna se kuvatekstuuri-node, joka sisältää tekstuurin, johon sen hetkinen tekstuurikartta leivotaan. Lähdemateriaalin shader pitää myös muuttaa hetkellisesti "Emission" shaderiksi kun subsurface ja kiiltävyysskartoja leivotaan. Kaikkien näiden asetusten varmistaminen pystyy automatisoimaan python skriptillä, jonka ansiosta aikaa säästyy.

Leivontaskripti varmistaa että lähde- ja kohdemateriaalien nodeverkostoissa on oikeat kytkökset ja oikeat nodet valittuna, ennen kuin leipominen aloitetaan. Skripti myös asettaa automaattisesti oikean leivonta-asetuksen jokaiselle erilaiselle tekstuurikartalle. Kun kaikki tekstuurikartat on leivottu, skripti palauttaa nodeverkostoille oletuskytkökset.

## 5.4 Hiukset ja karvat

3D hiuksia pystytään piirtämään nykyään pääsääntöisesti kahdella eri tekniikalla. Ensimmäinen ja kevyempi menetelmä on nimeltään "hair cards". Hair cardeja tehdään mallintamalla hahmon hiukset polygoni suikaleilla ja sitten lisäämällä näihin suikaleisiin läpinäkyvä hiustekstuuri, jotta hiukset näyttäisivät koostuvan monesta yksittäisestä hiuksesta.



Kuva 13: Hahmo jolla on haircard hiukset. Vasemmanpuoleisella haircardit raakana meshinä. Oikeanpuoleisella haircarderiin lisätty alpha (läpinäkyvyys) tekstuuri, joka antaa illuusion yksittäisistä hiuksista. (Hansen 2023)

Toinen yleinen menetelmä, jolla luodaan hiuksia, on “spline hairs”. Suomeksi “splinihiukset”. Se on hiusten määrittelyn tapa, jossa jokainen yksittäinen hius koostuu yhdestä kurvi-spliniestä. Tämän jälkeen shader ohjelma piirtää näytölle meshiputkilon kuvastamaan hiusta, jonka se tekee splinitiedon ja datapisteiden avulla (hiuksen paksuus, muoto, väri, yms.).

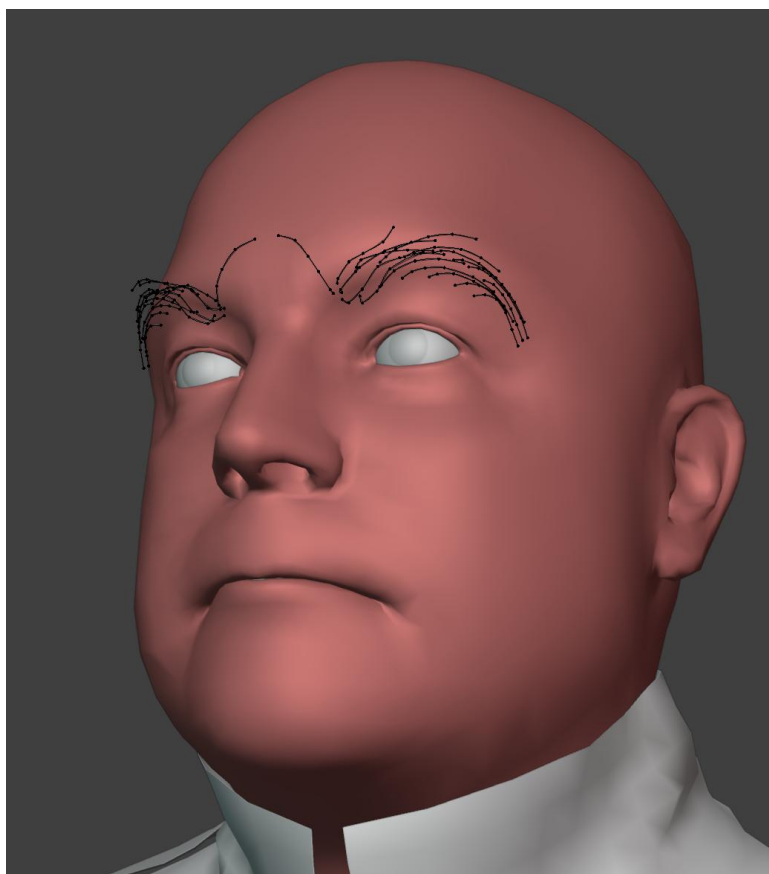
Tämän kouluttaja-avatarin luomiseen tullaan käyttämään splinihiuksia, koska niitä on erittäin helppo luoda Blenderissä ja myös koska Unity Digital Human –lisäosa tukee niitä.

#### 5.4.1 Hiuspinta

Jotta Blenderin splinihiuksia voidaan käyttää, pitää valmistella aluksi pinta, jolle hiuksia voidaan luoda. Nämä pinnat voidaan helposti nostaa alla olevasta pään retopomallista. Pinnan tulee olla riittävän suuri että kaikkien hiusten juuret mahduttavat siihen. Blenderin splinihiusjärjestelmä vaatii, että hiuspinnalla on toimivat UV koordinaatit. Pinnalle tullaan myös luomaan tiheystekstuuri, joten UV koordinaattien kannattaa täyttää koko UV alue. UV koordinaatit pystyy helposti generoimaan Blenderin sisäänrakennetulla “Unwrap Angle Based” toiminnolla.

### 5.4.2 Kontrollihiukset

Pinnan valmistelun jälkeen voidaan lisätä varsinainen hiusobjekti. Object modessa lisätään hiusobjekti ensin valitsemalla hiuspinta ja sitten lisäämällä "Hair Curves" objekti, esimerkiksi shift+A valikon kautta kohdasta Curve -> Empty Hair. Kun Curves objekti luodaan tällä tavalla, objekti automaattisesti linkittyy pintaobjektiin. Curves objekti myös automaattisesti parentoituu hiuspinta -objektiin. Koska Curves ei oletukselta sisällä mitään visuaalista käyrädataa, jota voisi valita helposti 3D näkymässä, pitää objekti etsiä scenen hierarkiasta, jotta sen voi valita aktiiviseksi.



Kuva 14: Lopullisia hiuksia harvemmat kontrollihiukset edit modessa.

Curves objektin Sculpt modessa voidaan alkaa luomaan karvoja haluttuun pintaan. Sculpt moden "Add" työkalulla voi lisätä yksittäisiä hiuksia. Blenderin "N Paneelin" (3D näkymän valikko, joka avataan oletuksella N painikkeella) "Tool" osiosta kannattaa aluksi tarkistaa, että yksi "Add" työkalun klikkaus lisää ainoastaan yhden hiuksen. Oletukselta tämä "Count" arvo voi olla korkeampi kuin yksi.

Manuaalisesti lisättyjen karvojen tiheyttä kannattaa pitää alhaisena, koska myöhemmin tullaan lisäämään modifiereiden avulla lisää hiuksia. Myös lopullisten karvojen muodon ohjattavuus säilyy. Tällaisia “kontrollihiuksia” kannattaa lisätä kuvan 14 kuvastamalla tavalla, eli ottamalla referenssikuvasta mallia mihin eri suuntiin karvat sojottavat eri kohdissa hiusmassaa. Kontrollihiusten pituudella ei ole kauheasti merkitystä tässä vaiheessa, koska sitä tullaan kuitenkin hienosäättämään myöhemmin modifiereiden avulla.

### 5.4.3 Hiusmodifietit



Kuva 15: kulmakarvojen modifietit

Blenderin sisäänrakennettujen hiusmodifiereiden avulla voidaan muutamasta kontrollihiuksesta luoda valmis, tiheä hiusmassakokonaisuus. Kuvassa 15 näkee kuinka valmis modifier stack vaikuttaa Curves-objektin ulkonäköön. Tärkeimpiä Curves objektin modifiereita ovat “Interpolate Hair Curves” ja “Set Hair Curve Profile” nimiset modifietit.

“Interpolate Hair Curves” modifietri generoi nimensä mukaan uusia karvoja objektin hiuspintaan interpoloimalla Curves objektin olemassa olevaa käyrädataa.

Kun modifieri lisätään objektiin ensimmäistä kertaa, pitää modifierin “Surface” parametriin täsmentää mitä hiuspintaa käytetään interpoloimista varten. Tämän asetuksen lisäksi, kaksi muuta tärkeintä parametria Interpolate Hair Curves modifierissa ovat sen “Density” sekä “Mask Texture” parametrit.



Kuva 16: Kulmakarvojen erillisessä hiuspintaobjektissa näkyvä mustavalkoinen tiheyskartta.

“Density” eli interpoloitujen hiusten tiheys on helposti ymmärrettävä parametri. Luku pitää monessa tapauksessa kuitenkin nostaa melko korkealle, jotta tiheys alkaa näyttää uskottavalta. “Mask Texture” parametri viittaa myös hiusten tiheyteen. Parametriin liitetään mustavalkoinen hiusten tiheyskartta, kuten kuvassa 16. Tiheyskartta on erittäin olennainen hiusmassan muodon määrittämiselle, koska ilman sitä modifieri interpoloi koko hiuspintaobjektin pinta-alalle uutta karvaa.

“Set Hair Curve Profile” -niminen modifieri taas määrittää yksittäisen karvan profiilin muodon. “Radius” -nimisellä parametrilla voi säätää hiuksen paksuutta. Ihmisen hiuksen paksuus voi vaihdella 17 mikrometrinä jopa 181 mikrometriin (Ley 1999), joten kannattaa huomioida referenssihenkilön yksilöllisen hiustyyppin paksuutta ja missä kohtaa sen hetkinen työn alla oleva hius/karva on referenssi-

henkilössä. Pitää myös huomioida että "Radius" parametri ottaa nimensä mukaan hiuksen säteen, joten haluttu paksuusluku pitää puolittaa kun se syötetään parametrikenttään.

Muita erittäin tärkeitä modifiereita hiusmassan muodon kehittämiseksi ovat "Clump Hair Curves", "Trim hair curves" ja "Hair curves noise".

#### 5.4.4 Viimeistely

Kun karvojen muoto on saatu viimeistelyä, pitää Curves objektin modifierit leipoa "Apply" painikkeilla. "Apply:minen" pitää suorittaa järjestyksessä modifier listan huipulta alkaen. Ilman leipomista, hiuskarvoja ei pysty animoimaan Blenderin puolella ja vieminen pelimoottoriin vaikeutuu. Blenderissä hiusten animointi edellyttää kuitenkin yhden eritysmmodifierin lisäämisen hiusobjektiin. "Surface Deform" -modifieri siirtää käyriä sen hiuspinnan mukaan. Kun tämä modifieri lisätään hiusten/karvojen Curves objektiin ja Curve objektin "surface" tai hiuspinta objektia animoidaan esimerkiksi luilla, liikkuvat karvat uskottavasti pinnan mukana.

Karvoille voi luoda myös materiaalin, mutta huomioitavaa on se, että blenderissä tehtyä hiusmateriaalia ei voi siirtää Unity -pelimoottoriin.

## 5.5 Hahmon vartalo ja rigaus



Kuva 17. Hahmon kehona toimiva valmis meritaisteluasun 3D malli. Mallintajat: Viivi Piironen (Asu), Sonja Sillanpää (Asu), Ville Nissinen (Kengät).

Hahmon vartalona tulen käyttämään Kuvan 17 meritaisteluasua. Asu oli Metropolian 2024 vuoden innovaatiokurssin suorittaneiden 3D opiskelijoiden mallintama.

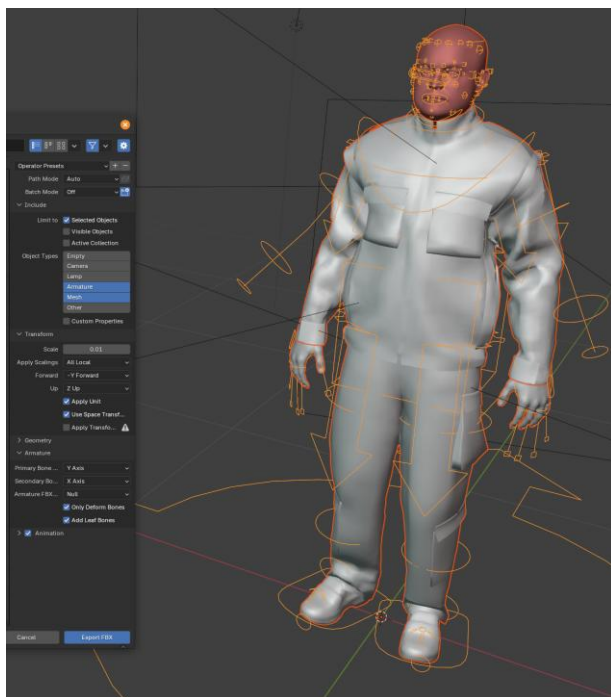
Hahmo rigataan Blenderin "Rigify" rigityökalulla. Rigify on Blenderin ilmainen lisäosa, joka ei ole oletukselta käynnistettynä. Kun lisäosa tai "addon" on käynnistetty, Blenderin 3D näkymään voi lisätä "Add" valikon (Shift + A) kautta ihmishahmon Rigify "Metarigin". Metarig on lopullisen rigin yksinkertaistettu jäljennös, jolla voi helposti asetella rigin luut ja nivelet oikeille paikoille suhteessa hahmon kehoon. Kun metarigin asettelu on valmista, rigiobjektin datapaneelistä voi "Generate Rig" painikkeella automaattisesti generoida lopullisen rigin.

## 6 Projektiosuus 2: Hahmon vienti Unity-pelimoottoriin

Projektin toisessa osuudessa kytketään hahmon eri osat toimimaan reaaliaikaisesti Unity –pelimoottorissa. Tähän tullaan käyttämään Unity demotiimin "DigitalHuman" lisäosaa, sekä tiimin "Hair" lisäosaa, jotta reaaliaikaiset splinihiukset

piirtyvät hahmolle. Netissä ei ole juurikaan dokumentaatiota kyseisistä työkaluista, joten iso osa tulevasta tiedosta on peräisin henkilökohtaisesta kokeilusta. Iso osa tiedosta on myös kerätty tutkimalla demotiimin julkaisemaa ”DigitalHuman sample” pakettia, joka sisältää valmiin ihmishahmon. Valmiilla hahmolla oli oikein kytketyt iho- ja silmämateriaalit, joista otin mallia.

## 6.1 Hahmon vieminen Blenderistä

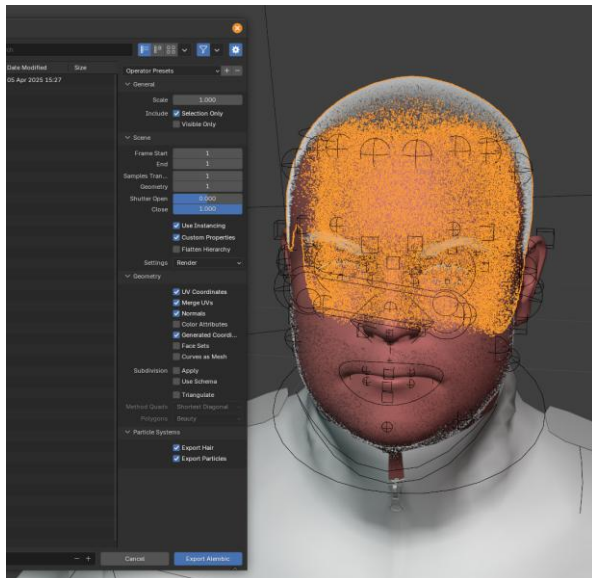


Kuva 18: FBX Vientiasetukset.

Ihmishahmo tullaan viemään Blenderistä kahdessa eri muodossa: FBX ja Alembic. Kehon meshidata, kuten pää ja vartalo viedään samanaikaisesti hahmon animoidun rigin kanssa. Tässä tapauksessa valitsen hahmon pään, hampaat, kädet, kehon ja luurangon samanaikaiseen valintaan. Luuranko tulee olemaan valinnan aktiivinen objekti. Aktiivisen objektin voi erottaa visuaalisesti siitä että sen ääriviivat ovat värjäytyneet kirkkaammalla oranssin sävyllä, kuten kuvassa 18.

FBX vientiasetuksien ”Include” osiosta muokkasin muutamia asetuksia. Aktivoitin ”Selected Objects” ruksituslaatikon, joka varmistaa että vain valitut objektit viedään. ”Object Types” moninapista voi määrittää vietyjen objektien tyyppin. Aktivoitin ”Mesh” ja ”Armature” napit, koska veitävänä on hahmon malli ja luuranko.

“Transform” osiosta “Scale” kannattaa pienentää lukuun 0.01, jotta Unityn puolella skaalat eivät olisi satakertaiset. “Armature” osiosta aktivoin “Only Deform Bones” ruksilaatikko, jotta Blender vie ainoastaan hahmon meshejä deformat luut, eikä mm. animaatorigin luita.



Kuva 19: Hiusten Alembic vientiasetukset.

Kun viedään splinihiuksia Blenderistä Unityyn ne pitää olla “Alembic” muodossa koska Blenderin FBX exporteri ei pysty viemään splinipohjaista hiusdataa. Alembic asetuksissa pitää vain kytkeä päälle “Selection Only” ja muuttaa “Frame End” ensimmäiseen ruutuun, jotta tiedostoon ei tallennu turhaa animaatiodataa.

## 6.2 Digital Human Package:n asentaminen

```
"dependencies": {
  "com.unity.demoteam.digital-human": "https://github.com/Unity-Technologies/com.unity.demoteam.digital-human.git",
  "com.unity.demoteam.hair": "https://github.com/Unity-Technologies/com.unity.demoteam.hair.git",
  ...
}
```

Teksti 2: tarvittavat paketit lisättyinä manifest.json tiedostoon. Kolme pistettä kuvastaa tiedoston/datan muuta sisältöä.

Tarvittavat Digital Human Package lisäosat voidaan asentaa Unity projektin luomisen jälkeen muokkaamalla projektin riippuvaisuusasetuksia. Näitä asetuksia

voi muokata navigoimalla Unity-projektin kansiorakenteessa "Packages/" kansioon ja muokkaamalla "manifest.json" tiedostoa. Muokkaamiseen riittää mikä tahansa tekstieditori. Unity demotiimin "DigitalHuman" paketti, sekä "Hair" paketti asentuvat lisäämällä tekstin 2 rivit "dependencies" sarakkeiden sisälle uusille riveille. (Unity Demoteam 2023) Kun manifest.json tallentuu ja Unity Editoriin palataan, alkaa Unity automaattisesti asentamaan lisäosia projektiin.

### 6.3 Pään mallin valmistelu pelimoottorin puolella

Hahmon FBX tiedoston voi tuoda Unityyn yksinkertaisesti raahaamalla tiedoston editori-ikkunan "Assets" kansioon. FBX assetin "Inspector" paneeliin pääsee painamalla sitä hiiren näppäimellä. Inspector paneelista muutetaan kaksi asetusta. "Convert Units" ruksituslaatikko kytketään pois päältä, jotta malli importtaantuu oikeassa skaalassa. "Read/Write" laatikko kytketään päälle, jotta hahmon pään meshille voidaan luoda "Skin Attachment Target" myöhemmin DigitalHuman paketin avulla. Tämä mahdollistaa splinikarvojen yhdistämisen meshipintaan (Unity Learn 2023). Lopuksi asetukset vahvistetaan painamalla "Apply" -painiketta.

Hahmon pää tarvitsee ihomateriaalin. Materiaalin shader-ohjelma, johon materiaali pohjautuu, pitää muuttaa "DigitalHuman Skin" shaderiksi. Jos DigitalHuman paketti on asennettuna, shaderin pitäisi löytyä osiosta ShaderGraphs -> DigitalHuman Skin. Seuraavaksi ihomateriaalille pitää luoda oma diffuusioprofiili. Diffuusioprofiili säättää ihomateriaalin Subsurface scatteringiä. Diffuusioprofiili lisätään editorin Assets kansioon kohdasta Create -> Rendering -> HDPR Diffusion profile. Diffuusioprofiiliassetin voi sitten lisätä ihomateriaaliin. Lopuksi Blenderissä leivotut pään tekstuurikartat lisätään ihomateriaaliin. Kun tuo Unity editoriin tekstuurikarttoja, pitää jälkeinpäin varmistaa että Unity tulkitsee tekstuurit oikein. Yksittäisen tekstuurikartan asetuksia voi säätää Inspector -paneelista. Basecolor, roughness ja specular tekstuurikarttojen "Texture Type" voi pitää oletuksella. NormalMap tekstuurikartan "Texture Type" pitää muuttaa asetukseen "Normal map". Muuten ihomateriaalin pintanormaalit näyttävät virheellisiltä. Roughness- ja specularikartat eivät ole väritekstuureita, joten niissä ei kuulu olla sRGB väritransformaatiota. Näiltä tekstuurikartoilta kytketään transformaatio pois päältä

“sRGB (Color Texture)” ruksituslaatikosta. Apply –painiketta pitää painaa, jotta tekstuurien muutokset tulevat voimaan.

Jotta pään meshin pintaan voidaan kiinnittää asioita, sille pitää luoda “Skin Attachment Data” assetti. Assetti liitetään malliin valitsemalla peliympäristössä oleva GameObject, jossa on mallia piirtävä “SkinnedMesh Renderer” -komponentti. Gameobjektiin lisätään “Skin Attachment Target” -komponentti. Komponentin “Attach Data” kenttään lisätään aikaisemmin luotu Skin Attachment Data assetti. Lopuksi data rakennetaan painamalla komponentin “Rebuild” nappia.

#### 6.4 Digital Human silmät

Yksinkertainen silmän pintamalli tuodaan normaalisti fbx muodossa Blenderistä Unityyn. Inspectorissa laitetaan fbx asetuksista taas Convert Units pois päältä. DigitalHuman “Eye Renderer” komponentin ansiosta silmän 3D mallilla ei tarvitse olla UV kartoitusta lainkaan. Eye Rendererin käyttämistä helpottaa kuitenkin se, että silmän malli on käännetty Blenderissä siten, että pupilli osoittaa Z akselin mukaisesti. Tällöin kun malli tuodaan Unityyn, sitä ei tarvitse kiertää, jotta se osotaisi Unityssä eteenpäin, eli positiivisen Z suuntaan. Silmän voi kiinnittää hahmoon jompaan kumpaan silmäluuhun “Parent Constraint” komponentilla. Luun orientaatio ei välttämättä ole oletukselta oikeanlainen (+Z eteenpäin, +Y ylöspäin), joten silmän meshiobjektin voi ensin parentoida tyhjään gameobjektiin, jonka avulla annetaan hahmon silmän luun ja itse silmän meshin välille korjaava lisäkierto. Tyhjä GameObject parentoidaan hahmon luurankoon Parent Constraint –komponentilla.

Silmälle luodaan seuraavaksi “Digital Human Eyes” shaderiin pohjautuva materiaali. Silmämateriali tarvitsee myös oman diffuusioprofiilinsa. Pelinäkymässä silmän meshille lisätään materiaali, sekä Eye Renderer komponentti.

```

void AsgParameterUpdateCPU()
{
    Vector3 osMarkerL = (1.1f * geometryRadius) * Vector3.Normalize(Vector3.forward + Vector3.left);
    Vector3 osMarkerR = (1.1f * geometryRadius) * Vector3.Normalize(Vector3.forward + Vector3.right);
    Vector3 osMarkerT = (1.1f * geometryRadius) * Vector3.Normalize(Vector3.forward + 0.35f * Vector3.up);
    Vector3 osMarkerB = (1.1f * geometryRadius) * Vector3.Normalize(Vector3.forward + 0.35f * Vector3.down);

    if (asgMarkerPolygon != null && asgMarkerPolygon.childCount == 4)
    {
        //
        // 1
        // 0 --- 2
        // 3
        //
        osMarkerL = this.transform.InverseTransformPoint(markerPositions[0]);
        osMarkerR = this.transform.InverseTransformPoint(markerPositions[2]);
        osMarkerT = this.transform.InverseTransformPoint(markerPositions[1]);
        osMarkerB = this.transform.InverseTransformPoint(markerPositions[3]);
    }
}

```

Kuva 20: EyeRenderer.cs skriptistä kuvankaappaus. Ohjelmoija on kommenttien avulla luonut diagrammin joka selittää asg markkereiden sijoittamisen silmän ympärille.

Eye Renderer –komponentti ei kuitenkaan toimi vielä, koska siltä puuttuu referenssi “asg polygoniin”. Eye Renderer pystyy luomaan tällaisen Asg polygonin avulla silmälle silmäonkalon luoman reunavarjostuksen. Tällainen järjestelmä on kehitetty Eye Rendererille todennäköisesti sen takia, koska tavallisten renderöintimenetelmien avulla efektin luominen voisi tuottaa hitaita ja huonolaatuisia tuloksia. Itse Asg polygoni on erittäin yksinkertainen. Unityscenenen pitää vain lisätä tyhjä GameObject, jolla on neljä tyhjää GameObject lasta. Nämä neljä lapsiobjektia sitten asetellaan silmäkulmiin kuvan 20 koodikommentin mukaisesti, eli silmän sisäkulmaan, yläkeskiosaan, ulkokulmaan ja alakeskiosaan omat gameobjektit. Lopulta neljän markkeriobjektin gameobject vanhempi lisää Eye Renderer –komponentin “Asg Marker Polygon” kenttään.

Asg markerit pitää vielä kiinnittää silmäluomien meshiin. Tämä onnistuu käyttämällä aikaisemmin luotua pään Skin Attachment Data assettia. Jokaiselle neljälle silmän Asg marker gameobjektille pitää lisätä “Skin Attachment” skriptikomponentti. Komponentin asetuksissa “Target” kenttään pistetään pään meshin kiinnitysdata. Lopulta komponentin “Attach” painikkeen avulla markeri kiinnitetään meshin pintaan.

## 6.5 Hiukset ja karvat Unityssa

Hiusten ja karvojen tuominen toimii hiukan erilailla Unityssa. Unityn “Package Managerista” eli paketinhallinnasta pitää ottaa käyttöön Alembic paketti, jotta

hiusten ja karvojen alembic tiedostoja voi tuoda moottoriin. Paketin asentamisen jälkeen Alembic tiedostot voi tuoda Unityn Asset –kansioon hiirtä raahaamalla. Tuomisen jälkeen, alembic assetin tuontiasetuksista “Scale Factor” arvo pitää muuttaa arvoon 1.0, jotta hiusmalli olisi tuotuna oikeassa koossa moottoriin. Jos Demoteam Hair –paketti on asennettuna, editorin Assets –kansioon pitäisi pystyä luomaan uusi assetti nimeltä “Hair Asset”. Inspector –ikkunasta voi muokata Hair Assetin asetuksia. Asetuksien “Settings Basic” osiosta vaihetaan “Type” asetus asentoon “Alembic”. Seuraavaksi assetin asetuksiin ilmestyy “Settings Alembic” osio. Tämän osion asetuksista määritellään “Alembic Asset” kohtaan haluttujen hiuksien/karvojen alembic tiedosto. “Strand Diameter” asetuksilla voi muokata yksittäisten hiuksien paksuutta, jos alembic tiedostoon tallennetut paksuustiedot ovat vääränlaiset. Lopuksi painamalla “Build Strand Groups”, Unity generoi Hair Assetille hiuksidatan alembic assetin pohjalta (Unity Learn 2023).

Hius Assetin asettaminen hahmolle toimii lisäämällä Unity scenessä tyhjälle GameObjectille “Hair Instance” nimisen komponentin. Hair Instance komponentin “System Contents” osiossa on valintalaatikko, johon voi lisätä halutun Hair Assetin. Kun Hair Asset on määritetty komponenttiin, painetaan “Reload” painiketta (Unity Learn 2023).

Jotta hiukset kiinnittyisivät hahmon pintaan (tässä tapauksessa pään meshiin), hiusten asetuksiin myös määritetään pään “Skin Attachment Data”. Määrittely onnistuu menemällä Hair Instance komponentin “Strand Settings” osioon ja avaamalla asetukset painamalla pientä kynän kuvaketta. Asetuksien “Settings Skinning” osion “Roots Attach” asetukseen voi määritellä pään Skin Attachment Data:n (Unity Learn 2023). Hiuksien pitäisi nyt liikkua hahmon mukana. “Strand Settings” tai tästä lähtien säieasetuksissa, tullaan muuttamaan monta muutakin asetusta. Säieasetuksien “Settings Physics” osiosta voi muokata hiussimulaation käyttäytymistä. Tätä kouluttaja-avataria varten säädän asetuksia siten, että simulaatio käytännössä kytketään pois. Tämä onnistuu säätämällä “Global Position” asetusta täysin numeroon, 1.0, ja “Interval” pudotusvalikosta valitsemalla 1 millisekunnin vaihtoehdon.

Hiusten materiaalin voi muuttaa säieasetuksien “Settings Rendering” osion materiaalikohdasta. Demotiimin Hair lisäosan mukana ei tule valmista hius shaderia, kuten iholle ja silmille, mutta sisällettynä on Shader Graph noodi, jolla voi rakentaa oman hius-shaderin. Noodi on nimeltään “Hair Vertex”. Lisäämällä noodi shader graph näkymään ja liittämällä noodin “surfacePositionOS”, “surfaceNormalOS” ja “surfaceTangentOS” ulostulot verteksi ulostulon “Vertex”, “Normal” ja “Tangent” sisääntuloihin, alkaa materiaali jo piirtämään hiuksia (Unity Demo-team 2023).

## 7 Tulos



Kuva 21: Kuvakaappaus hahmosta Unity-pelimoottorista.



Kuva 22: Kuvakaappaus hahmon Blender versiosta.

Varsinainen lopputulos on liitteen 2 reaaliaikainen applikaatio. Kuvat 21 ja 22 ovat staattisia näytteitä hahmosta.

## 8 Loppusanat

Opinnäytetyön tavoitteena oli reaaliaikaisen kouluttaja-hahmon tuotannon proses-  
sikuvaus ja dokumentaatio. Unity pelimoottorissa kootu lopputulos on näyte siitä  
kuinka DigitalHuman Unity -lisäosien käyttöönotto on mahdollista pienemmänkin  
tuotannon kontekstissa. Kuitenkin, lisäosien vähäisen dokumentaation takia, tä-  
män projektin lopputulos ei ole aivan samalla tasolla kuin Unity demotiimin toteu-  
tuksissa. Tilaa hahmon parantamiselle on vielä. Mm. kasvojen animointi ilmeni  
suureksi haasteeksi. Uskottavien ilmeiden ja silmien räpäytyksien toteuttaminen  
vaatisi vaihtuvien tekstuurikarttojen lisäksi naaman meshille "Blend shapeja".  
Tällä hetkellä hahmon ilmeitä ajaa vain luut.

Realististen 3D ihmishahmojen saavuttaminen on ollut vuosikymmenien aikana  
haasteena viihdeteollisuudessa. Edistykset ihmishon renderöinnissä ja kasvojen

uskottavassa animoimisessa, sekä näyttelijöiden piirteiden kaappaamisessa teknologioiden, kuten Light Stage kuvaaminen, ovat olleet suuria. Realistiset ihmis-kasvot on edelleen erittäin haastava tavoite 3D artistille, vaikka tuotantoa helpottavia työkaluja ja teknologioita on ilmestynyt.

## Lähteet

Price, Andrew 2016. Photorealism Explained. Verkkovideo.  
<https://www.youtube.com/watch?v=R1-Ef54uTeU>

Ratajczyk Dawid 2019. Uncanny Valley in Video Games: An Overview. Sivu 137.  
[https://www.researchgate.net/publication/338663039\\_Uncanny\\_Valley\\_in\\_Video\\_Games\\_An\\_Overview](https://www.researchgate.net/publication/338663039_Uncanny_Valley_in_Video_Games_An_Overview)

d'Eon, Eugene, Luebke, David. 2007. GPU Gems 3, Chapter 14. Advanced Techniques for Realistic Real-time Skin Rendering. Kirja.  
<https://developer.nvidia.com/gpugems/gpugems3/part-iii-rendering/chapter-14-advanced-techniques-realistic-real-time-skin>

Debevec, Paul. 2012. The Light Stages and Their Applications to Photoreal Digital Actors.  
<https://ict.usc.edu/pubs/The%20Light%20Stages%20and%20Their%20Applications%20to%20Photoreal%20Digital%20Actors.pdf>

Nvidia, 2013. GTC 2013: Face Works (2 of 11). Verkkovideo.  
<https://www.youtube.com/watch?v=5d1ZOYU4gpo>

Reed, Nathan. 2014. Advanced Skin Shading with FaceWorks.  
<https://www.reedbeta.com/talks/faceworks/gtc-2014-faceworks.pdf>

Activision R&D, 2013. Activision R&D Real-time Character Demo. Verkkovideo  
<https://www.youtube.com/watch?v=l6R6N4Vy0nE>

Epic Games, 2021. Announcing MetaHuman Creator. Verkkosivu.  
<https://www.epicgames.com/site/en-US/news/announcing-metahuman-creator-fast-high-fidelity-digital-humans-in-unreal-engine/>

Schoennagel, Mark. 2023. Creating Realistic Digital Humans in Unity featuring "Enemies". Verkkovideo.  
<https://www.youtube.com/watch?v=tVoB9s6YJo4>

Zmejevskis, Lukas. 2022. Cross Polarization Photogrammetry. Artikkel.  
<https://www.pix-pro.com/blog/cross-polarization>

Courtois, Leia. 2024. How reference pictures can be used to perfect the likeness. Blogipostaus.  
[https://www.linkedin.com/posts/leia-courtois\\_characterartist-arnold-schwarzenegger-sculpting-activity-7211696310817615872-Kmn4](https://www.linkedin.com/posts/leia-courtois_characterartist-arnold-schwarzenegger-sculpting-activity-7211696310817615872-Kmn4)

Blender Manual. Multiresolution Modifier.  
<https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/multi-resolution.html>

Blender Developer Documentation, 2022. Blender 3.2 Release notes, Sculpt, Paint, Texture. Verkkosivu.  
[https://developer.blender.org/docs/release\\_notes/3.2/sculpt/](https://developer.blender.org/docs/release_notes/3.2/sculpt/)

Wikipedia, 2025. Gamma correction. Artikkel.  
[https://en.wikipedia.org/wiki/Gamma\\_correction](https://en.wikipedia.org/wiki/Gamma_correction)

Ley, Brian, 1999. Diameter of a Human Hair. Verkkosivu.  
<https://hypertextbook.com/facts/1999/BrianLey.shtml>

Unity Demoteam, 2023. Digital Human Package README. Verkkosivu.  
<https://github.com/Unity-Technologies/com.unity.demoteam.digital-human>

Unity Demoteam, 2023. Digital Human Hair Package README. Verkkosivu.  
<https://github.com/Unity-Technologies/com.unity.demoteam.hair>

Unity Learn, 2023. Set up Character Hair. Verkkosivu.  
<https://learn.unity.com/tutorial/set-up-character-hair?uv=2021.2&projectId=635a945cedbc2a39658709de#636076b4edbc2a3971b08c18>

Unity Manual, 2025. Diffusion Profile.  
<https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/Diffusion-Profile.html>

## Kuvalähteet

Kuva 1. Ahmad Azaini Manaf, Fatihah Ismail, Mohd Rosli Arshad & SungPil Lee. 2022. Familiarity and Overcoming of Uncanny Valley towards Computer-Generated Imagery Characters in Malaysian Film. Sivun 182.  
[https://www.researchgate.net/publication/367178663\\_Familiarity\\_and\\_Overcoming\\_of\\_Uncanny\\_Valley\\_towards\\_Computer-Generated\\_Imagery\\_Characters\\_in\\_Malaysian\\_Film#pf2](https://www.researchgate.net/publication/367178663_Familiarity_and_Overcoming_of_Uncanny_Valley_towards_Computer-Generated_Imagery_Characters_in_Malaysian_Film#pf2)

Kuva 2. Nvidia. "Ira" Lifelike human face rendering. 2013. Verkkosivu.  
<https://www.nvidia.com/en-us/geforce/community/demos/>

Kuva 4. Blender Studio. Realistic Human Base. 2022. Verkkosivu.  
<https://studio.blender.org/training/realistic-human-research/chapter/design-blocking-sculpting/>

Kuva 11. Martin Guldbaek. Color Zones of the Face. Verkkosivu.  
<https://goolbegg.artstation.com/blog/ygg3/color-zones-of-the-face>

Kuva 13. Hansen, Helena. So.. How to haircards? 2023. Verkkosivu.  
<https://www.artstation.com/blogs/helenaehhansen/NN4G/so-how-to-haircards>

## **Liitteet**

### **Automaattinen tekstuurikarttojen leivontaan tarkoitettu python skripti**

Python skripti ajetaan blender ohjelmassa.

**Realiaikainen demoapplikaatio, jossa kouluttaja-avatar**

Yksinkertainen Unity-applikaatio, jossa voi pyöritellä kameraa hahmon ympärillä. Kameraa voi kääntää liikuttamalla hiirtä samalla kun hiiren vasen painike on painettuna pohjaan.