

Laura Eronen

3D-vihollishahmon mallintaminen



Tradenomi
Graafikko
Syksy 2022



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä(t): Eronen Laura

Työn nimike: 3D-vihollishahmon mallintaminen

Tutkintonimike: Tradenomi (amk), tietojenkäsittely

Asiasanat: 3D, videopeli, pelihahmo, 3D-mallintaminen

Tässä opinnäytetyössä käsitellään 3D-vihollishahmon luomisen eri työvaiheita suunnittelusta valmiin 3D-mallin riggaamiseen. Työ toteutettiin toimeksiantajan pyynnöstä ja työn tarkoituksena oli dokumentoida, miten 3D-pelihahmoja toteutetaan työympäristössä.

Opinnäytetyö oli jaettu teoria- ja projektiosuuteen, joista teoriaosuudessa käytiin läpi 3D-hahmonluomisen eri vaiheet ja kerrottiin, mitä tulee ottaa huomioon hahmoja suunnitellessa. Projektiosuudessa vuorostaan suunniteltiin ja mallinnettiin toimeksiantajan työtehtävän pohjalta vihollishahmo käyttäen Blenderiä. Mallin tekstuurit luotiin beikkaamalla ne raskasmallista mallinnusohjelman sisällä ja yhdistämällä ne yhdeksi tekstuuriatlakseksi käyttäen Photoshopia.

Työn lopputuloksena saatiin valmiiksi rigattu hahmo toimeksiantajan peliprojektia, Beverage: the Last Callia, varten. Hahmon animaatioiden tekeminen jouduttiin ajanpuutteen vuoksi jättämään projektista pois, mutta hahmo kerrettiin testata peliprojektin sisällä.

Abstract

Author(s): Eronen Laura

Title of the Publication: Modeling a 3D Enemy Character

Degree Title: Bachelor of Business Administration. Business Information Technology

Keywords: 3D, video game, video game character, 3D modeling

This thesis will go over the different work stages of creating a 3D enemy character, from planning to rigging the finished 3D model. This work was carried out by the request of the client and the objective of this Bachelor's thesis was to document how 3D game characters are produced in a work environment.

The thesis was divided into a theory and a project part. The theory part goes through the different stages of 3D character creation and explains what should be considered when designing characters. In the project part, the enemy character was designed and modeled based on the client's assignment using Blender. The textures of the model were made by baking them from the high-poly model within the 3D package and combining them into one texture atlas using Photoshop.

The result of the work was a fully rigged character for the client's game project, Beverage: The Last Call. Due to lack of time, the animations for the character had to be left out of the project, but the character was tested within the game project.

Sisällys

1	JOHDANTO.....	4
2	HAHMOSUUNNITTELU.....	5
2.1	Hahmosuunnittelun perusperiaatteet	5
2.2	Hahmosuunnittelun työnkulku	5
3	3D-MALLINTAMISEN TYÖNKULKU.....	8
3.1	Model sheet	8
3.2	3D-mallintaminen.....	9
3.3	3D-veistäminen	11
3.4	Unwrappaaminen.....	12
3.5	Beikkaaminen	13
3.6	Teksturointi	15
3.7	Riggaus	17
4	PROJEKTITYÖ	21
4.1	Suunnittelu	21
4.2	3D-mallintaminen.....	22
4.2.1	Pää.....	22
4.2.2	Ruumis.....	26
4.2.3	Jalat	28
4.2.4	Siivet.....	29
4.2.5	Yksityiskohtien veistäminen	30
4.3	Mallin optimointi.....	32
4.3.1	Retopologia	32
4.3.2	Unwrappaus	32
4.3.3	Beikkaus	34
4.3.4	Teksturointi	36
4.3.5	Riggaus	37
4.4	Testaus	39
5	JOHTOPÄÄTÖKSET	41
6	LÄHTEET.....	42

SYMBOLILUETTELO

2D-sprite	Osittain läpinäkyvä kaksiulotteinen grafiikkamenetelmä, jota hyödynnetään peliympäristön liikkuvissa elementeissä, kuten hahmoissa.
3D-malli	Mallinnettu tai veistetty digitaalinen kappale, joka koostuu polygoneista.
Beikkaus	3D-mallin pinnan datan siirtäminen toiselle mallille muuttamalla se tekstuurikartaksi (engl. bake).
Diffuusikartta	Tekstuurikartta, joka määrittää 3D-mallin pinnan värin (engl. diffuse map).
Digitaalinen veistäminen	3D-mallin manipulointi käyttäen työkaluja, joilla voi vetää, työntää ja silottaa mallin pintaa (engl. digital sculpting, 3D sculpting).
Kevytmalli	3D-malli, jonka polygonimäärä on mahdollisimman matala (engl. low-poly model).
Materiaali	Tietokonegrafiikassa materiaali sisältää väritiedon lisäksi muitakin ominaisuuksia, kuten kiiltoisuutta tai metallisuutta.
Modifikaattori	3D-mallinnusohjelman automatisoitu toiminto, jolla voi muokata mallia muuttamatta sen perusgeometriaa (engl. modifier).
Normaalikartta	Tekstuurikartta, joka vaikuttaa siihen, miten valo osuu 3D-mallin pintaan (engl. normal map).
Pelimoottori	Ohjelmisto, jolla voi luoda pelejä.
Polygoni	Kolmi- tai nelikulmaisia monikulmioita, joista 3D-malli koostuu.
Raskasmalli	3D-malli, jonka polygonimäärä on suuri (engl. high-poly model).
Reuna	Polygonin reuna, joka yhdistää kaksi verteksiä (engl. edge).
Teksturoidi	Tekstuurikarttojen luominen ja 3D-mallin pinnoittaminen niiden avulla.
Topologia	3D-mallin polygonien sommittelu, jolla määritellään, miten osat liittyvät toisiinsa tai järjestetään.

UV-kartta	2D-tila, johon 3D-mallin pinta on purettu.
Unwrappaus	3D-mallin pintojen ”purkaminen” kaksiulotteiseen UV-tilaan luomalla mallin reunoista saumoja (engl. unwrapping).
Varjostin	Ohjelmisto, joka määrittää miten 3D-ympäristö piirtyy näytölle (engl. shader).
Verteksi	Polygonin kulmapiste (engl. vertex).

1 JOHDANTO

3D-mallintaminen ja hahmosuunnittelu ovat kehittyneet paljon viimeisen 30 vuoden aikana. 3D-pelien pioneereina voidaan nähdä Quake (1996) ja Super Mario 64 (1996), jotka olivat ensimmäisiä 3D-pelejä, jotka hyödynsivät 3D-malleja myös pelaaja- ja vihollishahmoissa, eikä vain peliympäristössä. Aikaisemmat 3D-pelit, kuten Wolfenstein 3D (1992) ja Doom (1993) hyödynsivät 3D-grafiikkaa peliympäristössä, mutta hahmot koostuivat 2D-spriteistä, jotka esitettiin useista esitetyistä katselukulmista.

Opinnäytetyön päätarkoitus oli kehittää hahmosuunnittelu- ja 3D-mallintamistaitoja sekä luoda materiaalia aiheesta kiinnostuneille. Työn aikana käydään läpi hahmosuunnittelun eri työvaiheet sekä 3D-mallintamisen ja -veistämisen perusteet. Opinnäytetyön projektiosuudessa tullaan suunnittelemaan ja mallintamaan pelivalmis 3D-vihollishahmo, jonka toimeksiantaja toimii kajaanilainen peliyhtiö, Rust0 Games Oy.

2 HAHMOSUUNNITTELU

Hahmosuunnittelu on tärkeä osa kaikkia tarinaan perustuvia mediatyyppejä, olipa ne sitten elokuvia, kirjoja tai videopelejä. Hahmojen avulla katsoja saadaan uppoutumaan tarinaan ja siksi hahmosuunnittelua varten on varattava tarpeeksi paljon aikaa, jotta hahmot voidaan rakentaa perusteellisesti. [1.]

2.1 Hahmosuunnittelun peruseriaatteet

Hahmot voidaan jakaa kahteen ryhmään: taidevetoisiin hahmoihin ja tarinavetoisiin hahmoihin. Taidevetoisia hahmoja usein käytetään peleissä, jotka keskittyvät enemmän pelattavuuteen ja kognitiiviseen uppoutumiseen, jolloin hahmojen tausta voidaan jättää tekemättä tai tarvittavat tiedot lisätä jälkepäin, kun hahmon visuaalinen muoto on saatu valmiiksi. Koska pelaaja ei saa hahmosta mitään selville tarinan kautta, on hahmon tuotava tarina ja luonteen piirteet esille sen ulkomuodolla. Luonteen piirteitä ja hahmon tarinan voi ilmaista monilla eri tavoilla, kuten muodoilla, väreillä, asusteilla, ilmeillä, äänillä ja asennoilla. Hahmon ominaisuuksia voidaan sen lisäksi liioitella, jotta niiden ilmaiseminen pelaajalle on selkeää. [1.]

Tarinavetoiset hahmot vuorostaan tuovat piirteitään esille luonteen ja tapojen avulla, mikä tarkoittaa, että kaikkia hahmon ominaisuuksia ei tarvitse tuoda esille sen ulkomuodon kautta. Tämä ei kuitenkaan tarkoita sitä, että hahmon ulkonäöllä ei ole merkitystä. Hahmon on oltava huomiota herättävä ja kiinnostava, mutta myös uskottava pelin yhteydessä. Pelin hahmojen on myös erotuttava toisistaan, jotta pelaaja ei sekoita niitä toisiinsa. [1;2.]

2.2 Hahmosuunnittelun työnkulku

Hahmosuunnittelu usein aloitetaan luomalla ohje, jota noudatetaan hahmon designia suunniteltaessa. Tämä ohje voi olla niinkin yksinkertainen kuin pieni valikoima avainsanoja, jotka määrittelevät hahmon tärkeimmät ominaisuudet, kuten koon, iän, muodon, käyttäytymisen ja niin edel-

leen. Kun lista on luotu, voidaan sen vastapainoksi lisätä sanoja, jotka ovat käsitteellisesti vastakkaisia alkuperäisten sanojen kanssa, mikä antaa hahmolle kontrastia ja syvyyttä. Avainsanaluettelo muutetaan sitten kirjalliseksi kuvaukseksi hahmosta eli high conceptiksi. High conceptia voi verrata low conceptiin, joka käsittelee hahmon kehityksiä ja muita yksityiskohtia, joita ei ole helppo tiivistää. [2;3;4.]

Kun hahmon kirjallinen kuvaus on valmis, seuraava vaihe on yleensä alustavien luonnosten ja konseptitaiteen luominen. Luonnostelutapoja on monia, mutta tyypillisin tapa on luoda hahmosta mustavalkoisia siluetteja. Siluetteja voidaan kutsua thumbnailiksi ja niiden tarkoituksena on antaa taiteilijan tutkia useita erilaisia ideoita nopeasti, joista on esimerkki kuvassa 1. Kun thumbnailia on tarpeeksi, taiteilija ja hänen tiiminsä voivat valita mieluisimmat kuvat sekä yhdistellä niiden osia keskenään. Valittuja kuvia työstetään sitten eteenpäin lisäämällä niihin enemmän yksityiskohtia ja harmaansävyjä antaen hahmolle kolmiulotteisemman muodon. Kun hahmon siluetti on valmis, voidaan siihen lisätä värejä, joko maalaamalla värit suoraan thumbnailin päälle tai piirtämällä hahmolle yksityiskohtaisempi viivapiirros. Mustavalkoisten siluettien tapaan taiteilija maalaa useita väri variaatioita hahmolle tutkien samalla, mitkä niistä toimivat parhaiten. [2;3.]



Kuva 1. Robotti ötököiden siluetteja.

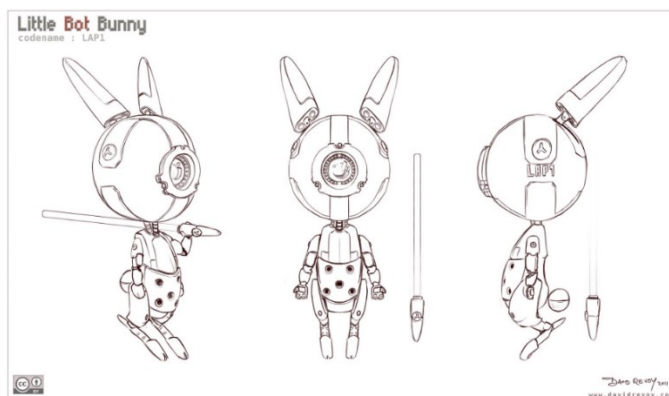
Viimeisenä vaiheena valmiille hahmolle tehdään yksityiskohtainen character sheet, jota muut taiteilijat, kuten animaattorit ja 3D-mallintajat voivat käyttää apuna. Tyypillinen character sheet sisältää useita kuvia hahmosta eri asennoissa ja kuvattuna eri kuvakulmista. Jos hahmolla on asusteita tai rekvisiittaa, voidaan niistäkin tehdä erillisiä lähikuvia sekä lyhyitä kuvailevia kuvatekstejä ja muistiinpanoja. [2.]

3 3D-MALLINTAMISEN TYÖNKULKU

3D-mallintamista voi lähestyä monelta eri taholta riippuen teknisistä rajoituksista, pelin taiteellisesta tyylistä ja työntekijöiden hierarkiasta. Uuden sukupolven konsolit ja pelitietokoneet kykenevät pyörittämään raskaampia pelejä paremmin mobiililaitteisiin verrattuna, mikä tarkoittaa, että hahmoa suunniteltaessa on hyvä ottaa huomioon kohdelaite, jolle peliä rakennetaan. Kohdelaitteen rajoitukset voivat myös vaikuttaa pelin taidesuuntaan; mobiilipelit ovat usein tyylieltyjä ja yksinkertaisia, kun taas konsoli- ja tietokonepelit pyrkivät tekemään vaikutuksen grafiikoillaan. Teknisten rajoitusten lisäksi työntekijöiden määrä vaikuttaa työnkulkuun. Suurissa AAA-peliyhtiöissä hahmosuunnittelun eri työvaiheet voidaan jakaa monelle eri työntekijälle. Yksi työntekijä voi työskennellä konseptitaiteen parissa, toinen hoitaa 3D-mallintamisen ja kolmas hahmon animoimisen. Pienemmissä yhtiöissä yksi henkilö on usein vastuussa kaikista työvaiheista.

3.1 Model sheet

Ennen 3D-mallintamisen aloittamista on hahmolle luotava model sheet. Model sheet toimii 3D-mallintajan apuna mallinnusvaiheessa, mistä on hyötyä varsinkin silloin, kun mallintaja on eri henkilö kuin hahmon suunnittelija. Kuvassa 2 on tyypillinen model sheet, jossa hahmo on kuvattu kolmesta eri ortografisesta kuvakulmasta: edestä, sivulta ja kulmasta. Jos hahmo on hyvin yksityiskohtainen, on model sheettiin syytä lisätä kuvia tarpeellisista kohdista epäselvyyksien välttämiseksi. [5;6.]



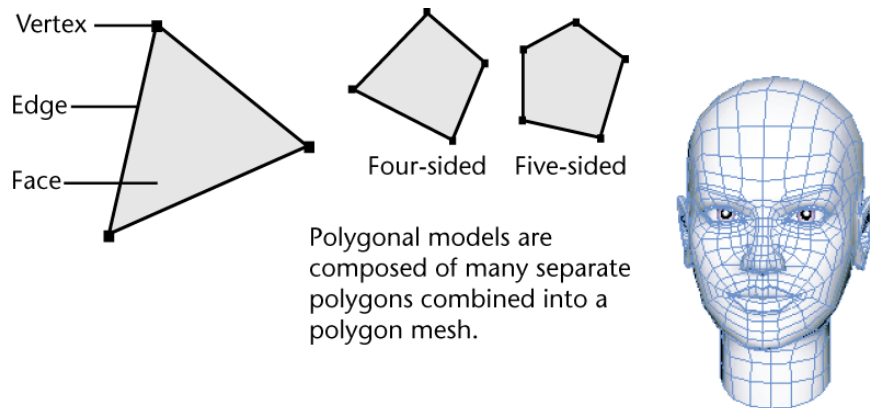
Kuva 2. Model sheet, jossa hahmo on kuvattuna kolmesta eri kuvakulmasta. (Revoy, D. 2011.)

Kun model sheet on valmis, voidaan se tuoda valittuun 3D-mallinnusohjelmaan. Siellä se asetetaan tekstuuriksi kahdelle tai useammalle 3D-tasolle, jotka näkyvät taustalla hahmoa mallintamassa. [5;6.]

3.2 3D-mallintaminen

Tapoja mallintaa on monia riippuen mallinnusohjelmasta ja itse mallista, mutta tämän opinnäytetyön tarkoituksena on kuitenkin käsitellä vain peliteollisuudessa käytettyjä mallintamistapoja. Peliteollisuudessa käytettävistä ohjelmista yleisimmät ovat Blender, Autodesk Maya ja Autodesk 3ds Max. Mallinnusohjelman sisällä 3D-malli rakennetaan luomalla kolmiulotteisia pintoja, joita kutsutaan polygoneiksi. Kuvassa 3 selostetaan, miten polygonit ovat geometrisia pintoja, jotka koostuvat vähintään kolmesta verteksistä ja niiden välisistä reunoista. Verteksi on ikään kuin piste kolmiulotteisessa tilassa, ja sillä on myös normaalisti kutsuttu ominaisuus, joka määrittää suunnan, johon verteksi osoittaa. Yhdistämällä verteksit toisiinsa luodaan reunoja niiden välille. Verteksin ja reunojen muodostamaa muotoa kutsutaan polygoniksi. [2.]

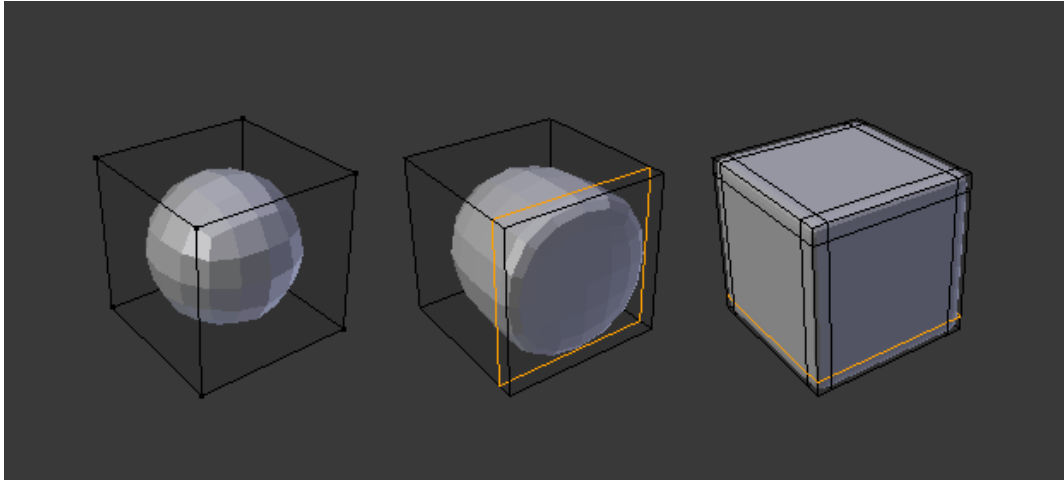
Yleisesti 3D-mallin polygonit koostuvat neljästä verteksistä ja näitä polygoneja kutsutaan nelikulmioiksi. Jos polygonissa on viisi tai useampi verteksi, niitä kutsutaan silloin n-goneiksi. N-gonien ja kolmionmuotoisten polygonien käyttäminen malleissa ei yleisesti ottaen ole suotavaa, sillä ne saattavat aiheuttaa ongelmia renderöidessä valaistuksia tai mallia animoitaessa. Ongelmista huolimatta suurin osa pelimoottoreista muuttaa nelikulmiot kolmioiksi automaattisesti, koska graafiset laitteistot renderöivät niitä nopeammin. Jos pelimoottorin luoma triangulointi eli kolmioiksi muuttaminen ei vastaa haluttua lopputulosta, taiteilija voi trianguloida mallin manuaalisesti mallinnusohjelmassa. [2;7.]



Kuva 3. Polygonin eri osat selostettuna. (Autodesk Knowledge Network. 2016.)

Kaksi yleisintä tapaa polygonimallintamisessa on laatikko- ja reunamallinnus. Laatikkomallinnus tarkoittaa mallintamista, jossa mallintaja aloittaa mallintamisen yksinkertaisesta muodosta, kuten kuutiosta, sylinteristä tai pallosta. Reunamallinnuksella tarkoitetaan mallin rakentamista mallintamalla silmukoita yksittäisistä polygoneista, jotka luovat mallin muodon. Mallintaja voi muokata 3D-mallia useilla tavoilla, kuten polygonien ja reunojen ekstruusiolla, luomalla uusia reunoja polygonien läpi sekä liikuttelemalla yksittäisiä verteksejä tai verteksiryhmiä. [2.]

Mallinnustyökalujen lisäksi mallintaja voi hyödyntää modifikaattoreita. Modifikaattorit ovat automaattisia toimintoja, joiden avulla mallia voidaan muokata tuhoamatta sen topologiaa tai alkuperäistä muotoa. Modifikaattorit vaikuttavat ainoastaan siihen, miten malli renderöidään ohjelman sisällä, mikä tarkoittaa, että toiminnot eivät ole pysyviä ennen kuin ne kollapsoidaan. Yleisimmin käytettyjä modifikaattoreita ovat Mirror, Subdivision Surface, Twist ja Bend. Mirror-modifikaattoria käytetään symmetristen mallien luomiseen, koska se tekee mallista peilatun kopion mallintajan valitseman symmetria-akselin perusteella. Mirror-modifikaattorin avulla mallintaja voi keskittyä mallintamaan vain toisen puoliskon mallista, sillä se tekee muutokset automaattisesti peilatulle puoliskolle. Kuvassa 4 esitelty Subdivision Surface -modifikaattori lisää mallin polygonimäärää sen parametrien mukaan. Mallintaja voi myös päättää halutaanko toiminnolla myös pyöristää mallia tai vain lisätä polygoneja mallin muotoa muuttamatta. Twist- ja Bend-toiminnoilla mallia voi kiertää ja taivuttaa valitun akselin mukaisesti. Twist-modifikaattoria voi käyttää esimerkiksi ruuvien mallintamiseen ja Bend-modifikaattoria renkaiden mallintamiseen. [2;8;9;10.]

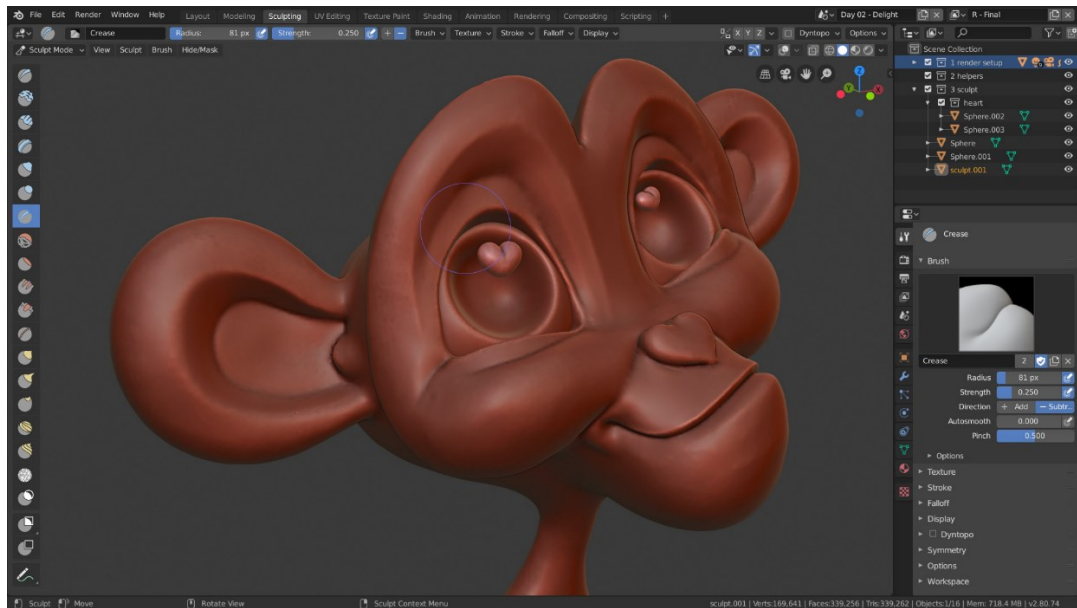


Kuva 4. Kuutiolle lisätty Subdivision Surface -modifikaattori, johon on lisätty reunasilmukoita. (Blender Documentation Team.)

3.3 3D-veistäminen

3D-veistäminen, tai toiselta nimeltään digitaalinen veistäminen, on tapa muotoilla 3D-objekti digitaalisesta massasta, jota voi muovata samalla tavalla kuin oikeaa savea. Veisto-ohjelman sisäisillä työkaluilla massaa voi muokata työntäen, vetäen, puristaen ja tasoittaen, mikä mahdollistaa yksityiskohtaisien veistoksien luomisen, kuten kuvassa 5 on tehty. Veistämisen voi aloittaa alusta rakentamalla yksinkertaisen mallin itse, tai käyttämällä valmista pohjamallia. Veistäjä voi käyttää yhden veistoksen luomiseen satojakin tunteja mallin monimutkaisuudesta ja veistäjän taidosta riippuen.

Kuten oikeankin veistoksen veistäminen, digitaalinen veistäminen aloitetaan yksinkertaisista muodoista. Kun veistäjä on tyytyväinen tulokseensa, voi hän siirtyä hienompiin yksityiskohtiin. Koska digitaalinen veistäminen käyttää paljon tietokoneresursseja ja vaatii huomattavaa prosessointitehoa, ohjelma hidastuu mitä enemmän topologiaa veistoksella on. [11.]



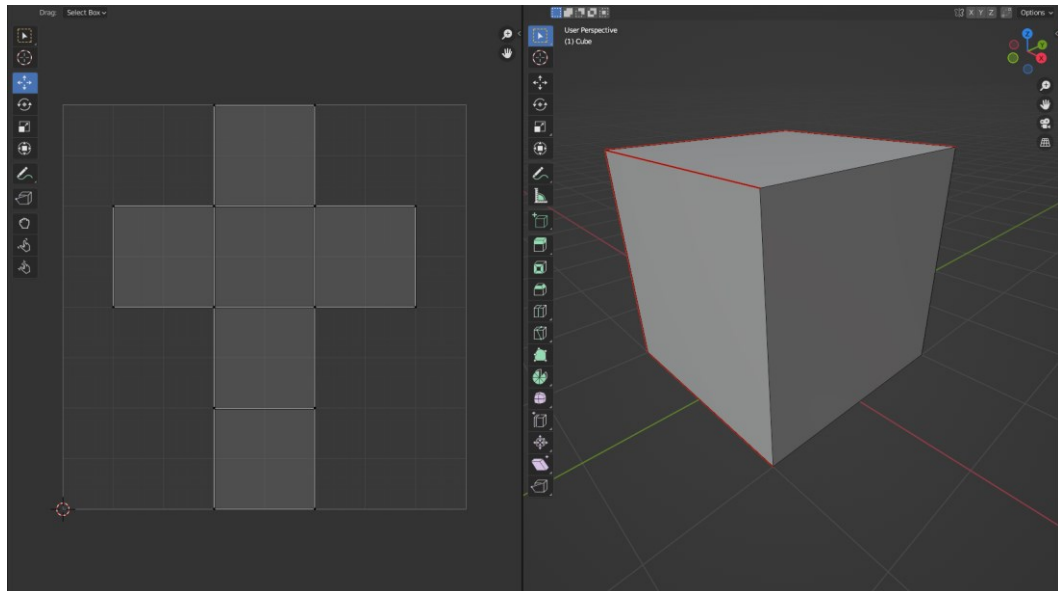
Kuva 5. Blenderissä veistetty apinahahmo. (Blender Documentation Team.)

3.4 Unwrappaaminen

3D-mallintamalla hahmo saa kolmiulotteisen muodon, mutta ilman tekstuureja mallin pinta on tyypillisesti yksivärinen. Tekstuureja ei kuitenkaan voi lisätä 3D-mallin pinnalle ennen kuin se on unwrapped eli muutettu kaksiulotteiseksi. Tätä kaksiulotteista kuvaa kutsutaan UV-kartaksi, jossa U ja V kirjaimet kuvastavat vaak- ja pystysuunnassa olevia koordinaatteja, joiden välille kartta piirretään.

Unwrappaaminen aloitetaan valitsemalla mallin reunoja, jotka muutetaan saumoiksi. Nämä saumat toimivat samalla tavoin kuin oikean elämän vaatekappaleiden saumat, ja niiden avulla mallin eri osat leikataan irti toisistaan, kuten kuvassa 6 on esitelty. Joissain tapauksissa tietokone osaa tehdä unwrappingin itse, mutta mitä monimutkaisemmasta mallista on kyse, sen vaikeampaa tietokoneen on tuottaa toimiva ja käyttökelpoinen UV-kartta itse. Kun malli on unwrapped, UV-saaret järjestellään UV-kartalle joko manuaalisesti tai automaattisesti käyttäen mallinnusohjelman pakkausohjelmaa. Koska UV-kartalla ei ole paljon tilaa, on UV-saaret aseteltava mahdollisimman lähellä toisiinsa. UV-saarten tekstuurit saattavat tällöin vuotaa viereisille

saarille, etenkin tekstuurin resoluutiota muuttaessa. Yleisesti ottaen UV-saaria ei myöskään kannata asettaa toistensa päälle, paitsi jos malli on symmetrinen, jolloin symmetriset saaret voidaan peilata ja asettaa vastaavien saarien päälle. [12;13.]



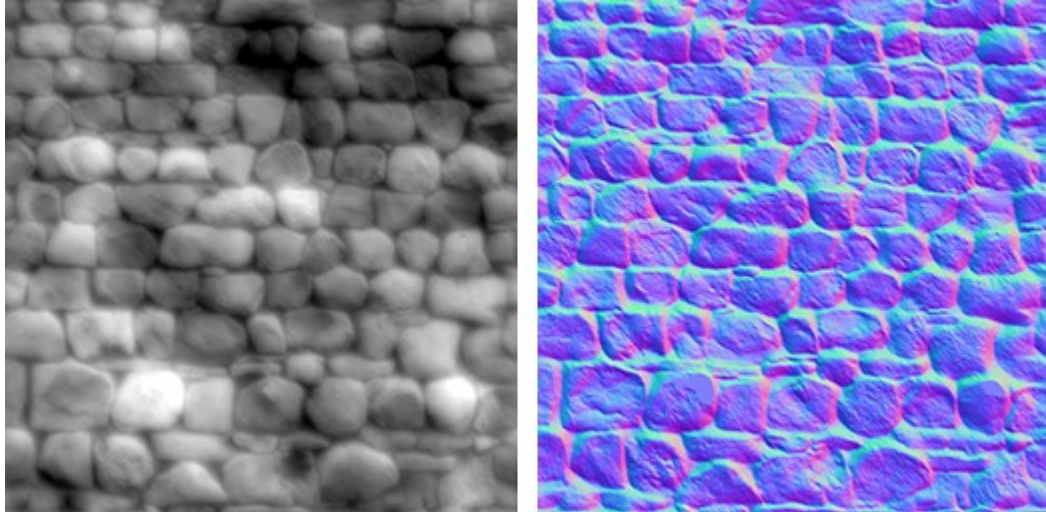
Kuva 6. Kuution saumat ja unwrapattu UV-saari.

3.5 Beikkaaminen

Raskaat 3D-veistetyt mallit käyttävät liian paljon laitteiston prosessointitehoa, joten niiden käyttö videopeleissä ei ole mahdollista. On kuitenkin tapoja siirtää raskasmallien pintojen dataa kevytmalleihin tekstuurikartoilla. Tekstuurikarttoihin voidaan laskea normaali-, kohouma- ja Ambient Occlusion -kartat.

Normaali- ja kohoumakartat vaikuttavat siihen, miten valo osuu 3D-mallin pintaan luoden illuusion syvyydestä, vaikka mallin pinta olisikin täysin sileä. Normaalikartta saa kevytmallin näyttämään yksityiskohtaisemmalta ilman suurta määrää topologiaa. Ero normaali- ja kohoumakarttojen välillä ei ole suuri, koska niillä on periaatteessa sama käyttötarkoitus. Kuvassa 7 esitelty kohoumakartta luo illuusion syvyydestä harmaansävyillä, joista musta kuvastaa syvyyttä, valkoinen kohoumaa ja harmaa muuttumatonta korkeutta. Normaalikartta sisältää enemmän dataa ja on

näistä kartoista uudempi, minkä vuoksi se on suuremmissa käytössä kohoumakarttaan verrattuna. Toisin kuin kohoumakartta, kuvassa 7 oleva normaalikartta käyttää vihreää, punaista ja sinistä väriä luomaan illuusion syvyydestä. [12.]



Kuva 7. Kiviseinän kohouma- ja normaalikartta. (Unity Documentation.)

Ambient Occlusion - tai AO-kartta on harmaansävyinen tekstuurikartta, joka sisältää valaisutietoa. AO-karttaa ei yleensä käytetä omana karttanaan, vaan se sisällytetään diffuusi-, spekulari- tai metallisuuskarttoihin antaen 3D-mallille realistisemmän valaistuksen korostamalla sen varjoja. Samoin kuin normaalikartat, AO-kartan voi beikata raskasmallista, mutta monet pelimoottorit pystyvät renderöimään sen reaaliajassa, jonka voi nähdä kuvasta 8. Tällöin beikkaamista ei tarvita ollenkaan. AO:n reaaliaikainen renderöinti kuitenkin vaatii koneen GPU:lta enemmän prosessointitehoa, varsinkin jos 3D-ympäristössä on paljon objekteja, minkä takia sitä ei käytetä tai se korvataan jollakin muulla post-processing -menetelmällä. [2;14.]



Kuva 8. Peliympäristö, jossa on pelimoottorin laskemat AO-kartat pois ja päällä. (Hayden, L. 2021.)

3.6 Teksturointi

Samoin kuin normaali- ja AO-kartat, 3D-mallin muutkin tekstuurikartat ovat kaksiulotteisia, mutta niitä ei tyypillisesti beikata raskasmallista. Mallin teksturointitapa riippuu taiteilijasta ja peliyhtiön käytänteistä, mutta yleisimmät tavat ovat käsin maalaaminen ja kuvamanipulointi käyttäen digitaalista kuvanmuokkausohjelmaa sekä proseduraalinen generointi noodipohjaisessa materiaalieditorissa. [12.]

Yleisimmät tekstuurikartat, joita pelien aseteissa käytetään, ovat diffuusi-, karkeus-, metallisuus ja läpinäkyvyyskartat, joista kolme ensimmäistä on esitelty kuvassa 9. Diffuusikartta on tekstuurikartoista yleisin ja sen avulla määritetään mallin pinnan väri ja kuviointi. Joissain tapauksissa diffuusikartta on mallin ainut kartta, jolloin siihen lisätään valaisu- ja varjotietoa esimerkiksi AO-kartan avulla. Nykypäivänä diffuusikartan kanssa käytetään muitakin karttoja, jolloin siihen ei tarvitse sisällyttää muuta kuin väritietoa. [15.]

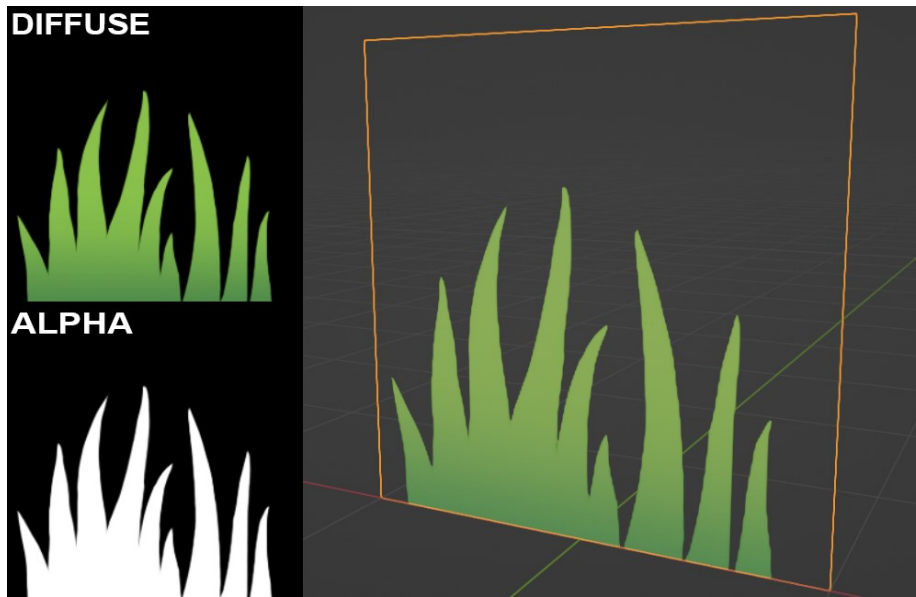
Karkeuskartta on harmaasävyinen tekstuurikartta, jolla voidaan muuttaa pinnan kiiltoisuutta. Mitä tummempi kartta on, sitä kiiltävämpi on mallin pinta. Metallisuuskartta on karkeuskartan tavoin harmaasävyinen ja sen avulla mallin pinnan heijastavuutta voidaan muuttaa. Kartassa valkoinen lisää heijastavuutta ja musta poistaa sitä. Karkeus- ja metallisuuskarttoja käytetään

usein yhdessä, sillä ne vaikuttavat toisiinsa. Esimerkkinä näiden kahden kartan käytöstä yhdessä voisi käyttää metallista ämpäriä, joka on ruostunut. Metallisuuskartassa ruostuneet osat maalaan mustiksi poistaen niiden heijastavuutta, ja karkeuskartalla ämpärin pinnalle voidaan lisätä epätäydellisyyksiä, kuten sormenjälkiä maalaamalla ne kartalle tummalla harmaansävyllä poistaen samalla niiden kiiltoisuutta ja metallisuutta. [2;16.]



Kuva 9. Eri tekstuurikarttojen vaikutukset mallin pinnassa.

Läpinäkyvyyskartalla voidaan määrittää, mitkä mallin osat ovat joko osittain tai kokonaan läpinäkyviä. Läpinäkyvyyskarttaa käytetään esimerkiksi ruohon, puun lehtien ja hiusten teksturoinnissa ilman monimutkaisten yksityiskohtien mallintamista käyttämällä yksinkertaista litteää polygonitasoa, josta on esimerkki kuvassa 10. Läpinäkyvyystietoa voi tallentaa muihinkin tekstuuri-karttoihin, kuten diffuusi- ja normaalikarttoihin, mutta sitä voi käyttää erillisenä harmaansävyisenä tai mustavalkoisena maskikarttana, jossa valkoinen väri on täysin kiinteä ja musta täysin läpinäkyvä. [2;17.]

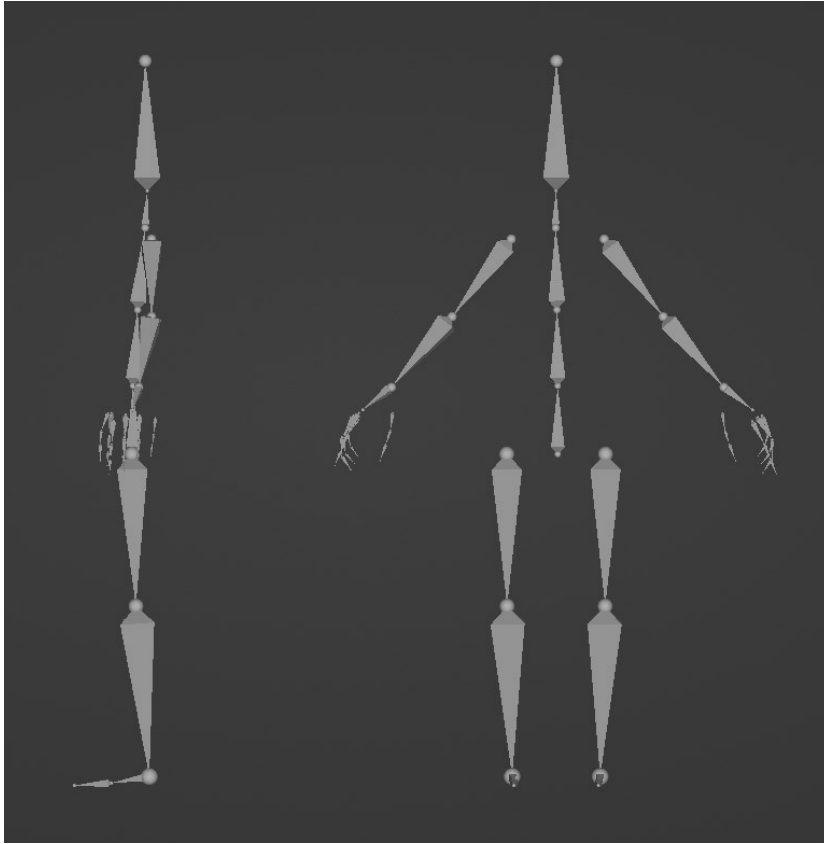


Kuva 10. Ruohonkorsien diffuusi- ja läpinäkyvyyskartat ja kuinka ne näkyvät tason pinnalla.

3.7 Riggaus

Ennen kuin valmista 3D-mallia voi alkaa animoimaan, se täytyy rigata ensin. Riggaamisella tarkoitetaan rigin eli nivelistä, luista ja muista kontrollereista koostuvan digitaalisen luurangon tekoa. Rigit voivat olla yhtä yksinkertaisia tai monimutkaisia riippuen itse mallista ja sen käyttötarkoituksesta. Kuvan 11 yksinkertainen rigi voi valmistua alle tunnissa, mutta monimutkaisemmat voivat viedä päiviä tai jopa viikkoja. [18.]

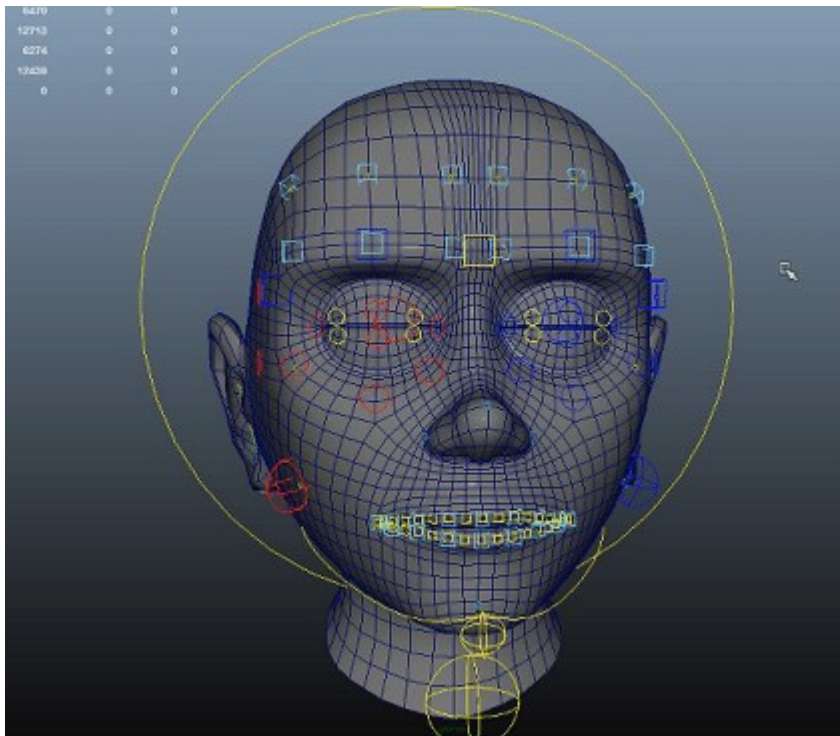
3D-mallinnusohjelmat usein sisältävät joko yhden tai useamman valmiin rigin, joita voi käyttää pohjana jatkokehitystä varten, mutta riggaaja myös voi luoda rigin itse luu kerrallaan. Rigin tekeminen aloitetaan luomalla root joint eli juurinivel. Rigin luut ohjaavat toisiaan hierarkkisessa järjestyksessä, mikä tarkoittaa, että ylintä luuta eli parent-luuta liikuttamalla kaikki sen alapuolella olevat luut eli child-luut liikkuvat sen mukana. Tämän kaltaista rigiä kutsutaan nimellä Forward Kinematics tai lyhennettynä FK. Juurinivel on hierarkiassa ylin luu, joten kaikki muut rigin luut ovat yhteydessä siihen joko suoraan tai toisten luitten kautta. Rigiä luodessa luut on hyvä asettaa samoin, miten ne olisivat oikeassakin luurangossa ja useimmiten juurinivelenä toimii hahmon lantio. [18.]



Kuva 11. Kahdella jalalla kulkevan hahmon rigi.

Inverse Kinematics tai lyhennettynä IK toimii päinvastoin kuin FK eli child luuta liikuttamalla tietyt parent-luut liikkuvat sen mukana. Yleisesti IK:ta käytetään hahmojen raajoissa, jolloin esimerkiksi nilkan luuta liikuttamalla koko jalka liikkuu sen mukana. IK useimmiten vaatii rajoitteita, jotta se toimii toivotulla tavalla. Aiemmin mainitussa esimerkissä nilkan luuta liikuttamalla liian korkealle voi aiheuttaa polven nivelen taipumisen väärään suuntaan. Rajoitteita lisäämällä voidaan esimerkiksi estää luuta liikkumasta tietyn rajan yli tai pyörimästä akselinsa ympäri. [18.]

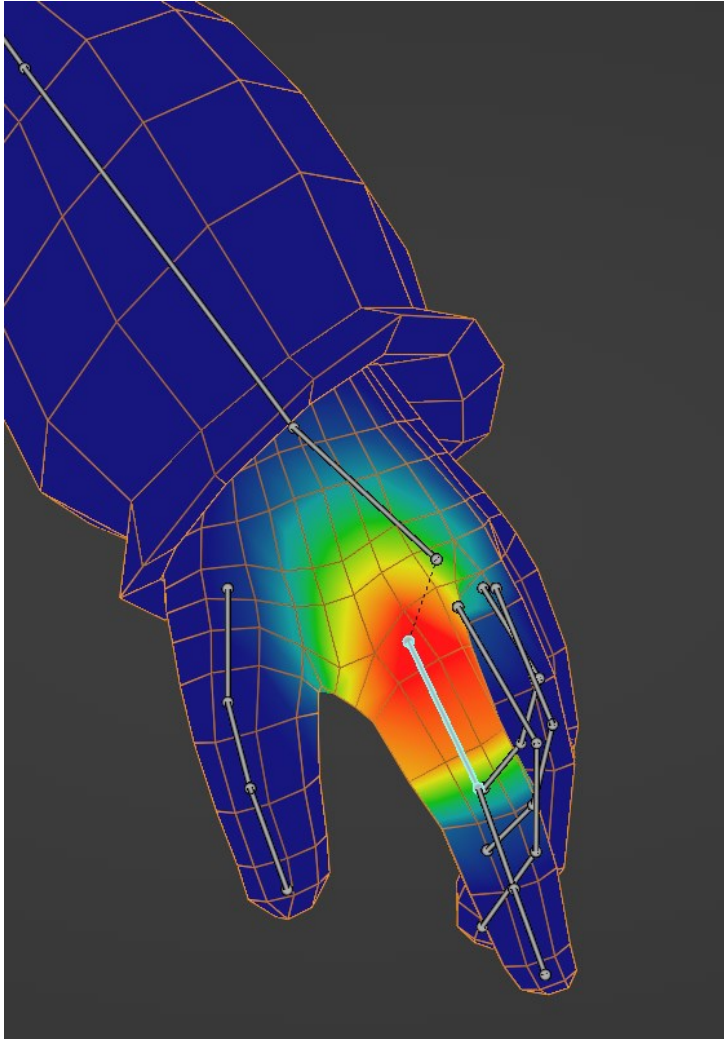
Hahmon kasvot usein vaativat erillisen rigin, josta on esimerkki kuvassa 12, sillä normaalit luut eivät aina sovellu monimutkaisten ilmeitten animoimiseen. Näissä tapauksissa mallille lisätään deformereita eli muodonmuuttajia. Muodonmuuttajia on useita erilaisia, joilla on omat hyötynsä ja haittansa, minkä takia hahmon kasvorigit usein koostuvat monista muodonmuuttajista. Yleisin muodonmuuttaja on Blend Shape, joka saa nimensä siitä, että sitä voidaan sekoittaa päälle ja pois kertoimen arvoilla 0 ja 1 välillä. Esimerkiksi silmäluomia voi kontrolloida Blend Shape -muodonmuuttajan avulla, jossa 0 vastaa neutraalia asentoa eli avonaisia silmiä, 0,5 vastaa puoliksi auki-naisia silmiä ja 1 vastaa kiinni olevia silmiä. [18;19.]



Kuva 12. Hahmon kasvorigi ja sen kontrollerit. (Riggingdojo.)

Kaikki muodonmuuttajat eivät kuitenkaan sovellu peliteollisuuteen, sillä monet pelimoottorit eivät tue niitä. Peliteollisuutta varten tehdyt rigit ovat myös yksinkertaisempia, koska monimutkaiset verteksen muutokset voivat hidastaa peliä. Esimerkiksi Muscle-muodonmuuttaja, joka nimensä omaisesti simuloi hahmon lihaksia on liian raskas peliteollisuutta varten, mutta sitä käytetään paljon elokuvateollisuudessa, varsinkin hyperrealistisia hahmoja animoidessa. [19;20.]

Jotta rigiä voi käyttää mallin animoimiseen, se täytyy ensin yhdistää malliin kiinni. Tämä työvaihe on nimeltään skinning. Nimestään huolimatta, skinning ei suinkaan tarkoita mallin repimistä auki, vaan painoarvojen antamista mallin vertekseille. Blender ja monet muutkin 3D-mallinnusohjelmat pystyvät luomaan painoarvot automaattisesti. Tietokone laskee luun painoarvot sitä lähellä oleville vertekseille, mutta ne ovat harvoin optimaalisia, minkä vuoksi painoarvoja täytyy muokata manuaalisesti. Blenderissä painoarvoja voi muokata Weight Paint -tilan työkaluja käyttäen. Painoarvoa voi säätää arvojen 0 ja 1 välillä maalamalla verteksejä, jossa 0 näkyy tummansinisena ja 1 näkyy punaisena. Kuvassa 13 esitellään, kuinka näiden kahden arvon väliset painoarvot näkyvät mallin pinnalla sinisen ja punaisen välisenä gradienttina. [21.]



Kuva 13. Hahmon käden skinnaaminen.

Weight painntaaminen on hyvä aloittaa maalaamalla jokaisen luun vaikuttamat verteksit punaisella eli antamalla niille maksimaalisen määrän painoarvoa. Luiden välisillä liitoskohdilla tulisi olla laajempi arvoalue, muutoin näiden alueiden polygonit voivat animoidessa muuttaa muotoaan ei-toivotulla tavalla. Painoarvojen vaikutusta voi testata muuttamalla rigin asentoa Pose-tilassa. Tällä tavoin on myös helpompi nähdä, jos jokin luu kontrolloi mallin verteksejä, joita sen ei pitäisi. [21.]

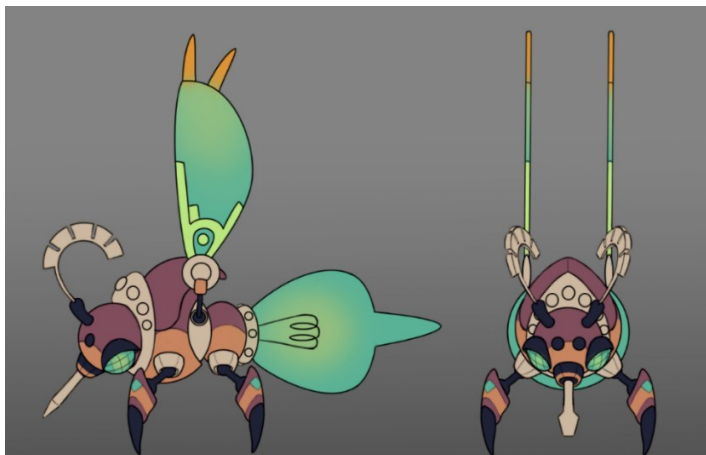
4 PROJEKTITYÖ

4.1 Suunnittelu

Projektityö aloitettiin käymällä toimeksiantajan kanssa läpi, minkälaisen vihollishahmon he halusivat. Vihollishahmon oli tarkoitus näyttää kullankaivuurobotilta, joka sopisi temaattisesti heidän peliprojektinsa steampunk-tyyliseen kenttään. Robotin tuli olla myös pienikokoinen ja sen päätarkoituksena oli varastaa pelaajahahmolta esineitä, kuten terveysjuomia, minkä jälkeen robotti yrittäisi sitten paeta pelaajaa. Toimeksiantajalla oli jo entuudestaan konseptikuvia, joista he halusivat variaatioita, mutta he antoivat vapaat kädet luoda myös muunlaisia konsepteja. Toimeksiantajan suosikkikonsepti oli ötökkämäinen, joten se teema esiintyi hyvin selvästi uusissakin konsepteissa.

Uusia konseptikuvia tehdessä kerättiin paljon referenssimateriaalia eri internetsivustoilta, jotta hahmosuunnittelu helpottuisi. Suurin osa referensseistä oli muiden pelien konseptikuvia, mutta joissakin konsepteissa apuna käytettiin käsin tehtyjä koruötököitä.

Valitusta konseptista tehtiin mallintamisen helpottamiseksi model sheet, jossa robotti oli kuvattu sekä sivulta että edestä. Kuvassa 14 oleva robotin model sheet ei kuitenkaan ollut aivan täydellinen, sillä eri kuvakulmat eivät sopineet yhteen eikä aikataulun vuoksi näitä ongelmia keritty korjaamaan. Model sheetista oli silti erittäin paljon hyötyä, sillä se toimi referenssikuvana, jonka avulla hahmon eri ruumiinosat saatiin mallinnettua oikeankokoisiksi ja -muotoisiksi.



Kuva 14. Robottihahmon model sheet, jossa hahmo on kuvattuna sivulta ja edestä.

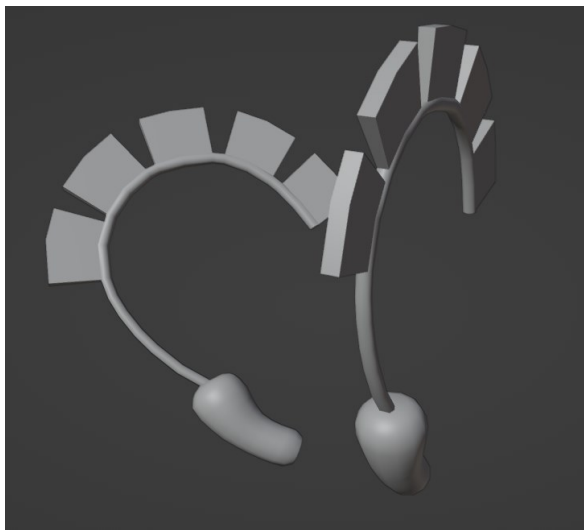
4.2 3D-mallintaminen

Vihollishahmon mallintamiseen käytettiin Blenderiä. Mallinnusohjelma valittiin, sillä sitä oli käytetty paljon jo entuudestaan ja se oli ilmainen, joten sen käyttöön ei tarvittu maksullista lisenssiä. Blender on avoimen lähdekoodin 3D-mallinnusohjelma, joten siihen voi halutessaan ladata lisäosia. Tässä projektityössä on kuitenkin käytetty Blenderin oletusversiota.

Hahmon mallintaminen aloitettiin luomalla ensin yksinkertaiset kevytmallit sen eri ruumiinosista, jotta niiden koot saataisiin oikeanlaisiksi jo heti projektin alussa. Kevytmalleja on myös helpompi muokata pienemmän polygonimäärän takia, joten suurtenkin muutosten tekeminen ei olisi vaivalloista. Kevytmallin pohjalta luotiin myös raskasmalli, josta beikattiin tekstuurikarttoja.

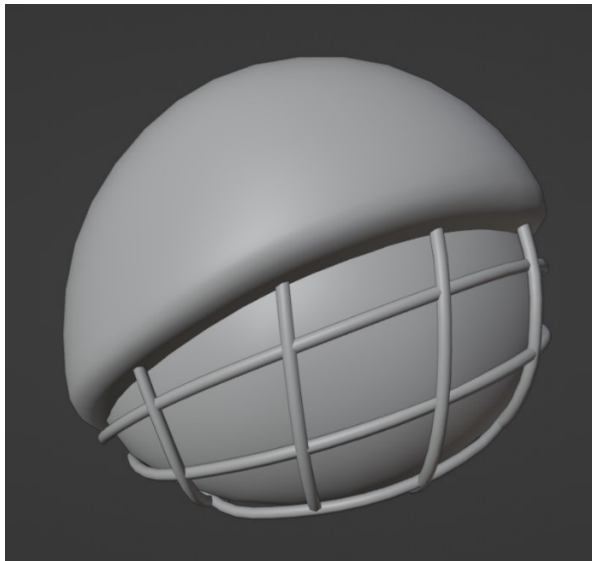
4.2.1 Pää

Vihollishahmon pään mallintaminen aloitettiin mallintamalla pallo, johon kaikki muut pään osat kiinnitettäisiin. Eturuumiin valmistuttua pää upotettaisiin osittain sen sisälle, jossa pää pystyisi rotaatioitumaan akselinsa ympäri vapaasti. Kuvassa 15 on robotin tuntosarvet, jotka tehtiin käyttäen Curve-työkalua, jotta se saataisiin kaarevaan muotoonsa helposti. Hammasratasta muistuttavat osiot tehtiin valitsemalla polygoneja mallin pinnalta ja työntämällä niitä ulospäin ekstruusion avulla.



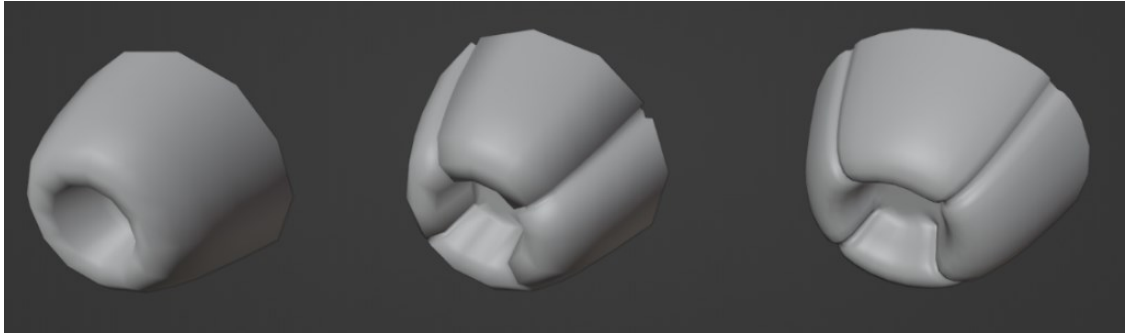
Kuva 15. Robotin etusarvien kevytmallit.

Silmät luotiin palloista, joiden muotoa muokattiin Sculpt-tilan työkaluilla, kuten Grab- ja Smooth-työkaluilla. Koska Blenderin oletuspallot sisältävät kolmioita oli silmien topologiaa muutettava siten, että kolmiot poistettiin ja korvattiin ne nelikulmioilla. Muutos tehtiin, sillä nelikulmionmuotoisia polygoneja on helpompi muokata, ja ne tekevät mallin pinnasta sileämmän. Silmäluomet tehtiin kopioimalla silmät ja suurentamalla niitä siten, että ne peittävät ne. Sen jälkeen silmäkulmista poistettiin alempi puolisko ja jäljelle jäänyt reikä korjattiin käyttämällä Grid Fill -työkalua, joka luo uutta nelikulmiomuotoista topologiaa. Grid Fill -työkalu toimii vain silloin, kun mallissa olevan reiän reunoilla on tasalukuinen määrä verteksejä. Vaihtoehtoisena työkaluna voi käyttää Fill-työkalua, mutta nelikulmioiden sijasta se tuottaa kolmioita. Reiän voi peittää myös n-gonilla, jonka voi luoda käyttäen New Face from Edges -työkalua, jolloin mallintaja voi tehdä reunasilmu-
kat manuaalisesti käyttämällä Knife-työkalua. Viimeiseksi silmien pinnalle lisättiin ritilämäiset yksityiskohdat käyttäen Curve-työkalua, jotka voi nähdä kuvassa 16.



Kuva 16. Robotin silmän raskasmallin eri osat.

Kuvassa 17 on esitelty hahmon niin sanottu nenä, jonka mallintaminen aloitettiin sylinteristä. Blenderin luomien sylinterien päädyissä on n-gonit, joten ne poistettiin ja toiseen puoliskoon tehtiin uutta topologiaa sekä luotiin kolo ruuvimeisselille. Nenälle lisättiin myös sen muotoa pitkin kulkevat sisennykset.



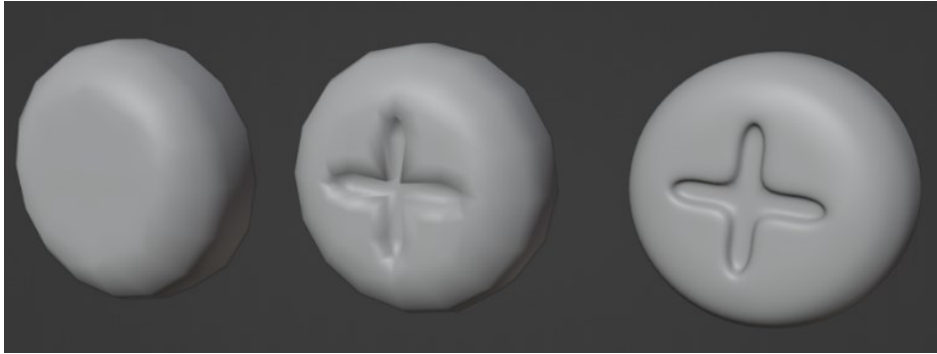
Kuva 17. Nenän mallintamisen vaiheet, jossa on kevytmalli sekä raskasmalli ennen ja jälkeen Subdivion Surface -modifikaattorin lisäämistä.

Kuvassa 18 on robotin nenään yhdistetty sylinteristä tehty ruuvimeisseli, jonka runkoon lisättiin reunasilmukka, jotta verteksien leventäminen ei vaikuttaisi muuhun runkoon. Teräosaa kavennettiin ja meisselille lisättiin ylimääräisiä reunasilmukoita, joiden avulla se piti muotonsa, kun sille lisättiin Subdivion Surface -modifikaattori.



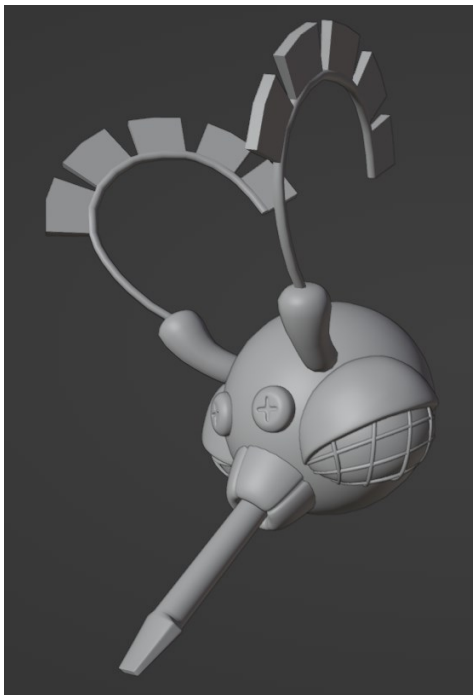
Kuva 18. Valmis raskasmalli ruuvimeisselistä, joka on upotettu robotin nenään.

Kuvassa 19 on sylinteristä tehty pään ruuvi, jonka reunoja pyöristettiin Bevel-työkalulla ja litistettiin Scale-työkalulla saaden oikean muotonsa. Ristin muotoinen reikä lisättiin ruuviin muokkamalla sen topologiaa ja käyttämällä ekstruutiota. Lopuksi ruuville lisättiin Subdivion Surface - ja Mirror-modifikaattorit, jotta se saatiin sileämmän näköiseksi ja kopioitua helposti pään toiselle puolelle.



Kuva 19. Ruuvin mallintamisen vaiheet, jossa on kevyt malli sekä raskas malli enne ja jälkeen Subdivision Surface -modifikaattorin lisäämistä.

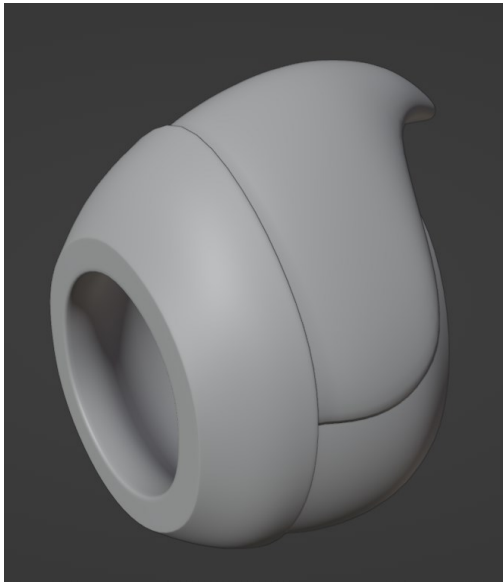
Toimeksiantaja oli muutoin tyytyväinen kuvassa 20 olevan pään muotoon, mutta tuntosarvia tarvitsi hieman muokata. Ne olivat liian lähellä eturuumista, jolloin ne pään liikuessa saattaisivat osua tai työntyä eturuumiin sisälle. Korjaus oli kuitenkin yksinkertainen, sillä sarvia tarvitsi vain liikuttaa eteenpäin.



Kuva 20. Viimeistelty raskas malli robotin päästä.

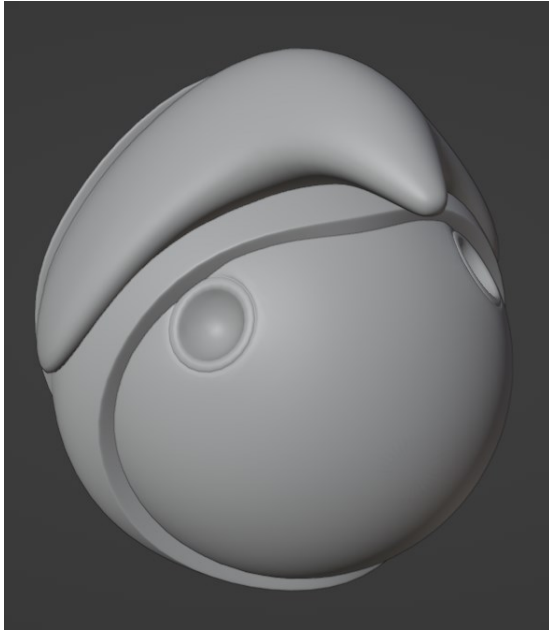
4.2.2 Ruumis

Kuvassa 21 kuvattu robotin ruumis koostui kolmesta osasta: eturuumiista, johon pää upotettiin, takamuksesta, johon hehkulamppu kiinnitettäisiin, ja pallonivelestä, joka toimisi kahden aiemman osan kiinnityskohtana. Eturuumis luotiin pallosta, joka kaiverrettiin ontoksi, jotta sen sisälle voitaisiin upottaa sekä pää että pallonivel. Eturuumiin päälle lisättiin myös takasuojus, joka mallinnettiin kopioimalla eturuumiin topologiaa ja käyttäen Solidify-modifikaattoria, kun suojus oli saatu oikeanmuotoiseksi.



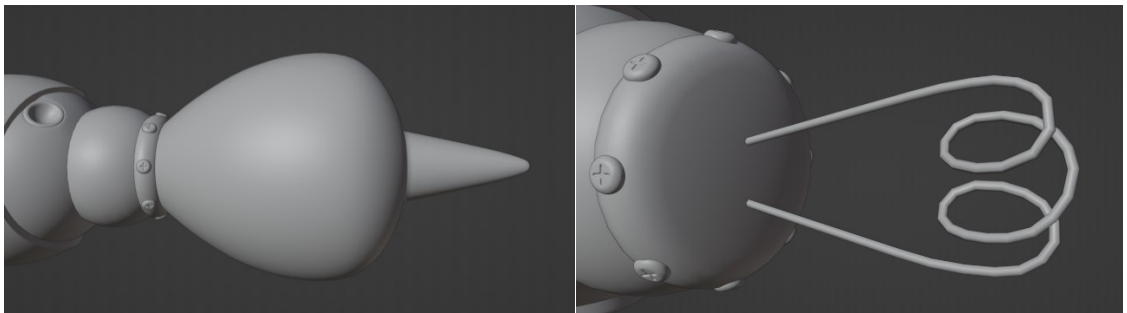
Kuva 21. Eturuumiin raskasmallin pohjaversio.

Kuvassa 22 kuvattu pallonivel tehtiin pallosta, joka asetettiin osittain eturuumiin sisälle. Pallonive-
len tarkoituksena oli toimia sekä ruumiinosien liitoskohtana, mutta myös mahdollistaa osien ro-
taation nivelen akselin ympäri. Palloniveleen tehtiin myös kolot, joihin siivet voitaisiin yhdistää.



Kuva 22. Ruumiin pallonivelen raskasmalli eturuumiin sisällä.

Robotin takamus tehtiin pallosta, josta poistettiin vertikaalisesti sen toinen puolisko. Takamuksen muotoa litistettiin käyttäen Scale-työkalua, jotta se näyttäisi soikiomaisemmalta. Kuvassa 23 kuvattu hehkulamppu tehtiin käyttäen sylinteriä, joka sai muotonsa lisäämällä reunasilmukoita käyttäen Loop Cut -työkalua ja muokkaamalla niiden kokoa Scale-työkalulla. Takamuksen ja hehkulampun väliin lisättiin pienempi ja litistetympi kopio takamuksesta, joka toimi lampun kantana ja johon hehkulanka kiinnitettiin.

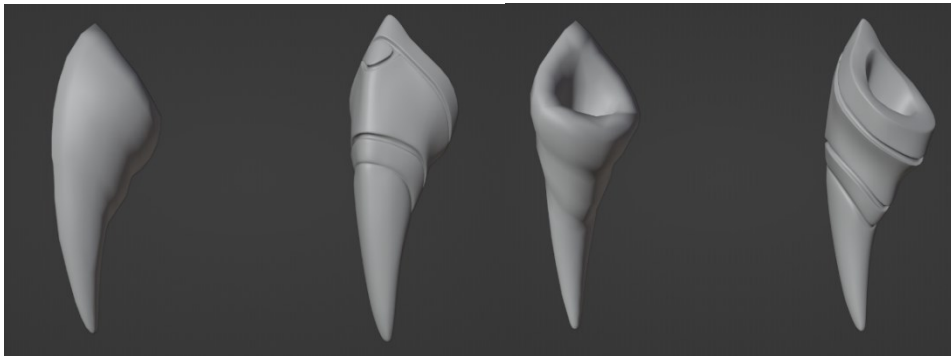


Kuva 23. Robotin takamuksen eri osat sekä hehkulampun sisällä oleva hehkulanka.

4.2.3 Jalat

Robotin olisi pystyttävä liikkumaan sekä ilmassa että maalla, joten se tarvitsisi jalat. Jalat saatiin robotin ruumiinosiin kiinni käyttäen nenän kopioita, sillä ne muotonsa ansiosta toimivat hyvinä liitoskohtina jalkojen palloniveleä varten. Eturaajat kiinnitettiin eturuumiiseen ja takajalat takamukseen.

Kuvassa 24 on kuvattu jalkojen päihin luodut yksittäiset kynnet, joiden päällä robotti pystyisi seisomaan. Pallonivelelle lisättiin liitoskohta kaivamalla kolo sitä varten. Kynnen kuviointi saatiin tehtyä lisäämällä reunasilmukoita ja käyttämällä ekstruutiota. Jotkin reunasilmukat saivat aikaan kulmikkaan muodon, joka saatiin korjattua käyttäen Smooth-työkalua, jonka avulla voitiin tasoittaa liian lähellä toisiaan olevia verteksejä.



Kuva 24. Jalkojen kynsien kevytmalli ja raskasmallin pohjamalli kuvattuna edestä ja takaa.

Ruumiin ja kynnen liitoskohtien välille luotiin pallonivel, joka on esitelty kuvassa 25. Nivel tehtiin kahdesta identtisestä pallosta, jotka liitettiin toisiinsa käyttäen Bridge-työkalua. Työkalua käyttäen pallojen väliin saatiin sylinterin muotoinen osio, joka antoi nivelelle tarvittavan määrän pituutta.

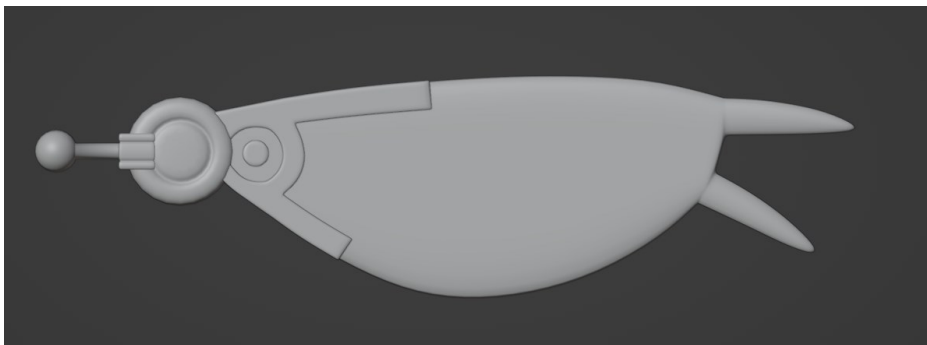


Kuva 25. Jalan pallonivel liitettynä kynteen.

4.2.4 Siivet

Kuvassa 26 on robotin siipi, joka mallinnettiin käyttäen tasoa, johon lisättiin lisää topologiaa käyttäen ekstruutiota sen reunoissa. Siivelle lisättiin myös Solidify-modifikaattori, jonka avulla siitä saatiin kolmiulotteinen. Siiven kuviointi toteutettiin samalla tavoin muodostamalla sen muoto ensin käyttäen tasoa ja lisäämällä siihen Solidify-modifikaattori.

Siipi kiinnitettiin robotin ruumiiseen käyttäen kopiota jalkojen pallonivelestä, josta leikattiin ylempi pallo pois. Pallonivelen ja siiven väliin lisättiin kiekkomainen osio, joka luotiin käyttäen sylinteriä. Lopuksi kiekon ja nivelen liitoskohtaan lisättiin nelikulmion muotoinen osa.

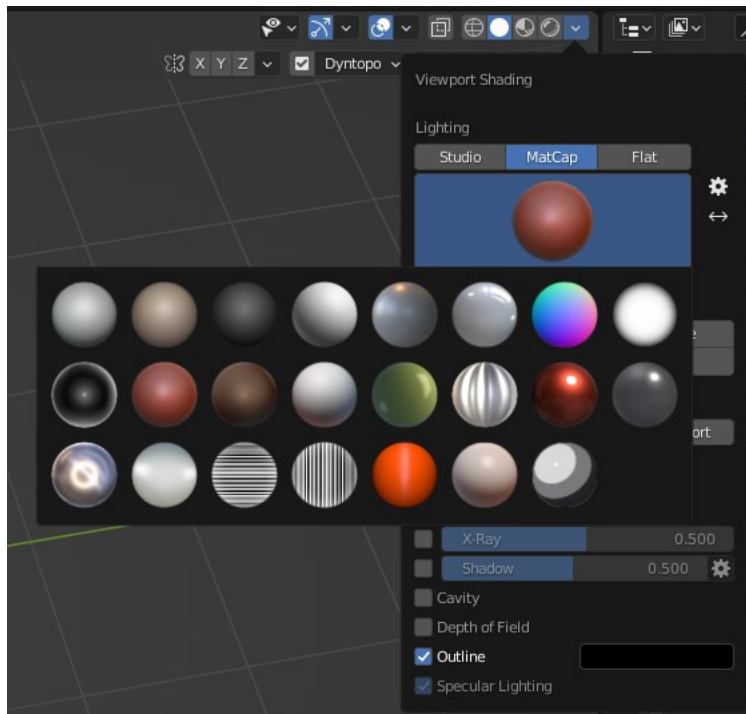


Kuva 26. Siiven eri osat kuvattuna päältäpäin.

4.2.5 Yksityiskohtien veistäminen

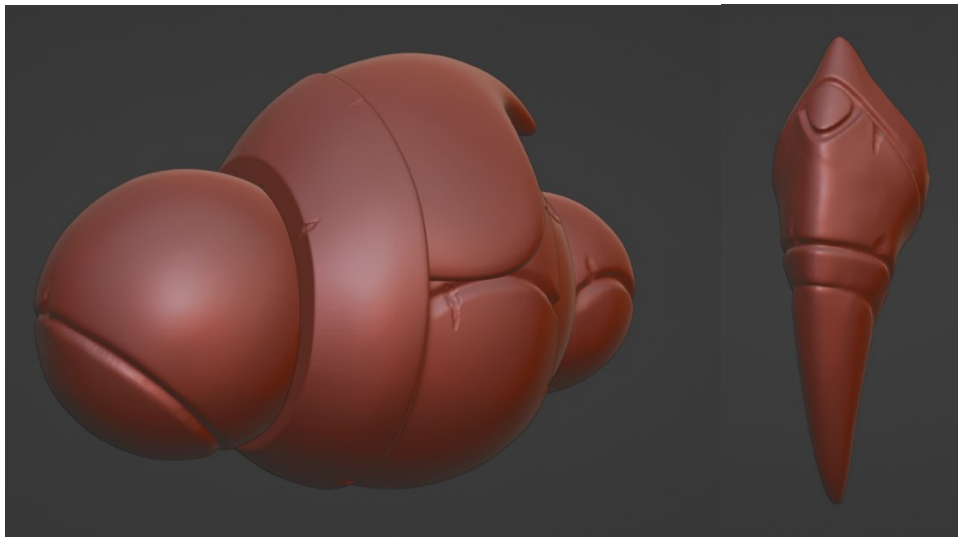
Mallin eri osien pohjamallit saatiin valmiiksi, mutta niitten pintaan oli vielä lisättävä yksityiskohtia veistämällä. Veistäminen aloitettiin nostamalla mallien osien polygonimäärää käyttämällä Subdivision Surface -modifikaattoria, sillä veistäminen vaatii korkean polygonimäärän. Sculpt-ikkunasta löytyy myös kaksi muuta työkalua, joiden avulla voi vaikuttaa mallin topologiaan. Dyntopo eli dynaaminen topologia on työkalu, jonka avulla voi lisätä tai poistaa topologiaa maalaamalla mallin pintaa käyttäen veistämissiveltimiä. Asetuksien kautta voi muokata miten paljon uutta topologiaa Dyntopo tuottaa ja vaihtaa määrän muuttumattomaksi, jolloin se miten kaukana tai lähellä malli on ei vaikuta uuden topologian lisäämiseen. Remesh vuorostaan rakentaa koko mallin geometrian uudelleen pyrkien pitämään topologian yhtenäisenä. Asetuksien kautta voi muuttaa Remeshin resoluutiota eli miten paljon uutta topologiaa lisätään tai vähennetään. Resoluution muuttamisen kanssa on oltava erittäin varovainen, sillä Remesh voi tuottaa miljoonittain uusia polygoneja, jos resoluution säätää liian suureksi. Pahimmassa tapauksessa koneen teho ei riitä laskemaan uutta topologiaa ja ohjelma sulkeutuu, josta syystä resoluutiota on hyvä nostaa pieni määrä kerrallaan. Remesh on kuitenkin kevyempi Dyntopoon verrattuna, joten sitä käytettiin tämän projektin mallin veistämisen yhteydessä.

Veistämisen ajaksi mallin Viewport-materiaali muutettiin kuvassa 27 olevien asetuksien kautta savenpunaiseksi. Valikosta voi valita monia erilaisia materiaaleja, joiden tarkoitus on helpottaa mallintajaa näkemään pinnan muodot helpommin eri väreissä ja valaistuksissa.



Kuva 27. Kiinteiden materiaalien näkymän valikko valittavista materiaaleista.

Toimeksiantajan pyynnöstä pinnan vaurioita ei saanut olla paljoa, mutta robotin tuli näyttää hie-
man kuluneelta. Kuvassa 28 näkyy, kuinka Clay Strips -sivellintä käyttämällä mallin pintaan veis-
tettiin naarmuja ja lommoja niihin alueisiin, jotka todennäköisimmin voisivat olla vaurioituneet.



Kuva 28. Ruumiin, pään ja kynnen raskasmallit, joihin on veistetty kulun.

4.3 Mallin optimointi

Vaikka robotin eri osat oli saatu valmiiksi, sitä ei vielä pystynyt teksturoimaan tai riggaamaan, sillä se oli aivan liian raskas käytettäväksi millään pelialustalla, koska raskasmallin polygonimäärä oli yli 1,3 miljoonaa. Malli piti ensin retopologoida eli siitä oli luotava kevytmalli. Kevytmallien teko on saatavilla ohjelmia, jotka on luotu juuri retopologiaa varten, kuten esimerkiksi 3DCoat ja CozyBlanket, mutta monien mallinnus- ja veistämishjelmien sisältä löytyy myös omat työkalut kevytmallien luontia varten. Robotin kevytmalli luotiin Blenderin sisällä käyttäen sen oletustyökaluja, mutta halutessaan Blenderiin voi asentaa lisäosia, kuten RetopoFlowin, joka antaa käyttöön hyödyllisiä työkaluja, joiden avulla retopologia helpottuu.

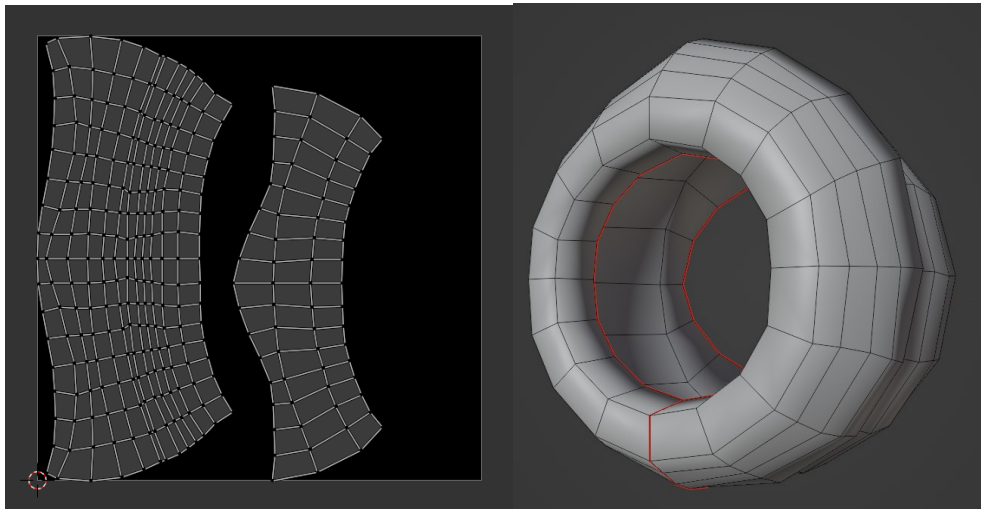
4.3.1 Retopologia

Koska robotin raskasmallien pohjina oli käytetty kevyempiä malleja, niitä pystyttiin hyödyntämään kevytmallia luodessa. Myös jotkin robotin osat kuten tuntosarvet ja raajojen pallonivelet eivät vaatineet retopologiaa, sillä ne olivat tarpeeksi kevyitä käytettäväksi. Veistettyjen osien kevyemmistä versioista oli myös jätetty kopioita, jotta niistä olisi helppo muokata kevytmalleja poistamalla ylimääräisiä reunasilmukoita. Lopullisen mallin polygonimäärä saatiin tiputettua yli 1,3 miljoonasta vähän yli 6,7 tuhanteen.

Retopologian apuna käytettiin myös Shrinkwrap-modifikaattoria, joka lisättiin tietyille kevytmalleille, kuten silmäluomille ja jalkojen kynsille. Shrinkwrap-modifikaattorin avulla kevytmalli saatiin kietoutumaan mallin valitun raskasmallin ympärille.

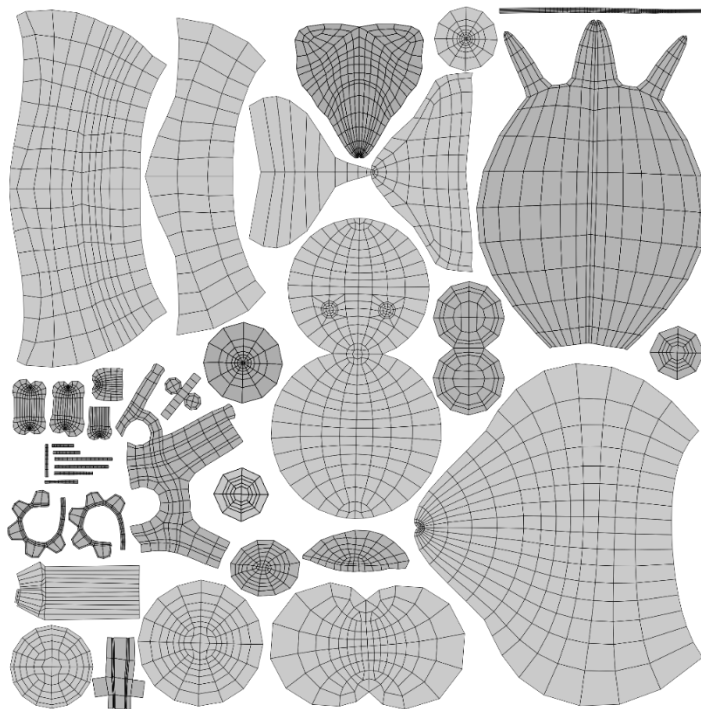
4.3.2 Unwrappaus

Unwrappausta tehdessä robotin eri osat pidettiin vielä erillisinä objekteina, kuten kuvasta 29 voi nähdä, sillä esimerkiksi tuntosarviin ja raajoihin lisättäisiin Mirror-modifikaattori, jolloin mallin unwrappaukset peilautuisivat myös sen kopiolle. Peilattujen osien UV-saaret olisivat myös alkuperäisten päällä, jonka avulla pystyttiin säästämään huomattavasti tilaa, sillä lopullisessa UV-kartassa kaikkien robotin osien UV-saarten tulisi mahtua yhteen.



Kuva 29. Eturuumiin UV-saumat ja levitetty UV-saari.

Osien unwrappausten valmistuttua, osat voitiin liittää yhdeksi malliksi, mikä helpottaisi UV-saarien järjestelyä. Blender osaa pakata UV-saaret yhdeksi UV-kartaksi automaattisesti, mutta tässä tapauksessa saaret aseteltiin manuaalisesti, kuten kuvasta 30 voi nähdä. Saaret pyrittiin pitämään suorassa joko horisontaalisesti tai vertikaalisesti, muutoin esimerkiksi normaalikarttoja beika-tessa olisi voinut esiintyä artefakteja.

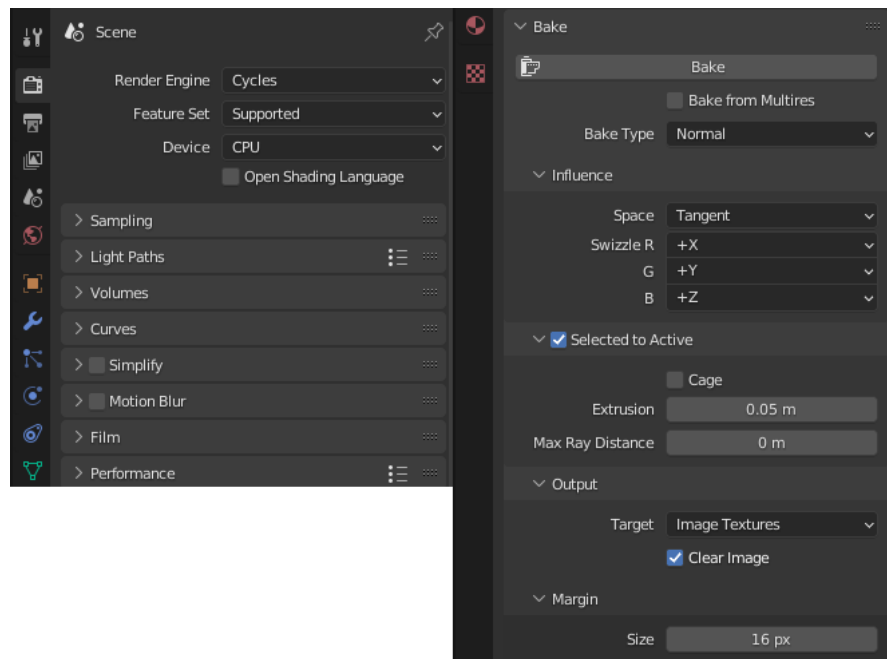


Kuva 30. Robotin osien UV-saaret järjesteltynä yhdeksi UV-kartaksi.

4.3.3 Beikkaus

Beikkauksen ajaksi robotin eri osat eroteltiin toisistaan erillisiksi objekteiksi, sillä kaikki näkyvillä olevat mallit vaikuttavat normaalikarttojen beikkaamiseen. Tällä tavoin esimerkiksi eturuumiin normaalit eivät vuoda pään normaalien päälle. Beikkaamisen ajaksi Blenderin renderöintimoottori vaihdettiin Cyclesiin, sillä se on ainut moottori, jonka avulla voi beikata tekstuurikarttoja.

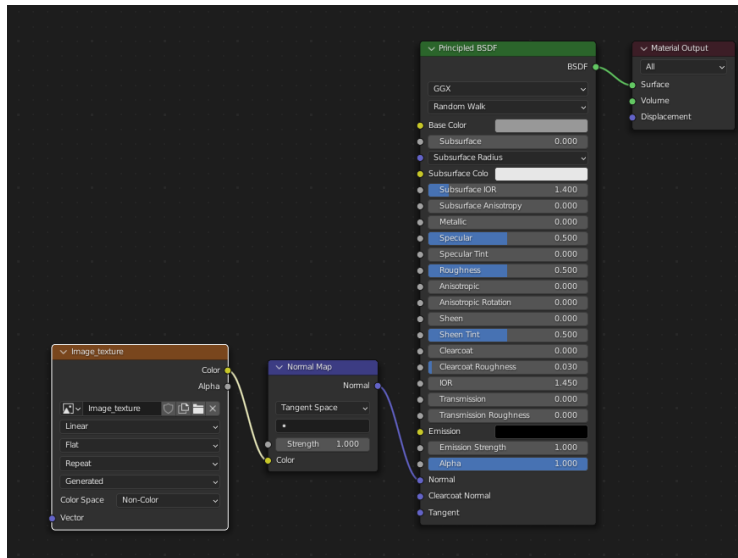
Kuvassa 31 näkyvän Render Properties -ikkunan sisältä löytyvän Bake-osion asetuksista muutettiin Bake Type normaalikartaksi. Pudotusvalikon kautta voi myös esimerkiksi valita yksittäisen tekstuurikartan kuten AO- tai diffuusikartan, mutta halutessaan useita tekstuurikarttoja voi beikata samanaikaisesti valitsemalla Combined. Asetuksista valittiin myös Selected to Active aktiiviseksi, mikä tarkoittaa, että ensimmäiseksi valitusta mallista beikataan tekstuurikarttoja viimeiseksi valitun mallin pinnalle. Valinnan alapuolella olevaa Extrusion-liukusäädintä nostettiin 0,05:een, minkä avulla pystyttiin vähentämään normaalikartoissa esiintyviä artefakteja.



Kuva 31. Beikkaukseen liittyvien asetusten valikot.

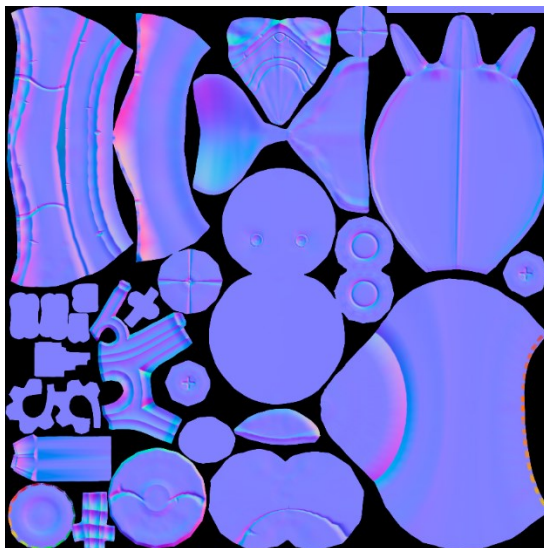
Kuvassa 32 on kuvattuna mallille luotu materiaali, jota ilman beikkaaminen ei olisi ollut mahdollista. Materiaalina käytettiin Blenderin oletusmateriaalia, johon lisättiin Image Texture -noodi, jonka Color Space muutettiin Non-Coloriksi, ja Normal Map -noodi, jonka avulla normaalikartta

piirtyisi oikein myös Viewportissa. Viimeiseksi Image Texture -noodi valittiin aktiiviseksi painamalla sitä, jolloin noodin ympärille ilmestyi valkoiset ääriiviivat. Valinnan avulla Blenderille kerrottiin, mihin noodiin tekstuuri kartta haluttiin tallentaa.



Kuva 32. Yksinkertainen materiaali normaalien beikkaamista varten.

Jokaisen beikkauksen jälkeen normaalikartta tallennettiin tekstuureja varten tehtyyn kansioon, sillä jokainen uusi beikkaus tallentuu edellisen päälle. Kuvassa 33 on kuvattuna kaikkien robotin osien beikatut normaalit yhdistettynä yhdeksi tekstuuriatlaksiksi, joka tehtiin käyttämällä Photoshopia.



Kuva 33. Robotin osien normaalikartat yhdistettynä yhdeksi normaalikartaksi.

4.3.4 Teksturointi

Teksturointiprosessi oli yksinkertainen, sillä robotti tarvitsi vain diffuusi-, emissiivisyys- ja läpinäkyvyyskartat, jotka ovat kuvattuna mallin pinnalla kuvassa 34. Metallisuus- tai karkeuskartat eivät olleet tarpeellisia, koska ne eivät vaikuttaisi mallin ulkonäköön paljoa peliprojektin sisällä, sillä se käyttää Cel shading -renderointitekniikkaa, joka saa peliympäristön mallit näyttämään sarakuvamaisilta.

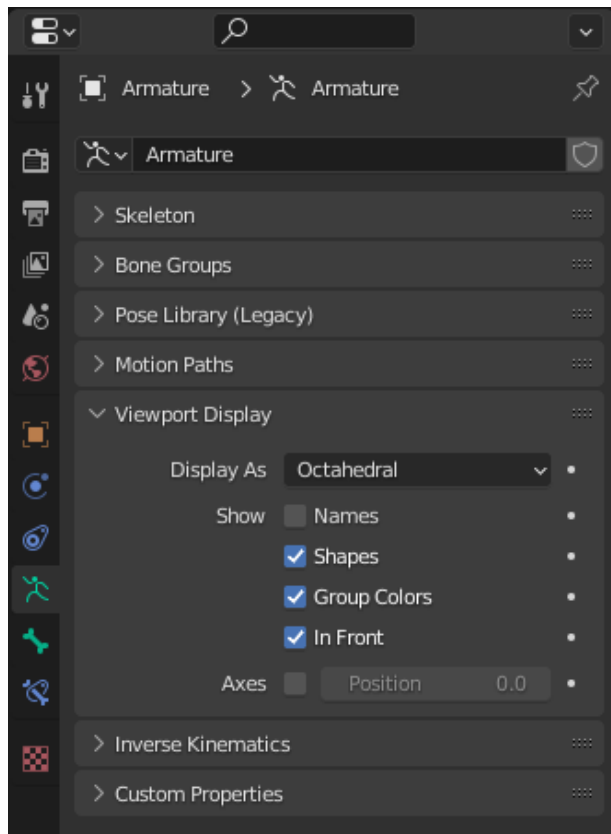
Diffuusikartta maalattiin Photoshopissa käyttämällä normaali- ja UV-karttaa apuna. Normaalikartan avulla pystyttiin esimerkiksi näkemään oranssin ja viininpunaisen värin kohtaamiskohdat ja UV-kartan avulla pystyttiin välttämään värien vuotaminen toisten UV-saarien päälle. Emissiivisyyskartta tehtiin diffuusikartan kopiosta, johon määriteltiin valaistuvat osat maalaamalla muut osat mustiksi. Emissiivisyyttä lisättiin esimerkiksi robotin silmille, siiville ja hehkulampulle. Läpinäkyvyyskartta tehtiin täysin valkoisesta tekstuurikartasta, johon lisättiin vaalean harmaata niihin kohtiin, jotka haluttiin osittain läpinäkyviksi, tässä tapauksessa vain robotin hehkulamppu tulisi olemaan läpinäkyvä.



Kuva 34. Testirendi robotista diffuusi-, emissiivisyys- ja normaalikartan kanssa.

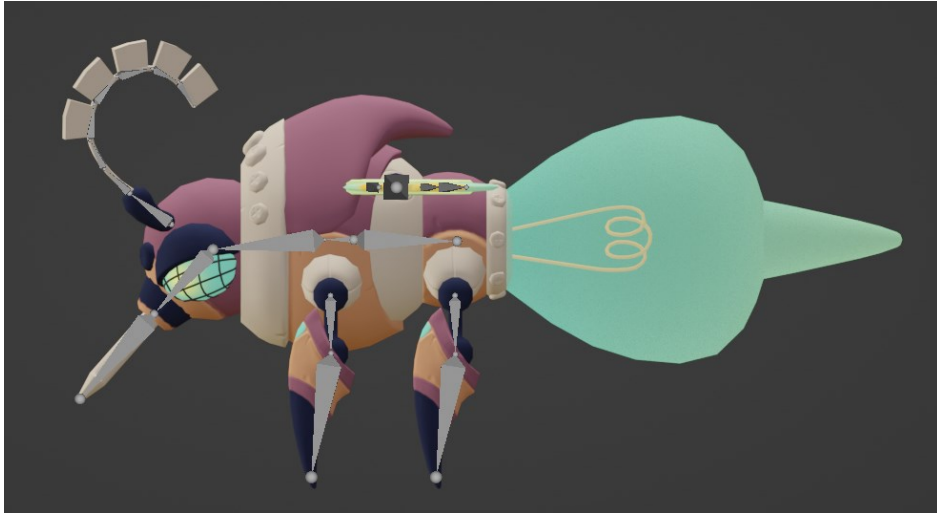
4.3.5 Riggaus

Hahmon riggaaminen aloitettiin lisäämällä juurinivel, joka tässä tapauksessa oli robotin ruumiin palloniveltä kontrolloiva luu. Kuvassa 35 olevasta Object Data Properties -valikosta muutettiin Viewport-asetuksia siten, että luut näkyisivät mallin edessä riippumatta siitä, mistä suunnasta mallia katsoisi. Luita lisättiin yksi kerrallaan kaikille robotin liikkuville osille, kuten eturuumiille, päälle, ruuvimeisselille ja takamukselle. Jalkojen, tuntosarvien ja siipien luista tehtiin vain vasemmanpuoleiset, sillä ne pystyttäisiin kopioimaan käyttämällä Symmetrize-toimintoa. Luut nimettiin niiden osien mukaisesti, joita ne kontrolloisivat ja kopioitavien luiden nimien perään lisättiin `_L`-merkintä, sillä Symmetrize-toiminto korvaisi ne `_R`-merkinnällä automaattisesti.



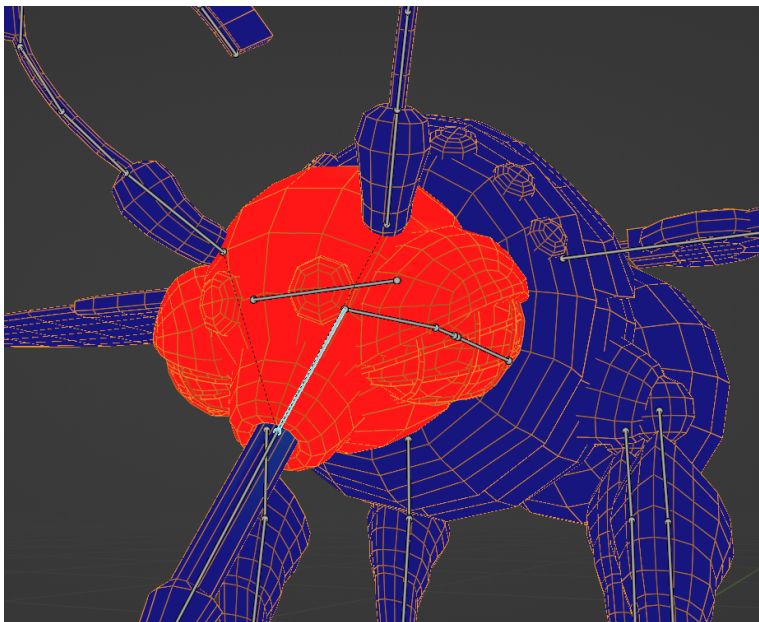
Kuva 35. Rigin näkymän asetuksien valikko.

Kuvassa 36 on valmis rigi, joka kiinnitettiin robottiin valitsemalla `With Empty Groups` -toiminto. `With Empty Groups` ei laske automaattisia painoarvoja mallille eikä se myöskään korvaa tai poista aikaisemmin tehtyjä painoarvoja, jos rigin joutuu ottamaan irti mallista.



Kuva 36. Valmis rigi kuvattuna sivulta.

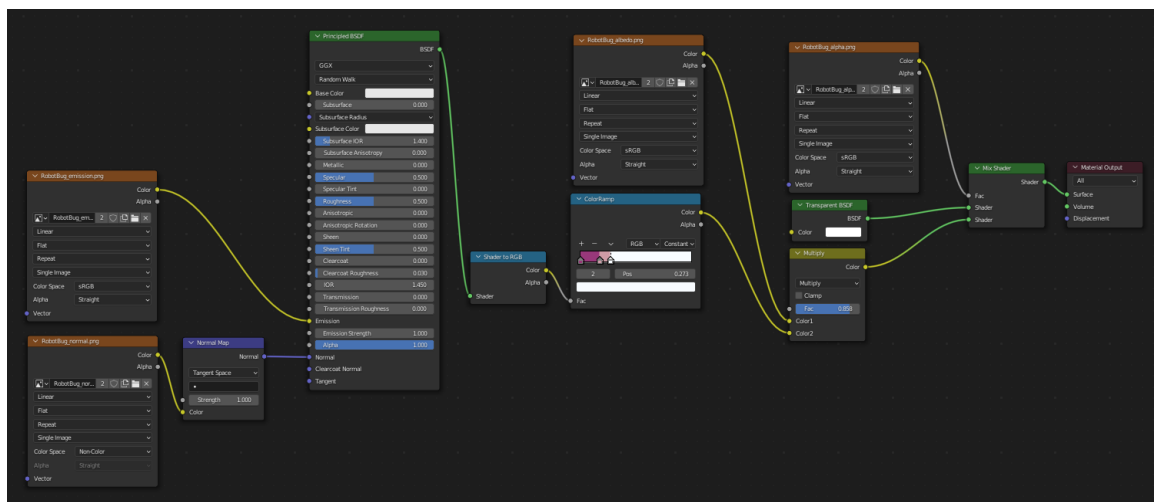
Viimeiseksi mallin eri osille maalattiin painoarvot Weight Paint -tilaa käyttämällä. Ensimmäisenä maalattiin tärkeimmät liikkuvat osat kuten pää, eturuumis, takamus, siivet ja jalat, joista pään painoarvot näkyvät kuvassa 37. Näiden osien luiden Pivot Pointien oli oltava erittäin tarkkoissa paikoissa, jotta osat pystyisivät liikkumaan oikein akselinsa ympäri. Rigi jouduttiin ottamaan irti muutaman kerran, jotta luiden asentoa ja paikkaa voitiin muuttaa. Myös esimerkiksi etusarville ja siiville lisättiin pari luuta lisää, sillä alkuperäinen määrä ei sallinut näitä osia liikkumaan halutulla tavalla.



Kuva 37. Robotin näkymä Weight Paint -tilassa.

4.4 Testaus

Valmis malli testattiin peliprojektin sisällä muuttamalla se ensin FBX-tiedostoksi vientiasetuksien kautta. Valitettavasti pelin varjostimet eivät tukeneet osittaista läpinäkyvyyttä, joten mallista ei voitu ottaa kunnollista kuvaa. Siispä viimeinen renderöity otos otettiin Blenderissä käyttämällä yksinkertaista sarjakuvamateriaalia, joka matki peliprojektin varjostinta. Kuvassa 38 on kuvattuna robotin uusi materiaali, joka tehtiin lisäämällä Shader to RGB - ja ColorRamp-noodit, joiden avulla varjoista saatiin kovemmat. ColorRamp-noodi ja robotin diffuusikartan sisältävä Image Texture -noodi yhdistettiin MixRGB-noodiin, jonka Blending mode muutettiin Multiplyksi, minkä avulla voitiin vaikuttaa varjojen väriin. Seuraavaksi lisätiin Mix Shader -noodi, johon yhdistettiin aiemmin tehty MixRGB-noodi sekä Transparent BSDF -noodi ja robotin läpinäkyvyyskartan sisältävä Image Texture -noodi, joiden avulla robotin hehkulampusta saatiin osittain läpinäkyvä. Viimeiseksi Mix Shader -noodi yhdistettiin Material Output -noodiin, jonka avulla materiaali saatiin näkymään mallin pinnalla.



Kuva 38. Sarjakuvamateriaalin noodit noodieditorissa.

Kuvassa 39 olevaa viimeistä renderöityä kuvaa varten robotista otettiin kaksi kopiota, jotka aseteltiin lähemmäksi toisiaan pystysuunnassa. Hahmojen asentoja muutettiin Pose-tilassa siten, että katsoja näkisi ne mahdollisimman monipuolisista kuvakulmista.

Seuraavaksi tilaan lisättiin muutama valo, sillä materiaaliesikatselunäkymän valaistus ei näy renderöidyssä näkymässä. Valoina käytettiin pointtivaloja, jotka aseteltiin robottien eturuumiiden yläpuolelle, jotta hahmot saataisiin valaistua yläpuolelta.

Viimeiseksi tilaan lisättiin kamera, joka aseteltiin robottien eteen. Kameran resoluutiota muutettiin, sillä molemmat hahmot eivät olisi mahtuneet oletuksena asetettujen rajojen sisälle. Valmiista kuvasta poistettiin tummanharmaa tausta Photoshopissa ja se tallennettiin PNG-tiedostona, jotta tausta pysyisi läpinäkyvänä.



Kuva 39. Renderöity kuva kahdesta valmiista robotista.

5 JOHTOPÄÄTÖKSET

Opinnäytetyön päätarkoituksena oli kartuttaa omia 3D-mallintamisen ja -veistämisen taitoja sekä saada lisäkokemusta työntajalle työskentelemisestä. Työn tärkeimpinä oppimistehtävinä olivat teoriaan syventyminen ja eri mallintamistyövaiheiden optimointi sekä aikatauluista kiinni pitäminen.

Koen onnistuneeni opinnäytetyön toteutuksessa omasta mielestäni hyvin. Työtä helpotti suuresti se, että olin kirjoittanut suurimman osan teoriaosuudesta osana Ajankohtaisseminaari ja asiakirjoittaminen -kurssia, jonka suoritin loppuvuodesta 2021. Seminaarikurssin myötä minulle jäi vain muutaman kappaleen lisääminen ja itse projektityön toteutus jäljelle.

Valitettavasti ajan puutteen vuoksi en kerennyt tehdä hahmolle animaatioita. Olisin myös halunnut testata Blenderiin asennettavia lisäosia, mutta projektityön aloittamisen venymisen takia aikaa ei jäänyt tarpeeksi niiden opettelemiseen. Työ toteutettiin täten oletusohjelmilla, jotka olivat riittäviä, mutta lisäosat olisivat mahdollisesti voineet nopeuttaa tiettyjä työvaiheita. Jatkossa pyrin pitämään aikatauluista tiukemmin kiinni, jotta työstä ei tarvitsisi karsia haluttuja asioita pois. Aikataulujen venymisestä ja animaatioiden puuttumisesta huolimatta toimeksiantajat olivat tyytyväisiä valmiiseen tuotokseen.

Opinnäytetyön aihe oli jo ennestään itselleni tuttu, mutta en ollut aikaisemmin perehtynyt sen teoriaan näin syvällisesti. Koin oppineeni uusia asioita sekä syventämään aiikeisempaa tietoa, mitä itselläni jo oli. Aihetta voisi jatkokehittää esimerkiksi tutkimalla erilaisia rigejä ja kuinka niitä voi hyödyntää videopelien hahmoja animoidessa.

Haluaisin lopuksi antaa suuret kiitokset Rust0 Gamesin väelle mahdollisuudesta tehdä opinnäytetyön heidän kauttansa. Varsinkin ohjeistus ja palaute auttoivat erittäin paljon hiomaan hahmon ulkonäköä paremmaksi ja näkemään parannettavat osat, joita yksin en olisi saattanut huomatakaan.

6 LÄHTEET

1. Haglund, V. 2012. CHARACTER DEVELOPMENT AND ITS UTILIZATION FOR CONVERGENT MEDIA FORMATS. <http://www.diva-portal.org/smash/get/diva2:615435/fulltext01.pdf>
2. Terävä, T. 2017. Workflows for Creating 3D Game Characters. https://www.theseus.fi/bitstream/handle/10024/131241/Terava_Tapio.pdf?sequence=1&isAllowed=y
3. Solarski, C. 2012. Sponsored Feature: Drawing Basics and Video Game Art: Character Design. GameDeveloper.com 4.10.2012. <https://www.gamedeveloper.com/art/sponsored-feature-drawing-basics-and-video-game-art-character-design>
4. Wikipedia. High concept. https://en.wikipedia.org/wiki/High_concept
5. Zagrobelna, M. 2014. Design and Draw a Model Sheet of a Werewolf Warrior. Design.tutsplus.com 29.12.2014. <https://design.tutsplus.com/tutorials/design-and-draw-a-model-sheet-of-a-werewolf-warrior--cms-22834>
6. Heikkilä, E. 2017. A Guide to Building a 3D Game Character. https://www.theseus.fi/bitstream/handle/10024/133905/Heikkila_Elisabet.pdf?sequence=1&isAllowed=y
7. PolyCount Wiki. Polygon Count. http://wiki.polycount.com/wiki/Polygon_Count
8. Blender Documentation Team. Blender 2.93 Manual: Mirror Modifier. <https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html>
9. Blender Documentation Team. Blender 2.93 Manual: Subdivision Surface Modifier. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/subdivision_surface.html
10. Blender Documentation Team. Blender 2.93 Manual: Simple Deform Modifier. https://docs.blender.org/manual/en/latest/modeling/modifiers/deform/simple_deform.html?highlight=simple%20deform

11. Heginbotham, C. What is 3D Digital Sculpting? Conceptartempire.com <https://conceptartempire.com/what-is-3d-sculpting/>
12. Ohra-aho, O. 2018. 3D-mobiilipelihahmon digitaalinen veistäminen. https://www.theseus.fi/bitstream/handle/10024/150819/Thesis_OutiO.pdf?sequence=1&isAllowed=y
13. Denham, T. What is UV Mapping & Unwrapping? Conceptartempire.com. <https://conceptartempire.com/uv-mapping-unwrapping/>
14. Hayden, L. 2021. What is Ambient Occlusion? Thewiredshopper.com 19.8.2021. <https://thewiredshopper.com/ambient-occlusion/?nonitro=1>
15. Cryengine. Diffuse Maps. <https://docs.cryengine.com/display/SDKDOC2/Diffuse+Maps>
16. Polycount Wiki. 2015. Texture types. http://wiki.polycount.com/wiki/Texture_types
17. Polycount Wiki. 2015. Transparency map. http://wiki.polycount.com/wiki/Transparency_map
18. Slick, J. 2020. Prepare a 3D Model for Animation With Rigging. Lifewire.com <https://www.lifewire.com/what-is-rigging-2095>
19. Zwerman S, Okun JA. 2010. Visual Effects Society Handbook: Workflow and Techniques. https://books.google.fi/books?id=ulcvdLf0h8IC&pg=PT335&lpg=PT335&dq=facial+rig+deformers&source=bl&ots=MP0kFE3ih3&sig=L_NWgw1ASW_ViaIF2sjY-oPHpKIA&hl=en&sa=X&ved=0ahUKEwjoppzXmPLPAhUGVywKHUixBw0Q6AEIT-zAl#v=onepage&q=facial%20rig%20deformers&f=false
20. Pluralsight. 2014. Rigging Guideline for the Artist: What's Important for a Good Rig? Pluralsight.com <https://www.pluralsight.com/blog/film-games/rigging-guideline-artist-whats-important-good-rig>
21. Pluralsight. 2014. Understanding Skinning - The Vital Step for Any Rigging Project. Pluralsight.com <https://www.pluralsight.com/blog/film-games/understanding-skinning-vital-step-rigging-project>

KUALÄHTEET

1. Oma kuva
2. Revoy, D. 2011. Free 3D model-sheet: Little Bot Bunny. Davidrevoy.com <https://www.davidrevoy.com/article103/little-bot-bunny-free-3d-model-sheet>
3. Autodesk Knowledge Network. Polygonal Modeling. 2016. <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-7941F97A-36E8-47FE-95D1-71412A3B3017-htm.html>
4. Blender Documentation Team. Blender 2.93 Manual: Subdivision Surface Modifier. https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/subdivision_surface.html
5. Blender Documentation Team. Blender 3.3 Manual: Sculpting Introduction. https://docs.blender.org/manual/en/latest/sculpt_paint/sculpting/introduction.html
6. Oma kuva
7. Unity Documentation. Normal map (Bump mappin). <https://docs.unity3d.com/2017.2/Documentation/Manual/StandardShaderMaterialParameterNormalMap.html>
8. Hayden, L. 2021. What is Ambient Occlusion? Thewiredshopper.com 19.8.2021. <https://thewiredshopper.com/ambient-occlusion/?nonitro=1>
- 9–11. Omaa kuvia
12. Riggingdojo. Face Rigging Rigging 101. <https://www.riggingdojo.com/face-rigging-rigging-101/>
- 13–39. Omaa kuvia