

Elina Talasmäki

# 3D-MALLINTAMISEN JA -ANIMOINNIN HYÖDYNTÄMINEN 2D-PELITUOTAN- NOSSA

Opinnäytetyö

Viestintä

Maaliskuu 2016



**KYAMK**  
University of Applied Sciences

<b>Tekijä/Tekijät</b>	<b>Tutkinto</b>	<b>Aika</b>
Elina Talasmäki	Medianomi	Maaliskuu 2016
<b>Opinnäytetyön nimi</b>		56 sivua
3D-mallintamisen ja -animoinnin hyödyntäminen 2D-pelituotannossa		
<b>Toimeksiantaja</b>		
MZD-projekti		
<b>Ohjaaja</b>		
Marko Siitonen, Brenda Jiménez Rodriguez		
<b>Tiivistelmä</b>		
<p>3D- ja 2D -grafiikoiden yhdistäminen peleissä tai muussa mediassa ei ole mikään uusi tai poikkeuksellinen asia. Kyseistä tapaa on käytetty jo pitkään ja useammalla eri tavalla, jotta saavutettaisiin haluttuja hyötyjä. Nämä voidaan jakaa esimerkiksi käytännöllisiin ja esteettisiin etuihin. Käytännössä molempien tekniikoiden käytöstä voi hyötyä esimerkiksi rahan ja ajan säästöllä, kun taas visuaalisesti yhdistelemällä näitä tyylejä voi saada aikaan estetiikkaa, joka ei pelkästään toisella onnistuisi.</p> <p>Tämän opinnäytetyön tehtävä on alustavasti kartoittaa 3D-mallintamisen ja animoinnin hyötyjä ja haittoja 2D-ympäristössä, sen visuaalisia mahdollisuuksia ja rajoituksia. Lisäksi se tarkastelee sitä, onko metodista haastajaksi 2D-animaatiolle ajansäästöllisistä näkökulmista. Projekti, johon 3D:n käyttöä harkitaan, tulisi olemaan muuten täysin 2D-grafiikoilla ja -moottorilla tuotettu. Pelihahmot olisi kuitenkin tarkoitus alun perin mallintaa ja animoida 3D-ohjelmassa. Tarkoituksena on selvittää, onko tällä tavalla mahdollista saada aikaan yhtenäistä, mutta silti mielenkiintoista visuaalista efektiä peliin niin, että sitä on järkevää käyttää täysimääräisesti useamman hahmon mallinnukseen ja animointiin.</p> <p>Teoriaosassa käydään lävitse tärkeimpiä 3D:n käsitteitä ja termejä, 3D:n käyttöä peleissä sekä sen ja 2D:n eroja. Lisäksi teoriakappaleessa analysoidaan yleisiä syitä sille, miksi näitä kahta eri grafiikkatyylä voidaan halutaan yhdistellä toisiinsa. Käytännön osassa selvitetään käytetyn 3D-ohjelman mahdollisuuksia mallintamalla yksinkertainen hahmo ja testaamalla erinäisiä työkaluja ja työskentelytapoja. Kaikkiaan tavoite on nähdä, minkälaisia visuaalisia tuotoksia näillä keinoilla voi saavuttaa, ja siten kartoittaa tuloksia tulevaa peliprojektia varten.</p>		
<b>Asiasanat</b>		
3D, 2D, pelituotanto, mallinnus		

<b>Author (authors)</b>	<b>Degree</b>	<b>Time</b>
Elina Talasmäki	Bachelor of Culture and Arts	March 2015
<b>Thesis Title</b>		56 pages
Utilisation of 3D-modeling and Animating in 2D Video Game Development		
<b>Commissioned by</b>		
MZD-project		
<b>Supervisor</b>		
Marko Siitonen, Brenda Jiménez Rodriguez		
<b>Abstract</b>		
<p>Combining 3D and 2D graphics in games or in other media is not in any way new or exceptional thing to do. Both have been utilised together for a long time in many different ways to accomplish desired results. These benefits can be divided into practical and aesthetic ones. In practice, using both techniques can be beneficial money- and time-wise. However, by combining these two different styles it is possible to achieve the kind of aesthetic that would not be possible to obtain with just one of the styles.</p> <p>The purpose of this thesis was to initially survey the pros and cons of using 3D in 2D environment, the visual possibilities and restrictions it presents. In addition, it analyses whether the technique could challenge 2D animation in a time conservative aspect. The project in which using 3D is considered would otherwise be produced with 2D graphics and engine. The player characters, however, would be modeled and animated in a 3D program at first. The goal was to figure out whether it would be possible to reach consistent, yet at the same time interesting visual effect for the game in a way that it would be sensible to utilise 3D fully as a model and animate all the player characters.</p> <p>The theory part covers the most important concepts and terms, the usage of 3D in games and the differences it has with 2D. Additionally, it analyses the common reasons for how and why the two graphic styles could be combined. In the practice section the possibilities of a 3D programme are examined by modeling a simple character and testing the different tools and working methods presented. In all, the goal was to find out what kind of visual output could be reached by these means and in that way map out the results for upcoming game projects</p>		
<b>Keywords</b>		
3D, 2D, game development, modeling		

## SISÄLLYS

1	JOHDANTO.....	5
2	PROJEKTIN TAUSTAT .....	6
2.1	Projektin kuvaus .....	6
2.2	Visuaaliset tavoitteet ja 3D-mallinnuksen rooli.....	7
3	3D-MALLINNUKSEN TEORIAA JA VAIHEITA.....	9
3.1	Peruskäsitteitä ja termistöä.....	9
3.1.1	Tärkeitä käsitteitä .....	9
3.1.2	Mallinnustekniikat.....	11
3.1.3	Teksturointi.....	14
3.1.4	Animointi .....	21
3.2	3D-mallinnus peleissä.....	23
3.3	3D- ja 2D -asettien ja animaation eroavaisuudet ja yhdistäminen.....	25
3.4	3D:n ja 2D:n yhdistämisen perusteita .....	30
4	KÄYTÄNTÖ .....	30
4.1	Lähtökohdat.....	31
4.2	3D-mallin hyödyt projektissa .....	31
4.3	Työvaiheiden kuvaus .....	32
4.4	Ongelmia ja ratkaisuja .....	44
5	LOPUKSI .....	46
	LÄHTEET.....	49
	KUVALUETTELO .....	54

## 1 JOHDANTO

Kun puhutaan 3D:n ja 2D:n yhdistämisestä peleissä, tarkoitetaan usein nimenomaan 2D-elementtien tai tekstuurien käyttöä kolmiulotteisessa ympäristössä. Esimerkiksi 2D-pelihahmojen käyttö 3D-peliympäristössä on hyvin yleistä sellaisissa peleissä, kuin Pokémon Black ja White (Frushtick 2011) ja Paper Mario Sticker Star (Elec Man EXE 2010). Toinen tyypillinen tapa yhdistää 2D:tä ja 3D:tä peleissä on luoda kaksiulotteisen ympäristön illuusio kolmiulotteisessa moottorissa esimerkiksi teksturoinnin avulla. Esimerkiksi Square Enixin Bravelly Default- ja Bravelly Second -pelit pyrkivät jäljittelemään satukirjamaista tyyliä käyttäen kuitenkin hyväkseen 3DS-konsolin 3D-toimintoa (BuiBui 2016). Samoin 3D-moottoria hyödynnettiin myös pelissä Ori and the Blind Forest, josta Bravelly-sarjasta poiketen kehkeytyi lopulta kuitenkin puhtaasti 2D-tuotos. Tapoja yhdistellä 2D:tä ja 3D:tä on siis monia.

Toisaalta 3D-elementtien käyttö 2D-alustoilla on ollut kauan käytössä animaation kentällä ja tullut yhä suosituimmaksi modernissa animaatioissa. Muiden muassa sellaiset elementit kuin tuli ja erilaiset hiukkaset pyritään generoimaan 3D-ohjelmistoilla. Näin on tehty esimerkiksi animaatioelokuvassa Egyptin Prinssi (Mainard 2016). Myös aikaa ja rahaa pyritään säästämään hyväksikäyttämällä 3D-mallinnusta perinteisen 2D-animaation rinnalla. Nykyään, kun 3D:n käyttö on kehittynyt parempaan suuntaan halvemmissakin tuotannoissa, pystytään sitä käyttämään jo pelkästään visuaalisena tehosteena ilman, että saavutetaan muita hyötyjä.

Tässä opinnäytetyössä pyrin ensisijaisesti tutkimaan sitä, miten 3D-mallintamista ja lopulta animaatiota voi hyödyntää 2D-peliprojektissa niin, että lopputulokset sulautuvat pelimaailmaan näyttämättä siitä irrallisilta. Peli tuotetaan 2D-moottorilla, johon ei ole mahdollista tuoda suoraan 3D-malleja, jonka vuoksi on myös selvitettävä, kuinka siirros eri formaattia tukevaan moottoriin onnistuu. Tavoitteena on kuitenkin, että metodin käytöllä on lopputuloksen kannalta positiivisia tyyllillisiä vaikutuksia pelin ulkoasuun, ja siten sen käytöstä on projektille erityistä etua. Lopuksi pyritään selvittämään, onko 3D:n käyttö ajansäästöllisesti hyödyllistä. Tarkoitus on siis kartoittaa mallinnuksen mahdollisuuksia, hyötyjä ja haittoja ennen projektin tuotannon aloittamista.

## 2 PROJEKTIN TAUSTAT

Tässä luvussa kuvaillaan MZD (Miracle Zombie Dream) -peliprojektin taustaa, tavoitteita ja visuaalisia visioita sekä 3D-mallinnuksen mahdollisuuksia sen tarpeiden täyttämässä.

### 2.1 Projektin kuvaus

MZD-projekti lähti käyntiin alun perin Pinja Kuusion, Essi Lehtivarjon, ja Laura Päivisen huomattua Construct 2 -pelimoottorin potentiaali 2D-pelin luonnissa heidän työstäessään aiempaa tilaustyötä. Tilatussa pelissä oli huomattavan paljon asiakkaan asettamia rajoituksia kehittämisen suhteen. Eritoten tästä syystä pelin kehittäjille syntyi tahto koetella Construct 2:n tarjoamia mahdollisuuksia laajempialaisesti. Kaikilla MZD-projektiin osallistuvilla on lisäksi kiinnostusta päästä toteuttamaan itseään riippumattomassa peliprojektissa vapain käsin ja oman ajan kanssa sekä oppimaan pelien kehitys- ja tuotantoprosesseista enemmän käytännössä.

Construct 2 on moottori, joka mahdollistaa nimenomaan 2D-pelien ohjelmoinnin ja kokoamisen. Construct on kehitetty niin, että käyttäjän olisi mahdollisimman helppo manipuloida ohjelman sisään tuotuja asetteja eli peli-elementtejä sekä koodia. Ohjelmaa voi käyttää ilmaiseksi tietyissä rajoissa, mutta jos on tarkoituksena tehdä yhtään suurempaa tuotosta, on lähes välttämätöntä hankkia lisensoitu, maksullinen versio, joka sallii ohjelman paljon laajemman käytön. Koska työryhmän edellisessä peliprojektissa tarvittiin kyseisen version toimintoja ja lisenssi hankittiin projektin jäsenten nimiin, myös MZD-projektissa on käytössä ohjelman täysi versio.

Projekti itsessään on vielä hyvin aluillaan, mutta suunnittelutyö on alkanut vauhdikkaasti. Tarkoituksena on tuottaa episodisesti luotava ja ilmestyvä selainpeli, jonka lajityyppi on alustavasti vain 2D side scroller. Muut mahdolliset lajit ovat vielä jossain määrin avoimia, mutta on erittäin todennäköistä, että peli tulee edustamaan useampaa erilaista tyyppiä. Peliin on kuitenkin suunniteltu jokseenkin syvällistä combopohjaista taistelumekaniikkaa, joten sen lajityyppi tulee mahdollisesti inspiroitumaan myös esimerkiksi taistelupeleistä. Pelattavia hahmoja tulisi useampia ja hahmoilla olisi erilaiset taistelu- ja toi-

mintatyylit. Sekä tarina että pelimekaniikat ovat tärkeässä roolissa, eikä kumpaakaan haluta jättää huomiotta toisen kustannuksella. Peli aiotaan ainakin alustavasti tarjota ilmaiseksi pelattavaksi, mutta projektissa uskotaan olevan potentiaalia myös mahdolliseen kaupalliseen tuotteeseen, mikäli se onnistuu toivotulla tavalla.

## 2.2 Visuaaliset tavoitteet ja 3D-mallinnuksen rooli

Visuaalisesti pelin on tarkoitus olla tietyssä määrin ”kieli poskessa” tehty ja lähes parodisesti japanilaista anime-tyyliä ja kliseitä jäljittelevä. Pelistä halutaan tehdä ulkomuodoltaan yhtä aikaa yltiösöpö (röyhelöitä, suuria silmiä, kimalusta) ja samalla yhdistää karuja sekä häiriintyneitä elementtejä kuten verta ja suolenpätkiä. Tavoitteena on, että lopputuloksesta tulisi omalla tavallaan hyvinkin ahdistava herttaisesta ulkokuorestaan huolimatta, tai jopa sen ansiosta. Samaan aikaan olisi toivottavaa, että peli ei uppoisi kuitenkaan massaan herttaisella ulkomuodollaan, vaan erottuisi vastaavankaltaisten pelien lukuisesta joukkiosta. Pelin parodisten elementtien ei myöskään ole tarkoitus olla karikaatyyrisiä tai pilkkaavia lähdemateriaalia kohtaan, vaan niiden on enemmänkin tarkoitus olla kunnianosoitus lajityypilleen.

3D:n rooli tässä pelissä on vielä hyvin teoreettisessa vaiheessa, eikä sitä alun perin edes harkittu käytettävän. Idean tultua ilmoille haluttiin selvittää, mikäli olisi mahdollista luoda 3D:tä käyttäen mielenkiintoinen ja luova visuaalinen ilme. Samalla tahdotaan kuitenkin säilyttää pelin ulkomuodon yhtenäisyys. Tämän opinnäytetyön pääasiallinen tavoite onkin selvittää, voiko mallinnuksen keinoin luoda peliin omaleimaisia, visuaalisesti vaikuttavia efektejä ilman, että ne näyttävät liian irrallisilta pelin 2D-elementeistä. Toinen tärkeä tutkimuskysymys on se, olisiko pelin päähahmojen animaatio tehokkaampaa ja nopeampaa luoda 3D-ohjelmassa, kuin jollakin 2D-tekniikalla.

Pelianimaatiosta puhuttaessa eräs merkittävä peruste on sprite sheettien hyödyntäminen. Sprite tarkoittaa mitä tahansa yksittäistä graafista elementtiä, joista peli lopulta rakentuu, eikä tällainen yksikkö itsessään ole animoitu. Sprite sheet taas on usesta spritekuvasta koostuva kuvakokonaisuus, joka niin säästää tilaa kuin myös nopeuttaa tahtia, jolla kuvia saadaan esiin loppu-

tuotoksessa. Tämä perustuu siihen, että animaatio-ohjelmille ja pelimoottoreille on yksinkertaisempaa siirtyä yhdessä kuvassa toiseen kohtaan, kuin noutaa kokonaan eri kuva joka kerta animaation edetessä (kuvat 1 a ja b).



Kuvat 1 a ja b. Vasemmalla kuvassa on yksinkertainen sprite, kun taas oikealla kuvassa on sprite sheet, jossa näkyvät kaikki Spritet tytön juoksuanimaatiosta (Lambert 2013).

Toisin sanoen sprite sheet-animaatiossa luodaan illuusio liikkeestä vaihtamalla sheetissä olevaa kuvaa peräkkäin nopeassa tahdissa (Lambert 2013). Tämä on myös se muoto, jollaiseksi MZD-projektissa 3D-animaatio muutetaan sen jälkeen, kun animaatiot ja niiden osat on saatu valmiiksi 3D-ohjelmassa. Tämän johdosta lopullinen tuotos ei käytännössä ole enää 3D:tä. Pelissä pääpaino ei missään tapauksessa tule olemaan 3D:ssä. Vaikka lopulta päädyttäisiin siihen tulokseen, että tätä tekniikkaa käytetään, tultaisiin 2D-formaattiin muunnettuja spritejä muokkaamaan vielä kuvankäsittelyohjelmassa jälkikäteen. Tällä tekniikalla tahdotaan saavuttaa uniikki ja mieleenpainuva tulos.

Vaikka mahdollisia 3D-mallinnusta ja animaatiota hyödyntäen tehtyjä spritejä ei lopulta edes käytettäisi pelissä esimerkiksi tyylin sopimattomuuden vuoksi, on mahdollista hyväksikäyttää luotua materiaalia esimerkiksi rotoskooppaamalla liikkeitä. Rotoskooppauksessa käytetään olemassa olevia kuvasarjoja tai videoita ja niiden peräkkäisiä frameja, jotka ovat yksittäisiä kuvaruutuja ja joista videot ja animaation koostuvat. Niistä liike jäljitellään läpi ja käytännössä siis kopioidaan, jotta saadaan aikaan vaikkapa kävelyanimaatio pelihahmolle (Fairley 2014). Peruste tämän metodin mahdolliselle käytölle on se, että 2D-animaatiossa liikkeen suunnittelu ja sen toteuttaminen vaatii usein yhtä suuren työtaakan jokaisen animaation kohdalla. 3D-mallia käyttäessä kaikkia as-

setteja ei tarvitse luoda alusta jokaisen animaation kohdalla, vaan sama pohjatyö mahdollistaa yksittäisten animaatioiden kehittämisen aloittamisen suoraan.

### 3 3D-MALLINNUKSEN TEORIAA JA VAIHEITA

Tässä kappaleessa käsitellään 3D-mallinnuksen teoriaa, siihen liittyvää termistöä, käyttöä ja eroavaisuuksia 3D:n sekä 2D:n välillä.

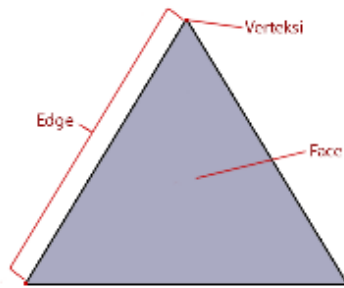
#### 3.1 Peruskäsitteitä ja termistöä

3D on tietokoneella tuotettua, matematiikkaan perustuvaa kolmessa ulottuvuudessa hahmotettavaa grafiikkaa. 3D:n tekemisen ja tuottamisen helpottamiseksi on kehitetty erinäisiä ohjelmia erilaisiin tarkoituksiin, ja näiden ohjelmien työkalut auttavat rakentamaan kolmiulotteisia tuotoksia. Valmiit mallit voidaan tuoda ulos 3D-ohjelmasta joko niissä olevien renderöintiprosessien avulla kaksiulotteiseen kuvamuotoon, kuten staattiseksi kuvaksi tai videokuvaksi, tai tulostaa 3D-printterien avulla fyysiseksi esineeksi. (Ghosh 2014.)

##### 3.1.1 Tärkeitä käsitteitä

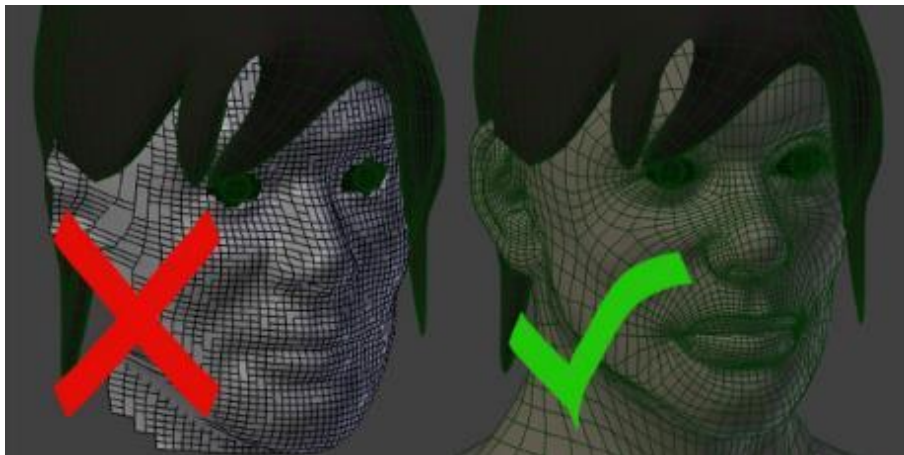
Polygoni on kaikkein käytetyin geometrinen yksikkö 3D-maailmassa. Se koostuu kolmesta eri yksikköryhmästä; vertekseistä eli pisteistä, edgeistä eli sivuista sekä faceista eli pinnoista, jotka jäävät vähintään kolmen edgen väliin (kuva 2 a). Polygoni on näiden elementtien kokonaisuus ja 3D-malli koostuu usean polygonin summasta. Tätä summaa kutsutaan polygoniverkoksi. (Digitaltutors 2014.)

Lisäksi polygoneja on kolmea eri tyyppiä: kolmikulmainen ja -sivuinen triangle tai tri-polygoni, nelikulmainen ja -sivuinen quad-polygoni sekä viisi- tai useampikulmainen ja -sivuinen ngon-polygoni (kuva 2 b). Näistä quad-polygoni on yleisesti hyväksytyin mallintamisessa, sillä se ei vääristy niin helposti animoidessa ja on muutenkin miellyttävämpi käsitellä mallintaessa, sillä sitä on helppo jakaa kahtia (Mayden 2015). Quad-polygonit voidaan myös jakaa tri-polygoneiksi ilman suurempia vaikeuksia.



Kuvat 2 a ja b. Vasemmalla kolmion mallinen polygoni, joka koostuu kolmesta verteksistä, kolmesta edgestä ja yhdestä facesta. Oikealla eri polygonityypit vasemmalta oikealle: tri-polygoni, quad-polygoni ja ngon-polygoni (teachmecone 2016)

Polygoniverkoilla on niin kutsuttu topologia, joka tarkoittaa sitä, miten eri polygonit ovat kiinni toisissaan ja kulkevat pitkin geometriaa. Hyvä topologia on soljuvaa, selkeälinjaista sekä kulkee geometrian mukaisesti, jolloin mallia on helppo ymmärtää, kun taas huono menee sattumanvaraisesti joka suuntaan ja tekee mallista epäselkeän (kuva 3). Huono topologia voi aiheuttaa polygoniverkon pintaan vääristymiä muun muassa animoinnin yhteydessä. (Digitaltutors 2014.)



Kuva 3. Esimerkki huonosta ja hyvästä topologiasta (D J T 2015)

Mallinnuksessa tärkeä asia on myös polygonimäärät. Polygonimäärät tarkoittavat yksinkertaisuudessaan sitä, kuinka monta polygonia mallissa on. Mitä suurempi polygonimäärä on, sitä raskaampia malleista tulee ja sitä enemmän ne vievät tilaa.

3D-mallintaminen on usein hidasta ja itseään toistavaa aktiviteettia, kun mallia tehdään manuaalisesti. Tämän vuoksi on tärkeää opetella käyttämään ohjelmista löytyviä niin sanottuja hotkeys-pikanäppäimiä. Nimensä mukaan niiden

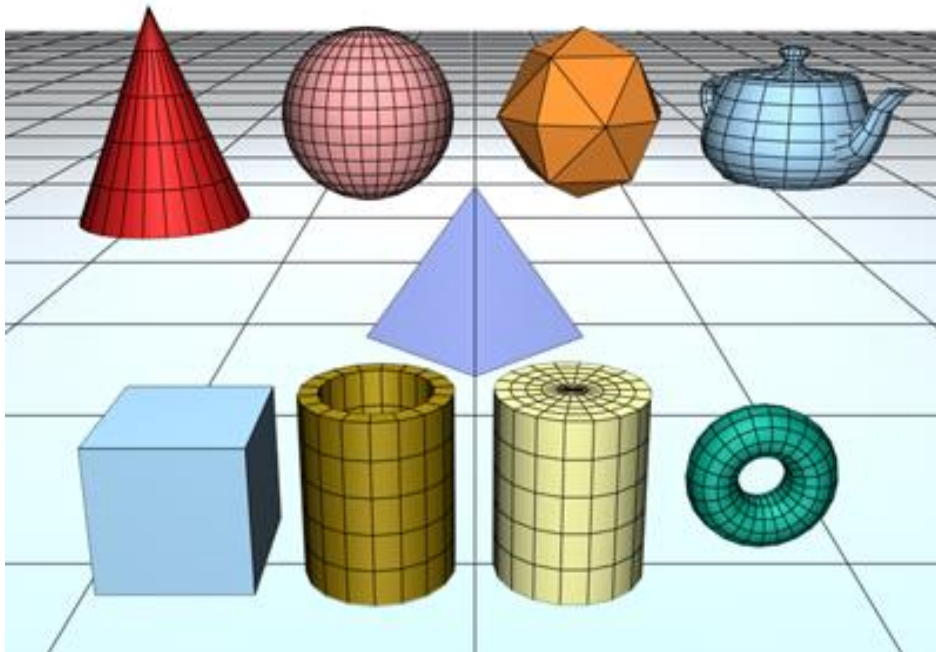
avulla pystyy näppäimistön kautta valitsemaan nopeasti esimerkiksi halutun työkalun, joka muuten saattaisi olla useamman valikon ja hiirennäppäyksen takana. Niiden avulla työskentelystä tulee huomattavasti nopeampaa ja mielekkäämpää. (Gahan 2009, 169.)

### 3.1.2 Mallinnustekniikat

Slickin (2014) mukaan 3D-mallinnustekniikoita on useita eri tyyppiä:

- Polygonimallinnus
- NURBS-mallinnus
- Digitaalinen muotoilu
- Proseduraalinen mallinnus
- Kuvaan(generoitu) perustuva mallinnus
- 3D-skannaus

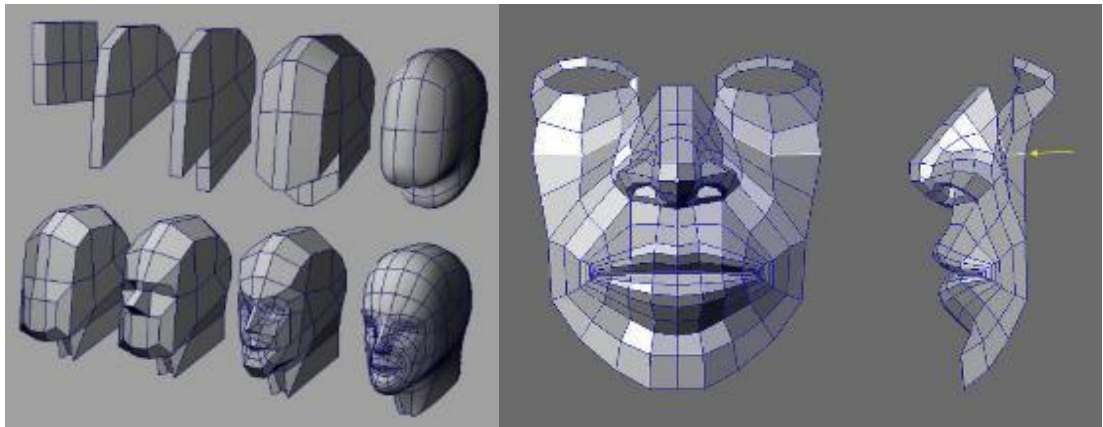
Polygonimallinnus perustuu nimensä mukaan polygonien manipulointiin ja rakentamiseen. Polygonimallinnuksen voi vielä erikseen jakaa kahteen eri alaryhmään, box- eli laatikkomallinnukseen sekä edge-mallinnukseen. Boxmodeling-mallinnustekniikassa mallia lähdetään rakentamaan valmiista geometria-alkiosta (kuva 4), joita 3D-ohjelmissa on yleensä tarjolla käytettäväksi.



Kuva 4. Kuvassa on joukko 3ds Max-ohjelmasta löytyviä geometria-alkioita. Nämä alkioitkin ovat polygoniverkkoja, joskin yksinkertaisia sellaisia (autodesk 2016).

Kun on valittu sopiva valmis geometria-alkio, sitä muokataan asteittain muun muassa sellaisten työkalujen kuin ekstrude ja subdivide avulla monimutkaisemmaksi, kunnes saavutetaan haluttu muoto (kuva 5 a).

Edge-mallinnuksessa lähdetään liikkeelle luomalla yksinkertaisia yhden tai useamman edgen silmukoita tai jonoja, joilla tehdään mallin yleispiirteitä. Näitä ovat esimerkiksi hahmon siluetti tai silmän ääriiviiva. Vähitellen kyseisiä piirteitä täydennetään ja yhdistetään lisäämällä faceja sekä enemmän edgejä. Edge-mallinnus mahdollistaa yksityiskohtaisemman työn teon kuin boxmodeling-mallinnus heti kättelyssä, ja tämän vuoksi sitä käytetään mieluusti tarkkuutta vaativien kohteiden tekoon, kuten esimerkiksi ihmiskasvoihin (kuvat 5 a ja b).

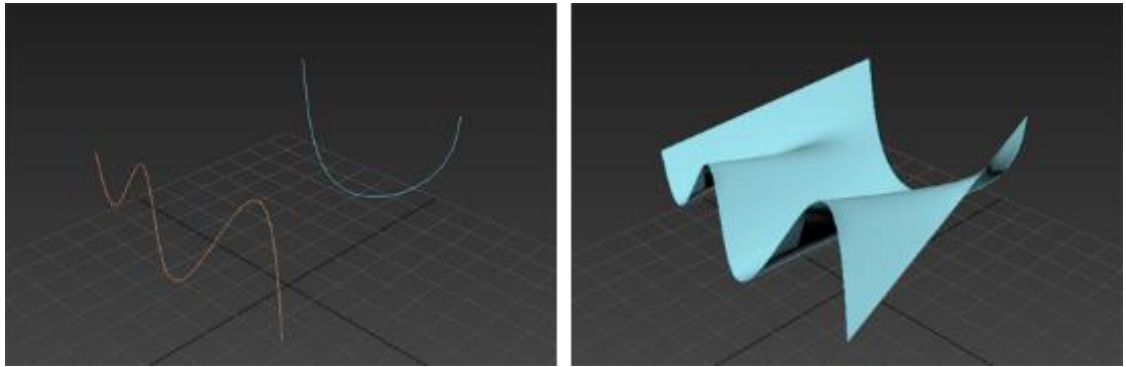


Kuvat 5 a ja b. Ensimmäisessä kuvassa on esimerkki polygonimallinnuksesta ja toisessa edge-mallinnuksesta. Huomaa, kuinka edge-tekniikassa päästään nopeasti tarkempaan yksityiskohtien työstöön. (Jones 2015.)

Näitä kahta tekniikkaa käytetään usein myös sekaisin esimerkiksi ihmistä mallintaessa. Kasvot, kädet ja jalkaterät voivat olla mielekkäämpiä tehdä edge-mallinnuksella ja loppuvartalo boxmodeling-tekniikalla. Lopuksi aikaansaadut erilliset polygoniverkot voidaan yhdistää kokonaiseksi ihmismalliksi. (Slick 2014.)

NURBS (Non-Uniform Rational B-Spline) -mallinnus perustuu kolmiulotteisessa tilassa oleviin käyriin, jotka 3D-ohjelma yhdistää luomalla käyrien välille sileitä pintoja (kuva 6). Kyseisten käyrien asentoa ja muotoa muutetaan CV (control vertices) -kahvojen avulla. NURBS on ollut suosittu esimerkiksi teknisten mallien, kuten autojen ja arkkitehtuurin, suunnittelussa sen vuoksi, että NURBS-malliin on helppo saada erilaisia kulmia ja yksityiskohtia ilman vääristymiä mallin pinnassa. NURBS-tekniikalla tehdään nimenomaan staattisia

malleja, sillä niiden animointi voisi olla hankalaa ja aiheuttaa mallin geometriaan vääristymiä.



Kuva 6. Esimerkki NURBS-käyrien yhdistämisestä

Digital sculpting eli digitaalinen muotoilu on uudempi tulokas 3D-mallinnuksen saralla ja on vienyt jonkin verran NURBS-mallinnuksen suosiota esimerkiksi elokuva-alalla. Digitaaliseen kuvanveistoon perustuvilla uusilla muotoiluohjelmilla (kuten ZBrush) on mahdollista saada aikaan helposti ja nopeasti todella yksityiskohtaista ja orgaanista jälkeä. Myös polygonitiheys saa olla äärimmäisen korkea ilman minkäänlaisia haittatekijöitä. Sen sijaan, että mallintaja joutuisi rakentamaan mallia hitaasti pala palalta, digitaalisissa muotoiluohjelmissa 3D-polygoniverkkomassaa muokataan lähes samalla lailla, kuin muotoilijat käsittelevät savea. Nämä ohjelmat tosin vaativat pitkälti piirtopöytien käytön tai vastaavien fyysisten kalustojen hallitsemisen, sillä ne simuloivat painetta, jota luonnollisesti tarvitaan muotoiluun. (Slick 2014.)

Proseduraalinen mallinnus tarkoittaa käytännössä sitä, että malli tai mallit luodaan ohjelmallisesti tiettyjen säädettyjen algoritmien mukaan ilman, että malliin kajotaan välillä ihmisen toimesta. Tätä mallinnustapaa käytetään sellaisissa tapauksissa, kun jonkin asian tuottaminen olisi liian hankalaa, kallista ja aikaa vievää mallintajalle itselleen. Esimerkiksi puiden lehtien tiheyttä olisi järjestöntä säätää manuaalisesti. Jotkin ohjelmat mahdollistavat kokonaisien ympäristöjen luomisen proseduraalisesti, oli se sitten metsikkö tai kaupunkimaisema. (Suridge 2013.)

Kuvaan perustuva mallintaminen toimii siten, että 3D-ohjelmaan syötetään useampia 2D-kuvia halutusta kohteesta, esimerkiksi patsaasta tai rakennuksesta, ja ohjelma luo kuvien pohjalta algoritmien avulla 3D-mallin. Tämä tekniikka mahdollistaa nopean ja rahaa säästävän 3D-mallinnuksen. (Slick 2014.)

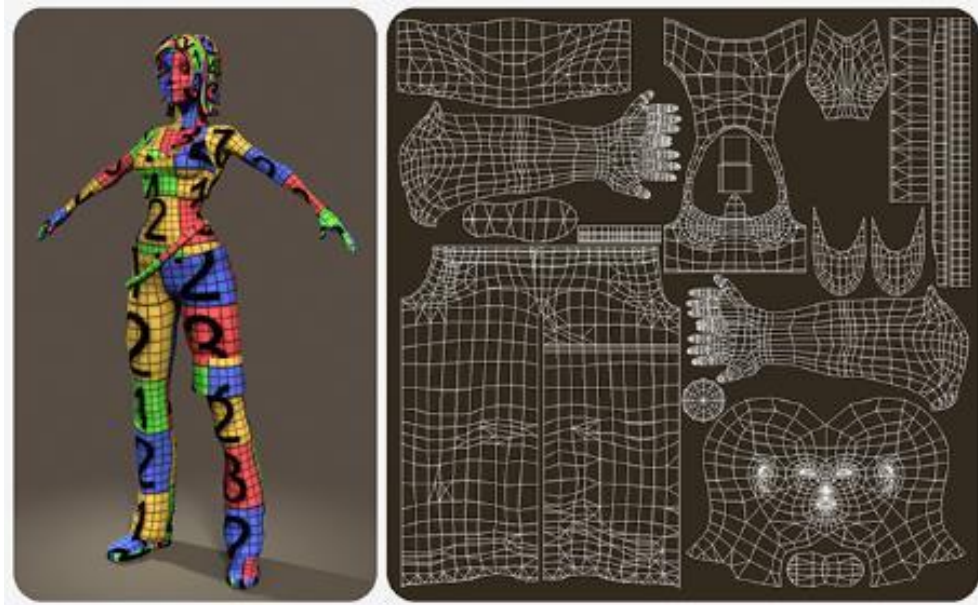
3D-skannaus on tapa tehdä oikeista esineistä ja ihmisistä äärimmäisen tarkkoja polygoni- tai NURBS-geometriaverkkoja. Tätä käytetään muun muassa elokuva-alalla, kun reaali maailman asiasta halutaan tarkka kopio elokuvaan syystä tai toisesta. (Slick 2014.)

Vaikka 3D-skannaus on usein enemmän ison budjetin projektien käytössä, on mahdollista kotioloissakin tavallaan käyttää tätä tekniikkaa; samaa tarkkaa laatua se ei kuitenkaan tarjoa (samandniko 2014).

### 3.1.3 Teksturointi

Kun 3D-malli on saatu valmiiksi, on seuraava ja visuaalisesti mallin viimeistelevä vaihe on tekstuurien luominen. Tekstuurit ovat 3D-mallin pintaan projisoitavia 2D-kuvatietoja tai bittikarttoja, joilla on mahdollista antaa mallille sellaisia ominaisuuksia kuin värit, kirkkaus, kiiltävyys ja heijastus. Nämä efektit luodaan yleensä erillisillä tekstuurityypeillä ja niiden yhteiskäytöllä saadaan aikaan vakuuttava ulkopinta malliin. (Slick 2016.)

Jotta malliin on helpompaa laittaa tekstureita, on sille luotava UV-asettelu eli UV-kartoitus. Tämä tarkoittaa sitä, että mallin geometria ikään kuin avataan tasaiseksi pinnaksi (kuva 7), sillä olisi kovin hankalaa luoda tekstureja monimutkaisen geometrisen mallin pinnalle (Slick 2016). Poikkeuksen tähän tuovat nykyään sellaiset ohjelmat kuten Zbrush ja Mudbox, joissa pystyy helposti maalaamaan suoraan mallin päälle (unityhacker 2012). UV:t itsessään ovat 2D-tekstuurien koordinaatteja ja ne vastaavat verteksien paikkoja 3D-mallin pinnalla, joissa U vastaa yleensä horisontaalista tekstuurisuuntaa ja V vertikaalista. Toisinaan puhutaan myös UVW:sta, jolloin koordinaatteihin on lisätty vielä syvyystekijä, jota kirjain W edustaa (Baker 2010).



Kuva 7. Vasemmalla malli ja oikealla mallille tehty UV-kartta (Ward 2013)

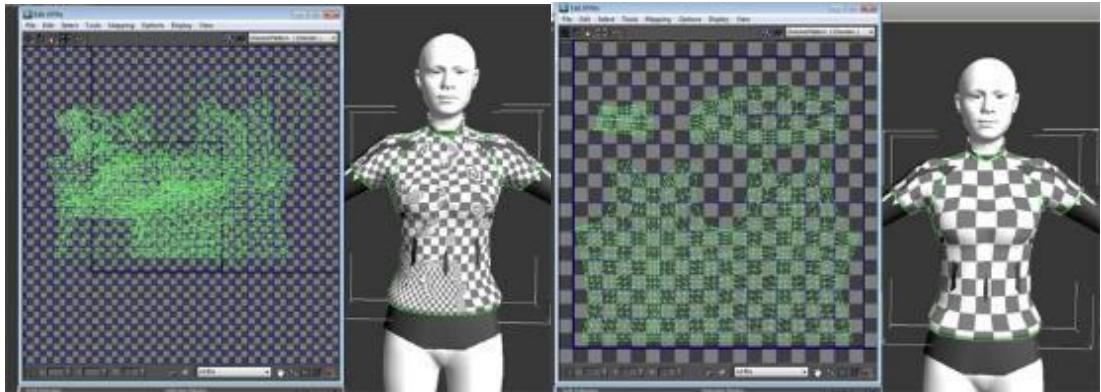
Kartoituksen voi jakaa karkeasti kahteen tyyppiin: automaattiseen ja manuaaliseen. Automaattisessa kartoituksessa 3D-ohjelmalla on niin sanottuja valmiita pohjia mallille, joiden mukaan 2D-kuva asettuu valittujen polygonien päälle. Näissä tapauksissa mallille ei siis itse luoda UV-karttaa ja ne toimivat yleisesti vain yksinkertaisen geometrian kanssa. Tyyppejä on useampia, mutta tärkeimmät ovat spherical eli pallon mallinen, cylindrical eli sylinterin mallinen sekä planar eli tasomallinen kartoitus (Kuva 8). (Masters 2014.)



Kuva 8. Vasemmalta oikealle: pallon mallinen kartoitus, sylinterin mallinen kartoitus ja tasomallinen kartoitus (Masters 2014)

UV:n kartoitus on helppo aloittaa tällaisella automaattisella tekniikalla. Jotta olisi mahdollista kartoittaa monimutkaisempia asioita, kuten kokonainen ihmishahmo, voi UV:n avata manuaalisesti. Tällä tavalla mallin saumojen (kohdat, joista UV:osat on leikattu erilleen) paikkoja voi siirtää haluamallaan tavalla, mikä taas sallii siistimmän UV:n ja sitä myötä tekstuurin aikaansaamisen. Lisäksi manuaalisesti voi tarvittaessa siirtää jopa yksittäisiä UV-pisteitä kerrallaan. (Slick 2016.)

Kartoitus on valmis, kun UV:t on aseteltu toivotulla tavalla niin, että koordinaatit on jaettu tasaisesti mallin pinta-alalla ja saumat on saatu sopiviin kohtiin, joissa ne on helppo piilottaa tai korjata (kuva 9). Tämän jälkeen valmis UV-kartta voidaan viedä ulos 3D-ohjelmasta toiseen ohjelmaan kuten Photoshopiin, jossa 2D-tekstuuri luodaan UV-kartan avulla. Kun tekstuurit on luotu, ne tuodaan takaisin 3D-ohjelmaan ja voidaan laittaa suoraan mallin päälle kiitos valmiin UV-kartan. Tämänkin jälkeen UV:ta voi vielä muokata, jos tekstuuri ei istu täysin.

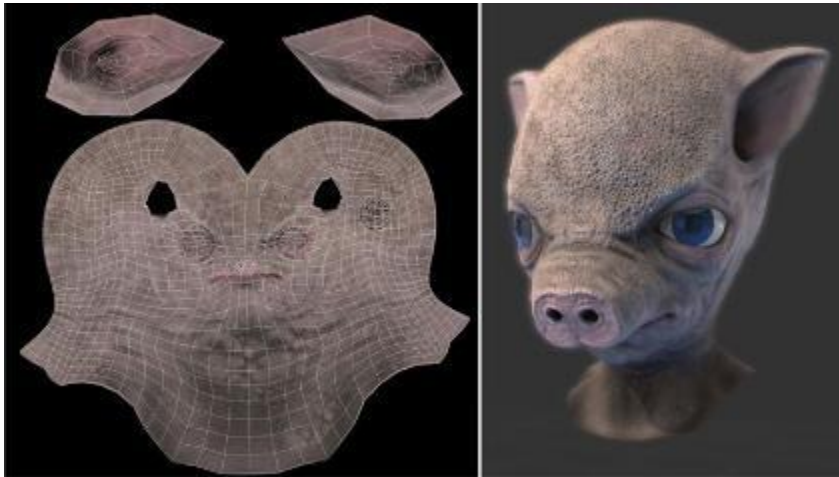


Kuva 9. Vasemmalla näkyy esimerkki huonosta UV-kartasta ja oikealla siisti, valmis UV-kartta. Materiaali 3D-mallin paidassa on checker-ruudukkokuviio, joka auttaa hahmottamaan paremmin, kuinka UV on aseteltu (Grinchesku 2015).

Yleisimmin käytettyjä tekstuurityyppejä ovat:

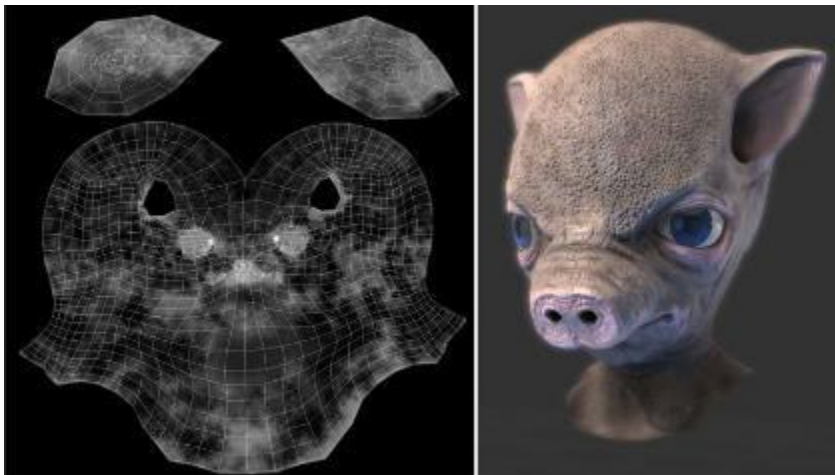
- Diffuse-kartta (tai Color-kartta) eli värikartta
- Specular-kartta eli kiiltokartta
- Bump-kartta
- Normal-kartta
- Displacement-kartta
- Transparency-kartta
- Ambient occlusion-kartta

Värikartat ovat kaikkein yleisimpiä tekstuurikarttoja. Niissä voi käyttää suoraan valokuvaa tai teksturoijan itsensä maalaamaa 2D-tekstuuria. Värikartoilla voidaan yksinkertaisimmillaan antaa esimerkiksi lattiaan laattatekstuuri valokuvalla ja monimutkaisimmillaan antamaan värit kokonaisuudelle hahmolle kaikessa monimutkaisuudessaan (kuva 10). Se on pohja koko tekstuurille, muttei kuitenkaan välttämättä ainoa kartta, joka yksittäiselle mallille annetaan. (Slick 2014.)



Kuva 10. Kuvassa näkyy olennolle luotu värikartta. Kartassa näkyvät myös UV:t. (Russell 2014.)

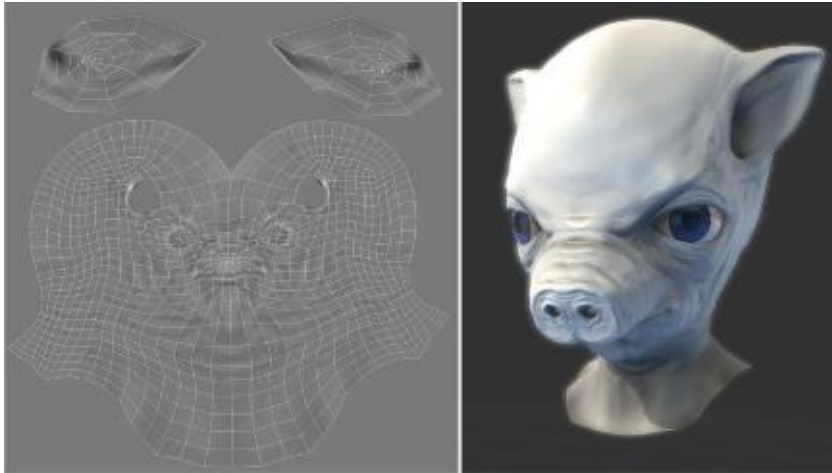
Kiiltokartta nimensä mukaan kertoo 3D-ohjelmalle mihin, kohtaan mallia tulee enemmän tai vähemmän kiiltoa (Kuva 11) (Russell 2014).



Kuva 11. Kuvattuna on kiiltokartta/specula-kartta. Erityisesti silmien ympärillä ja kärsän alueella voi nähdä kuinka kartta on vaaleampi, eli kiiltoa tulee noille alueille enemmän. (Russell 2014.)

Normal-kartta ja bump-kartta luovat tekstuurin pintaan illuusion kolmiulotteisista yksityiskohdista valaistuksen avulla. Tekstuurin pinta ei kuitenkaan oikeasti muuta syvyyttään. Toisin sanoen tietyistä kulmista voi nähdä kuinka yksityiskohdat ovat oikeasti täysin latteita. Näiden karttojen avulla voidaan säästää mallin tiedostokoossa, sillä malli voi itse olla matalapolygoninen ja silti näyttää hyvinkin yksityiskohtaiselta. (Slick 2014.)

Bump-kartta on näistä kahdesta vanhempi ja myös rajoittuneempi, joten sillä voi saada aikaan lähinnä pieniä ja yksinkertaisia yksityiskohtia (kuva 12), kuten puisen lankun syitä.



Kuva 12. Bump-kartta, jossa näkyy pieniä muutoksia (Russell 2014).

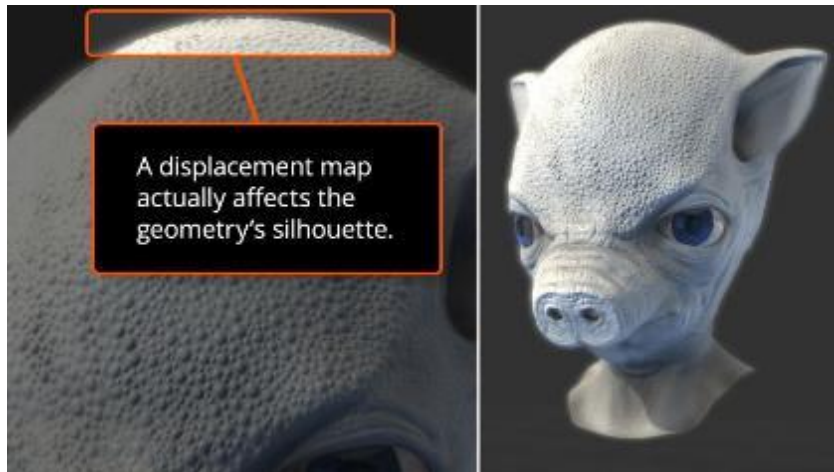
Normal-kartta taas on uudempi kartoitustekniikka ja salliikin huomattavasti suurempia ja tarkempia yksityiskohtia. Tämä johtuu siitä, että toisin kuin bump-kartat, jotka esitetään harmaansävyissä, normal-kartat hyödyntävät RGB:tä eli punaista, vihreää ja sinistä. Värit ovat suoraan verrannollisia 3D-ohjelman X-, Y- ja Z-koordinaattitietoihin. Tämän johdosta 3D-ohjelma hahmottaa tarkasti, missä asennossa pintojen kuuluu olla. (Russell 2014.)

Jotta normal-kartta voidaan luoda, on 3D-mallista tehtävä kaksi versiota: matalapolygoninen ja korkeapolygoninen (kuva 13). Korkeapolygonisesta versiosta "leivotaan" tekstuurirenderöinnin avulla normal-kartta, joka projisoidaan matalapolygonisen mallin päälle (Blender Reference Manual). Tekstuurirenderöinti tai tekstuuri-baking on prosessi, jossa halutut polygoniverkkotiedot muunnetaan tekstuurimuotoon. Toisin sanoen kokonaisen 3D-mallin pinta-alan voi muuntaa tekstuuriksi. Tätä tekniikkaa hyödynnetään renderöintiaikojen lyhentämiseksi. (Autodesk Knowledge Network 2014.)



Kuva 13. Kuvassa vasemmalta oikealle: ensimmäisenä matalapolygonimalli, johon on lisätty tekstuurirenderöity normal-kartta, keskellä pelkkä matalapolygonimalli ja viimeisenä korkeapolygonimalli, josta tekstuurirenderöinti tehtiin. (F 2013.)

Displacement-kartta poikkeaa kahdesta edellämainitusta siinä, että se nimenomaan muuttaa mallin pintaa fyysisesti (kuva 14). Tämä vaatii sen, että 3D-mallin polygoneja jaetaan. Vaikka yksityiskohdat ovat miellyttävämpiä tällä kartalla kuin bump- ja normal-kartoilla, on displacement-kartta raskaampi käyttää ja vie enemmän aikaa renderöidä. (Russell 2014.)



Kuva 14. Kuvassa näkee, kuinka hahmon päässä on patteja, jotka on luotu displacement-kartan avulla (Russell 2014).

Muita yleisiä käytössä olevia karttoja ovat transparency map eli läpinäkyvyyskartta (kuva 15) ja reflection map eli heijastuskartta. Molemmissa kartta kertoo 3D-ohjelmalle, minkä alueen mallista tulisi olla joko läpinäkyvä tai heijastava riippuen kartasta. (Slick 2014.)



Kuva 15. Hahmolle on luotu tatuointi läpinäkyvyyskartan avulla (Russell 2014).

Ambient occlusion simuloi 3D-mallille sellaista valaistusta ja varjoja, jollaisia pilvisenä päivänä voisi olla. Se ei ole täysin todenmukainen, mutta näyttää ihmissilmään mielekkäältä ja tekee mallista realistisemmän oloisen korostaessaan yksityiskohtia, joita ei välttämättä muuten näkisi (kuva 16). Tarkalleen ottaen se mittaa eri pintojen etäisyyden toisistaan ja luo niiden välille varjoja. (Rosen 2009.) Ambient occlusionia on kuitenkin nykyään hankalampi käyttää esimerkiksi peliympäristöissä, sillä niin monet objektit pelimaailman sisällä ovat liikuteltavia. Koska ambient occlusion sisältyy staattisena mallin tekstuurireihin, on ymmärrettävää, ettei se liiku elementtien mukana. Paikallaan pysyvissä objekteissa ja joissain tapauksissa pelihahmoissa tätä tekniikkaa kuitenkin voi käyttää hyväksi. (DWSMatt 2015.)



Kuva 16. Vasemmalta oikealle: hahmo ilman tekstuuria, hahmo ambient occlusion-kartan kanssa ja lopuksi hahmo, jolla on sekä värit että ambient occlusion (Rosen 2009).

Ambient occlusion yleensä tekstuurirenderöidään 3D-mallista ja lopputulos lisätään suoraan värikarttaan Photoshopissa tai vastaavassa ohjelmassa. On myös olemassa renderöintitekniikoita (mm. SSAO eli screen space ambient occlusion), jotka emuloivat ambient occlusion -valaistusta tosiajassa pelien sisällä. Nämä tekniikat toimivat luonnollisesti paremmin liikkuvien ja animoitujen elementtien kanssa, mutta vaativat huomattavasti enemmän tehoa kohdetietokoneelta.

Viimeinen osio tekstuurien luonnissa on niiden kokoaminen 3D-ohjelmassa. Tämä tapahtuu siten, että ohjelmassa tarjolla oleville oletusmateriaaleille valitaan shader eli geneerinen pintamateriaali, johon kyseisen shaderin mahdollisuuksien mukaan voidaan lisätä aiemmin luotuja 2D-tekstuureja (esimerkiksi

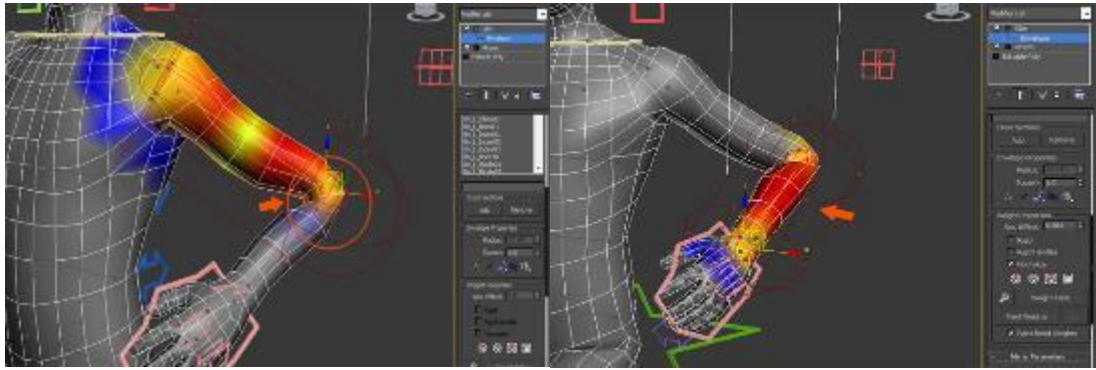
värikartta ja bump-kartta). Shaderit ovat niin sanottuja pienoishjelmia, jotka pitävät sisällään käskyjä 3D-ohjelmalle siitä, kuinka valoa tulee käsitellä mallin pinnassa reaaliajassa (Ahearn 2009, 95). Erilaiset shaderit sallivat hieman erilaisia muutoksia. Sen lisäksi, että shader voi saada valaistustiedot aiemmin mainituilla tekstuuri-kartoilla, voi tietoja myös säätää vapaasti muiden muassa sellaisilla parametreillä kuin diffuse-väri, läpikuultamattomuus, heijastavuus, spekulaaarisuus, hehku ja niin edespäin. Näin saadaan aikaan lopullisen tekstuurin pinta. (Slick 2014.)

### 3.1.4 Animointi

Kun 3D-malli on saatu valmiiksi, voidaan aloittaa sen animointi. Animoinnin eri vaiheet ovat riggaus, skinnaus ja itse animaatio. Riggaus tarkoittaa niin sanottua liikuteltavan luurangon luomista mallille. Skinnaus taas viittaa mallin liittämiseen luotuihin luihin. Nämä prosessit vaikuttavat kriittisesti siihen, kuinka helppoa itse hahmon animoiminen tulee olemaan. (Chopine 2011, 94-95.)

3D-malli vaatii liikkuaakseen luurangon, hyvin samaan tapaan kuin ihminenkin vaatii. Niin sanottu riggi koostuu luista ja nivelistä, joiden avulla animaattori voi saada hahmon liikkumaan. Ensimmäinen tehtävä rigin rakentamisessa on luurangon luominen sekä sen kokoaminen ja asettaminen paikalleen. Luiden pitäisi olla niitä vastaavissa paikoissa hahmomallin anatomian mukaan. Seuraavaksi luut yhdistetään hierarkisesti toisiinsa, jotta ne olisivat kiinnittyneet oikeista paikoista ja seuraisivat toisiaan realistisesti liikuteltaessa. Kaikki luut päätyvät asteittain pääniveleen, joka asetetaan yleensä hahmon keskusta, ”lantioluuhun”. (Slick 2014.)

Riggauksen jälkeen on mallille tehtävä skinnaus. Skinnauksen tehtävä on sitoa mallin geometria ja luut toisiinsa niin, että luita liikuteltaessa mallikin liikkuu. Jotta lopputuloksesta tulisi hyvä niin, että malli liikkuu luiden mukana luonnollisesti ja vääristymättä, pitää skinnaus yleensä säätää manuaalisesti. Esimerkiksi ihmishahmoa skinnatessa kyynärvarren ja olkavarren taiteessa mallin verteksit täytyy jakaa tasaisesti molempien luille, jotta lopputulos olisi hyvä (kuva 17). (Masters 2014.)



Kuva 17. Vasemmalla näkyy, kuinka malli on vääristynyt taiteen kohdalta. Oikealla taiteen verteksit on jaettu paremmin ja taive käyttäytyy luonnollisemmin. (Kumari 2012.)

Animointi voidaan aloittaa, kun skinnaus on saatu valmiiksi. Animoinnin helpottamiseksi löytyy joitain työkaluja, kuten esimerkiksi forward kinematics ja inverse kinematics. Nivelestä, johon on määrätty forward kinematics alaspäin, lähtevät muut nivelet ja luut ovat kyseisen ensimmäisen yksilön liikealaisuudessa. (Slick 2014.) Inverse kinematicsissa (IK) vastakkaisesti jonon viimeinen luu asetellaan animaattorin kädestä ja loput luut seuraavat liikettä. Ilman IK:ta jokainen luu pitäisi asetella yksitellen oikeaan asentoon, mikä tekisi animaatiosta pitkän päälle hankalaa ja hidasta. (Blender Reference Manual.) Mikäli animaattori hakee realistista liikettä, on eräs hyödyllinen taktiikka animaation helpottamiseksi tiettyjen nivelien liikkumisen rajaaminen. Esimerkiksi oikealla ihmiselläkään kaikki nivelet eivät pyöri vapaasti joka suuntaan, joten luonnollisesti tämä on otettava myös 3D-animaatiossa huomioon. (Slick 2014.)

Kasvoanimaatioon käytetään usein morph target -työkalua, jossa 3D-mallia kopioidaan ja kopioiden kasvojen polygoniverkkoa liikutellaan, ja jokainen muutos tallennetaan erillisiksi targeteiksi eli kohteiksi. Siten animaattori voi säätää tallennettuja muutoksia ja liikkua vapaasti niiden välillä ja näin saada aikaan ilmeitä. (Murdock 2012, 608.)

Se, miten hahmoja lopulta liikutellaan, riippuu jonkin verran ohjelmasta. Periaatteet ovat kuitenkin samat. Tapoja on kaksi: keyframe- eli avainruutuanimaatio ja liikekaappaus (motion capture). Avainruutuanimaatiossa 3D-mallille tehdyt tärkeimmät muutokset liikkeessä, muodossa ja nopeudessa tallennetaan niin kutsutuiksi avainruuduiksi, jonka jälkeen 3D-ohjelma täyttää tallennettujen avainkuvien välissä olevat tyhjät paikat liikkeellä (kuva 18). Tämän jälkeen avainruutuja voi luoda lisää muokkaamalla mallia haluamissaan kohdissa. Liikekaappauksessa näyttelijälle puetaan sensorein päällystetty puku,

jota useat kamerat seuraavat ja tallentavat liikkeen. Tämä liike sitten liitetään hahmomallin riggiin ja näin saadaan hahmo liikkumaan. (Grey 2016.)



Kuva 18. Oranssilla värjättyt hahmot on aseteltu avainruuduiksi, jotka sitten 3D-ohjelma automaattisesti yhdistää väliruuduilla (siniset hahmot). (IEEEEXPLORE 2009.)

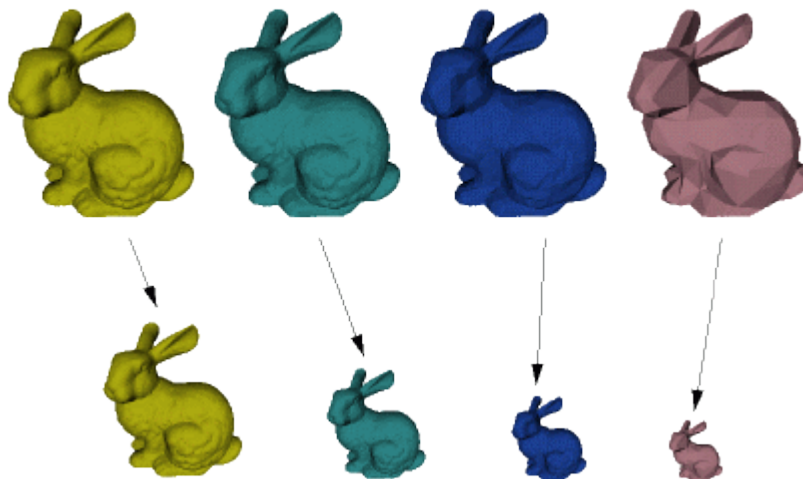
Lopuksi 3D-mallintajan on aina huomioitava, että 3D-ohjelmia on useita erilaisia, ja kaikissa niissä ei ole täsmälleen samoja työkaluja tai ominaisuuksia. Vaikka selostetut 3D:n perusteet ovat hyvin yleismaallisia, on aina olemassa se mahdollisuus, että jokin elementti ei vastaa kuvausta täysin jossain tietyssä 3D-ohjelmassa. Jokainen 3D-tuotannon yksittäinen aihepiiri (mallinnus, teksturointi ja animointi) on valtavan laaja ja vaatii jo yksinäänkin useiden vuosien aktiivista harjoittelua, jotta tekniikan voi todella hallita täysin.

### 3.2 3D-mallinnus peleissä

Kun puhutaan puhtaasti 3D-peleistä ja niiden mallinnuksesta verrattuna muiden medioiden mallinnusvaatimuksiin, ovat metodit ja työskentelytavat varsin lähellä toisiaan, kuten Masters (2014) toteaa blogissaan. Tästä huolimatta eroja löytyy. Esimerkiksi elokuvien 3D-mallinnukseen verrattuna peleissä on huolehdittava jatkuvasti polygonirajoituksista. Elokuvissa polygonimäärät saavat olla äärimmäisen korkeita, sillä elokuvat esirenderöidään, kun taas peleissä lukemattomien assettien renderöinti tapahtuu tosiajassa. Toisin sanoen kaikki, mitä pelissä tapahtuu, rakennetaan pelaajan silmien eteen tapahtumahetkellä, kun taas elokuvissa osat on kasattu kokoon etukäteen (Silverman 2013). Peleissäkin käytetään esirenderöityjä 3D-animaatioita, mutta yleensä vain välianimaatioissa, joka edistävät tarinaa ja joiden etenemiseen pelaaja ei voi vaikuttaa (Holmes 2012, 67). (Masters 2014.) Polygonirajoitukset vaihtelevat myös riippuen käytetystä pelimoottorista ja konsolista (Kennedy 2013, 55).

Koska lopullisessa pelissä tulee olemaan paljon muitakin toimintoja, kuin vain 3D-asettien tosiaikainen renderöinti, (kuten käyttöliittymä, äänitiedostot ja fyysikat), on ensisijaisen tärkeää, että polygonimääriä säädellään tarkkaan. Jos peli ylikuormittuu, putoaa sen frame rate (se, kuinka monta kuvaruutua sekunnissa näkyy), mikä taas käytännössä hidastaa peliä ja pahimmassa tapauksessa tekee siitä käyttökelvottoman. (Ahearn 2008, 103.)

Koska peleissä mallien polygonimäärille on tiukat rajoitukset, on kehitetty erilaisia tapoja säästää muistia ja nopeuttaa renderöintiä. Yksi näistä tavoista on normal-karttojen hyödyntäminen, sillä ne luovat illuusion yksityiskohdista (ks.luku 3.1.3). Toinen tekniikka, jota hyödynnetään pelituotannossa, on niin kutsuttu levels of detail (LOD) eli yksityiskohtatasot (F 2013). Tämä tarkoittaa sitä, että peliaseteista on tehty useita eri polygonitason versioita. Mitä lähempänä asettia pelaaja tai kamera on, sitä yksityiskohtaisempi asetti latautuu. Tämä tarkoittaa sitä, että yksinkertaisempi versio latautuu, kun kamera on kaukana ja monimutkainen versio kameran ollessa lähellä (kuva 19). Tämä pätee sekä pelihahmoihin että muihin peleissä oleviin asetteihin, kuten rakennuksiin tai puihin. (Newman 2016.)



Kuva 19. Mitä matalampi polygoninen kanimalli on, sitä kauempana se näytetään, jolloin tuskin huomaa mallin olevan yksinkertaisempi (Krus 2016).

Joissain tapauksissa käytetään tekniikkaa, jossa pelaajalta näkymättömiin jäävät polygonit poistetaan mallintajan toimesta ja pelimoottori tarpeen mukaan käyttää oikeaa mallia. Esimerkiksi FPS (ensimmäisen persoona ammunta) -peleissä pelaajan käsissä oleva asemalli on monimutkaisempi, kuin ne aseet, joita pelaajan näkemät viholliset kantavat. Lisäksi pelaajan aseesta voidaan poistaa perä, jos se ei muutenkaan näkyisi pelaajalle. Tällaiset objektit tosin

eivät voi olla sellaisia, joita pelaaja voisi pudottaa pelissä maahan mallin ulko-muodollisten poikkeavuuksien vuoksi. (Silverman 2013.)

3D-peleissä mallit luodaan, UV-kartoitetaan, teksturoidaan ja lopuksi animoi-daan tarpeen tullen. Se, miten näitä prosesseja lähestytään ja mitä työmene-telmiä ja ohjelmia käytetään, riippuu jonkin verran mallintajan tottumuksista ja pelitalon toimintatavoista sekä varakkuudesta. 3D-ohjelmia käytetään lukuisia erilaisia. Muiden muassa Zbrush ja Mudbox ovat suosittuja yksityiskohtaisten mallien teossa ja tekstuurimaalaamisessa, kun taas 3ds Max käy hyvin esi-merkiksi matalapolygonisten mallien tekoon ja Maya taas toimii miellyttävästi 3D-mallin animoinnissa. Vaikka kyseiset ohjelmat olisivatkin optimaalisempia joihinkin tiettyihin tarkoituksiin, ei niitä käytetä eksklusiivisesti yhteen tehtä-vään, sillä ne toimivat myös ristiin. Nykyään myös ilmaiset 3D-ohjelmat vetä-vät vertoja maksullisille vastineilleen. Tästä esimerkkinä Blender, jossa on laaja valikoima eri työkaluja eri tarkoituksiin. (Helps 2015.)

### 3.3 3D- ja 2D -assetien ja animaation eroavaisuudet ja yhdistäminen

Kaikessa yksinkertaisuudessaan 3D:n ja 2D:n ero on siinä, että 3D:llä on le-veyden ja korkeuden lisäksi syvyysulottuvuus. Sama pätee asetteihin ja ani-maatioon; 2D-assetit ovat kaksiulotteisia, esimerkiksi valokuvia, piirroksia tai maalauksia. Spritet ovat myös 2D-kuvatiedostoja (ks. luku 2.2). 3D taas tar-koittaa kolmiulotteista, joten tietokoneella luodut 3D-mallit ja maailma, jossa elämme, ovat kolmiulotteisia. (Weinberg 2016.)

Animointitapojen suhteen eroja löytyy jonkin verran. 3D-animoinnissa työpro-sessi on suhteellisen suoraviivainen. Mallille joko luodaan itse riggi eli luut, joita liikutella, tai vaihtoehtoisesti käytetään 3D-ohjelmien tarjoamia valmiita luusysteemeitä. Kun luut ovat paikoillaan, hahmo animoidaan avainruu-tuanimaatiota tai liikekaappausta käyttämällä (ks. luku 3.1.4). Avainruu-tuanimaatio on tarkkaa mutta hidasta, kun taas liikekaappaus suhteessa no-peampaa. Haittapuolena liikekaappauksessa on sen hintavuus, sekä se, että kaapattu liike vaatii usein paljon pieniä korjauksia. (Gray 2016.) Hyvä puoli 3D-animaatiossa yleisesti on se, että kun hahmo kerran luodaan, sitä voidaan uusiokäyttää halutusti kuin nukkea tai näyttelijää (Meroz 2016).

2D omaa 3D:ta useampia animointitapoja näistä kahdesta vanhempana tekniikkana. Perinteistä avainruutuanimaatiota käytetään myös 2D-animaatiossa, mutta toisin kuin 3D:n kanssa, 2D:ssä animaattorin täytyy paitsi piirtää avainkehukset, myös piirtää ja täyttää väliruudutkin itse. Tämä tekee 2D-avainruutuanimaatiosta jossain määrin työläämpää kuin mitä vastaava on 3D:ssä, vaikkakin nykyään tietokoneohjelmat helpottavat tätäkin metodia.

Kaksi muuta usein käytettyä 2D-animaatiotapaa ovat pala-animaatio ja rotoskooppaus. Pala-animaatiossa kaikki animoitavat elementit luodaan erillisistä palasista, kuten nimestäkin voi arvata (kuva 20). Näitä paloja liikuttelemalla saadaan aikaan animaatio, mutta toisin kuin avainruutuanimaatiossa, pala-animaation kuvaruutujen luonnissa liikutaan ensimmäisestä ruudusta toiseen ja kolmanteen ja niin edespäin, kunnes päädytään viimeiseen. Tässä tekniikassa ei siis ensiksi tehdä ohjaavia avainruutuja animaation pääkohdista. Samaan tapaan kuin 3D:ssä, pala-animaatiossa luotuja asetteja voidaan uudelleenkäyttää, mikä helpottaa prosessia. (Sanders 2014.)



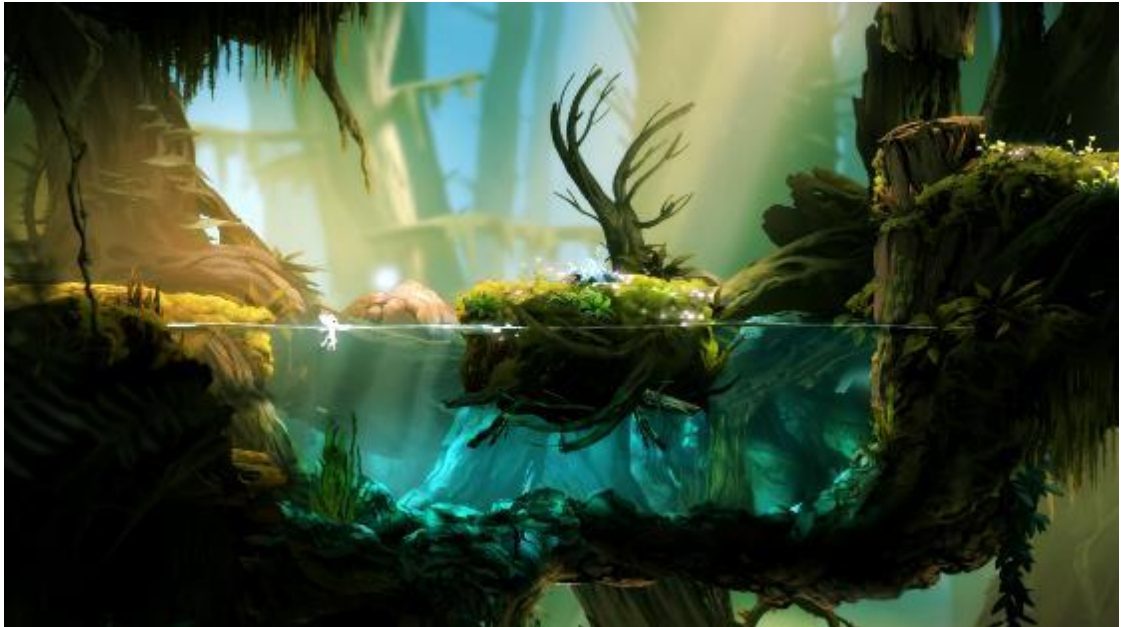
Abigail Archer, Grossology © Nelvana Limited, Corus® Entertainment Inc.

Kuva 20. Kuvassa näkyvät pala-animaatiohahmon palaset (toonboom 2016).

Rotoskooppauksessa animaation liikkeet luodaan kopioimalla ne läpi jo valmiista materiaalista, kuten videokuvasta tai toisesta animaatioista, ruutu kerrallaan. Tämä prosessi on kohtalaisen hidas, mutta sekin on helpottunut tietokonetöskentelyn ansiosta. (Fairley 2014.)

3D:n ja 2D:n eroavaisuuksiin voidaan laskea myös kuvaruutumäärä. Yleinen kuvaruutumäärä videoissa ja animaatioissa on 24 kuvaa sekunnissa. 2D-animaatioissa voi kuitenkin luistaa tästä määrästä jättämällä esimerkiksi joka toisen ruudun välistä ja vielä hitaammassa liikkeessä jopa useamman animaation kärsimättä. 3D:n kanssa vastaava ei toimi, sillä muutos on liian selkeä ja vaikuttaa liikaa animaation sulavuuteen. (Meroz 2016.)

2.5D- tai pseudo-3D -termejä käytetään usein pelialalla kuvailemaan pelejä, jotka yhdistävät 2D:ä ja 3D:ä tietyllä tavalla. Tällaisia pelejä ovat 2D-pelit, jotka on tehty näyttämään kolmiulotteisilta, esimerkiksi Ori and the Blind Forest (kuva 21). Toinen esimerkki ovat 3D-pelit, jotka on kuvattu tietystä pysyvistä 2D-asemasta kuten Trine 2 (kuva 22).



Kuva 21. Ori and the Blind Forest on 2D side-scroller -peli, joka muistuttaa ulkonäöltään ja toiminnaltaan 3D peliä. Siksi sitä voi kutsuta 2.5D-peliksi (Makuch 2015).



Kuva 22. Trine on 3D side-scroller –peli, jossa kamera seuraa pelihahmoa pysyvästi sivusuunnassa, eikä hahmo pysty liikkumaan syvyysuunnassa. Liikkuminen onnistuu ainoastaan 2D-tasolla eteen-, taakse-, ylös- ja alaspäin. Tätäkin peliä voi siis kutsua 2.5D-peliksi. (Steighner 2011.)

Yleensä, jos 3D:tä ja 2D:tä yhdistetään peleissä, se tapahtuu 3D-moottorissa. Tämä tarkoittaa sitä, että vaikka tarkoituksena on luoda 2D-peli, pelin osaset kootaan niiden kaksiulotteisuudesta huolimatta niin sanotusti 3D-ympäristöön. Tämä ei kuitenkaan tee pelistä 3D-peliä. Hyötyinä voi nähdä sen, että 3D-moottorissa tehtyyn 2D peliin voi lisätä myös kolmiulotteisia elementtejä, toisin kuin 2D-moottorissa (thedaian 2011). Jos 3D-objektia haluaa hyödyntää 2D-pelimoottorin kautta pelissä, on objekti ensin muunnettava kaksiulotteiseen muotoon, eli renderöitävä kuviksi. Tällöin 3D ei enää teknisesti ole kolmiulotteista, mutta muistuttaa sitä ulkoisesti.

Peleissä on jo kauan yhdistetty 2D:tä ja 3D:tä, eikä pelkästään 3D-mallien 2D-tekstuurien muodossa. Vanhemmissa 3D-peleissä näkee useasti osittain tai kokonaan 2D-formaattiin esirenderöityjä ympäristöjä ja taustoja. Esimerkkejä tästä näkee Resident Evil 3: Nemesis ja Final Fantasy 8 -peleissä (Kuva 23 a ja b). 3D-pelien käyttöliittymät ovat myös melkein aina 2D-grafiikalla tuotettuja.



Kuvat 23 a ja b. Kuvassa a vasemmalla on Resident evil 3: Nemesis esirenderöity ympäristö ja kuvassa b oikealla on pelistä Final Fantasy 8 esirenderöity maisema, jossa näkyy 3D-peelaajahahmoja mukana (Vas 2013).

Monissa peleissä pelimaailma on 3D:tä ja pelihahmo 2D:nä, kuten Pokemon Black & White -peleissä (kuva 24 a). Toisissa peleissä tämä on päinvastoin. Esimerkiksi Bastion-pelissä (kuva 24 b) pelin hahmo on animoitu 3D:nä ja maailma on kuvattu isometrisesti 2D:nä. Isometrisessä näkymässä 2D- tai 3D-objektista näkyy kolme sivua, mutta ilman luonnollista perspektiivivääristymää (total fabrication 2016).



Kuvat 24 a ja b. Vasemmalla Pokemon Black & Whiten kaksiulotteinen pelihahmo kolmiulotteisessa ympäristössä (Nintendolife). Oikealla Bastion-peli 3D-hahmoineen ja 2D-isometrisine maailmoineen (Supergiant Games)

3D:tä ja 2D:tä käytetään yhdessä muuallakin kuin peleissä, muun muassa elokuvissa, animaatioisarjoissa, mainoksissa ja arkkitehtuurissa. Nimenomaan elokuvissa ja animaatioisarjoissa yhdistellään usein 2D- ja 3D-animaatiota. 2D-animaatioissa saatetaan käyttää 3D-taustoja ja -efektejä (toonboom 2016).

### 3.4 3D:n ja 2D:n yhdistämisen perusteita

Syyt 3D:n ja 2D:n yhdistämiseen voidaan jakaa teknisiin ja tyyllisiin. Teknologian kehityksestä huolimatta pelit kohtaavat yhä useita rajoitteita toiminnassaan. Vaikka modernit konsolit ja tietokoneet kykenevätkin lukemaan sujuvasti tuhansia kertoja raskaampia pelejä, kuin esimerkiksi kymmenen vuotta sitten, on pelien yksittäisten elementtien luomisessa silti hyvä olla säästeliäs. Pelistä halutaan aina tehdä mahdollisimman kevyt mahdollisten ongelmien minimoiseksi sen saavuttaessa pelaajien kädet. Tästä syystä esimerkiksi esirenderöityjä taustoja käytetään nykyäänkin peleissä sen vuoksi, että ne säästävät pelin tehoja.

Toinen syy tekniikoiden yhdistämiselle on esteettisyys. Yksi yleisimmistä tavoista käyttää näitä kahta graafista tyyppiä sekaisin on nimenomaan tekstuurien hyödyntäminen 3D-malleissa. Tätä tehdään sekä teknisistä että tyyllisistä syistä. Ilman 2D-tekstuureita 3D-malli olisi ikään kuin harmaansävyinen patsas, minkä vuoksi on tarpeellista lisätä siihen tekstuurit. Lisäksi tekstureilla voi saada aikaan myös erilaisia visuaalisia efektejä, mikä puolestaan situoutuu osin tekniseen osapuoleen. Monipuolinen teksturointi saattaa hyvässä tapauksessa tarkoittaa sitä, että itse mallissa voidaan säästää polygoneissa, mikä puolestaan vähentää pelin kuormitusta.

Nykyään, kun teknologia kehittyy, voi eri grafiikkatyyppejä yhä vapaammin yhdistellä mielensä mukaan. Monet modernit 3D:n ja 2D:n sekoitukset on tehty puhtaasti, jotta saavutettaisiin toivottu visuaalinen lopputulos. 2D-elokuvissa 3D-partikkeliefektejä käytetään siksi, että se helpottaa työntekoa huomattavasti. Pelien käyttöliittymät toimivat 2D:nä ulkomuodoltaan hyvin ja niitä on helppo rakentaa kyseisellä tavalla. Tämä tarkoittaa käytännössä sitä, että 3D:n ja 2D:n sekoittaminen ei ole vanhentunut tai työläs konsepti, vaan sen mahdollisuudet ja perustelut laajenevat jatkuvasti teknologian kehittyessä.

## 4 KÄYTÄNTÖ

Tässä kappaleessa selvitan työn käytännön puolta ja sitä, kuinka työskentelin selvittäessäni 3D-mallinnuksen mahdollisuuksia MZD-projektin kannalta.

## 4.1 Lähtökohdat

MZD-projekti on vielä konseptisuunnittelu- ja teoriatasolla. Kuultuani projektista toivoin kuitenkin, että voisin osallistua jotenkin ja auttaa pelin luomisessa. Koska olin tutustunut jossain määrin 3D-mallintamiseen ennen projektin alkua, pohdin, voisiko tätä tekniikkaa jotenkin käyttää hyväksi. On huomioitava, että minulla ei todella ole merkittävää kokemusta 3D-mallintamisesta tai animaatioista. Kiinnostuksen riittäessä tämä ei mielestäni kuitenkaan ole este. Vaikka hahmojen lopullista ulkonäköä tai pelin maailman ilmettä ei oltu lyöty lukkoon aloittaessani työskentelyn, tämä ei estänyt minua aloittamasta testauksia minulle tutuimmassa 3D-ohjelmassa eli 3ds Maxissa. Sen tiesin, että peli olisi muun muassa animetyyliä parodioiva 2D-side scroller ja että jos 3D-malleja käytettäisiin, niihin lisättäisiin vielä jälkikäteen efektejä toisessa ohjelmassa, kuten Photoshopissa.

## 4.2 3D-mallin hyödyt projektissa

Ajattelin, että 3D:n käytöstä aiheutuisi useampia hyötyjä. Ensimmäisenä tuli mieleen se, kuinka sulavaa ja kaunista animaatiota olisi mahdollista saada aikaan kohtalaisen pienellä vaivalla, sillä ainakin 3ds Maxissa on valmiita animaatiotyökaluja, jotka helpottavat animointia todella paljon. En kuitenkaan ollut varma siitä, miten animointi toimisi jossain toisessa ohjelmassa.

Koska peli tehdään 2D-moottorilla, ei minun tarvitse huolehtia mallin viemisestä 3D-moottoriin. Tämän vuoksi malli olisi myös mahdollista tehdä useammasta osasta, vaikka esimerkiksi 3D-peleissä on useimmiten toivottavaa, että hahmomallit ovat saumattomia kokonaisuuksia. Tapauksessa, jossa 3D-moottoria käytettäisiin, joutuisin varmistamaan kaikkien tekstuurien ja animaatioiden sujuvan siirtymisen mallin mukana. Ongelmia olisi todennäköisesti syntynyt, sillä kaikki 3D-moottorit eivät tue esimerkiksi tiettyjä animointitapoja, joita 3D-ohjelmat tarjoavat. Se, mistä minun tulee huolehtia, on 3D-mallin sujuva renderöinti jälkimuutoksia ja 2D-moottoria varten.

Itse mallin tekemisen ei pitäisi ole turhan aikaa vievää, kun puhutaan 2D-side-scrollerista, sillä resoluutio ei tule olemaan valtavan suuri. Tämän vuoksi 3D-mallista ei tarvitse tehdä kovin monimutkaista, ellei sitten itse niin halua. Kui-

tenkin ainakin renderöintiäika nousee, mikäli mallilla on tuhattoman paljon polygoneja. Renderöinnin jälkeen hahmomalli ei enää ole 3D:tä, mutta visuaalisesti muistuttaa sitä. Lopullisen animaation kuvaruutujen määrästä ei myöskään tarvitse huolehtia, sillä Construct 2 pelimoottori jaksaa pyörittää isompia-kin sprite sheetejä.

### 4.3 Työvaiheiden kuvaus

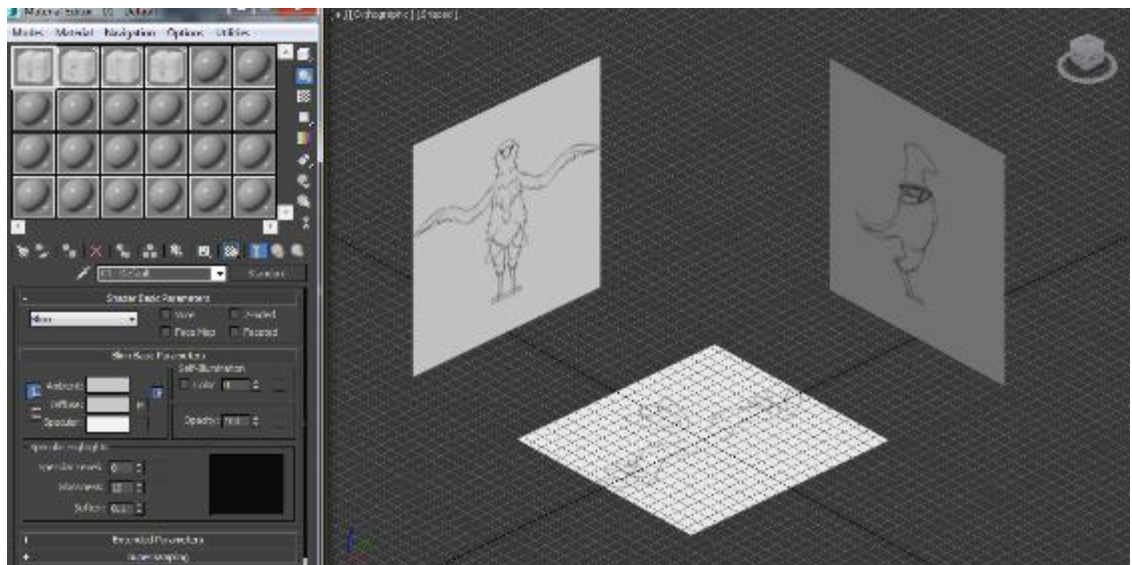
Aloitin prosessin tekemällä taustapiirroksia hahmosta, jonka valitsin testimalliksi. Ohjelmaksi otin 3ds Maxin, sillä se on minulle tutuin 3D-ohjelma. Lisäksi tiedän sen sisältävän paljon valmiita mallinnus- ja animaatiotyökaluja, joita voisi hyödyntää tehokkaamman työskentelyn edesauttamiseksi.

Hahmo, josta tuli testeri tässä projektissa, on pieni korppimainen olento, jonka avulla selvittelin mahdollisuuksiani ohjelman rajoissa. Syy, miksi tein korpin, oli puhtaasti se, ettei pelin hahmojen lopullista design-tyyliä ollut suunniteltu ajatustasoa pidemmälle. Lisäksi peliin on tarkoitus tulla myös vihollishahmoja sekä eräänlaisia maskottiolentoja, joten esimerkiksi tekemääni riggia voisi olla mahdollista hyödyntää myös lopullisessa projektissa, vaikka itse malli ei tyyliin sopisikaan.

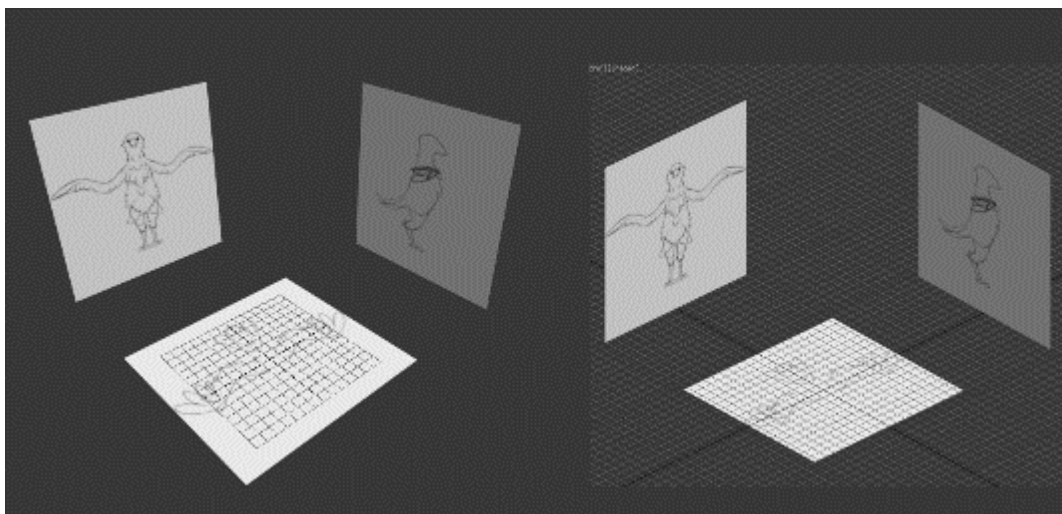
Yksi kohtaamistani hyvin tärkeistä seikoista hahmon hyvän ulkonäön takaamiseksi oli tekstuurit ja kartat. Vaikka 3ds Max on minulle tutuin 3D-ohjelma, ei minulla ole näiden ohjelmien kanssa työskentelyssä pitkää historiaa. Kokeukseni rajoittuu ennen kaikkea opintojaksoihin koulutukseni aikana sekä muutamaan itse tehtyyn malliin. Tämä tietenkin tarkoittaa sitä, että joudun työskennellessäni oppimaan yrityksen ja erehdyksen kautta, ja olen pitkälti internetin apujen varassa. Vaikka opin ja olen oppinut työskennellessäni valtavasti, huomaan kuitenkin, että minulla on vielä paljon opittavaa.

Erityisesti minulla oli ennakkohuolia teksturoinnin kannalta. Aiemmillä oppitunneilla oli käynyt ilmi, että muun muassa tekstuurit, joita tein, olivat hyvin vajavaisia ja rumia verrattuna siihen, mitä ne voisivat olla. Tästä syystä mallitkin näyttivät vielä todellista kehnommilta. Tämän uuden mallin kanssa koetin selvittää, mitä minun olisi minimissään tehtävä, jotta saisin aikaan sellaista jälkeä, jota sopisi laittaa peliin. En tässä vaiheessa tiennyt, minkälaista tyyliä pelissä yleisesti lähdetään hakemaan, joten kokeilin erilaisia vaihtoehtoja.

Ensimmäinen asia, jonka tein, oli niin kutsutun studion luominen. Tämä käytännössä tarkoittaa sitä, että toin aiemmin tekemäni mallikuvat hahmosta edestä, sivusta ja tässä tapauksessa ylhäältä kuvattuna 3ds Max -ohjelmaan. Nämä kuvat laitoin tekstuureina kolmelle luomalleni pinnalle eli planelle, jotka asettelin haluamilleni paikoille 3D-näkymässä (kuva 25). Muutin myös näkymän perspektiivisestä ortografiseksi, jota yleisesti käytetään mallintaessa. Tämä johtuu siitä, että ortografisessa näkymässä ei ole luonnollista perspektiivin illuusiota, jolloin malliinkaan ei tule odottamattomia vääristymiä (kuva 26).

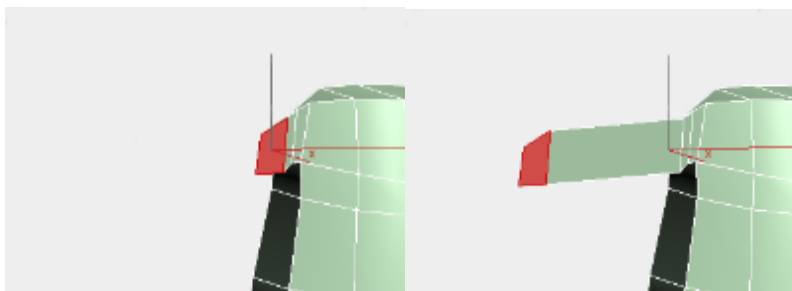


Kuva 25. Vasemmalla näkyy materiaalieditori, johon toin kuvat haluamilleni materiaalipaikoille. Nämä materiaalit siirsin kolmelle tasolle, jotka näkyvät oikealla. Kuvat on laitettu diffuse-paikalle bitmappeina, eli vain kuvina.



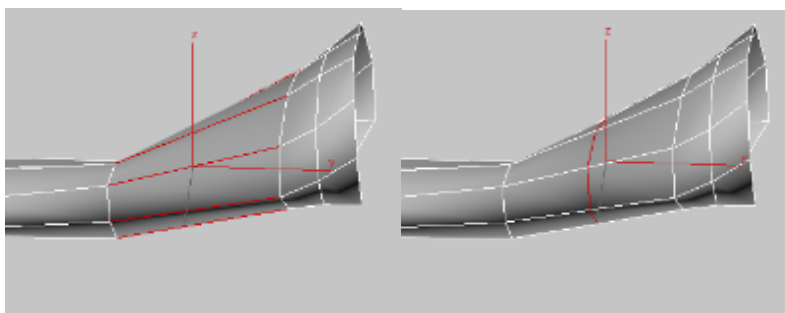
Kuva 26. Vasemmalla näkymä perspektiivissä ja oikealla ortografisesti

Ryhtyessäni luomaan mallia valitsin tekniikaksi laatikkomallinnuksen, sillä kyseinen tapa on minulle tutuin ja siten nopein tehdä. Tärkeimmät työkalut minulle tässä tekniikassa olivat ekstrude ja connect perinteisten siirto-, koonmuutos- sekä pyöritystyökalujen rinnalla. Ekstrude-työkalulla voi ikään kuin työntää uuden polygonimassan ulos valitusta alueesta (kuvat 27 a ja b). Tämä on kätevä tapa lisätä polygonimassaa halutusti.



Kuvat 27 a ja b. Kuvassa a on valittu haluttu kohta, josta polygon ekstrudataan. Oikealla kuvassa b on käytetty extrude-työkalua.

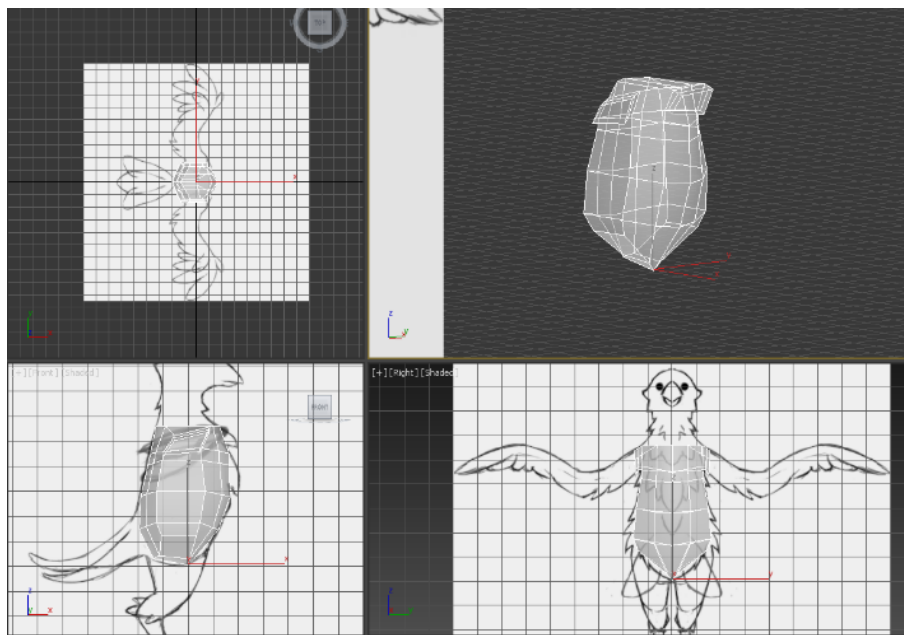
Connect-työkalulla pystyin lisäämään edge-luuppeja haluamiini kohtiin ja jälleen näin lisäämään polygonien määrää, mikä mahdollisti mallin yksityiskohdistamisen (kuvat 28 a ja b). Työskennellessäni minulla oli auki neljä 3D-näkymää. Yksi edestäpäin, yksi sivusta, yksi ylhäältä ja viimeinen oli sitä varten, että pystyin vapaasti pyörimään mallin ympärillä.



Kuvat 28 a ja b. Kuvassa a olen valinnut edge-luoppien välistä paikan, johon haluan uuden luopin ja kuvassa b olen käyttänyt connect-työkalua, joka loi uuden luopin halutulle paikalle.

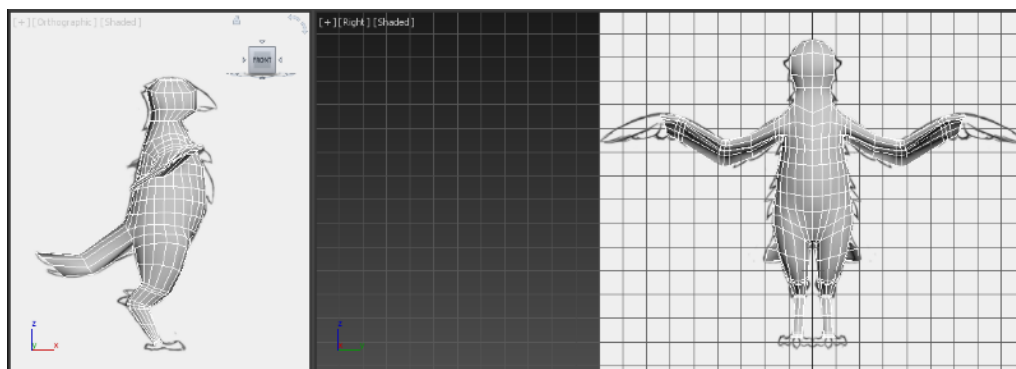
Aloitin työn kuutionmallisesta geometria-alkiosta. Jotta pystyin muokkaamaan sitä, muutin sen editable poly -muotoon. Kun ensimmäiset muodot on saatu aikaan ja rintakehän alkukantainen malli oli valmis, poistin siitä vertikaalisesti puolet ja käytin mirror-modifieria, joka peilaa työstössä olevan geometrian haluttuun suuntaan. Näin on mahdollista työstää vain toista puolta mallista ja samalla toinen puoli muokkautuu automaattisesti identtiseksi. Läpinäkyvyyttä

käytin aika ajoin, jotta pystyin näkemään taustalle tuomani mallikuvat ja työskentelemään niiden ohjeistuksella (kuva 29).



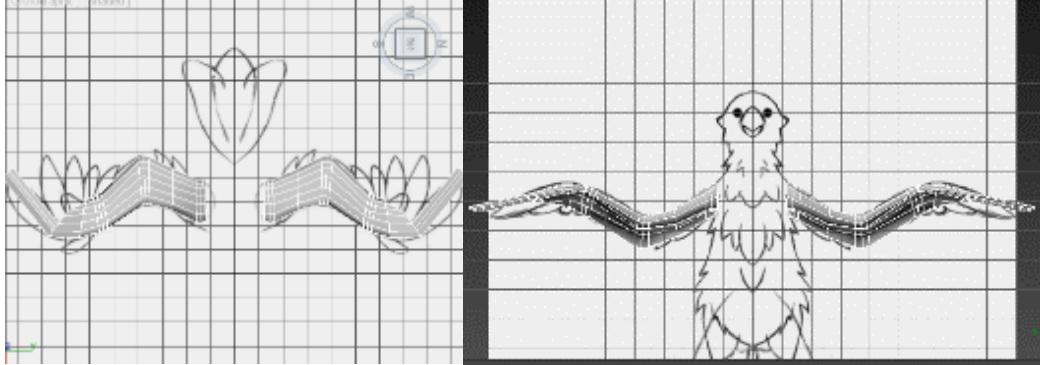
Kuva 29. Karkea rintakehä läpinäkyvänä

Tästä lähdin muovaamaan käsien ja jalkojen alkuja. Samalla myös lisäsin polygoneja jo olemassa olevaan verkkoon, jotta pystyin tekemään mallista yhä yksityiskohtaisemman. Muokkasinkin rintakehää ja vatsan aluetta oikeampaan suuntaan ja siistimmiksi niin, että muoto ja topologia olisivat paremmassa kunnossa. Tein myös hahmolle yksinkertaisen pyrstön sekä päänalun tässä vaiheessa. Käsia ja jalkoja tehdessäni minun piti muistaa, että taivealueisiin on hyvä laittaa useampia edge-luuppeja, jotta malli animoidessa taipuisi kyseisiltä kohdilta mielekkäästi (kuva 30).



Kuva 30. Hahmon malli yksityiskohtaisempana

Kun olin saanut tärkeimmät muodot aikaiseksi, aloitin käsivarsien ja kämmenien sekä sormien tarkemman kehittämisen. Olin suunnitellut lintuhahmolle siipimäiset kädet ja sormien paikalla oli sulat (kuva 31). Samalla tavalla, kuin muutkin pääpolygonimassasta ulkonevat osat, tein ne ekstruden avulla.

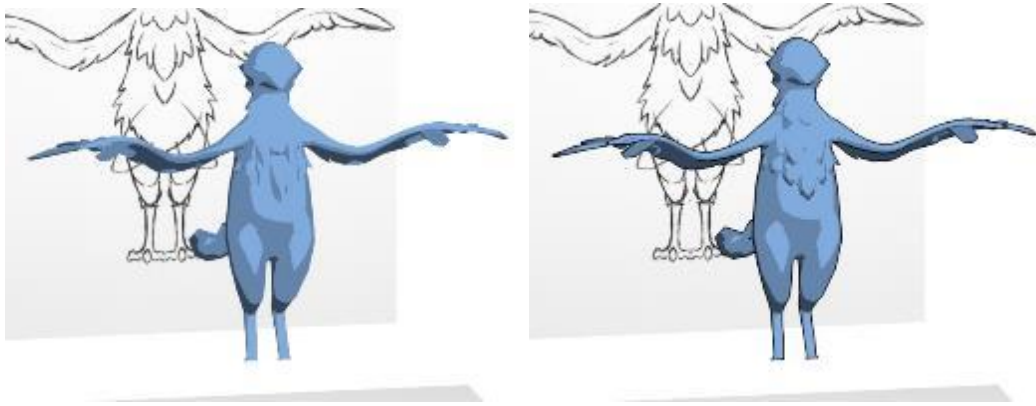


Kuva 31. Siipien kehitystä

Käsien valmistumisen jälkeen siirryin pään muokkaamiseen. Tässä vaiheessa luovuin mirror-modifierista ja liitin peilatus puolen polygoniverkkoon niin, että malli oli kiinteä kokonaisuus. Tämä minun oli tehtävä sen vuoksi, että halusin muokata ja etenkin suurentaa edge-luuppeja pään ja kaulan ympärillä. Jos mirror-modifier on päällä tällaista tehdessä, voi hahmon keskisauma aueta. Tämä ei ole toivottavaa. Työskentelin edelleenkin vain toisella hahmon puoliskolla, sillä loisin mirror modifierin uudestaan, kun sitä myöhemmin tarvitsisin. Pään saatua suurpiirteisen muotonsa, oli vuorossa joidenkin yksityiskohtien luominen. Tässä vaiheessa tein hahmolle sen rintaröyhelöt. Tämäkin tapahtui ekstrudea käyttäen, jonka jälkeen muutin niiden mallin oikeanlaiseksi. Myös pyrstö pääsi lähemmäs lopullista muotoaan. Jalkoihin lisäsin pituutta ja kaulasekä kylkihöyhenet muodostuivat.

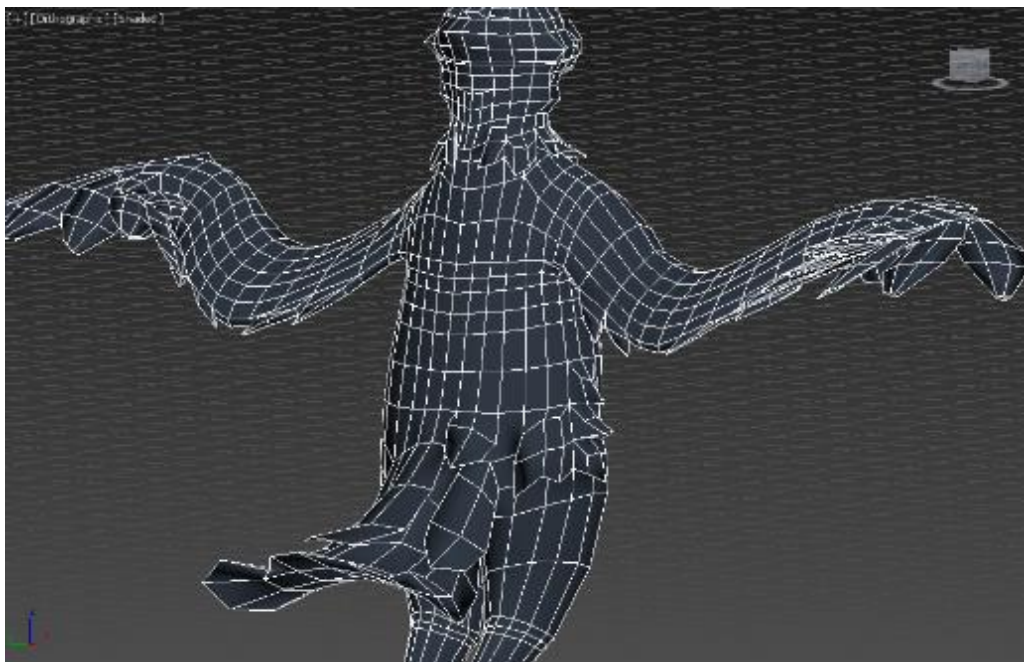
Koska olen malttamaton, koetin jo tässä vaiheessa joitain 3ds Maxin materiaaleja mallin päälle, sillä niitä varten hahmoa ei tarvitse välttämättä UV-kartoittaa. Oletin tässä vaiheessa edellisen tilauspeliprojektin vuoksi, että aseteista tulee mahdollisesti cel-varjostettuja, eli varjot ovat hyvin yksinkertaiset ja selkeärajaiset. Tovin etsinnän jälkeen löysin Ink 'n Paint -nimisen materiaalin 3ds Maxin syövereistä. Sen vuoksi, että lopullinen malli renderöidään sellaisenaan 2D-formaattiin, sain mahdollisuuden käyttää näitä valmiita materiaaleja. Jos malli oltaisiin viety 3D-moottoriin, on hyvin todennäköistä, että kyseinen mate-

riaali olisi täytynyt uudelleenrakentaa täysin moottorin sisällä. Tässä materiaalissa pystyy paitsi valitsemaan varjon laajuutta, myös kuinka monta tasoa varjoon tulee. Tarpeen mukaan löytyy myös mahdollisuus lisätä niin sanotut musterajat (kuvat 32 a ja b). Sen lisäksi, että tällaista materiaalia käyttämällä voitaisiin säästää aikaa, on se myös sen verran yksinkertainen, että mahdollisuudet hahmoanimaation visuaaliseen jälkimuokkaukseen ovat hyvät.



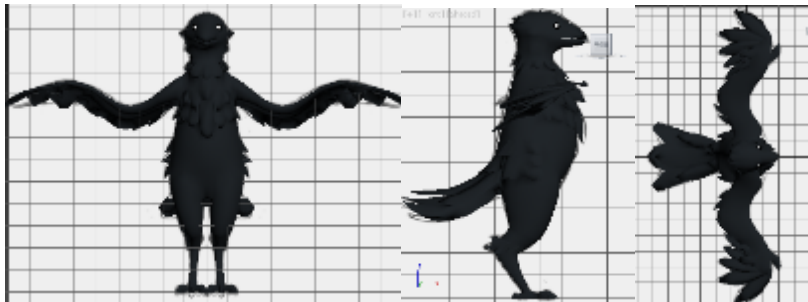
Kuvat 32 a ja b. Kuvassa a näkyy pelkkä kaksitasoinen varjostus ja kuvassa b on lisäksi vielä musterajat.

Kokeilujeni jälkeen mallinsin jalkaterät ja varpaat sekä lisäsin sulkia pyrstöön (kuva 33). Hahmon silmät tein erillisiksi elementeiksi sen vuoksi, että malli on itsessään kohtalaisen matalapolygoninen. Ajattelin silmien teon olevan hankalaa suoraan malliin ilman, että päähän silmien alueelle olisi tullut muuta polygoniverkkoa tiheämpi alue. Silmille annoin alustavasti kiiltävän materiaalin.



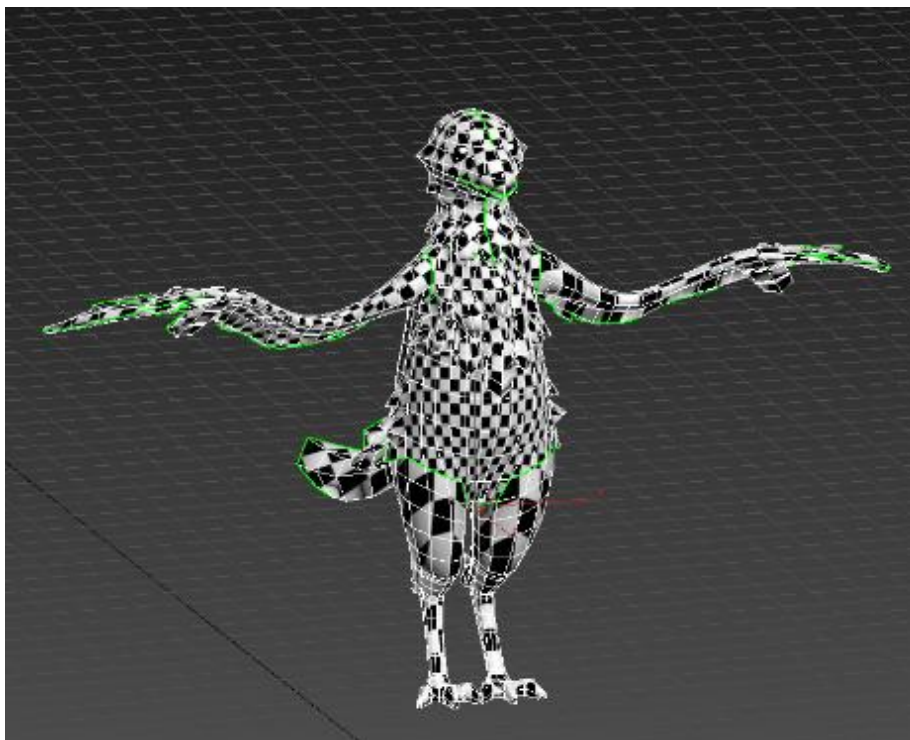
Kuva 33. Päälimmäiset pyrstösulat ekstrudattiin olemassaolevasta geometriasta.

Alunperin en ajatellut, että tekisin korpille aukeavaa suuta, mutta muutin mielteni lopulta. Suun ei ollut tarpeellista aueta, mutta en ajatellut siitä tulevan suuria ongelmia myöhemmin. Ennen kuin poistin muokkaamatta jättämäni puoliskon mallista, jotta voisin laittaa hahmolle jälleen peilauksen päälle, lisäsin vielä yksityiskohtia linnun jalkaan. Sen jälkeen kun peilasin mallin ja kiinnitin peilauksen mallin toiseen puolikkaaseen, hahmo oli jotakuinkin valmis (kuva 34) ja saatoinkin aloittaa UV-kartoituksen.

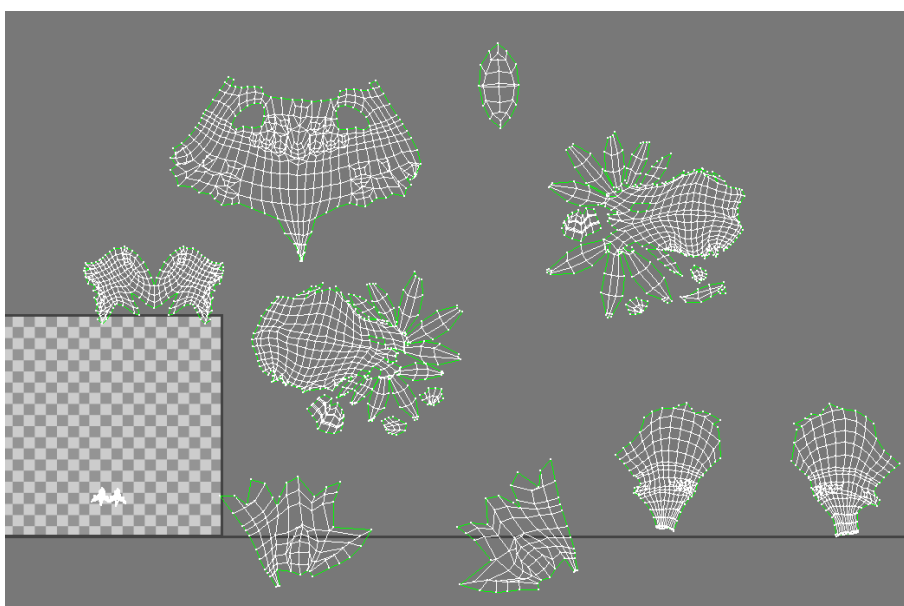


Kuva 34. Valmis malli edestä, sivusta ja ylhäältä

Jotta pystyin aloittamaan UV-kartoituksen, oli minun valittava modifiontilistalta Unwrap-UVW -editointityökalun. Kun se oli valittu, pystyin aloittamaan polygoniverkon saumakohtien merkitsemisen. Saumapaikat valitaan yleensä sen mukaan, miten ne on helppo piilottaa ja korjata tarpeen vaatiessa. Tässä tapauksessa saumakohtien paikoilla ei ole niin paljon väliä, sillä tekstuuri tulee olemaan pitkälti yksivärinen ja mahdollisimman yksinkertainen. Tällöin saumoja ei välttämättä edes näe, vaikka niitä olisikin mallin pinnalla. Pysin kuitenkin pitämään saumat älykkäissä paikoissa, kuten käden takareunassa ja kaulahöyhenten alla. Kaikki raajat erotellaan yleensä torsosta ja pääkin erotellaan. Minun tapauksessani erottelin myös pyrstön ja suun sisäosan, sekä joitain yksittäisiä sulkia. Tässä vaiheessa laitoin korppihahmolle checker- eli ruutukuvion, jotta paremmin näkisin kuinka tasaisesti kartat asettuvat (kuva 35). Kun saumapaikat oli valittu, vahvistettiin saumat ja pystyin aloittamaan karttasioiden erottelun. Tämän jälkeen valitsin kaikki polygonit yhdestä saumoilla erotelluista osioista ja painoin quick peel -työkalua, joka irrottaa kyseisen alueen nopeasti ja helposti muista pinnoista. Kun tämä on tehty pystyin muokkaamaan ja asettelemaan irrotettua karttapalasta UV-editorissa, kunnes ruutukuvio näytti hyvältä mallin päällä (kuva 36). Toistin tämän sarjan jokaisen saumoilla erotetun palasen kohdalla, kunnes kaikki oli irrotettu ja siistitty.

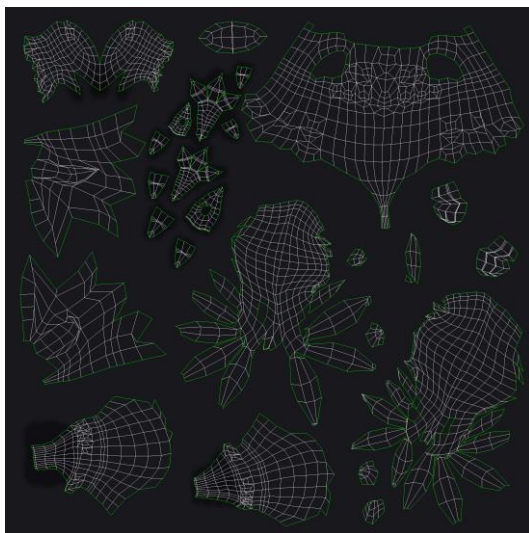


Kuva 35. Jalkoja, pyrstöä ja vasenta kättä ei ole vielä kartoitettu. Tämän näkee ruutukuvion epätasaisuudesta.



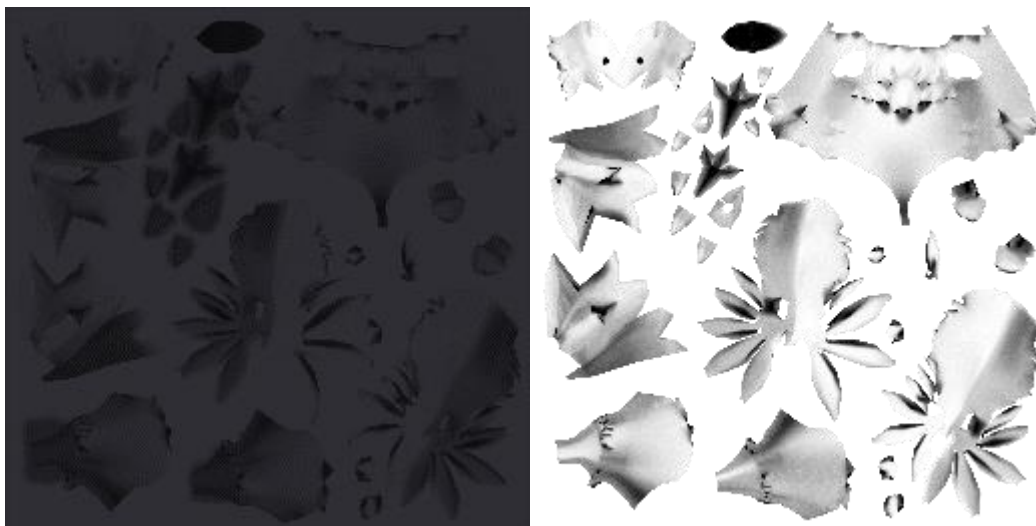
Kuva 36. Kuva on kaapattu UV-editorista. Kaikki muut osat paitsi jalkaterät on kartoitettu. Lopulta kaikki osaset asetellaan vasemmalla olevan pienen ruudun sisään.

Kun olin tehnyt UV-kartoituksen, pystyin viemään UV-karttamallin Photoshopiin ja siellä loin värin tulevalle diffuse- eli värikartalle. UV-kartoituksen avulla pystyin ensinnäkin näkemään, mihin kohtaan minun pitäisi laittaa vaikka tummempaa väriä, jos haluaisin hahmon nokasta tummemman (kuva 37). Toiseksi se tästä lähtien määrittäisi paikat, joihin värit menevät mallin päällä.



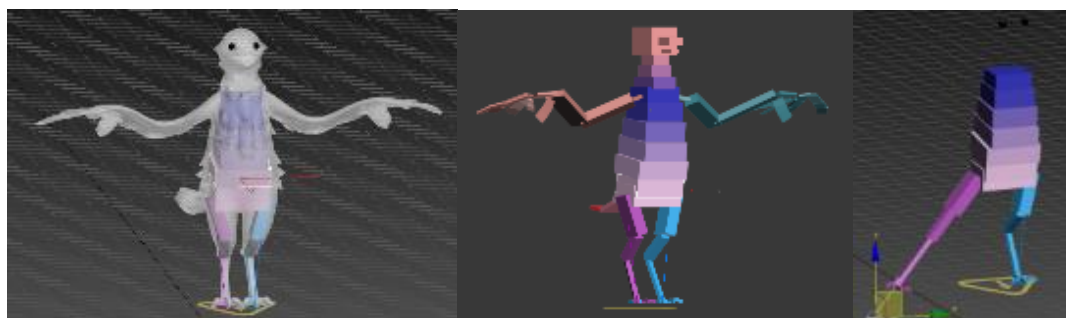
Kuva 37. Värikartta, jossa päällä näkyy UV-kartoitus.

Kun olin tehnyt yksinkertaisen värikartan, halusin, että malli saisi vähän yksityiskohtia. Selvitettyäni mikä voisi toimia törmäsin ambient occlusion -karttaan ja ajattelin sen olevan hyvä vaihtoehto, sillä se simuloi neutraalia kevyttä valoa (kuva 38). Tämä tarkoitti sitä, että minun pitäisi ”leipoa” eli renderöidä valaistus tekstuuriksi. Ensimmäinen asia, jota minun täytyi tehdä, oli muuttaa 3ds Maxissa renderöintisysteemin Mental Rayksi. Seuraavaksi valitsin tekstuurirenderöinti-valikosta ambient occlusion -vaihtoehdon ja nostin siellä myös sample määrän, jotta lopputuloksesta ei tulisi rakeinen. (Corijn 2016.) Tämä luotu kartta asettui UV-kartoitukseni mukaisesti ja oli siis täysin käyttövalmis saman tien. Liitin ao-kartan suoraan aiemmin luomaani värikarttaan. Lisäksi kokeilin bump-karttaa nostamaan korpin nokan ja jalkojen pintaa, mutta en nähnyt efektiä tähdelliseksi, joten jätin sen pois.



Kuva 38. Vasemmalla ao-kartta on lisätty värikarttaan. Oikealla on pelkkä valmis ao-kartta.

Tässä vaiheessa päätin, että on aika siirtyä hahmon riggaamiseen. Siitä, että mallia ei tarvitse siirtää 3D-moottoriin, on tässäkin tapauksessa hyötyä, sillä voin käyttää 3ds Maxin CAT-riggaus systeemiä hyväkseni. Se on helpompi ja nopeampi, kuin jos joutuisin tekemään itse luut, linkkaamaan ne sekä lisäämään raajoihin FK- ja IK-työkaluja. Kaikki nämä ovat jo valmiina CAT-luissa. Toisin sanoen tällöin esimerkiksi hahmon raajat taipuvat luonnollisemmin, jos hahmon jalka-alustasta otetaan kiinni ja sitä liikutellaan. Raajojen luut eivät myöskään lähde pois paikoiltaan, vaikka niitä kiskottaisiin poispäin muusta luurangosta. Minun tehtäväni oli vain luoda tarvittavat luut. Ensimmäiseksi lisäsin CAT-alustan, josta sitten lähdin rakentamaan luurankoa lantiosta alkaen. Lisäksi tässä luusysteemissä on mahdollista luoda luut vain yhdelle jalkalle ja kädelle ja sen jälkeen peilata ne vastakkaiselle puolelle. Pidin prosessin ajan mallia läpinäkyvänä, jotta saisin luut kohdilleen (kuva 39).



Kuva 39. Vasemmalla näkyy keskeneräinen luuranko läpinäkyvän mallin sisällä, keskellä taas valmis luuranko. Viimeisessä kuvassa luurangon jalkaa kiskotaan poispäin sen vartalosta, mutta se ei silti rikkoudu kiitos FK- ja IK-työkalujen.

Kun luuranko oli valmis, siirryin skinnaukseen. Valitsin skin modifierin, jotta pystyisin aloittamaan operaation. Sen työkaluilla valitsin kaikki luomani luut ja lisäsin ne listaan, jonka avulla ohjelma liittää ne malliin. Näin malli oli siis kiinni luissa. Ongelmana oli kuitenkin luonnollisesti se, että ohjelma ei osaa tehdä työtä erityisen tarkkaan ja joku osa mallista voi olla liittyneenä väärään luuhun (kuvat 40 a ja b). Esimerkiksi mallin käden ei pitäisi olla kiinni jalkaluussa. Lisäksi vaaditaan hienosäätöä taivekohdille, jotta liikkeestä tulisi mahdollisimman luonnollista. Skinnatessa verteksit jaetaan luiden kesken tietyn asteisesti (tai painoisesti), jotta saavutetaan haluttu tasapainoinen jako.



Kuvat 40 a ja b. Kuvassa a näkee, kuinka siiven alaosassa verteksejä on kiinni väärässä luussa, mikä johtaa siiven vääristymiseen. Kuvassa b ongelma on korjattu ja verteksit on liitetty oikeaan luuhun.

Skinnaustakin helpottaa se, että mallin verteksien painopisteet on mahdollista peilata ja kopioida puolelta toiselle. Skinnauksen valmistuttua pystyin siirtämään animointiin.

Animointi on myös mukavan helppoa 3ds Maxissa, jos on käyttänyt CAT-luuserankoa. Mallille voi antaa jonkun valmiista animaatioista, joita ohjelmassa on tarjolla. Nämäkin animaatiot vaativat paljon hienosäätöä, jos niitä käyttää omassa mallissaan. Itse lisäsin perusihmiskävelysyklin hahmolleni. Näin olento lähti liikkeelle, joskin könkäisesti ja oudosti heiluen. Ajatus tässä on kuitenkin se, että hahmon liikkeitä muokataan haluttuun suuntaan. Tietysti liikkeen voi luoda itse alusta alkaen, mutta valmis liikesykli voi auttaa oman animaation työstämisessä. Kun aloitin animaation teon, loin ensimmäiseksi tason, jolle muutokset, jota CATmotion editorissa tein, tallentui. Sitte menin CATmotion editoriin, jossa on tarjolla hyvin perusteellinen joukko erilaisia liikeparametreja, joilla pystyin vaikuttamaan siihen, kuinka mikäkin luu liikkuu ja reagoi. Näitä parametreja hiomalla sain aikaan tyydyttävän kävelyanimaation korpilleni.

Lopuksi vielä reneidoin kävelyanimaation kuviksi, jotka kokosin Photoshopissa muutamaksi pieneksi esimerkkikävelyanimaatioksi (kuva 41). Kahdessa on hieman eri sävyisillä tekstuureilla varustettu korppi, jonka animaatio on esitetty ortografisesti sivulta kuvattuna, niin kuin se olisi peliin laitettu. Yhteen animaatioon laitoin kamera-ajon, jotta olisi mahdollista nähdä korppi joka suunnasta.



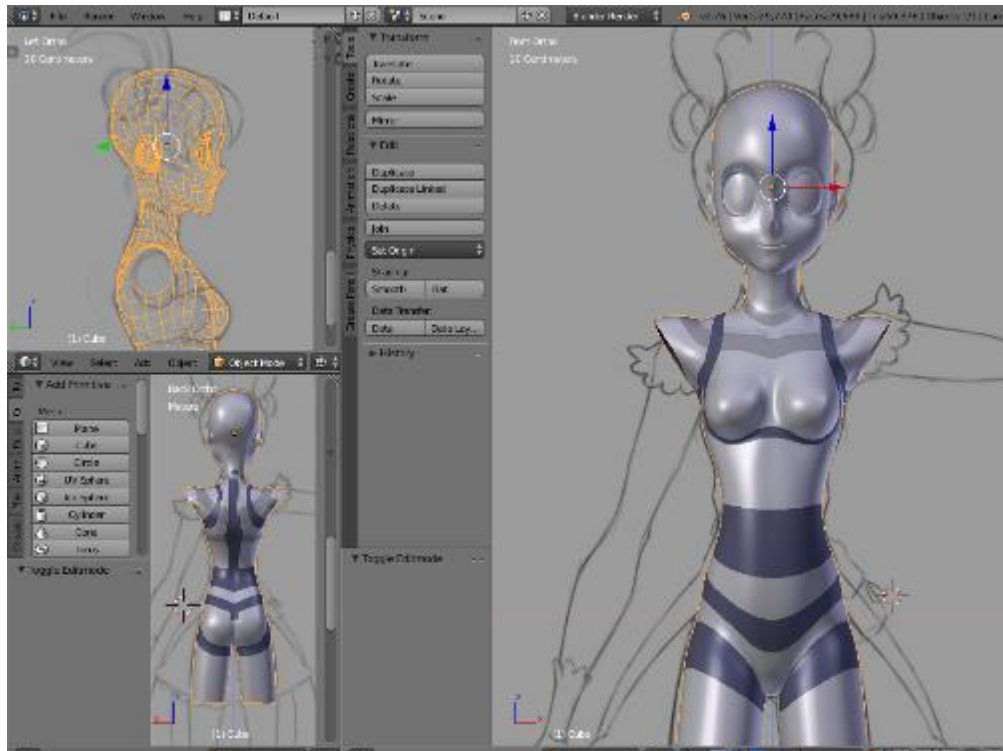
Kuva 41. Kolme kuvaa koko animaatiostakuvasarjasta

Renderöin lopulta liikeanimaatioksi vain korpin, jolla oli teksuurina värikartta, johon oli upotettu ao-kartta, mutta renderöin yksittäisiä malliesimerkkejä myös cell-varjostuksesta ilman väri- ja ao-karttaa sekä niiden kanssa (kuva 42).



Kuva 42. Vasemmalla pelkkä värikartta ao:n kanssa, keskellä pelkkä cel-varjo ja viimeisenä näiden kahden edellisen yhdistelmä

Tarkoitukseni oli tutkia toisenkin 3D-ohjelman (nimellisesti Blenderin) työkaluja, mutta valitettavasti yliarvioin nopeuteni uuden ohjelman käytön suhteen. Opin kyllä perusmallintamistyökalujen käytön ja edge-mallinnuksen perusperiaatteet ja metodit. Valitettavasti kuitenkin kunnianhimoissani harjoitusmalli, jota ryhdyin työstämään, oli liian hidastekoinen aloittelijalle. Sen loppuun saattaminen jää minulle vielä tehtäväksi (kuva 43). Lisäksi, kun ryhdyin tekemään tätä uutta mallia, olin saanut tässä vaiheessa hahmosuunnitelmia muilta ryhmän edustajilta, joiden pohjalta tein taustapiirroksia mallille. Kävi kuitenkin ilmi, että nämä suunnitelmat eivät olleet lopullisia, joten mallikaan ei tule olemaan käypä loppujen lopuksi, mutta jos ei mitään muuta niin ainakin saan harjoitusta sen avulla.



Kuva 43. Keskenräinen hahmomallini Blenderissä

#### 4.4 Ongelmia ja ratkaisuja

Vastaani tuli monenlaisia ongelmia. Ensimmäinen asia, joka aiheutti minulle harmaita hiuksia, oli mallin sormien tilalla olleet sulat. Sen lisäksi, että en ole kovin kokenut mallintaja ja normaalitkin ihmissormet tuottavat ongelmia, olivat sulat niin lähellä toisiaan ja limittäin, että niitä oli kovin hankala mallintaa. Ongelma oli kaikei juuri se, että yritän tehdä vaikeampia asioita, kuin mitä taitotasoni sallivat. Kun ruvetaan muuttamaan anatomiaa tarkoituksella vähemmän realistiseksi ja karikatyyrisemmäksi, olisi hyvä, jos hallitsisi ensin perusteet. Saman sain todeta, kun aloitin Blenderissä uuden mallin työstämisen. Koska hahmo on animettyinen, on sillä hyvin erilaiset kasvat verrattuna normaaliin ihmiseen. Niiden mallintaminen osoittautui hyvin haastavaksi. Näissä tapauksissa kuitenkin pääsin jotenkin eteenpäin vain yrittämällä, kunnes onnistuin, eli tekemällä ja korjaamalla uudestaan ja uudestaan. Sormiongelma olisi voinut ratketa myös niin, että olisin mallintanut sormet hieman kauemmaksi toisistaan haralleen. Se olisi mahdollisesti helpottanut työntekoa. Myös pään ja nokan kanssa oli jotain ongelmia sen vuoksi, että pelkään lisätä geometriaa liialti yksittäisiin kohtiin. Mitä pidemmälle olen kuitenkin edennyt mallintamisessa,

sitä enemmän olen ymmärtänyt, että kyllä geometria saa olla vähän tiheämpää alueilla, joissa yksityiskohdat ovat kaikkein tärkeimpiä. Silloin pitää vain pitää huoli mallin hyvästä topologiasta.

Yksi suurimmista päänvaivantuottajista oli ambient occlusion -kartan teko. Tapoja, joilla kartan voi tehdä, on useampia. Koska asia oli minulle täysin uusi, sekoitin niitä usein toisiinsa. Lisäksi monet tutoriaalit, joita internetistä löysin, kertoivat hyvin sekavasti, kuinka luoda tällainen kartta. Pitkien taisteluiden jälkeen sain kuitenkin aikaan kyseisen kartan ja ymmärsin paremmin, mitä tarkalleen tehdä. Yksi helppo ratkaisu, joka minulle valkeni tämän asian osalta aivan liian myöhään, oli jälleen se fakta, että malli jota rakensin, tulitisiin vieämään 2D-moottoriin. Tämä tarkoittaa sitä, että voisin luoda haluamani valon 3D-näkymään mallin kanssa, ja minun ei tarvitsisi välttämättä tehdä siitä erikseen karttaa. Renderöinhän kuitenkin lopullisen tuotoksen juuri siinä paikassa ja valon efektit renderöityisivät mukana. Tämä keino ei tosin toimi, jos halutaan esimerkiksi sekä ambient occlusion -efekti että cel-värjäys mallin pinnalle, sillä cel-värjäys vaatii kovempaa valaistusta toimiakseen. Lisäksi jos näkymässä on sekä kova valo että ambient occlusionin yleinen valaistus, ao-valaistus häivittäisi cel-värjäyksen varjot.

Tämä seikka valojen kanssa on myös ongelma itsessään. Voimakkaiden valojen ja varjojen aiheuttamista mallin pinnassa pitäisi välttää, koska 2D-moottorissa ei ole 3D-valaistusta. Tästä syystä valaistus pelimaailmassa voisi poiketa mallista huomattavasti, jos ei ole varovainen. Tämän vuoksi ambient occlusion -kartta olisikin hyvin optimaalinen ratkaisu. Jos kuitenkin halutaan käyttää cel-värjäystä, pitäisi valon olla voimakas. Tähän ratkaisuna näen mahdollisimman neutraalin valopaikoituksen, eli sen, että valonlähde laitetaan hahmon lähelle eteen ja ylös. Sama ongelma oli myös korpin silmien kanssa. Itsessään hyvin kiiltävä materiaali tarvitsee valaistuksen toimiakseen. Tämän ratkaisin niin, että laitoin pienen, staattisen kiiltopisteen mustaan tekstuuriin ja laitoin tämän silmiin. Näin näyttää siltä, kuin silmät kiiltäisivät vähän.

UV-kartan luominen malliin, johon tulee näinkin yksinkertaiset tekstuurit, ei ollut hirvittävän hankalaa, mutta kämmenet ja sormet tuottivat tässäkin ongelmia. Ratkaisu olisi myös sama, kuin aikaisemmin: jos sormet olisivat erillään, olisi tämäkin tehtävä ollut helpompi.

Yritin myös luoda kiiltoa korpin nokkaan specular-kartalla, mutta syystä tai toisesta efekti hävisi jossain vaiheessa materiaalista, vaikka se hetkellisesti oli-kin toiminut hyvin. Hohtoa ei ole kovin vaikea laittaa spesifiin paikkaan. Se vaatii vain kokomustaa karttaa, jossa kiiltävät kohdat jätetään vaaleammiksi ja sitten kartta laitetaan sille tarkoitettuun paikkaan materiaalissa. Epäilen, että kun vaihtelin renderöintisysteemeitä 3ds Maxin sisällä, joku niistä ei kyennyt tuottamaan kiiltoa lopulliseen renderiin.

Skinnauksessa lieväksi hankaluudeksi osoittautui vertekseille jaettujen alueiden tai painojen peilaaminen mallin toiselle puolelle. Aika ajoin tämä toiminto ei ottanut onnistuakseen sujuvasti, mikä hidasti työtäni runsaasti. Sitkeästi yrittämällä sain kuitenkin hyviä lopputuloksia.

Yksi suurimmista niin sanotuista ongelmista on lopulta 3ds Max itsessään. Se on helppo ja hyvä ohjelma suhteellisen nopeaan mallinnukseen ja animaatioon, mutta ilman opiskelijalisenssiä se on valtavan kallis. Ratkaisu tähän olisi käyttää ilmaista Blenderiä, mutta Blenderissä ei tosiaan ole yhtä helppoja työkaluja riggaukseen ja animointiin. Toisin sanoen verrattuna 3ds Maxiin Blenderissä työskentely voi olla huomattavasti hitaampaa.

Koetin viedä valmista 3ds Max mallia Blenderiin, mikä tämäkin osoittautui hankalaksi. Se ei kuitenkaan ollut mahdotonta, kun muutin mallin tiedostomuodon obj.-muotoon ja sain mallin siirrettyä. Samalla mallin polygonit kuitenkin jakautuivat tri-polygoneiksi, mikä taas ei ole optimaalista, mikäli hahmo halutaan animoida Blenderissä. En vielä tiedä, mikä on nopea ratkaisu tämän korjaamiseen. Polygonit voi aina manuaalisesti muuttaa takaisin quad-muotoon, mutta tämä olisi todella työlästä ja hidasta.

## 5 LOPUKSI

Työskentelyprosessini aikana minulle avartui huomattavasti parempi käsitys siitä, mitä 3D:n hyödyntäminen MZD-projektissa voisi käytännössä tarkoittaa, ja millaisia hyötyjä siitä voisi olla.

Ensinnäkin 3D-ohjelmien sisäiset materiaalit, kuten 3ds Maxin ink 'n paint, cel-varjostus ja mustemateriaali vaikuttavat todella hyviltä tavoilta antaa 3D-mal- lille sopiva tunnelma tai tyyli yksinkertaisesti. Tämä toimisi kuitenkin niin, että

kun malli on renderöity, sitä voi vielä helposti muokata pinnalta visuaalisesti ohjelmissa kuten After Effects tai Photoshop. Lisäksi esimerkiksi ink 'n paint - materiaali mahdollisesti vastaisi peliin tulevan maailman tyyliä. En ole myöskään vielä kolunnut kaikkia mahdollisuuksia, varsinkaan muilla ohjelmilla. Joi-tain näitä materiaaleja käyttämällä pystyisi välttämään kokonaan UV-kartoitus- osion; tosin vain siinä tapauksessa, että mallista tulee yksivärinen, tai jos se tehdään useammasta erillisestä osasta. Tämäkään ei ole täysi mahdottomuus sen vuoksi, että mallia ei käytetä 3D-moottorissa. En kuitenkaan pidä tällaista tekniikkaa parhaana mahdollisena, sillä UV-kartoituksen avulla voidaan saada tärkeitä yksityiskohtia mallin pinnalle. Se, mikä 3D-rendereissä saattaa tuottaa tuskia myöhemmin, on juuri se fakta, että renderöidystä animaatiosta on vai- kea ottaa frameja eli kuvia välistä ilman, että animaatio muuttuu tökkiväksi. Tämä tarkoittaa sitä, että jos halutaan lisätä visuaalisia efektejä toisissa ohjel- missa, pitää muutokset tehdä jokaiselle framelle. Koska frameja on 24 sekun- tia kohden ja animaatioita tulisi olemaan paljon erilaisia, ei tämä välttämättä ole mielekästä.

Koko animaatioprosessin nopeudessa on monia muuttujia. Kuten kävi ilmi, 3ds Maxissa riggaus ja animaatio oli tehty varsin helpoksi, minkä johdosta työskentelyaika lyheni jonkin verran. Jos käytetty ohjelma on kuitenkin joku toinen, on hyvin todennäköistä, että nämä samat työt ottavat enemmän aikaa, jos ei ole käytössä yhtä monipuolisia animaatiotyökaluja. Itse mallin animaa- tiovalmiiksi saanti (mallinnus, unwrappaus, teksturointi, riggaus, skinnaus) vie suhteessa enemmän aika, kuin esimerkiksi pala-animaatiossa hahmojen ko- koaminen, mutta kun malli on valmis, sitä voidaan usein uusiokäyttää tarpeen tullen useammallakin tavalla. Siinä missä 2D-tekniikoillakin on kaikki animaa- tiot tehtävä, on myös 3D:llä. Tämä tarkoittaa muutamia luetellakseni, juoksu-, kävely-, hyppy-, kyykistymis-, taistelu-, kääntymis- sekä välianimaatioita. Kaikki edellä mainitut on luotava useammalle eri hahmolle. Koska 3D-malleja on mahdollista uusiokäyttää ja niiden riggejäkin kierrättää, on kohtuullista sa- noa, että tässä mielessä 3D saattaa voittaa 2D:n nopeudella. 3D-animaation renderöinti vie oman aikansa ja resoluutiot ja muut koot ja mitat vaikuttavat tä- hänkin aikaan, mutta tässä vaiheessa minulla ei ole tarvittavia tietoja, jotta voi- sin sanoa tarkkoja aikoja. Lisäksi renderöintiäikoihin vaikuttavat käytetyn ko- neen tehot.

Jos en olisi ylimielisyyksissäni ruvennut tekemään Blenderissä ensimmäiseksi kohtalaisen monimutkaista hahmomallia, saattaisi minulla olla nyt parempaa tietoa Blenderin työkaluista, materiaaleista ja riggausmahdollisuuksista. Nyt kuitenkin kävi näin, koska yliarvioin nopeuteni oppia lähes tuntemattoman 3D-ohjelman käyttöä. Kuitenkin huomasin, kiitos aiemman kokemukseni 3ds Maxin kanssa, että Blenderin käyttöliittymää oli suhteellisen helppo ymmärtää sen sekavuudesta huolimatta. Jos olisin tullut käyttämään sitä ilman mitään kokemusta 3D-animaatiosta, tehtävä olisi ollut huomattavasti haastavampi ja kestänyt paljon kauemmin. Pystyn nyt kuitenkin mallintamaan ilmaisessa ohjelmassa, mikä on meriitti jo itsessään.

Yksi seikka, joka myös osaltaan vaikuttaa näiden prosessien nopeuteen, on henkilökohtainen kokemus tekniikan käytössä. Esimerkiksi meidän työryhmästä minä mahdollisesti tuotan nopeimmin 3D-malleja ja animaatiota, kun taas joku toinen ryhmän jäsen hallitsee pala-animaation paremmin. On siis vaikea arvioida tarkasti, mikä työtekniikka on lopulta projektissa nopeampaa, sillä vertailukohteina on tekniikoiden lisäksi kaksi eri työntekijää.

Loppujen lopuksi ajansäästöllisesti 3D-tekniikka ja 2D-tekniikat saattavat olla hyvinkin tasoissa. Kuitenkin tämä projekti, jossa valmiita 3D-renderöityjä frameja olisi tarkoitus vielä muokata halutun erilaisen visuaalisen tyylin saavuttamiseksi, tekee 3D-tekniikasta työläämmän ja hitaamman.

3D-mallinnuksella ja -animoinnilla voi tämän opinnäytetyön tulosten mukaan siis saada aikaan mielenkiintoisia visuaalisia vaihtoehtoja, jotka on mahdollista asettaa erilaisiin 2D-ympäristöihin, ilman että ne poikkeavat liiaksi ympäristöstään. Ajankäytöllisesti nimenomaan MZD-projektissa 3D:n käyttö osoittautui mahdollisesti enemmän aikaa vieväksi kuin jokin vaihtoehtoinen 2D-animaatiotekniikka. Visuaaliset ja tekniset perusteet ovat siis jossain määrin ristiriidassa. Tämän vuoksi meidän on punnittava hyödyt ja haitat ja päätettävä, mikä on paras ratkaisu projektin kannalta.

## LÄHTEET

Ahearn, L. 2009. 3D Game Textures. Create Professional Game Art Using Photoshop. USA: Elsevier, Inc.

Anna. 2015. Unity 3D. Saatavissa: <http://videogamedesign24.com/unity-3d/>, videogamedesign24.com [Viitattu 12.3.2016].

Autodesk knowledge network. 2014. Render to texture. Saatavissa: <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-76F1E8AE-9E8F-40A7-A948-471D17E09DA9-htm.html>, knowled-ge.autodesk.com [Viitattu 22.3.2016].

Baker, J. 2010. What exactly is UV and UVW Mapping? Saatavissa: <http://gamedev.stackexchange.com/questions/6911/what-exactly-is-uv-and-uvw-mapping>, gamedev.stackexchange.com [Viitattu 20.3.2016].

Blender Reference Manual. [https://www.blender.org/manual/rigging/posing/inverse\\_kinematics.html](https://www.blender.org/manual/rigging/posing/inverse_kinematics.html), www.blender.org [Viitattu 21.3.2016].

BuiBui. Bravely Default: Where the Fairy Flies on Nintendo 3DS. Saatavissa: [http://www.teenink.com/reviews/video\\_game\\_reviews/article/672585/Bravely-Default-Where-the-Fairy-Flies-on-Nintendo-3DS/](http://www.teenink.com/reviews/video_game_reviews/article/672585/Bravely-Default-Where-the-Fairy-Flies-on-Nintendo-3DS/), teenink.com [Viitattu 3.3.2016].

Chopine, A. 2011. 3D Art Essentials. The Fundamentals of 3D Modeling, Texturing, and Animation. New York, London: Focal Press.

Corijn, L. ambient occlusion baking. Saatavissa: <http://www.laurensco-rijn.com/articles/ambient-occlusion-baking>, www.laurensco-rijn.com [Viitattu 24.3.2016].

D J T. 2015. Talking topology Saatavissa: <http://theillusionden.blogspot.fi/2015/06/talking-topology.html>, theillusion-den.blogspot.fi [Viitattu 15.3.2016].

DWSMatt. 2015. SSAO vs. AO maps? Saatavissa: [https://www.reddit.com/r/Unity3D/comments/3nchrp/ssao\\_vs\\_ao\\_maps/](https://www.reddit.com/r/Unity3D/comments/3nchrp/ssao_vs_ao_maps/), www.reddit.com [Viitattu 22.3.2016].

Elec Man EXE. 2010. So this is going to be 3D 2D. Saatavissa: <http://www.gamafaqs.com/boards/997828-paper-mario-sticker-star/55415559?page=1>, gamafaqs.com [Viitattu 3.3.2016].

F, D. 2013. VIDEO GAMES, FILM, WHAT'S THE DIFFERENCE? IT'S STILL 3D MODELING RIGHT? Saatavissa: <http://www.thegnomon-workshop.com/news/2013/03/video-games-film-whats-the-difference-its-still-3d-modeling-right/>, www.thegnomonworkshop.com [Viitattu 21.3.2016].

Fariley, R. 2014. What is Cutout Animation. Saatavissa: <http://www.videomaker.com/article/f06/17164-the-horrors-of-rotoscoping>, www.videomaker.com [Viitattu 24.3.2016].

Frushtick, R. 2011. Why wasn't 'Pokemon Black' and 'White' made for 3ds and why are pokemon still 2D. <http://www.mtv.com/news/2462736/why-wasnt->

pokemon-black-and-white-made-for-3ds-and-why-are-pokemon-still-2d/, mtv.com [Viitattu 12.3.2016].

Gahan, A. 2009. 3ds Max Modeling For Games: Insiders Guide to Game Character, Vehicle, and Environment Modeling. Canada: Focal Press.

Ghosh, D. 2014. Theory of 3D modelling. Saatavissa: <http://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>. Saatavissa: [www.sketchup-ur-space.com](http://www.sketchup-ur-space.com) [Viitattu 13.3.2016].

Gray, A. Introduction to 3D Animation. Saatavissa: <http://www.animationarena.com/introduction-to-3d-animation.html>, [www.animationarena.com](http://www.animationarena.com) [Viitattu 23.3.2016].

Grinchesku, I. 2015. Question: what is the purpose of unwrapping 3d models? Saatavissa: <https://github.com/Arahroid/UVChecker-map/issues/2>, [github.com](https://github.com) [Viitattu 20.3.2016].

Helps, A. 2015. What 3D modelling software do video game companies mostly use? Saatavissa: <https://www.quora.com/What-3D-modelling-software-do-video-game-companies-mostly-use>, [www.quora.com](http://www.quora.com) [Viitattu 22.3.2016].

Holmes, D. 2012. A Mind Forever Voyaging: A History of Storytelling in Video Games. Seattle: CreateSpace Independent Publishing Platform.

IEEEExplore. 2009. Multiresolution Mean Shift Clustering Algorithm for Shape Interpolation. Saatavissa: [http://ieeexplore.ieee.org/ieee\\_pilot/articles/05/ttg2009050853/figures.html](http://ieeexplore.ieee.org/ieee_pilot/articles/05/ttg2009050853/figures.html), [ieeexplore.ieee.org](http://ieeexplore.ieee.org) [Viitattu 24.3.2016].

Jones, M. 2015. MDU115 – Stages of the 3D Production Pipeline – Part The First. Saatavissa: <https://crossjebus.wordpress.com/2015/06/17/mdu115-stages-of-the-3d-production-pipeline-part-the-first/>, [crossjebus.wordpress.com](http://crossjebus.wordpress.com) [Viitattu 14.3.2016].

Kennedy, S. R. 2013. How to Become a Video Game Artist. New York: Watson-Guptill Publications.

Krus, M. Levels of Detail. Saatavissa: <http://mkrus.free.fr/CG/LODS/>, [mkrus.free.fr](http://mkrus.free.fr) [Viitattu 21.3.2016].

Kumari, S. 2012. Complete Human Character Rig In 3D Studio Max, Part 5 Skinning. Saatavissa: <http://cgi.tutsplus.com/tutorials/complete-human-character-rig-in-3d-studio-max-part-5-skinning--cg-18775>, [cgi.tutsplus.com](http://cgi.tutsplus.com) [Viitattu 22.3.2016].

Lambert, S. 2013. An Introduction to Spritesheet Animation. Saatavissa: <http://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>, [gamedevelopment.tutsplus.com](http://gamedevelopment.tutsplus.com) [Viitattu 12.3.2016].

Mainard, J. 2009. All In – Dreamworks Animation Goes All 3D. [https://library.creativecow.net/mainard\\_jim/magazine\\_all\\_in/1/](https://library.creativecow.net/mainard_jim/magazine_all_in/1/), [li-brary.creativecow.net](http://library.creativecow.net) [Viitattu 12.3.2016].

Makuch, E. 2015. Ori and the Blind Forest Dev Wants to Continue the Story, Possibly With a Movie. Saatavissa: <http://www.gamespot.com/articles/ori-and->

the-blind-forest-dev-wants-to-continue-the/1100-6426538/, [www.gamespot.com](http://www.gamespot.com) [Viitattu 24.3.2016].

Masters, M. 2012. Understanding Skinning – The Vital Step for Any Rigging Project. Saatavissa: <http://blog.digitaltutors.com/understanding-skinning-vital-step-rigging-project/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 22.3.2016].

Masters, M. 2014. Understanding UVs - Love Them or Hate Them, They're Essential to Know. Saatavissa: <http://blog.digitaltutors.com/understanding-uvs-love-them-or-hate-them-theyre-essential-to-know/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 20.3.2016].

Masters, M. 2014. What's the Difference? A Comparison of Modeling for Games and Modeling for Movies. Saatavissa: <http://blog.digitaltutors.com/whats-the-difference-a-comparison-of-modeling-for-games-and-modeling-for-movies/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 20.3.2016].

Mayden, A. 2015. Modeling with quads or triangles – what should I use? Saatavissa: <http://blog.digitaltutors.com/modeling-with-quads-or-triangles/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 14.3.2016].

Meroz, M. What is 3D Animation Compared to 2D Animation? The Core Differences. Saatavissa: <https://www.blopanimation.com/what-is-3d-animation/>, [www.blopanimation.com](http://www.blopanimation.com) [Viitattu 24.3.2016].

Murdock, K. L. 2012. Autodesk 3ds Max 2013 Bible. The Comprehensive, Tutorial Resource. Indianapolis: John Wiley & Sons, Inc.

Newman, J. LEVEL OF DETAIL. What is LOD (Level Of Detail)? Saatavissa: <https://jnewman96.wordpress.com/year-2/oo-design/level-of-detail/>, [jnewman96.wordpress.com](http://jnewman96.wordpress.com) [Viitattu 21.3.2016].

Nintendolife. 2011. Pokemon Black and White. Saatavissa: [http://www.nintendolife.com/games/ds/pokemon\\_black\\_and\\_white/screenshots](http://www.nintendolife.com/games/ds/pokemon_black_and_white/screenshots), [www.nintendolife.com](http://www.nintendolife.com) [Viitattu 24.3.2016].

Rosen, D. 2009. Ambient occlusion for characters. Saatavissa: <http://blog.wolfire.com/2009/09/ambient-occlusion-for-characters/>, [blog.wolfire.com](http://blog.wolfire.com) [Viitattu 22.3.2016].

Russell, E. 2014. Understanding the Difference between Texture Maps. Saatavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016].

samandniko. 2014. DIY Action Figures with 3D printing! Saatavissa: <https://www.youtube.com/watch?v=yh3MSIEsh2w>, [www.youtube.com](http://www.youtube.com) [Viitattu 26.3.2016].

Sanders, A-L. 2014. What is Cutout Animation. Saatavissa: <http://animation.about.com/od/faqs/f/What-Is-Cutout-Animation.htm>, [animati-on.about.com](http://animation.about.com) [Viitattu 24.3.2016].

Silverman, D. 2013. 3D Primer for Game Developers: An Overview of 3D Modeling in Games. Saatavissa: <http://gamedevelopment.tutsplus.com/articles/3d-primer-for-game-developers-an-overview-of-3d-modeling-in-games--gamedev-5704>, [gamedevelopment.tutsplus.com](http://gamedevelopment.tutsplus.com) [Viitattu 21.3.2016].

Slick, J. 2014. 7 Common Modeling Techniques for Film and Games. Saatavissa: <http://3d.about.com/od/3d-101-The-Basics/a/Introduction-To-3d-Modeling-Techniques.htm>, 3d.about.com [Viitattu 14.3.2016].

Slick, J. 2016. Surfacing 101 - Creating a UV Layout. Saatavissa: <http://3d.about.com/od/3d-101-The-Basics/a/Surfacing-101-Creating-A-UV-Layout.htm>, 3d.about.com [Viitattu 20.3.2016].

Slick, J. 2016. Surfacing 101 – Shading Networks and Surface Materials. Saatavissa: <http://3d.about.com/od/Creating-3D-The-CG-Pipeline/a/Surfacing-101-Shading-Networks-And-Surface-Materials.htm>, 3d.about.com [Viitattu 21.3.2016].

Slick, J. 2014. Surfacing 101 – Texture Mapping. Saatavissa: <http://3d.about.com/od/3d-101-The-Basics/a/Surfacing-101-Texture-Mapping.htm>, 3d.about.com [Viitattu 21.3.2016].

Slick, J. 2016. What is Rigging? Saatavissa: <http://3d.about.com/od/Creating-3D-The-CG-Pipeline/a/What-Is-Rigging.htm>, 3d.about.com [Viitattu 21.3.2016].

Standard Primitives. Saatavissa: <http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/index.html?url=files/GUID-66152BDE-BA64-423F-8472-C1F0EB409E16.htm,topicNumber=d30e26059>, docs.autodesk.com [Viitattu 18.3.2016].

Steighner, M. 2011. Understading Review: Trine 2. Saatavissa: <http://videogamewriters.com/review-trine-2-31980>, videogamewriters.com [Viitattu 24.3.2016].

Supergiant games. Bastion. Saatavissa: <https://www.supergiantgames.com/games/bastion/>, www.supergiantgames.com [Viitattu 24.3.2016].

Suridge, J. 2013. Modelling by numbers: part One A. Saatavissa: [http://www.gamasutra.com/blogs/JayelindaSuridge/20130903/199457/Modeling\\_by\\_numbers\\_Part\\_One\\_A.php](http://www.gamasutra.com/blogs/JayelindaSuridge/20130903/199457/Modeling_by_numbers_Part_One_A.php), www.gamasutra.com [Viitattu 24.3.2016].

teachmecone. Class 04. Saatavissa: [http://www.teachmecone.com/?page\\_id=882](http://www.teachmecone.com/?page_id=882), www.teachmecone.com [Viitattu 14.3.2016].

thedaian. 2011. What are the pros/cons of using a 3D engine for 2D games? Saatavissa: <http://gamedev.stackexchange.com/questions/15790/what-are-the-pros-cons-of-using-a-3d-engine-for-2d-games>, game-dev.stackexchange.com [Viitattu 24.3.2016].

toonboom. 2D/3D Mixed Media. Saatavissa: <https://www.toonboom.com/solutions/2d-3d-mixed-media>, www.toonboom.com [Viitattu 24.3.2016].

toonboom. Cut-out Animation. Saatavissa: [http://docs.toonboom.com/help/harmony-12/premium/Content/\\_CORE/\\_Workflow/003\\_Animation\\_Workflow/007\\_H2\\_Cut-out\\_Animation.html](http://docs.toonboom.com/help/harmony-12/premium/Content/_CORE/_Workflow/003_Animation_Workflow/007_H2_Cut-out_Animation.html), docs.toonboom.com [Viitattu 24.3.2016].

Total Fabrication. What is an isometric drawing? Saatavissa: [http://metal.brightcookie.com/2\\_draw/draw\\_t3/htm/draw3\\_2\\_2.htm](http://metal.brightcookie.com/2_draw/draw_t3/htm/draw3_2_2.htm), metal.brightcookie.com [Viitattu 24.3.2016].

unityhacker. Shaders, Materials, and Textures. Saatavissa <https://create3dgames.wordpress.com/2012/07/24/shaders-materials-and-textures/>, create3dgames.wordpress.com [Viitattu 21.3.2016].

Vas, G. 2013. Video Games With The Most Memorable Pre-Rendered Backgrounds. Saatavissa: <http://kotaku.com/video-games-with-the-most-memorable-pre-rendered-backgr-499808427>, kotaku.com [Viitattu 24.3.2016].

Ward, A. 2013. Game Character Creation Series: Kila Chapter 3 - UV Mapping. Saatavissa: <http://cgi.tutsplus.com/articles/game-character-creation-series-kila-chapter-3-uv-mapping--cg-26754>, cgi.tutsplus.com [Viitattu 20.3.2016].

Weinberg, G. 2016. Dimensions (physics): What is 1D, 2D, 3D, and 4D? How is it easily understood by a beginner. Saatavissa: <https://www.quora.com/Dimensions-physics/What-is-1D-2D-3D-and-4D-How-is-it-easily-understood-by-a-beginner>, www.quora.com [Viitattu 24.3.2016].

## KUVALUETTELO

Kuvat 1 a ja b. Vasemmalla kuvassa on yksinkertainen sprite, kun taas oikealla kuvassa on sprite sheet, jossa näkyvät kaikki Spritet tytön juoksuanimaatiosta. Lambert, S. 2013. An Introduction to Spritesheet Animation. Saatavissa: <http://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>, gamedevelopment.tutsplus.com [Viitattu 12.3.2016].

Kuva 2 b. Vasemmalla kolmion mallinen polygoni, joka koostuu kolmesta verteksistä, kolmesta edgestä ja yhdestä facesta. Oikealla eri polygonityypit vasemmalta oikealle: tri-polygoni, quad-polygoni ja ngon-polygoni. teachmecone. Class 04. Saatavissa: [http://www.teachmecone.com/?page\\_id=882](http://www.teachmecone.com/?page_id=882), www.teachmecone.com [Viitattu 14.3.2016].

Kuva 3. Esimerkki huonosta ja hyvästä topologiasta. D J T. 2015. Talking topology Saatavissa: <http://theillusionden.blogspot.fi/2015/06/talking-topology.html>, theillusion-den.blogspot.fi [Viitattu 15.3.2016].

Kuva 4. Kuvassa on joukko 3ds Max-ohjelmasta löytyviä geometria-alkioita. Nämä alkioitkin ovat polygoniverkkoja, joskin yksinkertaisia sellaisia. Autodesk knowledge network. 2014. Render to texture. Saatavissa: <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-76F1E8AE-9E8F-40A7-A948-471D17E09DA9-htm.html>, knowledge.autodesk.com [Viitattu 22.3.2016].

Kuvat 5 a ja b. Ensimmäisessä kuvassa on esimerkki polygonimallinnuksesta ja toisessa edge-mallinnuksesta. Huomaa, kuinka edge-tekniikassa päästään nopeasti tarkempaan yksityiskohtien työstöön. Jones, M. 2015. MDU115 – Stages of the 3D Production Pipeline – Part The First. Saatavissa: <https://crossjebus.wordpress.com/2015/06/17/mdu115-stages-of-the-3d-production-pipeline-part-the-first/>, crossjebus.wordpress.com [Viitattu 14.3.2016].

Kuva 7. Vasemmalla malli ja oikealla mallille tehty UV-kartta. Ward, A. 2013. Game Character Creation Series: Kila Chapter 3 - UV Mapping. Saatavissa: <http://cgi.tutsplus.com/articles/game-character-creation-series-kila-chapter-3-uv-mapping--cg-26754>, cgi.tutsplus.com [Viitattu 20.3.2016].

Kuva 8. Vasemmalta oikealle: pallon mallinen kartoitus, sylinterin mallinen kartoitus ja tasomallinen kartoitus. Masters, M. 2014. Understanding UVs - Love Them or Hate Them, They're Essential to Know. Saatavissa: <http://blog.digitaltutors.com/understanding-uvs-love-them-or-hate-them-theyre-essential-to-know/>, blog.digitaltutors.com [Viitattu 20.3.2016].

Kuva 9. Vasemmalla näkyy esimerkki huonosta UV-kartasta ja oikealla siisti, valmis UV-kartta. Materiaali 3D-mallin paidassa on checker-ruudukkokuva, joka auttaa hahmottamaan paremmin, kuinka UV on aseteltu. Grinchesku, I. 2015. Question: what is the purpose of unwrapping 3d mod-els? Saatavissa: <https://github.com/Arahroid/UVChecker-map/issues/2>, git-hub.com [Viitattu 20.3.2016].

Kuva 10. Kuvassa näkyy olennolle luotu värikartta. Kartassa näkyvät myös UV:t. Russell, E. 2014. Understanding the Difference between Texture Maps. Saa-tavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016]

Kuva 11. Kuvattuna on kiiltokartta/specula-kartta. Erityisesti silmien ympärillä ja kärsän alueella voi nähdä kuinka kartta on vaaleampi, eli kiiltoa tulee noille alueille enemmän. Russell, E. 2014. Understanding the Difference between Texture Maps. Saa-tavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016].

Kuva 12. Bump-kartta, jossa näkyy pieniä muutoksia. Russell, E. 2014. Understanding the Difference between Texture Maps. Saa-tavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016].

Kuva 13. Kuvassa vasemmalta oikealle: ensimmäisenä matalapolygonimalli, johon on lisätty tekstuurirenderöity normal-kartta, keskellä pelkkä matalapolygonimalli ja viimeisenä korkeapolygonimalli, josta tekstuurirenderöinti tehtiin. F, D. 2013. VIDEO GAMES, FILM, WHAT'S THE DIFFERENCE? IT'S STILL 3D MODELING RIGHT? Saatavissa: <http://www.thegnomonworkshop.com/news/2013/03/video-games-film-whats-the-difference-its-still-3d-modeling-right/>, [www.thegnomonworkshop.com](http://www.thegnomonworkshop.com) [Viitattu 21.3.2016].

Kuva 14. Kuvassa näkee, kuinka hahmon päässä on patteja, jotka on luotu displacement-kartan avulla. Russell, E. 2014. Understanding the Difference between Texture Maps. Saa-tavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016].

Kuva 15. Hahmolle on luotu tatuointi läpinäkyvyyskartan avulla. Russell, E. 2014. Understanding the Difference between Texture Maps. Saa-tavissa: <http://blog.digitaltutors.com/understanding-difference-texture-maps/>, [blog.digitaltutors.com](http://blog.digitaltutors.com) [Viitattu 21.3.2016].

Kuva 16. Vasemmalta oikealle: hahmo ilman tekstuuria, hahmo ambient occlusion-kartan kanssa ja lopuksi hahmo, jolla on sekä värit että ambient occlusion. Rosen, D. 2009. Ambient occlusion for characters. Saatavissa: <http://blog.wolfire.com/2009/09/ambient-occlusion-for-characters/>, [blog.wolfire.com](http://blog.wolfire.com) [Viitattu 22.3.2016].

Kuva 17. Vasemmalla näkyy, kuinka malli on vääristynyt taiteen kohdalta. Oikealla taiteen verteksit on jaettu paremmin ja taive käyttäytyy luonnollisemmin. Kumari, S. 2012. Complete Human Character Rig In 3D Studio Max, Part 5 Skinning. Saatavissa: <http://cgi.tutsplus.com/tutorials/complete-human-character-rig-in-3d-studio-max-part-5-skinning--cg-18775>, [cgi.tutsplus.com](http://cgi.tutsplus.com) [Viitattu 22.3.2016].

Kuva 18. Oranssilla värjätty hahmot on aseteltu avainruuduiksi, jotka sitten 3D-ohjelma automaattisesti yhdistää väliruuduilla (siniset hahmot). IEEEExplore. 2009. Multiresolution Mean Shift Clustering Algorithm for Shape Interpolation. Saatavissa: [http://ieeexplore.ieee.org/ieee\\_pilot/articles/05/ttg2009050853/figures.html](http://ieeexplore.ieee.org/ieee_pilot/articles/05/ttg2009050853/figures.html), [ieeexplore.ieee.org](http://ieeexplore.ieee.org) [Viitattu 24.3.2016].

Kuva 19. Mitä matalampi polygoninen kanimalli on, sitä kauempana se näytetään, jolloin tuskin huomaa mallin olevan yksinkertaisempi. Krus, M. Levels of

Detail. Saatavissa: <http://mkrus.free.fr/CG/LODS/>, mkrus.free.fr [Viitattu 21.3.2016].

Kuva 20. Kuvassa näkyy pala-animaatiohahmon palaset. toonboom. Cut-out Animation. Saatavissa: [http://docs.toonboom.com/help/harmony-12/premium/Content/\\_CORE/\\_Workflow/003\\_Animation\\_Workflow/007\\_H2\\_Cut-out\\_Animation.html](http://docs.toonboom.com/help/harmony-12/premium/Content/_CORE/_Workflow/003_Animation_Workflow/007_H2_Cut-out_Animation.html), docs.toonboom.com [Viitattu 24.3.2016].

Kuva 21. Ori and the Blind Forest on 2D side-scroller -peli, joka muistuttaa ulkonäöltään ja toiminnaltaan 3D peliä. Siksi sitä voi kutsua 2.5D-peliksi. Makuch, E. 2015. Ori and the Blind Forest Dev Wants to Continue the Story, Possibly With a Movie. Saatavissa: <http://www.gamespot.com/articles/ori-and-the-blind-forest-dev-wants-to-continue-the/1100-6426538/>, www.gamespot.com [Viitattu 24.3.2016].

Kuva 22. Trine on 3D side-scroller -peli jossa kamera seuraa pelihahmoa pysyvästi sivusuunnassa, eikä hahmo pysty liikkumaan syvyysuunnassa. Liikkuminen onnistuu ainoastaan 2D-tasolla eteen-, taakse-, ylös- ja alaspäin. Tästäkin peliä voi siis kutsua 2.5D-peliksi. Steighner, M. 2011. Understading Review: Trine 2. Saatavissa: <http://videogamewriters.com/review-trine-2-31980>, videogamewriters.com [Viitattu 24.3.2016].

Kuvat 23 a ja b. Kuvassa a vasemmalla on Resident evil 3: Nemesis esirenderöity ympäristö ja kuvassa b oikealla on pelistä Final Fantasy 8 esirenderöity maisema, jossa näkyy 3D-pelaajahahmoja mukana. Vas, G. 2013. Video Games With The Most Memorable Pre-Rendered Back-grounds. Saatavissa: <http://kotaku.com/video-games-with-the-most-memorable-pre-rendered-backgr-499808427>, kotaku.com [Viitattu 24.3.2016].

Kuva 24 a. Vasemmalla Pokemon Black & Whiten kaksiulotteinen pelihahmo kolmiulotteisessa ympäristössä. Oikealla Bastion-peli 3D-hahmoineen ja 2D-isometrisine maailmoineen. Nintendolife. 2011. Pokemon Black and White. Saatavissa: [http://www.nintendolife.com/games/ds/pokemon\\_black\\_and\\_white/screenshots](http://www.nintendolife.com/games/ds/pokemon_black_and_white/screenshots), www.nintendolife.com [Viitattu 24.3.2016].

Kuva 24 b. Vasemmalla Pokemon Black & Whiten kaksiulotteinen pelihahmo kolmiulotteisessa ympäristössä. Oikealla Bastion-peli 3D-hahmoineen ja 2D-isometrisine maailmoineen. Supergiant games. Bastion. Saatavissa: <https://www.supergiantgames.com/games/bastion/>, www.supergiantgames.com [Viitattu 24.3.2016].