

TEOLLISUUDEN 3D-MALLIN OPTIMOINTI REAALIAIKARENDERÖINTIIN

Niko Klemettinen

Opinnäytetyö
Joulukuu 2012

Mediatekniikan koulutusohjelma
Tekniikan ja liikenteen ala



Tekijä(t) KLEMETTINEN, Niko	Julkaisun laji Opinnäytetyö	Päivämäärä 10.12.2012
	Sivumäärä 58	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty (X)
Työn nimi TEOLLISUUDEN 3D-MALLIN OPTIMOINTI REAALIAIKARENDERÖINTIIN		
Koulutusohjelma Mediatekniikka		
Työn ohjaaja(t) NIEMI, Kari		
Toimeksiantaja(t) Movya Oy		
<p>Opinnäytetyön toimeksiantajana toimii Movya oy. Toimeksiantaja on yksi merkittävimmistä innovatiivisten monimediaratkaisujen tuottajista suomessa.</p> <p>Teollisuuden 3D-mallit ovat usein liian raskaita reaaliaikaiseen visualisointiin, kuten peleihin ja tuotekonfiguraattoreihin. Opinnäytetyön tehtävänä oli selvittää kuinka teollisuuden 3D-malli saadaan toimimaan sujuvasti reaaliaikavisualisoinnissa. Ongelmaan haettiin ratkaisuja optimoinnin keinoin, viimeisimpiä tekniikoita hyödyntäen.</p> <p>Tietoperustassa käsitellään 3D -mallin geometrian ja visualisoinnin perusteita sekä optimointiprojektin eri vaiheita. Lisäksi osiossa käydään läpi reaaliaikasovelluksen 3D -mallille asettamat rajoitukset. Opinnäytetyön käytännön osiossa käydään läpi optimointiprojektin kulku suunnitteluvaiheesta siihen pisteeseen, että 3D -malli voidaan viedä reaaliaikasovellukseen.</p> <p>Opinnäytetyön tuloksena syntyi ohjeistus teollisen 3D -mallin optimoinnista reaaliaikarenderöintiin. Ohjeistuksessa käydään läpi yleisimmät 3D -mallin optimointitekniikat. Samalla esitellään tekniikoita, joilla voidaan havaita ja korjata 3D -geometriassa piileviä ongelmia. Ohjeistuksen avulla Movya Oy:n on tulevaisuudessa mahdollista toteuttaa vaivatta teollisuuden 3D -mallin optimointiprojekti reaaliaikasovellusta varten. Ohjeistus toimii lisäksi hyvänä perehdytystyökaluna uusille työntekijöille</p>		
Avainsanat (asiasanat) 3D, 3D max, pelimoottori, optimointi, Autodesk,		
Muut tiedot		

Author(s) KLEMETTINEN, Niko	Type of publication Bachelor's Thesis	Date 10.12.2012
	Pages 58	Language Finnish
		Permission for web publication (X)
Title INDUSTRIAL 3D-MODEL OPTIMIZATION FOR REALTIME RENDERING		
Degree Programme Media Engineering		
Tutor(s) NIEMI, Kari		
Assigned by Movya Ltd		
<p>The assigner of the bachelor's thesis was Movya Ltd. The assigner is significant producer of innovative digital multimedia solutions in Finland.</p> <p>Industrial 3Dmodels are usually too heavy and precise for real-time rendering solutions, such as games and configurators. The purpose of this bachelor's thesis was to find out how heavy models can be used in real-time rendering. The problem was researched and resolved by using optimization methods.</p> <p>In the theory part of thesis the basics of 3D models is discussed including theory about geometry and basics of visualization. as well a the limits that realtime solutions set to the model in question. In the practical part of the thesis, the implementation of an optimization project is studied from the planning phase to the situation where model is optimized and ready to go for visualization.</p> <p>The result of the thesis was a tutorial about the optimization project of 3D models for a real-time solution. In the manual most common optimization tricks for 3D models are presented. At the same time the thesis illustrates how to spot and fix hidden problems of 3Dgeometry.</p> <p>With this manual Movya Ltd. can accomplish a 3D- optimization project. The enterprise and can also use this manual for orientation of new employees.</p>		
Keywords 3D, 3D max, Game engine, Optimization, Autodesk, Real-time		
Miscellaneous		

SISÄLTÖ

KUVIOT	3
KÄSITTEET JA LYHENTEET	4
1 TYÖN LÄHTÖKOHDAT.....	5
1.1 TAUSTA JA TOIMEKSIANTAJA	5
1.2 TAVOITTEET JA TEHTÄVÄT.....	6
2 LAITEKOKOONPANO JA OHJELMISTOT	7
2.1 LAITEKOKOONPANO	7
2.2 OHJELMISTOT	8
2.2.1 Autodesk 3ds Max 2013.....	8
2.2.2 Away 3D & Flash	8
2.2.3 AWD plugin	9
2.2.4 Vray 2.0	9
2.2.5 Deep Exploration.....	10
3 3D-VISUALISOINNISTA	10
3.1 YLEISTÄ VISUALISOINNISTA.....	10
3.2 3D-VISUALISOINTI	11
3.3 3D -VISUALISOINTI TEOLLISUUDESSA.....	13
3.4 ANIMAATIO JA SEN HISTORIA.....	13
3.4.1 INTERAKTIIVISET JA REAALIAIKAISET 3D-ANIMAATIOT	16
3.4.2 Yleistä.....	16
3.4.3 Pelimoottorit.....	17
4 3D-MALLINNUS- JA -VISUALISOINTITEKNIIKAT	18
4.1 KÄYTÄNTEET 3D-PROJEKTISSA.....	18
4.2 3D-GEOMETRIA	21
4.2.1 Geometrian rakenne.....	21
4.2.2 Normaalivektori	23

4.2.3	Yksinkertainen ja monimutkainen geometria	23
4.3	TEKSTUROINTI.....	24
4.3.1	Shaderit	24
4.3.2	Tekstuuri	25
4.3.3	Tekstuuri ja UV-mapping.....	25
4.3.4	Bump-, Normal-, Displacement- ja Transparency-kartat	26
4.4	VALAISTUS	29
4.4.1	Säteenjäljitys.....	29
4.4.2	Valotyypit.....	30
4.5	KAMERAT.....	31
4.5.1	Kameratypit.....	31
4.5.2	Kameraefektit	32
4.6	ANIMOINTI.....	33
4.6.1	Keyframe-animaatio	33
4.6.2	Kaavioilla animointi.....	33
5	RENDERÖINTI.....	36
6	TEOLLISUUDEN 3D-MALLIN OPTIMOINTI KÄYTÄNNÖSSÄ.....	37
6.1	YLEISTÄ	37
6.2	PROJEKTIN MÄÄRITTELY JA SUUNNITTELU.....	37
6.3	TUOTANTOVAIHE.....	39
6.3.1	3D-mallin kääntäminen.....	39
6.3.2	Mallin tuominen käsiteltäväksi	39
6.3.3	Mallin jäsentely ja karsiminen	40
6.3.4	Geometrian ongelmien tunnistaminen.....	41
6.3.5	Uuden geometrian luominen.....	42
6.3.6	Geometristen ongelmien korjaaminen	45
6.3.7	UVW-karttojen määrittely	47
6.3.8	Teksturointi.....	48
6.3.9	Animointi.....	50
6.3.10	Referenssikuvan renderöinti	51
6.3.11	Optimoidun mallin jättäminen	52
7	POHDINTA.....	55
8	LÄHTEET	56

KUVIOT

KUVIO 1. Visualisointi syvyysakselin vaikutuksesta koordinaatistoon.	12
KUVIO 2. Edvin Catmullin mallinnus ihmiskädestä.	15
KUVIO 3. Kohtaus John Lasseterin Luxo JR -lyhytelokuvasta.	15
KUVIO 4. Polygonin rakenne esitettynä kuutiolla.	21
KUVIO 5. Esimerkkejä erilaisista polygonirakenteista.	22
KUVIO 6. Esimerkki Lambert- ja Phong-shadereiden eroista.	24
KUVIO 7. Vasemmalla pallo bump karttalla, oikealla displacement karttalla.	27
KUVIO 8. Tangenttiin perustuvalla normaalikartalla teksturoitu kiikari.	28
KUVIO 9. Transparency kartan vaikutus geometriaan ja tekstuuriin.	29
KUVIO 10 Porrastettu curve.	34
KUVIO 11 Lineaarinen curve.	34
KUVIO 12 Kaareva curve.	35
KUVIO 13. Geometrian virheitä.	41
KUVIO 14. Smoothing groupin vaikutus geometrian ilmentymiseen.	43
KUVIO 15. Valon jakautuminen tasaisesti normaalivektoreiden suuremman määrän myötä.	44
KUVIO 16. Esimerkki tarkkuustason suhteesta objektin etäisyyteen.	45
KUVIO 17. AWD plugin, tiedoston tallennus .awd -muotoon	54

KÄSITTEET JA LYHENTEET

Algoritmi	Algoritmi on kuvaus siitä miten jokin tehtävä tai prosessi suoritetaan.
CAD	Computer Aided Design, mikä tarkoittaa tietokoneavusteista suunnittelua.
Globaali valaistus	Global illumination on joukko epäsuoran valaistuksen laskennallisia algoritmeja.
Interpolointi	Kahden arvon välissä olevien arvojen laskeminen.
Optimointi	Pyrkimys nykyistä parempaan tai parhaaseen mahdolliseen ratkaisuun.
Konfiguraattori	Ohjelma, jolla voidaan vaikuttaa tuotteen ominaisuuksiin.
Kohtaus	Scene (engl.). Tilanne, joka koostuu useasta samaan tilaan sijoitetusta objektista.
Pelimoottori	Ohjelma, joka vastaa objektien mallintamisesta ja piirtämisestä näytölle.
Polygoni	Monikulmio.
Renderöinti	2D-kuvan muodostaminen 3D-mallista.
Shader	Objektin pinnan varjostusalgoritmi.

1 TYÖN LÄHTÖKOHDAT

1.1 Tausta ja toimeksiantaja

Opinnäytetyön taustat perustuvat työelämälähtöiseen ongelmaan, johon lähdettiin hakemaan ratkaisuja kokeilemalla erilaisia optimointiratkaisuja käytännössä. Toimeksiantajan asiakaskunta koostuu teollisuusyrityksistä, joiden tuotteita visualisoidaan erilaisiin esitystapoihin. Yksi tällainen esitystapa on tuotekonfiguraattori, jonka valmistamisesta Movya Oy:llä on vankka osaaminen. Tuotekonfiguraattorissa asiakkaan tuotetta voidaan katsella eri kuvakulmista sekä mahdollisuuksien mukaan räätälöisä asiakkaan tarpeiden mukaisesti. Esimerkiksi tuotteeseen voidaan lisätä tai poistaa ominaisuuksia, tai vaihtaa osien värejä.

Tuotekonfiguraattorit on yleensä tehty suurimmaksi osaksi staattisilla kuvilla. Tämä tekniikka vaatii kuitenkin huomattavan paljon resursseja 3D-visualistilta, joka joutuu pahimmassa tapauksessa visualisoimaan useita satoja kuvia eri kuvakulmista. Kuvien suuri määrä hankaloittaa myös itse konfiguraattorin jakamista eteenpäin, esimerkiksi internetin välityksellä. Ratkaisu olemassaolevaan ongelmaan on rakentaa konfiguraattori helposti jaettavan, mahdollisimman monella eri alustalla toimivan pelimoottorin päälle. Pelimoottorilla pystytään visualisoimaan reaaliaikaisesti 3D-malleja sekä tekemään niille erilaisia muutoksia. Pelimoottorit poistavat tuotekonfiguraattorille aikaisemmin asetetut rajoitteet ennalta määrättyjen kuvakulmien suhteen, mutta luovat samalla omat rajoitteensa. Yksi tällainen rajoite on itse 3D-malli. Teollisuuden 3D-mallit ovat usein erittäin tarkkoja, niin

sanottuja tuotantomalleja, eivätkä sellaisenaan sovellu reaaliaikaisille esitystekniikoille, kuten pelimoottoreille. Opinnäytetyössä käsitellään teollisen 3D -mallin optimointia pelimoottoreille ammattitaidolla, viimeisimpiä tekniikoita hyödyntäen.

Toimeksiantaja oli Movya Oy, joka tarjoaa innovatiivisia ja korkeatasoisia, digitaalisia monimediapalveluita ja -tuotteita teollisuuteen. Asiakaskunta koostuu suurista metsä-, konepaja- sekä rakennustuoteyhtiöistä. Yritys työllistää tällä hetkellä 15 henkilöä, joiden työkenttä koostuu muun muassa ohjelmoinnista, 3D-visualisoinnista, videotuotannosta sekä visuaalisesta suunnittelusta. (Movya 2012.)

1.2 Tavoitteet ja tehtävät

Opinnäytetyön tavoitteena oli tuottaa toimeksiantajalle kattava dokumentaatio 3D-mallien optimoinnista. Tätä dokumentaatiota voidaan hyödyntää tulevaisuudessa optimointiin liittyvissä ongelmissa. Dokumentaatio on myös hyvä perustietopaketti toimeksiantajan tuleville työntekijöille, jotka tulevat väistämättä kohtaamaan teollisuuden raskaiden 3D-mallien ongelmia joko perinteisiä tai reaaliaikaisia visualisointimenetelmiä käyttäessään.

Opinnäytetyötä tehtäessä saatiin vastaus myös optimointiin liitettävään tuotantotekniseen kysymykseen: onko tuotantotehokkuuden kannalta kannattavaa lähteä optimoimaan teollisuuden 3D-malleja pelimoottoreille vai olisiko tällaiset mallit syytä rakentaa täysin uudelleen niillä rajoituksilla, jotka niiden esitystapa asettaa?

Tämän opinnäytetyön tekemisen aikana toimeksiantaja tutki pelimoottorin ohjelmointia, mikä tuki työn edistymistä. Työn tekemisen yhteydessä oli mahdollista päästä kokeilemaan 3D-mallin käyttäytymistä pelimoottorilla käytännössä. Tutkimuksen tulokset antavat toimeksiantajalle hyvän perustan

luoda tehokkaita sekä visuaalisesti näyttäviä 3D-tuotekonfiguraattoreita. Toimeksiantajan on lisäksi helpompi lähteä markkinoimaan tuotetta kahdesta erisyydestä. Tieto kulutetuista resursseista tekee tulevien asiakasprojektien resurssoinnista helpompaa, ja näin ollen auttaa projektien hinnoittelussa. Lisäksi konkreettinen esimerkki tuotteen toiminnallisuudesta ja visuaalisesta näyttävyydestä helpottaa vakuuttamaan asiakkaan toimeksiantajan ammattitaidosta.

2 LAITEKOKOONPANO JA OHJELMISTOT

2.1 Laitekoonpano

Tehokkuus on tärkeä avainsana ammattimaisessa 3D-mallien käsittelyssä. Optimoimattoman 3D-mallin käsittely vaatii tietokoneelta tehokasta suoritinta, näytönohjainta sekä riittävää määrää muistia. Tietokoneen tehontarve riippuu käsiteltävien mallien sekä polygonien määrästä. Pienemmän kohtauksen (alle tuhat objektia ja alle satatuhatta polygonia) esittämiseen riittää 4 gigabittiä RAM-muistia ja 512 megabittiä näyttömuistia. Puhuttaessa ammattimaisesta laitekoonpanosta tarvitaan vähintään 8 gigabittiä RAM-muistia ja suositusten mukaisesti myös vähintään 1 gigabitti näyttömuistia. Työskenneltäessä 3D-ohjelmien parissa on yhteensopivuusongelmien välttämiseksi kannattavaa valita videokortiksi kortti, joka on löydettävissä eniten käytetyn 3D-ohjelman tukilistalta. (3ds Max. System requirements 2012.)

Kaksi monitoria nopeuttaa työskentelyä, tutkimusten mukaan noin 20 -30 %. 3D-ohjelmien käyttöliittymä näyttää suuren määrän eri näkymiä samaan aikaan, joten kahdella monitorilla työskentely vähentää ikkunoiden jatkuvaa avaamista ja sulkemista. Kahdesta monitorista on hyötyä myös tilanteissa, joissa visualisointi vaatii tuekseen referenssikuvaa tai toimintaohjetta. (Anderson, Colvin, Tobler 2003.)

2.2 Ohjelmistot

2.2.1 Autodesk 3ds Max 2013

Autodesk 3ds Max on kolmiulotteisten mallien luomiseen, animointiin ja renderöintiin luotu tehokas ja helppokäyttöinen ohjelma, joka sisältää kattavat työkalut 3D-visualisteille, animaattoreille sekä visuaalisten efektien tuottajille. 3ds Max on erittäin laajasti käytössä oleva ohjelma, sillä se soveltuu useisiin eri tarkoituksiin insinööritasoisien tuotantomallien visualisoinnista peli- ja elokuvateollisuuden asettamiin haastaviin ominaisuuksiin. Yksi 3ds Maxin ominaisuuksista on tehokas MassFX -fysiikkamoottori, joka mahdollistaa erilaiset fysiikan lakeihin perustuvat ilmiöt, kuten painovoimaan perustuvan putoamisen sekä uusimpana ominaisuutena kankaiden repeytymisen. Laaja käyttäjäkunta luo ohjelman ympärille tiiviin yhteisön, jossa apua on saatavilla lähes kaikkiin ongelmiin. Mikäli jotakin valmista ominaisuutta ei ohjelmasta löydy, voidaan sellainen ohjelmoida 3ds Maxin omalla scripti-kielellä, MAXScriptillä. MAXScriptin ansiosta 3ds Maxiin on saatavalla monia tuotantoa tehostavia lisäosia. (3ds Max – tuotteet 2012.)

3ds Maxin valitsemiseen tämän projektin päätyökaluksi vaikutti suuresti se, että ohjelma löytyy valmiiksi toimeksiantajalta ja se tulee olemaan tulevaisuudessakin tärkein ohjelma 3D-mallien käsittelyssä. Lisäksi tekijän oma aikaisempi tietämys ohjelmistosta ei antanut edellytyksiä harkita ohjelman vaihtoa. Lähes kaikki työssä käsiteltävät optimointi- ja visualisointitekniikat pystytään toteuttamaan muillakin vastaavilla ohjelmilla, joista esimerkkeinä mainittakoon Autodesk Maya tai Cinema 4D.

2.2.2 Away 3D & Flash

Away 3D on avoimen lähdekoodin luokkakirjasto 3D-grafiikan visualisointiin Flash playerissä. Kirjastolla on mahdollista luoda reaaliaikaista 3D-animaatiota, minkä vuoksi Away 3D:tä voidaan kutsua niin sanotuksi 3D-moottoriksi. Moottorin kehitys alkoi vuonna 2007 Alexander Zadorozhnyin ja Rob Batemanin toimesta.

Away 3D:n 4.0 versio mahdollistaa kuvan muodostuksen hyödyntäen näytönohjaimen muistia. (Away 3d Features 2012.)

2.2.3 AWD plugin

AWD plugin on Away 3D -tiimin kehittämä lisäosa, jonka avulla voidaan kääntää 3D-editointiohjelman tiedostot Away 3D -ohjelmalle sopivaan muotoon. Away 3D plugin on saatavilla yleisimpiin 3D-editointiohjelmiin (3ds Max, Blender ja Maya.) AWD pluginilla käännettyyn tiedostoon voidaan lisätä tekstuureja, animaatioita ja paljon muuta 3D-mallin muodostamiseen tarvittavaa tietoa. (Adding the AWD plugin 2012.)

2.2.4 Vray 2.0

Vray 2.0 on Chaos Groupin kehittämä tehokas renderöintimoottori fotorealististen kuvien luomiseen 3D-objekteista. Vray käyttää kuvan muodostamiseen globaalin valaistuksen -algoritmeja, kuten fotoni- ja irradianssikarttaa. Näiden algoritmien avulla lasketaan se miten valonsäteily heijastuu, sitoutuu ja hajaantuu objektien pinnoille. Globaalin valaistuksen ansiosta Vray on erittäin laskentatehokas työkalu kuvien muodostukseen. Vrayn suosio perustuu sen helppokäyttöisyyteen, laskennalliseen tehokkuuteen sekä sen erinomaisiin algoritmeihin, joiden ansiosta realistisen valaistuksen luominen on helppoa. Kuvan renderöintivaiheessa on mahdollista jakaa tietty animaatiosekvenssi laskettavaksi usealle eri koneelle tai renderöintifarmille. Vray 2.0 asennetaan 3ds Max -ohjelmaan liitännäisenä. Liitännäinen pitää olla asennettuna jokaiselle koneelle, jolla laskentaa aiotaan suorittaa. (Chaos Group 2012)

Vray 2.0 -moottoria käytetään opinnäytetyössä, koska se on toimeksiantajalla käytössä pääsääntöisenä renderöintimoottorina. Vray:tä käytettiin opinnäytetyön tekniikkaosiossa referenssikuvan luomisessa.

2.2.5 Deep Exploration

Deep Exploration on Right Hemispheren kehittämä 3D-mallien visualisointiin ja kääntämiseen tarkoitettu ohjelmisto. Deep Explorationia käytetään projektissa mallien kääntämiseen CAD -formaattista 3ds Maxin ymmärtämään muotoon. Tiedostoja käännettäessä pystytään vaikuttamaan kriittisesti tulevan 3D-mallin geometrian laatuun sekä tarkkuuteen erityisillä tesselaatioarvoilla. Tesselaatiolla tarkoitetaan 3D-mallin pinnan jakamista kolmioihin. Tesselaatioarvo määrittää kolmioiden määrän eli polygonimäärän. 3D-mallin polygonimäärä vaikuttaa suoranaisesti siihen kuinka hyvin malli tulee toimimaan virtuaalimallissa. (Deep Exploration™ 5.7 Standard Edition 2012.)

3 3D-VISUALISOINNISTA

3.1 Yleistä visualisoinnista

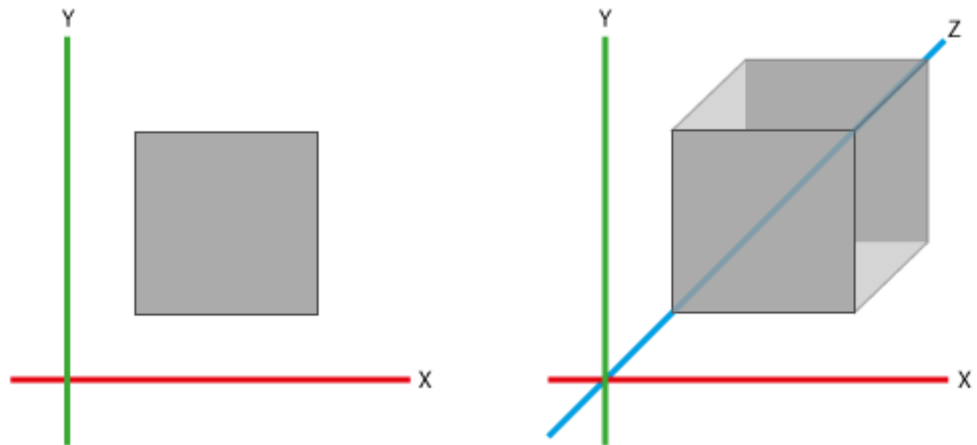
Visualisoinnilla tarkoitetaan jonkin asian havainnollistamista näköaistille. Tarkemmin määriteltynä visualisointi on tiedon esittämistä ihmisen henkilökohtaista ymmärrystä tukevaan muotoon. Visualisoinnin klassinen esimerkki on joko kuvio tai kaavio, jota käytetään kuvaamaan jotakin tiettyä tietoa mahdollisimman selkeällä tavalla. Tiedon selkeyttäminen on visualisoinnissa kuitenkin ongelmallista, sillä kohderyhmät usein ymmärtävät tiettyjä esitystapoja paremmin kuin toisia. (Ware 2004. 1-6.). Visualisoinnilla pyritään aina parantamaan tiedon ymmärrettävyyttä, joten on tarkkaan mietittävä, millä tapaa tiedon esittää, sillä esitystapa voi huomaamatta vääristää esitettäviä tuloksia. Hyvä esimerkki tällaisesta tilanteesta on pylväsdiagrammi, jonka pylväiden pohjatasoksi on määritelty jokin muu kuin nolla-arvo (Tufte 2004.)

Visualisointi toimii ihmiselle myös eräänlaisena apuvälineenä tiedon paikannukseen sekä siihen syventymiseen. Visualisointi ei siis ole pelkkiä kuvia tai kaavioita, vaan se parantaa tiedon painoarvoa ja auttaa löytämään yhä enemmän tärkeitä asioita visualisoidusta tiedosta. (Ware 2004, 317.)

Visualisointia käytetään havainnollistamisvälineenä päivittäin monissa eri tarkoituksissa. Opetuksessa visualisointi helpottaa opiskelijaa muistamaan opiskeltavan asian paremmin. Usein visualisointiin saatetaan liittää tehokeinoksi musiikkia, joka oikein käytettynä tehostaa visualisoinnin vaikutusta. Tieteessä visualisointia käytetään myös tavalliselle ihmiselle vaikeaksi ymmärrettävän tiedon selkeyttämisessä. Taloudessa ja yrityksissä visualisoinnilla pyritään ilmaisemaan yrityksen tarkoitusperiä mahdollisimman selkeästi: esimerkiksi markkinointi rakentuu usein vahvasti visualisointiin. Yrityksen visuaalinen ilme rakentuukin usein markkinoinnin ulosannin kautta, minkä vuoksi yritysten tulisi miettiä tarkkaan, minkälaisen kuvan he haluavat asiakkailleen näyttää. On syytä muistaa, että myös taiteilijan luoma taideteos on visualisointia, sillä taiteessa taiteilija pyrkii välittämään ihmisille tiettyjä tunnetiloja ja ajatuksia. (Tiedon visualisointi 2011.)

3.2 3D-visualisointi

Yksi visualisoinnin keinoista on 3D-visualisointi, jossa perinteiseen esitystapaan lisätään syvyyslementti. Käytännössä tämä tarkoittaa kolmannen, syvyys- eli Z- akselin lisäämistä visualisoinnin koordinaatistoon perinteisten horisontaalisen X- akselin ja vertikaalisen Y- akselin tueksi (kts. kuvio 1). Nämä kolme edellä mainittua akselia luovat niin sanotun 3D-avaruuden. (Goldstone 2009, 9.) 3D-avaruutta kuvaa parhaiten ihmiselle tuttu näkökenttä, jossa esiintyy aina kolme pääelementtiä: pituus, korkeus ja syvyys. Sijoittamalla objektit tähän tilaan saadaan luotua ihmiselle luonnollinen perspektiivi eli näkökulma.



KUVIO 1. Visualisointi syvyysakselin vaikutuksesta koordinaatistoon.

3D-visualisointi on mukana ihmisten jokapäiväisessä arjessa. Asiaan ei välttämättä kiinnitetä huomiota, mutta 3D-visualisointi on läsnä päivittäin kulutetussa viihteessä ja markkinoinnissa sekä epäsuorasti teollisuuden apuvälineenä tuotteiden suunnittelussa ja toteutuksessa.

3D-visualisointi avaa rajattomat mahdollisuudet visualisointiin, sillä tehokkaiden tietokoneiden avulla lähes minkä tahansa kuvan luominen on mahdollista. 3D-visualisoinnilla pystytään luomaan kuva jostakin, jota ei fyysisesti ole olemassa tai jota olisi mahdotonta visualisoida muutoin järkevästi objektin suuren koon tai hankalan sijainnin vuoksi. Esimerkkinä edellisestä toimii jokin vanha, jo tuhoutunut rakennus tai avaruusalus, jonka kuvaaminen sille luonteavassa ympäristössä olisi erittäin kallista tai jopa mahdotonta.

3D-visualisointi avaa rajattomat mahdollisuudet pelien ja elokuvien käsikirjoittajille ja teollisuuden tuotesuunnittelijoille. Tämä onkin yksi syy siihen, miksi elokuva- ja peliteollisuus ovat 3D-visualisoinnin suurkuluttajia perinteisen teollisuuden rinnalla. (Puhakka 2008, 23-24.)

3.3 3D -visualisointi teollisuudessa

Teollisuuden visualisoinnilla tarkoitetaan esimerkiksi suunnitelmien, piirustusten tai simuloinnin esittämistä. Teollisuuden visualisointi perustuukin usein jonkin tuotteen suunnittelun tai testauksen ympärille, milloin voidaan puhua niin sanotusta tietokoneavusteisesta suunnittelusta eli CADista. 3D CAD -malleja alettiin hyödyntää teollisuudessa jo 1980-luvulla, jolloin ensimmäinen 3D-mallin luomiseen suunniteltu CAD-sovellus julkaistiin. Tietokoneiden ja ohjelmistojen kehityksen tuloksena teollisuuden 3D-mallit ovat kehittyneet niin laadukkaiksi, että niitä pystytään nykyisin hyödyntämään koulutus- sekä markkinointimateriaalien tuotannossa.

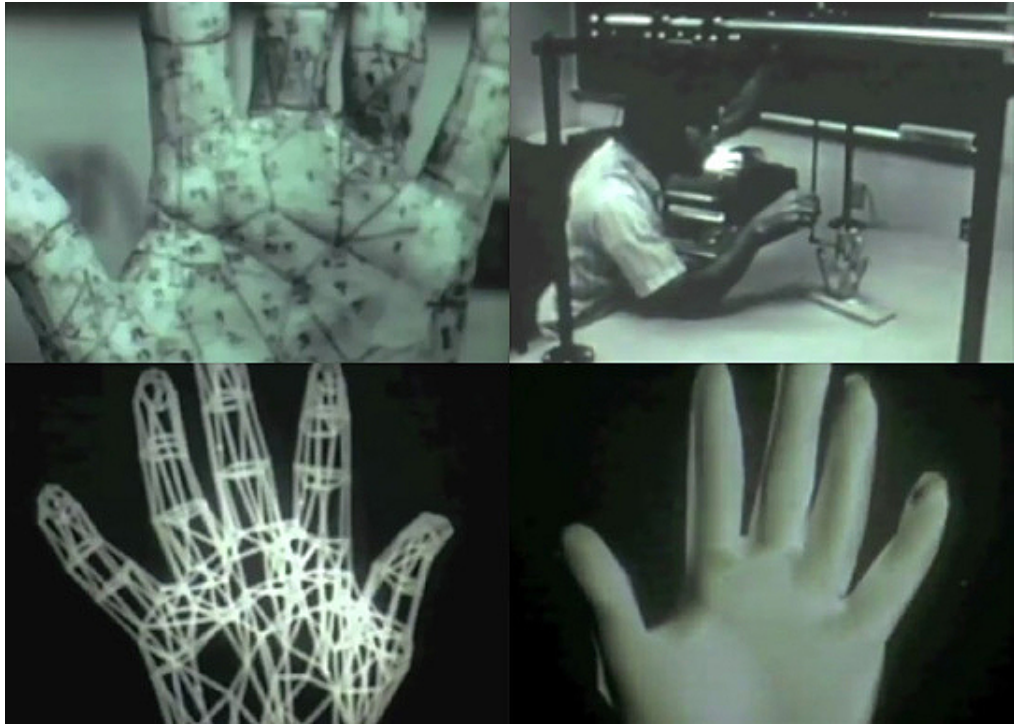
3.4 Animaatio ja sen historia

3D-malli on harvoin staattinen kappale. Usein sitä halutaan elävöittää liikkeellä, jolloin kappaleen liikettä kutsutaan animaatioksi. Animaatio on tekniikka, jossa elokuva toteutetaan kuva kovalta. Animaation toteuttamiseen on monia eri tapoja. Yksi vanhimmista tavoista on kuvien piirtäminen yksi kerrallaan, jolloin illuusio liikkeestä syntyy siitä, kun piirretyt kuvat esitetään peräkkäin tietyllä nopeudella.

Vuosikymmenten aikana on nähty useita eri tapoja tuottaa animaatiota. Yhdeksi kuuluisimmista tekniikoista on noussut J. Stuart Blacktonin vuonna 1906 vakiinnuttama Stop motion -tekniikka, jossa animaatio tuotetaan liikuttamalla staattisia esineitä vähän kerrallaan, ja kuvaamalla ne sitten filmille. Emil Cohl esitteli valtaväestölle animointitekniikan, joka tunnetaan piirrosanimaationa. Siinä kuvat tuotettiin piirtämällä ne käsin tai koneellisesti. Piirrosanimaatio nousi tärkeimmäksi animointitavaksi 1920-luvulla ja säilytti asemansa aina ensimmäisen 3D-animaation syntyyn asti. (The History of 3D Computer Animation 2010)

William Fetter lanseerasi termin "computer graphics" kehitellessään ensimmäisen 3D-ihmishahmon, The Boeing man:in, 1960-luvulla. Vuonna 1976 nähtiin 3D-animaatiota ensimmäistä kertaa elokuvassa Futureworld, jota varten Ed Catmull mallinsi ja animoi ihmisen käden ja kasvot (kts.kuvio 2). Futureword käynnisti uuden aikakauden. Uutta tekniikkaa hyödynnettiin useissa eri elokuvissa ja mainoksissa, minkä seurauksena 3D-animaatio alkoi vakiinnuttaa paikkaansa animaatioteollisuudessa. (The History of 3D Computer Animation 2010)

Vuonna 1979 Lucasfilmsin tuotantoyksiköksi perustettu Pixar on ollut yksi merkittävimmistä vaikuttajista 3D-animaation kehityksen historiassa. 1970-luvun lopulla Pixar alkoi kehittää ohjelmistoja ja erikoistehosteita yhdessä elokuvien erikoistehosteita tuottavan Industrial Light & Magicin kanssa. Steve Jobs osti Pixarin vuonna 1986 ja alkoi panostaa yrityksen laitevalmistukseen. Tämän seurauksena Pixar julkaisi oman tietokoneensa, Pixar Image Computerin. Tietokoneen myynti ei vastannut yrityksen odotuksia, joten Pixar antoi työntekijälleen John Lasseterille toimeksiannon luoda esittelyvideo Luxo Jr:n (kts. kuvio 3), jolla havainnollistettaisiin tietokoneen tehokkuutta. Pixarin tietokoneen heikko menestys jatkui, mutta Lasseterin luoma esittelyvideo herätti kiinnostusta suuremmassa yleisössä. Lopulta kiinnostus poiki yhteistyösopimuksen Walt Disneyn kanssa. Pixarin ja Disneyn yhteistyöstä syntyi menestyselokuva Toy Story, joka on toiminut innoittajana monille muille täyspitkille 3D-animaatioille. (Our story 2012)



KUVIO 2. Edvin Catmullin mallinnus ihmiskädestä. (A Computer animated hand 2004)



KUVIO 3. Kohtaus John Lasseterin Luxo JR lyhytelokuvasta (Luxo JR 2007)

3D-animaatiota voidaan tuottaa monella eri tapaa. Tietyn objektin eri osia tai pisteitä voidaan animoida joko käsin tai osittain automatisoidusti. Ilman ulkopuolista animointitietoa voidaan tuottaa animaatiota niin sanottujen keyframejen avulla. Keyframe't eli avainruudut ovat tärkeitä ajanhetkiä animaatioissa. Animaattori määrittelee animoitavalle objektille alku- ja loppuavainruudut, minkä jälkeen tietokone laskee näiden pisteiden välisen liikkeen. Animaatiota voidaan tuottaa myös kaappaamalla reaali maailman liikettä, milloin puhutaan motion capture -tekniikasta. Motion capture'ssa eli liikkeen kaappauksessa näyttelijän liike tunnustetaan ja kaapataan tietokoneelle, minkä avulla dataa voidaan käyttää animaation tuottamiseen.

3.4.1 Interaktiiviset ja reaaliaikaiset 3D-animaatiot

3.4.2 Yleistä

Reaaliaikaista 3D-animaatiota käytetään interaktiivisessa mediassa, kuten peleissä ja simulaatioissa. Tällaisissa sovelluksissa käyttäjä on vuorovaikutuksessa piirrettävän kuvan kanssa. Käyttäjä voi siis itse määrätä animaation kulun samalla kun animaatiota esitetään. Tavoitteena reaaliaikaisessa 3D-animaatioissa on näyttää aina vähintään se määrä kuvia, minkä ihmissilmä vaatii illuusioon sulavasta liikkeestä.

Kuvien esittämisen arvoa kuvataan ja mitataan kuvina sekunnissa (fps) tai hertzeinä (Hz). Yhden kuvan sekuntinopeudella käyttäjän on vaikea havaita interaktiivisuutta, todennäköisesti huomio keskittyy vain seuraavan kuvan odottamiseen. Nopeutta kasvattamalla kuuteen kuvaan sekunnissa interaktiivisuuden tunne alkaa kasvaa. Saavutettaessa viidentoista kuvan sekuntivauhtiin voidaan puhua reaaliaikaisesta animaatiosta, jossa käyttäjä pystyy reagoimaan ja toimimaan hänelle näytetyn animaation mukaisesti. On syytä tiedostaa, että usean kuvan samanaikainen näyttäminen vaatii paljon laskentatehoa tietokoneelta. Tehontarve kasvaa lineaarisesti, mikäli näytettävien

kuvien määrää kasvatetaan. Tehokkainta onkin pyrkiä näyttämään 15-72 kuvaa sekunnissa, sillä tätä ylittävää osaa ei voida enää ihmissilmällä havaita. Nykyisin grafiikkapiiri hoitaa 3D-animaation renderöinnin. Aikaisemmin se jäi keskussuorittimen laskettavaksi. Nykytekniikka mahdollistaa siis suurempien tietomäärien käsittelyn, mikä johtaa realistisempaan renderöintitulokseen.

Reaaliaikaisen 3D-animaation tehokeinoina voidaan käyttää linssiheijastuksia, liikesomeutta ja syvyysterävyttä. Näillä efekteillä pyritään tuomaan animaatioihin ihmissilmälle tuttua realismia. Reaaliaikainen kuvanmuodostus onkin parantunut huomattavasti grafiikkapiirien kehittyessä. (Akenine-Möller, Haines, Hoffman 2008, 1 - 4.)

3.4.3 Pelimoottorit

Interaktiivinen animaatio vaatii toimiakseen ohjelmarungon, joka toimii rajapintana käyttäjän ja 3D-maailman välillä. Pelimoottorit tarjoavat joustavan ja uudelleenkäytettävän ohjelmistorajapinnan, jota ohjelmoijat käyttävät luodessaan interaktiivisia pelejä ja sovelluksia. Pelimoottori tarjoaa ydintoiminnallisuuden sovellukselle. Näitä toimintoja ovat tyypillisesti kuvan luonti, fysiikkamoottori, muistin hallinta ja monet muut sovelluksen toiminnallisuuden perusominaisuudet.

Pelimoottorit voidaan jakaa kolmeen ryhmään: kaupallisiin, avoimen lähdekoodin sekä itse tuotettuihin pelimoottoreihin. Kaupalliset pelimoottorit pyritään luomaan mahdollisimman monikäyttöisiksi niin, että niitä pystytään hyödyntämään useissa eri sovelluksissa ainoastaan sovelluksen käyttölogiikkaa ja ulkonäköä muuttamalla. Pelin kehittäjä ei siis voi vaikuttaa pelimoottorin rakenteeseen, ja tämä on otettava huomioon sovellusta rakennettaessa. Avoimen lähdekoodin pelimoottorit poikkeavat kaupallisista moottoreista siten, että pelinkehittäjä voi halutessaan muokata pelimoottorin rakennetta sopimaan omiin tarkoituksiinsa. Avointen ja kaupallisten pelimoottorien eduiksi

katsottakoon niiden nopea käyttöönotto, joka jättää aikaa muulle sovelluskehitykselle. Täysin itse luotu pelimoottori on toimivin ratkaisu, mikäli valmista, omaan käyttöön soveltuvaa pelimoottoria ei löydy. Tällöin pelimoottorin kaikki ominaisuudet saadaan vastaamaan nimenomaisesti käyttäjän omaa tarvetta. Pelimoottorin luominen tyhjästä vaatii resursseja ja näin ollen sitä on aina tarkoin harkittava. (Game Development Team Composition Study 2003.)

4 3D–MALLINNUS- JA -VISUALISOINTITEKNIIKAT

4.1 Käytänteet 3D-projektissa

On muutamia toimintatapoja, jotka on syytä oppia tuntemaan toimittaessa 3D-mallien kanssa. Vakiintuneiden toimintatapojen tärkeys korostuu silloin, kun työskennellään isommissa tuotantotiimeissä useiden eri animaattorien ja mallintajien kanssa. Mitä aikaisemmin nämä tekniikat ja käytänteet oppii, sitä helpommaksi työskentely mallien kanssa ajan myötä muuttuu.

3D-projektin johdonmukaisuutta ja prosessia helpottavat käytänteet:

1. Hanki referenssimateriaalia, tiedä mitä olet tekemässä.
2. Järkeistä ja vakiinnuta nimeämiskäytänteet.
3. Muista tallentaa riittävän usein.
4. Kysy muiden mielipidettä tuottamastasi työstä.
5. Kirjaa ylös prosessiin liittyviä tärkeitä asioita.

Referenssimateriaali on yleensä yksi tärkeimmistä elementeistä 3D-mallia käsiteltäessä. Referenssimateriaalilla tarkoitetaan jotakin kuvaa tai sen osaa, johon tulevilla 3D-visualisoinnilla pyritään. Referointimateriaalia on syytä kerätä jatkuvasti, ei pelkästään jonkin projektin tiimoilta vaan jokapäiväisen havainnoinnin kautta. Esimerkiksi tietämys siitä, miltä jokin puu tai metalli näyttää ja tuntuu, auttaa sinua visualisoimaan saman asian 3D:llä. Projektia työstäessä on syytä selvittää, minkälaista referenssimateriaalia tarvitaan tietyn asian visualisoimiseksi 3D:llä. Esimerkiksi projektissa, jossa asiakas toimittaa visualisoitavaksi 3D-mallin tuotteesta, olisi hyvä pyytää referenssimateriaaliksi jo olemassa olevia kuvia laitteesta - tai vaihtoehtoisesti kokonainen laite - mikäli se on mahdollista. Epäselvyyksien välttämiseksi asiakkaalta tulisi myös tiedustella tuotteen maalauksessa käytettyjen värien sävykoodeja sekä esimerkki tuotteen mittasuhteista. Referenssimateriaalin keräykseen voidaan laskea myös asiakkaan toiveet animaatioon. Asiakas voi esimerkiksi kertoa tuotteesta tiettyjä kohtia, joita halutaan korostaa. Toisaalta asiakas voi myös esittää toiveen siitä, mitä animaatiossa tulee jättää näyttämättä. Asiakalta kerätyn referenssimateriaalin perusteella voidaan luoda käsikirjoitus tai esimerkkikuva siitä, mihin tulevilla visualisoinnilla pyritään. (Chopine2011, 13-19.)

Nimeämiskäytänteillä tarkoitetaan tiedostojen ja objektien nimeämistä projektin edetessä. Nimeämiskäytänteet tulee pitää mahdollisimman loogisena koko projektin keston ajan. Nimeämiskäytänteiden tärkeys tulee esille erityisesti silloin, kun projektiryhmän koko kasvaa, ja tiedostoja vaihdetaan tiheästi animaattoreiden, mallintajien ja ohjelmoijien kesken. Loogisesti nimetyn projektin rakenne selviää helposti ja siihen on helppo päästä mukaan. Tämän hyödyn voi myös itse todeta palaamalla myöhemmin vanhaan, hyvin nimettyyn projektiin, johon on helppo päästä sisään. Esimerkki asiakasprojektin nimeämisestä: Projektissa visualisoidaan autoa, jonka merkki on kuvitteellisesti VOLTZ ja malli Z2. 3D -projektitiedoston nimi voisi tällöin olla Voltz_z2_Versionumero_PPKKVV.Max. 3D-projektitiedoston sisältä löytyvä auton geometriaobjekti voidaan nimetä Voltz_z2. Auton geometriaobjektin sisältä

löytyvä auton rengas voidaan puolestaan nimetä
Voltz_z2_Rengas_vasen_etu. (Chopine2011, 17.)

3D-ohjelmat ”kaatuilevat” jatkuvasti, minkä vuoksi on syytä muistaa tallentaa työstämänsä materiaali riittävän usein. Ihannetilanne olisi kytkeä päälle automaattinen tallennus, joka tallentaa tehdyn työn tietyin aikaväleihin samalle tiedostonimelle. Raskaiden 3D-mallien tallentaminen saattaa joskus kestää minuutteja, joten tällaisissa tilanteissa automaattisen tallennuksen käyttö ei ole järkevää. Raskaiden mallien kohdalla tulisi tiedosto tallentaa aina, kun jokin merkittävä muutos on tehty. Tallennusten välille tulisi määritellä maksimiaika, esimerkiksi yksi tunti. Mikäli projektissa tehdään suuria muutoksia, olisi järkevää tallentaa työtiedosto uudelle tiedostonimelle. Tällainen nimeäminen voidaan toteuttaa esimerkiksi versionumeroinnilla. Uudelle nimelle tallennus kannattaa muutoinkin suorittaa päivittäin. Tällaista tallennuskäytäntöä toteuttamalla saadaan aikaan eräänlainen versionhallinta, jonka avulla projektissa voidaan palata taaksepäin, mikäli jokin menee pieleen. (Chopine2011, 17.)

Kritiikki auttaa kehittymään, vaikka sen vastaanottaminen voi joskus olla vaikeaa. 3D-projektissa - samalla tavoin kuin muissakin luovissa projekteissa - tulee pyytää mielipiteitä ja kommentteja kollegoilta ja asiakkaalta. Kommenttien avulla voidaan korjata mahdollisia virheitä, joita tekemäsi työhön on huomaamattasi jäänyt. Näitä virheitä syntyy, kun projektia luodessa ihmissilmä tottuu näkemäänsä, eikä enää havaitse tehtyjä virheitä. Hyviä kanavia kommenttien pyytämiseen ovat asiaan perehtyneet yhteisöt ja foorumit, työyhteisö ja asiakas. Kaikki kommentit kannattaa kirjata ylös, ja kritiikin keräämisen jälkeen on syytä suodattaa kritiikistä tärkeimmät asiat, joihin reagoidaan.

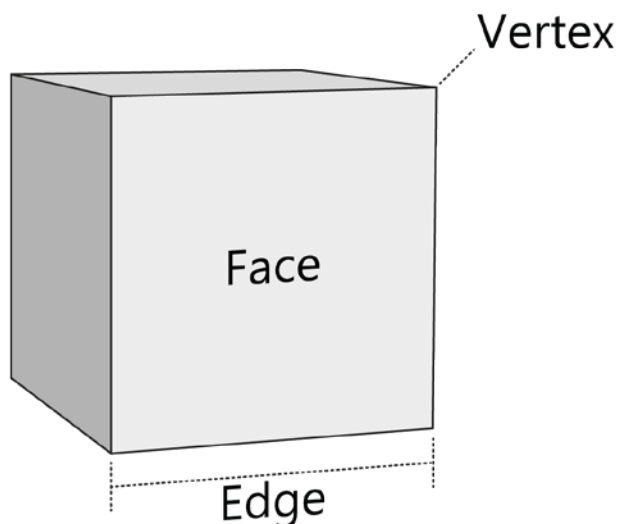
3D-projektissa yksi tärkeimmistä työkaluista on kynä ja muistivihko. On järkevää kirjata ylös kaikki se, mitä itse luokittelee tärkeäksi tiedoksi projektin etenemisen kannalta. Mikäli mahdollista, olisi hyvä kantaa mukana muistiinpanovälineitä lähes joka paikassa. Oman ideoinnin auttamiseksi voi kirjata muutamalla sanalla tai nopealla piirroksella asioita, jotka inspiroivat jokapäiväisessä elämässä.

Paperille tulee kirjata myös asiakkaan nimi ja yhteystiedot sekä kaikki päivämäärät projektin etenemisen suhteen. Nämä asiat helpottavat jokapäiväistä työtäsi. (Chopine 2011, 13-19.)

4.2 3D-geometria

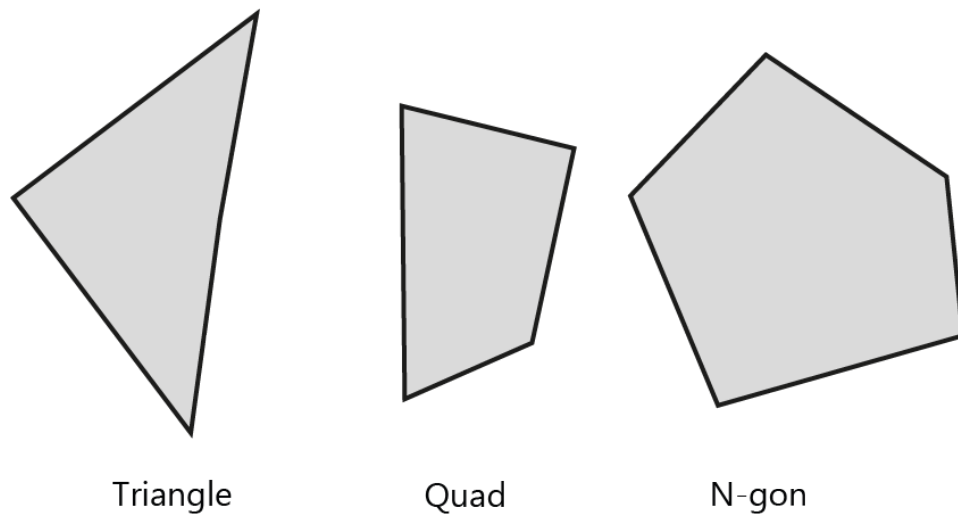
4.2.1 Geometrian rakenne

3D-ohjelmassa voidaan luoda niin sanottuja primitiiviobjekteja, jotka voivat olla kuutioita, palloja tai sylintereitä. Nämä kaikki objektit rakentuvat polygoneista. Helpoiten objektien rakenteen saa havainnollistettua ottamalla käsittelyyn kaikille tutun objektin, kuution. Kuution neliönmuotoiset sivut ovat polygoneja. Polygonien sisälle jääviä osia nimitetään faceiksi. Kuution reunaviivoja kutsutaan edgeiksi, ja kuution kulmien kärkipisteitä kutsutaan vertexipisteiksi (kts kuvio 4). Nämä polygonin osat voidaan jakaa myös niiden ulottuvuuksien mukaan: facella on aina pituus- ja leveysulottuvuus, edgellä leveysulottuvuus, vertexillä ei yhtään ulottuvuutta. Näiden pisteiden muodostamalla kuutiolla on leveys-, syvyys- ja korkeusulottuvuus.



KUVIO 4. Polygonin rakenne kuutiolla esitettynä

Polygoni voi rakentua rajattomasta määrästä vertexipisteitä ja edgejä, minkä vuoksi sen ei aina tarvitse olla neliö. Vähäisimmillään polygoni voi rakentua kolmesta vertexipisteestä ja edgestä, mitkä muodostavat kolmion. Kolmiot ovat yksinkertaisimpia polygoneja, joten ne ovat tietokoneelle helpoimpia objekteja käsitellä. Neljän vertexipisteestä muodostamaa polygonia kutsutaan Quadiksi ja kaikkia yli neljän vertexipisteestä muodostamia polygoneja kutsutaan n-goneiksi (kts. kuvio 5). Mitä enemmän polygonissa on vertexipisteitä, sitä raskaammaksi sen laskeminen tietokoneelle muodostuu. (Chopine, A. 2011. 22)



KUVIO 5. Esimerkkejä erilaisista polygonirakenteista

Aikaisemmin mainitussa kuutioesimerkissä voidaan huomata, että kuution faceilla on jaettuja edgejä ja vertekspisteitä. Nämä jaetut vertekspisteet liittävät kuution eri pinnat yhtenäiseksi verkoksi. Tätä verkkoa kutsutaan polygonverkoksi. 3D-mallin pinta muodostuu tästä verkosta. Kuutio on hyvä esimerkki suljetusta polygoni verkosta, sillä sen kaikki reunat on jaettu facejen kesken. Avoin polygoniverkko voi olla esimerkiksi kahdesta quadista muodostuva mittaamattoman ohut paperiarkki, jonka edget muodostavat kappaleen reunat. (Chopine 2011, 21-25.)

4.2.2 Normaalivektori

Normaalivektori on vektori, joka osoittaa kohtisuoraan polygonin etupintaan nähden. Mikäli vektori kuvattaisiin pallomaisen polygoniobjektin pinnassa, pallo näyttäisi piikikkäältä. Tämä johtuisi siitä, että jokaisen polygonin keskellä olisi suora viiva ylöspäin kuvaamassa vektoria. Normaalivektori on tärkeässä asemassa valonjäljityksessä. Normaalivektoreita ei tarvitse erikseen määritellä, vaan ne luodaan automaattisesti. Tavallisesti normaalivektoria ei näy. On kuitenkin hyvä muistaa, että normaalivektorilla voi olla kaksi suuntaa, sisään- tai ulospäin. Tämä johtuu polygonin sisä- ja ulkopinnasta. Vektorin suunta voi muuttua jos polygonin pinta on väärinpäin. Väärään suuntaan osoittavan polygonipinnan huomaa helpoiten siitä, ettei se ota vastaan valoa, koska sen normaalivektori osoittaa muuhun polygoniverkkoon nähden sisäänpäin. (Puhakka 2008, 41-43.)

4.2.3 Yksinkertainen ja monimutkainen geometria

3D-geometria voidaan jakaa kahteen pääryhmään: monimutkaiseen eli tarkkaan ja yksinkertaiseen eli epätarkkaan geometriaan. Näistä kahdesta geometriatyypistä käytetään nimityksiä highpolygon ja lowpolygon, viitaten niiden sisältämään polygonimäärään. Esimerkkinä matalalla polygonimäärällä tietyt muodot menettävät yksityiskohtia ja alkavat näyttää kulmikkailta. Kaikki lowpolygon-mallit eivät kuitenkaan näytä huonolaatuisilta, sillä niiden parantamiseen on kehitetty normal mapping sekä bump mapping -tekniikat. Käytetty renderöintitapa ja haluttu detailitaso ovat siis aina huomioitavia muuttujia, kun puhutaan low ja high polygon -malleista. (Finney 2006, 415-418.)

4.3 Teksturointi

4.3.1 Shaderit

Shaderit eli varjostustekniikat tuovat 3D-mallin pintamateriaaliin syvyyttä. Kuvitellaan esimerkiksi pallo, joka on tasavärinen joka puolelta. Tasavärinen, varjostamaton pallo näyttää litteältä 2D-kuvalta. Kun tällaista objektia varjostetaan, saadaan sen pintamateriaaliin syvyyttä ja realismia. Varjostetun objektin väri muuttuu niin, että lähimpänä valonlähdettä oleva kohta näkyy puhtaimpana ja tummenee aina liikuttaessa valonlähteestä poispäin.

Varjostus voidaan laskea useilla eri algoritmeilla, perustuen polygonin tai normalin asentoon. Lähes kaikista 3D-ohjelmista löytyviä shadereita ovat Lambert-, Gouraud-, Phong- ja Blinn-shaderit. Varjostustavat muuttavat pääsääntöisesti varjostuksen tasaisuutta ja kiillon yksityiskohtaisuutta. Lambert-shading on varjostustavoista yksinkertaisin. Siinä tarkastellaan kunkin polygonin asentoa valonlähteeseen nähden (kts kuvio 6.). Kehittyneemmissä varjostustekniikoissa, kuten Phongissa ja Blinnissa, käytetään useampia muuttujia. Näin saadaan tasaisempi ja aidompi lopputulos. (Chopine 2011, 139-143.)



Lambert



Phong

KUVIO 6. Esimerkki Lambert- ja Phong-shadereiden eroista

4.3.2 Tekstuuri

Tekstuurissa määritellään 3D-objektin pinnan väri, kiilto ja tasaisuus. Tekstuuri voi olla tasainen väri, bittikarttakuva tai näitä kahta yhdistelemällä tuotettu proseduraalinen kuvio. Tekstuurissa voidaan myös määritellä pinnan epätasaisuus normaali-, displacement- tai bump-karttojen avulla. Näitä karttoja käytetään näyttävyyden lisäämiseen 3D-malliin muuttamatta sen polygonirakennetta.

(Chopine 2011, 151.)

4.3.3 Tekstuuri ja UV-mapping

Koska tekstuurit ovat 2D-kuvia, täytyy niiden asettuminen 3D-objektin pinnalle määritellä. Asettuminen voidaan määritellä texture- tai UV-mapping -tekniikoilla. Texture mapping -tekniikassa tekstuuri projisoidaan objektin pinnalle. Projisoinnilla tarkoitetaan tekstuurin heijastamista 3D-objektin pintaan. Tekniikkaa voidaan helposti ymmärtää ajattelemalla videotykillä seinään heijastettua kuvaa. Heijastettava kuva on tekstuuri ja seinä toimii tässä tapauksessa tasaisena 3D-objektina. Tekstuurikartta voidaan heijastaa myös muihin kuin tasaisiin objekteihin. Kuutiomaisessa 3D-objektissa tekstuuri jaetaan ensin kuuteen osaan ja heijastetaan sitten objektiin, jolloin tekstuuri saadaan asettumaan oikein kuution jokaiselle sivulle. Texture mapping -tekniikka toimii hyvin yksinkertaisille objekteille ja onkin oikein käytettynä erittäin tehokas tapa määrittää tekstuurien asettumista 3D-objekteille. Vaativille, paljon yksityiskohtaisia ja voimakkaita muotoja omaaville objekteille on käytettävä UV-mapping tekniikkaa. UV-mapping tekniikassa käytetään koordinaatteja tekstuurin asettumisen määrittelyyn. Vaikka kyse on 3D-objektista, ei tekstuurin määrittelyyn käytetä XYZ-koordinaatteja. Määrittely tehdään käyttämällä horisontaalisia U- ja vertikaalisia V-leveysasteita. UV-koordinaatteja määriteltessä 3D-objektin pinta täytyy tasoittaa. Tasoittaminen on yksinkertaista puhuttaessa polygonigeometriasta. Jokainen face täytyy sijoittaa tasaiselle 2D-pinnalle. Manuaalinen tasoittaminen muuttuu vaikeammaksi mallin geometrian muuttuessa monimutkaisemmaksi. Normaalisti tasoittamisessa käytetään niin

sanottuja unfold- ja unwrap-tekniikoita, jotka automaattisesti kuorivat 3D-objektin pinnan ja tasoittavat sen UV-kartaksi. (Chopine 2011, 152-153.)

4.3.4 Bump-, Normal-, Displacement- ja Transparency-kartat.

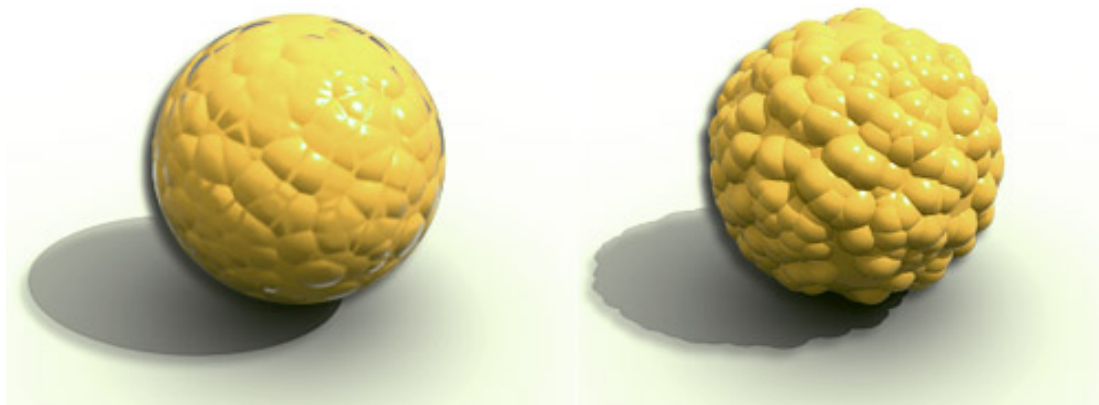
3D-mallinnuksessa tai optimoinnissa ei jokaista pientä yksityiskohtaa voida mallintaa geometriatasolla. Mikäli näin tehtäisiin, 3D-mallin polygoni määrä nousisi liian suureksi. Pienten yksityiskohtien esittämiseen on kehitetty tekniikka, jolla pystytään visualisoimaan pinnan epätasaisuutta vaikuttamatta 3D-mallin polygonimäärään. Tekniikka hyödyntää bittikarttoja pienten yksityiskohtien lisäämiseen malliin kuvanmuodostus- eli renderöintivaiheessa.

Näissä kartoissa jokainen väri kontrolloi tiettyä arvoa. Esimerkiksi mustalla viitataan arvoon yksi, kun taas valkoisella viitataan arvoon nolla. Kahdella luvulla ja niiden välimuodoilla saadaan kuvattua useita eri vaihteluita, kuten kohouma tai painauma, läpinäkyvä tai läpinäkymätön. Nämä kaksi ääreisarvoa riittävät useimmissa - ei kuitenkaan kaikissa - karttatyypeissä muodostamaan illuusion geometrian muutoksesta. Tästä syystä jokainen karttatyyppi tulee avata tapauskohtaisesti. (Chopine 2011, 158.)

Bump-kartta on korkeuden ja syvyyden kuvaukseen käytetty musta-valkokartta. Valkoinen on kartan korkein ja musta matalin kohta. Tietokone muodostaa illuusion syvyyden muutoksesta mustavalkokartan ja geometrian normaalien avulla. (Chopine, A. 2011. 159) Bump-kartta muodostaa illuusion geometrian muutoksesta ainoastaan 3D-mallin pintaan, se ei tee muutoksia varjoihin eikä siluettiin (kts kuvio 7). Bump-kartta soveltuu parhaiten pienen epätasaisuuden, kuten appelsiinin tai golfpallon karheuden visualisointiin. (Akenine-Möller, Haines, Hoffman 2008, 183-184.)

Displacement-kartalla on paljon yhtäläisyyksiä verrattuna bump-karttaan: molemmissa illuusio syvyyden muutoksesta luodaan mustavalkokartalla. Suurin eroavaisuus näiden kahden välillä on se, että displacement-kartta vaikuttaa geometriaan. Displacement-kartan vaikutus geometriaan näkyy selkeiden 3D-

mallin muodostamassa varjossa ja siluetissa. (kts kuvio 7) (Chopine 2011, 160.)



KUVIO 7. Vasemmalla pallo bump kartalla, oikealla displacement kartalla. (Displacement mapping 2010)

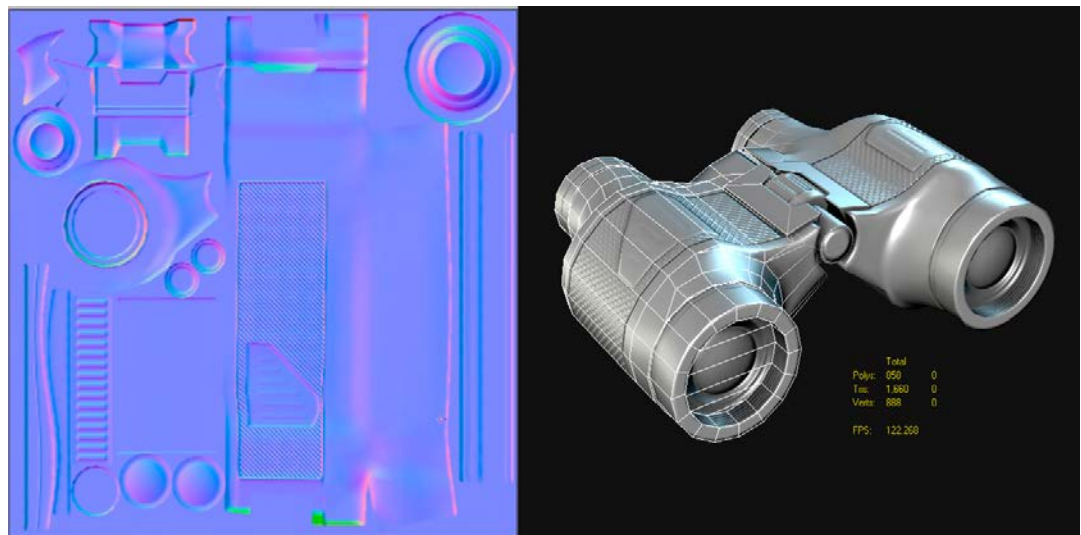
Normal-kartta poikkeaa bump- ja displacement-kartoista sisältämällä informaation kaikilta kolmelta akselilta. Normal-kartta käyttää red-, green- ja blue- arvoja, eli RGB-arvoja X-, Y- ja Z -akselien simulointiin. Näin ollen normal-kartta sisältää kolme kertaa enemmän tietoa kuin mustavalkokuvat, joita bump- ja displacement-kartoissa käytetään. Lisääntyneen tiedon myötä voidaan 3D-maailmassa aina muodostaa tarkka illuusio pinnan muodosta kameralle. Normal-karttaa käytetään erityisesti reaaliaika sovelluksissa, kuten peleissä. Karttaa käytetään näissä sovelluksissa, koska se on tietokoneelle helpoin ja laadukkein tapa muodostaa pinnan epätasaisuutta nopeasti.

Normal-kartta muodostetaan yleensä high-polygon -mallista, minkä jälkeen se annetaan tekstuuriksi low-polygon -mallille. Näin vähän geometriaa sisältävä malli saadaan näyttämään korkeatasoiselta (kts. kuvio 8). Kartan muodostamiseen käyteeään high-polygon mallin UV-kartan RGB-kanavia. Molempien mallien UV-koordinaattien tulee olla samanlaiset, jotta kartta asettuu

oikealle paikalleen low-polygon mallille. (Akenine-Möller, Haines, Hoffman 2008, 187-190.)

Normal-karttoja on useita eri tyyppisiä. Niiden muodostus, käyttö ja laskentatapa vaihtelevat riippuen siitä, käsitelläänkö normaaleja tangentin, objektin vai 3D-maailman koordinaatistossa.

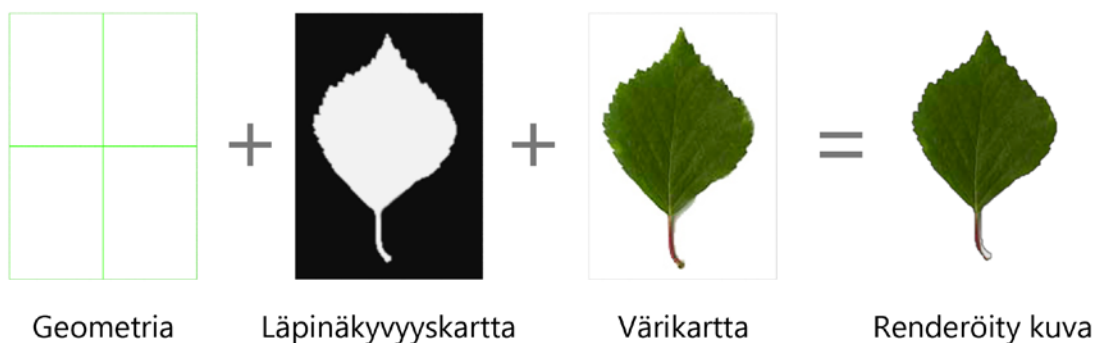
Tässä opinnäytetyössä käsitellään ainoastaan tangenttiin perustuvaa normal-kartan muodostusta ja käyttöä, sillä se on yleisemmin käytössä reaaliaikasovelluksissa.



KUVIO 8. Tangenttiin perustuvalla normal-kartalla teksturoitu kiikari. (Normal maps 2009.)

Transparency- eli läpinäkyvyyskartta vaikuttaa objektin läpinäkyvyyteen. Sitä voidaan käyttää tekstuuripinnoissa tiettyjen asioiden piilottamiseen ja näyttämiseen. Kartta on mustavalkoinen. Musta osa on täysin läpinäkyvä ja valkoinen läpinäkymätön. Näiden kahden ääreisarvon väliin jäävät harmaansävyt ovat läpinäkyvän- ja näkymättömän välimuotoja. Läpinäkyvyyskarttaa hyödynnetään usein tasaisten objektien, esimerkiksi lehtien, visualisointiin. Lehden geometria on tasainen, neljä polygonia sisältävä levy (kts. kuvio 9). Levyyhyn on laitettu tekstuuri, jossa läpinäkyvyyskartta piilottaa värikartan sekä

geometrian ylimääräiset osat. Näin syntyy illuusio tarkkaan mallinnetusta lehdestä. Tämän tyyppisiä, erittäin kevyitä malleja voidaan usein hyödyntää tuomaan helposti ja kevyesti lisää yksityiskohtia kohtauksen taustalle. (Chopine 2011, 160-161.)



KUVIO 9. Läpinäkyvyyskartan vaikutus geometriaan ja tekstuuriin.

4.4 Valaistus

4.4.1 Säteenjäljitys

3D-visualisoinnissa pyritään jäljittelemään reaalimailman fysikaalisia ilmiöitä myös valon osalta. Tällaisia ilmiöitä ovat esimerkiksi valon heijastuminen ja taittuminen. Reaalimailmassa jokaisesta valoa säteilevästä kappaleesta lähetetään iso määrä valonsäteitä, jotka heijastuvat sekä taittuvat matkalle osuvista pinnoista. Aina, kun valo taittuu tai heijastuu, sen voimakkuus pienenee ja väriarvo muuttuu edellisen pinnan mukaan, kunnes se saavuttaa katsojan silmän. Jokaisen säteen seuraaminen on kuitenkin valitettavan hidas tapa laskea valaistusta. 3D-visualisoinnissa käytetään usein säteenjäljitystä.

Säteenjäljityksessä valon etenemistä tarkastellaan päinvastaisessa järjestyksessä,

ihmissilmästä valonlähteesen. Näin on mahdollista jättää laskematta kaikki se, mitä katsojan hypoteettinen silmä ei näe.

Säteenjäljityksessä lasketaan kokonaisvalaistusta, mukaan lukien suorat ja heijastuneet valonlähteet. Tätä nimitetään globaaliksi valaistukseksi. Globaali valaistus perustuu kahden algoritmin vuorovaikutukselle; säteenjäljityksen heijastukselle ja radiositeetin diffuusiolle. (Puhakka 2008 406-408)

4.4.2 Valotyypit

Valon visuaaliseen ilmeeseen vaikuttaa voimakkaasti valonlähteen tyyppi.

Yleisesti käytettyjä valonlähteitä ovat:

1. Pistevalo (pointlight)
2. Spottivalo (spotlights)
3. Suunnatut valot (directional lights)
4. Aluevalot (area lights)
5. Lineaariset valot (linear lights)
6. Ympäröivä valaistus (ambient lighting)

Pistevalo säteilee valoa tasaisesti kaikista suunnista. Tätä valoa voidaan verrata reaaliympäristön hehkulamppuun, sillä pistevalot ovat 3D-maailman yksinkertaisimpia valotyyppisiä. Niiden laskenta on kuitenkin hidasta, sillä ne säteilevät valoa joka suuntaan.

Spottivalo säteilee valoa vain sille annettuun suuntaan muodostaen kartiota muistuttavan valokeilan. Spottivalon valokeilan kulma voi vaihdella, jolloin valaistavaa kohdetta saadaan valaistua tarvittaessa enemmän tai vähemmän.

Reaalimaailmassa spottivaloa muistuttavat eniten taskulampun sekä teatterivalaistuksen keilamaiset valot.

Suunnatun valon säteet kohtaavat kohteensa usein rinnakkain. Edellämainitun käyttäytymisen johdosta valoa hyödynnetään kuvaamaan kaukana olevia valonlähteitä, kuten tähtiä. Aurinko on eniten käytetty suunnattu valonlähde.

Aluevalot ovat käytännöllisiä pienten alueiden valaisuun. Aluevalot ovat yleensä neliömäisiä, pehmeää valoa tuottavia valonlähteitä. Aluevaloja voidaan käyttää luomaan pehmeää, valokuvaustudiomaista valaistusta.

Lineaarisella valonlähteellä tarkoitetaan pitkää valonlähdettä, jolla jäljitellään reaalimaailman loisteputkivalaistusta.

Ympäröivällä valaistuksella tarkoitetaan valoa, joka säteilee tasaisesti kaikista suunnista. Näin ollen se ei muodosta realistista varjoa. Ympäröivää valaistusta käytetään määrittämään tietyn kohtauksen valaistuksen tasoa. Näin ollen sitä ei juurikaan käytetä ilman muita valaistustyyppiejä. (Kerlow 2009, 226-233.)

4.5 Kamerat

4.5.1 Kameratyypit

Vaikka kamerat eivät varsinaisesti liity 3D-mallin optimointiin, täytyy ne kuitenkin käydä läpi, sillä 3D-maailmassa kuvanmuodostus perustuu aina siihen, että jotakin tiettyä tilaa katsotaan kameran läpi. Tietyntyypisiä kameroita voidaan animoida ja liikuttaa ohjelmasta toiseen. Optimoinnissa voidaan luoda kamera referenssikuvaa varten ja käyttää samaa kameraa reaaliaikasoveluksessa. Näin saavutetaan paras vertailukohta kahden renderöintituloksen välille. 3D-maailmassa voi olla useita eri kameratyyppiejä eri käyttötarkoituksia varten. Yleisimmin käytettyjä kameratyyppiejä ovat vapaa ja kohdennettu kamera. Nämä kameratyyppit löytyvät molemmista tässä opinnäytetyössä käytettävistä

sovelluksista. Vapaalla kameralla ei ole niin sanottua kohdepistettä, joten se täytyy aina suunnata erikseen. Vapaata kameraa pystyy kääntämään rajoittamattomasti jokaiseen suuntaan koordinaatistossa. Kohdennettu kamera katsoo aina johonkin tiettyyn pisteeseen, jota merkitään symbolilla. Symbolia liikuttaessa kameran positio kestää samana, mutta sen kääntökulma eli rotaatio muuttuu. Muilta asetuksiltaan kamerat ovat identtisiä. (Cameras 2012)

4.5.2 Kameraefektit

Ihmissilmä muodostaa jatkuvasti virheitä näkökenttään: se sumentaa nopeasti liikkuvia kohteita ja pystyy tarkentamaan vain tiettyyn pisteeseen näkökentässä. Silmällä on tapana myös taittaa valoa silmän sarveiskalvolla ennen kuin valo osuu verkkokalvolle. Ilman kameraefektejä, kamera muodostaisi aina täydellisen terävän kuvan 3D-maailmassa. Esimerkkinä kuva hyttysestä, jonka takana näkyy vuori. Ihmissilmä voi tarkentaa vain hyttyseen, jolloin vuori näkyy taustalla sumeana. Kamera puolestaan näyttää sekä vuoren että hyttyksen terävänä. (Cameras 2012)

Silmän ja tosielämän kameran näkökenttää simuloivaa efektiä kutsutaan focukseksi tai depth of fieldiksi eli DOF:iksi. Suomennettuna voidaan puhua syvyysterävyydestä. Syvyysterävyys efekti löytyy opinnäytetyössä käytetyistä ohjelmista. Syvyysterävyyden käytöllä voidaan samaan aikaan korostaa visualisoitavia objekteja ja sumentaa taustalta ylimääräisiä asioita. Jossain tapauksissa tätä käytetään tehokeinona optimoinnissa. 3D-kohtauksen tausta voi koostua 2D-objekteista, jotka sumennetaan syvyysterävyyden avulla.

Motion blur eli liikesumennus tekee nimensä mukaisesti liikkuvista objekteista epätarkkoja. Liikkeensumennuksella jäljitellään vähäisissä määrin ihmissilmän hitautta. Lähtökohtaisesti sillä jäljitellään kameran suljinajasta johtuvaa epätarkkuutta. Tämä on tuttu ilmiö kaikille niille, jotka ovat yrittäneet valokuvata pimeässä nopeasti liikkuvaa asiaa huonolla kameralla. Liikaa liikettä tallentuu yhteen kuvaan ja kuvasta muodostuu sumea. (Chopine 2011, 195-198.)

Lens flare eli linssiheijastus on efekti, joka johtuu fyysisen kameran linssistä. Linssissä valo taittuu väärin muodostaen hajavaloa linssin sisään, jolloin valo kulkee linssiltä toiselle ja aiheuttaa efektin. Tosielämässä tämä efekti pyritään minimoimaan valokuvista. 3D-maailmassa tämä efekti halutaan lisätä lopputulokseen tuomaan realistisuutta. Linssiheijastusta voidaan käyttää tehokeinona korostamaan 3D-objektin kiiltoa. Linssiheijastus voidaan myös tuottaa molemmissa opinnäytetyössä käytetyissä ohjelmissa. (Linssiheijastus. 2000.)

4.6 Animointi

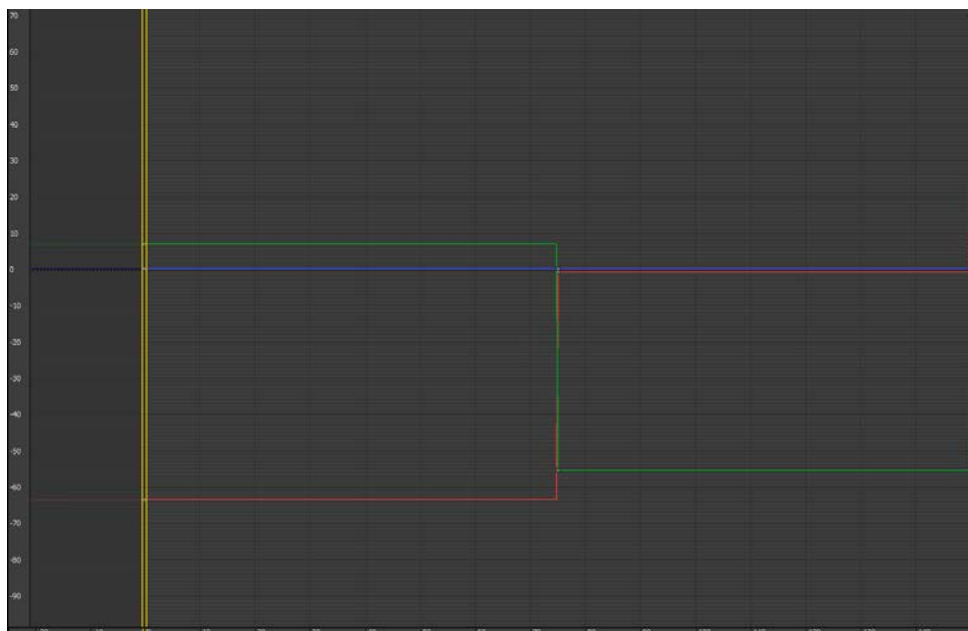
4.6.1 Keyframe-animaatio

Keyframe-animaatiossa määritellään kahdesta kolmeen eri asentoa: alku-, keski- ja loppuasennot. Kukin asento kiinnittyy määriteltyyn ajanhetkeen aikajanalla. Näiden ajanhetkien väliin jäävät asennot muodostetaan automaattisesti. Tällaista automaattista muodostusta kutsutaan interpolaatioksi. Esimerkkinä keyframe-animaatiosta voi toimia animaatio oven sulkeutumisesta. Ensimmäisellä avainhetkellä ovi on auki ja toisella avainhetkellä ovi on kiinni. Näiden kahden asennon väliin jäävät asennot muodostetaan automaattisesti. (Chopine 2011, 111.)

4.6.2 Kaavioilla animointi

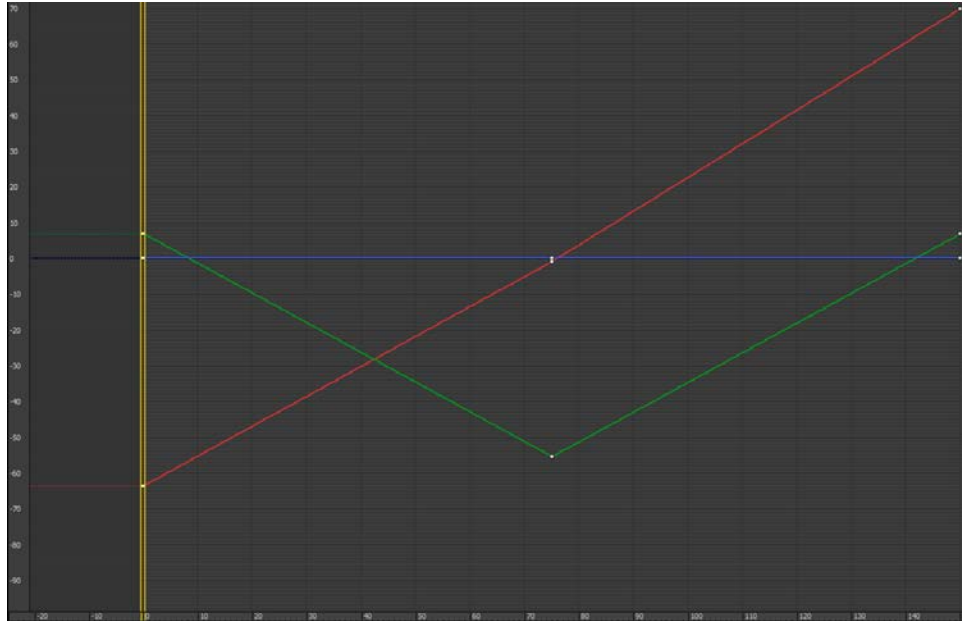
Tärkein animaation tekemiseen käytetty graafinen apuväline on aikajana. Aikajana on kuvaus animaation kulusta. Se näyttää jokaisen hetken animaation alusta loppuun. Aikajana on yksinkertaisin kaavio animaation luomiseen, mutta samalla sen kuvaama aika on elementti, joka on mukana kaikissa animointikaavioissa. Aikajanalla käytetään vain yhtä symbolia eli keyframea kuvaamaan tiettyä asentoa. 3D-objektin (3D-malli, kamera, valo) asento ja paikka määritellään kolmella akselilla. Yksi keyframe pitää sisällään tiedon sijainnista, asennosta ja skaalauksesta kaikilla kolmella akselilla. Näitä keyframen arvoja

voidaan tarkastella ja animoida jokaisella eri akselilla aikaan nähden. Tätä kutsutaan curveilla animoinniksi. Animaatio curve näyttää kahden keyframen väliin muodostuvan, interpoloidun animaation arvot curveina. Interpoloitua curvea voidaan animoida tai muuttaa. Animaatio curvella on kolme eri päätyyppiä: porrastettu, lineaarinen ja kaareva curve. Curven muodostuminen riippuu siitä, millä tapaa sen interpolaatio lasketaan. Porrastettu interpolaatio luo nykyään animaation. Porrastetussa animaatiossa liikeestä puuttuu kiihtyvyyttä, ja liike tapahtuu aina seuraavan keyframen kohdalla. Porrastetun animaation curve näyttää nimensä mukaisesti porrasmaiselta (kts. kuvio 10); portaan nousu tapahtuu aina seuraavan keyframen kohdalla. (Chopine 2011, 112-114.)

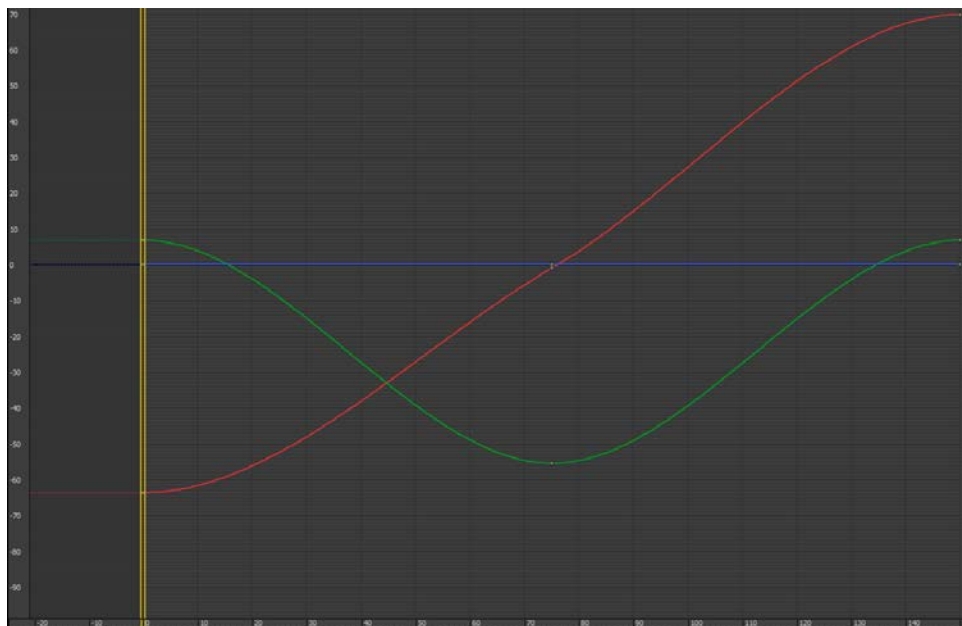


KUVIO 10. Porrastettu curve.

Lineaarisessa interpolaatiossa liike on sulavampaa, sillä se tapahtuu lineaarisesti kahden keyframen välillä (kts. kuvio 11). Lineaarisessa animatiossa liikkeitä ei kuitenkaan ole sulautettu, vaan animaatio nytkähtää keyframejen välissä. Sulava animaatio saadaan luotua kaarevalla curvella (kts. kuvio 12), jossa liike kiihtyy ja hidastuu luonnollisesti. Kaarevalla curvella myös eri keyframejen väliset muutokset saadaan sulaviksi, joten nykimistä ei oikein käytettynä tapahdu. (Chopine 2011, 112-114.)



KUVIO 11. Lineaarinen curve.



KUVIO 12. Kaareva curve.

Curvet ovat erinomainen työkalu animaation hienosäätöön, ne myös helpottavat erilaisten, toistuvien animaatioiden luomista. Curvejen kanssa animoidessa tulee aina pitää silmällä mitä animaatiossa tapahtuu. On hyvä pitää curvejen muokkaustyökalua toisella näytöllä ja tarkkailla toisesta näytöstä animaatiossa tapahtuvaa muutosta. (Chopine 2011, 112-114.)

5 Renderöinti

Renderöinti on 2D-kuvan muodostamista 3D-kohtauksesta. Helpommin sanottuna renderöinti on kuvainnollistamista, jossa tietokone laskee 3D-kohtauksesta valokuvamaisen kuvan. Renderöinnissä 3D-malliin lasketaan valot, materiaalit ja efektit niin, että lopputuloksena on valokuvamainen tulos 3D-mallista. Renderöinti suoritetaan aina viimeisenä 3D-kohtauksen tai mallin visualisoinnissa. Se on työvaihe, joka täytyy kuitenkin pitää mielessä työn alusta asti. Kuten aikaisemmin on viitattu, 3D-mallia luodessa tulee aina miettiä, millä tavalla se tullaan renderöimään. Renderöinti voidaan suorittaa monella eri tavoin. Tässä opinnäytetyössä sen toteuttamiseen käytetään kahta eri tapaa. Perinteinen renderöintitapa tuottaa tuloksekseen sarjan kuvia tai yhden kuvan. Kuva muodostuu renderöintimoottorille ennaltamäärättyjen asetusten mukaisesti. Kuvaa voidaan katsoa renderöinnin jälkeen vain siitä kuvakulmasta, jossa kamera on ollut renderöintiä määriteltäessä. Ainoa tapa vaikuttaa kuvaan renderöinnin jälkeen on tehdä siihen muutoksia kuvankäsittelyohjelmassa. Perinteinen renderöintitapa vaatii paljon konetehoa, sillä kaikki 3D-objektit pyritään näyttämään hyvällä laadulla. Realistinen valaistus vaatii myös ison osan konetehosta. Perinteisen kuvan renderöinti voi viedä sekunneista tunteihin, riippuen kuvakoosta sekä 3D-objektien monimutkaisuudesta ja pintamateriaaleista. (Chopine 2011, 219.)

Toinen renderöintitapa on jo aikaisemmin sivuttu reaaliaikainen renderöinti. Reaaliaikainen renderöinti vaatii kevyemmän 3D-mallin, sillä siitä tulee olla mahdollista muodostaa useita kuvia sekunnissa. Reaaliaikaisessa renderöinnissä interaktio kuvan kanssa jatkuu renderöinnin edetessä. Näiden kahden renderöintitavan vertailu keskenään on vaikeaa, sillä ne on suunniteltu eri käyttötarkoituksiin. Reaaliaikaista renderöintitapaa käytetään peleissä, kun taas perinteistä renderöintiä käytetään still-kuvissa ja animaatioissa. Perinteistä renderöintitapaa voidaan käyttää referenssinä reaaliaikaisen renderöinnin tulokselle. Molemmissa renderöintitavoissa täytyy muistaa vedostaminen. Usein ensimmäisellä renderöintiyrityksellä ei päästä haluttuun lopputulokseen. Ennen

viimeistä renderöintiä tai 3D-mallin lähettämistä reaaliaikarenderöintiin, kannattaa 3D-mallista ottaa useita koerenderöintejä eli niin sanottuja koevedoksia. Näin vältetään virheitä lopullisessa kuvassa. (Chopine 2011, 219.)

Reaaliaikarenderöintiin tehtävistä 3D-malleista kannattaa ottaa koerenderöintejä perinteisin menetelmin jo valmistusvaiheessa ennen kuin malli siirretään reaaliaikarenderöintiin. Mallia on helpompi testirenderöidä 3D-mallinnusohjelmassa kuin siirtää malli jokaisella kerralla erikseen reaaliaikamoottorille. Näin virheet saadaan karsittua pois jo ennen kuin 3D-malli siirretään reaaliaikamoottorille. (Kerlow 2009, 196-200.)

6 TEOLLISUUDEN 3D-MALLIN OPTIMOINTI KÄYTÄNNÖSSÄ

6.1 Yleistä

Käytäntöosuudessa käydään läpi optimointiprojektin eteneneminen. Tarkoituksena on selkeyttää projektin kulkua suunnittelusta aina valmiin mallin luovutuskuntoon saattamiseen asti. Käytännön esimerkeissä käydään läpi sitä, miten raskas geometria saadaan optimoitua reaaliaikamoottorille teksturoinnin tehokeinoja hyödyntämällä. Samalla tunnistetaan yleisimmät virheet 3D-mallin geometriassa ja käydään läpi niiden korjaaminen vaiheittain.

6.2 Projektin määrittely ja suunnittelu

Projektin aloituksessa on hyvä tehdä ensimmäiseksi tarkka suunnitelma siitä, mitä tuloksia haetaan ja kuinka ne saavutetaan. Tämän suunnitelman tekemistä kutsutaan projektin määrittely- ja suunnitteluvaiheeksi. 3D-mallin optimointiprojektiin ryhdyttäessä, on syytä määritellä ainakin seuraavat asiat:

Mitä visualisoinnissa halutaan näyttää?

Kuinka paljon resursseja optimointiin on käytettävissä?

Mikä on suurin sallittu polygonimäärä lopullisessa 3D-mallissa?

”Mitä visualisoinnissa halutaan näyttää?” on yksi tärkeimmistä kysymyksistä koko projektin määrittelyssä. Se määrittelee mihin projektissa pyritään ja selkeyttää samalla koko projektin kulkua. Käytännössä kysymys määrittelee sen, mitä kaikkea mallista voidaan karsia pois tai esittää huonommalla laadulla ilman että visualisoinnin alkuperäinen idea kärsii.

Visualisoinnin tavoitteen määrittelyn jälkeen on syytä miettiä siihen käytettäviä resursseja. Käytännössä on mietittävä, monenko tunnin työllä päästään haluttuun lopputulokseen. Mikäli resurssit eivät riitä visualisoinnin esittämiseen parhaalla mahdollisella laadulla, voidaan visualisoinnin ja optimoinnin laadusta karsia tiettyyn tasoon asti.

Projektille on hyvä määritellä minimilaatu. Laadun alentamisella tarkoitetaan tässä tapauksessa optimoinnin tason laskemista, mikä puolestaan tarkoittaa raskaampaa 3D-mallia. Tämän vuoksi suunnittelu- ja määrittelyvaiheessa täytyy tietää polygonien maksimimäärä tulevalle renderöintimoottorille.

Määrittelyvaihe kertoo mitä projektissa ollaan tekemässä, ja mitkä ovat projektin rajoitteet. Suunnitteluvaiheessa laaditaan edellämainittujen määritysten perusteella suunnitelma siitä, miten haluttuun lopputulokseen päästään. Tällaista dokumenttia nimitetään tuotantokäsikirjoitukseksi. Tuotantokäsikirjoitus toimii tuotantovaiheen runkona ja näin ollen apuvälineenä projektissa työskenteleville työntekijöille. 3D-visualistin on syytä laatia oma toimintasuunnitelma jokaiselle optimointiprojektille. Tämä suunnitelma toimii 3D-mallintajalle itselleen muistilistana projektissa huomioitavista asioista.

6.3 Tuotantovaihe

6.3.1 3D-mallin kääntäminen

Teollisuuden 3D-mallista puhuttaessa tarkoitetaan CAD-mallia. CAD, eli suunnittelumallit vaativat erityisesti tarkkuutta. CAD-mallien geometria rakentuu NURBS-käyristä. NURBS-käyrät ovat matemaattisia pintojen määrittämiseen käytettyjä käyriä, joilla pystytään muodostamaan tarkkoja pintojen muotoja.

NURBS-käyrillä muodostettu pinta olisi mahdotonta renderöidä reaaliaikaisesti. Tästä syystä 3D-malli tulee kääntää polygonigeometriaksi. Topologian muutos voidaan tehdä eri 3D-ohjelmissa. Optimoinnin kannalta parhaaksi ratkaisuksi on muodostunut Deep exploration -ohjelman käyttö, koska siinä voidaan määrittää pinnan tesselaatio. Tesselaatiolla tarkoitetaan sitä, kuinka moneen polygoniin NURBS-pinta jakautuu. Purkimyksenä on pitää tesselaatio arvot sellaisina, että geometria säilyy virheettömänä, mutta polygonimäärä ei kasva liian suureksi. Järkevämpää on kääntää mieluummin liikaa polygoneja sisältävä malli kuin liian vähän polygoneja sisältävä malli. Mallin kääntämisen jälkeen polygoniverkko tallennetaan, jolloin se on valmis optimoitavaksi.

6.3.2 Mallin tuominen käsiteltäväksi

Käännetty 3D-malli tuodaan 3D max -ohjelmaan, missä varsinainen optimointi suoritetaan. Malli tuodaan 3D maxiin import-toiminnolla. Import-toiminnolla on mahdollista avata 3D-malleja, jotka ovat 3D maxille epänatiivissa muodossa. Import-vaiheessa voidaan määrittellä se, mitä alkuperäisestä mallista halutaan tuoda 3D maxiin. Alkuperäiset animaatiot, kamerat sekä valot kannattaa jättää pois 3D-mallin tuontivaiheessa. Pääsääntöisesti mukaan kannattaa ottaa vain smoothing groupit sekä materiaalit. Alkuperäiset materiaalit korvataan myöhemmin paremmin reaaliaikavisualisointiin soveltuvilla materiaaleilla. Tärkeintä on saada tieto siitä, mikä materiaali kuuluu mihinkin 3D-mallin osaan. Tämä tieto nopeuttaa tulevaa teksturointia huomattavasti. Tuontivaiheessa tulee myös määrittellä mittayksikkö, jota mallissa tullaan käyttämään. Mittayksikön

valinnassa pitää miettiä mallin reaali maailman kokoa. Jos puhutaan erittäin pienestä objektista, tulee mittayksikkönä käyttää millimetrejä. Isojen objektien kanssa mittayksikkönä toimii parhaiten senttimetri tai metri.

6.3.3 Mallin jäsentely ja karsiminen

Teollisuuden 3D-malli täytyy jäsenellä uudelleen heti kääntämisen jälkeen. Mallin eri osat ovat suoraan suunnitteluohjelmasta tuotuna yleensä tuotteen kasaamiseen liittyvässä järjestyksessä. Tällainen tosielämän kasaukseen ja purkuun perustuva jäsentely ei ole paras ratkaisu 3D-mallin rakenteen luomiseksi. Mallille tulee miettiä järkevämpi jäsentelytapa, joka voi liittyä muunmuassa siihen, miten malli tullaan animoimaan tai miten sitä tullaan käsittelemään. Esimerkiksi kannettava tietokone on CAD-mallissa jäsenelty niin, että koko tietokone on yksi pääryhmä, jonka alla on alarymiä aina pienimpiin komponentteihin. Tällainen objekti on järkevämpää jäsenellä kahteen pääryhmään: näyttö ja muu runko. Nämä kaksi ne ovat todennäköisimmin osat, joita tullaan kyseisessä mallissa animoimaan.

3D-mallin jäsentely on käytännössä olemassa olevien objektiryhmien purkamista ja uudelleen kasaamista järkevämmiksi kokonaisuuksiksi. Tätä uudelleen jäsentelyä tehtäessä jokainen iso kokonaisuus tulee käytyä jollain tasolla läpi. Tämä on erinomainen mahdollisuus karsia mallista pois kaikki se, mitä ei tulla näyttämään lopullisessa visualisoinnissa. Pääsääntöisesti pois karsitaan kaikki mitä ei näy. Lisäksi voidaan poistaa objekteja, joiden merkitys lopulliseen visualisointiin on vähäistä. Ennen suurempaa karsimista tulee miettiä, voidaanko joitain pieniä objekteja käyttää isompien osien normal- tai displacement-karttojen osina. Esimerkiksi auton sylinterinkannessa oleva moottorin valmistajan logo säilytetään toistaiseksi geometriana, jotta se voidaan myöhemmin kääntää osaksi normaalikarttaa.

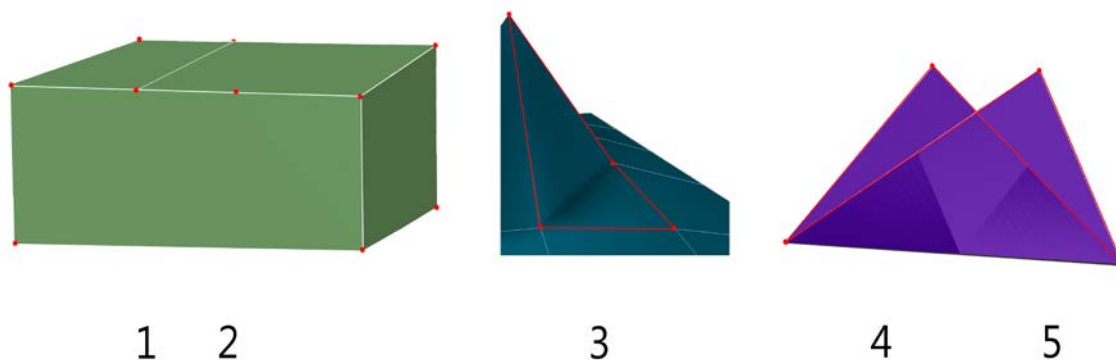
Uudelleenjäsentely kannattaa suorittaa huolella, sillä hyvin jäsenelty 3D-objekti on nopea käsiteltävä projektin edetessä.

6.3.4 Geometrian ongelmien tunnistaminen

Jäsentelyn jälkeen tulee 3D-mallista karsia pois kaikki huono geometria. Jo jäsentelyvaiheessa kannattaa tarkkailla geometriaa; mikäli jokin osio näyttää heti liian raskaalta, kannattaa se mallintaa uudelleen kevyemmäksi geometriaksi. Huonon geometrian määrittely vaihtelee, sillä eri ohjelmat käsittelevät geometriaa hieman eri tavoilla. Reaaliaikaisessa renderöinnissä huonoksi geometriaksi luokitellaan kaikki se geometria, joka aiheuttaa ongelmia lopullisen geometrian ilmentymisessä. Tällainen ongelmallinen ilmentymä voi aiheutua virheellisestä geometriasta, jolloin se näkyy virheellisenä lopputuloksena pinnassa. Ongelmalliseksi ilmentymäksi voidaan myös nimittää liian raskaan geometrian aiheuttamaa tahmaamista lopullisessa reaaliaikasovelluksessa.

Virheitä 3D-mallin geometriassa ovat (kts kuvio 13):

1. Irtonaiset edget
2. Irtonaiset verteksit
3. Vääntyneet quadit
4. Itsensä lävistävä geometria
5. Päällekkäinen geometria



KUVIO 13 Geometrian virheitä.

Lisäksi virheiksi geometriassa lasketaan enemmän kuin neljä verteksiä sisältävät quadit sekä liian raskas geometria. Liian raskaaksi geometriaksi voidaan luokitella silmämääräisesti geometria, joka sisältää liikaa polygoneja verrattuna esitettyihin yksityiskohtiin.

Geometrian virheitä syntyy, kun mallia käännetään eri käyttötarkoituksia varten topologiasta toiseen. Samoja virheitä geometriaan voi syntyä aina kun geometriaa muutetaan. Virheitä voi etsiä silmämääräisesti, mutta niiden etsimiseen on myös apuvälineitä, kuten 3D max:in STL Check, joka on erinomainen työkalu useimpien virheiden paikantamiseen. STL Check etsii 3D-mallista virheet ja näyttää ne käyttäjälle. STL Check ei kuitenkaan korjaa virheitä automaattisesti.

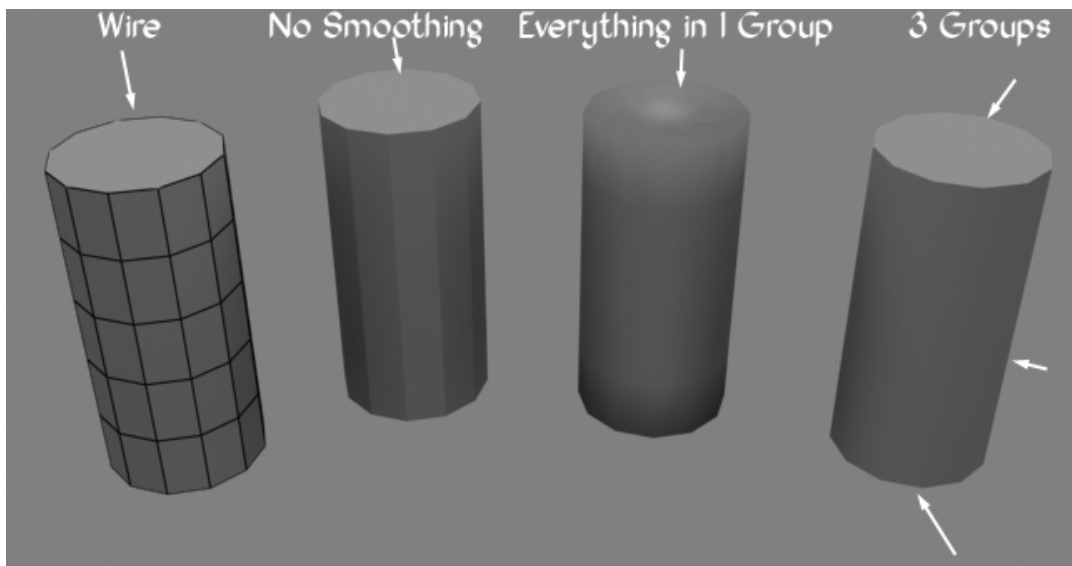
Geometrian tarkastaminen ja korjaaminen voidaan tehdä projektin alkuvaiheessa sellaiselle geometrialle, jota ei tulla enää projektin edetessä muuttamaan. Tarkastus on kuitenkin tehtävä uudelleen, mikäli geometriaa muutetaan tai luodaan lisää. (Chopine 2011, 36.)

6.3.5 Uuden geometrian luominen

Teollinen 3D-malli täytyy aina mallintaa joiltain osin uudelleen, mikäli sitä halutaan käyttää reaaliaikasovelluksessa. Uudelleenmallintaminen koskee kaikkia niitä osia, joiden tarkkuustaso on liian suuri esitettävää asiaa varten tai jotka sisältävät paljon huonoa geometriaa.

Liian raskasta vanhaa geometriaa voidaan yrittää karsia ja näin ollen myös keventää. Työkalut tähän löytyvät 3D max:in modifiereistä. Pro optimizer modifier on työkalu, jolla voidaan vähentää verteksejä automatisoidusti juoksuttamalla prosentti arvoa 100 ja nollan välillä. Arvo 100 tarkoittaa sitä, ettei yhtään verteksiä yhdistetä tai poisteta. Arvoa pienentämällä verteksiluku vähenee. Käytännössä arvoa voidaan pienentää, kunnes mallissa alkaa ilmentyä liian suuria geometrisiä muutoksia.

Geometrian uudelleenmallintaminen tehdään polygonimallinnuksena. Geometriaa luodessa pyritään esittämään asiat mahdollisimman pienellä polygonimäärällä. Parhaaseen lopputulokseen päästään, kun geometrian mallinnusvaiheessa huomioidaan karttojen sekä pehmenysten vaikutus lopulliseen geometriaan. Geometria tulee olla sellaista, että sitä pystytään pehmentämään erilaisten smoothing groupsien tai modifiereiden avulla. Smoothing groupsit (kts. kuvio14) pehmentävät 3D-mallin pintaa lisäämättä siihen geometriaa. Ne vaikuttavat ainoastaan mallin varjostukseen eli shadeaukseen. (Smoothing groups 2012.)

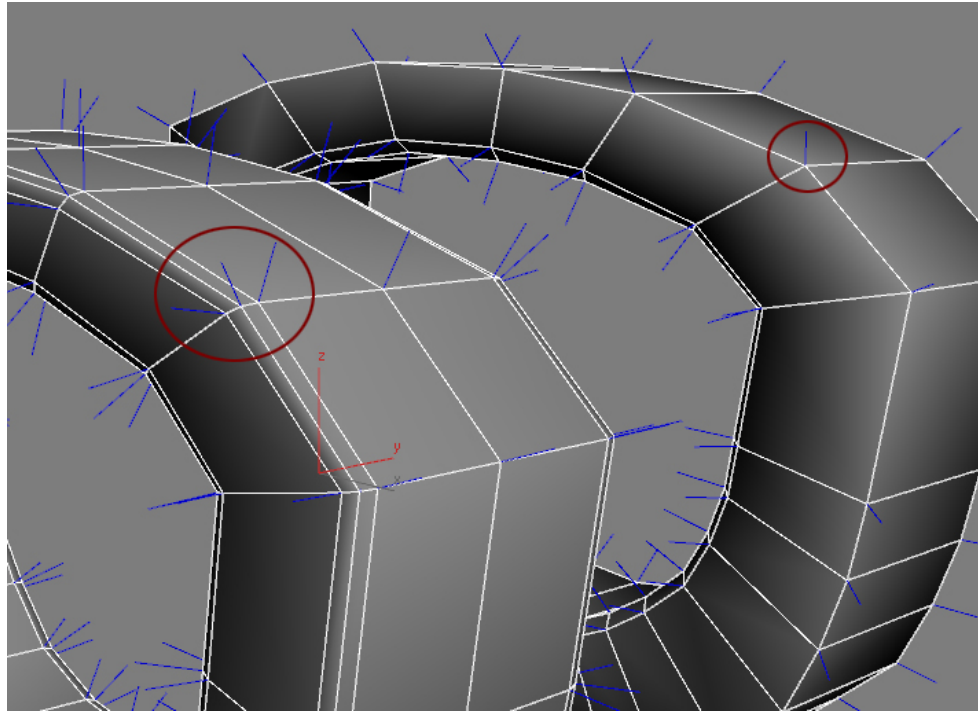


KUVIO 14 Smoothing groupin vaikutus geometrian ilmentymiseen (Smoothing Groups 2010)

Smoothing modifierit lisäävät tietyissä tapauksissa geometriaa ja näin pehmentävät mallia. Pääsääntöisesti hyvää lowpolygon-geometriaa on geometria, joka näyttää käyttötarkoituksessaan riittävältä smoothing groupsien avulla, mutta josta saadaan vielä parempaa erilaisten pehennysmodifiereiden avulla.

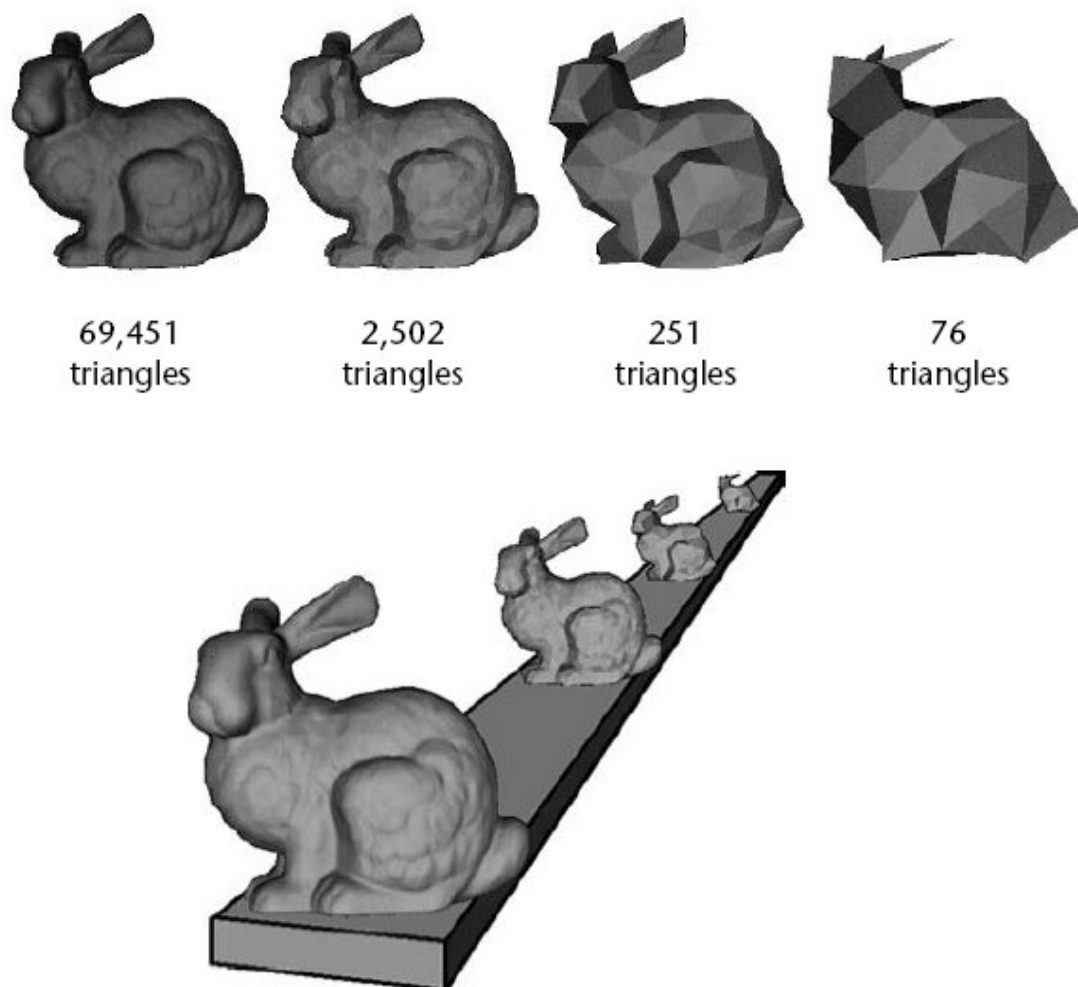
Mallinnusvaiheessa on kiinnitettävä huomiota etenkin yli 90 asteen kulmassa oleviin reunoihin 3D-mallissa, sillä nämä ovat alueita, joissa helpoiten esiintyy pehennysongelmia. Yli 90 asteiset kulmat tulee pyöristää eli camferoida.

Camferoinnissa on hyvä jakaa kulma kolmeen osaan. Kulman jakamisella saadaan enemmän normaalivektoreita, minkä myötä valo voidaan jakaa tasaisemmin. Valon jakautuessa tasaisesti saadaan pehmennys toimimaan oikein myös reunoissa. (kts. kuvio 15)



KUVIO 15. Valon jakautuminen tasaisesti normaalivektorien suuremman määrän myötä. (Normals 2010.)

Low polygon -mallinnuksessa tulee aina miettiä mallinnettavan objektin tärkeyttä tulevassa kohtauksessa. Tällaista tarkkuustason määrittystä kutsutaan termillä Level Of Detail eli LOD. Käytännössä Level Of Detaililla tarkoitetaan sitä, että lähellä olevien mallien tulee olla yksityiskohtaisempia kuin kauempana oleien, ns. toissijaisten objektien (kts kuvio 16). Kannattaa siis jo suunnitteluvaiheessa miettiä, mitä halutaan näyttää ja kuinka tarkasti. Tarkalla tarkkuustason määrittelyllä saadaan tehokkaasti vähennettyä polygonimääriä ja nopeutettua mallinnusprosessia. (Level of Detail & Visibility 2002)



KUVIO 16. Esimerkki tarkkuustason suhteesta objektin etäisyyteen. (Level of Detail & Visibility 2002.)

6.3.6 Geometristen ongelmien korjaaminen

Projektin edetessä vaiheeseen, jossa geometrian mallinnustyö on valmis, tulee geometriaan jääneet virheet korjata pois. Virhetyypit käytiin läpi aiemmin, nyt niitä tulee etsiä silmämääräisesti sekä STL check toiminnon avulla.

Irtonaisia edgejä voidaan etsiä STL checkin avulla käyttämällä open edges -valintaa. Toinen hyvä keino paikantaa avoin edge on objektin pehmennys turbosmooth modifierin avulla. Turbosmooth näyttää ongelmakohtat reikinä polygoniverkossa. Kun ongelmat on paikannettu, voidaan smoothaus poistaa

käytöstä. Avoimet edget tulee aina liittää viereisiin edgeihin. 3D-malliin voi jäädä avoimia edgejä, mutta vain sellaisiin objekteihin, joista poygoniverkko ei enää jatku.

Irtonaisia verteksejä ei tarvitse erikseen paikantaa objektista, vaan ne voidaan poistaa suoraan Edit vertices -valinnan alta löytyvällä Remove isolated vertices -valinnalla.

STL check ei osaa paikantaa vääntyneitä quadeja. Tämän ongelman voi paikantaa antamalla objektille Turn to poly -valinnan modifier-valikosta. Turn to poly -valinnan parametreista valitaan Require Planar Polygons, ja sen treshold-arvoa juoksutetaan ylöspäin. Treshold arvon ollessa välillä 0-50 pitäisi vääntyneiden quadejen jakautua kahdeksi kolmioksi. Arvo haetaan sopivaksi niin, ettei se vaikuta muuhun kuin ongelmakohtiin. Turn to poly -modifier voidaan välillä käyttää pois päältä, jotta nähdään mihin quadeihin se on vaikuttanut.

STL check tunnistaa jossain tapauksissa itsensä lävistävän geometrian päällekkäisenä geometriana. Itsensä lävistävä geometria on kuitenkin virhe, jonka havaitsee useimmiten selkeiden paljaalla silmällä. Yleensä verteksien uudelleenjärjestely korjaa ongelman, mutta quad voidaan myös poistaa kokonaan ja korvata uudella. Korvaamisessa on muistettava liittää quad huolellisesti osaksi muuta polygoniverkkoa.

Mikäli objektissa on päällekkäistä geometriaa, STL check osoittaa sen tehokkaasti. Päällekkäinen geometria voidaan poistaa välittömästi.

Normaalivirheet näkyvät pintojen shadeauksessa ja ne voidaan korjata edit normals -modifierillä.

6.3.7 UVW-karttojen määrittely

UVW-kartta täytyy määrittellä, jotta 2D-kuva saadaan asettumaan oikein 3D-mallin pintaan. 3D Maxissa UVW-karttoja voidaan määrittellä kahdella eri modifierillä: UVW-unwrap ja UVW-map. Molemmat modifierit vaikuttavat UVW-kartan koordinaatistoon. UVW-map -modifierillä voidaan projisoida tekstuuri objektin pintaan. Käytännössä parametreista valitaan mappingtyyli, joka vastaa 3D-objektin muotoa. Mapping-tyylejä on useita erilaisia sopimaan eri muodoille: sylintereille, palloille, laatikoille ja tasaisille levyille. UVW-map -toiminto on tehokas tapa UVW-karttojen määrittelyyn yksinkertaisille objekteille.

UVW-unwrap on määrittelymenetelemä vaikeammille 3D-objekteille, koska se antaa mahdollisuuden nähdä ja editoida UVW-karttoja. Unwrap modifier mahdollistaa 3D-objektin pinnan nimensä mukaan. Unwrappauksessa valitaan laajoja polygonipintoja UVW-karttaan, jonka jälkeen ne levitetään tasaisiksi levyiksi UVW-alalle. Kun koko objekti on kuorittu auki levyiksi, eikä mikään levy ole toisen päällä, voidaan UVW kartta tallentaa kuvakäsittelyohjelmaa varten.

Käytännössä objektille annetaan Unwrap UVW -modifieri modifier valikosta. Tämän jälkeen voidaan valita mapattavat polygonipinnat. Pintojen valitsemisen jälkeen voidaan avata UV-editori, mistä päästään määrittelemään, kuinka objektin pinta kuoritaan ja levitetään auki. Määrittely tehdään mapping-valikon kautta, joka antaa valittavaksi kolme eri mappaus-vaihtoehtoa: flatten-, normal-, unfoldmapping.

Flatten mapping jakaa polygonipinnan sektioihin pintojen kulmalukujen mukaan. Esimerkiksi laatikko saadaan jaettua kuuteen eri UV-sektioon antamalla alle 90 asteen arvoja. Lisäksi voidaan määrittää välistys, joka on UV-palojen väliin jäävä alue. Välistys on hyvä pitää mahdollisimman pienenä, jotta UVW-kartan ala saadaan parhaiten käyttöön. Normal mapping tasoittaa pintoja normaalivektoriin perustuen, jolloin pintoja pystytään avaamaan eri kulmista, kuten ylhäältä, edestä, takaa jne. Normal mapping on hyvä työkalu, kun sitä käytetään hillityn kokoisille pinnoille. Unfold mapping kuorii objektin pyrkien pitämään kaikki UVW-

kartan palat toisissaan kiinni. Esimerkiksi kuutio jakautuu kuuteen osaan, jotka on tikattu toisiinsa kiinni.

UVW-editorilla voidaan myös suoristaa UVW-karttojen vääristymiä sekä liittää UVW-karttojen osia toisiinsa. Kaikilla työkaluilla pyritään samaan lopputulokseen, virheettömään UVW-karttaan joka täyttää tehokkaasti UVW-alan. Lopputulos on karttojen määrittelyssä tärkeintä.

Määrittelyiden valmistuttua kannattaa UVW-modifier liittää osaksi objektin muuta dataa. Tämä tapahtuu klikkaamalla hiiren oikeaa painiketta Unwrap-UWV-modifierin päällä, ja antamalla valikosta ”collapse to” valinta. UWV-kartta häviää modifier stackista ja menee talteen objektin dataan.

UVW-karttoja tehdessä on syytä pitää mielessä UVW-tilan käyttö. Käytännössä ihannetilanne olisi sellainen, missä UVW-kartta täyttäisi kokonaan sille annetun tilan. Hukkatilan minimoinnilla saadaan etenkin pienikokoisten tekstuurien kohdalla suurta hyötyä. Tekstuurin tarkkuus saadaan paremmaksi nostamalla tekstuurikartan kokoa.

6.3.8 Teksturointi

Teksturointi voidaan aloittaa, kun objektin UVW-kartat ovat kunnossa. Teksturoinnissa 3D-objektille määritellään väri, kiiltoaste sekä pinnan epätasaisuus. Teksturointia voidaan käyttää myös optimoinnin tehokkeinona hyödyntämällä normaalikarttoja.

Teksturoinnissa on hyvä lähteä liikkeelle objektin väristä, sillä tietyt muodot korostuvat eri tavoin eri väreillä. Tummat värit korostavat pinnan epätasaisuutta, kun taas vaaleat vievät sitä pois. Värin määrittelyllä ensimmäisenä vältytäänkin bump-mappien turhalta säätämiseltä. Väri voidaan määrittellä tietyillä RGB-arvoilla tai bitmap-kuvalla. Bitmap-kuvan tulee olla UVW-kartan sisäpuolella, jotta se näkyy 3D-objektin pinnalla. 3D Maxissa pinnan väri voidaan myös maalata suoraan 3D-objektin pintaan canvas paint -työkalulla. Tällöin kuvasta

muodostuu automaattisesti bitmap-kuva, joka vastaa objektin UVW-avaruutta. Värien määrittämisen jälkeen voidaan määrittellä kiiltoaste, joka käytännössä määrää sen, miten valo heijastuu tai hajaantuu objektin pintaan.

Samalla tavoin kuin objektille annetaan väri- tai heijastusarvoja, voidaan sille antaa bump-map lisäämään pintaan epätasaisuutta. Reaaliaikarenderöinnissä hyödynnetään normaalikarttoja bump-mappeina. Normaalikartta on bump-kartta, jolla voidaan tehdä matalan polygonimäärän omaavasta mallista korkeapolygonimallin näköinen.

Normaalikartta luodaan teollisuuden 3D-mallista niistä objekteista, joissa tarvitaan paljon yksityiskohtia. Geometrisesti tällaisia yksityiskohtia ei voida loppukäytössä esittää liian suuren polygonimäärän takia.

Normaalikartan luomisessa käytetään apuna high polygon -mallia, joka on siis teollisuuden 3D malli. Normaalikartan luomiseen täytyy mallintaa high polygon -mallia pääpiirteittäin vastaava low polygon -malli. Tämä malli sisältää vain high polygon -mallin muodon. Low polygon- mallista saadaan high polygon -mallin näköinen, kun sille muodostetaan normaalikartta high polygon -mallista.

Käytännössä normaalikartan luonti tapahtuu 3D Maxissa. 3D Maxissa high polygon -mallin muodoista projisoidaan kartat low polygon- mallille. Tämän jälkeen normaalikartta viedään kuvankäsittelyohjelmaan, missä objektin pintaan voidaan lisätä samoja yksityiskohtia, joita normaalisti oltaisiin lisätty high polygon -malliin bump-mapilla.

Pienten yksityiskohtien lisäämisen jälkeen normaalikartta tuodaan low polygon -mallin normal bump-mapiksi. Low polygon -malli näyttää low polygon -mallilta aina renderöintiin asti.

6.3.9 Animointi

Animointi voidaan tehdä 3D-mallille siinä vaiheessa, kun sen geometriaan ei enää tehdä suuria muutoksia. Toisin sanoen 3D-mallia voidaan animoida mallinnus- ja optimointityön päätyttyä. Kameraa puolestaan voidaan animoida projektin missä tahansa vaiheessa, sillä kamera kulkee mukana projektin alusta loppuun. Geometriaa voidaan puolestaan korvata uudella, jonka vuoksi sen aiempi animoiminen voi olla turhaa.

Animaation kulku määritellään projektin suunnitteluvaiheessa.

Reaaliaikasovelluksissa kameraa tarvitsee harvoin animoida, sillä reaaliaikasovelluksen käyttäjä määrittelee kameran animaation kulun lähes kokonaan. 3D-objektille pystytään puolestaan määrittelemään erilaisia animaatioita, joita toteutetaan tietyin ehdoin reaaliaikasovelluksessa.

Esimerkkinä voidaan käyttää reaaliaikasovelluksessa renderöitävää autoa, jonka konepellin aukeaminen on ennalta animoitu. Käyttäjä laukaisee animaation klikkaamalla konepeltiä.

Animointi suoritetaan 3D Maxissa, vaikka sen laukaisemisen määrytykset tuleekin tehdä myöhemmin reaaliaikasovelluksessa. Animointiin käytetään keyframe-tekniikkaa, jota hienosäädetään curveilla. Keyframe-animaatio muodostetaan aikajanelle niin, että kaikki tietyille mallille tulevat liikesarjat animoidaan peräkkäin. Liikesarjojen keyframe-välit tulee kirjata valmiiksi erilliseen tekstitiedostoon. Näin ohjelmoija voi tulevaisuudessa ohjelmoida 3D-mallin toistamaan jonkin tietyn animaation napin painaluksella.

Seuraava esimerkki käsittelee animaationsarjojen merkitsemisen Away 3D - reaaliaikarenderöintiä varten. Tätä merkintätapaa voidaan helposti soveltaa muihinkin reaaliaikasovelluksiin. Esimerkkinä voidaan käyttää animointia auton toiminnallisuuksista. Auton toiminnallisuuksia ovat tyhjäkäynti, joka animoidaan tärinänä, kiihdytys, joka visualisoidaan auton nojautumisena taaksepäin sekä jarrutus visualisoituna auton nojautumisena eteenpäin. Aikajanelle nämä toiminnallisuudet animoidaan peräkkäin: tyhjäkäynti, kiihdytys, jarrutus.

Tyhjäkäynti on toistuva animaatio, joka sijoittuu keyframeille 0-20. 0 on ensimmäinen ja 20 viimeinen keyframe. Kiihdytys animoidaan heti tämän jälkeisille keyframeille 21-50. Viimeisenä animoidaan jarrutus keyframeille 51-70.

Keyframet kirjataan tekstitiedostoon seuraavassa muodossa: liikkeen nimi, aloituskeyframe, lopetuskeyframe. Lisäksi voidaan kirjata omia kommentteja ja huomioita lisäämällä lauseen eteen risuaita (#).

Auton liikkeet.

tyhjakaynti 0 20

kiihdytys 21 50

jarrutus 51 70

Tekstitiedostolle annetaan kuvaava nimi ja se tallennetaan .txt -tiedostona samaan kansioon, johon lopuksi tallennetaan myös kaikki tekstuurit ja valmis 3D-malli.

6.3.10 Referenssikuvan renderöinti

3D-mallista tulee renderöidä testikuvia teksturointityön edetessä.

Testirenderöinnit paljastavat virheet ja auttavat luomaan virheettömän sekä halutun lopputuloksen. Referenssikuva renderöidään siinä vaiheessa, kun työ on valmis. Referenssikuva toimii apuvälineenä sille, kuka tulee viemään 3D-mallin reaaliaikasovellukseen. Referenssikuva toimii siis esimerkkinä siitä, millainen renderöintituloksen tulisi olla reaaliaikasovelluksessa.

Referenssikuva renderöidään 3D Maxista käyttäen Vray 2.0 -renderöintimoottoria. Vray käyttää referenssikuvan laskemiseen sata- tai jopa tuhatkertaisen ajan verrattuna reaaliaikasovellukseen. Tämän vuoksi reaaliaikasovelluksella ei tulla ikinä saamaan samanlaista renderöintitulosta. Referenssikuva toimiikin vain esimerkkinä siitä, miten visualisti oli kaiken suunnitellut. Hyvässä referenssikuvassa näkyy se, miten tekstuurien pitäisi toimia, millaista geometrian tulisi olla ja millaista tunnelmaa haetaan.

Referenssikuvan renderöimistä varten täytyy 3D Maxissa määritellä kuvan valot. Valojen asettelussa on kannattavaa hyödyntää valo- ja videokuvauksesta tuttuja tehokeinoja. Valojen määrä on hyvä pitää hillittynä, sillä usein jo muutamalla järkevästi asetellulla valolla saadaan valaistua 3D-objekti riittävän hyvin. Valoina kannattaa käyttää sellaisia valotyypppejä, jotka löytyvät lopullisesta reaaliaikamoottorista.

Valot ovat välttämättömyys referenssikuvan renderöinnin kannalta. Valojen asettelun jälkeen on hyvä miettiä, käytetäänkö kameraefektejä vai ei. Mikäli on tiedossa, että jokin tietty kameraefekti toimii ohjelmassa, johon mallia ollaan optimoimassa, voidaan se tuoda mukaan jo referenssikuvaan.

Vray:n renderöintiasetukset kannattaa pitää yksinkertaisina. Hyvät Vray:n asetukset simuloivat mahdollisimman pitkälle reaaliaikarenderöinnin laskentatapaa. Vray:n renderöintiasetukset eivät kuitenkaan ole tärkeässä osassa, vaan pääasia on, että saadaan luotua selkeä kuvaus siitä, mihin visualisoinnissa pyrittiin.

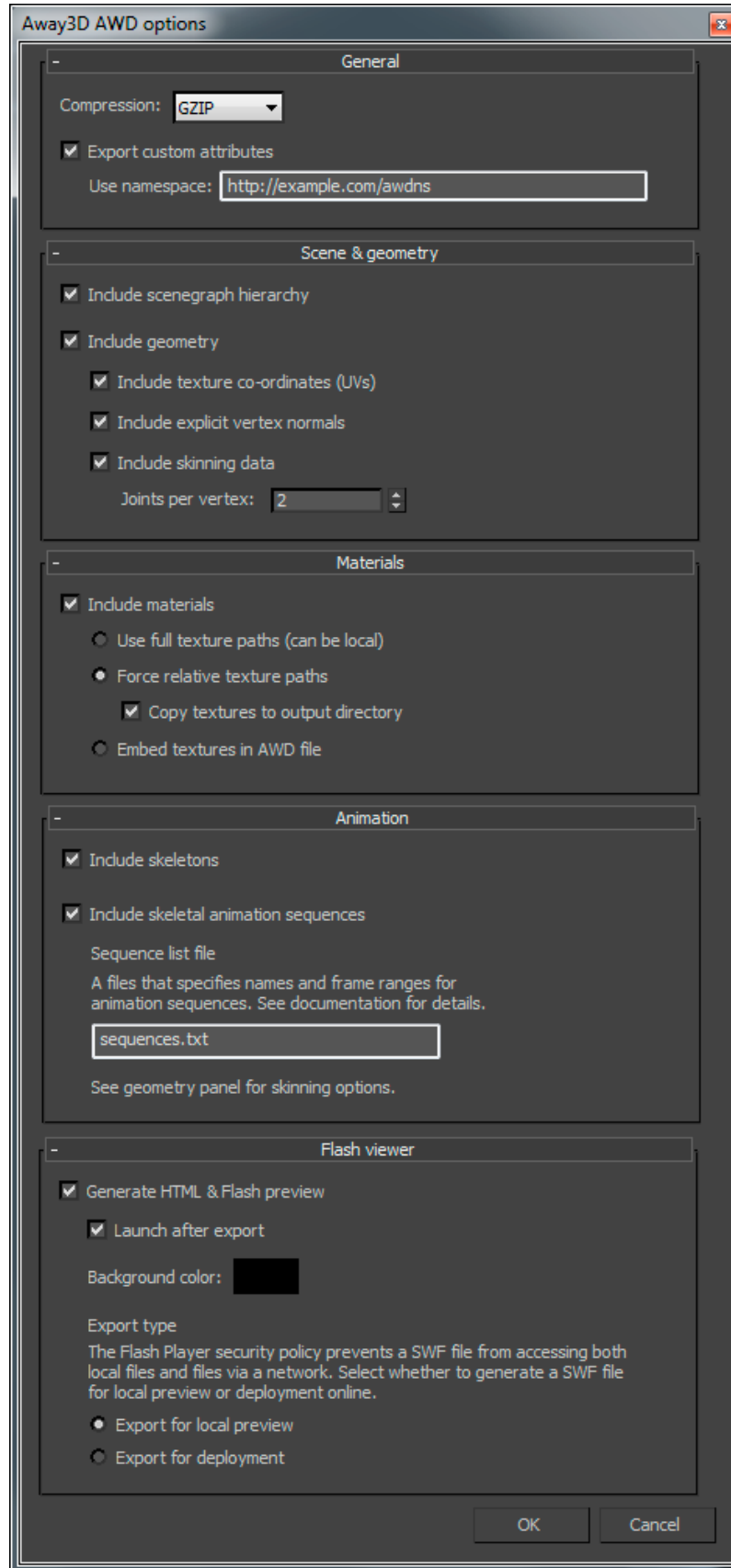
6.3.11 Optimoidun mallin jättäminen

Optimoinnin valmistuttua kaikki kerätty aineisto tulee kerätä samaan paikkaan, jotta se voidaan myöhemmin viedä reaaliaikasovellukseen. Yleisesti tämä tarkoittaa kaiken tiedon keräämistä mahdollisimman hyvin jäsennellysti yhteen paikkaan. 3D-malli voi tarvita paljon eri kuvia ja tiedostoja toimiakseen. Käytännöllisin tapa kaiken tiedon keräämiseen samaan paikkaan on tallentaa tiedosto 3D Maxista suoraan Zip paketiksi. Tämä tapahtuu 3D Maxin tallennusvalikosta, "save as archive" -kohdasta. Optimointiprojektin alussa luotu kansiorakenne tulisi olla selkeä, ja kaikkien tiedostojen tulisi olla tiettyjen projektien alla. "Save as archive" on syytä suorittaa varmuustoimenpiteenä, sillä se kaivaa mahdollisesti johonkin toiseen kansiorakenteeseen jääneet tekstuurit yhteen paikkaan.

Mikäli tiedetään mihin reaaliaikarenderöintiohjelmaan mallia ollaan viemässä,

voidaan käyttää 3D Maxiin asennettavia exporter-lisäosia. Opinnäytetyössä 3D-malli viedään suoraan Away 3D -ohjelmaan. Tästä syystä käytetään AWD-pluginia, jolla kaikki 3D-mallin sisältämät tiedot saadaan tallennettua Away 3D -ohjelman ymmärtämään muotoon. AWD-Plugin asennetaan kopioimalla sen tiedostot 3D-maxin plugin-kansioon, minkä jälkeen plugin on asennettu ja valmis käytettäväksi. 3D-malli tallennetaan AWD-muotoon valitsemalla 3D Maxin export-valikosta "export" -valinta. Tallennustyyppistä valitaan Away 3D (*.AWD) ja tiedostolle annetaan nimi. Valinnan hyväksymisen jälkeen aukeaa kuviossa 17 esitetty valikko. Valikon kautta voidaan valita, mitä kaikkea malliin sisällytetään. Mikäli mallia halutaan katsella vielä ennen kuin se annetaan eteenpäin, voidaan oletusasetukset jättää valituiksi flash viewer -kohtaan. Tällöin malli aukeaa Flash playeriin esikatseltavaksi.

Malli voidaan luovuttaa eteenpäin ohjelmoijille, kun kaikki sen tiedot on kerätty yhteen. Projektin viimeisessä vaiheessa on muistettava hyvät nimeämiskäytännöt sekä selkeät kansiorakenteet. Näin muiden on helppo jatkaa siitä, mihin optimointiprojekti päättyy.



KUVIO 17. AWD plugin, tiedoston tallennus .awd muotoon.

7 POHDINTA

Opinnäytetyöllä lähdettiin hakemaan ratkaisua ongelmaan, miten teollisuuden 3D-malli saataisiin kevennettyä niin, että sitä pystyttäisiin hyödyntämään reaaliaikasovelluksessa. Ongelmaa lähdettiin ratkaisemaan optimoinnilla. 3D-geometrian optimointiin on monia hyviä keinoja sekä toteutustapoja. Työhön valittiin asiakkaan tarpeisiin parhaiten sopivat optimointikeinot. Valinta suoritettiin määrittelemällä tyypillinen käsiteltävä 3D-malli Movya Oy:ssä. Tältä pohjalta pystyttiin valitsemaan opinnäytetyöhön asiakkaalle parhaiten sopivat optimointikeinot.

Opinnäytetyön tietoperustassa käsiteltiin 3D-mallin geometrian perusteet optimoinnin sekä reaaliaikarenderöinnin näkökannalta. Käytännön osuudessa käytiin läpi sellasia geometrian virheitä, joita useimmiten esiintyi Movya Oy:n käyttämissä 3D-malleissa. Asiakokonaisuudet olivat erittäin laajoja, eikä niitä voitu käsitellä tämän opinnäytetyön puitteissa laajemmin. Asiat pyrittiin käsittelemään selkeästi sekä niin, ettei opinnäytetyötä jouduttu sitomaan mihinkään tiettyyn 3D-työkaluun. Vaikka työssä pyrittiinkin ohjelmistoriippumattomuuteen, esiteltiin siinä kuitenkin optimointityötä nopeuttavia näkökulmia myös Movya Oy:n käytöstä löytyvillä ohjelmilla.

Opinnäytetyön tuloksena syntyi ohjeistus 3D-mallin optimoinnista. Ohjeistus toi Movya Oy:lle tietämystä siitä, mitä kaikkea 3D-mallille tulisi tehdä, jotta se voitaisiin renderöidä reaaliaikaisesti. Ohjeistusta voidaan käyttää nykyisten ja uusien työntekijöiden perehdyttämiseen, lisäksi työn tekeminen antoi kattavan tietoperustan opinnäytetyön tekijälle. Opinnäytetyön tekeminen vaati monen työssä esiintyvän asian testaamista käytännössä. Käytännön testaus antoi tämän työn tekijälle hyvän tietämyksen siitä, kuinka paljon optimointiprojektiin tulee varata aikaa. Tämän myötä myös Movya Oy voi arvioida paremmin optimointiprojektiin vaadittavia resursseja. Opinnäytetyön tuloksia pohdittaessa voidaan todeta, että opinnäytetyössä käsitellyt optimointikeinoja on mahdollista hyödyntää myös perinteisten visualisointimentelmien kanssa. Opinnäytetyön

tuloksena syntynyt ohjeistus onkin hyvä ohjeistus kaikille raskaiden 3D-mallien kanssa työskenteleville henkilöille.

8 LÄHTEET

3ds Max. System requirements 2012. Viitattu 11.10.2012.
<http://usa.autodesk.com/3ds-max/system-requirements/>

3ds Max – tuotteet 2012. Viitattu 11.10.2012.
<http://www.autodesk.fi/adsk/servlet/pc/index?siteID=448412&id=14644225>

Adding the AWD plugin. 2012. Viitattu 21.10.2012
http://www.away3d.com/example/3ds_Max_workflow

Anderson, J, A. Colvin & J. Tobler, N. 2003 Productivity and Multi-Screen Displays. Tutkimus. University of Utah.
<http://www.ergotron.com/Portals/0/literature/whitePapers/english/Multi-Mon-Report.pdf>

Away 3d Features 2012. Viitattu 22.10.2012
<http://away3d.com/features/>

Akenine-Möller, T. Haines, E. Hoffman, N 2008 Real time rendering third edition. Massachusetts: A K Peters, Ltd.

A Computer animated hand. 2004. Viitattu 4.11.2012
<http://www.animationmagazine.net/wordpress/wp-content/uploads/A-Computer-Animated-Hand-510x364.jpg>

Cameras 2012. Viitattu 7.11.2012
<http://docs.autodesk.com/3DSMAX/15/ENU/3ds-MaxHelp/index.html?url=files/GUID-23F7F6D9-09A9-47D7-BCBC-759B464F5367.htm,topicNumber=d30e350756>

Chaos Software 2012. Viitattu 11.10.2012
<http://www.chaosgroup.com/en/2/vray.html>

Chopine, A. 2011 3D Art Essentials. United states: Focal Press.

Deep Exploration™ 5.7 Standard Edition 2012. Viitattu 15.10.2012
http://www.righthemisphere.com/products/dexp/de_std.html

Displacement mapping 2010. Viitattu 6.11.2012

<http://www.tomshardware.co.uk/slideshow/Displacement-Mapping,0101-161355-0-2-3-0-jpg-.html>

Finney, K. 2006. 3D game programming all in one. Third edition. Course Technology PTR

Game Development Team Composition Study - Changes over time. 2003 Viitattu 18.10.2012

<http://web.cs.wpi.edu/~id111x/c05/slides/intro.ppt>

Goldstone, W 2009. Unity Game development essentials. Birgingham: Packt Publishing

Kerlow. I. 2009 The Art of 3D computer animation and Effects. USA:John Wiley & Sons, inc.

Level of Detail & Visibility 2002. Viitattu 14.11.2012

<http://www.cs.virginia.edu/~gfx/Courses/2004/RealTime/lecture06.LOD1.ppt>

Linssiheijastus 2000. Viitattu 7.11.2012

<http://www.laitetekniikka.com/digikuvaus/linssiheijastus.html>

Luxo JR. 2007. Viitattu 22.10.2012

<http://www.virtualworldlets.net/Resources/Hosted/Resource.php?Name=LuxoJr>

Movya 2012, Movya Oy:n verkkosivut. Viitattu 11.10.2012

<http://movya.fi/>

Normal maps 2009. Viitattu 6.11.2012

<http://cg.tutsplus.com/tutorials/autodesk-3d-studio-max/how-to-bake-a-flawless-normal-map-in-3ds-max/>

Normals 2010. Viitattu 20.11.2012

<http://dl.dropbox.com/u/1306131/Normals.jpg>

Our story 2012 Viitattu 18.10.2012

<http://www.pixar.com/about/Our-Story>

Puhakka, A. 2008. 3D-grafiikka. Helsinki: Esa Print Oy.

Smoothing Groups 2010. Viitattu 22.11.2012

http://www.vigville.com/forum_images/SmoothingGroups.jpg

Smoothing groups 2012. Viitattu 14.11.2012

<http://docs.autodesk.com/3DSMAX/13/ENU/Autodesk%20ds%20Max%202011%20Help/index.html?url=../files/WSf742dab041063133-18033351112a1ce86c6-7e67.htm,topicNumber=d0e494280>

The History of 3D Computer Animation 2010. Viitattu 18.10.2012

http://www.ehow.com/facts_5070359_history-computer-animation.html

Tiedon visualisointi 2011 Viitattu 22.10.2012

<http://datajournalismi.fi/2011/09/tiedon-visualisointi/>

Tufte, E. R. 2004 The visual display of quantitative information . Cheshire: Graphics Press.

Ware, C. 2004. Information Visualization - Perception for Design. San Fransisco: Morgan Kaufmann Publishers.