
Tuotantolinjaston myyntiesittely 3D-animaatiolla



Ammattikorkeakoulun opinnäytetyö

Tietotekniikan ko

Riihimäki, kevät 2015

Toni Salow



Riihimäki
Tietotekniikan ko
Mediajärjestelmät

Tekijä	Toni Salow	Vuosi 2015
Toimeksiantaja	Pivatic Oy	
Työn nimi	Tuotantolinjaston myyntiesittely 3D-animaatiolla	

TIIVISTELMÄ

Tämän opinnäytetyön toimeksiantajana oli ohutmetallityöstölinjastojen tuottaja Pivatic Oy. Työn tarkoituksena oli toteuttaa tuotantolinjaston myyntiesittely 3D-animaatiolla valmiiden CAD-mallien perusteella, joka sisälsi mallien siirron ohjelmasta toiseen, esivalmistelut animaatiota varten, itse animoinnin sekä lopullisen visualisoinnin.

Työn teoriaosuudessa käydään läpi 3D-tekniikan perusteet mallinnuksen, animoinnin ja visualisoinnin kannalta sekä tutkitaan eri 3D-ohjelmien välisiä eroja. Teoreettisen osuuden tavoitteena on luoda lukijalle selkeä kuva 3D-visualisointiprojektin kaikista työvaiheista sekä kartoittaa eri tarkoituksiin suunniteltujen ohjelmistojen hyödyt ja haitat.

Käytännön osuudessa käydään läpi käytännön toteutuksen työvaiheet ja niissä esiintyneet ongelmat. Tässä osuudessa verrataan myös vaihtoehtoisia toteutustapoja eri työn vaiheille.

Työssä käytettiin koulutuksen myötä saatua tietoa ja opittuja taitoja, sekä Pivaticissa henkilökunnan ammattiosaamista. Teknisen puolen toteutuksessa avusti esimerkiksi 3ds Max Help ja 3D Animation Essentials -verkkojulkaisu. Työ vaati myös jatkuvaa kontaktia alan ammattilaisten kanssa, jotta linjasto saatiin esiteltyä halutulla tavalla.

Avainsanat Animaatio, optimointi, tietokoneintegroitu tuotanto, visualisointi

Sivut 48 s. + liitteet 1 s.

Riihimäki
Information Technology
Media Systems

Author	Toni Salow	Year 2015
Commissioned by	Pivatic Oy	
Subject of Bachelor's thesis	Sales presentation of a production line via 3D-animation	

ABSTRACT

The thesis was commissioned by a production line fabricator for sheet metal, Pivatic Oy. The purpose of the thesis was to produce a sales presentation of a production line via 3D-animation based on finished CAD models, which included the transfer of models between applications, the preparations for animation, the animation itself, as well as the final visualization.

In the theoretical part we go through the basics of 3D-technology in terms of modeling, animating and visualizing, as well as exploring the differences between 3D programs. The theoretical part aims to create a clear vision for the reader of the 3D-visualization workflow and to identify the advantages and disadvantages of various software programs designed for given purposes.

In the practical part we review the steps of the implementation itself and go through of all the problems encountered. We also compare different implementation alternatives to the final version.

The thesis took advantage of the skills and knowledge gained from education, and the professional expertise of Pivatic staff. Assisting in the technical side of the implementation were for example the virtual publications 3ds Max Help and 3D Animation Essentials. The thesis required constant contact with professionals in their field to produce a sales presentation in the desired manner.

Keywords Animation, optimization, computer integrated manufacturing, visualization

Pages 48 p. + appendices 1 s.



SISÄLLYS

1	JOHDANTO.....	1
2	3D-TEKNOLOGIA.....	2
2.1	3D-mallit.....	2
2.1.1	Tilavuusmalli ja pintamalli.....	4
2.1.2	Editable Mesh.....	4
2.1.3	Editable Poly.....	5
2.1.4	Body Object.....	6
2.1.5	Transformaatiot.....	7
2.2	Animaatio.....	9
2.2.1	Keyframe-animaatio.....	9
2.2.2	Proseduraalinen animaatio.....	10
2.2.3	Kamera-animaatio.....	11
2.3	Rigging.....	12
2.3.1	Hierarkiat.....	12
2.3.2	Linkitys.....	13
2.4	Visualisointi renderöintiä varten.....	14
2.4.1	Valaisu.....	14
2.4.2	Materiaaliparametrit ja shader-algoritmi.....	15
2.5	Renderöinti.....	16
2.5.1	Render-algoritmi.....	16
2.5.2	Renderöintimoottorin ja ympäristön asetukset.....	17
2.5.3	Gamma-asetukset.....	20
2.5.4	Lopullinen tiedostotyyppi.....	20
2.6	Network Rendering.....	21
3	3D-OHJELMIEN VÄLISET EROT.....	22
3.1	Ohjelmien yhteensopivuus.....	22
3.2	Mallien optimointi.....	23
4	TUOTANTOLINJASTON MYYNTIESITTELYN TYÖNKULKU.....	25
4.1	CAD-mallien valmistelu.....	25
4.1.1	Mallien jälkioptimointi.....	28
4.2	Animaation esivalmistelut.....	30
4.2.1	Linkittäminen.....	31
4.3	Toiminnallisuuden animointi.....	33
4.3.1	Objektin läpinäkyvyyden animointi.....	35
4.4	Kamera-animointi.....	36
4.5	Hienosäätö.....	37
4.5.1	Materiaaliparametrit ja teksturointi.....	37
4.5.2	UV Mapping.....	39
4.5.3	Valaistus.....	40
4.5.4	Renderöinti.....	43
4.6	Jälkitoimenpiteet.....	45
4.6.1	Videonkäsittely.....	45

5 YHTEENVETO	47
LÄHTEET	48

Liite 1 Mesh Quality -script

TERMISTÖ

Anti-aliasing	Näytteenottotarkkuus, joka pienellä arvolla voi tuottaa renderöityyn kuvaan sahalaitaisia pintoja.
Arch & Design, A&D -material	Mental Ray –renderöintimoottorin yleisimmin käytetty materiaali, joka on kehitetty pääosin tuotevisuaalisointiin.
Assembly	Kokoonpano, joka rakentuu monesta eri osasesta. Esimerkkinä polkupyörä, joka rakentuu rungosta, ohjaustangosta ja renkaista.
Autodesk Backburner	Ohjelma, joka toimii taustamanagerina tehtävien hajauttamisessa verkoston yli eri tietokoneille, kuten renderöinti.
Bi-cubic	Taso-objekti, jossa yksittäiset polygonit omaavat kaarevia pintoja. 3ds Maxissa tämä objektityyppi on nimeltään Patch.
Bitmap	Bittikartta, digitaalinen kuva.
Bitrate	Tiedonsiirtonopeus, jolla video enkoodataan. Tämä arvo määrittää myös verkosta videota katselevan kaistan tarpeen.
Body	3D-mallirunko, josta puhutaan työkennellessä ohjelmalla SolidWorks.
Caustics	Efekti, joka syntyy kun valo taittuu heijastavasta pinnasta toiseen pintaan.
Child	Objekti, joka on linkitetty toiseen objektiin.
CMYK	Yksi väriavaruuksista, jota yleisimmin käytetään painotuotteissa.
Composite	Materiaali, joka kompositoi maksimissaan 10 eri materiaalia yhteen.
Controller	Toiminto, jolla kaikkea animaatiota käsitellään.
DPI, Dots Per Inch	Näytöllä pikseleitä tai paperilla mustepistettä yhtä tuumaa kohti.
Edge	Reuna, joka muodostuu kahden kär-

	kipisteen välille polygonipinnassa.
Environment Map	”Ympäristön tausta”, jota voidaan hyödyntää realistisen valaistuksen luomisessa IBL-tekniikan kanssa, tai luomalla objekteista heijastuvat yksityiskohdat Environment Mapiksi asetetusta bitmapista.
Export	Tiedoston tai mallin vienti ohjelmasta.
Face, Polygon	Taso-objekteja, jotka täyttävät objektin kärkipisteiden yhdistävät reunat.
Final Gather	Algoritmi, joka laskee epäsuoran valon kimpoamisen objekteista toisiin objekteihin.
Forward Kinematics	Animaatioketju, jossa yksi tai useampi hallitseva objekti vaikuttaa kaikkiin sen alaisiin.
Frame	Yksi kuva animaatiossa tai videossa esiin, joita esiintyy siis tyypillisesti 24-30 kuvaa sekunnissa.
FullHD	Teräväpiirtokuva, joka yltää näytöissä 1920x1080 pikselin tarkkuuteen.
Global Illumination, GI	Algoritmi, joka ottaa huomioon valon kimpoamisen pinnoista -> kapaleet saavat epäsuoraa valoa ja renderöinnistä tulee realistisempi. Käytetään usein yhdessä Final Gather-algoritmin kanssa.
H.264	Videonpakkausformaatti, joka on yksi yleisimmin käytetyistä formaateista videomateriaalin nauhoituksessa, pakkaamisessa ja jakamisessa.
IBL, Image Based Lighting	Valaistuksen renderöintitekniikka, jolla tavoitellaan todellista valaistusta hyödyntämällä kuvan (Environment Map) tai värin tarjoamaa informaatiota.
Import	Tiedoston tai mallin tuonti ohjelmaan.
Ink ‘n Paint -material	Materiaali, joka luo sarjakuvaisen vaikutuksen. Tarjoaa tasaisen varjostuksen mustutetulla reunuksella.

Inverse Kinematics	Animaatioketju, jossa hierarkian alin objekti vaikuttaa kaikkiin sen yläpuolella oleviin objekteihin.
JPEG	Häviöllinen kuvatiedoston pakkausformaatti, joka soveltuu hyvin tietokoneella ja verkossa esitettäviin kuviin.
Keyframe	Avainkehys, jonka animaattori asettaa aikajanelle. Avainkehys tallentaa liikkumistiedon, ja kahden avainkehysten välillä tapahtuva liike muuttuu animaatioksi.
NVIDIA:n Mental Ray	Yleiskäyttöinen renderöintimoottori, joka voi tuottaa fyysisesti oikeita simulaatioita valaistuksista, heijastuksista ja valon taitumisesta.
Mbps	Lyhenne tiedonsiirtotermistä megabittejä sekunnissa (megabit per second).
Mesh	3D-objektin geometrinen malli, jossa sen perusmuoto koostuu kärkipisteistä, jotka yhdistetään reunoilla.
Modifier	Objektien muokkaustapa, joka kattaa monimutkaisemmat ja usein automaattiset toimenpiteet kuten objektin poikkileikkaus tai peilikuvan luonti.
Motion blur	Epätarkkuus kuvassa, millä simuloidaan valon realistinen näkyminen liikkuesssa.
Named Selection Set	Nimetty valinta objekteista, jotka voi myöhemmin valita sen nimen perusteella luettelosta.
NURBS	Vektorimuotoinen objekti, jota käytetään lukemattomiin grafiikkamenetelmiin sekä tietokoneavusteiseen suunnitteluun.
Omni light	Suora valonlähde, joka ampuu valonsäteitä kaikkiin suuntiin yhdestä lähteestä.
Parent	Objekti, johon on linkitetty yksi tai useampi objekti.
Part-tiedosto	Tiedosto, joka sisältää kokoonpanon osien 3D-mallin ja rakenteen.

Patch	Pintatyyppi, joka omaa hieman kaa-revia polygonitasoja. Käytännössä pintatyyppi on polygonipinnan ja NURBS-käyrän risteytys.
Photon Mapping	Algoritmi, jota käytetään epäsuoran valon vaikutuksesta syntyvien efek-tien luontiin pinnoissa, kun rende-roidään Mental Ray-renderöintimoottorilla.
Png	Häviötön kuvatiedoston pakkaus-formaatti, joka sisältää läpinäky-vyysskanavan (Alpha). Alpha-kanavalla voidaan kuvankäsittelyssä luoda läpinäkyvyyttä kuviin.
Helper, Point Helper	Apuobjekti, jota käytetään animoin-nin tai mallien käsittelyn apuna.
Rasterization	Yksinkertainen render-algoritmi, joka soveltuu yksinkertaisiin ja no-peaa renderöintiäikää vaativiin tar-koituksiin.
Ray casting	Keskiverto render-algoritmi, joka soveltuu geometrian ja siihen koh-distuvan yksinkertaisen optiikan renderöintiin.
Ray tracing	Hyvin kehittynyt render-algoritmi, joka soveltuu hyvin realististen tu-losten saavuttamiseen pitkän rende-röintiajan kustannuksella.
Renderfarm	Monen tietokoneen hyödyntämä ko-konaisuus kuvan tai animaation ren-deröinnissä 3D-ohjelmasta.
Render to Elements, Render Passes	Renderöintitapa, jolla vain haluttu elementti voidaan renderöidä kuvas-ta (esim. valotus, heijastukset, väri).
Renderöinti, Rendering	Kuvan luonti mallista, antaa animaatiolle ja malleille viimeisen ulko-asun.
Rig (hierarkinen)	Järjestelmä, jolla animoija pystyy hallitsemaan ja kontrolloimaan ani-moitavaa objektiä tietyllä tavalla.
Riggaaja	Henkilö, joka suorittaa riggauksen.
Riggaus	Kontrollijärjestelmän luonti, jota animaation tekijä voi hyödyntää animaatioprosessissa.

Scanline	3ds Max:n vakio-renderöintimoottori, joka renderöi scenen riveittäin ylhäältä alas.
Scene	3D-ohjelmissa kohtaus tai kokonaisuus asioita, mitä käsitellään samaan aikaan.
Script	Ohjelma tai käskyjono, joka tulkitaan tai suoritetaan toisen ohjelman avulla tietokoneen prosessorin sijaan.
Shader	Prosessi, jolla muutetaan pintojen värien käyttäytymistä heijastaessa valoisuutta.
Skylight	Päivänvaloa tavoitteleva valomalli. Taivaan värin pystyy itse määrittämään, tai sen voi määrätä käyttäytymään bittikartan mukaan.
Spline	Käyrä, joka interpoloidaan kahden päätepisteen ja kahden tai useamman tangenttivektorin välillä.
Standard -material	Suoraviivainen materiaali pinnoille, joka simuloi pintojen heijastavia ominaisuuksia.
Still	Liikkumaton kuva, tai elokuvassa yksi filmin ruutu.
Surface Modeling	3D-objektien luomistapa kontrollipisteitä käyttämällä.
UV/UVW Mappaus	Työkalu, jolla kontrolloidaan materiaalien ja bittikarttojen näkymistä objektien pinnalla.
Visibility track	Rata, joka ohjaa objektin näkyvyyttä.
Vertex	Yksi polygonipinnan muodostavista kärkipisteistä.
VUE File Renderer	Renderöintimoottori, jolla tuotetaan jokaisesta framesta komentopohjainen .vue -tiedosto.



1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on perehtyä 3D-ohjelmien väliseen yhteensopivuuteen ja ongelmiin, kun tarkoituksena on toteuttaa arkkitehtikuvien perusteella tehdyistä 3D-malleista myyntiesittely animaationa. Animaation toteutusprojekti kytkeytyy suoraan mallien monimutkaisuuteen ja objektien lukumäärään, joten todellisten, fyysisten koneenosien perusteella luodut mallit ovat usein hyvin monimutkaisia ja sellaisenaan epäsoivia animaatioon.

3D-mallit suunnitteluun ja prototyyppien tekemiseen (CAD) ovat yleensä toteutettu hyvin insinöörimäisesti, toisin kuin animaatio ja visualisointi. Nämä mallit ovat hyvin parametrisia eli mallit on rakennettu todellisten mittojen mukaan. Tämä mahdollistaa hyvin tarkkojen mallien luomisen joita voidaan nopeasti muokata vastaamaan rakennemuutoksia. Nämä mallit ovat kuitenkin hyvin joustamattomia, ja poikkeaminen parametrisesta muokkaustavasta vaatii ohjelmalta mallin uudelleenlaskemista. CAD-mallit ovat myös hyvin raskaita, eli niiden tiedostokoot ovat suuria ja mallien pinnat omaavat suuren määrän polygoneja. Tämä ei ole ongelma jos 3D-malli pysyy luomisohjelmassaan, mutta siirtäessä mallin toiseen mallinnusohjelmaan ongelmia syntyy usein (Beanboxanimation 2013).

Tässä työssä on käytetty animaation ja visualisoinnin toteutukseen ohjelmaa Autodesk 3ds Max, alkuperäisten mallien käsittelyyn ja niiden siirtoon ohjelmaa SolidWorks 3D CAD sekä lopulliseen renderöintiin 3ds Maxin mukana tulevaa verkkorenderöintiohjelmistoa Autodesk Backburner. Videon jälkikäsittelyyn käytettiin ohjelmaa Adobe Premiere Pro CS6.

2 3D-TEKNOLOGIA

Suurelta osin televisiossa ja tietokoneen näytöllä esiintyvät kuvat, oli ne sitten paikoillaan olevia tai liikkuvia kuvia, on toteutettu kameralla tai videokameralla. Teknologia sai alkunsa vuonna 1951, kun ensimmäisellä videonauhurilla saatiin tallennettua reaaliaikaisia kuvia televisiokameroista muuttamalla kameran elektrisiä impulsseja ja tallentamalla sen informaation magneettiselle videonauhalle (Wikipedia 2015a).

Internetiä selailtaessa esim. autoja, huonekalukatalogeja tai vaatteita, kuvat jotka ruudulla näkyvät eivät välttämättä ole valokuvia, vaan kokoelma pikseleitä joiden kokoonpano on suoritettu tietokoneruudulla (Techonomy 2015).

Kameralla kuvattavalle teknologialle on viimeisten vuosikymmenien aikaan tullut haastaja, 3D-teknologia, joka lupaavasti näyttää etenevän digitaalisen esittämisen huipulle. Arviolta yli puolet Suomen televisiossa esiintyvistä mainoksista hyödyntää 3D-teknologiaa tavalla tai toisella. ”Kymmenen vuotta sitten noin 20 % autoteollisuuden markkinoinnista luotiin digitaalisesti ja 80 % oli tuotettu perinteisellä valokuvauksella. Tänä päivänä nuo luvut ovat päinvastaiset”, sanoi Midcoast Studion digitaalisten palvelujen johtaja Jay Dunstan vuonna 2012.

3D-kuvan luomisprosessi voidaan jakaa karkeasti kolmeen osaan: mallinnus, kokonaisuuden kasaaminen ja animointi, sekä visualisointi.

Ensimmäiseksi suunnitteilla olevasta tai fyysisestä objektista luodaan 3D-malli, joka voidaan toteuttaa proseduraalisesti mallinnusohjelmalla tai skannaamalla malli tietokoneelle fyysisistä objekteista.

Toiseksi mallinnetuista objekteista luodaan kokonaisuus (layout), eli esimerkiksi mallinnetun tuolin istuinosan alle siirretään siihen mallinnetut jalat. Jos tarkoituksena on tehdä kuvan sijasta video, luodaan mahdollinen liike (animaatio) objekteissa tai näiden välillä ajan edetessä.

Kolmanneksi luodaan toimenpiteet renderöintiin, eli luodaan malleille materiaalit, sopiva valaistus ja valitaan tarkoitukseen sopiva renderöintimoottori. Näitä renderöintimoottoreita on kehitetty eri tarkoituksiin, kuten fotorealististen lopputulosten saavuttamiseen simuloimalla oikeaa valaistusta tai taiteellisen lopputuloksen saavuttamiseen, missä fotorealisuus ei ole pääosassa. Lopuksi 3D-malleista renderöidään kuva tai video, ja tarvittaessa työstöä jatketaan videon-/kuvankäsittelyohjelmistossa. (Wikipedia 2015b.)

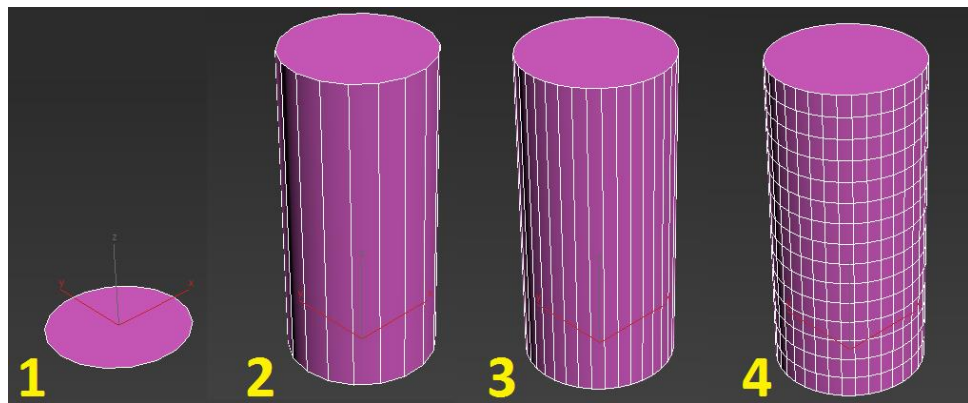
2.1 3D-mallit

3D-mallien luonti aloitetaan lähes aina ns. alkeisobjekteista, kuten kuutio, pallo tai sylinteri. Jos mietitään niiden koostumista, esimerkiksi kuutiohan koostuu neljästä neliöpinnasta. Jokainen neliöpinta muodostuu reunoista, ja pinnalla on myös 4 kulmaa. Nämä neliöpinnat ovat nimeltään polygoneja, ja niistä puhutaan nimellä face tai polygon. Reunat jotka ympäröivät neliöpinnan kutsutaan edgeiksi. Kun puhutaan polygonin kulmista, ei tarkoiteta minkä asteisen kulman ne luovat vaan kärkipisteitä, joista kulmat

muodostuvat. Näitä pisteitä kutsutaan vertex-pisteiksi. Yhteenvetona polygonipinnat koostuvat siis neliöpinnasta, reunoista ja kärkipisteistä. Hieman syvemmälle mentäessä nämä 3D-mallit siis omaavat ulottuvuuksia:

- Kärkipisteellä ei ole pituutta, leveyttä eikä korkeutta.
 - Reunalla on yksi ulottuvuus: pituus.
 - Polygonilla on kaksi ulottuvuutta: pituus ja leveys.
 - Kuutiolla on kaikki ulottuvuudet: pituus, leveys ja korkeus.
- (Chopine 2011, 3.0.)

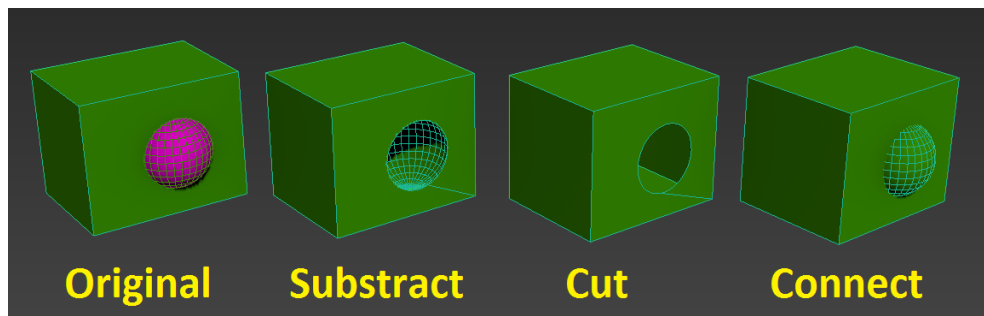
Esimerkkinä alkeisobjektin luomisesta on kuvassa 1 visuaalinen esitys sylinteri-alkeisobjektin luomisen vaiheista. Objektiin valitaan vaiheittain säde (1), korkeus (2), sivujen määrä (3) ja korkeuden segmentit (4). Alkeisobjekti sellaisenaan on aina tilavuusmalli, eli se on sisältä ”kiinteä” (Autodesk 3ds Max Help 2015 – Creating an Object).



Kuva 1. Cylinder-objektin luonti

Alkeisobjektia voi käyttää sellaisenaan, tai sille voi suorittaa perustoimenpiteitä (kuva 2) kuten

- toisen objektin ”vähennys” (subtract), jolla voi käytännössä tehdä esimerkiksi reiän tai poistaa osan objektista
- toisen objektin ”leikkaus” (cut), jolla leikataan kahden objektin yhdistämä osa esimerkiksi kokonaan pois
- kahden objektin ”yhdistäminen” (connect), jolla kahden objektin polygonipinnat yhdistetään toisiinsa



Kuva 2. Osa perustoimenpiteistä objekteille

Nämä perustoimenpiteet muuttavat mallin aina pintamalliksi, tehden niistä ”onttoja”, jolloin niiden alkuperäisiä parametreja kuten pallon halkaisija ei voi muuttaa enää. Tämän ilmiön huomaa helposti kuvan 2 toimenpiteestä Cut. Jos malliin halutaan tehdä muita kuin perustoimenpiteitä, täytyy se muuttaa joko Editable Polyksi tai Editable Meshiksi, jolloin työskentely tehdään aliobjektitasolla.

Pintamalleja on monia erityyppisiä, mutta tässä opinnäytetyössä käsitellään kolmea yleisimmin käytettyä tyyppiä (s. 4-6), ottaen huomioon eri ohjelmien välisen käytön: Editable Mesh, Editable Poly ja Body Object. (Autodesk 3ds Max Help 2015 – Basics of Creating and Modifying Objects.)

2.1.1 Tilavuusmalli ja pintamalli

Tilavuusmallien tarkoituksena on että objektit ovat usein osia, jotka lopulta valmistetaan valamalla, työstämällä tai puristamalla, ja ne voidaan rakentaa CAD-ohjelmassa yhdistämällä yksinkertaisia alkeisesineitä joita kutsutaan geometrisiksi alkimuodoiksi. Näitä muotoja on esimerkiksi kiinteä pallo, kartio, sylinteri tai suorakaide ja niitä yhdistellään käyttämällä (3-ulotteisia) Boolean-toimintoja ja lineaarisia muutoksia.

Eri Boolean-toiminnoilla muotoja voidaan lisätä objektista ja niitä voidaan vähentää siitä (tehdä reikiä). Esimerkiksi kasvattamalla suorakulmaisen tilavuusmallin sylinterin muotoista reikää tarkoittaa triviaalia muutosta – reiän tekevää sylinteriprimitiivin sädettä muuttamalla saadaan suurempi reikä. Tilavuusmallissa tämä toimenpide määrittää sen muodon sekä mallinnushistorian – alkuperäinen objekti ja siihen tehdyt muokkaukset luovat siis yhtenäisen mallin. Toisin sanoen tämä objekti on rakennettu lisäämällä alkeisobjekteja toisiinsa ja yhdistämällä ne.

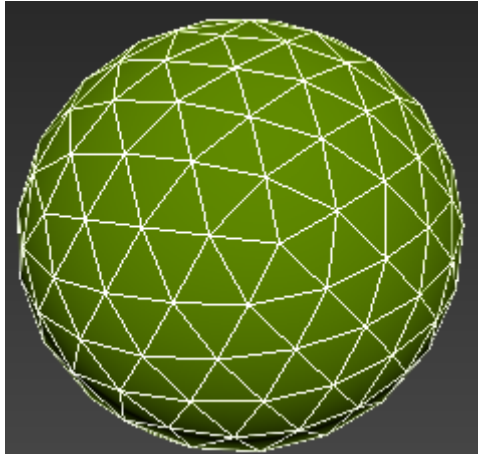
Tilavuusmallin muokkaustapa on ristiriidassa pintamallien kanssa, missä muokkaustoimenpiteiden suorittaminen ei tapahdu samalla tavalla. Vaikka sylinterimäisen pintamallin polygonit ovat helposti muokattavissa hierarkisessa mielessä, reiän sisäpinnat muodostuvat polygoneista ja niiden luonti tarkoittaa alkuperäisen polygonimallin luomistapaan kajoamista. Lisäksi huomioitavana seikkana mallin polygonimäärää täytyy kasvattaa säilyttääkseen yhtä hyvän tarkkuuden. (Watt 2000, 2.2.)

2.1.2 Editable Mesh

Polygonien ei aina tarvitse muodostua neliöpinnoista. Ne voidaan luoda millä tahansa määrällä reunoja ja kärkipisteitä (aina kuitenkin sama määrä kärkipisteitä kuin reunoja), mutta vähintään kolme. Kolmesta kärkipisteestä ja reunasta muodostuu kolmiopinta, jota kutsutaan nimellä triangle. Nämä yksinkertaiset pinnat ovat tietokoneelle kevyttä käsiteltävää, jonka takia niitä usein käytetään esimerkiksi pelimoottoreissa (Chopine 2011, 3.1).

Editable Mesh tarjoaa työkalut mesh-objektin manipulointiin, joka koostuu siis kolmiopinnoista ja omaa seuraavat aliobjektitasot: vertex (kärkipiste), edge (reuna) ja face (pinta), kuva 3. Suuren osan objekteista 3ds

Maxissa voi muuntaa Editable Meshiksi, mutta esimerkiksi spline-objekteja ei voi muuntaa, sillä muunnoksen jälkeen ne omaavat vain vertex-pisteitä, ei edge- tai face-operaatioita (Autodesk 3ds Max Help 2015 – Editable Mesh).

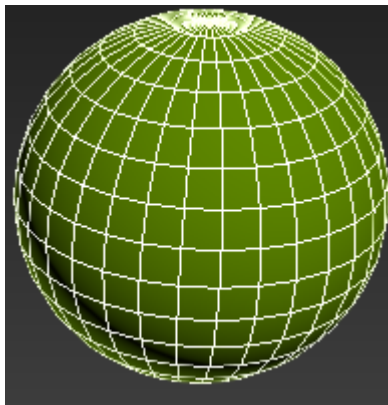


Kuva 3. Editable Mesh -objektin kolmiopinta

2.1.3 Editable Poly

Usein nelisivuinen polygonipinta ei ole neliö, vaan niiden reunat voivat kaikki olla eripituisia. Nelisivuista polygonipintaa kutsutaan nimellä quad, joka on lyhenne termistä quadrangle. Quadit ovat toinen tärkeä polygonipinta mallinnuksessa, sillä ne punoutuvat yhteen siistiksi, järjestäytyneeksi ruudukoksi (kuva 4) ja vaativat vain yhden lävistävän reunan rikkomista, jos ne halutaan muuttaa yksinkertaisiksi kolmiopinnoiksi. Nämä ja monet muut ominaisuudet ovat syynä siihen, että monet mallintajat käyttävät pelkästään quadeja mallinnuksessa (Chopine 2011, 3.1).

Editable Poly on muokattavissa oleva objekti viidellä objektin ala-tasolla: vertex (kärkipiste), edge (reuna), border (raja), polygon (monikulmio) ja element (elementti). Sen käyttö on samankaltainen Editable Mesh -objektin kanssa, ja se sisältää työkalut objektin käsittelyyn polygon-verkkopintana eri aliobjektitasoilla (Autodesk 3ds Max Help 2015 – Editable Poly).

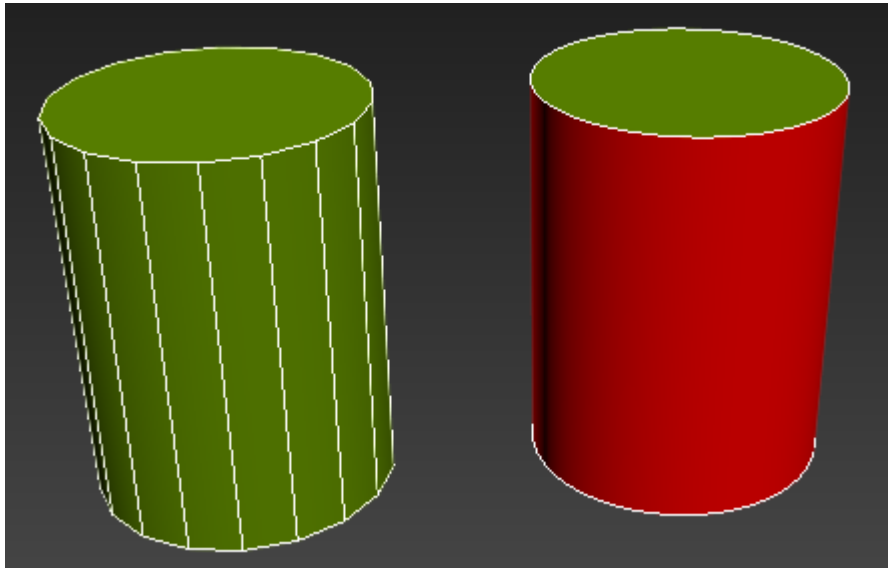


Kuva 4. Editable Poly -objektin neliöpinta

2.1.4 Body Object

Body Objecteja on mahdollista luoda 3ds Maxissa, mutta sen pääasiallinen käyttö on tukea ohjelmaan tuontia muista ohjelmista kuten Autodesk Inventor, Alias, Autodesk Showcase (Autodesk 3ds Max Help 2015 – Body Objects).

Työskennellessä Body Objectien kanssa muokkausta on mahdollista tehdä neljällä eri objektialatasolla: operand (operandi), edge (reuna), face (pinta) ja element (elementti). Koska Body Objectit eivät ole verkkopintoja vaan ne on määritelty proseduraalisesti, niiden rakenne poikkeaa normaalista 3ds Maxista löytyvistä objekteista. Esimerkiksi pyöreän objektin kuten sphere tai cylinder koko ympäryspinta voi koostua yhdestä tasopinnasta. Kuvassa 5 verrataan verkkopinnoista muodostuvaa objektia Body Object -tyyppiseen objektiin (Autodesk 3ds Max Help 2015 – Working with Body Objects at the Sub-Object Level).



Kuva 5. Verkkopintainen malli (vas.) ja Body Object (oik.)

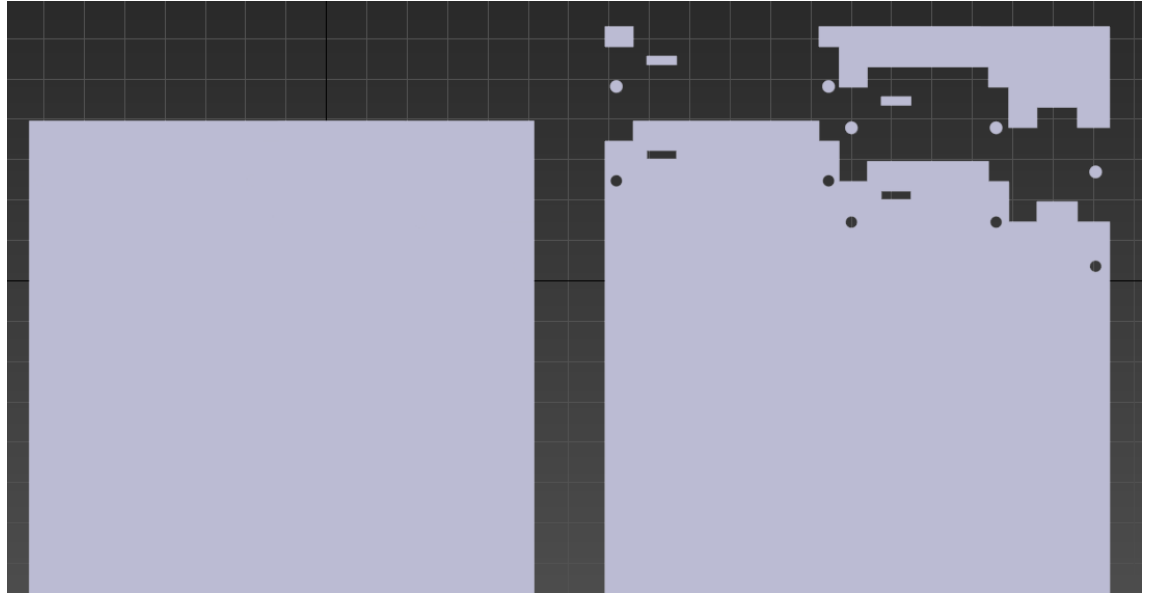
2.1.5 Transformaatiot

Malleille on mahdollista suorittaa kolme perustoimenpidettä: move (siirto), rotate (kääntö) ja scale (skaalaus), kuva 6. Objektiin todellinen paikka ja asento koordinaatistossa määräytyvät aina käytettävän koordinaatiojärjestelmän mukaan, joka vakiolta on näkymän (view) tai 3D-maailman (world) koordinaatisto. Koordinaatiojärjestelmä on mahdollista muuttaa myös esimerkiksi suhteessa näyttöön (screen), jolloin xyz-akselit ovat näytössä horisontaalinen, vertikaalinen ja syvyys (Autodesk 3ds Max Help 2015 – Reference Coordinate System).



Kuva 6. Perustoimenpiteet mallien käsittelylle

Omien päätelmiäni mukaan animaatioissa usein käytetään silmäkääntötemppeja, joilla saadaan katsoja uskomaan jotain tapahtuvan, vaikka todellinen toteutus on ollut muuta. Esimerkiksi kuvassa 7 esiintyvässä tasopinnassa (vas.) näyttäisi, että se on yksi, yhtenäinen objekti. Todellisuudessa tasopinta koostuu monesta eri objektista (oik.), jotka on saumattomasti liitetty yhteen. Näin saadaan animoitua osien irtoaminen pelkällä moveperustoimenpiteellä.



Kuva 7. Silmäkääntötemppe tasopinnalla

2.2 Animaatio

Yksinkertaisin kuvaus animaatiosta on ryhmä still-kuvia, jotka ovat hie- man erilaisia toisistaan ja ne esitetään peräkkäisessä järjestyksessä riittä- vän nopeasti niin, jotta silmämme uskovat jonkin liikkuvan. Mallianimaa- tiota tehtäessä on siis välttämätöntä toteuttaa 3D-mallien liikkuminen kuva kuvalta siten, että yleisö uskoo kuvissa esiintyvien 3D-mallien elossa olon (Beane 2012, 6.2).

Elokuviissa tavallinen kuvanopeus on 24 kuvaa sekunnissa, se on 30 kuvaa sekunnissa. Animaatiota tehdessä hahmojen tai mallien liikkeitä täytyy siis ohjata ohjaamisesta 24 kertaa sekunnissa luodakseen esityksen. 90 minuutin elokuva sisältää 129 600 kuvaa, ja jokaista kuvaa käsitellään, tutkitaan ja analysoidaan, jotta varmistutaan, että lopputulos on täydellinen (Beane 2012, 6.2).

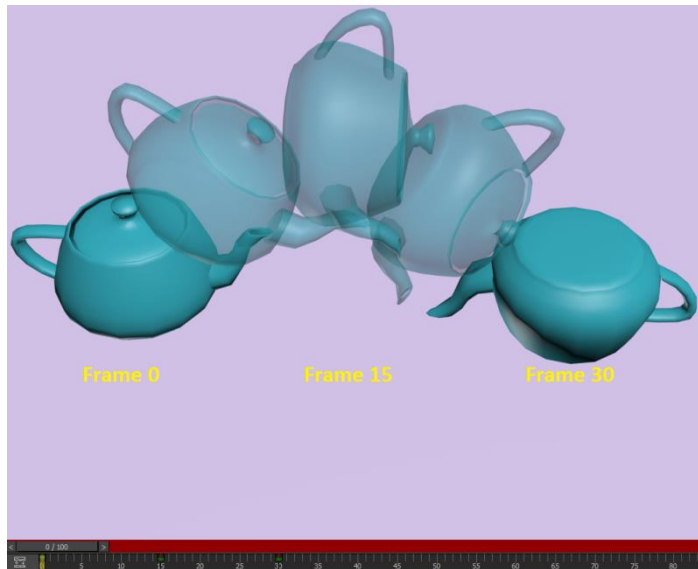
Jokaista animaation toimenpidettä ohjaa Controlleri, joka hallinnoi kolmea osa-aluetta:

- Se säilyttää animaation keyframet (ks. 2.2.1)
- Se säilyttää proseduraalisen animaation asetukset (ks. 2.2.2)
- Se interpoloi animaation aloitus- ja lopetuspisteen väliset framet. Tä- mä tarkoittaa sitä, että tietokone osaa interpoloidusti tuottaa liikkeen näiden paikkatietojen välillä.

(Autodesk 3ds Max Help 2015 – Controller.)

2.2.1 Keyframe-animaatio

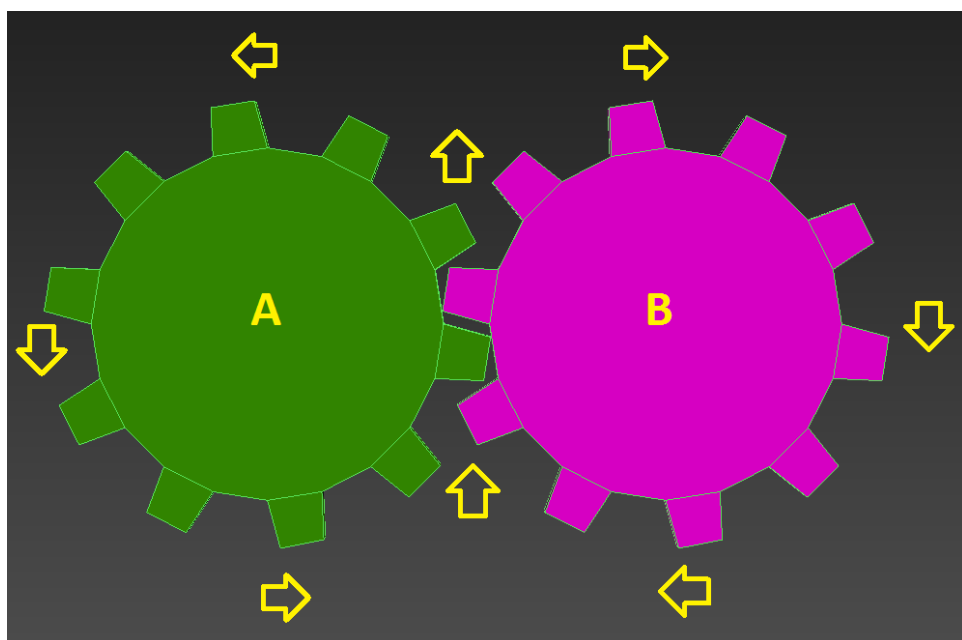
Keyframe on piirustus tai asento, joka luodaan näytettäväksi tietyssä ku- vassa. 3D-animaatiossa keyframe toimii merkinä, joka sijoitetaan aikaja- nalle. Nämä siis ovat avaintietoa objektin asennosta tai paikasta, jonka liikkeestä annetaan aloitus- ja lopetustiedot, ja tietokone interpoloi liik- keen näiden välillä, kuva 8. Käytännössä objekti asetetaan haluttuun paik- kaan ja asentoon ja luodaan keyframe, siirrytään aikajanalla, muutetaan objektin paikkaa ja/tai asentoa sekä luodaan uusi keyframe (Beane 2012, 6.2.1).



Kuva 8. Malli teepannusta eri ajanhetkillä

2.2.2 Proseduraalinen animaatio

Tämä animaatiotyyppi on kehitetty monimutkaisempien animaatioiden toteutukseen kuten simulaatiot (vesi, hiukset, kävely) ja objektien suhde toisiin objekteihin, ja se usein toteutetaan joko ohjelmasta löytyvillä työkaluilla tai erilaisilla scriptureilla. 3ds Maxista löytyy proseduraalisten animaatioiden toteuttamiseen monenlaista modifieriä, jolla siis objektin muotoa, läpinäkyvyyttä tai käyttäytymistä voidaan ajan muuttuessa manipuloida. Esimerkkinä proseduraalisesta animaatiosta on kahden hammasrattaan suhde toisiinsa, jolloin ratas A pyörimissuunta määrää rattaan B pyörimissuunnan (kuva 9) (Wikipedia 2014c).



Kuva 9. Hammasrattaiden suhde toisiinsa

2.2.3 Kamera-animaatio

Kamerat esittävät kohtausta tietystä näkökulmasta. Kameraobjektit simuloivat still-kuvia, elokuvia tai videokameroita reaali maailmassa.

Kameran näkymän voi säätää halutun linssin mukaisesti, eli näkymä olisi todentuntuinen kuin katsoessa oikean kameran linssin läpi. Kameranäkymä voi olla hyödyllinen geometrian muokkaamiseen sekä perustamalla kohtauksen renderöintiä varten.

On olemassa kahdenlaisia kameraobjekteja:

- Target Camera (kohteeseen katsova kamera) näyttää alueen kohdeobjektin ympärillä. Kamera ja sen kohdepiste voidaan animoida itsenäisesti, joten tämä kameratyyppi on helppokäyttöisempi silloin kun kamera ei liiku pitkin polkua.
- Free Camera (vapaa kamera) näkee alueen johon se on suunnattu. Vapaa kamera on käytännöllisempi kuin kameran sijainti ja asento määräytyy polun mukaan.

Jos halutaan animoida kamera sen näkökulmasta, voidaan luoda kamera ja animoida sen liike. Kamera-animaatio toimii samalla periaatteella kuin mallianimaatio, eli luodaan keyframet animaation aloitukselle ja lopetukselle, ja tietokone interpoloi näiden välin. Jos esimerkiksi halutaan lentää maiseman yli tai kävellä rakennuksen läpi, voidaan animoida liike ja halutessa kameran näkökenttä, jolla saadaan tuotettua esim. zoomaus kohtauksessa. (Autodesk 3ds Max Help 2015 – Cameras.)

2.3 Rigging

Kaikki 3D-geometria, joka tullaan animoimaan, tarvitsee jonkintyyppisen järjestelmän, josta animaattori saa vaaditun hallinnan ja joustavuuden halutun liikkeen toteuttamiseen. Tämä ohjausjärjestelmä on nimeltään rig (Beane 2012, 6.1).

Rigejä on olemassa monenlaisia: controller, constraint, helper, solver, wire parameter jne. Nämä rigit on kehitetty erilaisiin tarkoituksiin:

- Controllerit varastoivat perustoimenpiteiden keyframetiedot ja proseduraalisen animaation asetukset muutoksen interpolaation. Controllereita on kehitetty animoimaan esimerkiksi paikan, asennon, skaalauksen ja värin muutosta.
- Constraint on kehittyneempi controlleri, jota usein käytetään kahden objektin välistä suhdetta toisiinsa animaation toteuttamisessa. Esimerkkinä tähän voisi olla pallon nosto maasta, jolloin pallon paikka määräytyy nostohetkellä käden paikan mukaan, tai lentokone, joka kulkee ainoastaan spline-objektin muodostaman käyrän mukaan.
- Helper on apuobjekti, jota voidaan käyttää animaation tai työstön helpotuksessa. Nämä apuobjektit eivät renderöidy lopulliseen kuvaan, vaan ne antavat käyttäjälle visuaalista helpotusta työn eri vaiheissa, esimerkiksi etäisyyksien mitoitus.
- Solver on kinematiikan työkalu, jolla voidaan esimerkiksi vähentää korkealaatuisten hahmoanimaatioiden toteutukseen kuluva aikaa. Tästä aiheesta lisää luvussa 2.3.1.
- Wire Parameter sallii kahden objektin haluttujen parametrien linkityksen toisiinsa, eli esimerkiksi on mahdollista toteuttaa objektin A sijainnin vaikutus objektin B asentoon. Jos halutaan toteuttaa vaikutus moneen objektiin, voidaan luoda wire parameter -järjestelmä näiden objektien ohjaavien Helpereiden välille.

(Autodesk 3ds Max Help 2015 – Animation.)

2.3.1 Hierarkiat

Riggauksen olennainen osa on työskenteleminen hierarkioiden kanssa, tai oikeastaan päättäessä mitkä osat mallista ohjaavat toisia osia. Hierarkiassa ohjaava objekti on nimeltään parent, ja objektia, jota ohjataan, kutsutaan nimellä child. Hierarkian esityksessä näytöllä parent-objekti on child-objektin yläpuolella. On mahdollista luoda sisäisiä hierarkioita, jolloin alkuperäinen child-objekti toimii jonkin toisen objektin parent-objektina (Chopine 2011, 7.1).

Child-objekti voi liikkua, pyöriä ja skaalautua yksilöllisesti riippumatta parent-objektista, mutta kun parent-objekti liikkuu, child-objekti seuraa. On mahdollista luoda parent-suhde objektien välille, tai voidaan ryhmitellä objekteja jotta ne liikkuvat yksikkönä (Beane 2012, 6.1).

Monimutkaisempien animaatioketjujen tekemiseen, jota kutsutaan termillä kinematiikka, on kehitetty hierarkiatyypit Forward Kinematics (FK) ja Inverse Kinematics (IK). Nämä kaksi hierarkiatyyppiä ovat toistensa vastakohtia: FK-animaatioketjussa hallitseva (ja hierarkiassa ylimpänä) oleva

objekti vaikuttaa kaikkiin sen alaisiin, IK-animaatioketjussa taas hierarkiassa alimpana oleva objekti vaikuttaa sen yläpuolella oleviin objekteihin. Näistä kahdesta animaatioketjusta FK on yksinkertaisempi tapa toteuttaa hierarkia-animaatioita, mikä yksinkertaisena tosielämän esimerkkinä toteutuu ihmisen käden liikkeessä: olkapää on liikkeen ”hallitseva osa”, ja hierarkiamielessä kaikki sen alla olevat osat kuten kyynärpää, kämmen ja sormet liikkuvat olkapään liikkeen mukaan (Autodesk 3ds Max Help 2015 – Hierarchies and Kinematics).

2.3.2 Linkitys

Ennen linkityksen aloitusta on tärkeää miettiä linkityksen tarkoitusta ja sitä, miten sen toteuttaa. Toteutuksella on suuri merkitys mallin käytettävyyteen: hyvin suunniteltu hierarkia mahdollistaa monimutkaisten animaatioiden teon helposti, mutta huonosti suunniteltu voi tehdä animoinnin hyvin vaikeaksi. Kokonaisuuksien animointia varten on mahdollista tuottaa monta sisäkkäistä hierarkiaa, kuten jalan animoinnissa yksi hierarkia vyötäröstä nilkkaan ja toinen hierarkia kantapäästä varpasiin (Autodesk 3ds Max Help 2015 – Linking Strategy).

Objekteja on mahdollista linkittää myös käsiteltäväksi yhtenä objektina (Group) tai kokonaisuutena (Container). Nämä kummatkin tavat muuttavat hierarkian yhden objektin alle, mutta käyttäytyvät eri tavalla:

- Group luo objekteista teoreettisesti oman mallinsa, johon pystyy tekemään samoja toimenpiteitä kuin yksittäisiin objekteihin. Grouppiin on mahdollista siis tuoda modifiereita tai animoida sen läpinäkyvyyttä, ja kaikki geometria sen alla käyttäytyy grouppiin tehtyjen toimintojen mukaan.
- Container on organisoinnin helper-objekti, joka on suunniteltu helpottamaan objektien käsittelyä loogisten kokoelmien muodossa. Se käyttäytyy samalla tavalla kuin Group, mutta nämä kokonaisuudet on mahdollista tallentaa erilliseen tiedostoon, joka säilyttää geometriakokonaisuudet ohjelmien välillä.

(Autodesk 3ds Max Help – Group & Container.)

Toinen tehokas tapa yksinkertaisen animaation toteutukseen hierarkioiden sijaan on Named Selection Set eli nimetty valintakokonaisuus. On mahdollista valita scenestä halutut objektit, luoda näistä Named Selection Set ja myöhemmin käsitellä näitä objekteja helposti valitsemalla nimetty valintakokonaisuus. Tällä toimenpiteellä objektit eivät ohjelmassa linkity toisiinsa mitenkään, niiden liike ei vaikuta valintakokonaisuuden ulkopuolella muihin objekteihin, eikä hierarkia muutu alkuperäisestä poikkeavaksi, vaan objekteista luodaan kuvitteellinen valintakokonaisuus jota on helppo käyttää (Autodesk 3ds Max Help 2015 – Using Named Selection Sets).

2.4 Visualisointi renderöintiä varten

Valaisu, teksturointi ja renderöinti ovat 3D-kuvan tai videon tuotantoprosessin viimeisiä vaiheita, joiden yhteys toisiinsa tulee jatkuvasti ilmi työkennellessä. Nämä vaiheet ovat niin kytköksissä toisiinsa, että ei ole mahdollista muuttaa esimerkiksi valaistusasetuksia vaikuttamatta renderöintiin. Tässä luvussa käydään läpi jokaisen vaiheen perusteita, kuinka ne kytkeytyvät toisiinsa ja kuinka ne vaikuttavat visualisointiin (Beane 2012, 7.0).

2.4.1 Valaisu

Valaistus on erittäin tärkeää 3D-animaatioprojektissa, sillä valaistuksella ei ole tarkoitus ainoastaan välittää kuvan tunnelmaa ja tarkoitusta, vaan myös visuaalisesti kuvata kohtauksen sijaintia, kellonaikaa ja jopa säätä vakuuttavasti. Ymmärrys siitä, kuinka valo käyttäytyy todellisessa maailmassa, on tärkeää jotta pystyy luomaan tekaistun valon käyttäytymisen digitaalisessa maailmassa (Beane 2012, 7.9).

Kun halutaan luoda keinotekoinen valaisu, niin perinteinen lähestymistapa tähän on 3-pistevalaisu: luodaan siis avainvalo, täytevalo ja taustavalo. Avainvalo on pääkomponentti valaisussa, joka valaisee objektin edestäpäin. Vain yhtä valoa käyttämällä syntyy kuitenkin hyvin vahvat varjot, joita pehmentämään luodaan täytevalo. Lopuksi objektin taakse luodaan vielä taustavalo, joka antaa objektille hyvin erottuvat ulkoreunat (Chopine 2011, 13.0).

3D-ympäristön valaisu on vuorovaikutuksessa objektien kanssa monella tapaa. Valo käyttäytyy eri tavalla heijastuessa, taistuessa objektien (kuten lasi) läpi, ja kulkiessa läpikuultavan materiaalin kuten veden läpi. Se, kuinka valo käyttäytyy osuessaan 3D-objektiin, määräytyy objektin materiaalin shader-algoritmin perusteella, jota käsitellään luvussa 2.4.2 (Chopine 2011, 12.0).

3D-maailmassa valoa on kahdenlaista: epäsuora ja suora valo. Suoran valon heijastuminen objektiin toteutuu sen materiaalin shaderin perusteella, mutta valon jatkuva heijastuminen objektista pois päin on ainoastaan mahdollista epäsuoralla valolla, joka on Final Gather-algoritmin suorittama toimenpide. Jotta muut objektit voisivat ottaa vastaan tätä heijastuksen suorittamaa epäsuoraa valaistusta, täytyy ottaa käyttöön Global Illumination-algoritmi (GI). Perinteisen heijastuksen objektissa on mahdollista tehdä myös ”huijaamalla” eli käyttämällä Reflection Mappia. Tämä on siis materiaaliin asetettava bitmap, josta heijastus tulee ympäristön sijaan (Autodesk 3ds Max Help 2015 – Lights).

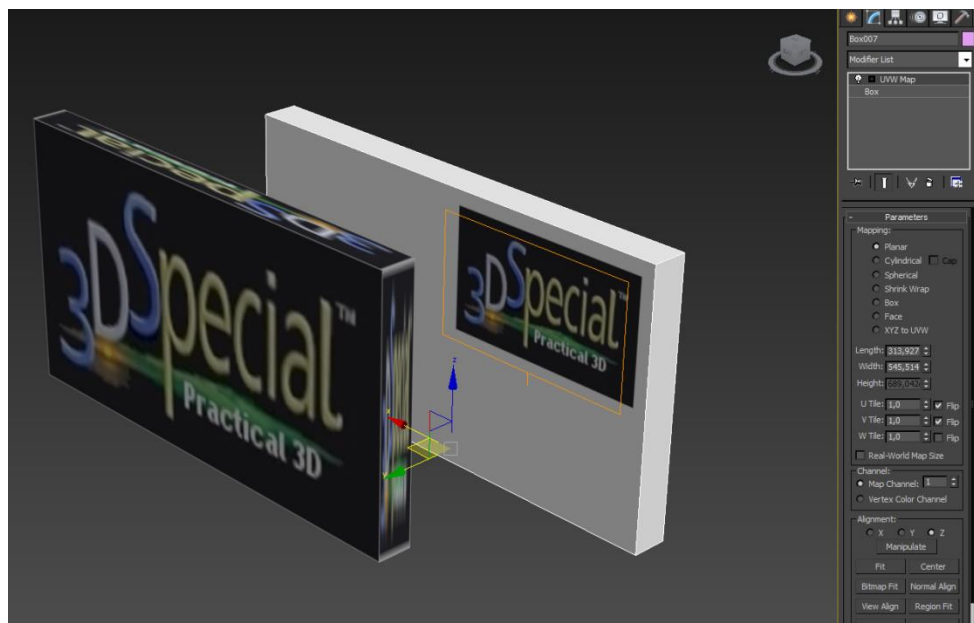
Valonsäteen osuessa läpinäkyvyyttä sisältävään objektiin, se kulkee objektin materiaalin läpi ja luo uuden valonsäteen. Jos pinta on käyrä, valonsäteen suunta muuttuu. Läpinäkyvä materiaali ja valaistus eivät myöskään ole yksiselitteisiä suhteessa toisiinsa, vaan uusi valonsäde ottaa vaikutuksia objektin paksuudesta ja väristä (Chopine 2011, 12.1-5).

2.4.2 Materiaaliparametrit ja shader-algoritmi

Objektin pintojen luomiseen tarvitaan aina jokin materiaali, joka omaa tietyn shader-algoritmin. Tällä luomisprosessilla yleensä tavoitellaan oikeiden objektien ulkonäön jäljittelyä tai sen artistista esitystapaa. Materiaalien ja tekstuurien luonnissa täytyy olla hyvä käsitys siitä, mikä tekee objektin näyttämään siltä kuin se on, ei niinkään fyysisessä mielessä vaan pinnan ja värin suhteessa. Ensiaskeleet oikealta näyttävään pintaan on siihen sopivan shaderin valinta, eli täytyy ymmärtää miten pinta käyttäytyy heijastaessaan valoa. Esimerkkinä keskeneräinen puupöytä; täysin uutena se ei ole heijastava, mutta omaa himmeän kiillon sen pinnalla, koska puukuidut eivät pakkaudu tiiviisti. Kun sama puupöytä ikääntyy ja sitä on käytetty aktiivisesti, puukuidut tiivistyvät, mikä muuttaa pinnan ulkonäköä tekemällä sen hieman heijastavaksi ja kiiltäväksi (Beane 2012, 5.33).

Teksturoinnissa usein käytetään bittikarttoja malleissa muodostamaan sen materiaalin täysin tai täydentämässä sitä. Esimerkiksi logon lisääminen malliin voidaan tehdä lisäämällä bittikartta materiaalin väriparametriin, tai halutessaan päällystää koko malli logolla. Usein tällä toimenpiteellä on sellaisenaan huonoja vaikutuksia: logon tilanteessa ohjelma ei ymmärrä kuinka bittikartta on tarkoitettu asettaa objektin päälle, vaan se vaatii materiaalin kartoituskoordinaatit (UV-koordinaatit), jotka saadaan tuotettua työkalulla UV Map. Tämä työkalu tarjoaa mahdollisuudet kuva käärimiseen eri tavoilla, sen skaalaukseen objektissa ja monet muut eri toiminnot (Beane 2012, 5.33).

Kuvassa 10 esiintyy kaksi box-primitiivistä tehtyä seinää, joihin on lisätty logot bitmappeina. Vasemmanpuoleisessa seinässä ei ole käytetty UVW Mappia, ja se simuloi tilannetta jossa bittikartta on suoraan linkitetty objektiin. Kuvasta selviävät myös UVW Mappauksen perustyökalut.



Kuva 10. Logon näkyminen seinällä ilman UVW Mappia, ja logon näkyminen seinällä UVW Mapilla (+ sen parametrit).

2.5 Renderöinti

Renderöinti on tuotannon loppuvaihe, jolla täydennetään geometriaa väreillä, valaistuksilla, varjoilla, heijastuksilla, jne. Tämän jälkeen geometria voidaan “tulostaa” valitulla render-algoritmilla kuvaksi tai videoksi. Yleensä vakioasetuksilla renderöity kuva tai video ei vastaa kuitenkaan sitä, miltä sen pitäisi näyttää, vaan renderöintiin vaikuttaa geometriaa täydentävien elementtien lisäksi monta eri seikkaa:

- Render-algoritmi, jota käytetään
- Renderöintimoottorin ja ympäristön asetukset
- Gamma-asetukset
- Lopullinen tiedostomuoto

(Autodesk 3ds Max 2015 – Rendering.)

Kuvassa 11 on verrattuna viewport-esikatselumalli ja NVIDIAN Mental Ray -renderöintimoottorilla renderöity malli teepannusta, joka on asetettu tasaiselle alustalle.



Kuva 11. Viewport-esikatselumalli (vas.) ja renderöity malli (oik.)

2.5.1 Render-algoritmi

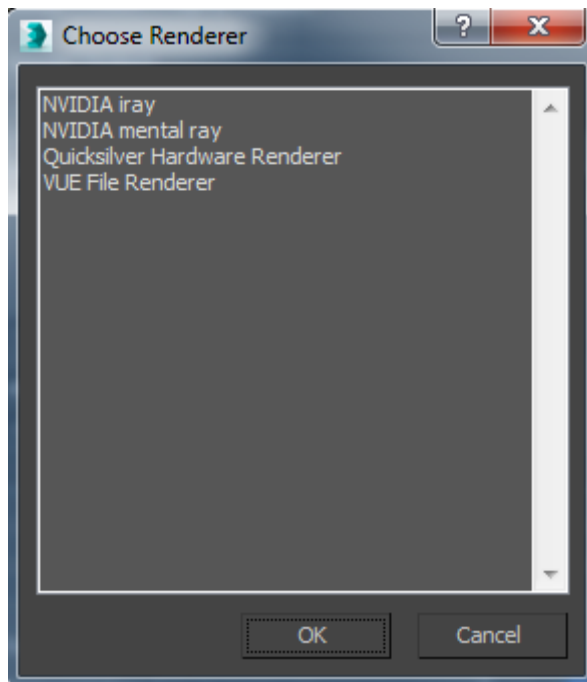
Render-algoritmeja on kehitetty jo vuosikymmeniä, mutta vain muutama tekniikka on säilynyt käytössä: rasterization, ray casting ja ray tracing. Rasterization-tekniikka (3ds Maxissa moottorina Default Scanline) on yksinkertaisin ja kaikkien näytönohjainten tukema renderöintitapa, jolla geometrian muodot piirretään pikseli pikseliltä tietokoneen näytölle. Tämä algoritmi ei ota huomioon tyhjiä alueita, joissa geometriaa ei esiinny eikä se tue kehittyneempää optiikkaa kuten luonnollinen valaistus. Rasterization-tekniikka on hyvä reaaliaikaisille sovellutuksille kuten videopelit, joissa lyhyt renderöintiaika on tärkeää.

Ray casting-tekniikka on keskivertainen renderöintitapa, joka scene renderöidään tietyistä näkökulmista katsottuna. Tämä algoritmi piirtää kuvan geometrian sekä heijastumisen yksinkertaisimmat optiikan lait. Ray casting eroaa kehittyneemmästä Ray tracing-algoritmista siten, että se ei osaa jäljitellä valonsäteiden jatkumista objektin pintaan osuessa; valo todelli-

suudessa kimpoaa objektista pois päin, ja taittuu kulkiessaan läpinäkyvän objektin kautta, kuten valaisu-kappaleessa läpi käytiin.

Viimeisenä algoritmeista on Ray tracing-tekniikka (3ds Maxissa moottorina mm. NVIDIAN Mental Ray), kaikkein kehittynein algoritmi, joka hyödyntää kehittyneempiä optisia simulointitapoja. Algoritmin päätarkoitus on mahdollistaa hyvin realististen lopputuloksien saavuttaminen pitkän renderöintiajan kustannuksella. Tämä algoritmi luo kuvan jäljittelemällä valon kulkua 3D-ympäristössä ja simuloimalla valon vaikutusta objekteihin, jotka se kohtaa. Ray tracing-tekniikka soveltuu parhaiten yleisesti markkinoinnin tai televisiossa esitettävän 3D-materiaalin renderöintiin, jossa realistinen lopputulos on tärkeämpi kuin lyhyt renderöintiaika. (Wikipedia 2015d.)

Kuvassa 12 esiintyy kaikki 3ds Maxin mukana tulevat renderöintimoottorit, joista Default Scanline Renderer käyttää rasterization-algoritmia, muut ray tracing-algoritmia. Kuvasta on jätetty pois VUE File Renderer, joka luo .vue -tiedoston eikä sitä käytetä suoraan 3ds Maxista kuvan tai videon renderöintiin.



Kuva 12. 3ds Maxin renderöintimoottorit

2.5.2 Renderöintimoottorin ja ympäristön asetukset

Renderöintimoottoreilla on niihin erikseen kehitetyt asetukset, mutta tässä työssä tarkastellaan Mental Ray –renderöintimoottorin asetuksia.

Mental Ray omaa siis laajimman skaalan eri asetuksista, joilla lopputulokseen voi vaikuttaa. Asetukset jakautuvat kahteen kategoriaan: renderöinti-asetukset ja valaisuasetukset, joiden mahdollisuudet ovat. Renderöinti-asetusten mahdollisuuksiin sisältyy perusteet selittäen..

- Anti-aliasing eli näytteenoton tarkkuus: voidaan määritellä, kuinka monta näytettä jokaisesta pikselistä otetaan, joiden mukaan sen sisälön väri määräytyy.

- Render-algoritmin valinta, joka MR-moottorissa on lukittuna ray tracing-algoritmiin. Lisäksi on mahdollista valita valon heijastumisen ja tiattumisen salliminen, sekä näiden maksimiarvot pikseliä kohden.
- Kamera-efektit, joihin sisältyy mm. liike-epäterävyuden (motion blur), kameravarjostimien ja syvyysvaikutelman säädöt
- Varjostusasetukset, jotka sisältävät mm. varjojen sallimisen, muodostumistavan ja pehmeiden.

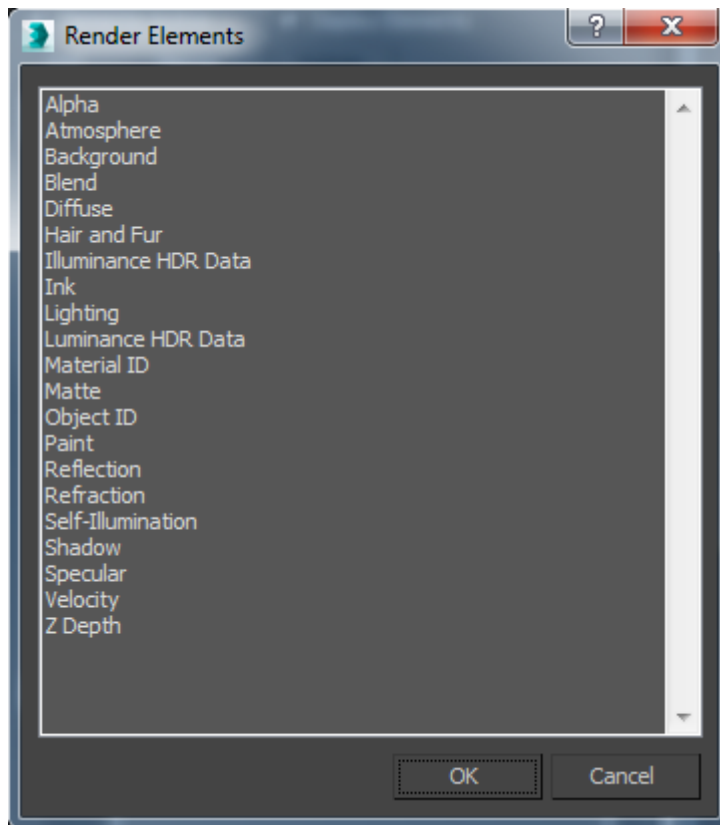
Valaisuasetukset jakautuvat karkeasti myös kolmeen eri kategoriaan..

- Ympäristön tai taivaan valaistuksen laskentatapa: Final Gather tai IBL (Image Based Lighting). Vakioasetuksilla laskentatapa on IBL, joka siis laskee valaistuksen ympäristön väristä tai siihen asetetusta bitmapista (Environment Map). Jotta tämä olisi käytännössä mahdollista, täytyy sceneen luoda Skylight -valo-objekti.
- Final Gather asetukset: tämä tekniikka on haluttaessa mahdollinen, ja sen asetukset sisältävät mm. kimpoavan valon vahvuuden ja tarkkuuden.
- Valon heijastuksien ja taantumisien asetukset objekteissa (Caustics & Photon Mapping [GI]): asetukset fotonien eli valokvanttien määrän jokaista valoa ja objektia kohden. Vakioasetuksilla caustics ei ole päällä, ja jotta sitä voidaan käyttää, täytyy valita vähintään yksi objekti tai valo joka tuottaa fotoneita, sekä yksi objekti joka vastaanottaa näitä.

(Autodesk 3ds Max Help 2015 – mental ray Renderer Interface.)

Scenestä on myös mahdollista renderöidä jokainen elementti omaksi kuvakseen, jotka myöhemmin kootaan kuvankäsittelyohjelmassa lopulliseksi kuvaksi, eli ne kompositoidaan. Tämä tapa on nimeltään Render to Elements (tai Render Passes), joka mahdollistaa esimerkiksi pelkästään varjostuksien, valotuksen tai heijastuksien renderöinnin. Render to Elements vaatii 3D-ohjelman lisäksi kuvankäsittelyohjelman sekä hieman lisätyötä kompositoinnin toteuttamiseen, mutta jos esimerkiksi valotus on vääränlainen ja sitä pitää muuttaa, voidaan renderöidä scenestä pelkkä valoisuus-elementti. (Beane 2012, 7.30)

Kuvasta 13 selviää 3ds Maxin mahdollisuudet renderata elementtejä erikseen scenestä.



Kuva 13. Render to Elements -ikkunan vaihtoehdot

Vaikka aihe on saanut vähän huomiota, on tärkeää ymmärtää ihmisen näköhavainnon rajat renderöintiä tehtäessä. Pääsyyinä tähän on se, että ihmisen näköaisti ja kehitetyt näytöt (TV, tietokone) ovat rajoittuneita. Renderöintimoottori voi simuloida lähestulkoon loputtoman valikoiman valon kirkkauksia ja värejä, mutta nykyiset näytöt eivät pysty käsittelemään niin paljoa, ja jotain renderöidystä datasta on hävitettävä tai pakattava. Ihmisen näköaistilla on myös rajansa, eikä näin ollen ole tarvetta tuottaa todella suuria tai tarkkoja kuvia luodakseen realismia (Wikipedia 2015d).

2.5.3 Gamma-asetukset

Kun kuvia ja videoita katsotaan eri näytöillä, tai kuva tulostetaan paperille, yleensä niiden kirkkaus on erilainen. Tämä johtuu Gamma-arvosta, joka määrittää kuinka lineaarinen siirtyminen täysin mustasta väristä täysin valkoiseksi väriksi on. Täydellinen siirtyminen onnistuisi gamma-arvolla 1, mutta tämäntyyppisiä näyttöjä ei ole olemassa. Gammakorjauksella korjataan siis näyttölaitteiden ja kuvantamisjärjestelmien epälineaarisuuksia ja kontrastieroja, jotta kuva näyttäisi samanlaiselta jokaisella kuvaruudulla että paperitulosteena.

Tietokonenäytöt ovat epälineaarisia laitteita, joiden perinteinen gamma-arvo vaihtelee 1.5 ja 2.2 välillä, ja näiden näyttöjen perinteinen gammakorjausarvo on 2.2. Käytännössä tämän korjaustoimenpiteen voi tehdä näytön asetuksissa, ja videopeleissä joissa valoisuusasteet ovat tärkeitä (kuten kauhupelit) gammakorjaus suoritetaan yleensä ennen pelin aloitusta. (Autodesk 3ds Max Help 2015 – Gamma Correction.)

2.5.4 Lopullinen tiedostotyyppi

Renderöinnissä aina valitaan tiedostotyyppi, johon lopullinen kuva tai animaatio renderöidään. Riippuen tiedostotyypistä, renderöinnissä on mahdollista valita myös lopullisen tiedoston asetukset pakkaukselle, värisyvyydelle ja sen laadulle (Autodesk 3ds Max Help 2015 – Render Output File Dialog).

Kuvaa renderöidessä tiedostotyyppi ja sen asetukset jakautuvat karkeasti kahteen osaan, jotka vaikuttavat lopputulokseen:

- Kuvakoko ja kuvasuhde: tällä siis määritellään millä resoluutiolla ja minkälaisella näytöllä se on tarkoitus näyttää. Resoluutio valitaan sen mukaan, onko lopullinen esitystapa tietokoneen/television näytöllä vai paperilla. Parhaan lopputuloksen saavuttamiseen verrataan esitystavan tyyppiä piirrettävän kuvan DPI-arvoon, joka siis näytöstä puhuttaessa tarkoittaa kuinka monta pikseliä ruudun jokaisella tuumalla esiintyy, ja paperilla puhuttaessa kuinka monta mustepistettä jokaisella tuumalla esiintyy. Nykynäytöt käyttävät lähes kaikki kuvasuhdetta 16:9 ja niiden resoluutioksi kuvan esittämisessä riittää 1920x1080 (arkisempänä terminä FullHD), joten perinteisesti 72 DPI riittää. Paperille tulostaessa tarvitaan suurempi DPI-arvo hyvän kuvanlaadun saavuttamiseksi, joka yleisesti on 300 DPI. Mitä suuremmalle paperille tulostetaan, sitä pienempi DPI-arvo yleensä kuitenkin valitaan, sillä kuva ei tulla katsomaan lähietäisyydeltä ja sen renderöintiäika sekä tiedostokokoon tulisivat hyvin suuriksi. Laskentakaavaksi esim A4-kokoiselle paperille tulostettavaksi on sen leveys 210x297mm, mikä tuumina on 8.27x11.69, kerrotaan 300 DPI:llä mistä saadaan 2480x3508 pikseliä. Tämä on siis tulostustarkoitukseen tarvittava renderöintiresoluutio. (Beane 2012, 16.1)
- Pakkaustapa: tämä määräytyy tiedostotyypin mukaan, jossa pystyy tiedostotyypistä riippuen valitsemaan värisyvyyden, mahdollisen alpha-kanavan säilytyksen ja lopullisen laadun. Monet eri kuvatiedostot on suunniteltu eri tarkoituksiin: yleisimmin verkossa esiintyvä

JPEG-tiedosto on häviöllinen pakkausformaatti, joka luo pienen tiedostokoon ja soveltuu sellaisenaan sopivaksi verkkoon. JPEG tukee 8-bittistä mustavalkokuvan tallennusta ja 24-bittistä värikuvan tallennusta, mutta välttämättömän pakkauksen seurauksena ei sovi jälkikäsitteilyyn tai tulostukseen. Jos kuvalle on tarkoitus tehdä jälkikäsitteilyä tietokoneella, sopiva kuvaformaatti on esimerkiksi PNG, joka on häviötön pakkaustapa ja se säilyttää tarvittaessa kuvasta alpha-kanavan. PNG tukee mustavalkokuvan tallennusta 8- ja 16-bittisellä värisyvyydellä, ja värikuvia 24- ja 48-bittisellä värisyvyydellä. Tulostustarkoituksiin sopiva tiedostoformaatti on TIFF, joka voidaan tallentaa häviöllisenä tai häviöttömänä kuvana värisyvyydellä 8-bit tai 16-bit, ja se tukee hyvin eri laitteiden väriavaruuksia kuten CMYK, jota usein käytetään kuvien painotuksessa. (Wikipedia 2015e.)

Kun tarkoituksena on renderöidä animaatio, ei ole suositeltavaa renderöidä suoraan videotiedostoon, sillä se pakkaa kuvia liikaa eikä tuota parasta mahdollista laatua, vaan paras tapa on renderöidä animaatio kuvasarjaksi jokainen frame kerrallaan. Kuvasarja säilyttää laadun pakkaamattomana, ja mahdollisen sähkökatkoksen, ohjelman kaatumisen tai muiden ongelmien tapahtuessa kuvasarjan renderöinti voidaan jatkaa siitä, mihin frameen ohjelma oli kerennyt renderöidä. Videotiedostoon renderöitäessä koko tiedosto ”jää kesken”, ja prosessi joudutaan aloittamaan alusta. Kuvasarjaksi rendattujen kuvien yhdistäminen videotiedostoksi voidaan suorittaa videonkäsittelyohjelmistossa, kuten Adobe Premiere Pro CS6. Videokäsittelyohjelmat omaavat laajemman skaalan eri pakkausalgoritmeja, ja lopullinen tiedosto saadaan paljon laadukkaammaksi (Autodesk 3ds Max Help 2015 – Image File Formats).

2.6 Network Rendering

Projekteissa jotka vaativat paljon suorituskykyä (kuten animaatioiden tai suurten kuvien renderöinti), voidaan käyttää useaa tietokonetta apuna. Tätä renderöintitapaa kutsutaan nimellä Network Rendering eli verkkorenderöinti. Tietokoneverkkoa, joka on suunnattu tähän tarkoitukseen, kutsutaan usein nimellä render farm.

Tätä tapaa käyttäessä jokaisella tietokoneella pitää olla client – pienempi versio renderistä joka kommunikoi palvelimen kanssa. Verkkorenderöinnissä yksi tietokone toimii palvelimena, joka määrää tehtävien jakamisen jokaiselle verkossa olevalle tietokoneelle, jotka renderöivät oman osansa työstä ja lähettävät sen takaisin palvelimelle, joka kasaa nämä palaset takaisin yhteen. Verkkorenderöinti animaation tapauksessa yleensä toteutetaan niin, että jokainen tietokone saa lopullisen kuvasarjan muodostamasta animaatiosta tietyn määrän kuvia renderöitäväksi.

Verkkorenderöintiä kannattaa hyödyntää vain suuremmissa projekteissa, sillä verkon välisen viestinnän ja tehtävähallinnan luomisessa menee oma aikansa. Chopine 2011, 16.7.)

3 3D-OHJELMIEN VÄLISET EROT

Polygonipinta kärsii monista haitoista kun objekti on monimutkainen ja omaa suuren määrän yksityiskohtia. Valtaosassa tietokonegrafiikkaa polygonien määrä objektin esityksessä vaihtelee muutamasta kymmenestä polygonista satoihin tuhansiin polygoneihin, millä on vakavia seurauksia renderöinti-aikaan ja mahdollisuuksiin käyttää kyseisiä objekteja animaatioissa tai virtuaalisessa ympäristössä. Muita ongelmia ilmaantuu animaatioissa jossa mallin täytyy jäljitellä objektin muotoa ja muuttaa muotoaan tai törmätä toisiin objekteihin.

Tästä huolimatta polygonimalli on tietokonegrafiikan ylintä antia. Sen hitaus on osana tehokkaiden algoritmien ja renderöitävien laitteiden kehitystä. Tämä on johtanut jossain määrin outoihin lopputuloksiin, missä on tehokkaampaa – kun mietitään renderöintiä – esittää muoto yksinkertaisemmilla elementeillä (polygons) kuin esittää se vähemmällä määrällä (ja tarkemmalla) mutta monimutkaisemmilla elementeillä, kuten parametrisillä bi-cubic -pinnoilla. Nämä pinnat ovat käytännössä kaarevia polygonipintoja, joiden muoto määräytyy NURBS-käyrien mukaan. NURBS-käyrät ovat vektorimuotoisia objekteja, joita käytetään lukemattomiin grafiikkamenetelmiin sekä tietokoneavusteiseen suunnitteluun. Yleisempi nimitystapa bi-cubic pinnoille on Patch, jolla nämä myös 3ds Maxissa tunnetaan. (Watt 2000, 2.0.)

3.1 Ohjelmien yhteensopivuus

3D-mallien eri esitystavoissa on omat etunsa ja haittansa, mutta kaikkiin olemassa oleviin ongelmiin ei ole yhtä yleistä ratkaisua. Pikemminkin tietyt mallinnustavat on kehitetty tiettyihin tarkoituksiin. (Watt 2000, 2.0.)

Viihdeteollisuudessa useimmiten käytetään ohjelmia kuten Autodesk Maya tai 3ds Max. Näillä ohjelmistoilla voi tehdä muutakin kuin vain mallinnuksen – kuten animaation, fyysisen dynamiikan simuloinnin, teksturoinnin, valaistuksen, renderöinnin ja kompositoinnin. Näiden ohjelmistojen tuetut mallinnustavat ovat polygonaalinen mallinnus (polys), epäyhtenäinen rationaalinen spline (NURBS) ja surface modeling.

Arkkitehtuurin ja tuote-visualisoinnin teollisuudessa mallinnetaan usein ohjelmistoissa kuten Autodesk AutoCAD, SolidWorks -tuotteet ja Google SketchUp Pro. Nämä ohjelmistot eivät tarjoa yhtä paljon kaikenkattavaa käyttöä animoinnin, dynamiikan tai renderöinnin puolesta, mutta ovat erinomaisia tarkan mittakaavan tuotemallinnuksessa. Nämä CAD-ohjelmistot pääosin käyttävät parametrissa tilavuusmallinnustekniikkaa, mikä käytännössä tarkoittaa sitä, että mallintaja piirtää esineen ääriviivat ja ohjelma täyttää sen sisustan. (Beane 2012, 2.10.)

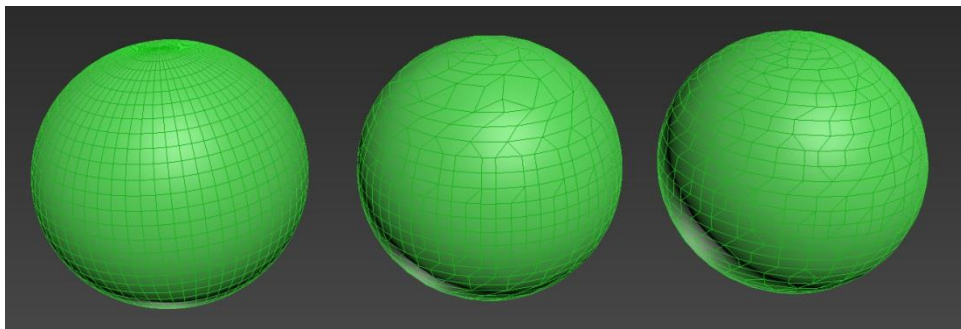
3D-mallintajat voivat viedä malleja tiedostoihin, jotka voidaan luoda Export-työkalulla muihin ohjelmiin siirtoa varten ja vastaanottaa Import-työkalulla, kunhan tiedoston metatiedot ovat yhteensopivia ohjelmien välillä (Wikipedia 2015f).

3.2 Mallien optimointi

Mallien ja scenen optimointiin vaikuttaa geometrian puolesta kaksi asiaa: kuinka monta objekta scenessä on ja kuinka polygonipintaa tai vertex-pistettä ne omaavat. Polygonipintojen tai vertex-pisteiden vähentäminen (toisin sanoen Meshin yksinkertaistaminen) onnistuu 3ds Maxissa helpoiten Modifiereillä, jotka ovat siis työkaluja joilla objektien geometriaan voidaan vaikuttaa. Näitä modifiereita on meshin yksinkertaistamiseen kaksi:

- Optimize-modifier, joka mahdollistaa automaattisen polygonipintojen ja vertex-pisteiden vähentämisen yhdistämällä niitä mahdollisuuksien mukaan ilman, että lopullinen geometria kärsii silmännähdessä huomattavasti. Tämä modifieri nopeuttaa vaadittavaa renderöintiä, mutta saattaa karsia tarkkoja yksityiskohtia sisältävää geometriaa käyttäjän huomaamatta.
- Multires-modifier, joka toimii samalla vähentämistekniikalla kuin Optimize, mutta sisältää tarkemmat asetukset polygonipintojen ja vertex-pisteiden vähentämiselle. Vähentäminen voidaan suorittaa joko poistamalla tietyn prosentin vertex-pisteistä tai rajaamalla poistomäärään vertex-pisteitä.

Modifierit ovat erityisen hyviä mallien optimointiin siksi, että ne säilyttävät alkuperäisen geometrian, johon käyttäjä voi halutessaan palata. Kuvassa 14 on vertailtu geometrian ulkomuotoja, kun alkuperäisestä 3480 polygonipinnan pallosta optimoitiin geometria. Optimize-modifierin vakioasetuksilla pallo optimoitiin 1518 polygonipintaan, ja lähelle samaa lopputulosta Multires-modifierilla päästiin säilyttämällä 44 % prosenttia vertex-pisteistä (1528 polygonipintaa).



Kuva 14. Alkuperäinen malli (vas.), malli Optimize-modifierilla (kesk.) ja malli Multires-modifierilla (oik.). Kuten kuvasta huomaa, suurta geometrian muutosta ei kummassakaan tapauksessa tapahtunut, mutta lähes 50% kevyempi malli tuotti hieman kulmikasta pintaa palloon.

Omien päätelmiäni mukaan puhuttaessa mallien optimoinnista ei ainoa ratkaisu ole käsitellä mallien pintojen monimutkaisuutta, vaan myös objektien tarpeellisuutta katsojalle. CAD-mallit luodaan todellisuuden mukaan, eli esimerkiksi malli tietokoneesta sisältää kaikki sen komponentit kuten kovalevyt, tuulettimet ja nämä koteloon kiinnittävät ruuvit. Jos tietokonetta halutaan esitellä vain ulkoapäin, ovat sen sisällä olevat esineet lisärasitteena tietokoneelle niin työstön kuin renderöinnin kannalta. Tähän ongelmaan kaksi ratkaisua:

- Turhien objektien piilottaminen: tämä tapa vähentää koneelta vaadittavaa tehoa, mutta haluttaessa objektit on mahdollista saada näkyviin.
- Turhien objektien poistaminen: tämä tapa omaa samat edut kuin piilottaminen, ja lisäetuna on työtiedoston koon pienentyminen. Haittana tässä tavassa on se, että objekteja ei ole mahdollista saada näkyviin enää, vaan ne on tuotava tarvittaessa uudestaan ohjelmaan.

(Autodesk 3ds Max Help 2015 – Modifiers.)

3ds Maxiin on kehitetty myös scriptityökalu, MAXScript, joka on todella tehokas lisä ohjelman käytössä. Sen käyttö päivittäin voi vaikuttaa huomattavasti tehokkuuteen, ja sillä on mahdollista toteuttaa toimenpiteitä, jotka eivät alkuperäisessä ohjelmistossa ole mahdollisia. Kuten lisäosien käytössä, myöskään ilmais- tai mainoskäyttöön tehtyjen valmiiden scriptien käyttö ei vaadi niiden luomiseen käytettävän ohjelmointikielen osaamista. (Autodesk 3ds Max Help 2015 – MAXScript for New and Casual Users.)

Tässä työssä MAXScriptiä käytettiin mallien mesh-pintojen yksinkertaistamiseen, mitkä olisi ilman scriptiä jouduttu suorittamaan objektikohtaisesti.

4 TUOTANTOLINJASTON MYYNTIESITTELYN TYÖNKULKU

Opinnäytetyön kohteena oli siis yrityksen tuottavin lävistyslinjasto, josta oli tuotettu 3D-suunnittelumalli (CAD) linjaston rakentamista varten. Tämän mallin esittelyyn oli aikaisemmin käytetty videokameralla kuvattua videomateriaalia ensimmäisestä rakennetusta linjastosta, mutta laitteen toimintaa ei saatu esiteltyä tarpeeksi yksityiskohtaisesti eikä sen mallinusohjelmalla ollut mahdollista tehdä fotorealistista tavoittelevaa visualisointia. Lähtökohtana opinnäytetyössä oli selvittää CAD-suunnittelumallien siirtomahdollisuudet visualisointiohjelmaan, toteuttaa sen toiminnan animaatio ja visualisointi sekä lopuksi renderöinti.

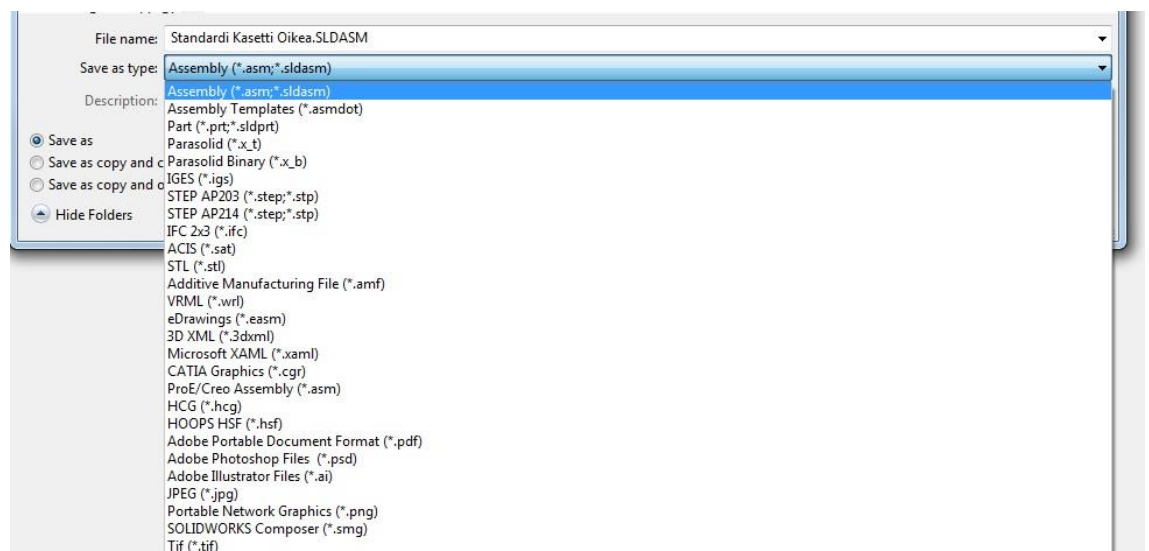
Koko työ jakaantui karkeasti neljään työvaiheeseen: CAD-mallien valmistelu (1), rigging ja animointi (2), hienosäätö (3) ja renderöinti (4). Käytännössä tuotantolinjasto jaettiin kolmeen osaan, joiden työvaiheet 1 ja 2 suoritettiin yksilöllisesti. Työvaiheet 3 ja 4 toteutettiin kerralla koko linjaston kohdalta.

Työnkulussa tuotantolinjaston kolmesta osasta puhutaan nimellä ”kelankäsittely”, ”lävistysyksikkö” ja ”CTW-asema”. Linjaston läpi kulkevasta metallilevystä puhutaan nimellä ”raina”.

4.1 CAD-mallien valmistelu

Tuotantolinjaston 3D-mallit oli toteutettu ohjelmalla SolidWorks 3D CAD, ja animaatio sekä visualisointi toteutettiin ohjelmalla Autodesk 3ds Max. Mallit täytyi siis saada siirrettyä ohjelmasta toiseen. Tästä eteenpäin SolidWorks -ohjelmasta puhutaan nimellä SW.

SW:ssä tuotantolinjastoja käsiteltiin eri mallien kokoonpanoina (assembly, .sldasm), eli kaikki linjaston osat olivat omissa työtiedostoissaan malleina, jotka esitettiin yhtenä kokoonpanona .sldasm -tiedostossa. Tällaista tiedostoa Maxiin ei sellaisenaan pystynyt viemään (export), joten täytyi tutkia Maxin hyväksymät tiedostotyytit tuontiin (import) sekä SW:n mahdollisuudet näitä tuottaa (kuva 15).



Kuva 15. Mahdolliset tallennusmuodot SolidWorksissa

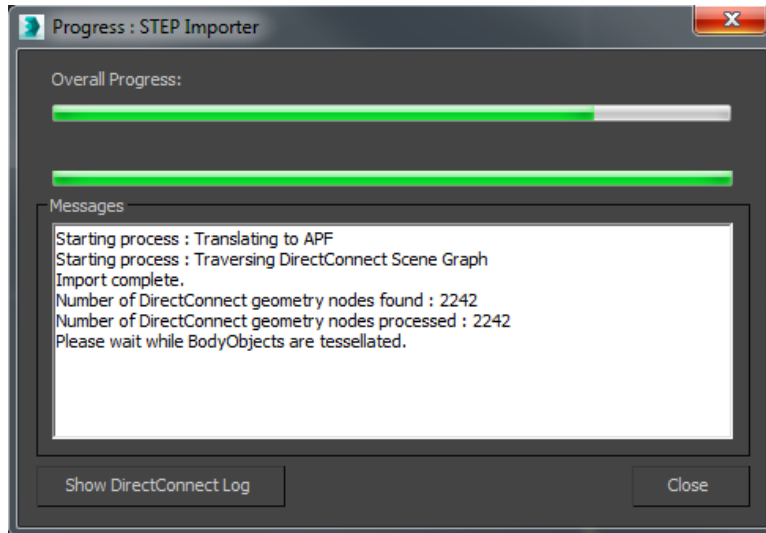
Taulukossa 1 on vertailtu ohjelmien välistä yhteensopivuutta mallien siirtoa koskien. Taulukossa esiintyy ohjelman kaikki mahdolliset 3D-grafiikan vientimahdollisuudet ja verrataan, onko tuonti 3ds Maxiin mahdollinen.

Taulukko 1. Tiedostotyyppien vienti- ja tuontimahdollisuudet

Tiedostotyyppi	Yhteensopivuus	Mallityyppi	Lopputulos
Assemblage (.sldasm)	Kiistanalainen	Body Object	SW täytyi olla asennettuna ja käynnissä, jotta siirto onnistui.
Part (.prt, sldprt)	Kiistanalainen	Body Object	SW täytyi olla asennettuna ja käynnissä, jotta siirto onnistui.
IGES (.igs)	Kyllä	Body Object	Toimi itsenäisesti, mutta suuri tiedostokoko.
STEP (.step, .stp)	Kyllä	Body Object	Toimi itsenäisesti, hyvä uudelleenkäyttö.
IFC 2x3 (.ifc)	Ei	-	Ei mahdollista tuoda.
ACIS (.sat)	Kyllä	Body Object	Erilliset asetukset mallin tuontiin ja selkeämpi hierarkia, mutta ei tunnisteita.
STL (.stl)	Kiistanalainen	Editable Mesh	Mallin osien automaattinen ”yhteenhitsaus” (Welding), joka yhdisti kaikki osat yhdeksi Editable Mesh :ksi.
VRML (.wrl)	Kyllä	Editable Mesh	Ei tunnisteita, mutta kevyt tiedosto.
CATIA Graphics (.cgr)	Kiistanalainen	Editable Mesh	Ei tunnisteita, jokainen mallin pinta omana objektinaan.
3D XML (.3dxml)	Ei	-	Ei mahdollista tuoda.
HOOPS HSF (.hsf)	Ei	-	Ei mahdollista tuoda.

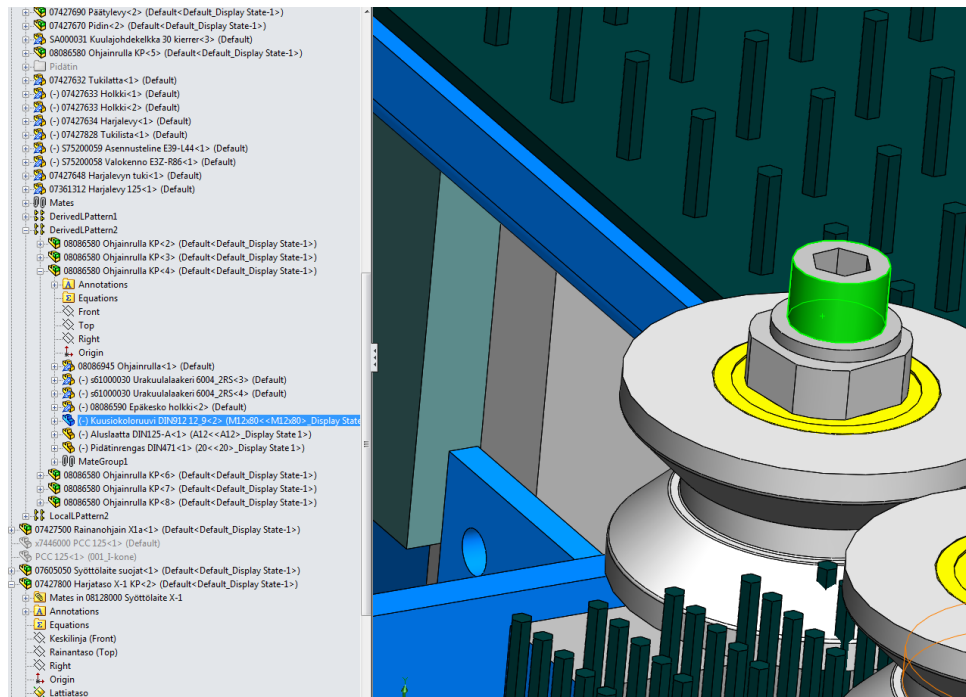
Ratkaisuksi päädyttiin tallentamaan ensiksi kokoonpanotiedosto (.sldasm) Part-tiedostoksi (.prt, .sldprt), jonka seurauksena saatiin itsenäisesti toimiva tiedosto 3D-geometriasta. Tämän tiedoston pystyi siis siirtämään sellaisenaan kansioden välillä, ja se säilytti tiedot geometriasta. Part-tiedoston luonnissa oli myös mahdollisuus valita, viedäänkö geometriasta pelkästään ulkoiset pinnat (Exterior Faces) vai otetaanko huomioon myös jokaisen objektin sisäkkäiset pinnat (Exterior Components). Vaihtoehdolla Exterior Faces lopputuloksena oli kevyempi malli, joka sisälsi vain esittelylle olennaiset pinnat.

Part-tiedostosta kuitenkin päädyttiin vielä viemään geometria STEP -muotoon, jolloin saatiin ilman SW:tä toimiva tiedostotyyppi, sekä saatiin automaattisesti suppressoitua malleja, tekemällä näistä kevyempiä. STEP on lyhenne termistä ”Standard for the Exchange of Product Data” ja se on tiedonsiirtoformaatti jota monet eri 3D-ohjelmat ymmärtävät, minkä seurauksena tämä valittiin lopulliseksi tuontiformaatiksi. Tämä siirtoprosessi suoritettiin automaattisesti 3ds Maxin omalla STEP Importer -lisäosalla (kuva 16).



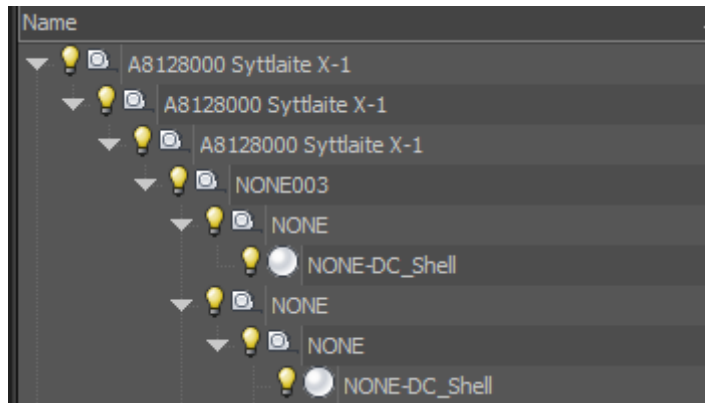
Kuva 16. STEP Importer

Mallit siirtyvät 3ds Maxiin joko Body Object tai Editable Mesh – muodossa. SW:ssä malleja käsitellään yleensä ”Bodyinä” eli jokainen alkuperäisen kokoonpanotiedoston malli on oma kokoonpanonsa, joka koostuu useista tilavuusmalleista. Malleja ei fyysisesti nimetä vaan jokaisella Bodyllä on oma tunnisteensa, ja näitä kasataan päällekkäin niin pienistä osista kuin muttereista ja ruuveista lopullisen tuotantolinjaston Bodyn alle (kuva 17).



Kuva 17. Body-kokonaisuus SolidWorksissä

3ds Maxiin tuotaessa nämä Bodyt muuttuvat Point Helpereiksi, jotka ovat yksi ohjelman monista apuobjekteista. Tämän seurauksena itse 3D-mallit ilmentyvät SW:n kokoonpanohierarkian mukaan jopa 10+ Point Helperin alle (kuva 18). Tämä ominaisuus heikensi huomattavasti 3ds Maxin työtotehokkuutta alussa, kun hierarkiaa ei pystynyt hyödyntämään samalla, koulussa opitulla tavalla.



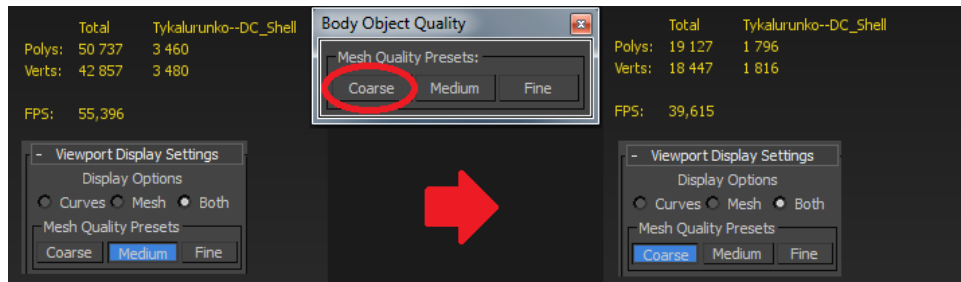
Kuva 18. Syöttölaitteen hierarkia Maxiin tuotaessa

4.1.1 Mallien jälkioptimointi

Suurin osa malleista joita käytettiin, oli arkkitehtimalleja, eli näiden mallien perusteella oli rakennettu fyysiset linjastot. Tämä johti siihen, että myös 3ds Maxiin tuotaessa kaikki aiemmin mainitut mutterit ja ruuvit siirtyivät mukana. Tämä vuorostaan vaikutti huomattavasti tiedostokokoon ja tietokoneelta vaadittuun suoritustehoon, joka lopulta paisui liian suureksi saatavilla olevaan suorituskäyttöön nähden. Malleja siirryttiin keventämään eli piilottamaan/poistamaan Max -työtiedoista turhat, silmille piilossa olevat objektit. Piilottamalla objekteja saatiin tarvittavaa suoritustehoa työstölle pienemmäksi, poistamalla saatiin pienennettyä tiedostokokoa. Näillä toimenpiteillä saatiin mallit sopivaksi työstölle.

Vaikka turhien objektien poisto tai piilotus oli suoritettu, polygonimäärä säilyi vieläkin turhan suurena. Ohjelmaan tuoduissa Body Objecteissa on mahdollisuus valita objektien pintojen laatu, Mesh Quality Preset, joka automaattisesti muuttaa objektien polygonimäärän karkeaksi (Coarse), keskiverroksi (Medium) tai hienoksi (Fine). Coarse-valinnalla tämä toimenpide pääosin karsii polygoneja kaarevista pinnoista, joita näiden mallien kohdalta oli paljon (mutterit, pultit, reiät, jouset). Näin pienten osien mesh-pintojen optimoinnilla siis pystyttiin saavuttamaan jopa 50 % vähempi polygonimäärä, ilman että silmä pystyi sitä tunnistamaan katseletäisytydeltä.

Ongelmana kuitenkin oli se, että jokaiselle objektille olisi täytynyt suorittaa tämä toimenpide erikseen, ja objekteja oli reilu 10 000. Tähän tarkoitukseen oli kuitenkin kehitetty monenlaista scriptiä, jonka avulla pystyttiin suorittamaan sama toimenpide kaikille valituille objekteille. Liitteessä 1 esiintyy työssä käytetty scripti, joka tuotti käyttäjälle ikkunan, josta pystyi valitsemaan valinnalle halutun pintojen laadun. Kuvassa 19 esitetään yksinkertaisen työkalukasetin muunnos Coarse-laatuiseksi malliksi, jossa valintana on yksi sen osanen.



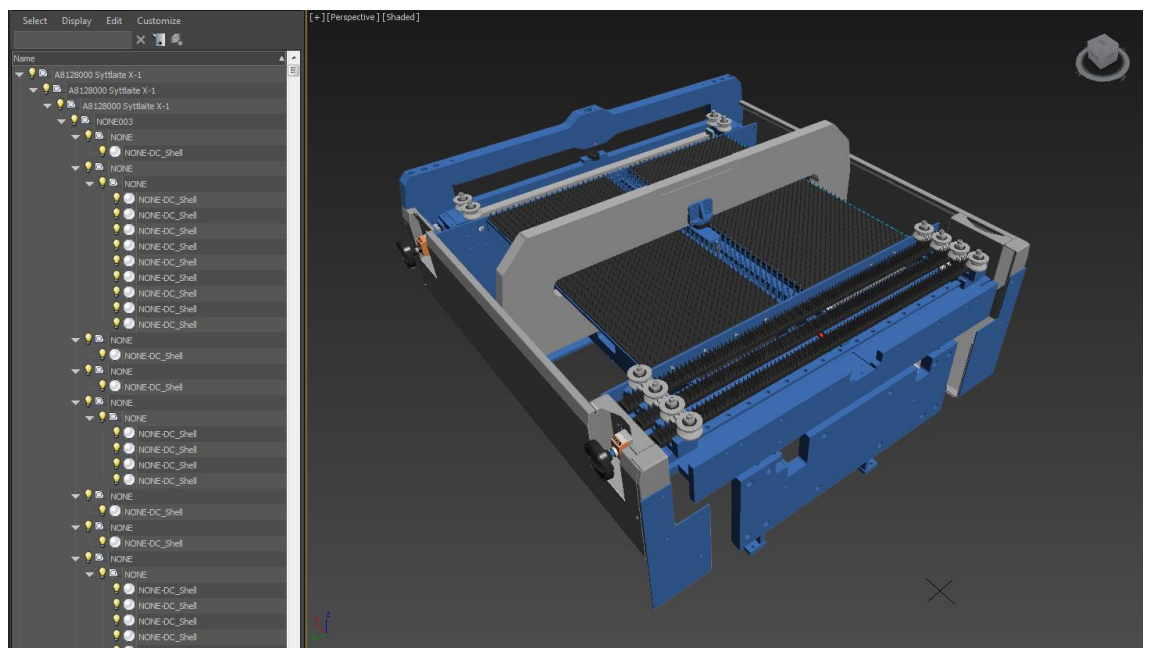
Kuva 19. Mesh Quality

Kuvasta huomaa hyvin, kuinka mallin sekä valinnan kokonaismääräinen polygoni sekä vertex -lukumäärät pienentyivät yli 50 %. Otettaessa huomioon, että kuvassa esiintyvä testi on otos yhden työkalukasetin optimoinnista, joka oli pieni osa linjaston kolmesta yksiköstä. Nämä toimenpiteet suoritettua kaikille ohjelmaan tuotaville malleille saavutettiin siis noin 50 % kevyemmät mallit, jotka vaikuttivat työstön tehokkuuteen, renderöinti-aikaan sekä tiedostokokoon.

4.2 Animaation esivalmistelut

Ennen varsinaista riggaus-/animointityötä linjastosta täytyi saada selville sen toiminta sekä mitä halutaan näyttää ja korostaa. Tämä tapahtui selvitystyönä alan asiantuntijoiden, yrityksen kotisivuilla olevien videoiden sekä julkisuudesta piilossa olevan materiaalin avulla. Osa linjaston toiminnasta on kilpaileviin yrityksiin verrattuna ainutlaatuista, ja yrityksen valtiona sen toimintaa ei opinnäytetyössä julkaistu.

Linjaston laajuuden vuoksi työ jaettiin kolmeen osaan, joihin tuotiin kaikki tarvittavat kokoonpanot. Nämä kokoonpanot olivat siis erillisten Point Helperien alla 3ds Maxiin tuodessa ja näin ollen Point Helperit toimivat hierarkiassa parent-objekteina, itse geometriset mallit child-objekteina. Kuvassa 20 esiintyy lävistysyksikön syöttölaitteen hierarkia ja sen geometria.



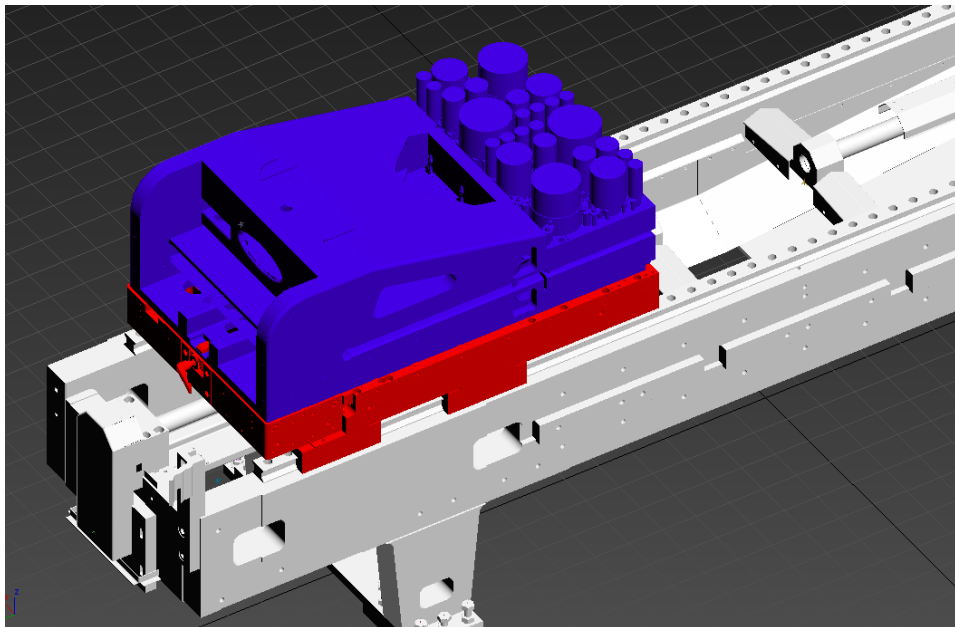
Kuva 20. Syöttölaite ja sen hierarkia

Kun yksi scene koostui monesta eri kokoonpanosta, mallit täytyi siirtää oikeille paikoilleen kuten palapeliä kootessa. Automaattiset kokoonpanojen hierarkiat tekivät tämän helpoksi, mutta sisäkkäisten hierarkioiden luonti oli myös tarpeen jotta saatiin toiminta animoitua.

4.2.1 Linkittäminen

Linjaston kaikki kolme osaa hyödynsivät eri linkitystekniikoita, kuten hierarkkinen parent-child -tekniikka ja Named Selection Set -tekniikka. Käytetään esimerkkinä linkityksen tarpeille kokoonpanoja lävistysyksikön työkalukasetit, jotka vaativat eniten toimenpiteitä.

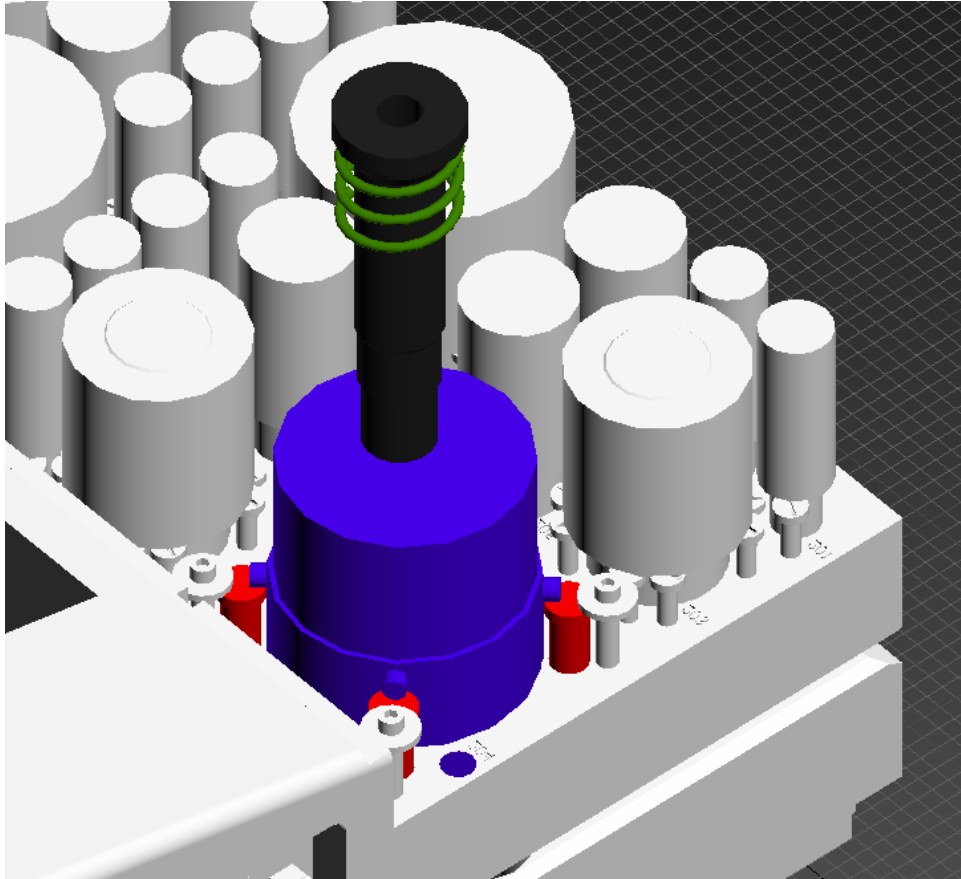
Työkalukasetit olivat siis omia kokoonpanojansa, joiden geometria 3ds Maxissa esiintyivät yhden kokonaisuutta nimeävän Point Helperin alla parent-child -tekniikalla. Työkalukasetti todellisuudessa liikkuu kuljettamismekanismin avulla, jonka linkitystä kasettiin ei kuitenkaan suoritettu samalla tekniikalla. Syynä tähän oli se, että jos kiskoilla liikkuvat kuljetusmekanismin osat sekä työkalukasetin alusta (pun.) linkitettäisiin työkalukasettiin (sin.), ei jatkossa mahdollisesti halutun kasetin vaihtoanimaation toteutus olisi käynyt niin helposti kuin mahdollista (kuva 21). Ratkaisuna päädyttiin käyttämään siis Named Selection Setiä, joka luotiin valitsemalla työkalukasetin Point Helper sekä kuljetusmekanismiin sisältyvät osat ja tekemällä näistä nimetty valinta, jota käytettiin animoinnissa.



Kuva 21. Sisuskalujen linkitys

Työkalukasetit kattoivat myös suurimman osan animaatiosta, ja siinä käytettiin linkitystä kuvaamassa toimintaa, joka oli yksinkertaisella parent-child -suhteella hoidettavissa. Kuvassa 22 valitsijaan (must.) on linkitetty sen palautusjouset (vihr.), sekä työkalumuotteihin (esimerkkinä yksi isoista muoteista, sin.) on linkitetty sen palautusjouset (pun.).

Vaikka kuvassa esiintyvä valitsija liikutti animaatiossa eri työkalumuotteja, tätä ei kuitenkaan voinut linkittää näihin, sillä työkalumuotteja liikutettiin yksi kerrallaan.

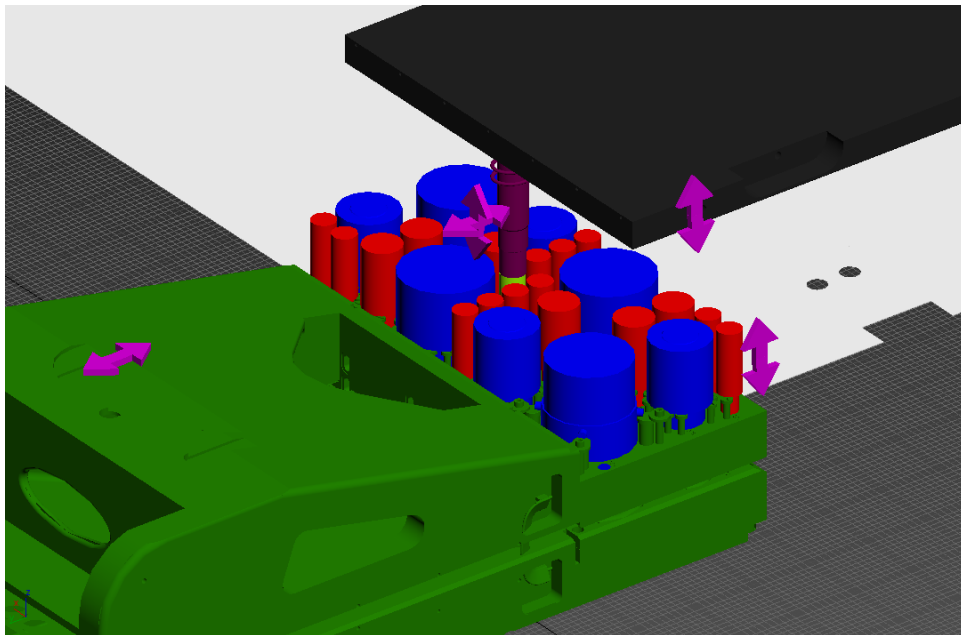


Kuva 22. Työkalukasetin linkitykset

4.3 Toiminnallisuuden animointi

Kaikki linjaston kolme osaa animoitiin samalla Keyframe-tekniikalla, ja tämä luku käsittelee tekniikkaa lävistämisen animaation kohdalta. Tämän tyyppinen teollisuuden animaatio jälkepäin todettuna vaati ainoastaan move-transformaation käyttöä, vaikka toteutus olisi ollut mahdollista tehdä monipuolisemmilla toimilla. Yksinkertainen Keyframe-toteutustapa oli ajallisesti tehokkainta, ja vaati ainoastaan hyvin tehtyä hierarkian ja linkityksen tarvetta. Animaation kuvanopeus toteutettiin PAL-standardin mukaan (25 fps).

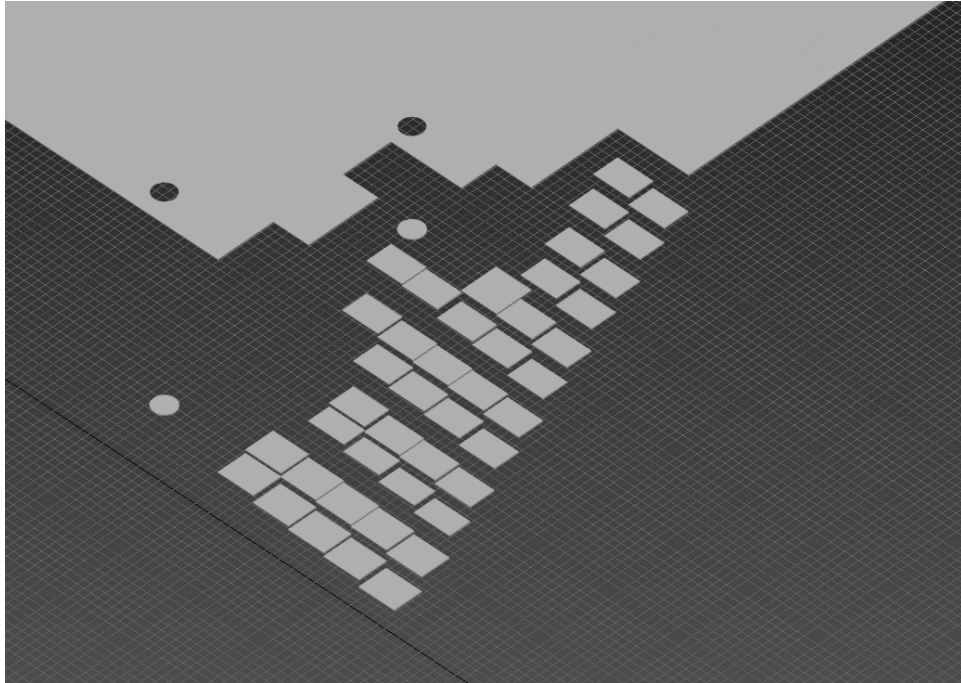
Tässä osassa linjastoa raina saapui siis aluksi syöttölaitteelle, joka siirsi rainaa tarpeen mukaan lävistystä varten. Rainan siirtyessä lävistysyksiköön siihen alettiin tuottaa erityyppisiä lävistyksiä (kuva 23), riippuen mikä työkalu muottien alla sijaitsi. Suuremmat työkalumuotit (sin.) sisälsivät työstöä varten kulmikkaat työkalut ja pienemmät muotit (pun.) pyöreät työkalut. Lävistysprosessi tapahtui neljän eri toiminnon seurauksena: puskinlevy (must.) painoi alas valitsijaa (violet.), joka vuorostaan painoi valittua muottia (kelt.) alaspäin, jonka työkalu teki lävistyksen rainaan kasetin (vihr.) nykyisen sijainnin perusteella.



Kuva 23. Lävistysprosessi

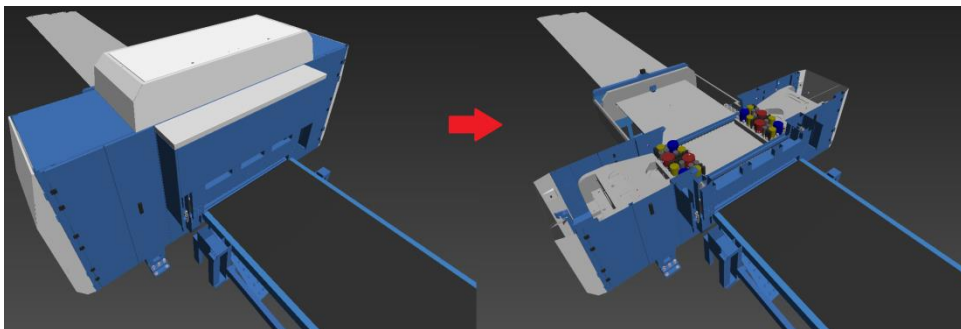
Animaatio tapahtui ensiksi siirtämällä puskinlevy vakioyksikön verran alaspäin niin että se kosketti valitsijaa, toiseksi siirtämällä nämä yhdessä muotin ylle ja kolmanneksi siirtämällä muotti näiden lisäksi alaspäin tehdäkseen lävistyksen. Samat toimenpiteet suoritettiin peilikuvana alkuperäisen tilanteen palauttamiseksi, ja siirryttiin seuraavan lävistyksen tekemiseen. Lävistyksen jälkeistä, kokonaisesta rainasta leikattua rainapalaa kutsutaan raportissa tästä eteenpäin kappaleena.

Rainan lävistys ei tapahtunut todellisuudessa, vaan kokonaisen rainan sisällä oli valmiiksi lävistysprosessin seurauksena syntyvä kappale, jossa lävistyksen hukkapalat olivat saumattomasti esitettyinä kunnes ne prosessissa irtoaisivat. Kuva 24 havainnollistaa prosessissa käytetyn silmäkääntötempun, joka yksinkertaisesti oli ajankäytöllisesti tehokkaampaa kuin lävistyksen todellinen tekeminen.



Kuva 24. Lävistysprosessin silmäkääntötempu

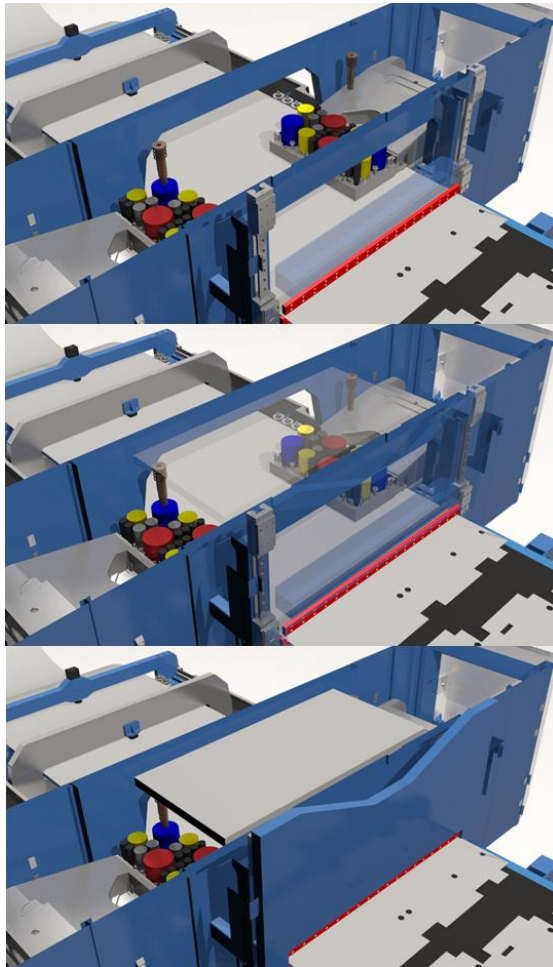
Yksikön toimintaa pyrittiin aluksi esittämään kameran sijaintia ja kuvakulmaa vaihtamalla, sillä suurin osa toiminnasta tapahtui sen ”sisällä”. Lopullinen, artistinen ratkaisu oli kuitenkin mallin yläosan häivytyksen animoitu Invisibility Track-käyrällä, jolla saatiin sisusta näytettyä helposti ilman monimutkaista kameran liikettä (kuva 25).



Kuva 25. Yläosa näkyvässä (vas.) ja yläosa täysin häivytyttyä (oik.)

4.3.1 Objektin läpinäkyvyyden animointi

Animoimalla häivytyks saatiin sisusta näkyviin, mutta olennaiset osat kuten työkaluvalitsijan puskinlevyn ja leikkurin toiminta haluttiin mukaan animaatioon. Ongelmana oli kuitenkin puskinlevyn ja leikkurin suuret koot, jotka peittivät alleen paljon olennaista toimintaa. Aluksi ratkaisuna tähän kokeiltiin kuvakulmien vaihtoa, mutta myöhemmin päädyttiin artistisesti näyttävältä näkyvyyden animoinnilta (kuva 26). Objekteihin lisättiin visibility trackit ja animoitiin toiminta näkyvän vain silloin, kun se oli olennaista. Esim. leikkuri oli siis näkyvässä vain, kun leikkaus suoritettiin.



Kuva 26. Visibility 0 % (ylh.), 20 % (kesk.) ja 100 % (alh.)

Kuitenkaan tämäkään ratkaisu ei ollut täysin sitä, mitä haluttiin. Kaukaa katsottuna puskinlevy ja leikkuri eivät näkyneet lähes ollenkaan ja läheltäkin ne ”sulautuivat” ympäristöön. Näkyvyys oli hyvä lähtökohta, ja haluttu tekninen ratkaisu löytyi lopulta materiaalien puolelta. Tämä ratkaisu käydään läpi luvussa 4.6.1.

4.4 Kamera-animointi

Alkunperin kamera-animaatioksi toteutettiin ”toimintaa seuraava kamera”, eli katseluetäisyys vaihteli sen mukaan, mitä toimintaa oli käynnissä ja mitä haluttiin näyttää. Tämä kamera-animointi suoritettiin siis ensimmäisenä vuorossa olleen linjasto-osan eli lävistysyksikön kohdalta ainoastaan. Huonona puolena tässä tavassa tuli vastaan se, että monen eri laitteen samanaikainen liike ei näkynyt ja huomio saattoi keskittyä väärään kohtaan laitteessa.

Kun yksikköä laajennettiin ja sitä alettiin tarkastelemaan kokonaisuutena, todettiin että parhaiten kaikki toiminta saadaan esitettyä yhdellä staattisella Target Cameralla, joka näyttää siis koko linjaston osasen samasta kuvakulmasta animaatioissa. Näin katse kiinnityi vain siihen mikä liikkui, ja kuvasta pystyi alusta alkaen erottamaan, mikä osa linjastosta oli esitettynä.

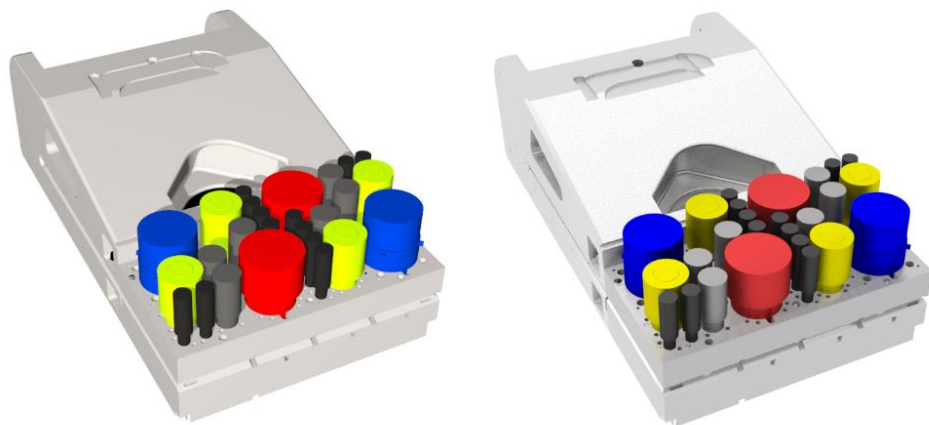
4.5 Hienosäätö

Vaikka animaatio oli tehty ja siitä selvisi hyvin linjaston käyttötarkoitus ja toiminta, ei se ollut todellisen eikä näyttävän näköinen. Työvaihe, jolla malli ja ympäristö tehdään vaikuttavan näköiseksi, on 3D-visualisoijan yleensä eniten aikaa vievä vaihe. Monet eri elementit kuten ihmisen iho tai eri metallityypit on koneellisesti hankala toteuttaa realistisen näköiseksi, ja ne onnistuvat vain oikeiden materiaalien, valaistusten ja mahdollisten lisätoimien seurauksena. Tässä työssä onneksi lopputulos ei tarvinnut olla täysin fotorealistinen, vaan näyttävä lopputulos oli päämääränä. Tarkoitus kun oli käyttää videota myynnin apuvälineenä.

Ulkonäöllisesti haluttuun pisteeseen päästiin työskentelemällä materiaalien, valaistusten ja linjastoon kiinnitettävien tarrojen (bitmap) kanssa.

4.5.1 Materiaaliparametrit ja teksturointi

Työstön alussa käytettiin ainoastaan Standard-materiaaleja oletusarvoisella Blinn-shadereilla, joilla värjättiin malli hahmottelemaan todellisuutta. Standard-materiaalin myötä käytettiin Scanline-renderöintimoottoria, jolla animaation suuntaa oli helppo hahmotella nopeiden renderöintien ansiosta. Kun suuntaa antava animaatio oli toteutettu ja linjaston esittelyn prioriteetit laitettu järjestykseen, vaihdettiin renderöintimoottoriksi Mental Ray sekä materiaaleiksi Arch & Design. Näiden materiaalien renderöidyt lopputulokset on havainnollistettu kuvassa 27. Kuvanlaatuun vaikuttaa myös käytettävä valaisu ja render-algoritmit, tämä kuva on ainoastaan havainnollistamassa valon käyttäytymistä eri algoritmeilla.



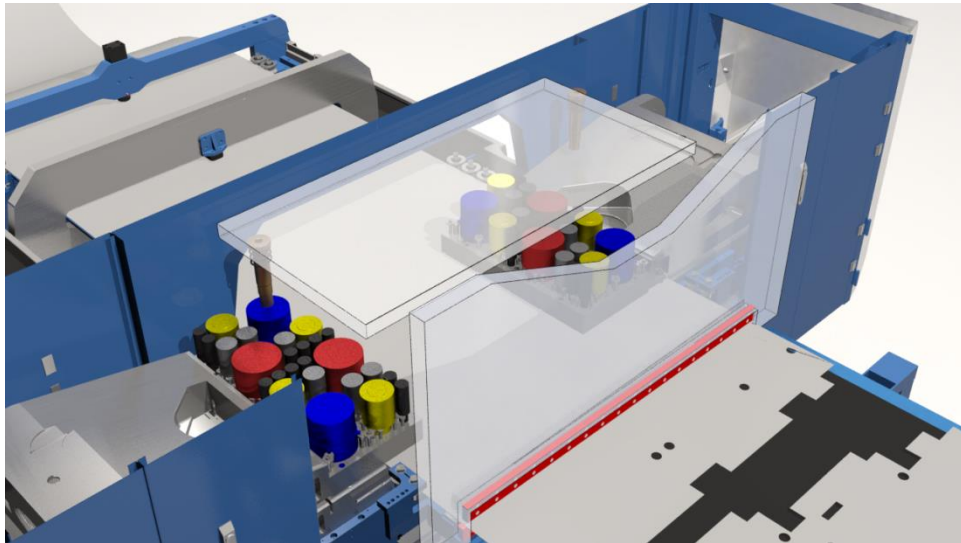
Kuva 27. Scanline (vas.) ja Mental Ray (oik.)

Ulkonäöllisesti tavoite oli tasapainoinen näyttävän ja fotorealistisen välillä, mikä lopulta saavutettiin oikean valaistuksen ja materiaalien yhdistelmän avulla.

Aiemmin todetun ongelman, eli puskinlevyn ja leikkurin toiminnan peittäminen, ratkaisu animoimalla näkyvyyttä ei tuottanut haluttua lopputulosta. Tälle ongelmalle ratkaisu löytyi materiaaleista.

Näille kahdelle objektille siis luotiin kopiot alkuperäisistä materiaaleista, jotka yhdistettiin Ink 'n Paint -materiaalin (Paint -ominaisuus pois käytöstä) kanssa yhteen Composite -materiaaliin. Näkyvyys asetettiin pysyvästi 20-30 % välille ja Ink 'n Paintilla sille luotiin selvästi erottuvat ulkoreunat. Näin objekti ei täysin peittänyt alleen oleellista toimintaa ja se silti erottui selvästi ympäristöstä.

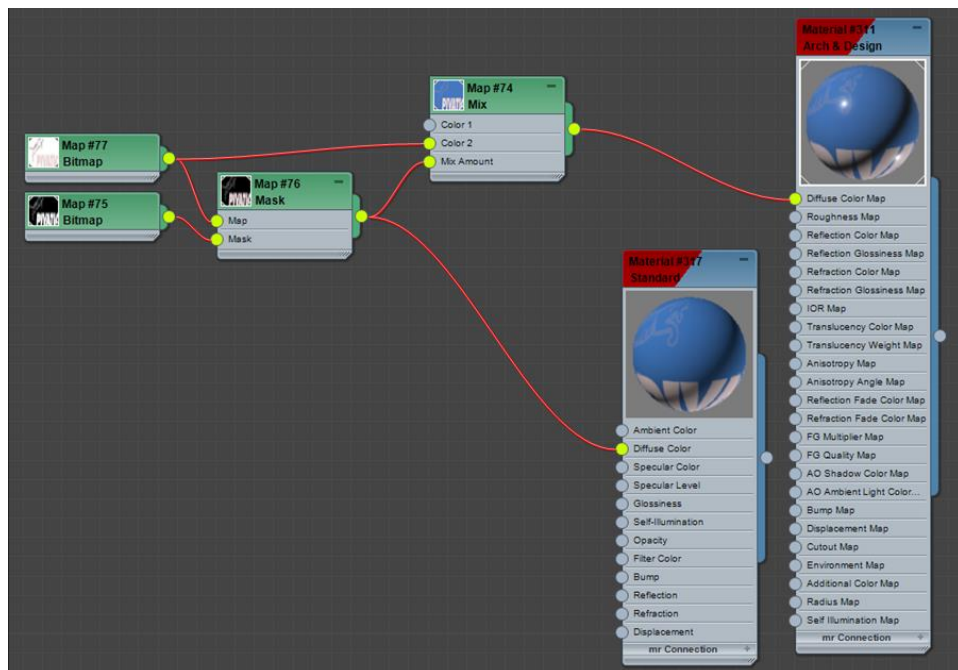
Lisätoimenpiteenä näitä kahta osaa liikuttavat, turhat objektit häivytettiin ympäriltä ja animoitu toiminta esitettiin artistisesti ilmassa leijuvilla objekteilla (kuva 28).



Kuva 28. Puskinlevyssä ja leikkurissa Composite: Arch & Design sekä Ink 'n Paint

4.5.2 UV Mapping

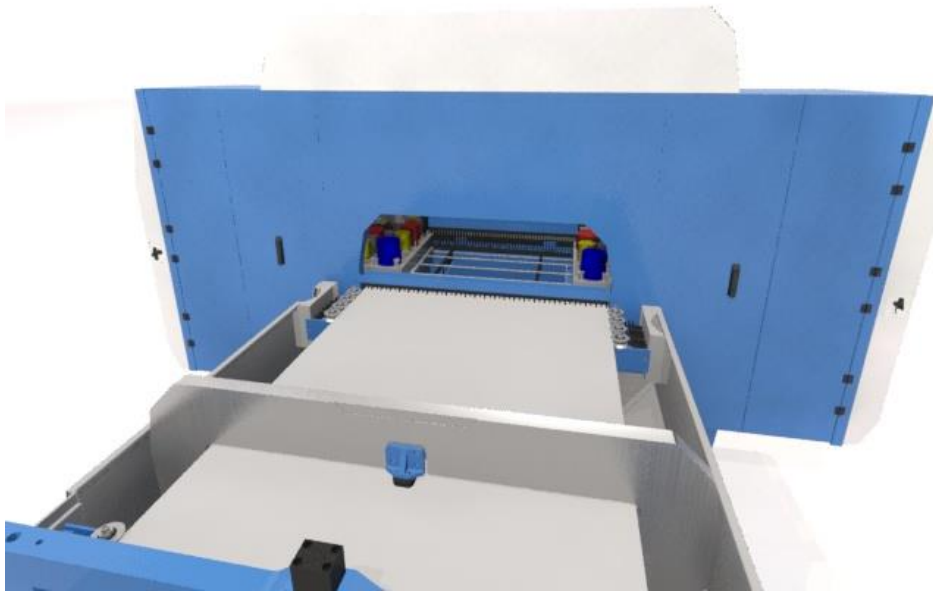
Työssä UV-mappausta tarvitsi ainoastaan yhdeltä osalta: yrityksen logon lisääminen muutamaan objektiin. Sitä alettiin toteuttaa valmiiden png-kuvien avulla scenessä sijaitseviin A&D -materiaaleihin, mikä ei kuitenkaan onnistunut kuten suunniteltu. Bitmapit saatiin objekteissa asettumaan UVW Mappauksella oikeaan kohtaan, mutta itse logo näkyi häivyttetyinä tai ei ollenkaan objektissa. Ongelmaksi selvytti A&D -materiaalin käyttäytyminen alpha-kanavan käytön kanssa, jolloin materiaalin taustaväri ei pääse näkymään alpha-kanavan läpi objektin diffuse-värissä. Tutkimuksen ansiosta selvisi, että oikeannäköiseen lopputulokseen päästään käyttämällä Mask-mapin ja materiaalin välissä Mix-mappia, asettamalla materiaalin värin ja bitmapin Color 1 ja 2 -parametreihin, ja Mix amount-parametriin Mask-map. Yksinkertainen ja toimiva ratkaisu saatiin saavutettua myös Standard-materiaalilla koulussa opetetulla tavalla, joka on A&D -ratkaisun alapuolella kuvassa 29.



Kuva 29. Onnistuneet bitmapin lisäykset Standard ja A&D -materiaaleihin

4.5.3 Valaistus

Kun työn alussa oli tavoitteena saada suuntaa antava animaatio eikä ulkonäölliseen puoleen keskitytty, toimi valaistuksena 3-pistevalaisu toteutettuna Omni lighteilla, jotka siis ampuvat valosäteitä yhdestä pisteestä jokaiseen mahdolliseen suuntaan. Kuvassa 30 lävistysyksikön ensimmäiset materiaali- ja valaisutestit, jossa päävalona toimi siis suora pistevalo epäsuoran, luonnollisen valon sijaan.



Kuva 30. 3-pistevalaisu toteutettuna pelkillä Omni lighteilla

Hyvää ulkonäköä tavoitellessa päävaloksi vaihdettiin luonnollista valoa simuloiva Skylight, ja epäsuoran valon sekä varjojen laskemiseen käytettiin Skylight from IBL -laskentakaavaa, joka oli hyvin nopea luoda eikä vaatinut parametreihin kajoamista. Skylight from IBL myös tuotti hyvän näköisen ja realistisen varjostuksen, mutta oli koneellisesti hyvin raskasta laskettavaa, eikä näin ollen sopinut animaatioon.

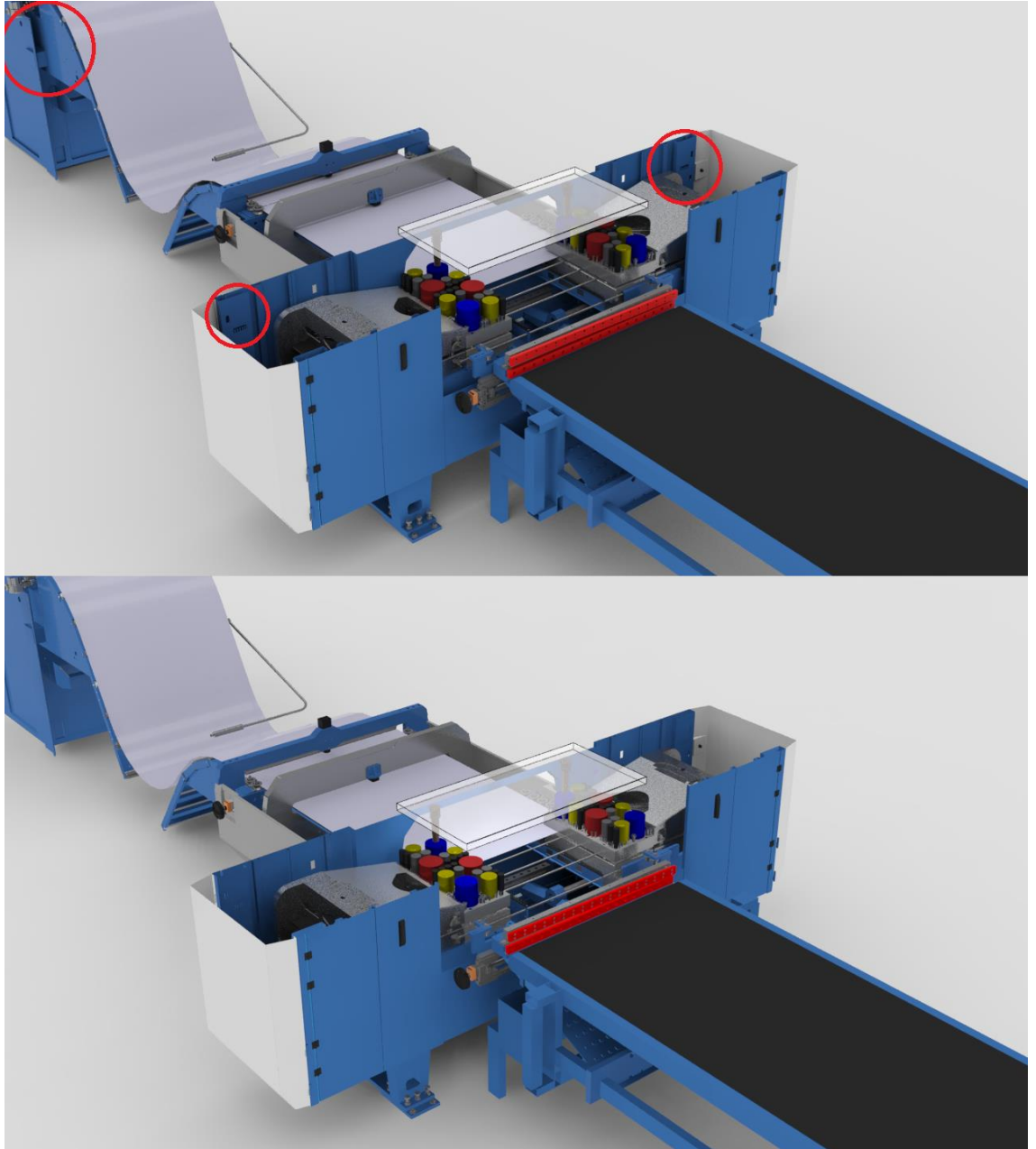
Epäsuoran valon laskemiseen vaihdettiin laskentakaavaksi Skylight from FG, ja otettiin käyttöön Final Gather ja Global Illumination. Tarkastellaan jälleen lopullista laatua linjaston keskipisteen, ja myös raskaimman kokonaisuuden eli lävistysyksikön valotusasetuksia ja niiden vaikutuksia renderöintiin animaation keskivaiheilla. Näitä asetuksia lähdettiin optimoimaan taulukon 2 mukaan. Tavoiteaika renderöinnille oli noin 1 minuutti/frame.

Taulukko 2. Valotustavan muuttaminen saman laadun säilyttäessä

Asetukset	Renderöinti-aika	Tilanne
Näytteenotto: 4min/16max Skylight from IBL, varjojen laatu: 2 Ei Final Gather Ei GI	17min 15sek	Kuva oli tarkka, varjostukset hyvännäköisiä, mutta renderöinti-aika kohtuuton.
Näytteenotto: 4min/16max Skylight from FG FG asetukset - Density: 0,1 - Rays per FG Point: 30 - Interpolate: 40 - Diffuse Bounces: 3 GI asetukset - Photons: 50 000 - Max Photons per Sample: 200 - Max Sampling Radius: 50cm	0min 55sek	Kuva oli tarkka, varjostukset hyvin lähellä aikaisempaa laatua, mutta varjostuksien yksityiskohdat saattoivat epäsuoran valon kimpoamisen seurauksena kadota/varjostuksia ei esiintynyt tarpeeksi.
Näytteenotto: 4min/16max Skylight from FG FG asetukset - Density: 0,5 - Rays per FG Point: 40 - Interpolate: 50 - Diffuse Bounces: 1 GI asetukset - Photons: 50 000 - Max Photons per Sample: 100 - Max Sampling Radius: 50cm	1min 05sek	Kuva oli tarkka ja varjostukset olivat hyvännäköisiä. Yksityiskohdat olivat suhteellisen hyvin havaittavissa, ja renderöinti-aika oli sopiva.

Valotustavan muuttamisella saatiin yli kymmenkertainen renderöintinopeus, ja vakioparametreilla ainoana haittana oli varjostuksen tarve silloin, kun sitä olisi haluttu. Tämä johtui siis epäsuoran valon kimpoamisesta, jolloin yksityiskohtia mallista saattoi valotuksen takia kadota. FG Densityä kasvattamalla varjoista saatiin vahvempia ja tummempia, mutta yksikön alapuolen varjot tulivat liian vahvoiksi. Rays per FG Point kasvattaminen korjasi tämän ilmiön.

Kuvassa 31 on verrattuna valotuksen toteutus alkuperäisellä Skylight from IBL -laskentatavalla (renderöintiajalla 17min 15sek), ja Skylight from FG -laskentatavalla (1min 05sek). Kuvasta huomaa, että lähes 17-kertaisen renderöintiajan aiheuttaneen valotuksen laskentakaavan lopputulos ei ollut sen arvoinen, kun lähes samaan lopputulokseen päästiin eri laskentakaavalla parametreja säätämällä. Ainoastaan kuvaan merkityt muutamat yksityiskohdat saattoivat hälventyä Final Gather -algoritmin seurauksena, mutta ne eivät olleet olennaisia toiminnan kannalta.

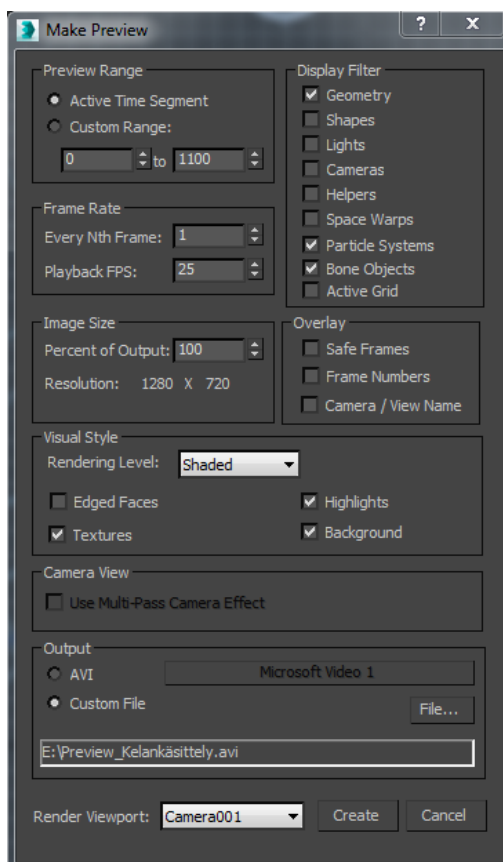


Kuva 31. Skylight from IBL (ylh.) ja Skylight from FG (alh.)

4.5.4 Renderöinti

Renderöinti suoritettiin kolmessa osassa, eli linjaston kaikki osat toteutettiin omana animaationaan. Tämä johtui siitä, että animaatioiden nopeus luotiin toimintojen todellisten nopeuksien perusteella, joka johti pitkiin (jopa yli 1000 framen) animaatioihin. Kun kokonaisuutena oli kuitenkin koko linjasto, nämä osat yhdistettiin jälkeenpäin videonkäsittelyohjelmassa.

Ennen videonkäsittelyyn sopivalla laadulla renderöimistä jokaisesta osasta luotiin Preview Renderöinti, jolla saatiin varmistettua että kaikki animaation elementit olivat oikein (kuva 32). Toisin sanoen varmistettiin, että scenestä löytyi kaikki tarvittavat objektit, animaatio oli jokaisen objektin kohdalta suoritettu ja se oli todellisen toiminnan nopeuden mukainen.



Kuva 32. Preview Render –ikkuna

Lopulliseksi laaduksi animaatiolle haluttiin HD-taso kuvasuhteen 16:9 mukaan, eli resoluutiolla 1280x720 pikseliä, ja lähtökohtana lävistysyksiön renderöintiajan optimoinnille oli seuraavanlainen:

- Polygoncount oli pienennetty scriptillä Coarse-laatuiseksi kaikissa Body Objecteissa
- Turhat, animaatioissa näkymättömät objektit oli poistettu scenestä, saavuttaen objektien lukumäärän 4882, ja näiden polygonimäärän 2 217 940.

- Tavoiteaika renderöinnille per kuva oli noin 1 minuutti, jolloin yhden animaation renderöinnin voisi suorittaa muutamassa tunnissa renderfarmilla, tai alle vuorokaudessa yhdellä tietokoneella.

Suurin osa laadullisesta ulkonäöstä oli toteutettu materiaalien ja valaistuksen kautta, joten lopullisen renderöinnin laatuun vaikutti renderöintimootorin kannalta näytteenottotaajuus sekä pakkaustavan valinta, joka määräytyy tiedostotyyppin mukaan. Esikatselurenderöinteihin näytteenottotaajuus on yleensä min1/max4, ja lopullisiin renderöinteihin min4/max16. Tämä oli otettu huomioon jo valaistusparametreja säätäessä, joten siitä ei tarvinnut tinkiä.

Renderöinti suoritettiin kuvasarjana PNG-formaatissa, joka soveltui hyvin häviöttömänä pakkausformaattina. PNG-formaatin bittisyvyudeksi valittiin default-vaihtoehto RGB 48-bittiä, ja kuvalle suoritettiin automaattinen gammakorjaus 2,2 -arvolla, joka on myös vakiolta 3ds Maxissa päällä. Jos animaation kuvista olisi haluttu osa tulostukseen/painoon, ne olisi renderöity TIFF-formaatissa suuremmalla resoluutiolla ja DPI-arvolla erikseen.

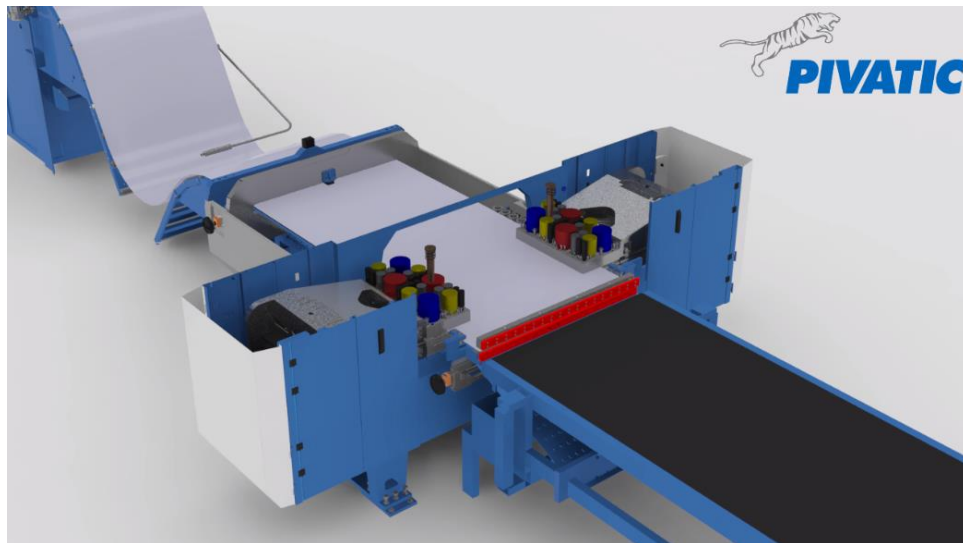
Lopullinen renderöinti toteutettiin hyödyntämällä koulusta löytyviä tietokoneita Network Rendering -tekniikalla, eli animaation muodostamat kuvat jaettiin renderöitäväksi koneiden kesken. Ohjelma, joka tämän työvaiheen suoritti, oli Autodesk Backburner, joka on 3ds Maxin lisäosana tuleva ohjelmisto.

4.6 Jälkitoimenpiteet

Animaatiot oli siis renderöity kuvasarjoiksi, joten vaadittu toimenpide seuraavaksi oli muunnos kuvasarjoista videotiedostoiksi. Tähän toimenpiteeseen kykenee monet videokäsittelyohjelmat, mutta tässä työssä käytettiin koululta löytyvää ohjelmaa Adobe Premiere Pro CS6. Lisäksi tähän mainostarkoitukseen menevään videotiedostoon haluttiin lisätä yrityksen logo.

4.6.1 Videonkäsittely

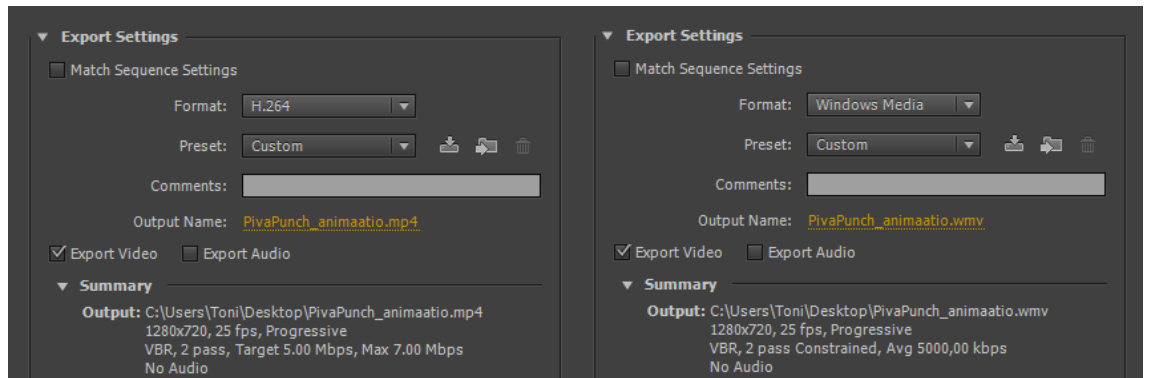
Kuvasarjat siis tuotiin ohjelmaan ja ne lisättiin aikajanelle, samoin tehtiin yrityksen logolle. Itse logo oli png-kuvatiedostona, ja se skaalattiin sopivaksi videon oikeaan yläkulmaan, näkyväksi koko esityksen ajan (kuva 33).



Kuva 33. Kuvakaappaus animaatiosta yrityksen logolla

Kolmesta osasta koostuvat kuvasarjat asetettiin peräjälkeen aikajanelle, ja vaihdos näiden välillä suoritettiin videonsiirtymisefektilä Dip to White, eli edellisen animaation loppu haalistui valkoiseksi ja uusi animaatio alkoi käänteisellä toimenpiteellä.

Musiikkia elokuvaan ei haluttu. Lopullinen tiedostomuoto luotiin kahteen eri formaattiin, sillä yrityksen käyttämä esitysohjelma hyödynsi parhaiten Windows Media-tiedostoa, kun nettisivuilla käytettiin tämän lisäksi mp4-videota. Videotiedoston pakkausalgoritmina käytettiin H.264-kodekkia, millä luotiin .mp4-tiedosto. Lisäksi luotiin valmiilla Windows Media-kodekilla pakattu video, mistä saatiin .wmv-tiedosto. Kummatkin tiedostomuodot luotiin käyttäen pohjana ohjelman valmista asetuskokoonpanoa HD 720p 25 (ei audio), muuttaen kohdebitrateksi 5 Mbps ja maximum bitrateksi 7 Mbps, kuva 34. Tämä on yleisesti sopiva ja myös YouTuben suosittelema bitrate HD-videolle, johon myyntiesittelyvideo lopullisessa muodossa muunmuuassa ladataan.



Kuva 34. Lopulliset videotiedostot. Vasemmalla .mp4-tiedostomuotoon tuleva, oikealla .wmv.

5 YHTEENVETO

Mallien siirtämiseen ohjelmasta toiseen on monta eri tapaa ja tiedostotyyppiä, mutta kannattaa perehtyä kunnolla siihen, miten malli käyttäytyy siirron jälkeen ja mitä sille ollaan tekemässä. Monet tiedostotyypit kuten taulukossa 1 esiintyvät STL tai CGR käyttäytyivät Maxiin tuodessa tavalla, millä animaation toteutus olisi hyvin vaikeaa tai jopa mahdotonta. Vastapainona tähän myös animaation toteutukseen soveltuvia tiedostotyyppejä oli monta, ja ohjelmasiirroksen jälkeen geometria pysyi usein samana, ainostaan hierarkian esitys ja linkitykset vaihtoivat muotoaan. Verratessa erityisesti tässä opinnäytetyössä käytettyjen ohjelmien yhteensopivuutta, tiedostotyypit STEP ja IGES olivat parhaita vaihtoehtoja, sillä ne toivat mallit ehjinä ohjelmaan ja säilyttivät objektien väliset hierarkiat. IGES-tiedostotyyppi kuitenkin loi hyvin suuren tiedostokoon, jolloin tällaisessa hyvin suuren polygoni-/objektimäärän omaavassa projektissa kevyempi STEP-tiedostotyyppi selviytyi voittajaksi.

Kokonaisuuden linkityksen tarve sekä animaatio olivat suuren työmäärän takana, mutta suurin osa näistä toimenpiteistä onnistui ilman monimutkaisempia työkaluja tai menetelmiä. Linkitystavan valinnassa otettiin huomioon, onko objekteja tarkoitus animoida uudelleen ja sisälsivätkö nämä ”ali-animaatioita”. Kaiken kaikkiaan animaatio ja objektien käyttäytymisen olivat hyvin yksinkertaisia, ja kun selvitystyö jokaisen työstövaiheen liikkuvista osista oli tehty, animaatio pystyttiin toteuttamaan helposti.

Visualisoinnissa ja lopullisessa renderöinnissä hyvää ulkonäköä lähdettiin aluksi hakemaan pääosin materiaalin ja renderöintiasetusten kautta, mutta ottaen huomioon linjaston kiiltävien ja heijastavien metalli- ja muovipintojen ulkonäön, valaistuksella oli hyvin suuri merkitys fotorealistsuutta ja näyttävää lopputulosta tavoitellessa.

LÄHTEET

Watt, A. 2000, 3D Computer Graphics (3rd Edition). Harlow, England: Pearson Education Limited.

Autodesk 3ds Max Help 2015

BeanBox Animation 2013. Viitattu 30.4.2015.

<http://www.beanboxanimation.co.uk/3d-modeling-for-animation-and-product-visualisation-projects/>

Beane, A. 2012, 3D Animation Essentials. Indianapolis, Indiana: John Wiley & Sons.

Chopine, A 2011, 3D Art Essentials – The Fundamentals of 3D Modeling and Animation. Oxford, United Kingdom: Elsevier Inc.

Techonomy 2012, Real or Rendered? by Alex Southern. Viitattu 27.4.2015.

<http://techonomy.com/2012/10/real-or-rendered-how-3d-imagery-is-changing-the-way-you-shop/>

Wikipedia 2015a, Video. Viitattu 29.4.2015.

<http://en.wikipedia.org/wiki/Video>

Wikipedia 2015b, 3D Computer Graphics. Viitattu 9.5.2015.

http://en.wikipedia.org/wiki/3D_computer_graphics

Wikipedia 2015c, Procedural Animation. Viitattu 10.5.2015.

http://en.wikipedia.org/wiki/Procedural_animation

Wikipedia 2015d, Rendering (computer graphics). Viitattu 28.4.2015.

http://en.wikipedia.org/wiki/Rendering_%28computer_graphics%29

Wikipedia 2015e, Image File Formats. Viitattu 10.5.2015.

http://en.wikipedia.org/wiki/Image_file_formats

Wikipedia 2015f, 3D Computer Graphics Software. Viitattu 1.5.2015.

http://en.wikipedia.org/wiki/3D_computer_graphics_software

MESH QUALITY -SCRIPT

```
try(destroyDialog ::bgaRoll)catch()
rollout bgaRoll "Body Object Quality"
(
    local presetProps = (#LowQualityVDS, #MediumQualityVDS, #HighQualityVDS, #IsoAngleDS, #IsoChordHeightVDS,
#FaceApproxAngleVDS, \
    #EdgeApproxAngleVDS, #FaceChordHeightVDS, #EdgeChordHeightVDS, #MaxEdgeLengthPctVDS)
    fn setMeshQuality objs mode: props:presetProps = if objs.count != 0 do
    (
        if GetCommandPanelTaskMode() != #create do SetCommandPanelTaskMode #create
        local vals = case mode of
        (
            #Coarse: #(on, off, off, 15, 0, 30, 0, 1, 0, 0)
            #Medium: #(off, on, off, 7, 0, 15, 0, .1, 0, 0)
            #Fine: #(off, off, on, 3, 0, 10, 0, .02, 0, 0)
        )
        with redraw off for body in objs where isKindOf body Body_Object do
        (
            for i = 1 to props.count do setProperty body presetProps[i] vals[i]
        )
    )
    group "Mesh Quality Presets:"
    (
        button btn_c "Coarse" pos:[8,25] width:60 height:20
        button btn_m "Medium" pos:[70,25] width:60 height:20
        button btn_f "Fine" pos:[132,25] width:60 height:20
    )
    on btn_c pressed do setMeshQuality selection mode:#Coarse
    on btn_m pressed do setMeshQuality selection mode:#Medium
    on btn_f pressed do setMeshQuality selection mode:#Fine
)
createDialog bgaRoll 200 55 style:(#style_titlebar, #style_sysmenu, #style_toolwindow)
```