

Toni Kontio

Ajoneuvon riggaus Mayassa

Metropolia Ammattikorkeakoulu
Medianomi
Viestinnän koulutusohjelma
Opinnäytetyö
7.5.2012

| | |
|--|--|
| Tekijä Otsikko | Toni Kontio Ajoneuvon riggaus Mayassa |
| Sivumäärä Aika | 50 sivua + 1 liite 7.5.2012 |
| Tutkinto | Medianomi |
| Koulutusohjelma | Viestinnän koulutusohjelma |
| Suuntautumisvaihtoehto | 3D-animointi ja -visualisointi |
| Ohjaaja | Jaro Lehtonen |
| <p>Opinnäytetyö keskittyy mekaanisten osien riggaukseen Mayassa. Riggaaminen on 3d-mallien valmistelemista animaatioon. Teoriavaiheessa keskitytään tutkimaan Autodesk Maya -ohjelman tarjoamiin riggausvälineisiin. Tarkoituksena ei ole käydä läpi Mayan kaikkia työkaluja, vaan keskittyä tutkimaan työkaluja joilla saa tehtyä mekaanisia esineitä, kuten ajoneuvoja. Vaikka ohjelmaksi on valittu Maya, niin osaa näistä esimerkeistä voi soveltaa myös muihin 3d-ohjelmiin, esim. Autodeskin 3ds Maxiin.</p> <p>Työvaiheessa rigataan tätä työtä varten mallinnettu helikopteri. Tavoitteena oli valita malli, jossa on paljon liikkuvia osia ja jonka esimerkeistä on helppo soveltaa myös muihin kulkuneuvoihin. Lopputuotteena on valmis 3d-malli, jonka toimivuutta esitellään animaatiolla.</p> | |
| Avainsanat | Opinnäytetyö, 3d, Maya, riggaus, ajoneuvo, helikopteri |

| | |
|---|--|
| Author Title | Toni Kontio Rigging vehicle in Maya |
| Number of pages Date | 50 pages + 1 appendices 7.5.2012 |
| Degree | Medianomi |
| Degree programme | Media |
| Specialisation option | 3D Animation and Visualization |
| Instructor | Jaro Lehtonen |
| <p>The Thesis concentrated on rigging mechanical objects in Maya, rigging understood as preparing 3d objects for animation. The theoretical focus lied on examining Autodesk Maya rigging tools. There was no purpose to examine every tool in Maya but to concentrate on those that could be best used for mechanical rigging, vehicles, for example. Because Maya was selected, some of the examples employed in the Thesis can also be used for other 3d programs, such as Autodesk 3ds Max, for example.</p> <p>The practical main implementation was rigging a helicopter that was modelled for the Thesis. The purpose was to select a model that would contain a lot of moving parts and whose modeled examples could be applied to other vehicles. The end product is a finished 3d model whose functionality is presented with an animation.</p> | |
| Keywords | Diploma work, 3d, Maya, rigging, vehicle, helicopter |

Sisällys

| | | |
|------|--|----|
| 1 | Johdanto | 3 |
| 2 | Käsitteitä | 4 |
| 3 | Riggaus yleisesti | 6 |
| 4 | Maya riggausvälineenä | 7 |
| 4.1 | Yleiskatsaus Mayaan | 7 |
| 4.2 | Nodet & Historia | 8 |
| 4.3 | Mayan käyttöliittymä | 9 |
| 4.4 | Projektiasetukset & Referointi | 11 |
| 4.5 | Objektien instanssointi | 11 |
| 4.6 | Parentointi ja ryhmät | 12 |
| 4.7 | Constraintit | 13 |
| 4.8 | Jointit, eli luut | 16 |
| 4.9 | IK-handle | 17 |
| 4.10 | Skinnaus | 17 |
| 4.11 | Deformerit | 18 |
| 4.12 | Connection Editor | 19 |
| 4.13 | Set Driven Key | 19 |
| 4.14 | Riggauksen huomioiminen mallintamisessa | 19 |
| 4.15 | Scriptaaminen Mayassa | 20 |
| 4.16 | Yleisiä vinkkejä riggaukseen Mayassa | 21 |
| 5 | Helikopteri | 23 |
| 5.1 | Helikopteri rigattavana mallina | 23 |
| 5.2 | Tuulilasinpyyhkimet | 24 |
| 5.3 | Kuomun aukaisumekanismi | 28 |
| 5.4 | Pääroottori | 30 |
| 5.5 | Takaroottori | 34 |
| 5.6 | Renkaat ja iskunvaimentimet | 35 |
| 5.7 | Aseistus | 39 |
| 5.8 | Riggauksen viimeistely | 47 |
| 6 | Yhteenveto ja pohdintaa | 49 |
| | Lähteet | 50 |
| | Liitteet | |
| | Liite1. Linkit rigin esittelyanimaatioon | |

1 Johdanto

Opinnäytetyössä keskitytään lähinnä mekaanisten osien riggaukseen Autodesk Maya -ohjelmalla ja oppia eri työmenetelmiä. Riggaaminen tarkoittaa 3d-mallien valmistamista animaatioon. Mekaanisissa esineissä, kuten ajoneuvoissa, tämä voisi tarkoittaa renkaiden tai muiden liikkuvien osien toimivaksi rakentamista. Riggauksen esimerkkejä voidaan soveltamalla käyttää myös muissakin 3d-ohjelmissa, esim. Autodeskin 3ds Maxissa.

Alun teoriavaiheessa käydään läpi Mayan keskeisiä työvälineitä joita tarvitaan mekaanisten osien riggaamisessa. Riggaustyövälineet käydään teoriavaiheessa yleisluontoisesti esitellen, mutta työvaiheessa näihin tutustutaan tarkemmin.

Rigattavaksi ajoneuvoksi on valittu tätä työtä varten mallinnettu helikopteri, jossa on mahdollisimman paljon liikkuvia osia. Opinnäytetyön esimerkit on kirjoitettu opasmaisesti, jotta niistä olisi apua aloitteleville riggaajille. Rigattuja esimerkkejä voidaan soveltaa muihinkin ajoneuvoihin tai muihin mekaanisiin laitteisiin. Vaikka valmis helikopterimalli sisältää myös materiaaleja, animaatiota ja rendauksia, niin tässä opinnäytetyössä niitä ei käsitellä. Valmis rigattu malli viimeistellään animaattorille ystävällisellä kontrollisysteemillä, jonka avulla animointi on mahdollisimman helppoa. Helikopterin riggaussysteemiä esitellään myös animaatiolla.

Vaikka lopputyö onkin kirjoittajan oma projekti, niin silti projektia pyritään peilaamaan työelämän vastaaviin. Näitä ovat esimerkiksi projektiasetukset, työn jakaminen ja 3d-mallin viimeistely animaattorille. Lopputyössä ollaan lähinnä riggaajan asemassa.

2 Käsitteitä

3d-visualisoinnin ja -animoinnin sanasto on suurin osa englanninkielisiä lainasanoja. Näiden sanojen ymmärtäminen on tärkeää tämän opinnäytetyön ymmärtämisen kannalta. Tässä kappaleessa perehdytään yleisimpiin 3d-maailmasta tuttuihin käsitteisiin. Itse työvaiheeseen liittyvää sanastoa käsitellään myöhemmin niiden omissa kappaleissa.

3d-grafiikka

3d-grafiikalla voidaan esittää kolmiulotteisia kuvia. 3d-grafiikan etuna perinteiseen kaksiulotteiseen grafiikkaan verrattuna on syvyysulottuvuus, jonka avulla esineitä voidaan esimerkiksi pyöritellä tilaulottuvuudessa. 3d-grafiikan tilaulottuvuudessa on kolme suuntaa, joita yleisimmin kutsutaan x-, y- ja z-akseleiksi. Akseleiden suunnat maailmassa on ohjelmakohtaisia. Autodesk Maya -ohjelmassa y-tarkoittaa oletuksena maailman pystyakselia, kun taas Autodesk 3ds Max -ohjelmassa se on z-akseli.

3d-mallintaminen

3d-grafiikka sisältää kolmiulotteisia objekteja, jotka täytyy mallintaa. Mallinnetut objektit koostuvat vertexeistä, eli pisteistä, joiden väliin muodostuu polygon-pinta. Voidaan kuvitella esim. laatikko, jonka jokaisessa kulmassa on verteksi; jokaista verteksiä yhdistää edge ja jokainen sivu on polygon. Edellä mainittua tapaa kutsutaan polygon-mallintamiseksi. Muitakin tapoja on, esim. nurbs-mallintaminen, joka liittyy enemmän vektoreihin. Tässä opinnäytetyössä on käytetty polygonmallintamista.

Riggaus

Riggaaminen on 3d-mallien valmistamista animaatioon. Riggaaminen tehdään ennen animointia lähes kaikille animoitaville hahmoille, ajoneuvoille, esineille, jne. Riggaamisella helpotetaan animointia. Riggaamisesta kerrotaan tarkemmin sen omassa kappaleessa.

Scene

Scene on 3d-mallinnusohjelman 3d-avaruus, näyttämö, jossa kaikki 3d-mallit, valaistus ja muut objektit yhdistellään yhdeksi kokonaisuudeksi. Scenejä voidaan yhdistellä toisiinsa. Näkymää, jossa sceneä tarkastellaan kutsutaan viewportiksi. Viewporteja voi olla useita, esim. Top, Right, Perspective. Mayassa jokaista viewportia kuvaa oma kameransa, jonka asetuksia voidaan muuttaa.

Animointi

Kaikki asiat, jotka kuvassa liikkuvat ovat jotenkin animoituja. Näitä ovat esim. hahmot, ajoneuvot, kamerat, valot jne.

(Key)frame

Animaatiossa aika määritellään usein frameilla. Jokainen frame vastaa yhtä kuvaa animaatiossa. Toistamalla frameja nopeasti peräkkäin, saadaan illuusio liikkeestä. Keyframe on animaatiossa määritelty frame, johon on tehty jotain tapahtumaa. Tämä tapahtuma voisi olla vaikka animaatiohahmon kävely, jossa jokainen jalan, käden, yms. ääriasento on tallennettu keyframeksi.

Attribuutti

3d-maailmassa attribuutti on jonkin asian tai esineen arvo tai ominaisuus. Esim. 3d-objektilla on useita attribuutteja, esim. sijainti (translate), suunta (rotate), koko (scale) ja näkyvyys (visibility). 3d-ohjelmissa käyttäjä voi muokata, animoida ja luoda uusia attribuutteja.

Deformointi

Objektien muokkauksesta animoinnin aikana käytetään sanaa deformointi. Deformointi muokkaa objektien pintarakennetta liikuttamalla esim. polygon-objektin vertexejä. Deformoimalla esineitä saadaan esim. ihmisen kasvot ilmehtimään tai käsi taipumaan.

Renderöinti

Renderöinti voidaan suomentaa kuvantamiseksi, mutta sana renderöinti on yleisesti käytössä. Renderöinnillä saadaan scenen halutut objektit näkyville monitorille. Reaaliaikaista renderöintiä tehdään jatkuvasti viewporteissa, mutta yleensä renderöinnillä tarkoitetaan kuvan laskemista, jossa on mukana valaistuksia, varjoja, shadereitä jne. Renderöity kuva tallennetaan erilliseksi kuvatiedostoksi.

Shader

Shaderit ovat tietokoneella laskettuja renderöintiefektejä. Mayassa shaderit ovat mm. erilaiset materiaalit, kuten esim. puu- ja metallipinnat. Näille shadereille voidaan laskea erikseen heijastuksia, korkokuvioita, läpinäkyvyyksiä jne. Kaikilla renderöitävillä objekteilla täytyy olla jokin shaderi, jotta se voi edes näkyä. Shadereitä löytyy Mayasta valmiilla asetuksilla tai niitä voi luoda itse.

3 Riggaus yleisesti

Riggauksella 3d-maailmassa tarkoitetaan 3d-mallin valmistelemista animointiin ja liikkeiden automatisointia. Otetaan esimerkiksi tätä lopputyötä varten mallinnettu helikopteri. Valmis 3d-malli ei kelpaa vielä animoitavaksi, koska olisi liian sekavaa ja vaikeaa animoida jokaista liikkuvaa palaa erikseen. Jotta helikopteria olisi miellyttävä animoida täytyy se ensin rigata. Riggaus rakennetaan objekteja yhdistämällä ja rakentamalla erilaisten apuobjektien avulla mahdollisimman yksinkertainen rigi, käyttöliittymä, jolla animointi on helppoa.

Riggaajan ja mallintajan täytyy tuntea rigattavan 3d-mallin toimintaperiaate, jotta sille voidaan tehdä uskottava ja helppokäyttöinen rigi. Jos esim. helikopterista haluaisi tehdä oikealla tavalla toimivan, täytyy jo mallinnusvaiheessa tutkia helikopterin käyttäytymistä. Ihmishahmon anatomian tutkimus riittävällä tasolla on tarpeen, jotta sille voi rakentaa rigin luustosta ja lihaksista. Riggauksen voi jakaa kolmeen seuraavaan vaiheeseen.

Ensimmäinen vaihe riggauksessa tarkoittaa objektien yhdistämistä ja niiden laittamista oikeaan hierarkiaan, arvojärjestykseen. Mayassa tämä tarkoittaa parentointia, josta lisää sen omassa kappaleessa. Tähän vaiheeseen kuuluu myös erilaisten apuobjektien lisääminen 3d-objekteihin, jotta animointia saadaan automatisoitua. Esim. helikopterin jousituksen tulisi toimia automaattisesti kun rengas koskettaa maahan.

Toinen vaihe on rakentaa mahdollisimman yksinkertainen kontrollijärjestelmä. Koskaan ei ole hyvä animoida suoraan luita tai animaatioissa näkyviä 3d-objekteja. Sen sijaan hyvä tapa on luoda animaattorille kontrolliojekteista rakennettu käyttöliittymä, jota hallitsemalla kaikki tarvittavat osat liikkuvat. Kaikki animaatio myös tallennetaan näihin kontrolliojekteihin.

Kolmas vaihe on riggauksen viimeistely. Tähän kuuluu riggauksen tarkistaminen testianimaatioilla. Kontrolliojektit freezataan, eli kaikki attribuutit nollataan, jotta animaattori voi asettaa niihin omia arvoja. Tähän kuuluu myös layereiden järjestäminen, jotta näkyvillä ovat vain ne objektit, jotka animaattorin tarvitsee nähdä. Lisäksi riggaajan tulee laittaa kaikki apuobjektit piiloon ja/tai lukkoon, jotta animaattori ei vahingossa tuhoa rigiä. Tarkoitus on, että animaattori voi liikutella vain kontrolliojekteja.

4 Maya riggausvälineenä

4.1 Yleiskatsaus Mayaan

Ensimmäinen versio Mayasta julkaistiin vuonna 1998 Alias Systems Corporation toimesta. Vuonna 2005 Autodesk osti Mayan ja uusi omistaja nimesi ohjelman Autodesk Mayaksi. Maya on ammattilaisten suosima ohjelma ja sitä käytetään varsinkin elokuva - ja peliteollisuudessa. Kirjoitushetkellä uusin versio Mayasta on 2012, mutta tässä opinnäytetyössä käytetään Maya 2011:sta. Versioilla ei ole paljon eroa tätä työtä tehdessä, koska riggaus ei ole juurikaan muuttunut näiden versioiden välillä.

4.2 Nodet & Historia

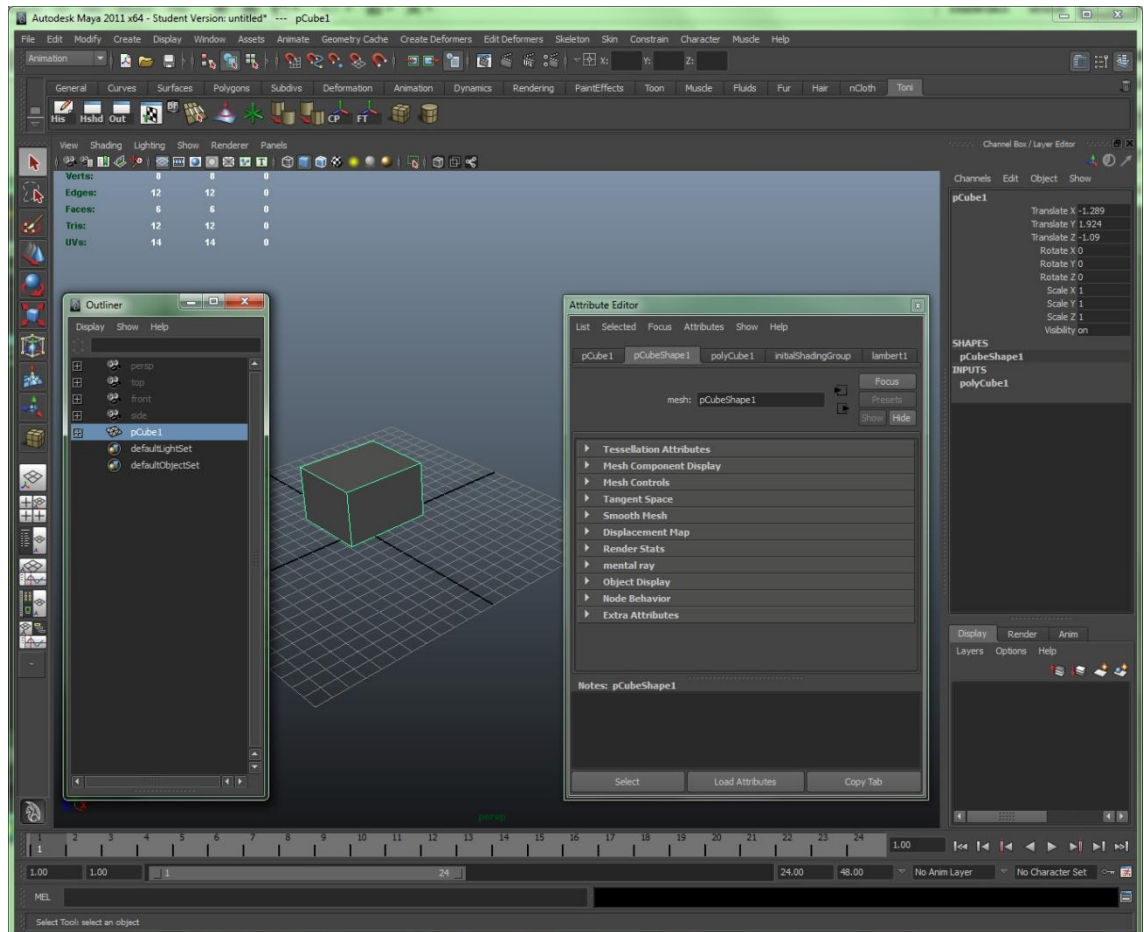
Mayan pintaa syvemmällä kaikki toiminnot perustuvat nodeihin. Nodet ovat scenessä olevia objekteja, jotka sisältävät jokainen omat attribuuttinsa. 3d-mallitkin muodostuvat nodeista, joissa on vähintään kolme eri nodea. Ensimmäinen on shape-node, joka sisältää tiedon 3d-objektin muodosta. Toinen on material-node, joka sisältää 3d-mallin shaderin, sen miltä objekti näyttää. Kolmas on transform-node, joka sisältää objektin attribuuteista sijainnin (translate), suunnan (rotate), skaalauksen (scale) ja näkyvyyden (visibility).

Nodet usein linkittyvät toistensa attribuutteihin ja tähän voi myös käyttäjä vaikuttaa. Nodejen käyttäminen ei ole välttämätöntä yksinkertaisissa projekteissa, mutta niiden ymmärtäminen auttaa monessa asiassa, mm. instanseissa, josta lisää myöhemmin. Nodeihin pääsee käsiksi Window\Hypergraph:Connections -valikon kautta. Myös Mayan sharedeita käsitellään nodeilla, joihin pääsee erikseen: Window\Rendering Editors\Hypershade.

Maya tallentaa melkein kaikki objekteihin tapahtuvat muutokset Historiaan. Historia tekee muutoksista uusia nodeja joita voidaan tarkastella Hypergraphista. Muutoksia voidaan usein poistaa ja muokata jälkikäteen. Historia kertyy nopeasti scenen kasvaessa, joten sitä tarvitsee tyhjentää aika ajoin. Yleensä tämä tehdään silloin, kun objekti on valmis. Scenen muuttuessa hitaaksi voi kyseessä olla Historian suuri koko. Historiaa ei saa poistaa täysin huolettomattomasti, koska myös objektien deformaatiot tallentuvat Historiaan. Historian poistoon löytyvät työvälaineet löytyvät Edit/Delete By Type -valikosta.

4.3 Mayan käyttöliittymä

Käyttöliittymä käydään läpi vain tasolla, joka vaaditaan tämän opinnäytetyön ymmärtämiseen.



Kuvio 1. Yleiskuva Mayasta. Attribute Editor on irroitettu oikeasta reunasta.

Mayan toiminnot on jaettu eri kategorioihin, esim: Animation, Polygons, Surfaces, Rendering, jne. Eri kategoriat valitaan vasemman ylänurkan pudotusvalikoista tai näppäimistöltä F2-F6. Kategorioiden valinta muuttaa ylärivin työkaluriviä aina Assets-valikosta eteenpäin. Osa työkaluista löytyy myös työkalurivin alapuolelta olevasta Shelfistä, eli hyllyltä. Shelfit ovat täysin kustomoitavia ja suositeltavaa onkin, että käyttäjä kokoaa oman hyllynsä eniten käyttämistä työkaluista.

Vasemmalla pystyrivillä ovat Mayan perustyökalut aina objektien liikkutuksesta skaalaamiseen. Samalla rivillä alhaalla on myös viewporteja muuttavia valintoja.

Oikealle pystyriville voidaan valita Channel Box / Layer Editor tai Attribute Editor. Channel Box näyttää valitun objektin perusattribuutit. Sen alapuolella on erilaisia layervalikkoja. Vaihtoehtoisesti oikealle puolelle voidaan valita Attribute Editor, joka on Channel Boxista kehittyneempi ja näyttää enemmän tietoja valituista objekteista. Palkkeja voi myös irroittaa, kuten Attribute Editor Kuvio 1:ssä.

Alhaalla oleva palkki sisältää aikajanan ja sen kontrollointiin tarkoitetut työkalut. Alla vasemmalla on myös MEL tai Python skripaukseen oleva harmaa tekstipalkki, josta voidaan suorittaa skriptejä. Harmaan palkin oikealla puolella oleva musta tekstipalkki näyttää kulloisenkin suoritettujen skriptin tiedot. Aivan alimmaiselle riville tulee ohjetekstejä valitun työkalun mukaan.

Mayan Outliner (Window\Outliner) näyttää kaikki scenen objektit ja niiden hierarkian. Outlineriä voidaan käyttää objektien järjestämiseen, nimeämiseen ja parentointiin. Lisäksi Outlineristä löytyy objekteihin liitetyt constraintit. Monipuolisuuden ansiosta Outliner onkin monesti riggaajalla jatkuvasti esillä. Parentoinnista ja constrainteista lisää niiden omissa kappaleissa.

Hypershaderia (Window/Rendering Editors/Hypershader) käytetään yleisimmin shadereiden tekemiseen, mutta riggaajallekin sillä on tärkeä rooli. Hypershaderissa voidaan yhdistellä objektien attribuutteja ja lisätä niille esim. laskutoimituksia.

Mayasta löytyy kaksi erilaista Hypergraph editoria (Window\Hypergraph), josta toinen Hypergraph: Hierarchy näyttää objektien hierarkian ja toinen Hypergraph: Connections objektien nodet ja niiden yhteydet attribuutteihin.

Mayassa objektien muokkaamiseen on kaksi eri toimintatapaa. Ensimmäinen on object-mode, jossa objektia käsitellään kokonaisuutena, mm. objektin sijaintia maailmassa tai pivotin sijaintia. Toinen on component-mode, jossa objektia voidaan käsitellä sisäisesti, esim. vertexien, edgejen ja polygonien muokkaukset tehdään tällä. Eri modeja voidaan vaihdella ylhäältä työkaluriviltä löytyviltä napeilta (select by object type/component type) tai vaihtoehtoisesti näppäimistöltä F8-F12.

4.4 Projektiasetukset & Referointi

Uusi työ on hyvä aloittaa valitsemalla Projektiasetukset. Projektiasetukset löytyvät Mayasta File\Project\ -valikosta. Tällöin kaikki projektiin liittyvät tiedostot, kuten 3d-mallitiedostot, tekstuurit, rendatut kuvat, yms. tallentuvat yhteen projektikansioon ja omiin alikansioihin. Jos työn alla on isompi projekti, esim. animaatio, kannattaa tehdä uusi projekti aina jokaiselle objektille. Tätä projektia varten on tehty omat projektikansiot: helikopterille, jokaiselle taustalle ja lisäksi jokaiselle kohtaukselle omansa.

Projekteja voidaan yhdistellä referoimalla (File\Reference Editor), esim. tässä työssä on referoitu helikopteri ja taustat kohtauksiin. Referoituja objekteja ei ole tarkoitus muokata, koska ne ovat vain ns. peilikuvia alkuperäisistä. Muokkaukset referoituihin objekteihin eivät vaikuta alkuperäisiin tiedostoihin. Referointi on tärkeä työkalu, jonka avulla voidaan testata myös keskeneräisiä malleja, kuten teksturoimattomia hahmoja kohtauksissa.

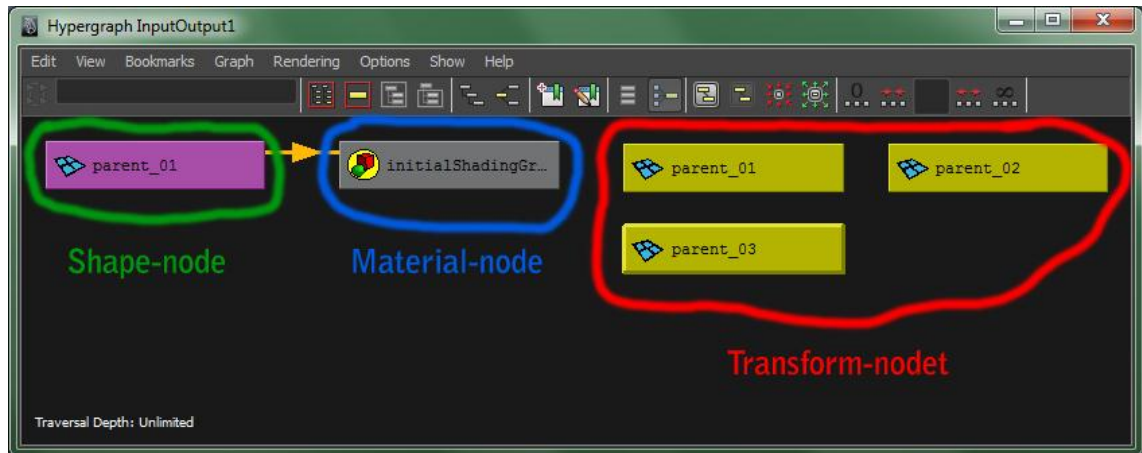
4.5 Objektien instanssointi

Objektien instanssointi tarkoittaa kopioimista, jossa jokaiselle kopioidulle objektille tehtävät muutokset vaikuttavat muihin kopioihin. Instanssoidut objektit ovat kuin peilikuvia toisistaan. Mayassa node-pohjaisuus voi aiheuttaa ongelmia instansseihin.

Joskus Mayassa objekteja instanssoidessa voi tulla vastaan yllättäviä ongelmia, jos ei tiedä mitä instanssien nodeille tapahtuu. Esim. jos skaalaa 3d-objektia objektimoodissa ei muihin instanssoituihin objekteihin tulekaan muutoksia. Tämä johtuu siitä, että jokaiselle instanssoidulle objektille luodaan oma transform-node, joka sisältää myös skaalaus-attribuutin. Jos haluaa, että skaalaus vaikuttaa kaikkiin instansseihin, niin täytyy skaalaaminen tehdä component-modessa, esim. vertexejä skaalamalla.

Toinen huomioitava asia instanssoiduissa objekteissa on se, että ne kaikki jakavat saman shape-noden, joka sisältää kaikkien instanssien muodon. Shape-noden poistaminen hävittää kaikkien instanssien tiedot niiden muodoista ja jäljelle voi jäädä vain transform- tai material-nodet. Shape-node näkyy myös Outlinerissä jokaisen

instanssi-objektin alla, joka on helppo poistaa vahingossa. Monesti riggaajat välttelevät Mayassa instanssoimista, koska se aiheuttaa helposti peruuttamatonta tuhoa.



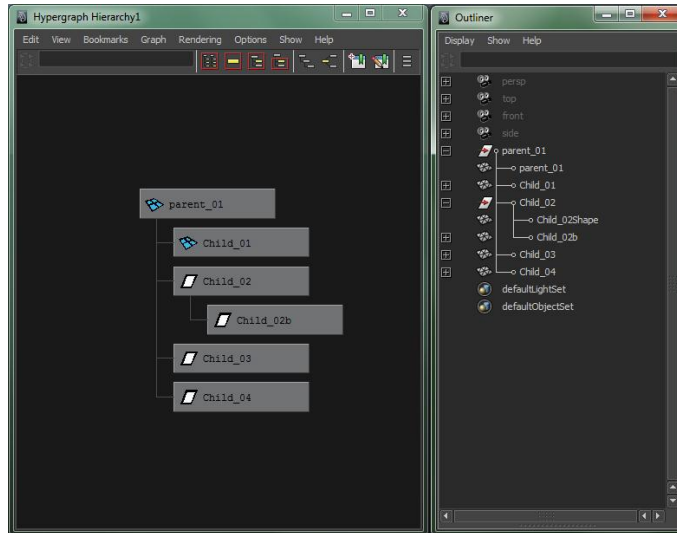
Kuvio 2. Kolme instanssoitua objektia jakavat saman shape- ja material-noden.

4.6 Parentointi ja ryhmät

Parentointi on riggauksen yksi tärkeimmistä asioista. Käytännössä parentointi yhdistää objekteja toisiinsa, jossa yksi objekti on parent, eli määräävä objekti ja siihen parentoidut objektit ovat childejä, jotka seuraavat parenttia. Parent-objektilla voi olla useita child-objekteja. Alemman tason child-objektit saavat attribuutit ylemmällä hierarkiatasolla olevalta parentilta, esim. jos tämä liikkuu, kääntyy tai muuttaa kokoa. Alemman tason objektit säilyttävät attribuuttiarvot, jotka niillä oli ennen parentointia. Node-tasolla jokaisen parentin transform-node on Mayan peruskoordinaatiossa, mutta child-objektit ovat suhteessa parentin koordinaatistoon. Parentoinnin hierarkiaa voi tarkastella Mayan Outlineristä tai Hypergraph Hierachystä jossa parentit ovat ylimpänä ja childit sen alla.

Parentointi voidaan Mayassa tehdä usealla tavalla. Yksi tapa on valita viewportissa ensin child ja sen jälkeen parent ja painaa näppäimistöltä P tai valikosta Edit\Parent. Parentoinnin peruminen tapahtuu painamalla Shift-P tai valikosta Edit\Unparent. Toinen tapa on mennä Outlineriin tai Hypergraphiin ja raahata hiiren keskinäppäimellä child parentin päälle. Parentointi voidaan perua vetämällä objekti pois samalla tavalla.

Ryhmät toimivat hieman parentoinnin tapaan, mutta erona on, että ryhmä on oma objektinsa, jolla on omat attribuuttinsa. Ryhmiä voidaan tehdä menemällä valikkoon Edit/Group (ctrl-g) tai perua Edit/Ungroup.



Kuvio 3. Hypergraph ja Outliner näyttävät samat objektit.

4.7 Constraintit

Constraintit ovat eräänlaisia objektien linkittäjiä. Constrainteilla usein automatisoidaan animaatiota, jotta jokaista liikettä ei tarvitse käsin animoida, vaan liike otetaan joltain toiselta objektilta. Niillä yhdistetään yleensä kaksi tai useampi objekti keskenään, joista määräävää objektia kutsutaan yleensä target-objektiksi. Target-objekti on kohde, johon muut objektit yhdistetään constrainteilla. Constraintteja voi olla yhdellä objektilla useita ja niiden painoarvoja voi muuttaa (weight). Tällä tavalla jonkun constraintin vaikutusta voidaan vähentää tai ottaa kokonaan pois päältä. Weighttien käsittely on yleinen tapa laittaa jokin toiminto päälle tai pois. Constraintteja käytetään myös silloin, kun ei haluta parentoida objekteja keskenään, mutta halutaan silti niiden vaikuttavan toisiinsa. Tässä kappaleessa käydään läpi vain constrainttien perustoiminnot. Lisäksi jokaisella constraintilta löytyvät omat lisäasetukset, joita osaa käsitellä työvaiheessa. Constraintit löytyvät Animation\Constrain -valikosta.

Mayassa on 11 erilaista constrainttia, jotka voi jakaa kolmeen ryhmään. Ensimmäiseen ryhmään kuuluvat: Point, Aim, Orient, Scale ja Parent. Tämän ryhmän constraintit

vaikuttavat pelkästään objektien transform-noden attribuutteihin. Toiseen ryhmään kuuluvat: Geometry, Normal, Tangent, Point on Poly, Closest Point. Nämä constraintit vaikuttavat objektien geometriaan, esim. polygoneihin ja vertexeihin. Kolmanteen ryhmään kuuluu vain yksi constraint: Pole Vector. Tämä viimeinen constraint vaikuttaa ainoastaan IK-Handleen. Seuraavissa kappaleissa constrainteja käsitellään tarkemmin.

Point-constrain

Tällä constraintilla saadaan kaksi tai useammat objektit pysymään samassa kohdassa. Constrainteilla yhdistetyt objektit ottavat translate-attribuutit target-objektilta. Objekteilla voi olla useita target-objekteja, jolloin weight-arvot jakaantuvat automaattisesti target-objektien kesken. Weight-arvoja voidaan animoida. Point-constraintin asetuksista löytyy maintain offset, jolla voi säätää, pysyykö objekti omalla sijainnillaan vai siirtykö targetin luokse.

Aim-constrain

Aim-constraintilla saadaan objektin haluttu akseli katsomaan target-objektin pivottia. Aim-constraineja käytetään usein kameroiden, valojen ja silmien kohdistamiseen.

Orient-constrain

Orient-constraintilla objekti pyörii oletuksena kaikilla akseleilla target-objektin mukana. Objektin rotate-attribuutit korvautuvat targetin vastaavilla.

Scale-constrain

Nimensä mukaisesti scale-constrain vaikuttaa objektien scale-attribuutteihin. Target-objektia skaalaamalla saadaan objektit muuttamaan kokoaan.

Parent-constrain

Parent-constrain toimii melkein samaan tapaan kuin tavallinen parentointi. Asetusten kautta voi valita halutut attribuutit jotka objekti saa targetilta. Pelkässä parentoinnissa

child-objekti on itsenäinen objekti, jonka sijaintia pystyy muuttamaan. Parent-constraintin etuna on, että parentointi voidaan laittaa päälle ja pois animoinnin aikana weightiä säätämällä. Tällä tavoin saadaan esim. hahmo nostamaan esine pöydästä. Parent-constraina käytetään myös silloin, kun objektit halutaan parentoida, mutta ei laittaa samoihin ryhmiin.

Geometry-constrain

Geometry-constrainin objektina tai targetina voi olla polygon, nurbs-pinta tai nurbs-käyrä. Tällä constrainilla objekti voidaan kiinnittää targetin geometriapinnalle. Esineet tarttuvat pintaan kiinni ja niitä voidaan liikuttaa pintaa pitkin. Objekti ei kuitenkaan rotatoidu pinnan muotojen mukaan. Objekti tarttuu kiinni omasta pivotistaan.

Normal-constrain

Tämä constrain toimii hieman Aim-constrainin tapaan. Tämäkin objekti suuntaa pivottinsa targettia kohden, mutta targetin normalin, eli pinnan myötäisesti. Tyypillisesti objekti kiinnitetään target-objektiin Geometry- ja Normal-constraineilla, jotta objekti saadaan kiinnittymään ja rotatoitumaan targetin pinnalla. Tällä tavalla voidaan objekti saada liukumaan epätasaisellakin pinnalla kohtuullisen helposti.

Tangent-constrain

Tällä constrainilla saadaan objekti rotatoitumaan targettina toimivan nurbs-kurvin suuntaisesti. Tangent-constraina käytetään tavallisesti Geometry-constrainin kanssa, jotta objekti saadaan pysymään kurvissa ja pyörimään sen muotojen mukaan. Tällä menetelmällä voitaisiin saada vaikkapa juna pysymään mutkaisella raiteella. Mayasta löytyy myös samankaltainen Motion Path -toiminto (Animation\Animate\Motion Paths), jossa on hieman enemmän toimintoja. Motion Pathia ei käsitellä tässä opinnäytetyössä.

Point on Poly -constrain

Tämän constrainin avulla objekti saadaan pysymään targetin pinnalla, kun objekti deformoi eli muuttaa muotoaan. Objekti voidaan kiinnittää targetin verteksiin, edgeen tai polygoniin. Constrainin avulla voidaan esim. kiinnittää deformoituviin renkasiin irtokiviä.

Closest Point -constrain

Tätä constrainia käytetään etäisyyksien laskentaan. Constrainia varten valitaan vain yksi objekti. Constrain luo kaksi locatoria. Ensimmäinen locator kiinnittyy objektin pintaan. Se toimii samaan tapaan kuin geometry-constrain, pysyen kiinni objektin pinnassa. Toinen locator on vapaasti liikuteltava. Eri etäisyydet löytyvät Attribute-editorista constrainin välilehden alta. Locatoreja voi halutessaan käyttää parentteina tai muiden constraintien target-objekteina.

Pole Vector -constrain

Pole Vector -constrainilla saadaan IK-handlen Pole Vector -attribuutin hallinta target-objektille. Yleinen käyttökohde tälle constrainille on mm. ihmisen kyynärpäiden tai polvien kohdistaminen, kun käytetään jointteja ja IK-handleja.

4.8 Jointit, eli luut

Mayassa luita ja niveliä kutsutaan jointeiksi. Jointeilla yhdistetään liikkuvia osia toisiinsa. Jointteja käytetään lähes aina orgaanisten olentojen riggaamisessa, mutta mekaanistenkin esineiden riggauksessa niillä on oma roolinsa. Mayassa IK-handleja ei voi kiinnittää esim. polygoni-objekteihin. Jos haluaa tehdä moniosaisia taittuvia nivel-esineitä, jossa tarvitaan IK-pohjaista liikettä, niin jointit ovat silloin pakollisia. Jointtiketjuja tehdessä parentointi tapahtuu automaattisesti, vaikka joskus jointtien rotaatioita saadaan hallittua paremmin tekemällä yksittäisiä jointteja ja parentoimalla ne jälkikäteen. Rigattuihin jointteihin objektit saadaan kiinni parentoimalla tai skinnaamalla. Jointit löytyvät Mayasta Animation/Skeleton -valikosta tai animation-

shelfistä. Jointteja käsitellään tarkemmin työvaiheessa, jossa rigataan tuulilasinyyhkimet.

4.9 IK-handle

IK-handlet ovat Mayassa yleisimmin jointteihin liitettäviä ketjuja, joilla voidaan yhdistää kahden tai useamman joint-ketjun alku- ja loppupää yhteen. IK-handlen loppupäähän muodostuu end effector ja luiden väliin muodostuu IK-handle. End effector on viewportissa näkymätön osa, mutta Hypergraphin kautta se näkyy. IK-handlea liikuttamalla end effector pyrkii laskemaan sijaintinsa ja asentonsa IK-handlen mukaan. Tämä näkyy luiden liikkumisena nivelten tapaan. IK-handlella on Mayassa useampia solvereita, jotka ovat erilaisia laskentatapoja, joilla end effector laskee sijaintinsa. Tässä opinnäytetyössä keskitytään käyttämään ikSCsolveria (Single Chain).

Tavallinen käyttökohde IK-handlelle ovat elävien hahmojen kädet ja jalat. Tässä opinnäytetyössä IK-handleja käytetään monella tapaa, josta lisää niiden omissa kappaleissa (mm. tuulilasinyyhkimet). IK-handleen liittyy läheisesti myös aiemmin mainittu Pole Vector -constrain. Tällä saatiin siis IK-handlen keskivaihe suunnattua haluttuun kohteeseen. Pole Vector -constrainin käyttö edellyttää ikRPsolverin (Rotate Plain) käyttöä IK-handlessa. Mayassa IK-ketjuja voidaan kiinnittää myös nurbskurveihin ja effectoreihin, mutta tässä opinnäytetyössä niiden tekniikoita ei käydä läpi. IK-handlet löytyvät Mayasta Animation/Skeleton -valikosta.

4.10 Skinnaus

Skinnauksella saadaan esim. polygon-objekti kiinnittymään useampaan jointtiin kerralla. Näin polygon-objekti saadaan taittumaan pehmeästi jointtien taittumakohdasta. Tavallisia käyttökohteita tälle tekniikalle ovat erilaiset orgaaniset hahmot, joiden taipuvien nivelten täytyy pysyä jointtien mukana. Mekaanisessa riggaamisessa tätä voisi käyttää esim. letkujen tekemiseen. Käytännössä polygon-objektista valitaan vertexit, jotka kiinnittyvät eri jointteihin eri painoarvoilla (weight).

Tässä työssä tälle tekniikalle ei ole paljoa käyttöä, eikä sitä käydä kovin tarkasti läpi. Skinnaukseen liittyvät työkalut löytyvät Animation/Skin -valikosta.

4.11 Deformerit

Deformerit ovat erilaisia pintojen muokkaajia, joita voidaan animoida. Näiden avulla saadaan nopeasti rigattua erilaisia taipumia, pullistumia tai muita yksinkertaisia muodonmuutoksia. Deformereita käytetään myös usein mallinnuksen apuvälineinä. Historia poistamalla objektilta saadaan deformerit pois niin, että muutokset säilyvät. Kaikkia deformereita ei tässä kappaleessa käydä läpi, mutta alla esitellään muutamia käytetyimpiä. Deformerit löytyvät: animation/create deformer -valikosta.

Blend Shape

Nimensä mukaisesti Blend Shape sekoittaa muotoja. Tätä varten valitaan base-objekti jolle muodon muutokset sekoitetaan target-objekteista. Target-objekteja voi olla useampia. Yleensä tarkoituksena on, että base-objekti on normaalissa muodossaan ja target-objektit muokkaavat sitä halutulla weight-arvolla. Normaali käyttökohde tälle tekniikalle voisi esimerkiksi olla muuttuvat kasvon ilmeet tai ajoneuvoon ilmestyvät lommot. Base- ja target-objektien täytyy perustua samaan objektiin, jotta sekoitukset toimivat. Tavallisesti base-objekti mallinnetaan ensin ja sen pohjalta tehdään eri muotoinen target-objekti. Blend Shape tunnetaan monesti Morph-nimellä muissa ohjelmissa, esim. 3ds Maxissa ja UDK:ssa.

Lattice

Lattice luo valitulle objektille kehikon, johon saa halutun määrän pisteitä (Lattice Point), joista kehikon muotoa voi muuttaa halutulla tavalla. Latticen pisteitä liikuttamalla liikkuvat myös sitä lähimpänä olevat vertexit. Latticen pisteitä voidaan myös animoida.

Nonlinear

Tähän ryhmään kuuluu useita deformereita, kuten: bend, twist, wave, jne. Näillä saa tehtyä nimensä mukaisia muokkauksia. Bendillä saa objektin taipumaan, twistillä vääntymään ja wavella aaltoilemaan. Bendiä on tässä työssä käytetty helikopterin lapojen taivuttamiseen. Squash-deformerilla litistetään helikopterin renkaita.

4.12 Connection Editor

Editorilla yhdistetään kahden valitun objektin attribuutteja toisiinsa. Tällä tavalla objektien valitut attribuutit ovat tavallaan "instanssissa" keskenään. Editorilla saadaan esim. objektin y-akselin liike kopioitua toisen objektin z-akseliin.

4.13 Set Driven Key

Tällä editorilla saadaan nopeasti yhdistettyä objektien attribuutteja toisiinsa hieman eri tapaan kuin Connection Editorissa. Editorissa valitaan kaksi objektia, joista toinen on driver, "kuljettaja" joka hallitsee attribuutillaan toisen objektin, drivenin attribuuttia. Tällä tekniikalla on usein käyttöä mekaanisessa riggaamisessa ja tässäkin lopputyössä editoria käytetään useasti, mm. tuulilasin pyyhkimessä, jossa pyyhkimen hallinta siirretään kontrolliohjelmalle. Editori löytyy valikosta: animation/animate/set driven key/set.

4.14 Riggauksen huomioiminen mallintamisessa

Myös mallintajan tulee huolehtia osaltaan 3d-mallin toimivuudesta animaatiossa. Mallintajan tulee tietää mitkä osat liikkuvat ja ovat yhteydessä toisiinsa. Mallintajan on hyvä tietää mistä kohdasta objektit rotatoivat. Yleensä pivotin kohdalla mallin rakenne on sellainen, että pivotin siirtäessä vahingossa paikkaa se voidaan liittää siihen helposti uudestaan. Objektien rotaatioiden on hyvä olla tasalukuja, jotta rotaatioiden epähuomiossa hävittyä ne voidaan helposti saada takaisin suoraan.

Mallintaja huolehtii mallin siisteydestä, jotta siinä ei ole esim. nGoneja, eli polygoneja joilla on enemmän kuin neljä verteksiä. NGonet voivat aiheuttaa ongelmia mm. rendauksessa tai dynamiikkalaskennoissa. Mallin scenen täytyy muutenkin olla puhdas, niin että siitä otetaan kaikki ylimääräiset Historiat, nodet ja materiaalit pois. Riggaajaa auttaa myös se, että mallin eri osat ovat nimettyjä ja kenties jaoteltu ryhmiin. Riggaaja kuitenkin päättää rigattujen esineiden lopullisen hierarkian.

4.15 Scriptaaminen Mayassa

Mayassa rigatessa pärjää monesti ilman scriptaamista. Silloin kun työkalujen toiminnot eivät riitä, voidaan apuna käyttää Mayan eri scriptaustapoja.

Yksinkertaisin tapa tehdä scriptejä ovat Expressionit. Nämä ovat yleensä tiettyihin attribuutteihin sidottuja lyhyitä scriptejä. Expressionilla voidaan esim. tehostaa jonkin liikkeen vaikutusta, mm. kertolaskulla. Tässä lopputyössä tehdään muutama Expressioni. Expressionit pohjautuvat MEL-scriptauskieleen, mutta eivät aina sellaisenaan toimi keskenään.

Expressionit ovat Mayassa scriptejä, jotka laskevat arvoja yhdeltä tai useammalta attribuutilta. Mayassa on kahdenlaisia expressioneja: particle expressionit, jotka kontrolloivat per-particle attribuutteja particle-objektilta, ja objektien attribuutti expressionit, jotka kontrolloivat per-object attribuutteja. (Wilkins, Mark R & Kazmier, Chris 2005, 37)

Lisäksi Mayasta löytyy MEL- ja Python-scriptauskielet joilla yleensä vaikutetaan Mayan toimintoihin yleisesti. Näitä scriptauskieliä käytetään usein esim. kun tiettyä toimintoa pitää tehdä kymmenille objekteille kerralla. Näillä kielillä tehdään myös Mayan monet lisäosat ja pluginit. Tässä lopputyössä ei käydä läpi näitä scriptauskieliä yhtä MEL-lausetta enempää.

MEL-scriptit ovat yleensä scenen ulkopuolella. Niiden suorittaminen aktivoidaan valitsemalla menu-valikosta tai klikkaamalla shelf-nappia. MEL-scriptejä voidaan käyttää myös scenen sisällä käyttämällä script-nodea: ominaisuutta, jota käytetään usein esim. suorittamalla scripti, kun scene avataan; tai varmistetaan, että MEL-scripti on saatavilla, kun scene lähetään jollekin toiselle joka työskentelee muualla. MEL-scriptien tärkein tehtävä on kuitenkin automatisoida Mayan monimutkaisia scenejä. (Wilkins, Mark R & Kazmier, Chris 2005, 95-96)

4.16 Yleisiä vinkkejä riggaukseen Mayassa

Tässä kappaleessa käydään läpi joitain vinkkejä tai ongelmanratkaisuja, joista suurin osa on tullut esille tätä työtä tehdessä.

Objektien linjaaminen keskenään

Mayasta puuttuvat perinteiset align-työkalut, mutta objektien linjaamiseen keskenään voi käyttää point-, orient- ja parent-constraintteja. Constraintit poistetaan käytön jälkeen.

Rigattujen esineiden kopioiminen

Constrainteilla yhdistetyt esineet menettävät linkitykset kopioitaessa. Linkitykset täytyy korjata käsin nodeja säätämällä tai tekemällä uudet constraintit. Samoin käy mm. IK-handleille ja deformereille.

Objektien freezaaminen ilman, että pivotti kääntyy

Joskus voi tulla tarvetta sille, että objektin täytyy olla rotatoitu, mutta rotaatioiden tulisi olla samalla freezattuja. Tämä tehdään luomalla esim. tyhjä group samaan kohtaan ja samalla rotaatiolla kuin haluttu objekti. Objekti parentoidaan tähän ryhmään. Objektin rotaatiot voidaan freezata ilman, että objekti kääntyy Mayan peruskoordinaation mukaan, koska objekti ottaa rotaatiot ylemmän tason groupilta.

Objektit deformatuvat rotatoidessa

Jos objektit muuttavat oudosti muotoaan kun niitä rotatoidaan, tämä johtuu siitä, että jollain ylemmän tason objektilla on epäsymmetristä skaalausta. Ongelma voidaan korjata freezaamalla skaalausarvot. Ennen riggausta skaalaukset on muutenkin hyvä freezata.

Objektit muuttuvat näkymättömäksi tai mustaksi

Joskus Maya kadottaa objekteista yhteyden shadereihin ja objektista tulee näkymätön. Tällöin objekti täytyy linkittää shaderiin uudestaan. Jos polygon-objekti muuttuu mustaksi, sen normalit eli pinta on kääntynyt nurin. Valitsemalla kaikki polygonit ja käyttämällä Reverse-toimintoa (Polygons/Normals/Reverse) ongelma korjaantuu.

Aim-constrain objektit flippaavat

Toisinaan aim-constrainteilla yhdistetyt voivat flipata väärään suuntaan, jos parent-objekti pyörii esim. 360-astetta. Tällöin voidaan käyttää aim-constraintin asetuksista World up type -pudotusvalikosta Object up. World up object:ksi valitaan aim-constraintin target objekti.

Erimuotoiset kontrolliobjektit

Perinteisesti kontrolliobjekteihin käytetään nurbs-käyriä, jotka löytyvät valikosta: Create/Nurbs primitives. Kontrolliobjekteihin voidaan käyttää myös fontteja, jotka löytyvät Create/Text -valikosta. Fonteista Windings 1, 2, 3 löytyy hyviä nuolikuvioita.

Internetin oppaat

Nykyään löytää harvemmin esineitä/asioita joita ei olisi ennen rigattu. Tämän takia kannattaa ensin etsiä riggausongelmiin apua internetin oppaista ja keskustelupalstoilta. Scriptaamiseen löytyy myös hyviä kirjoja, joista yhtä on lainattu tähän lopputyöhön.

Robotin riggaamiseen löytyy hyvä Lester Banksin tekemä video-opas. Siinä käydään läpi useita mekaanisia osia. (Banks 2011, Lesterbanks.com)

5 Helikopteri

5.1 Helikopteri rigattavana mallina

Tähän työhön rigattavaksi malliksi on valittu taisteluhelikopteri Apache AH-64D. Helikopteri sisältää paljon erilaisia liikkuvia osia, jonka esimerkeistä on helppo soveltaa muihin ajoneuvoihin tai mekaanisiin esineisiin. Seuraavissa kappaleissa käydään helikopterin osat läpi opasmaisesti, koska se on hyvä tapa selvittää osien riggaamista. Opasmainen kirjoitustyyli toimii hyvin myös siksi, että siitä on mahdollisesti lukijoille paljon apua.

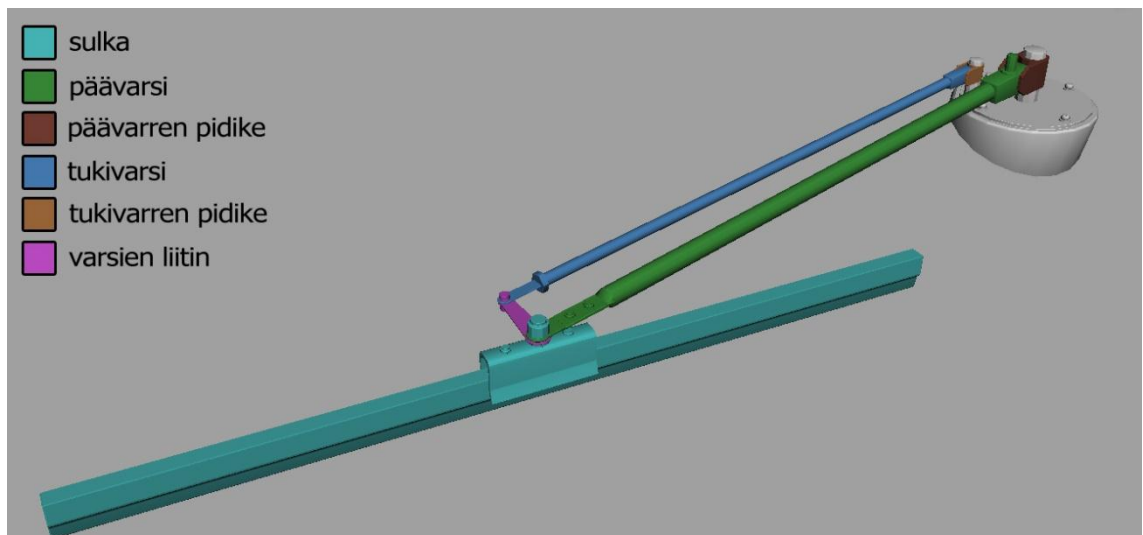
Tämän lopputyön aihe on ollut kirjoittajalle Mayan riggaukseen tutustumista. Kirjoittajalla on aiempaa kokemusta lähinnä 3ds Maxin työkaluista, joten kappaleet etenevät samalla kirjoittajan omana oppimisena. Työn alussa rigatut esineet voi kenties tehdä paremminkin tai erilailla. Tämän työn tarkoituksena on ollut oppia eri tapoja rigata esineitä.



Kuvio 4. Helikopterin mallinnus

5.2 Tuulilasinpyyhkimet

Ensimmäisenä rigataan tuulilasinpyyhkimet joita tulee kaksi kappaletta kuomun etuosaan. Tässä vaiheessa on selvää, että pyyhkimet ovat samanlaisia, mutta sulkaosa hieman erikokoinen. Tämän ansiosta molempia pyyhkimiä ei tarvitse erikseen mallintaa, mutta riggaaminen täytyy osittain tehdä kahteen kertaan. Ennen riggaamista pyyhkimen toiminta on mietitty jo mallinnusvaiheessa. Pyyhkimiin tulee yhteensä kuusi liikkuvaa osaa ja niiden apuobjektit. 2 vartta, 2 varren pidikettä, 1 liitin varsien välille ja 1 sulka. Aluksi pyyhkimet soviteltiin yksinkertaisilla polygon-cubeilla paikoilleen ja niin, että ne olivat lopullisen kokoisia. Sen jälkeen nämä objektit käännettiin suoraan maailman pivottiin nähden ja pyyhkimet voitiin mallintaa lopulliseen muotoonsa. Eri osien pivotit on hyvä miettiä etukäteen. Viimeisenä ennen riggausta objektit nimettiin ja niiden attribuutit freezattiin, mutta niin etteivät rotaatiot nolaudu niistä joissa se on oleellista.

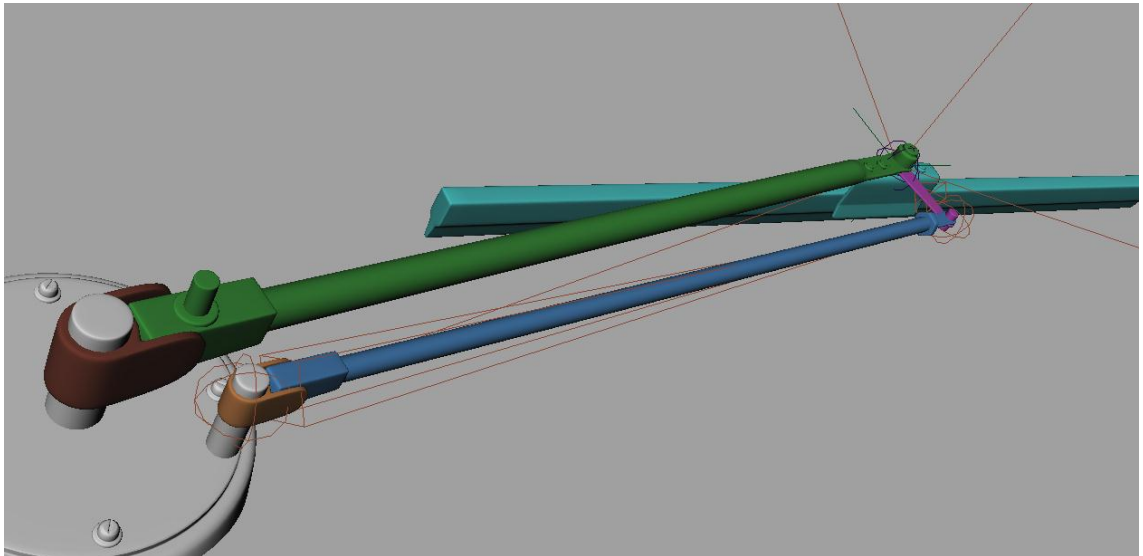


Kuvio 5. Tuulilasinpyyhkimen rakenne

Pyyhkimissä on kaksi vartta, joista isompi on liikkeen hallitseva osa ja pienempi varsi estää sulkaa pyörimästä. Isompi päävarsi voidaan rigata ensimmäisenä. Päävarren liikkeeseen tarvitaan käyttöön y-akseli, joka liikuttaa pyyhkimiä edestakaisin ja z-akseli, joka painaa pyyhkimiä tuulilasia vasten. Pivotin täytyy olla oikeassa asennossa, jotta liikkeessa pyyhkimet myötäilevät tuulilasin pintaa. Pyyhkimien liikettä voidaan vielä hienosäätää myöhemmin, kun pyyhkimet animoidaan ja lisätään kontrollien attribuutteihin. Päävarsi parentoidaan sen pidikkeeseen, jolloin varren z-akselin liikettä

hallitsee päävarsi ja y-akselin liikettä päävarren pidike. Päävarren päähän tehdään uusi locator, joka kiinnitetään päävarteen point constraintilla. Näin saadaan päävarrelle uusi pyörimätön kiinnityskohta. Tähän uuteen locatoriin parentoidaan sulka-objekti.

Seuraavaksi voidaan rigata tukivarsi. Tähän tarvitaan jointteja ja IK-handlea, koska liike on nivelmäistä. Aloitetaan tekemällä kahden jointin ketju, jonka ensimmäinen jointti alkaa tukivarren alusta ja loppuu varsien liittimen alkuun. Toinen jointti on varsien liittimien kokoinen. Jointteja ei tarvitse siirtää käsin paikoilleen, vaan apuna voidaan käyttää constraintteja. Constraintit poistetaan käytön jälkeen. IK-handle laitetaan IK-Handle toolilla (animation/skeleton) alkaen jointin alusta ja liittämällä se viimeiseen jointtiin. Solverina käytetään ikSCsolveria. Jos IK-Handlen lisäämisen jälkeen jointit pyörähtävät ympäri, niin tämä voidaan estää rakentamalla jointit ja IK-handle gridillä ja laittamalla ne paikoilleen valmiina. Toisena vaihtoehtona on käyttää eri solveria ja pyörittää IK-handlea twist-attribuutilla.

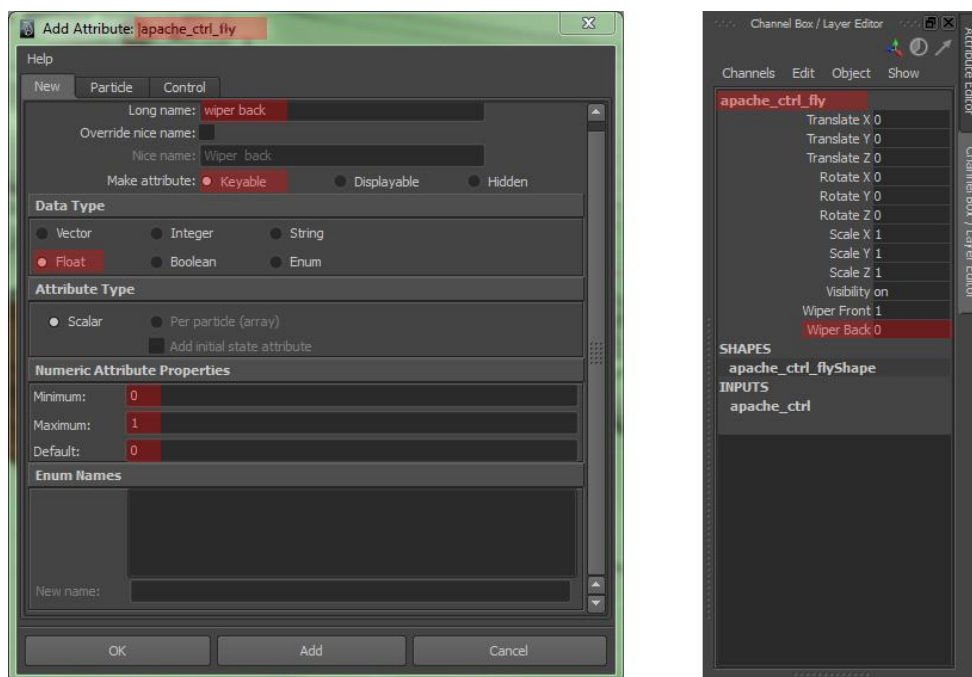


Kuvio 6. Tuulilasinpyyhkimen rakenne

Tukivarsi parentoidaan ensimmäiseen jointtiin, toinen jointti parentoidaan varsien liittimeen ja IK-handle parentoidaan locatoriin, joka tehtiin päävarteen. Tukivarren pidike saadaan katsomaan oikeaan suuntaan aim-constraintilla, jonka targetiksi voidaan laittaa tukivarsi. Aim-constraintin asetuksista täytyy määrätä, että se vaikuttaa vain y-akselilla, jotta tukivarren pidike pyörii vain yhdellä, sallitulla akselilla.

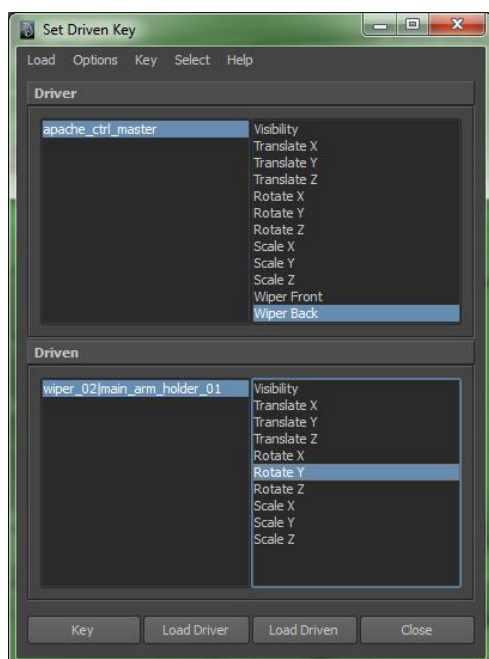
Viimeistelynä kaikki pyyhkimen osat apuobjekteineen laitetaan samaan grouppiin. Rigi toimii nyt jos sitä pyörittää päävarresta tai sen pidikkeestä.

Seuraavaksi tarkoituksena on siirtää rigin hallinta kontrolliobjektille ja animoida sen ääriasennot. Tätä varten tehdään haluttuun kohtaan kontrolliobjekti, josta lopulta helikopterin lentoa animoidaan. Kontrolliobjektiksi voidaan tehdä esim. Nurbs-Circle (Create/Nurbs/Primitives), joka nimetään `apache_ctrl_fly` (käytetään jatkossa nimeä Fly-kontrolli). Fly-kontrolliin parentoidaan kaikki helikopterin osat. Tehdään uusi attribuutti Fly-kontrollille. Uusi attribuutti tehdään Channel Boxin edit/add attribute -valikosta. Seuraavasta kuvasta (Kuvio 7) voi nähdä punaisella tärkeimmät asetukset. Otsikossa näkyy valittu objekti. Long name on uuden attribuutin nimi. Make attribute keyable tekee attribuutista animoitavan. Data type -otsikon alta valitaan Float, jotta saadaan desimaaliluvut mukaan ja pyyhkimiin voidaan animoida väliasennot. Numeric Attribute Properties -otsikon alta laitetaan edellä valitun arvon minimi, maksimi ja oletusarvot. Koska tuulilasinyyhkimiin tehdään yksinkertainen päälle/pois -toiminto, niin arvot ovat 0-1.



Kuvio 7. Vasemmalla attribuutin lisääminen ja oikealla valmis attribuutti

Seuraava tehtävä on animoida pyyhkimet. Tähän voidaan käyttää Set Driven Key -editoria (animation/animate/set driven key/set).



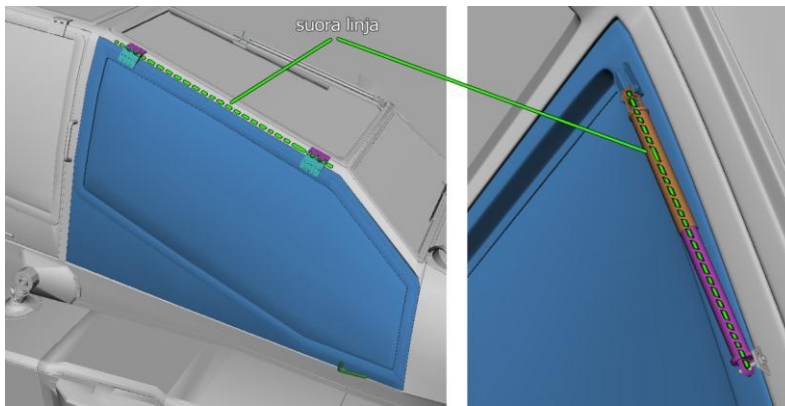
Kuvio 8. Pyyhkimen animointi uuteen attribuuttiin

Editorista valitaan Driver, objekti ja attribuutti jolla liikettä on tarkoitus hallita. Driveniin valitaan objekti joka liikkuu ja sen oikea attribuutti, eli tässä tapauksessa Rotate Y. Tässä vaiheessa on tärkeää asettaa attribuuteille aloituspiste. Tämä tehdään niin, että varmistetaan kontrolliohjelman wiper_back-attribuutin olevan arvoltaan 0: pyyhkimet ovat pois päältä. Samoin itse tuulilasinyyhkimen pitää olla oikeassa asennossa. Varmistusten jälkeen voidaan painaa Set Driven Key -editorista Key-nappia. Tämä tekee keyframen aloituspisteelle. Seuraavaksi laitetaan kontrolliohjelman wiper_back-attribuutti arvoon 1: pyyhkimet ovat päällä. Sitten liikutetaan itse pyyhkimet oikeaan asentoon ja painetaan editorista jälleen Key-nappia. Nyt pyyhkimien pitäisi toimia wiper_back-attribuutin arvoa muuttellessa.

Pyyhkimien liikkuminen ja sen siirtäminen uuteen attribuuttiin voidaan tehdä toisellakin tavalla. Tähän voitaisiin käyttää Hypershade-editorista löytyviä utility-nodeja. Multiply/Divide-nodea käyttämällä voitaisiin kertoa uuden attribuutin arvoa pyyhkimen rotaatiolla. Tätä tekniikkaa käytetään myöhemmin helikopterin aseistuksen männän jousituksessa.

5.3 Kuomun aukaisumekanismi

Helikopterissa on kaksi kuomun ovea, jotka molemmat ovat kaarevia muodoiltaan ja vaativat mallinnuksen osalta tarkkuutta, jotta esim. saranat toimivat kunnolla. Saranat on helpoin rakentaa niin, että ne ovat linjassa keskenään, jotta ovea avattaessa saranat eivät luiskahda paikoiltaan. Ovien sisäpuolelle kuuluu myös ovipumput, jotka estävät raskasta ovea putoamasta. Lisäksi ovissa on kahvat, jotka kääntyvät yhtäaikaan sisältä ja ulkoa.

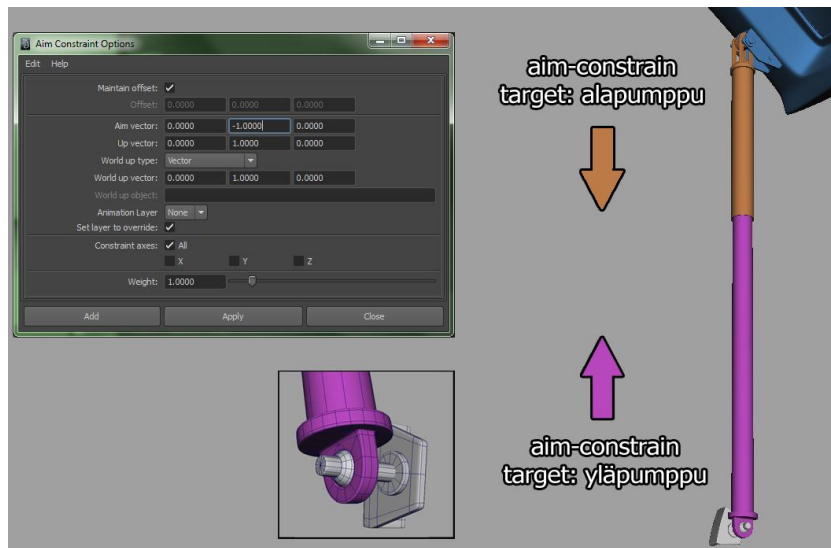


Kuvio 9. Vasemmalla ovi ulkopuolelta ja oikealla sisäpuolen ovipumppu

Saranat rigataan siten, että saranan toinen puoli on parentoitu helikopterin runkoon ja toinen ovenavauskontrolliin. Saranan osien väliin tulee tappi, jonka keskellä on ovien pivotti. Tässä tapauksessa saranalle tehdään uusi kontrolleri. Tämä voi olla parempi tapa kuin lisätä se uudella attribuutilla Fly-kontrolleriin. Näin toimitaan, koska ovi on läheisesti animaatiohahmon kanssa tekemisissä ja ovea animoidaan mieluummin lähellä hahmoa. Kontrollerin liikuttelu on muutenkin hienovaraisempaa kuin attribuutin arvojen säätely. Kontrolliohjelman rotaatiot täytyy nollata, jotta niistä voidaan liikuttaa ovea yhdellä rotate-attribuutilla. Oven kahvoille tehdään myös omat kontrolliohjelmat.

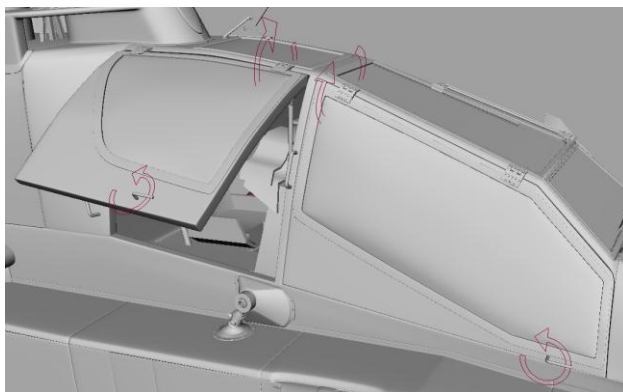
Oven pumput on mallinnettu ensin suoralla gridillä ja siirretty sen jälkeen paikoilleen. Pumpun molemmat päät on mallinnettu pyöreäksi, jotta pumpun päät saavat hieman elää liikkeen mukana. Tämä on tehty sen takia, että oven pumput eivät ole aivan suorassa linjassa oven avauslinjan kanssa. Ovipumpun ylempi osa (oranssi osa Kuvio 10) on parentoitu oveen ja alempi osa (pinkki Kuvio 10) helikopterin runkoon. Pumput saadaan suunnattua toisiinsa aim-constrainteilla. Aim-constraintin suuntausta voidaan

vaihtaa constraintin optionsista. Aim vector -riviltä löytyy x-, y-, z- akseleille omat lukunsa (Kuvio 10). Laittamalla aim vectorin jollekin sarakkeelle numeron 1 tai -1 muuttaa akselia, jolla aim-constrain katsoo targettia. Constrain axes on raksittu kohtaan All, jotta ylempänä valittu akseli voi pyöriä kohti targettia joka akselilla. Pumppujen toiminta on nyt täysin automatisoitua eikä animaattorin tarvitse huolehtia siitä.



Kuvio 10. Ovipumppuun tarvitaan kaksi aim-constraintia

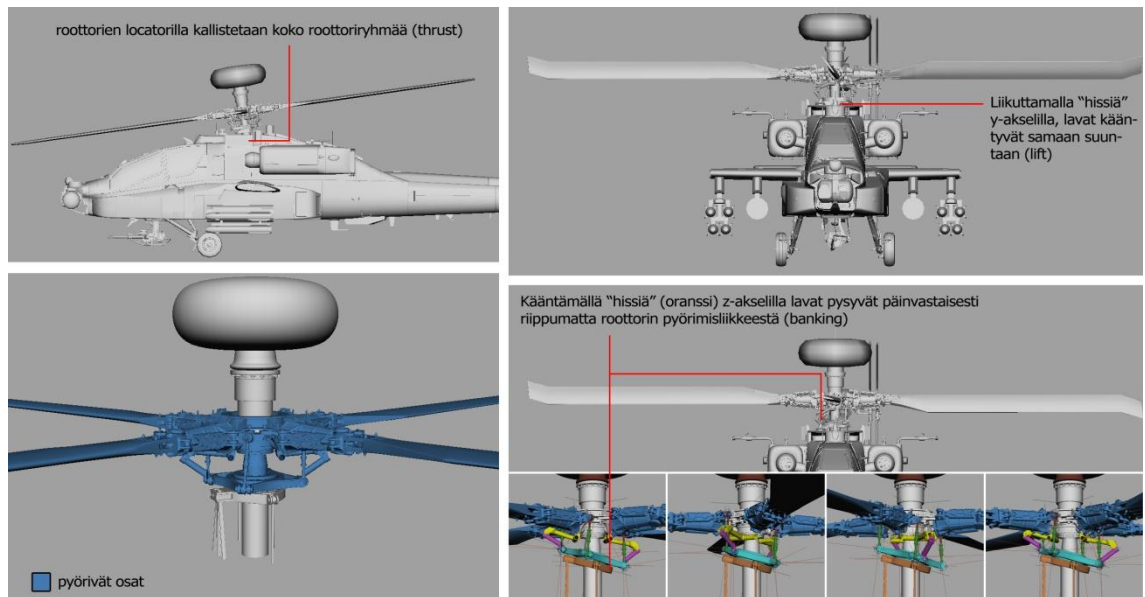
Ovirigin jatkokehittelyä ajatellen ovelle voisi lisätä animaatiohahmolle ystävällisemmän rigin, jossa oven kahvalle luotaisiin kontrolleri, jota nostamalla ovi nousisi. Tällöin hahmon käsi kontrolloisi oven nostoa. Tähän voisi käyttää jointeilla ja IK-handlella tehtyä vipua.



Kuvio 11. Valmiiksi rigatut ovet

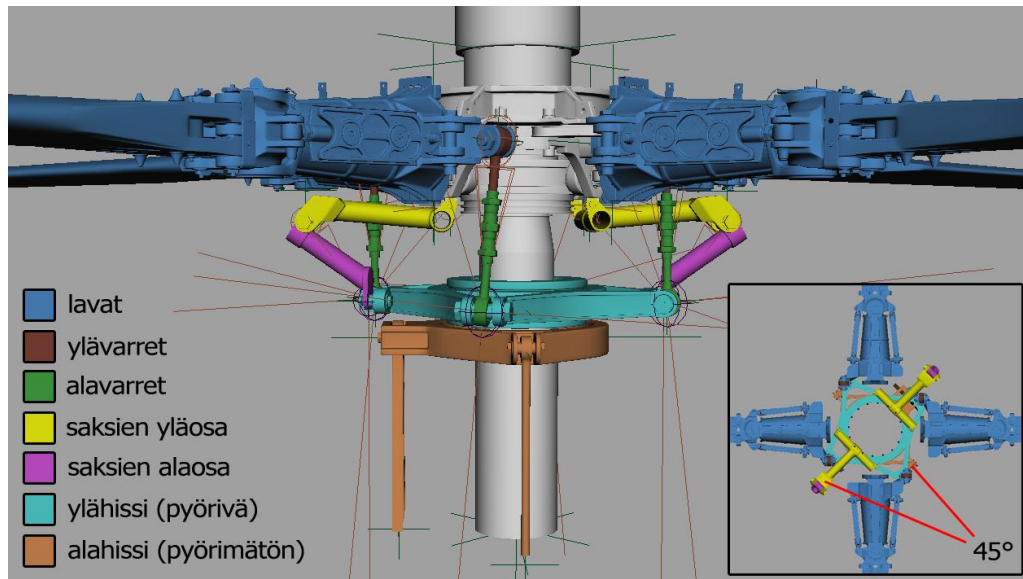
5.4 Pääroottori

Ennen roottorin mallintamista ja riggaamista sen toiminta kannattaa miettiä tarkkaan. Pääroottorille tarvitaan neljää erilaista liikettä. Ensimmäisenä roottorin päälliike, eli pyöriminen akselin ympäri (revolve). Toisena koko roottorin kallistaminen, jolla on ohjataan helikopterin lentosuuntaa (thrust) ja (side). Kolmantena on roottorin kaikkien lapojen kallistuminen yhtäaikaan, jolla säädetään helikopterin lentokorkeutta (lift). Neljäntenä lapojen täytyy vielä kallistua päinvastaisesti, jolla helikopteri kääntyy (banking).



Kuvio 12. Roottorin toiminta

Roottorin eri osat on hyvä laittaa aluksi ryhmiin selkeyden vuoksi. Kuvio 13:sta näkee liikkuvat osat vapaasti nimetyissä ryhmissään. Jokaisella ryhmällä on oma locatorinsa hierarkian huipulla, jonka tehtävä on hallita ryhmän liikettä ja johon constraintit ja muut apuobjektit laitetaan.



Kuvio 13. Pääroottorin eri osat

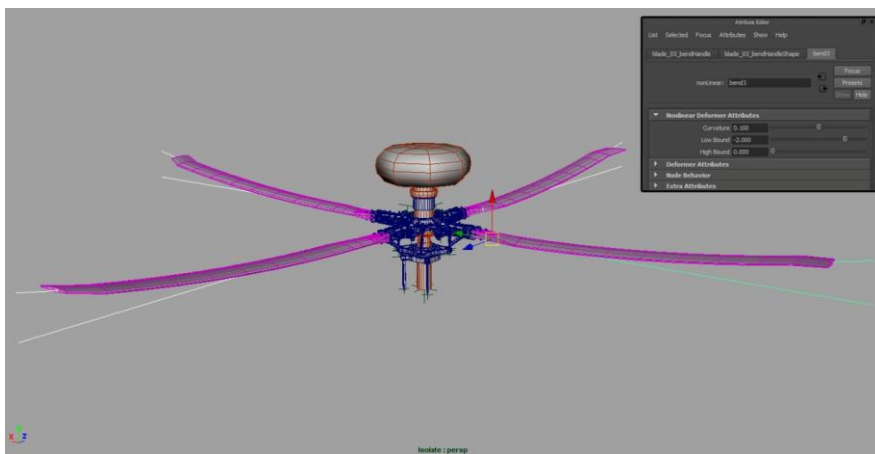
Roottorin kaikki osat laitetaan yhden locatorin alle, jolla kallistetaan koko pääroottoria (Kuvio 13 alin thrust-locator keskellä). Seuraavaksi osat laitetaan kolmeen ryhmään: ei pyöriin, pyöriin ja osaan joka säätelee lapojen liikettä (Kuvio 13 ylähissi). Ylähissi pyörii roottorin mukana, mutta sitä ei laiteta pyörivien osien kanssa samaan ryhmään, koska sen täytyy pyöriä itsenäisesti z-akselilla. Ylähissin z-akselin rotaatiolla hallitaan lapojen liikettä, jolloin lavat kääntyvät päinvastaiseen suuntaan riippumatta roottorin pyörimisliikkeestä (bank). Hissin y-akselin translate-attribuutilla lavat kääntyvät samansuuntaisesti (lift). Hissin on asetettu 45 asteen kulmaan. Pyörivät osat laitetaan oman locatorin alle, jolla animoidaan roottorin pyörimisliikettä (Kuvio 13 ylhäällä keskellä revolution-locator).

Lapojen riggaaminen aloitetaan tekemällä kolme locatoria. Ylin locator lapojen juureen, johon parentoidaan kaikki lavan osat. Näin lapojen liikettä hallitsee locator. Toiset locatorit sijoitetaan työntövarren ylä- ja alapäähän. Locatorien rotaatiot on tärkeää asettaa oikein. Näiden locatorien välille tehdään joint-ketju, ylhäältä alas. Alempan jointtiin parentoidaan alavarsi. Luuketjuille tehdään IK-handle, jonka kahva on alavarren locatorissa. IK-handlelle laitetaan point-constrain, jonka target on alavarren locator. Ylävarren locatorille laitetaan point-constrain, jonka targetiksi valitaan sen kohdalla oleva joint. Alavarren locatoriin lisätään parent-constrain, jonka targetiksi valitaan ylähissi. Lapojen locatorille lisätään aim-constrain, joka katsoo ylävarren locatoria.

Saksien alaosa on mallinnettu pyöreällä nupilla, jotta se saa liikkua vapaasti. Saksien yläosa kääntyy vain yhdellä akselilla. Sakset rigataan samaan tapaan kuin varretkin. Aluksi tehdään kolme locatoria jokaiseen kääntyvään pivottiin. Tehdään joint-ketju ylhäältä alas IK-handlella. Saksien alaosa ei saa olla samassa ryhmässä ylähissin kanssa, koska muuten saksien alaosa ottaa rotaatiot siltä, eikä näin ollen liiku vapaasti. Saksien alaosan locator laitetaan kiinni ylähissin parent-constrainilla.

Ylähissin y-translate- ja z-rotate-attribuutit määrätään alahissin hallintaan parent-constrainilla. Ylähissi ottaa y-rotaten ylimmältä locatorilta, jolla pyöritettiin roottoreita (Kuvio 13 ylin locator keskellä). Alahissi on pyörimätön ja liikkuu y-translatella ja z-rotatella. Alahissin kuuluu kolme vartta, jotka katsovat alas aim-constraineilla. Lapojen kääntymistä hallitaan nyt alahissin yhdellä locatorilla.

Lavat täytyy vielä rigata niin, että ne taipuvat sekä ylös että alas. Tämä on helpoin tehdä Bend-deformerilla, joka löytyy valikosta: Animation/Create deformer/Nonlinear/Bend. Jokaiselle lavalle tehdään oma Bend-deformer, jonka pivotti tulee lavan juureen. Bend käännetään rotate-työkalulla oikeinpäin niin, että bendin curvature attribuutti taivuttaa kaikkia lapoja samalla tavalla. Lapa on mallinnettu etukäteen moneen segmenttiin, jotta taipuma on mahdollinen. Bendiltä löytyy kaksi attribuuttia: low- ja high-bound, joilla säädetään pivotista lähteviä kurvin pituuksia. Tässä tapauksessa tarvitaan vain toista kurvia, joten riippuen bendin suuntauksesta toinen boundeista voidaan asettaa nolnaan. Toinen boundi voidaan laittaa maksimiin, jotta koko lapa taipuu.



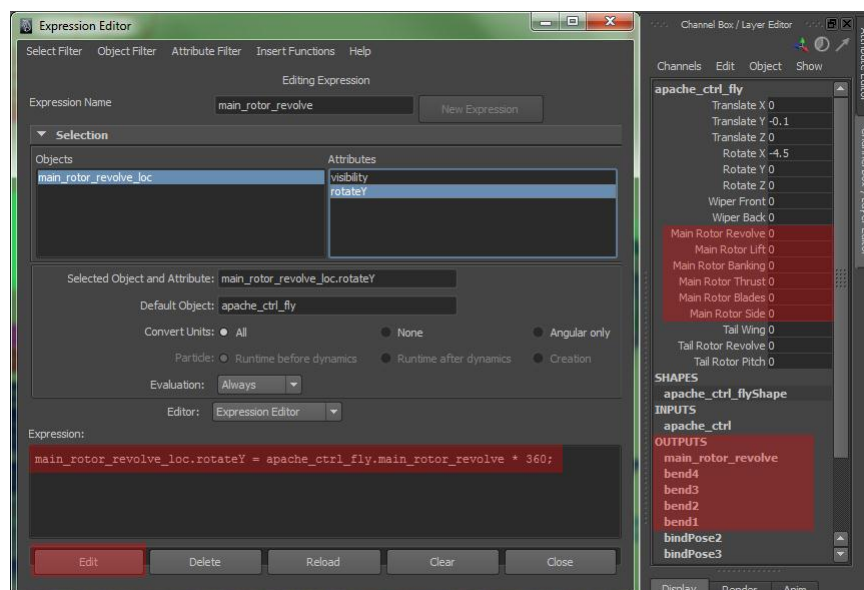
Kuvio 14. Jokaisella lavalla on oma bend-deformer

Viimeisenä siirretään roottorin kaikki toiminnot Fly-kontrollerille. Tarvitaan kuusi uutta attribuuttia: `main_rotor_revolve` (pyörittää roottoria), `_thrust` (kallistaa roottoria eteen ja taakse), `_side` (kallistaa roottoria sivuille), `_lift` (kääntää lavat samaan suuntaan), `_banking` (kääntää lavat eri suuntaan) ja `_blades` (taivuttaa lavat).

Main_rotor_revolve-attribuutille ei laiteta maksimiarvoa, mutta minimi voi olla 0, jos roottoria ei haluta pyörittää väärään suuntaan. Uusi attribuutti toimii parhaiten jos arvo 1 pyörittää roottoria täydet 360-astetta. Tämä on hyvä tapa tehdä pyöriviä objekteja ettei animaattorin tarvitse käyttää tuhatlukuja objektien pyörittämiseen. Tätä varten tarvitaan expressionia. Expressioniin pääsee käsiksi valitsemalla uusi attribuutti ja valitsemalla Channel Boxin valikosta Edit/Expressions. Ilmestyvään ikkunaan kirjoitetaan scripti:

```
main_rotor_revolve_loc.rotateY = apache_ctrl_fly.main_rotor_revolve * 360;
```

Ensimmäinen lauseke viittaa `main_rotor_revolve_loc` objektin y-akseliin. `=`-merkki tarkoittaa yhtäkuin tai samaa. Seuraava lauseke viittaa `apache_ctrl_master`-objektin `main_rotor_revolve`-attribuuttiin, jonka arvo kerrotaan 360:lla. `;`-merkki päättää lauseen, muulloin lause voisi jatkua toiselle riville. Käytännössä ensimmäinen lauseke saa arvon toiselta lausekkeelta. Edit-nappi tallentaa expressionin attribuuttiin. Muihin editorin asetuksiin ei tarvitse koskea.

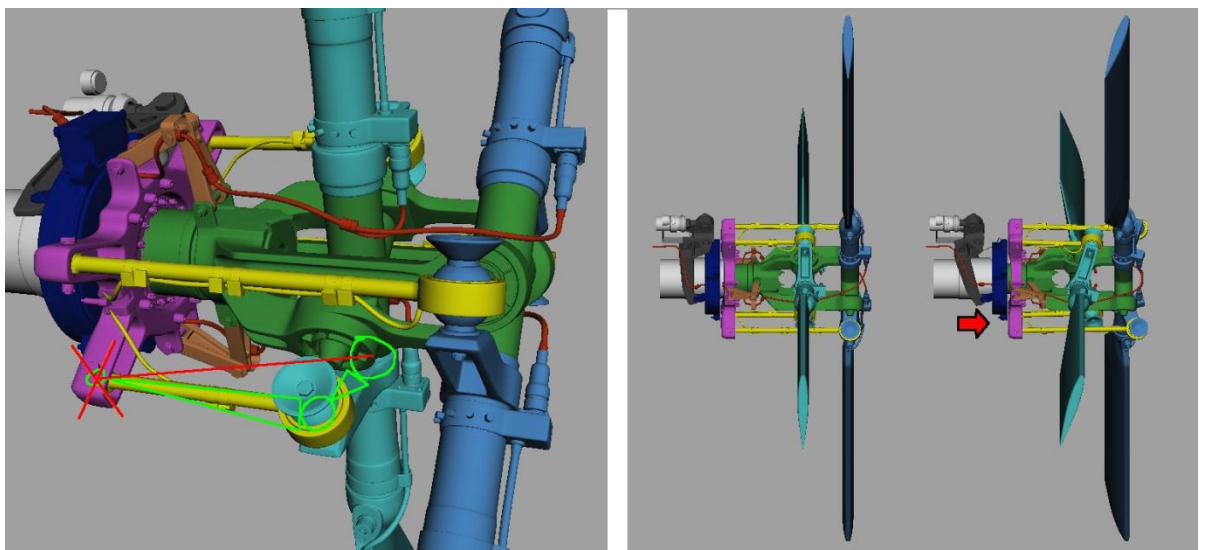


Kuvio 15. Vasemmalla Expression Editor. Oikealla valmiit attribuutit ja niiden outputit.

Loput roottorin toiminnot tehdään Set Driven Key -editorilla samaan tapaan kuin aikaisemmin tuulilasinpyyhkimissä. Roottorin eri osien astekulmat riippuvat rakennetusta mallista. Animaattorin on mahdollista saada lavat mahdollisiin asentoihin nykyisellä rigillä jos lift- ja banking-attribuutit laittaa ääriasentoihin. Rigin jatkokehittelyllä näihin on mahdollista tehdä expressionilla rajoitukset.

5.5 Takaroottori

Takaroottoriin tarvitaan kahta erilaista liikettä. Toinen pyörittää roottoria akselinsa ympäri (revolve) ja toinen kääntää lapoja (pitch). Lapoja käännetään samaan tapaan kuin pääroottorissakin. Kuvio 16:sta tummansininen (ei pyöri) ja pinkki (pyörii) rakenne työntää keltaisia varsia eteenpäin ja näin taivuttaa lapoja. Oranssit "sakset" on kiinnitetty pinkkiin "hissiin" ja vihreään osaan, joka ei työnny eteenpäin. Tummanharmaan osan tehtävä on työntää "hissiä" eteenpäin.

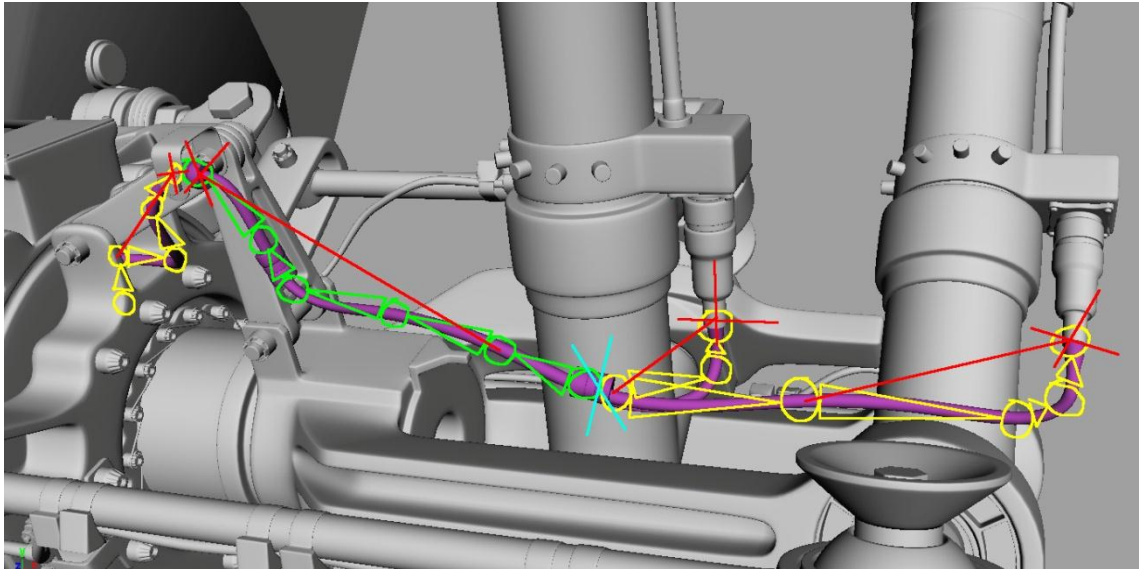


Kuvio 16. Takaroottorin toiminta

Roottorin riggaaminen ei eroa oleellisesti pääroottorin riggaamisesta, joten sen rakentamista ei ole tarpeen käydä läpi. Yksi uusi asia on lapoihin kiinnittyvät letkut.

Takaroottoriin tulee letku joka seuraa lapojen liikettä. Letkuihin tehdään kolme eri joint-ketjua, joihin tulee IK-handlet letkun kiinnityskohtiin. IK-handlet kiinnitetään parentoimalla tai parent-constrainteilla. Joittiketjujen väliin tehdään locator (Kuva 17

turkoosi risti), joka laitetaan kahdella point-constraintilla IK-handleihin. Tämä saa myös keskikohdan liikkumaan tasaisesti kahden kohteen välillä. Itse polygon-letku skinnataan luihin. Ensiksi katsotaan letku-objektin transform-attribuuttia, josta otetaan Inherits Transform pois päältä (Attribute Editor). Ottamalla edellä mainittu raksi pois, skinnattu objekti voidaan parentoida muihin objekteihin ilman että objekti yrittää liikkua omilla attribuuteilla. Tämä valinta voi siirtää objektin eri paikkaan, jolloin se siirretään käsin paikoilleen. Skinnaus onnistuu näin yksinkertaiselle objektille valitsemalla polygon-letku, joint-ketju, ja ottamalla valikosta: Animation/Skin/Bind Skin oletusasetuksilla. Nyt letkun polygon-verkon pitäisi seurata joint-ketjun liikkeitä. Skinnatun objektin Historiaa ei saa poistaa normaalisti, koska se poistaa myös skinnauksen. Tämän vuoksi skinnatulle objektille voidaan käyttää eri toimintoa, joka löytyy: Edit/Delete By Type/Non-Deformer History.



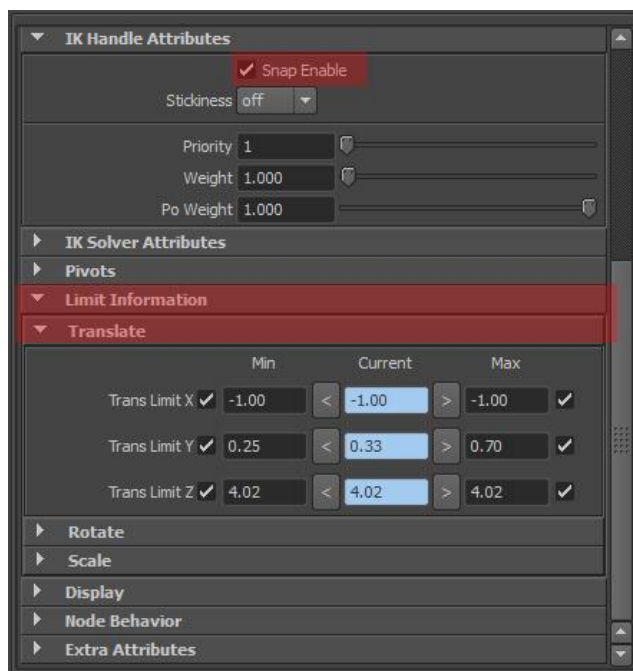
Kuvio 17. Taipuvan letkun rakenne

5.6 Renkaat ja iskunvaimentimet

Helikopteriin tulee laskutelineet eteen ja taakse. Etummaisets telineets ovat identtiset joiden renkaat pyörivät vain yhdellä akselilla. Takarenkaalla käännetään helikopterin suuntaa sen rullatessa maalla. Laskutelineillä on iskunvaimentimet ja pehmeät renkaat. Tavoitteena on tehdä laskutelineets, joiden iskunvaimentimet ja renkaiden lytistyminen toimivat automaattisesti helikopterin laskeutuessa. Rigi toimii siten, että animaattori päättää kohdan johon helikopteri laskeutuu tai josta se nousee. Tähän kohtaan

animaattori siirtää kontrollerin. Tätä varten tehdään Master-kontrolleri maan pinnan tasolle. Rigi ei ole parhaimmillaan kohtauksissa, jossa helikopteri laskeutuessaan pomppisi ja liikkuisi eteenpäin. Takalaskutelineen riggausta ei käydä läpi, koska sen toimintaperiaate on samankaltainen etulaskutelineiden kanssa.

Aluksi laitetaan eri osat omiin ryhmiinsä ja luodaan locatoreita ryhmien pivotteihin. Ryhmät pitää miettiä renkaan osalta niin, että vanne ja rengas pyörivät mutta jarrut eivät. Renkaan alle keskelle tehdään uusi kontrolliohjekti. Kontrolliohjektiä ei parentoida Fly-kontrolleriin, jotta se ei tule lentäessä mukana. Telineettä varten tehdään yhden jointin ketju IK-handlella. IK-handlelle laitetaan Snap Enable -raksi päälle, jotta IK-handle ei veny. Lisäksi laitetaan translatelle rajoitukset, jotta teline liikkuu vain y-akselilla sallitun verran. IK-ketju liitetään Point-constraintilla renkaan kontrolliohjektiin (Kuvio 18 siniset arvot).



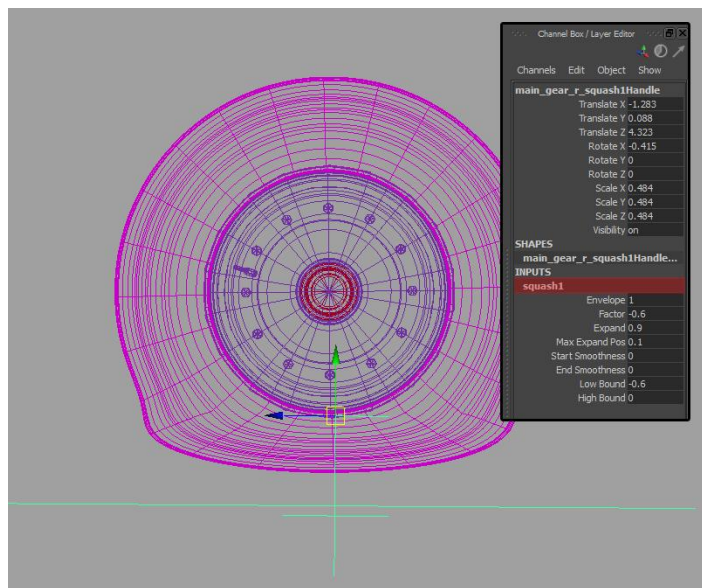
Kuvio 18. Laskutelineen IK-handle-asetukset

Renkaan pyöriminen tehdään automaattiseksi, kun animaattori liikuttaa Master-kontrolleria eteen tai taakse. Tämä saadaan aikaan expressionilla, joka laitetaan rengasta pyörittävään attribuuttiin. Tässä tapauksessa locatorin x-akseliin:

```
tire_revolve_loc_01.rotateX = apache_ctrl_master.translateZ * 80;
```


Toisen lausekkeen translateZ arvo kerrotaan 80:llä, jotta rengas pyörii oikealla nopeudella. Arvo riippuu täysin mallinnetusta renkaasta, joten arvo voi olla muutakin. Huomion arvoista on se, että jos helikopteria käännetään 180 astetta ja helikopteri rullaa toiseen suuntaan eivät renkaat osaa pyöriä oikeaan suuntaan. Expressionin jatkokehittelyllä tähän voisi tehdä tarkistuksen, joka katsoo, miten päin Master-kontrolleri on ja tehdä laskutoimitus sen perusteella.

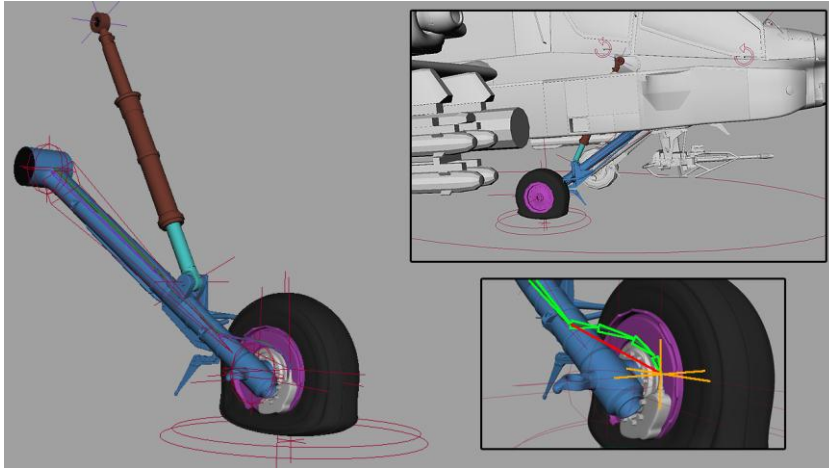
Rengas saadaan ruttuun Squash-deformerilla (Animation/Create Deformers/Nonlinear). Deformerin asetuksista Envelopella saadaan squashin vaikutus päälle ja pois. Factor säätää pullistumista, jota Expandilla voi laajentaa. Max Expand Pos:lla voi keskitason korkeuttaa säätää. Smooth pehmentää mahdollisia teräviä kulmia. Low ja High Bound:lla säädetään squashin vaikutusalueen kokoa. Deformer parentoidaan Master-kontrolleriin, jotta se jää maahan.



Kuvio 19. Squash-deformer

Jarruihin tulee letku jolle tehdään jointtiketju ja IK-handle. IK-handlea ei laiteta aivan ylimmästä jointista, koska jarruletkun tulee taittua vain alareunasta kun rengas liikkuu. Jarruletkun polygon-objekti kiinnitetään jointteihin skinnaamalla. Jointtiketju voidaan parentoida laskutelineen varteen ja IK-handle jarruun.

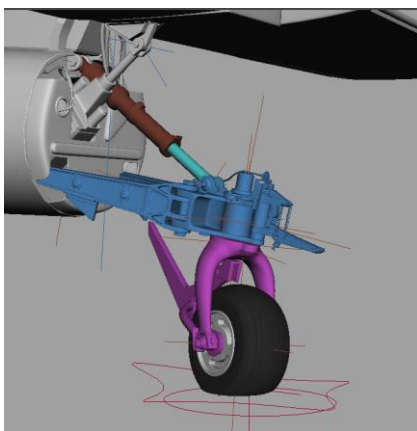
Laskutelineisiin kuuluu myös iskunvaimennin, joka rigataan aim-constrainteilla samaan tapaan kuin aiemmin ovipumput.



Kuvio 20. Valmis laskuteline. Oikealla alhaalla jarruletkun rakenne.

Tämän suuntainen rigi on suunniteltu toimimaan siten, että animaattori tietää mistä kohdasta helikopteri nousee tai laskee. Laskuteline voidaan rigata dynaamisemmaksi niin, että laskupaikkaa ei tarvitse päättää. Tämä voitaisiin tehdä käyttämällä parent-constrains renkaiden kontrolliin ja squash-deformeriin, joiden targettina olisi lentokontrolli. Renkaiden osuminen maahan voitaisiin toteuttaa siten, että parent-constrainsista otettaisiin weight pois päältä renkaan osuessa maahan. Weightin säätäminen voitaisiin toteuttaa käsin uudella attribuutilla tai automaattisesti expressionilla.

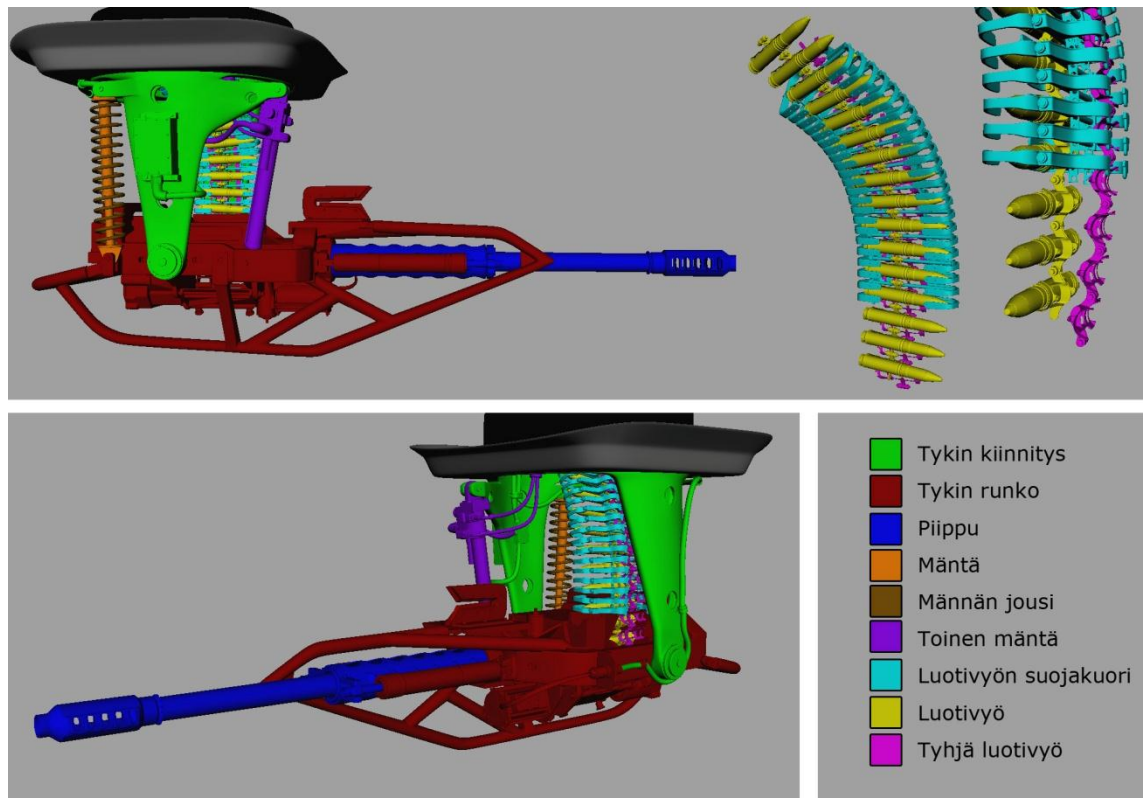
Tarvittaessa renkaiden ja jousituksen yhteisrakenne voitaisiin toteuttaa esim. Steve Hughesin tutoriaalin mukaan, jossa hän opettaa video-oppaiden avulla jousituksen rakentamisen alusta loppuun. (Hughes 2007, Portfolio)



Kuvio 21. Takalaskutelineen rigi valmiina

5.7 Aseistus

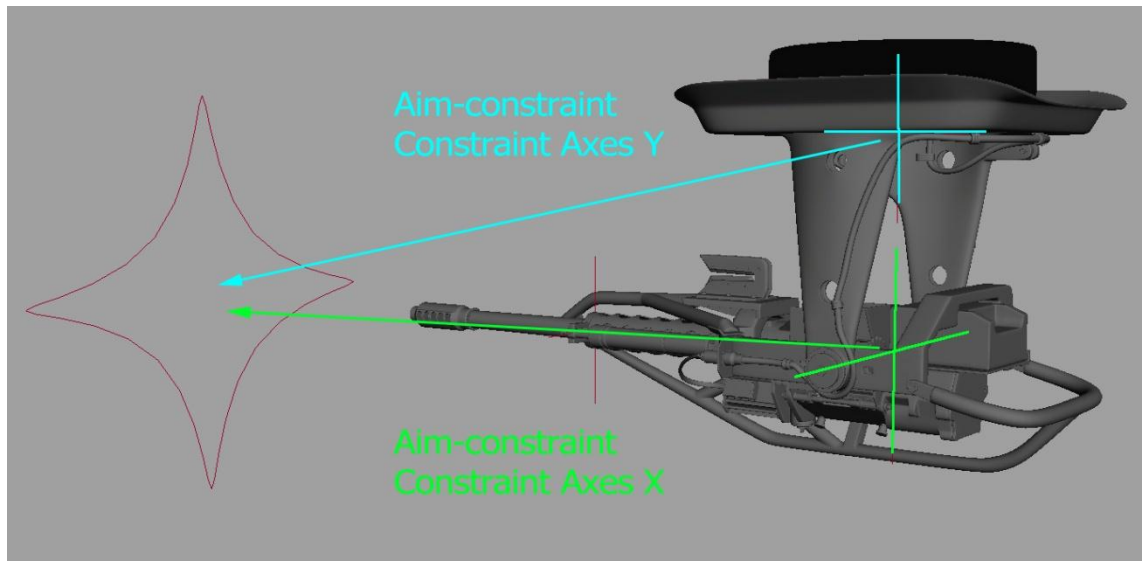
Helikopterille tehdään kahta erilaista aseistusta. Työläin osuus on tykki, joka sijoittuu helikopterin keulaan. Tykkiin olennaisena kuuluu myös keulan kameralaitteisto, joka laitetaan katsomaan tykin kohdetta. Toisena helikopterin "siipiin", eli asetelineisiin tulee molemmin puolin raketteja ja ohjuksia. Näitä ei ole tarpeen rigata muuten kuin laittamalla lentäviin osiin locatorit, joista niitä animoidaan. Ohjuksia ei parentoida normaalisti lentävään helikopteriin, koska ne täytyy voida irroittaa helikopterista lennon aikana omiksi objekteikseen. Tämän takia ohjukset täytyy laittaa kiinni runkoon parent-constrainteilla, joista weight laitetaan nolnaan, kun ohjus irroitetaan. Ohjusten lentorata jätetään animaattorin hoidettavaksi, joka voidaan säätää esimerkiksi CV-Curveilla ja Motion Path -työkalulla.



Kuvio 22. Tykin rakenne

Tykki toimii parhaiten jos sille tehdään kohde jota se seuraa. Tätä varten tehdään piipun eteen uusi kontrolliohje. Tarvitaan myös kaksi locatoria, joista toinen kääntää tykin kiinnitystä Y-akselilla ja toinen itse tykkiä X-akselilla. Locatorit katsovat

kontrolliohjetta Aim-constrainteilla, joiden asetuksista valitaan oikeat Constraint Axes:t.

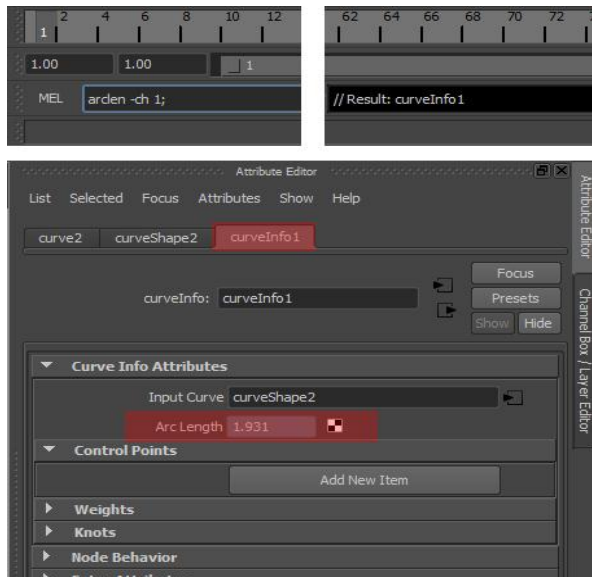


Kuvio 23. Tykin suuntaus kohteeseen

Ammusvyön tekeminen aloitetaan riggaamalla sen suojakuori. Suojakuori on ase-
liikkeen mukana taipuva ketju, jonka sisällä itse ammuksset liikkuvat. Ensimmäisenä
tehdään CV-Curve (Create/CV-Curve Tool) ja piirretään kurvi siihen kohtaan, jossa
suojakuoren pitäisi olla. Tässä tapauksessa riittää neljän pisteen CV-Curve. Seuraavaksi
on hyvä tietää CV-Curven pituus, jotta tiedetään rakentaa sille oikean mittainen
suojakuori. Mitataan CV-Curven pituus valitsemalla kurvi ja kirjoittamalla pieni MEL-
scripti Mayan vasemman alanurkan alueelle:

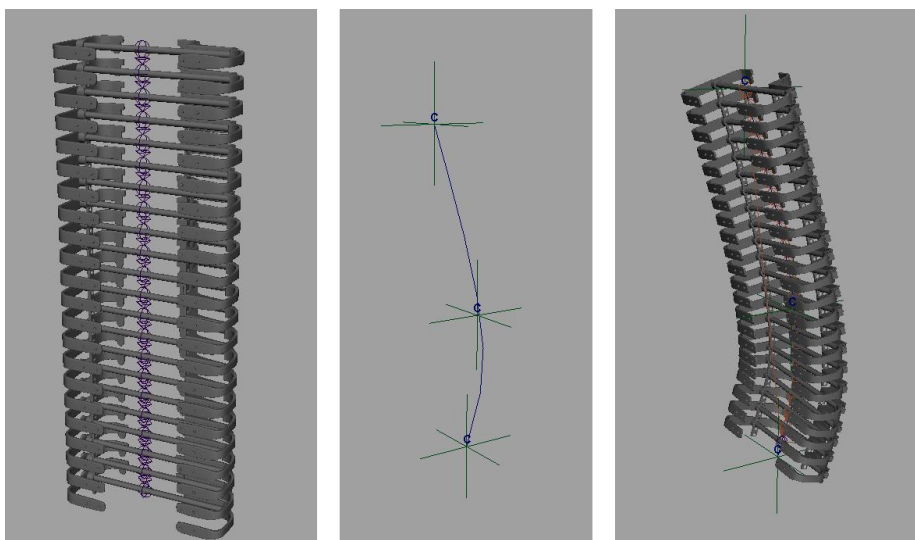
```
arclen -ch 1;
```

Enterin painallus suorittaa scriptin. Jos scripti onnistui, siitä tulee ilmoitus viereiselle
mustalle alueelle: //Result: curveInfo 1. Scripti tekee CV-Curvelle uuden curveInfo-
noden. Attribute Editorin curveInfo 1 -välilehdestä kohdasta Arc Length näkyy kurvin
pituus.



Kuvio 24. Ylhäällä scriptin kirjoitus ja alhaalla uusi node Attribute Editorissa

CV-Curvelle täytyy vielä tehdä kolme kontrollipistettä joista sitä voidaan taivuttaa. Tätä varten valitaan CV-Curvesta pisteitä valitsemalla CV-Curve ja hiiren oikean napin valikon kautta Control Vertex. Valitaan ylin Control Vertex ja valikosta Animation/Create Deformers/Cluster luodaan tälle vertexille cluster-niminen objekti, josta vertexiä voidaan liikuttaa. Kahdelle keskimmaiselle CV-Curven vertexille tehdään yksi yhteinen cluster ja alimmalle oma. CV-Curven käsittely on mielekkäämpää, jos sille tehdään vielä kolme locatoria, jotka sijoittuvat clustereiden päälle. Clusterit parentoidaan locatoreihin, jotta niiden kontrolli saadaan locatoreille.

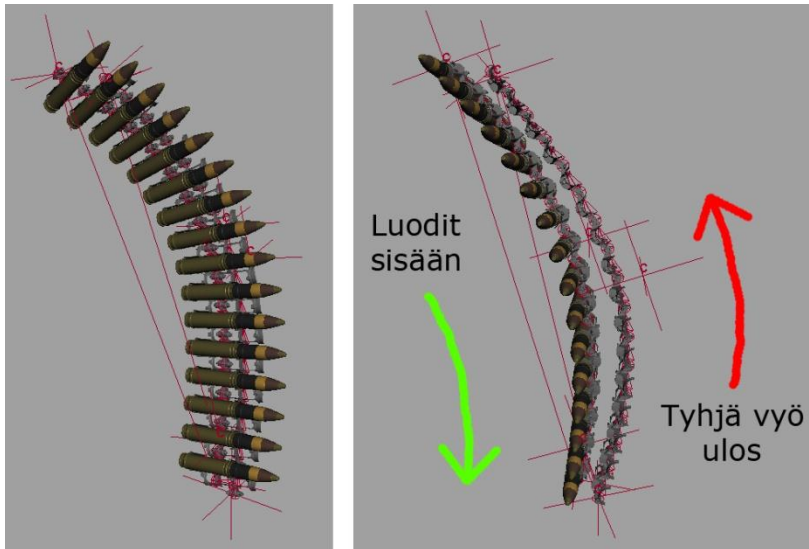


Kuvio 25. Suojakuoren eri rakennusvaiheet

Kun tiedossa on CV-Curven pituus, voidaan rakentaa saman pituinen suojakuori. Tätä varten on mallinnettu yksi pala suojakuorta, jonka keskelle taipumakohtaan tehdään yksi joint. Tätä yhtä palaa jointin kanssa kopioidaan niin monta kertaa kuin CV-Curven pituuteen tarvitaan. Mitta-apuna voidaan käyttää esim. samanpituista Polygon Cubea ja kopioimiseen Edit/Duplicate Special -valikon toimintoa. Jointit parentoidaan toisiinsa, jotta niistä tulee ketju. Samoin jokainen suojakuoren pala parentoidaan kohdalla olevaan jointtiin. Suojakuori saadaan kiinnitettyä CV-Curveen IK Spline Handle Toolilla. Työkalun asetuksista otetaan raksi pois kohdasta: Auto Create Curve - tällöin työkalu ei luo uutta curvea. Työjärjestys menee valitsemalla työkalu, alku joint, loppu joint ja CV-Curve.

Viimeinen vaihe on liittää suojakuori tykkiin. Aluksi tehdään yksi locator suojakuoren yläpuolelle kohtaan johon ketju kiinnittyy (nimetään: belt_mount_locator). Belt_mount_locator parentoidaan locator:iin joka pyörittää koko asetta Y-akselilla (tykin kiinnitys, Kuvio 22). Suojakuori parentoidaan kokonaisuudessaan aseeseen, jotta se pyörii ja taipuu tykin liikkeen mukana. Suojakuoren ylin locator saadaan tarttumaan uuteen belt_mount_locator:iin parent-constraintilla niin, että asetuksista laitetaan weightiä 0.6. Weightin määrä riippuu mallinnetusta suojakuoresta, tykistä ja ympäröivästä rakenteesta. Näin saadaan ketju toimimaan niin, että se ei osu reunoihin ja että se seuraa hieman tykin liikettä, mutta pysyy silti osittain ylhäällä kiinni. Belt_mount_locator:lle laitetaan vielä point-constraint y-akselille ja targetiksi tykkiä liikuttava locator. Tämä parantaa entisestään suojakuoren liikettä tykin katsoessa alas.

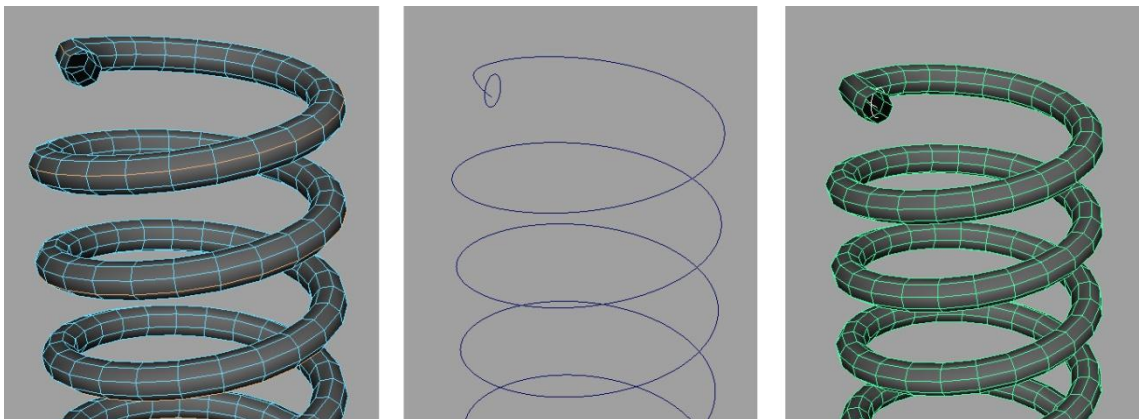
Itse ammusvyö tehdään samalla tavalla kuin suojakuori. Helpoin tapa tehdä tämä on tehdä kaksi erillistä ketjua, joista toinen vie mukanaan ammuksia kohti asetta ja toinen vie tyhjää vyötä ylös kohti helikopterin runkoa. Vyöketjun liikettä animoidaan Spline IK handlen Offset-attribuutilla, joka liikuttaa ketjua splinellä, mutta vain yhteen suuntaan, joka pitää huomioida ketjua rakentaessa. Offsetin hallinta siirretään kontrolliobjektille, joka hallitsee aseella ampumista. Ampumista varten tehdään uusi attribuutti, johon ampuminen animoidaan. Ampumis-attribuutin arvolla 0.5 ase laukeaa ja vyö liikahtaa pykälän verran. Arvolla 1 ase on suorittanut koko ampumisen. Vyön palautuessa normaaliin asentoon Offset animaation kurvi täytyy olla Step, jotta palautumisliikettä ei näy. Vyö ei siis liiku koko mittansa, vaan pelkästään luo illusion liikkumisesta.



Kuvio 26. Ammusvyön rakenne

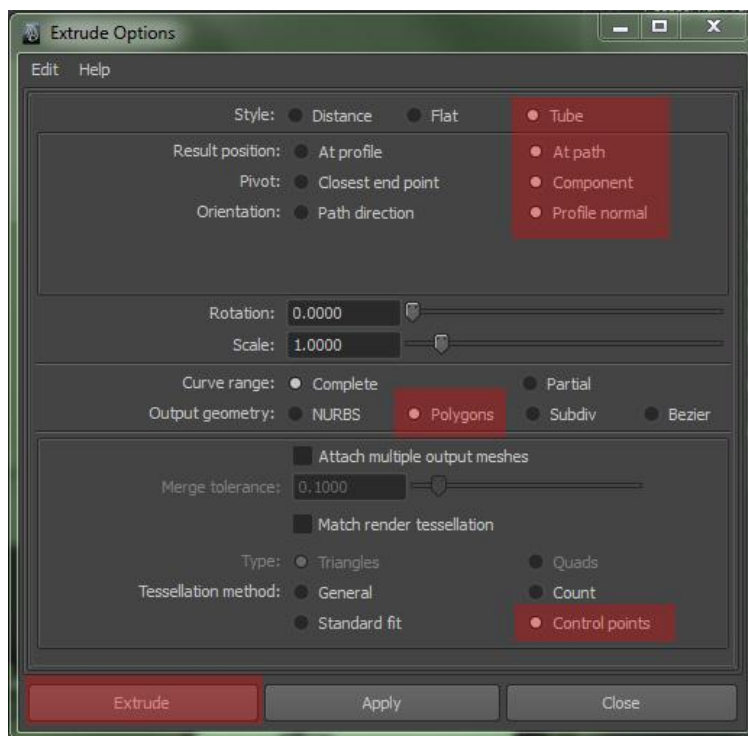
Tykin takaosassa on mäntä, jonka päälle tulee jousi. Mäntä voidaan rigata normaaliin tapaan aim-constrainteilla. Jousta varten tarvitaan Nurbs-Helix, jota ei oletuksena löydy Maya 2011:sta. Nurbs-Helix luodaan käyttämällä apuna Polygon-Helixiä. Polygon-Helix ei käy rigiin sellaisenaan, koska jousen pitää säilyttää muotonsa sitä skaalatessa.

Aluksi luodaan oikean kokoinen Polygon-Helix (Create/Polygon Primitives/Helix). Tämän jälkeen valitaan Polygon-Helixiltä yksi Edge-rivi ja sitten Modify/Convert/Polygon Edges to Curve. Tämä luo uuden Nurbs-käyrän, josta luodaan uusi Polygon-Helix. Vanha Polygon-Helix voidaan poistaa.



Kuvio 27. Polygon-Helixistä luodaan Nurbs-Helix, josta taas Polygon-Helix

Uudesta Nurbs-Helixistä luodaan uusi Polygon-Helix josta tulee lopullinen jousi, joka näkyy mm. rendatessa. Tehdään ensin uusi Nurbs-Circle, joka on jousen paksuinen. Valitaan Nurbs-Circle ja Nurbs-Helix ja valikosta Surfaces/Surfaces/Extrude Options laitetaan seuraavat asetukset:

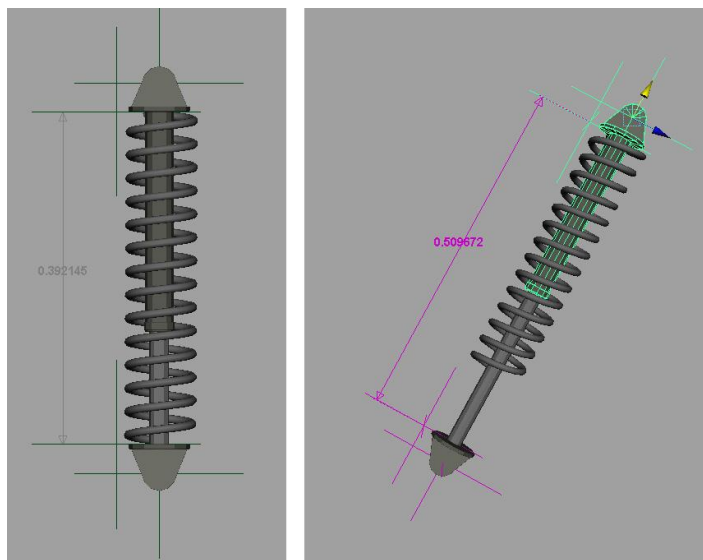


Kuvio 28. Nurbs-käyrästä muutetaan polygon-objekti

Seuraavaksi Nurbs- ja Polygon-Helixiltä poistetaan Historia, jotta ne eivät ole enää yhteydessä toisiinsa. Tätä voidaan testata liikuttamalla Nurbs-Helixiä, eikä Polygon-Helixin pitäisi seurata perässä. Tällä työtavalla saatiin myös Nurbs-Helix tarkasti Polygon-Helixin sisälle.

Polygon-Helix saadaan seuraamaan Nurbs-Helixiä Wire Toolin avulla, joka löytyy valikosta: Animation/Deformers/Wire Tool. Työkalun vihjeet opastavat valitsemaan ensin objektin jota deformoidaan: Polygon-Helixin ja painamaan Enteriä. Seuraavaksi valitaan deformoija: Nurbs-Helix ja painetaan Enteriä. Nyt Polygon-Helixin pitäisi venyä oikealla tavalla Nurbs-Helixiä skaalaamalla ilman että polygon-pinta muuttua näkyvästi muotoaan. Nurbs-Helix parentoidaan ylämäntään.

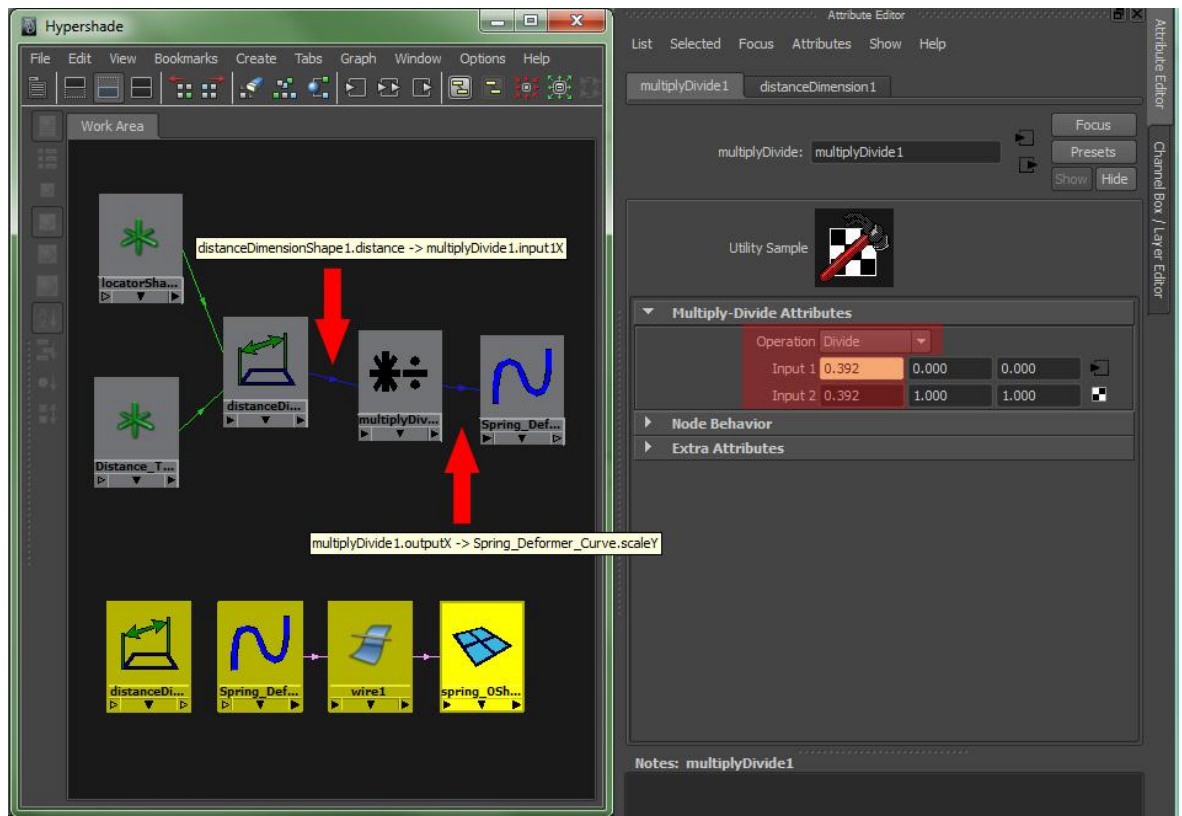
Seuraava tehtävä on saada jousi pysymään automaattisesti männän liikkeen mukana. Aluksi luodaan apuobjektit valikosta: Create/Measure Tools/Distance Tool. Tällä työkalulla luodaan kaksi locatoria jousen ylä- ja alapäähän, jonka väliltä mitataan etäisyys. Ylempi locator parentoidaan männän yläosaan ja toinen männän alaosaan.



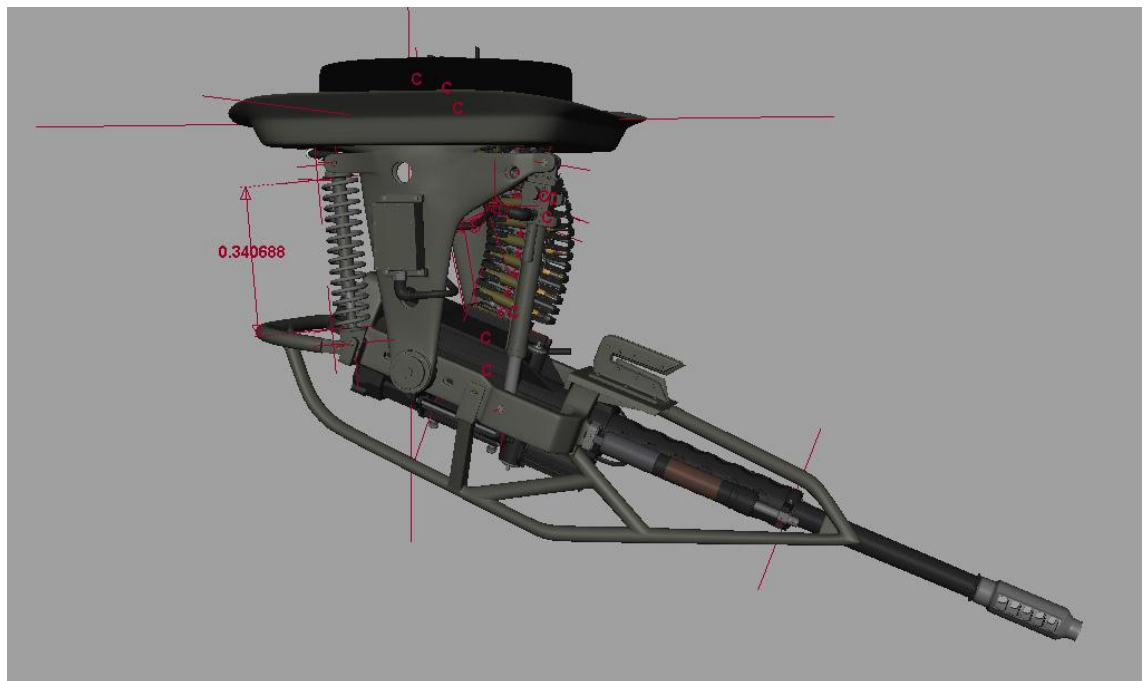
Kuvio 29. Distance Toolin locatorit parentoituna männän osiin. Jousi ei vielä veny.

Nurbs-Helix saadaan venymään Distance Toolin locatorien välillä jakamalla distanceDimension1-objektilta tuleva Distance-arvo ja antamalla se Nurbs-Helixin y-skaalaukselle. Valitaan Outlineristä distanceDimension1-objekti ja Nurbs-Helix, joka deformoi jousia. Mennään HyperShaderiin ja valitaan näkyviin Input and Output Connections. Tehdään uusi node valikosta: Create/General Utilities/Multiply Divide. Tehdään yhteys distanceDimensionShape1.distance -> multiplyDivide1.input1x. Näin saadaan siirrettyä etäisyyden mittajan arvo jakajaan. MultiplyDivide1-nodelta laitetaan Attribute-editorissa Operation pudotusvalikosta Divide päälle ja Input2:seen kopioidaan arvo Input1:stä. Lopuksi jakajalta viedään arvo Nurbs-Helixin y-skaalaukseen. Tehdään yhteys multiplyDivide1.outputX -> Spring_Deformer_CurveShape.ScaleY. Nyt jousi pysyy männän liikkeen mukana ja mäntä voidaan kiinnittää paikalleen.

Edellä mainittu esimerkki voidaan rakentaa sellaisenaan moniin ajoneuvoihin joilla on jousitus. Tällainen männän ja jousen rakenne on yleinen myös autoissa ja moottoripyörissä.



Kuvio 30. Jousen asetukset Hypershadessa



Kuvio 31. Valmis tykki

5.8 Riggauksen viimeistely

Rigi on nyt toimintavalmis, mutta riggaajan tulee vielä huolehtia, ettei animaattori riko rigiä vahingossa. Objektit tulee myös laittaa eri layereihin, joista kaikki locatorit, jointit ja IK-handlet voidaan piilottaa ja lukita. Lisäksi riggaajan on hyvä laittaa kaikki attribuutit lukkoon ja/tai piiloon kontrolliojekteista joita ei ole tarkoitus valita. Animaattorin on tarkoitus koskea vain kontrolliojektien attribuutteihin joita saa animoida. Layereitä ei myöskään kannata tehdä liikaa jos scenestä tulee iso ja se sisältää useita ajoneuvoja, taustoja, kameroita, valoja jne. Hyvä määrä tällaiselle ajoneuvolle voisi olla 2-3 Layeriä, joista yhteen tulee kontrolliojektit ja toisiin laitetaan muut osat.

Riggaajan tulee huolehtia rigatun objektin scenen siisteydestä. Scenen siisteyttä ei voi liikaa korostaa, koska pahimmillaan Maya kaatuu sotkuisiin malleihin, kun yhteen sceneen referoidaan useita erilaisia malleja. Maya kerää objekteihin lähes kaikista muutoksista Historiaa joka tulee poistaa, mutta ei kuitenkaan niin, että rigi rikkoutuu. Scenestä poistetaan turhat objektit, materiaalit, nodet ja layerit niin, että jäljelle jää vain rigattu esine tarvittavine osineen. Viimeisenä, apuna voi käyttää File/Optimize Scene Size -toimintoa, joka siivoaa automaattisesti valittuja osia.

Tämän lopputyön helikopteri ei ole hyvä sellaisenaan animaattorille. Helikopterissa on paljon yksityiskohtia ja se on raskas animoitava. Raskaille malleille mallinnetaan usein Proxy-malli. Proxy-malli on Lowpoly-malli, jossa on vain muutama sata polygonia. Se on laatikkomainen malli, joka on äärimmäisen kevyt animoitava ja jota voi käyttää esimerkiksi dynamiikan törmäystarkistuksissa. Animoitavassa Proxy-mallissa on samat toiminnot kuin Highpoly-mallissakin. Proxy-mallia ei tarvitse rigata monimutkaisesti uudestaan vaan sen eri osat voidaan parent-constraintilla liittää Highpoly-malliin. Proxy-animoiminen tapahtuu näin samoista kontrolliojekteista.

Lopuksi riggaaja testaa mallinsa animoimalla lyhyitä testianimaatioita, joilla on tarkoitus löytää mahdolliset virheet. Jos rigatussa mallissa on puutteita joita ei välttämättä tarvitse korjata, kuten mahdottomiin asentoihin kääntyviä esineitä, niin niistä on hyvä ilmoittaa animaattorille etukäteen ja kirjoittaa ne ylös Mayasta jokaiselta objektilta löytyvään Notes-kirjoitusalueeseen.



Kuvio 32. Valmis rigi

6 Yhteenveto ja pohdintaa

Lopputyön tavoitteena oli kirjoittajan oma oppiminen Mayan riggaamisessa ja se toteutui hyvin. Helikopterissa oli paljon erilaisia osia, jonka avulla riggaustyövälineet tulivat hyvin tutuiksi. Tavoitteena oli myös tehdä hyvä opas aloitteleville riggaajille.

Helikopterin rigi onnistui pääosin hyvin, mutta jatkokehittelyllekin jäi varaa. Riggaaminen tapahtui enimmäkseen kirjoittajan omien kokeilujen avulla ja ainoastaan muutamassa ongelmakohtassa apua tarvitsi etsiä internetistä löytyvistä oppaista. Internetin oppaita kannattaa muutenkin etsiä tällaisissa projekteissa, koska harvemmin löytyy sellainen rigi, jota ei olisi enemminkin tehty.

Suurin onnistuminen oli pääroottorin lapojen tekeminen. Niiden toiminta ei ollut aiemmin tuttua ja sen tutkimiseen kului aikaa. Toinen onnistuminen oli tykin tekeminen, joissa oli paljon erilaisia osia. Tämä työvaihe oli hyvin opettavainen ja sisälsi monia tekniikoita, joita pystyy käyttämään monissa teknisissä esineissä.

Jatkokehittelyä ajatellen rigi tarvitsisi monia rajoittimia, joiden avulla animaattori ei saisi laitettua eri osia mahdottomiin asentoihin. Helikopterin ovien rigaamisessa olisi pitänyt ajatella enemmän animaatiohahmoa, joka lopulta hallitsee ovien liikettä. Helikopterin siivissä oleva rakettiaseistus on nykyisellään kenties liian työläs. Tähän olisi voinut tehdä jonkinlaisen automaattisen käyttöliittymän, jotta jokaisen raketin parent-constraintia ei tarvitsisi säätää erikseen.

Pääosin lopputyöprojekti oli onnistunut ja kaikissa tavoitteissa onnistuttiin. Helikopteri tekee sen mitä sen tarvitsee, kaikki osat toimivat ja ovat riittävän helppoja animoida. Lopputyöhön tuli useita tekniikoita, joista voi soveltaa muihin 3d-malleihin ja kenties toisiin ohjelmiin. Mayan työkalut tulivat tutuiksi ja vastaaviin projekteihin on helppo tarttua jatkossa.

Lähteet

Wilkins, Mark R & Kazmier, Chris 2005. Mel Scripting for Maya Animators - Second Edition. San Francisco, CA, USA: Morgan Kaufmann.

Banks, Lester 2011. Robot rigging in Maya. Lesterbanks.com [verkkosivu]
< <http://lesterbanks.com/2011/04/robot-rigging-in-maya/> >
(luettu 26.4.2012)

Hughes, Steve 2007. Maya Rigging Video Tutorial: Piston and Spring. Portfolio [verkkosivu]
< <http://www.stevenhughes.dsl.pipex.com/tutorials.html> >
(luettu 26.4.2012)

Linkit rigin esittelyanimaatioon:

http://www.youtube.com/watch?v=vT_77FWr1Ec (hakusana: apache ah-64d rig)

<http://www.tonikontio.com/apache.mp4>

