



Case: Iron Sky

Tampereen ammattikorkeakoulu
Viestinnän koulutusohjelman opinnäytetyö
Visuaalinen suunnittelu
suuntautumisvaihtoehto
Kevät 2007
Lassi Aalto

OPINNÄYTETIIVISTELMÄ

Osasto Viestintä	Erikoistumisala Visuaalinen suunnittelu
Tekijä Aalto Lassi	
Työn nimi Case:Iron Sky	
Lopputyön laji Mediateko	
Työn valmistumisaika 07.06.07	Sivumäärä 36
<p>Tiivistelmä</p> <p>Tuotantoyhtiö Energia, pyysi minua avustamaan Iron Sky -elokuvan demon valmistamisessa, mallintamalla ja animoimalla siihen 3d:llä. Työ tapahtui keväällä 2007.</p> <p>Käyn työssäni läpi vaihteita, joita tarvittiin oman osioni aikaansaamiseksi.</p> <p>Opin tämän työn kautta paljon, ohjaajan kanssa työskentelystä, sekä yleisesti 3d-tuotantoprosessista.</p>	
Aineisto	
Asiasanat 3D, Mallinnus, animaatio, renderöinti	
Säilytyspaikka Tampereen ammattikorkeakoulu, Taide ja viestintä, Finlayson	
,Muita tietoja	

THESIS

SUMMARY

Department Media Programme	Area of specialisation Visual Design
Author Lassi Aalto	
Title Case: Iron Sky	
Sort of Final Thesis (Written / Project / Portfolio) Project	
Date 07.06.07	Number of pages 36
<p>Summary:</p> <p>Production house Energia, invited me to help them with the demo of their next feature Iron Sky, by modelling and animating with 3d. Work placed in spring 2007.</p> <p>In this thesis I will go trough work process i was required to do my share.</p> <p>With this project I learned how to work with director and 3d pipeline required in this kind of production.</p>	
Material (e.g. audio / video tape, photographs, slides, paintings, statues...)	
Key words 3D, Animation, Modelling, Rendering	
Filing Tampere Polytechnic, Art and Media, Finlayson.	
Other information	

Sisällys

1 Johdanto.....	4
2 Tarvittavat työvaiheet.....	5
2.1 Mallinnus.....	5
2.1.1 Polygonit.....	5
2.1.2 Nurbs.....	6
2.1.3 Subdivision Surface.....	6
2.2 Materiaalit.....	6
2.2.1 Color.....	7
2.2.2 Specular mapping.....	7
2.2.3 Luminosity.....	7
2.2.4 Transparency.....	8
2.2.5 Bump mapping.....	8
2.2.6 Displacement mapping.....	8
2.2.7 Normal mapping.....	9
2.3 Tekstuurit.....	9
2.4 Texture mapping.....	9
2.4.1 UV kartta (UV Map).....	10
2.4.2 3D-tekstuuri.....	10
2.5 Valaistus.....	10
2.5.1 Pistevalo (Point Light).....	11
2.5.2 Kohdevalo (Spotlight).....	11
2.5.3 Suuntaavavallo (Directional Light).....	11
2.5.4 Aluevalo (Area Light).....	11
2.5.5 Ympäröivävalo (Ambient Light).....	12
2.5.6 Kokonaisvalaistus (Global illumination).....	12
2.5.7 Radiositeetti (radiosity).....	12
2.5.8 Fotonikartoitus (Photon Mapping).....	13
2.5.9 Final Gathering.....	13
2.5.10 Caustics.....	13
2.6 Varjot.....	14
2.6.1 Syvyyskarttavarjot (Shadow map/ depthmap shadows).....	14
2.6.2 Säteenseuranta varjot (Raytraced Shadows).....	14

2.7 Riggaus	14
2.7.1 Forward / inverse kinematic	15
2.7.2 Luurangon ja hahmon geometrian liittäminen toisiinsa	16
2.8 Animaatio	17
2.9 Renderointi	18
2.9.1 Renderöinti kerrokset (Render Layer)	18
2.9.2 Renderöinti pyyhkäisyt (Render Pass)	19
2.9.3 Diffuse Passes	19
2.9.4 Specular Passes	19
2.9.5 Reflection Passes	20
2.9.6 Shadow Passes	20
2.9.7 Color Passes	20
2.9.8 Occlusion Passes	20
2.9.9 Beauty Passes	20
2.9.10 Mask Passes	20
2.9.11 Global Illumination Passes	20
2.9.12 Depth Passes	20
3 Työprosessi	21
3.1 Mallinnus	21
3.2 UV-mappaus	22
3.3 Mudbox	23
3.4 Materiaalit ja tekstuurit	23
3.5 Valaistus	24
3.6 Renderointi	24
Lähteet	28
Liitteet	29

1 Johdanto

Siihen aikaan keisari Pirk antoi käskyn. Kaikkien Second Brain oy 3D-osaajien oli tultava avustamaan Iron Sky-elokuvan demon teossa. Tämä avunpyyntö oli ensimmäinen ja tapahtui Mikko Monton ollessa Second Brainin toimitusjohtajana. Näin mekin menimme avustamaan, kukin omilla taidoillaan, omalle osiolleen.

Sama suomeksi. Tuotantoyhtiö energia kutsui meidät avustamaan uuden elokuvansa demon tekemisessä. Tarkoituksenani oli toimia mallinnusapuna, mutta lopulta päädyin tekemään enemmän kuin vain mallintamaan. Kerron nyt tästä prosessista, jossa mallinsin, teksturoin riggasin, animoin ja renderoin kohtauksen tähän teaser demoan.

Tämä kirjallinen osio on jaettu kahteen pääalueeseen, ensimmäisenä on yleinen teoriaosio, jossa kerron, mitä työhön tarvittavista työvaiheista. Toinen osio on kuvaus työprosessista, joka vaadittiin tähän mediatekoon. Jälkimmäisessä osiossa keskityn Iron Sky:n logona toimivan kotkan työstöön, koska työmääräni muissa kohtauksissa oli suhteessa paljon vähäisempää.

Mallinnus tapahtui perusmuodon luomisella entisen 2d-logo kuvan, innokkaan fanin tekemän pronssisen patsaan, sekä kotkan kuvien toimiessa referenssi-materiaalina.

Mallinnukseen käytin Autodesk Maya- ja Skymatter Mudbox- ohjelmia. Teksturointiin käytin edellä mainittujen ohjelmien lisäksi Adobe Photoshop CS3-ohjelmaa. Kaiken lopun työn kuten, valaistuksen, animoinnin ja renderöinnin tein Mayalla.

2 Tarvittavat työvaiheet

Aluksi käyn läpi teoriaa tarvittavista työvaiheista yleisesti, ja niiden vaikutuksista toisiinsa.

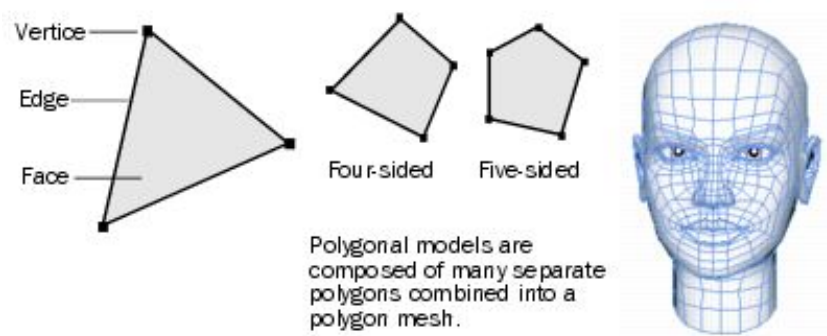
2.1 Mallinnus

Mallintaminen on 3D:n perusta, jonka päälle kaikki muu kasautuu. Mallinnettaessa luodaan kaikki objektit, joita kohtaukseen tarvitaan. Maya-ohjelmalla mallintamisen voi jakaa kolmeen kategoriaan mallinnustekniikoiden mukaan. Ne ovat polygoni-, nurbs- ja subdivision surface mallintaminen.

2.1.1 Polygonit

Polygonit ovat suorareunaisia kuvioita, jotka koostuvat kolmesta tai useammasta sivusta, jotka määräytyvät kolmiulotteisessa avaruudessa olevien pisteiden (vertex) ja niitä yhdistävien suorien viivojen (edge) mukaan. Polygonin sisäaluetta kutsutaan nimellä face. Polygonien muokkaaminen tapahtuu muokkaamalla sen peruskomponentteja joita ovat vertex, egde ja face.

Kolmisivuisia polygoneja kutsutaan triangeleiksi, nelisivuisia taas nimellä quad (quadrilateral). Tätä useampi sivuisia kutsutaan nimellä n-gon, mutta niitä ei suositella käytettäviksi, koska ne saattavat aiheuttaa ongelmia eri tekstuuriin kanssa tai konvertoitaessa pintaa toisenlaiseksi pinta-tyypiksi.



Kuva1 lähde Maya 8.5 Help files

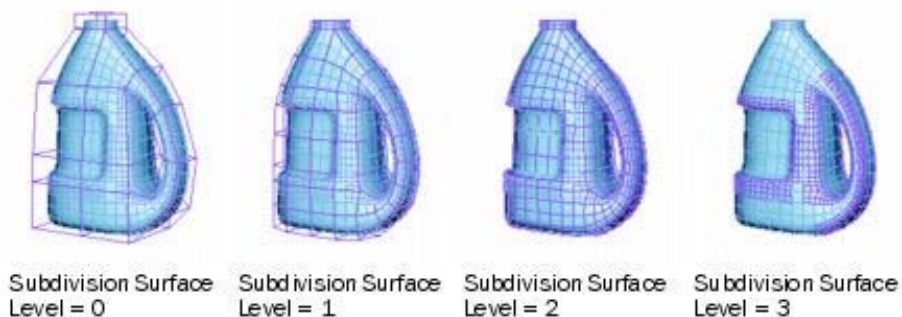
2.1.2 Nurbs

Non-Uniform Rational B-Splines. Matemaattisesti rakentuva käyrä tai pinta (Lehtovirta 2000, 210). Nurbs-pinnat poikkeavat polygonipinnoista erityisesti niiden kaarevuuden vuoksi. Nurbs-pinnoilla saa helposti aikaan luonnollisen kaaren huomattavasti pienemmällä määrällä pisteitä, kuin polygoni-pinnoilla, minkä vuoksi ne ovatkin suosittuja orgaanisessa mallinnuksessa.

2.1.3 Subdivision Surface

Mayan subdivision surface yhdistää sekä nurbs että polygon mallinnuksen parhaita puolia. Kuten nurbs-pinnat tuottavat myös subdivision pinnat kaarevan ja pehmeän muodon pienellä määrällä pisteitä. Subdivision pintaa voi puolestaan muokata kuten polygoneja, eli muokkaamalla peruskomponentteja. Subdivision pinnalla voi olla eri tarkkuudella yksityiskohtia eri alueilla pintaa, joista vähemmän kontrollipisteitä löytyy tasaisilla pinnanalueilla ja enemmän monimutkaisemmillä muodoilla.

Subdivision pintaa käyttämällä voi välttyä nurbs-pintojen ongelmilta, kuten repeilyltä, mitä ilmenee nurbs pintojen liikkuesssa.



Kuva2 lähde Maya 8.5 Help files

2.2 Materiaalit

Valmis mallinnettu malli näyttää harmaalta sileäpintaiselta muoviköntiltä, jotta siitä saisi realistisemman täytyy siihen lisätä tekstuurit ja materiaalit. Reaalimaailmassa, valon osuessa kappaleeseen osa siitä heijastuu ja osa imeytyy. Mitä sileämpi pinta on, sitä kiiltävämpi se on ja mitä karkeampi pinta on, sitä mattapintaisempi se on. Mayassa materiaalilla määritetään miten malli käyttäytyy siihen osuvan valon kanssa.

Tekstuureilla ja materiaaleilla luodaan vaikutelma mallin pintamateriaalista, joka kertoo katsojalle mistä aineesta se on tehty. Tekstuurit ovat erityisen tärkeässä asemassa tietokonepelien malleissa, joissa rajoitettu polygonien lukumäärä vähentää yksityiskohtien määrää. Tällöin teksturointi onkin usein se osa-alue, joka joko tekee tai ei tee mallista uskottavaa. Hyvä malli keskinkertaisilla tekstuureilla ei usein näytä hyvältä, kun taas heikompi malli hyvillä tekstuureilla voi olla erittäin onnistunut (McKinley 2005, 58).

Tekstuureilla voidaan kontrolloida monia pinnan eri ominaisuuksia, joiden avulla saadaan renderöitäessä erilaisia efektejä aikaan. Birnin (2002, 200) mukaan seitsemän yleisintä tekstuuria ovat color, specular, luminosity, transparency, bump, displacement ja normal. Nämä määrittävät kappaleen värin, kirkkaimmat kohdat, itsevalaisun, läpinäkyvyyden ja pinnan kuhruraisuudet.

2.2.1 Color mapping

Color mapping tunnetaan myös nimellä diffuse mapping, se korvaa mallin pinnan päävärin tekstuurilla. Color map säätelee pinnan hajoavan valon heijastumisen sävyä ja voimakkuutta. Joissain ohjelmissa Color ja Diffuse mapping on erikseen, jolloin color mapin sävyt kerrotaan diffuse mapin sävyillä.

2.2.2 Specular mapping

Specular mapping määrittää mitkä osat kappaleesta heijastavat kirkkaimmin valoa. Varsinainen valon heijastus syntyy kuitenkin valon ja kappaleen keskinäisen kulman mukaan, eikä specular mappingilla voida sitä määrittää. Specular mappingilla voidaan vaikuttaa heijastuvan valon sävyyn, kirkkauteen tai sen voi blokata kokonaan. (Birni, 2000, 206.) Vaaleat alueet specular mapissa tekevät heijastukset kirkkaammiksi, kun taas tummemmat alueet tekevät valon heijastumisesta vähemmän näkyvää. Specular mapin vaikutukset voidaan havaita vai alueilla, jossa tämä valon heijastuminen tapahtuisi ilmeikkain specular mappia.

2.2.3 Luminosity

Tunnetaan myös nimillä *incandescence*, *ambience* ja *constant mapping*. Se käyttää tekstuuria simuloimaan kappaleen itsevalaisu ominaisuuksia. Tämän mapin värit lisätään lopulliseen renderointiin riippumatta valaistuksesta.

2.2.4 Transparency

Tekstuuri, joka määrittää kappaleen läpinäkyvyyden. Tummemmat alueet ovat läpinäkyviä, kun taas vaaleat alueet läpinäkymättömiä. Sen sijaan että tekisi kappaleesta tasaisesti läpinäkyvän, voi *transparency map*illa osasta alueista vähemmän läpinäkyviä, kuten vaikka likaisessa ikkunassa.

2.2.5 Bump mapping

3D-pinnoite, joka näyttää kolmiulotteiselta. Pinnoitteen avulla voi kappaleen pinnasta tehdä epätasaisen näköisen. Kappaleen ääriviivojen muotoon se ei kuitenkaan vaikuta. *Bump mapping* käyttää pinnoitteen harmaan sävyjä syvyyden määrittämisessä. (Lehtovirta 2000, 199.) Toisin kuin *Lehtovirran* määritelmästä voi päätellä, ei pidä sotkea 3D-tekstuuriin, sillä *bump map* toimii myös 2D-pinnoitteena.

Bump map saa mallin reagoimaan valoon kuin siihen olisi lisätty enemmän yksityiskohtia. Mallissa ei kuitenkaan ole sen enempää yksityiskohtia, kuin aikaisemminkaan, joten *bump map*illa on rajoituksia. Se ei vaikutta kappaleen silhuettiin, vaikka sillä olisikin aikaan saatu karkea pinta. Se ei myöskään vaikuta varjoihin, joita kappale aiheuttaa muihin objekteihin, eikä kappaleeseen osuviin varjoihin. Kappaleessa *bump map*in ansiosta olevat yksityiskohdat eivät myöskään aiheuta varjoja kappaleeseen. Linja, jolla *bump map*illa käsitelty objekti leikkaa toisen objektin, ei myöskään muutu *bump map*illa miksikään.

Nämä kaikki ongelmat voidaan korjata *displacement map*illa.

2.2.6 Displacement mapping

On kappaleen muodon muuttamista kuvan avulla. Kappale muotoutuu siihen heijastetun kuvan mukaisesti. Vaaleat sävyt nostavat muotoja, tummat sävyt luovat kuoppia. (Lehtovirta 2000, 199).

Yksityiskohtainen displacement map vaatii, että kappale jaetaan moniin pieniin polygoneihin renderöitäessä, tästä syystä malli käyttää paljon enemmän polygoneja kuin mitä se olisi käyttänyt ilman displacement mappia. Liian pieni polygoni määrä aiheuttaa epämääräisen ja epäterävän jäljen, mutta resoluution lisääminen lisää myös vaadittavan muistin määrää ja renderöinnin kestoa.

2.2.7 Normal mapping

Normal map toimii samankaltaisesti kuin bump map, sekään ei oikeasti muokkaa kappaletta, vaan muodostaa kuviota renderöitäessä. Verrattuna bump mappiin normal map muokkaa määrätymmin pinnan normaaleja. Bump mapissa pikselin kirkkaus määrittää korkeuden, ja ympäröivien pikselien kirkkaus kulman. Normal mapissa tämä kulma tulee pikselille annetuista kolmesta arvosta jotka sijaitsevat kolmella mapin värikanavalla. Yleisin käyttö normal mapille on saada pienempi resoluutioinen malli näyttämään siltä, kuin siihen olisi siirretty yksityiskohdat korkea resoluutioisesta mallista. Tämä onnistuu automaattisesti laskemalla mallien erot.

2.3 Tekstuurit

Tekstuurit voidaan jakaa kahteen kategoriaan proceduraalisiin ja tiedostoihin perustuviin. Ensimmäiseksi mainitut ovat tietokoneen laskennallisesti tekemiä materiaaleja, joita voi muokata syöttämällä niille uusia arvoja. Jälkimmäiset vuorostaan ovat mikä tahansa ohjelman ymmärtämä kuvatiedosto tietokoneella. Maya-ohjelma ymmärtää myös Photoshop-ohjelman smart objekteja, joissa voi olla Illustrator-ohjelmasta siirrettyjä vektoritiedostoja, muuten tiedostotekstuurit ovat yleensä bittikarttoja kiinteällä resoluutiolla. Tämän vuoksi proceduraalisten tekstuurien suurin etu tiedostoihin perustuviin tekstuureihin nähden on niiden skaalautuvuus. Renderöitäessä suuria kuvia täytyy bittikarttatekstuurien olla vähintään yhtä suuria, kuin renderöitäessä niistä näkyvä alue, tai niiden resoluutio jää liian pieneksi ja niistä tulee sumeita. Proceduraalisilla tekstuureilla ei ole tätä rajoitusta.

2.4 Texture mapping

Erilaisia tapoja liittää tekstuuri pintaan ovat projektiot ja uv-koordinaatit, näiden lisäksi on vielä 3D-tekstuurit, jotka on syytä erottaa omaksi osiokseen. Projektioita ovat taso-, sylinteri-, pallo-, kuutio-, ja kutistuva projektiot. Projektioissa tekstuuri heijastetaan

kappaleen pinnalle erilaisista muodoista. Taso projektio on tasaiselta pinnalta heijastettava kuva, joka toimii hyvin heijastettavalta suunnalta, mutta venyy kappaleen muilta sivuilta, muut projektiot heijastetaan nimiensä mukaisista muodoista.

2.4.1 UV kartta (UV Map)

UV:t ovat kaksiulotteinen tekstuuri koordinaatisto, joka kulkee vertex komponentin informaation mukana polygon ja subdivision surface objekteissa. UV:n avulla määritellään kaksiulotteinen koordinaatisto ja tekstuurin sijoittuminen pinnalle. UV:t ovat merkkipisteitä, jotka määrittävät mitkä tekstuurin pikseleistä liittyvät mihinkin pinnan vertex-pisteistä.

UV-kartoitus (UV mapping) on prosessi, jossa luodaan ja muokataan kaksiulotteista mallia pinnasta tekstuurina käytettävän bittikarttakuvan päällä UV-tekstuurieditorissa. UV-koordinaatit ovat kappaleen pinnan pisteiden mukaan luotuja koordinaatteja. Niiden avulla tekstuuri on kuin käärepaperi johon kappale kääritään. UV-kartoitus on tärkeää tehdä ennen kuin mallia alkaa muokata Mudboxin tai Z-brushin kaltaisilla ohjelmilla, mikäli aikoo tehdä niissä mallinnetuista korkearesoluutioisista malleista normal tai displacement karttoja.

2.4.2 3D-tekstuuri

3d-tekstuuri on kolmiulotteinen proceduraalinen tekstuuri, jota käyttäessä kappale on kuin tehty tekstuurin määrittämästä aineesta, sillä siinä ei ole havaittavissa katkeamia kuviossa. 3d-tekstuuria voidaan verrata maalipurkkiin, jonne kappale laitetaan, jolloin materiaali tarttuu pintaan. Teksturoitaessa 3d-tekstuurilla animoitavaa hahmoa, jonka pinta tulee liikkumaan, pitää tehdä tekstuurille referenssiobjekti (texture reference object), joka lukitsee 3d-tekstuurin kiinni pintaan, jolloin se liikkuu pinnan mukana. Muutoin kohteen muuttaessaan muotoaan, tai liikkeessa tekstuuri pysyisi paikallaan, aivan kuin kappale uisi siinä.

Hyvin luoduilla tekstuureilla voidaan saada malli sisältämään huomattavan paljon enemmän yksityiskohtia kuin mitä sen geometriassa todellisuudessa on.

2.5 Valaistus

Valaistuksen voisi melkein ohittaa kokonaan, koska sehän on vain yksi tärkeimmistä asioista 3D-työskentelyssä. Kaikki video- ja valokuvaus perustuu valoon. Valo mahdollistaa asioiden näkemisen, heijastamalla eri asioista. Valitettavan usein huomaa ettei valaistusta ole mietitty tai toteutettu kunnolla useimmissa 3D-toteutuksissa.

3D-ohjelmissa on perusvalojen, joita ovat muun muassa spot, directional, area, point/omni ja ambient, lisäksi erilaisia tapoja, joilla simuloida luonnollisenvalon käyttäytymistä.

2.5.1

Pistevalo (Point Light)

Pistevalo (point/omni), on yksinkertaisin valonlähde käyttää 3D:ssä. pistevalo on kuin hehkulamppu roikkumassa huoneen katosta, kaikki sen valonsäteet lähtevät samasta pisteestä, joka suuntaan, samalla voimakkuudella.

2.5.2

Kohdevalo (Spotlight)

Kohdevalo (Spotlight) on suosituin valotyyppi 3d:ssä, sen hallittavuuden vuoksi. Kuten pistevalossakin kaikki sen säteet lähtevät samasta pisteestä, mutta sen sijaan että se lähettäisi valoa joka suuntaan se on rajoitettu valittuun suuntaan osoittavaan kartioon.

2.5.3

Suuntaavavallo (Directional

Light)

Suuntaavavaloa käytetään usein simuloimaan suoraa auringon valoa. Se tunnetaan myös nimillä distant, direct, infinite tai sunlight riippuen ohjelmasta. Suuntaava valo valaisee jokaisen objektin samasta kulmasta riippumatta objektin sijainnista suhteessa valoon.

2.5.4

Aluevalo (Area Light)

Aluevalo on valotyyppi, joka simuloi fyysisen valolähteen kokoa reaali maailmassa. Toisin kuin piste-, kohde- ja suuntaavanvalon kanssa, aluevalon kokoa muuttamalla 3D-ympäristössä muuttaa myös sen valaisua. Suurempi aluevalo valaisee suuremman alueen. Aluevaloista voi tehdä erimuotoisia, näitä muotoja ovat nelikulmiot, kiekot, pallot ja lineaarinen. Alue valoilla saavutettavan valon ja varjojen laatu tekee niistä

loistavan valinnan joihinkin realistisiin renderöinteihin, mutta niiden varjojen renderöiminen on usein hitaampaa ja monimutkaisempaa, kuin mitä olisi kohde- ja pistevalojen kanssa, koska rendereri joutuu levittämään useampia säteitä alueelle testaamaan aiheuttavatko ne osittaisia varjoja. Monet välttävätkin aluevaloja animaatioissa näiden hitaan renderöimisen vuoksi.

2.5.5

Ympäröivävalo (Ambient Light)

Ympäröivävalo, reaali maailmassa ympäröivävalo tarkoittaa kaikkea valoa, jota ympärilläsi on. Se sisältää taivaan valon, maasta heijastuneen valon ja kaikki muut valon lähteet. On kuitenkin havaittavissa, että reaali maailmassa ympäröivä valo valaisee eri sävyin ja voimakkuuksin eri alueilla, jotka vaihtuvat eri ympäristöissä. Harvoin tämä valo on jokapuolella tasaista. Tietokone 3D-grafiikassa useimmissa ohjelmissa on ympäröivävalo, joka epärealistisesti valaisee objekteja joka puolelta tasaisesti. Ensimmäinen sääntö näin toimivan ympäröivänvalon kanssa onkin, ettei sitä pidä käyttää.

2.5.6

Kokonaisvalaistus (Global illumination)

Kokonaisvalaistukseksi kutsutaan niitä renderöinti algoritmeja, jotka simuloivat valon heijastumista pintojen välillä. Näitä ovat radiositeetti, fotonikartoitus, final gather ja caustics.

2.5.7

Radiositeetti (radiosity)

Radiositeetti yksi tapa saada aikaan kokonaisvalaistus, jossa epäsuora valaistus lähetetään pintavärin hajaantuneena heijastuksena ja tallennetaan pinnan vertekseihin. Radiositeetti voidaan laskea progressiivisesti, joten valo voi kimpoilla niin kauan, kuin on tarpeen, jotta saadaan aikaan tarkka simulaatio oikeasta valaistuksesta. Valon kimpoamisten määrä on rajoitettu ainoastaan siihen käytettävissä olevaan laskuaikaan. Radiositeetti oli ensimmäisiä tapoja tuottaa kokonaisvalaistusta, minkä vuoksi jotkut ihmiset kutsuvat virheellisesti kaikkia kokonaisvalaistustekniikoita radiositeetiksi. Radiositeetti tallentaa valoinformaation kappaleen pinnan vertekseihin, jolloin kappaleen resoluutio vaikuttaa valon resoluutioon, tämän vuoksi on vaikea laskea nopeaa arviota valaistuksesta näkymään, jossa on korkea polygonien lukumäärä.

Kappaleiden liikkua joutuu radiositeetilaskennassa valaistuksen laskemaan uudelleen, mikä on vähentänyt sen suosiota animaatiokäytössä. Arkkitehdit sen sijaan suosivat radiositeettiä, koska liikkumattomien objektien kanssa voi valaistuksen laskea kerran, jonka jälkeen samaa valaistusta voi käyttää renderöimään vaikka kamera-ajon.

2.5.8 **Mapping)**

Fotonikartoitus (Photon

Radiositeetin kaltaisia tuloksia voi saavuttaa fotonikartoituksen (photon mapping) avulla. Tällöin luodaan erillinen fotonikartta johon kokonaisvalaistuksen arvot tallennetaan, minkä vuoksi fotonikartoitus ei ole riippuvainen geometrian resoluutiosta. fotonikartoituksen nopeus ja tarkkuus riippuu käytettyjen fotonien määrästä. Pienellä määrällä fotoneita voi tehdä nopean testirenderöinnin ja lisäämällä niitä saada aikaan tarkemman valaistuksen. Fotoneita voi ajatella eräänlaisena energiana, joka lähtee valosta ja etenee kimpoillein vastaan tulevista kappaleista kunnes sen energia on lopussa. Fotonien koko vaihtelee niille annettavien parametrien mukaan, jos niille on annettu arvoksi nolla ne mukautuvat scenen koon mukaan. Alhainen määrä fotoneja tekee valaistuksesta laukukkaan. Fotonien määrää lisäämällä lopputuloksesta tulee pehmeämpi, mutta tällöin renderöinti aika kasvaa myös. Pelkkää fotonikarttaa käyttämällä voidaan saada aikaan hyvä lopputulos, mutta joissain tapauksissa tarvittavien fotonien määrä on niin suuri, että renderöintiajat kasvavat liian pitkiksi, tällöin voidaan ottaa avuksi Final Gather.

2.5.9

Final Gathering

Final gather on renderöinti optio, jota voi käyttää yksinään tai pehmentämään fotonikarttaan perustuvaa kokonaisvalaistusta. Final gather aloittaa renderöimällä koko kuvan, jonka jälkeen se katsoo joka suuntaan, mitä on näkyvissä kameran näkökulmasta. Katsottavien suuntien määrää voi säätää säteiden ja näytteiden lukumäärää muuttamalla. Ottaessaan näytteitä eri suunnista se katsoo pintojen väriä ja valaistusta, lisätäkseen ne kokonaisvalaistukseen, sekä samalla se katsoo lähellä olevia fotoneja fotonikartassa, jotta se voi pehmentää fotonikartan avulla tehtyä kokonaisvalaistusta. Käytettäessä ilman fotonikartoitusta final gather lisää valoon yhden kimpoamisen, joissain ohjelmissa näitä valon kimpoamisia voi lisätä useammankin.

2.5.10

Caustics

Caustic tarkoittaa epäsuoraa valoa, joka ei hajoa tai leviä vaan aiheuttaa kuvion. Reaalimaailmassa caustic efektin voi havaita muun muassa valon taittuessa lasiesineiden läpi, valonsäteen heijastuessa peilistä, kiiltävästä materiaalista tai vedenpinnasta.

2.6 Varjot

Varjot ovat toinen puolisko valaistusprosessista, Reaalimaailmassa varjot syntyvät automaattisesti, eikä niiden ilmaantumista voi estää. Tietokonegraafiikassa varjot täytyy laskea erikseen, mikä on yksi hitaimmista laskettavista valaistuksessa. Varjot voidaan jakaa kahteen pääkategoriaan syvyyskarttoihin (depth map) ja säteenseurantaan (ray trace).

2.6.1

Syvyyskarttavarjot (*Shadow map/ depthmap shadows*)

map/ depthmap shadows

Ennen renderöinnin alkua rendereri laskee syvyyskartan jokaisen syvyyskarttavarjoa käyttävän valon näkökulmasta. Jokaisesta suunnasta mistä valo loistaa syvyyskartta laskee etäisyyden lähimpään varjon aiheuttavaan objektiin siinä suunnassa. Renderöitäessä, valo leikkaantuu syvyyskartan määrittämän etäisyyden verran.

2.6.2

Säteenseuranta varjot

(Raytraced Shadows)

Säteenseuranta varjot lasketaan seuraamalla valonsäteitä valon lähteen ja valaistavan objektin välillä. Säteenseuranta varjot lasketaan pikseli kerrallaan muun renderöinnin ohella. Säteenseuranta varjoilla on monia etuja syvyyskarttavarjoihin verrattuina, kuten ne ovat vaaleampia läpinäkyvien kappaleiden kanssa ja voivat poimia värejä läpinäkyvistä pinnoista. Niiden kanssa ei tarvitse säätää bias-arvoa korjatakseen valovuotoja. Ne ovat resoluutio riippumattomia. Ne luovat uskottavimmat ja realistisemmat pehmeät varjot, kun niitä käytetään aluevalonkanssa. Säteenseurantavalot on useimmissa tapauksissa kuitenkin hitaampi renderöidä kuin syvyyskartta varjot, sekä ne kuluttavat enemmän muistia.

2.7 Riggaus

Riggauksella tarkoitetaan sekä mallille luurangon, että sitä ajavan kontrollisysteemin luomista (Ford, Lehman, 2002). Animaatioita, ja erityisesti hahmoanimaatiota varten usein joutuu mallille tekemään luurangon, jonka avulla mallin geometria voi muokata ja animoida. Luilla on hierarkia, jonka mukaan ne liikkuvat, juuri (root) liikuttaa kaikkia luita ja siitä hierarkiassa alaspäin olevat liikuttavat mukanaan niitä alempia ja niin edelleen. Esimerkiksi, kun pyöräyttää olkapäätä, mukana liikkuvat myös kyynärpää, ranne ja sormet. Yleisimmin koko hahmon juuri sijaitsee lantiossa, joka on ainakin humanoidihahmoilla usein tasapainopiste.

Luita täytyy olla riittävästi, jotta hahmo pystyy muuttamaan muotoaan luontevasti. Jos käytössä on vain yksi nivel selkärangalle, ei useimmissa tapauksissa päästä hyvään lopputulokseen. Liian monta luuta aiheuttaa vuorostaan enemmän työtä animoitaessa, tuomalla lisää animoitavia kohteita.

Nivelet pitää asemoida samoin suhteessa 3d-avaruuteen (local rotation axis), jotta pyöritettäessä samalla akselilla kaikki nivelet liikkuvat samaan suuntaan. Esimerkiksi jos olkapäälle on määritetty akseli niin, että pyörytettäessä z-akselia positiiviseen, käsi kääntyy hahmosta eteenpäin. Pitäisi kääntämällä z-akselilla positiiviseen kyynärpään nivelestä, käsivarren kääntyä myös eteenpäin hahmosta katsottuna. Näin menettelemällä hahmo käyttäytyy luonnollisemmin ja on helpommin animoitavissa.

2.7.1 *Forward / inverse kinematic*

Perinteinen lähestyminen hahmon animointiin on pyörittää jokaista niveltä vuorollaan alkaen hierarkian alusta loppuun asti, jolloin esimerkiksi kättä animoitaessa aloitetaan olkapäästä ja siirrytään kyynärpäähän kautta ranteeseen ja niin edelleen. Tätä lähestymistapaa kutsutaan nimellä forward kinematics (suora kinematiikka). Tämän lähestymistavan etuna on automaattisesti luonnollisessa kaarella tapahtuvat liikkeet.

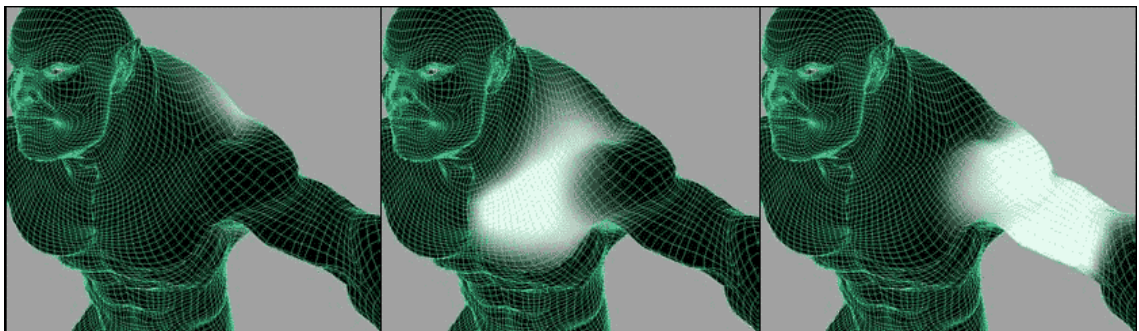
Toinen lähestymistapa on inverse kinematics (käänteinen kinematiikka). Tässä tapauksessa liikutetaan hierarkian viimeistä osaa, ja ohjelma ohjaa muiden nivelien liikkeitä. Esimerkiksi rannetta siirtämällä siirtyy kyynärpää automaattisesti. Tällä menetelmällä päästään nopeammin liikkeen ääriasentoihin, koska animoitavia kohteita on vain yksi, mutta liike on kuitenkin mekaanisemman oloista, koska haluttu

muodonmuutos tapahtuu liikuttamalla niveliä suoraviivaisesti, eikä niitä pyörittämällä. Yksi käänteisen kinematiikan käyttötarkoitus on estää hahmoa läpäisemästä erilaisia pintoja, kuten vaikka maankamaraa (Lew, 2003).

Eri ohjelmissa on näiden lisäksi erilaisia, mutta kuitenkin samoihin periaatteisiin perustuvia ratkaisuja, joilla voidaan helpottaa hahmon animointia. 3d studio Max ohjelmassa on character studio laajennus, jossa on valmis kaksijalkainen hahmo jonka voi liittää omaan geometriaansa, Maya-ohjelmasta taas löytyy full body ik työkalu, jolla voidaan luoda hahmon koko luurangolle rakenne, joka toimii saman aikaan sekä forward- että inverse kinematiikka periaatteilla.

2.7.2 Luurangon ja hahmon geometrian liittäminen toisiinsa

Valmis luuranko täytyy liittää hahmoon, jotta sillä voitaisiin ohjata geometrian muodon muutoksia. Hahmon geometrian jokainen ohjauspiste sidotaan jonkun nivelen alaiseksi, joka tämän jälkeen määrittää pisteiden liikkeitä. Yhden geometrian ohjauspisteen voi määrittää ottamaan vastaan liikeinformaatiota yhdeltä (rigid bind) tai useammalta (smooth bind) niveleltä, näiden vaikutusvoimakkuuksia säädetään maalaamalla jokaisen nivelen vaikutusalue ja voimakkuus.



kuva 3 paint weights

Vaikutusalueiden maalaaminen tapahtuu valitsemalla nivel, jonka jälkeen käytetään (paint weights) vaikutusalueen maalaustyökalua. Tällä työkalulla sitten maalataan hiiren tai piirtopöydän avulla 3d-tilassa objektin pinnalle harmaasävyillä, joista valkoiset osat määrittävät ne pinnan osat, joihin nivel vaikuttaa. Mitä kirkkaampi valkoinen osa on, sitä voimakkaammin vaikutus näkyy niveltä liikuttaessa. Yhdistelemällä useamman nivelen vaikutusta ohjauspisteisiin saadaan aikaiseksi pehmeämpiä siirtymiä ja illuusio liikkuvasta ihosta.

Tämä työvaihe on yksi koko hahmorakentamisen tärkeimpiä ja aikaa vievimpiä osia, jos jokin väärä nivel vaikuttaa väärin ohjauspisteisiin, syntyy vääristynyttä muodonmuutosta, minkä takia tässä työvaiheessa tuleekin olla erittäin tarkka.

Esimerkiksi, jos vasemman nilkan nivel ohjaakin vahingossa yhtä ohjauspistettä oikeassa nilkassa, niin joka kerta, kun vasenta nilkkaa liikutettaisiin, liikkuisi myös oikeassa nilkassa oleva piste, vaikkei oikean nilkan pitäisikään liikkua minnekään.

Vaikutusalueiden maalaamisen jälkeen geometria tottelee luurangon käskyjä, joskin kehossa voi olla useita ongelmakohtia, joissa geometria menee itsensä sisään.

Esimerkiksi, taivutettaessa käsivartta olkapäätä kohti, saattaa kyynärpäähän seudulla ilmaantua ongelmia. Yksi tapa korjata tällaiset ongelmat on kopioida taittuvan käsivarren geometria ja muokata kopiosta perusmallinnustekniikoita hyväksikäyttäen anatomisesti oikein toimiva, jonka jälkeen voidaan tämän uuden mallin avulla ohjata alkuperäinen muuttamaan muotoaan aina käden taittuessa. Tähän tarkoitukseen käytetään Mayassa työkalua blend shape, jota taas 3d Studio Maxissa vastaa morpher.

Animaation helpottamiseksi on suositeltavaa luoda toimivat animaatiokontrollit, eli tehdä omat valikot, joista löytää helposti animoitavat kohteet, sekä hahmoon liitetyt avusteet, joilla voidaan valikot valita helposti. Sen sijaan, että joutuisi joka kerta etsimään hahmon osan, jonka haluaa animoida, on helpompaa valita se lyhyestä listasta.

Esimerkiksi kasvon ilmeille ja sormille olisi hyvä olla omat valikkonsa. On huomattavasti tehokkaampaa valita kaikkia sormia kerralla kontrolloiva liukusäädin, kuin käydä liikuttamassa jokaista niveltä vuorollaan. Kasvoja varten on puolestaan hyvä tehdä valmiit valinnat jokaiselle äännteelle, jolloin huulisynkronoinnin tekeminen helpottuu huomattavasti. Keskittämällä animaatiokontrollit saman nodin alle, helpottaa avainkehysten (keyframes) hallintaa. Tämän voi tehdä käyttämällä character set-työkalua.. Character set:in ei tarvitse olla fyysinen hahmo, vaan se on kokoelma attribuutteja, jotka on tarkoitettu animoida yhdessä. Animaatiokontrollien tulisi olla intuitiivisia ja helposti löydettäviä, erityisesti, jos niiden tekijä ei itse toimi animaattorina.

2.8 Animaatio

Animaatiosta on tehty tuhansia kirjoja, jotka kuitenkin perustuvat Disneyn *illusion of life* -kirjaan. Kaikkien sen nyanssien selittäminen tässä ei ole käytännöllistä, joten tyydyn kertomaan siitä vain pinnallisesti.

Animoinnilla tarkoitetaan tapahtumaa, jossa jokin, joka ei itse pysty liikkumaan, laitetaan liikkumaan. Animointi lisää kuvaan uuden määreen, ajan (Lehtovirta, Nuutinen, 2000). Animointi tapahtuu tekemällä objektien parametreille muutoksia ajassa suhteessa niiden alkuperäisarvoihin ja asettamalla keyframeet kohtiin, joissa muutokset tapahtuvat. Kappaleen liikkeen pisteestä a pisteeseen b tapahtuu antamalla keyframe kappaleelle sen ollessa pisteessä a, jonka jälkeen siirrytään ajassa eteenpäin, ja siirretään kappale pisteeseen b ja annetaan sille keyframe. Se miten kappale käyttäytyy näiden keyframeiden välissä, riippuu animaatiokurveista. Animaatiokurvit ovat graafinen tapa ilmaista kappaleen arvojen vaihtelua ajassa.

2.9 Renderointi

Renderöinniksi kutsutaan prosessia, jossa renderöinti ohjelma laskee jokaiselle kuvan pikselille arvot, joita ovat punainen, vihreä, sininen, alpha ja Z-syvyys (Z-depth). Nämä laskutoimitukset perustuvat siihen mitkä objektit ovat pikselin kohdalla, mitkä valot sitä valaisevat ja kameran asetuksiin (Parrish, 2002 123). Renderöintiohjelmia on monia erilaisia, Mayassa mukana tule Mayan software-, hardware-, vektori- rendererit, sekä Mental Ray renderer. Näiden lisäksi on erillisiä renderöintiohjelmia kuten Renderman, Brazil, Maxwell ja V-ray.

Jokaisessa renderöintiohjelmassa on omat asetuksensa, joita säätämällä voi vaikuttaa renderöitävään kuvaan, sekä renderöinnin kestoon. Renderöitäessä animaatiota on hyvä muistaa, lopullisen renderöintiajan kasvavan reilusti, vaikka yhden kuvan renderöiminen toisilla asetuksilla veisikin vain vähän kauemmin. Tämän vuoksi renderöinti kannattaakin optimoida mahdollisimman hyvin.

2.9.1 Renderöinti kerrokset (*Render Layer*)

kerroksina renderöiminen tarkoittaa eri objektien renderöimistä erillisiin kuvatiedostoihin, jotka liitetään kompositio vaiheessa toisiinsa. Kerroksina

renderöiminen vaatii enemmän valmistelu aikaa kuin kaikkien objektien renderöiminen kerralla. Voisi kuvitella ettei kerroksilla renderöiminen ole järkevää, koska silloin kuluu enemmän aikaa objektien jakamisessa layereille, sekä niiden kompositoinnissa yhdeksi kuvaksi. Kerroksien renderöinnissä on kuitenkin monia etuja, se mahdollistaa suurten ja monipuolisten kuvien renderöimisen, jos tällaista kuvaa yrittää renderöidä kerralla, voi tietokoneesta loppua muisti ja se voi joutua lataamaan dataa kovalevyiltä, joka hidastaa konetta huomattavasti tai saattaa kaataa sen kokonaan. Paikallaan olevien objektien renderöinti nopeutuu huomattavasti, kun ei tarvitse renderöidä kuin ne kehykset (frame), joissa muutosta tapahtuu. Syväterävyyden aikaansaamiseksi voi parhaassa tapauksessa selvittää pelkällä taustalayerin sumentamisella kompositio-ohjelmassa, ilman että tarvitsee laskea sitä hitaasti rendererilla. Eri kerroksiin voi käyttää eri renderöinti asetuksia tai vaikka eri rendereriä. Kertaalleen renderöityjä elementtejä voi käyttää uudelleen muissa kohtauksissa. Renderöinti ongelmia voi kiertää, jos vaikka tietty efekti ei toimi läpinäkyvyyden kanssa, voi sen renderöidä erikseen ja kompositoida yhteen liittäen läpinäkyvyyden omaava elementti efekti layerin päälle. Lopulta voi tehdä nopeita värin ja kirkkauden korjauksia kompositio-ohjelmassa ilman hidasta uudelleen renderöintiä.

2.9.2 Renderöinti pyyhkäisy (Render Pass)

Renderöinti pyyhkäisy (Render pass), tarkoittaa kuvan eri attribuuttien renderöimistä erillisinä pyyhkäisyinä. Nämä pyyhkäisyvät ovat nimetty niiden ominaisuuksien mukaan, jotka ne eristävät, kuten varjot. Kun layereiden renderöiminen tarkoittaa ero objektien renderöimistä erikseen, tarkoittaa pyyhkäisyjen renderöiminen kuvan eri osien eristämistä toisistaan, kuten valaistus, varjot ja syvyysinformaatio. Yleisimmät pyyhkäisyvät ovat diffuse, specular, reflection, shadow, ambient, occlusion, beauty, global illumination, mask ja depth. Renderöinti pyyhkäisyjen avulla voi tarpeen vaatiessa vaihtaa kompositio-ohjelmassa vaikka renderöidyn väri pyyhkäisyasetuksia, jolloin punaiseksi renderöity auto vaihtaakin värinsä siniseksi muokattaessa sen sävyä. Näin toimiminen on huomattavan paljon nopeampaa, kuin koko kuvan uudelleen renderöinti. Renderöinti pyyhkäisyjen käyttö ja layereiden renderöinti eivät sulje pois toisiaan. Mitä enemmän passeja ja layereita renderöi, sitä enemmän on niiden erittelyssä työtä, mutta sitä enemmän on myös vaikutusmahdollisuuksia myöhemmin kompositiovaiheessa.

2.9.3***Diffuse Passes***

Diffuse pyyhkäisy on renderöinti, jossa kohde renderöidään, diffuse valaistuksen, värin ja tekstuurin kanssa, mutta ilman heijastuksia, kohokohtia (highlights) tai varjoja.

2.9.4***Specular Passes***

specular pyyhkäisyssä eristetään objektin kohokohtat (highlights). Specular pyyhkäisyä voi käyttää myös hohto (glow) efektin aikaan saamiseksi. lisäämällä sumennetun (blurred) kopion specular pyyhkäisystä kohokohtien päälle.

2.9.5***Reflection Passes***

Heijastuspyyhkäisy voi sisältää itseheijastuksia, heijastuksia toisista objekteista tai heijastuksia ympäristöstä. Usein pitää renderöidä useampia heijastuspyyhkäisyjä, varsinkin jos haluaa eristää eri objektien heijastukset.

2.9.6***Shadow Passes***

Varjopyyhkäisyssä renderöidään varjot.

2.9.7***Color Passes***

Väripyyhkäisy (color/ambient pass) näyttää värin ja tekstuurin, muttei sisällä mitään valoinformaatiota.

2.9.8***Occlusion Passes***

Occlusion pyyhkäisy tummentaa kuvaa perustuen objektien etäisyyksiin toisiinsa, mitä lähempänä objektit ovat, sitä tummempia ne ovat ja toisinpäin.

2.9.9***Beauty Passes***

Beauty pyyhkäisyllä tarkoitetaan koko kuvan renderöimistä kerralla.

2.9.10***Mask Passes***

Maski pyyhkäisy, jota myös matte- ja alpha pyyhkäisyksikin kutsutaan, on pyyhkäisy joka tuottaa maskin, mikä näyttää objektin paikan kuvassa.

2.9.11***Global Illumination Passes***

Kokonaisvalaistus pyyhkäisy eristää epäsuoranvalaistuksen kuvasta.

2.9.12***Depth Passes***

Syvyyspyyhkäisy tallentaa syvyysinformaation jokaisesta kuvan pisteestä.

3 Työprosessi

Demon 3d-osuuksie työstö tapahtui käytännössä viiden hengen voimin, Torssosen Samulin johdolla. Mallintelimme objekteja, jotka sitten lähetimme Samulille, joka koosti ja renderöi niistä kuvat lightwave-ohjelmalla. Ainoana poikkeuksena tähän ketjuun oli kotkalogo kohta, jonka toteutin Mayalla.

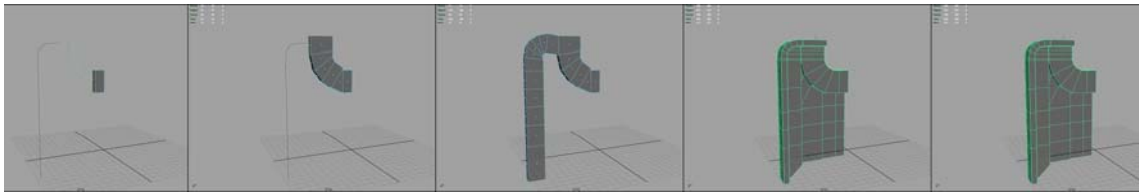
Aloitin kuvauksen varsinaisesta työstä, jota tein Iron sky demoon, keskittyen kuitenkin vain logokotka julisteeseen ja animaatioon, vaikka osallistuin myös muuhun työhön demossa.

Lähdemateriaalina työlle minulla oli käytettävissäni aikaisemmin tehty 2D-logokuva kotka patsaasta ja iron sky tekstistä, sekä elokuvan innokkaan fanin tekemä pronssivalos patsaasta. Näiden materiaalien pohjalta minun tuli rakentaa 3D-malli kotkasta, jota tulitisiin käyttämään muun muassa isossa julisteessa, sekä demossa.

2D-logokuvan kotka oli tyylielty ja muistutti enemmän papukaijaa kuin kotkaa. Aluksi lähdin mallintamaan patsaasta 2D-logokuvan näköistä, vaikka näinkin ettei se tulisi toimimaan suoraan 3D-mallina, näin toimimalla sain kuitenkin perusteltua ohjaajalle kantani, kun hän näki ne ongelma kohdat joita tästä siirtymisestä 3d-malliin tuli. Tässä vaiheessa kaivoin esiin googlen kuvahaulla referenssikuvia kotkasta, jolloin erot 2d-kuvan ja oikean kotkan välillä kävivät selviksi. Lopputuloksesta tuli kotkan ja 2d-kuvan välimuoto.

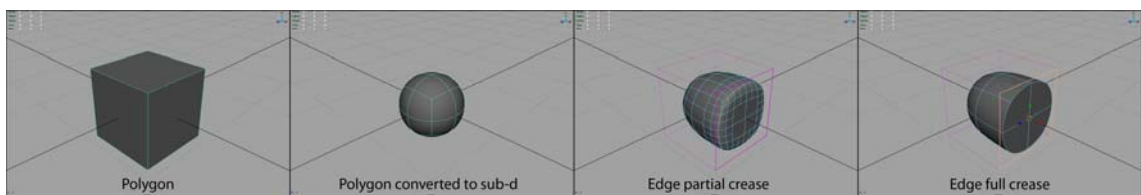
3.1 Mallinnus

Perusmallinnuksen aloitin mayassa muokkaamalla kuutiota, josta käyttämällä extrude työkalua extrudasin kurvia myöten siiven profiilin. Tätä siipeä muokkasin verteksi tasolla saavuttaakseni halutun muodon. Muokkasin polygoni verkkoa cut faces työkalulla niin, että kaikista polygoneista tuli quadeja eli nelikulmioita. Tein niistä nelikulmiota, koska silloin käänös polygonista subdivision surface muotoon onnistuu parhaiten, sekä teksturoitaessa displacement mapilla osaa ohjelma laskea alueen oikein. N-goneilla (monikulmio) tai triangeleina (kolmio) saattaa tulla virheitä.



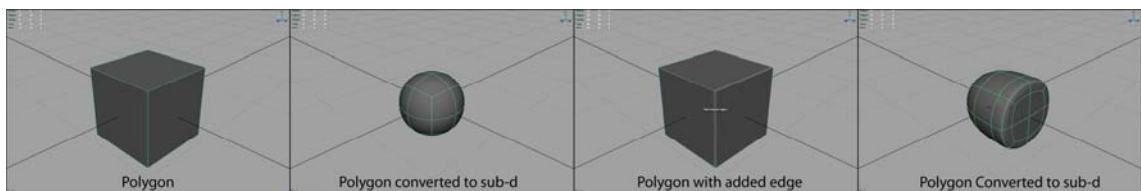
kuva 4 Siiven mallinnus.

Saadakseni terävähköt reunat aikaiseksi subdivision surfacelle oli käytettävissäni kaksi erilaista toiminta mallia. Ensimmäisessä vaihtoehdossa käytetään subdivision surfacen crease työkalua, jolla voidaan osittain tai kokonaan tehdä reunasta terävä. (kuva 5)



kuva 5

Toisena ja käyttämäni vaihtoehtona oli lisätä uusi edge kulman viereen niihin kulmiin joihin halusin terävän näköisen kulman. (kuva 6)



kuva 6

Tämä vaihtoehto oli tähän tarkoitukseen parempi, koska nämä muokkaukset eivät katoa konvertoitaessa kappale takaisin polygoni muotoon niin kuin olisi käynyt crease toimintoa käyttämällä.

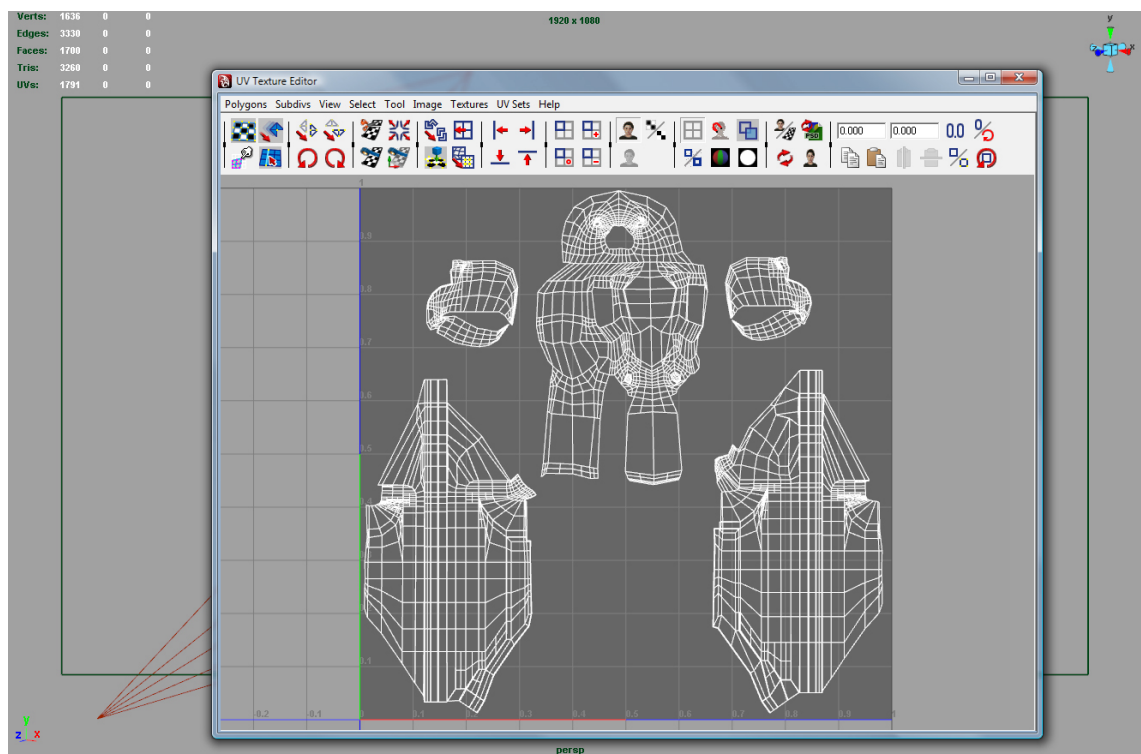
Muunsin mallin subdivision surfaceksi, koska tällöin sain käyttööni hyviä puolia sekä, polygon että nurbs mallintamisesta. Subdivision surface kappaletta voi muokata kuten polygon malliakin, mutta se on muodoiltaan kaareva kuin nurbs, eikä näin ollen ole niin riippuvainen yksityiskohtien määrästä kuin polygonit, pystyen kuitenkin esittämään kaarevat pinnat paljon tehokkaammin. Subdivision surface mallia muokkaamalla tein siiven haluttuun muotoon, jonka jälkeen kopion siitä peilikuvan minkä liitin alkuperäiseen näin saaden aikaan symmetriset siivet. Siipien väliin mallinsin samoja

menetelmiä hyödyntämällä kotkan pään ja vartalon. Lopulta yhdistin nämä yhdeksi objektiksi. Jalat tein jälleen kuutiosta muokkaamalla extrude työkalun avulla, extruudaamalla varpaat ja kynnet. Toisen jalan kopioin toisen pelikuvaksi. Kuu puolestaan oli vain yksinkertainen pallo. Saatuani mallin esikuvansa 2d-mallin näköiseksi alkoi ohjaajan ohjeilla uudelleen muokkaus. Tässä työ vaiheessa ohjaaja kertoi mitä halusi ja minä tein muutokset, jonka jälkeen usein palattiin edelliseen versioon, tämän jatkuen niin kauan kunnes ohjaajaa tyydyttävä malli oli valmis. Tässä työvaiheessa ohjaajan näkemyksen muokkaaminen oikeaan suuntaan tämän sitä huomaamatta oli erittäin tärkeää. Alkuperäisessä kaksiulotteisessa logossa ei kotka vastannut juurikaan oikeaa kotkaa ja muistuttikin ehkä enemmän papukaijaa. 3D-mallista tuli lopulta kotkan ja papukaijan välimuoto.

Tekstin mallintamisessa käytin apuna kuvaa jossa oli 2d-tekstiä, koska vektori muodossa sitä ei ollut saatavissa. Tekstin mallinnus tapahtui piirtämällä create polygon työkalulla tekstin kulmia mukaillen sen ääri viivat kirjain kerrallaan. Nämä ngonit jaottelin cut faces työkalulla quadeiksi. Lopulta extrudasin kirjaimista kolmiulotteiset. Sen sijaan että olisin käyttänyt aikaa reunojen pyöristämiseen, päätin kokeilla Mental Rayn round corners nodia, joka yhdistettynä Mentayl Ray materiaaliin laskee renderöitäessä kappaleelle pyöreät reunat. Round corners toimii samankaltaisesti kuin bump map ja liitetäänkin materiaaliin samoin kuin bump map. Vasta renderoituani kuvan havaitsin käyttäneeni vääränlaista materiaalia, eikä lopputulos ollut aivan hakemani. Tätä väärin renderöityä tulosta kuitenkin käytettiin lopulta.

3.2 UV-mappaus

Mallin valmistumisen jälkeen oli vuorossa sen uv-mappaus, jossa annetaan mallille koordinaatit, joiden mukaan tekstuurit tulevat sen pinnalle. Tässä prosessissa käytin enimmäkseen projektio mappoja, jotka projisoidaan halutusta suunnasta kappaleen päälle. Näitä tekemällä erilaisista kulmista, ja lopulta niiden yhteen liittäminen uv-tekstuureditorissa, aikaan sai yhtenäisen uv-mapin, jonka avulla projektin seuraava vaihe pääsi alkamaan.



kuva 7 UV-map

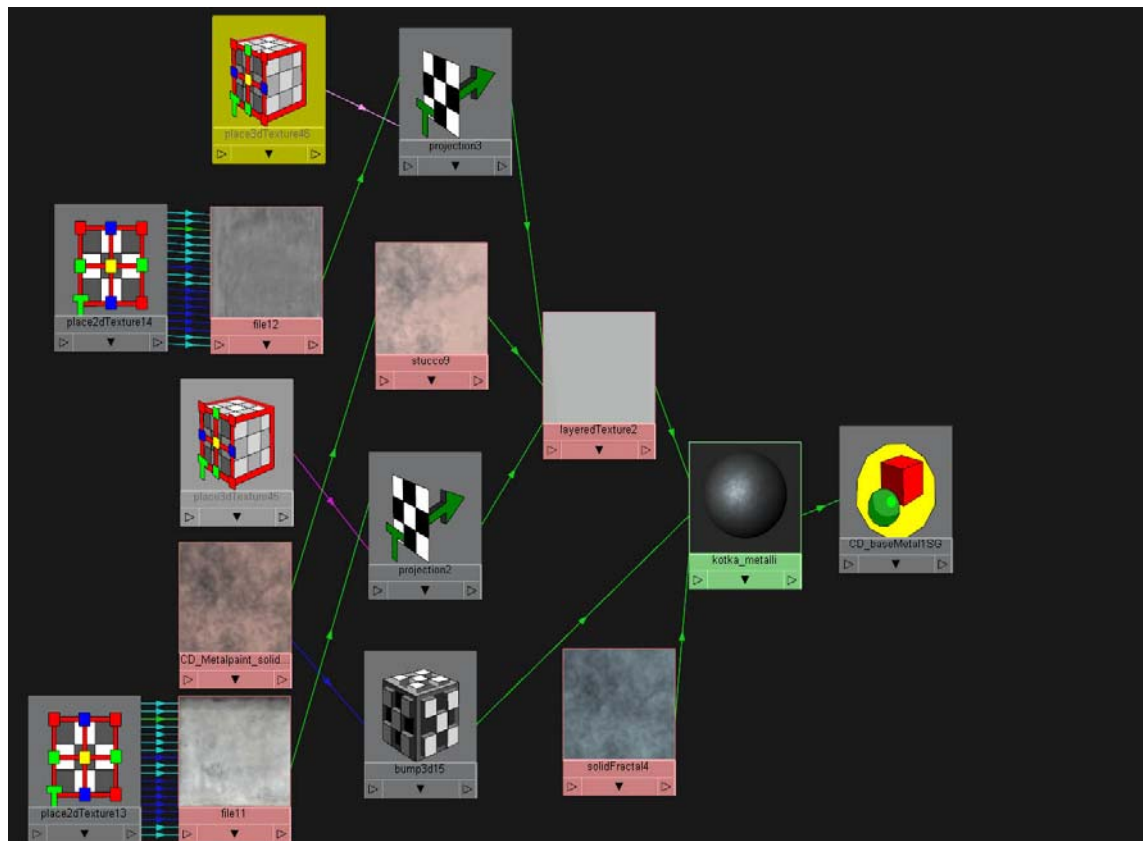
3.3 Mudbox

Mudbox on uudenlainen mallinnusohjelma. Se on melkoisen rajoittunut toiminnoiltaan, jos sitä vertaa 3d:n monipuolisimpiin työkaluihin kuten Maya Max, tai Xsi, mutta se

hoitaa sen pienen osa-alueen mihin se on tarkoitettu erinomaisesti. Mudboxissa yleensä toisesta ohjelmasta tuotua mallia muokataan erilaisilla siveltimillä, ja se pääsee käytettävyydeltään melko lähelle saven tai muovailuvahan muokkausta. Sillä voi nopeasti muokata sellaisia polygoni määriä, jotka saisivat monet muut ohjelmat polvilleen. Mudboxin suurin voima on kuitenkin sen mullistavat 3D-layerit. 3D-layerit ovat kuin Photoshopin layereita, joihin voi tehdä muokkauksia malliin, jonka jälkeen layerin voi poistaa, piilottaa tai sen läpinäkyvyyttä voi muuttaa. Tämä mahdollistaa non destructive (häviämättömän) mallintamisen, missä kappaleita voi muokata ja tarpeen vaatiessa poistaa muokkaukset nopeasti ilman kappaleen työlästä käsittelyä. Esimerkiksi kotkaa mallintaessa piti siihen tehdä kulumia ja naarmuja. Normaalein menetelmin olisin tehnyt naarmut ja kulumat samoin kuin nytkin, mutta, jos ohjaaja ei haluakaan tietyille alueille naarmuja olisi normaalilla työskentelytavalla eli verteksipisteiden siirtelyllä, niiden poistaminen ollut yhtä työlästä, kuin niiden tekeminen, varsinkin jos ei halunnut muuttaa muita yksityiskohtia muokattavalta alueelta. Mudboxin 3D-layerien ansiosta tämä onnistui vain pyyhkimällä alueelta naarmut pois erase siveltimellä. Mudbox työvaiheessa yksi tarkoituksista oli luoda mudboxissa korkearesoluutioisesta mallista displacement map, jonka avulla voidaan renderoitaessa luoda yksityiskohtia matalaresoluutioisempaan malliin. Kohdemateriaalin vaatiessa todella suurta resoluutiota displacement mapilta, tuli kuitenkin teknisiä esteitä vastaan, ilmeisesti Mudboxin 32bittisen muistinkäsittelyn vuoksi ei siitä saatu laskettua riittävän tarkkoja displacement mappeja, jolloin jouduin käyttämään korkearesoluutiosta mallia siirtäen sen suoraan Mayaan.

3.4 Materiaalit ja tekstuurit

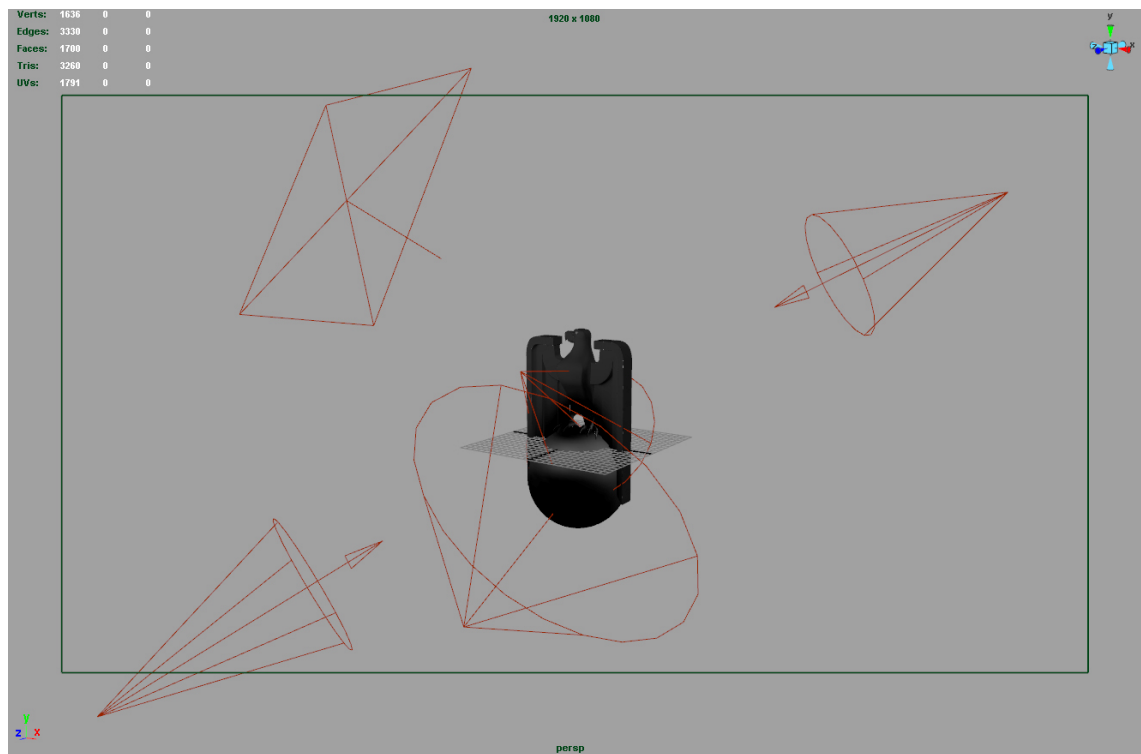
Kotkasta piti renderöidä 70cm*50cm julisteeseen tuleva versio, mikä asetti tekstuurien koolle suuret vaatimukset. Yli 10 000 pikselin tekstuurien välttämiseksi pyrin tekemään materiaalin proceduraalisilla tekstuureilla, mikä tarkoittaa tietokoneen ohjelmallisesti luomaa materiaalia, joka skaalautuu mille resoluutiolle tahansa. Diffuse kanavassa käytin layered tekstuurin avulla proceduraalisen tekstuurin lisäksi kolmea bittikarttatekstuuria, jotka tulivat materiaalin päälle pienellä opasiteetilla, jolloin ne vain rikkoivat hieman proceduraalisen tekstuurin säännönmukaisuutta. Nämä bittikartta tekstuurit projisoitiin malliin, enkä käyttänyt niihin mallin UV-karttaa. Specular highlighteille ja bump-mapille oli molemmille omat proceduraaliset tekstuurinsa. Tekstiin käytin alun perin samaa materiaalia, kuin kotkan jalkoihin. Kuun pintaan käytin kuutekstuuria, joita saa ladattua eri puolilta nettiä.



kuva 8 Kotkan pinnan materiaalin shading network.

3.5 Valaistus

Valaistuksessa päävalona on yksi iso area-light, joka käyttää raytrace (säteen seuranta) varjoja. Tällöin vältetään edelleen tekstuuripohjaisten varjojen resoluutio ongelmilta. Tämä valo oli myös ainoa, joka tuotti varjoja. Muuna valaistuksena oli muutama spotlight (kohde valo), jotka irrottivat kotkaa taustastaan, sekä kaksi negatiivisella arvolla varustettua spotlightia, jotka loivat kuun ja kotkan rajamaastoon tummemman alueen. Näiden negatiivisten valojen kanssa käytin hyödyksi light linkerä, jonka avulla laitoin valot vaikuttamaan vain haluamiini objekteihin. Toinen valoista tuli kuun alapuolelta, jolloin kuu olisi blokanut valon, ellei valon vaikutusta kuuhun olisi estetty. Valojen sijoittelun näkee kuvasta 9.



kuva 9 Valojen sijoittelu

3.6 Renderöinti

Renderoidessani kuvaa julisteeseen käytin Mental Ray rendererin Final gather ominaisuutta, joka simuloi valon kimpoilua kappaleesta toiseen. Varjot olivat säteenseuranta varjoja ja renderöinti resoluutio suurimmillaan 12 000*9 000 pikseliä.

3.7 Animaatio versio

Animaatio versiossa, joka demoon tuli piti alun perin olla useita hitaita kamera-ajoja, joissa kotkapatsas esitellään erittäin tiiviissä lähikuvissa, lopulta laajentuen kokokuvaan. Pulun jalkojen oli myös tarkoitus olla pelkästään jalat jolloin kaikki muu leikkaantuisi kuvasta pois. Pulun oli tarkoitus seisoskella kotkan pääläella ja ulostaa patsaan päälle, jonka jälkeen kamera ajaa nopeasti kokokuvaan ja Iron Sky -teksti lentää kuvaan kameran takaa.

Aloitin työskentelyn tekemällä ohjaajan ohjauksella kamera-ajot animoimalla useamman kameran perus keyframe animointi menetelmällä. Samoin menettelemällä tein tekstin animaation, joskin lisäsin sen loppuun vielä pienen tärähdyksen sen pysähtyessä nopeasta vauhdista.

Pulun mallintamiseen käytin samoja mallinnusmenetelmiä kuin kotkaankin. Ohjaaja olisi tässä vaiheessa halunnut pulusta lentävän, mutta se ei aikataulusta johtuen ollut enää mahdollista.

Pulun teksturointiin käytin Photoshop cs3 extended -ohjelmaa ja sen uusia 3d-layereita. Nämä uudet 3d-layerit mahdollistavat 3d-objektien tuomisen Photoshop-ohjelmaan, jolloin niitä voi pyörittää ja liikutella Photoshop-ohjelmassa. Mallien tekstuureita voi muokata kuten photoshopissa normaalistikin tavallisia 2d-kuvia, mutta aina tallennettaessa päivittyy myös 3d-layerilla olevan mallin tekstuuri. Tämä nopeuttaa työskentelyä hieman, koska silloin ei tarvitse joka kerta ladata tekstuuria uudelleen 3d-ohjelmaan, jossa sitten tarkastella muutosta.

Lisäksi tein pululle normalmapin mudboxilla. Normalmap on kuin bump map joka lasketaan korkearesoluutioisemmasta mallista, ja joka lisää yksityiskohtia malliin bump mapin tavoin. Tämän normal mapin lisäksi käytin toista bump-mappia, jossa oli tekstuurina kohinaa.

Rakensin pululle yksinkertaisen luu hierarkian create joint työkalulla. Tässä vaiheessa oli oleellista että jokaisen nivelen akselit olivat samansuuntaisia, jos näin ei olisi valittaessa useampi nivel kerralla ja pyöräytettäessä niitä taipuisi eri suunnan omaava akseli väärään suuntaan. Laitoin jalkoihin ik-solverit, joiden avulla niiden animoiminen inverse kinematiikkaa hyödyntämällä oli mahdollista. Saadakseni pulun animoinnin helpommaksi tein locator objectin, jonka linkkasin pulun luurangon juureen, jotta se seuraisi liikuteltaessa mukana. Tälle locator objektille annoin add attribute komennolla erilaisia arvoja, joita käytin myöhemmin mallin animoimiseen liitettynä ne ensin set driven keyllä ohjaamaan varsinaisten nivelten arvoja. Tällä menettelyllä sain kaikki luurangolle annettavat keyframet saman nodin alle, jolloin niiden muokkaaminen olin helpompaa. Tein kaksi circle shapea, joista toiseen linkkasin luurangon juuren ja toiseen linkkasin tämän edellisen. Näin ollen pystyin ensimmäisellä circlellä liikuttamaan luurankoa, siten että se ei vaikuttanut ik handlereihin, kun taas jälkimmäistä circleä liikuttamalla koko pulu liikkui. Silloin kun halusin animoida pulun jalkoja hyödyntämällä forward kinematiikkaa, annoin ik-handle objektien fk/ik blend asetukseen keyframe arvon 0, jolloin ik kytetytyi pois päältä ja fk tuli käyttöön, jos taas halusin takaisin ik-toimintoon, annoin blend arvoksi 1.

Varsinainen animaatio tapahtui straight a head menetelmällä, eli aloin liikuttamaan pulua ja siirryin ajassa sitä mukaan eteenpäin kuin tarve vaati, kunnes olin saanut aikaiseksi raaka animaation pulun liikkeistä. Tämän jälkeen hienosäädin tapahtumien kestoa siirtämällä keyframeja ja siirryin tekemään toissijaisia animaatiota pululle. Animaatiossa pulun yli laskeutua kotkan pääläelle, kävellä muutama askel, hiukan nokkaista kotkaa ja ulostaa sen päälle, jonka jälkeen kääntyä juliste esikuvansa asentoon. Askelten animoinnissa käytin hyväksi inverse kinematiikkaa, jolla annoin ik handlerille keyframen, jotta se pysyisi paikallaan ja animoin pulua eteenpäin, jolloin jalka pysyi paikallaan muun pulun liikkessa. Ohjaaja halusi pienen siipien sulkeutumisanimaation pulun laskeutumiseen mukaan, koska en ajan puutteen vuoksi ollut tehnyt kunnollista siipimekaniikkaa linnulle, jouduin animoimaan nämä muutamat frameit käsin mallintamalla. Otin niistä frameista, joissa siipien tuli liikkua pulusta kopion, jonka siipiä muokkasin näyttämään siltä miltä ne näyttäisivät niiden sulkeutuessa. Animoin näille malleille vielä vastaavan liikeradan muutamaan frameen kuin alkuperäiselläkin, jolloin liike epäterävyys ei näyttäisi katkeavan,

Renderointi vaiheessa malleista tekstuureista ja valoista lasketaan renderöinti-ohjelmalla kuvatiedosto. Tämä kuvatiedosto on vastaava kuin mikä tahansa muukin kuvatiedosto, eikä sisällä enää 3d- elementtejä ja on menettänyt kolmiulotteisuutensa ja kolmiulotteisen muokattavuutensa.

Monesti animaatioprojekteissa ohjaaja saattaa muuttaa mieltään jonkun tietyn kohdan suhteen, jolloin koko kuva pitäisi renderöidä uudelleen. Renderöinti on usein aikaa vievää puuhaa, eikä jokaisen pienen muutoksen takia koko kuvan uudelleen renderöinti kannata.

Kuvan eri elementit kannattaakin jakaa erillisiin render layereihin, jolloin haluttaessa muutoksia yhteen osioon kuvassa ei tarvitse renderöidä kuin se osa uudelleen, säästäten näin huomattavasti aikaa. Kappaleiden lisäksi erikseen voidaan renderöidä myös jokaisesta kappaleesta omat renderöinti kerrokset. Jakamalla kappale erilaisiin kerroksiin, kuten diffuse, shadow, specular, mask, transparency saadaan huomattavan paljon enemmän kontrollia kuvaan jälkikäsitteily vaiheessa. Mitä enemmän eri layereita ja passeja on käytössä, sitä enemmän alkutyötä niiden aikaansaamiseksi on, mutta silloin on myös enemmän vaikutusvaltaa lopputulokseen kompositio vaiheessa.

Tässä tapauksessa renderöin kotkan pulun ja tekstin omilla layereillään, enkä käyttänyt erillisiä renderpasseja, koska näiden materiaalit olivat jo ohjaajan hyväksymiä.

Renderöin kotkan ja pulun mayan software renderillä ja tekstin MentalRaylla, koska se käytti materiaaleja, jotka toimivat vain mentalrayn kanssa.

Renderöinnin nopeuttamiseksi renderöin kustakin layerista vai ne framet, jossa sen layerin objektit olivat näkyvissä. Alkuperäisessä versiossa pulu tuli kuvaan vasta 200 framen paikkeilla ja teksti framessa 440. Rendröinnin suoritin mayan batch rendererillä, ettei tarvitsisi turhaan kuluttaa koneen resursseja maya-ohjelman ajamiseen. Batch rendererille voi antaa komentorivillä parametrejä, joiden mukaan se renderöi scenet.

Kirjoitin tekstieditorilla bat-tiedoston, johon annoin renderöinti-parametrit.

Käynnistämällä tämän bat-tiedoston käynnistyi myös renderöinti. Tällä tavoin voi renderöidä monta sceneä peräkkäin ilman, että täytyy tulla väliajoin laittamaan seuraava scene renderöitymään.

4 Tulosten pohdinta ja johtopäätökset

En voi olla täysin tyytyväinen tuloksiin omalta osaltani. Ajan puutteen vuoksi muun muassa pulu on melko hätäisesti tehty. Sen normal map ei kestä lähempää tarkastelua, eikä sen teksturointi muutenkaan. Kaikenkaikkiaan pulu on se osa, jota haluaisin parantaa eniten. Sen lisäksi parantelua vaatii kotka malli, jota en saanut tehtyä aluksi kaavailemallani tavalla, jossa sen yksityiskohdat olisi mallinnettu mudboxilla ja renderöity käyttäen displacement mappeja. Vaikka kotkan malli olikin erittäin korkea resoluutioinen, ei sen resoluutio siltikään riittänyt niihin tiukkoihin lähikuviin jolla sitä kuvattiin. Mainitsin tästä useaan otteeseen mallintaessani kotkaa, mutta huoleeni reagoitiin toteamalla motion blurrin peittävän ongelmat. Inhottaa olla oikeassa tällaisessa tilanteessa.

Logotekstin kanssa ole lukuisia renderöinti ongelmia, johtuen omasta virheestäni käyttäen sitä väärän materiaalin kanssa, ja vielä renderöidessäni sitä väärällä rendererillä.

Kuu mallin kanssa kulutin aivan liian kauan aikaa saadakseni aikaan hyvännäköiset halkeamat kuun pintaa, halkeamat, joita siihen ei lopulta tullut ollenkaan, kun jalat korvattiinkin toisenlaisilla, jotka olivat kyllä huomattavasti paremmannäköiset kuin ensimmäiset versiot.

Kuun pinta julisteessa näyttää aivan hirvittävältä, koska siinä olevan tekstuurin resoluutio ei yksinkertaisesti ole riittävän suuri.

En ole ainoa, joka ei ollut demon tämän hetkiseen tilaan tyytyväinen, koska demon virallinen julkaisu on siirtynyt tämän hetkisten tietojeni mukaan elokuulle.

Johtopäätöksenä voidaan todeta, hiljaa hyvä tulee. Lopussa alkoi tulla kiire ja se näkyi jäljessä

5 Yhteenvetoa ja oman työn arviointia

Työskentely tapahtui enimmäkseen omalla tahdillani omissa oloissani, vaikkakin avotoimistossa työskennellessä omat olot ovat melko suhteellinen käsite. Ohjaajan ollessa paikalla pyysin ohjaajalta mielipidettä miten tulee edetä, jonka jälkeen tein hänen haluamansa muutokset. Ihmismieli on kuitenkin muuttuvainen ja päädyin tekemään monet kohtaukset useamman kerran uudelleen. Osia pitkään työstetyistä kamera-ajoiista ja kohtauksista eivät edes tulleet käyttöön.

Työprosessissa pääsin kokeilemaan itsellekin uusia työkaluja kuten mudboxia ja Photoshop CS3- ohjelman 3D-layereitä. Erityisesti mudbox tulee varmasti pysymään jatkossakin osana työkalu arsenaaliani. Sen tehokkaat 3D-layerit säästivät minut todennäköiseltä hermoromahdukselta, joka olisi seurannut lukuisia kotkan uudelleen muokkauksia, sen jälkeen kun jo monta kertaa hyväksytyn perusmuodon yksityiskohtiakin hienosäädetty.

Työssä kohdatut ongelmat aion kesän myötä analysoida tarkemmin ja etsiä niihin ratkaisut, enkä vain kierrä niitä kuten tämän projektin aikana.

Toisaalta, jos vähentää hiukan itsekritiikkiä, niin näyttäähän tuo juliste ihan mukavalta seinälläni ja lämmittää mukavasti tieto siitä että työnsä jäljen näkee kuitenkin melkoisen moni ihminen. Työskentelypaikkana Energian toimisto oli ihan mukava, jos ei lasketa renderifarmin tuottamaa korkeaa lämpötilaa ja jatkuvaa vetoa, joka tuli avoimesta ovesta, jolla yritettiin pitää lämpötiloja kurissa.

Lähteet

Birn, J. 2006. *Digital lighting & rendering second edition*. New riders

Learning Maya7 The Modelling & Animation Handbook. 2005

Lehtovirta, P. Nuutinen, K. 2000. *3d-sisältötuotannon peruskirja*. Jyväskylä:
Docendo.

Lew, J. 2003. *Learning 3d character animation DVD*

McKinley, M. 2005. *The game artist guide to Maya*. Sybex.

Parrish,D 2002. *3D Lighting and Compositing*. Premiere press.

Liitteet