



Normal map

High poly -mallin pinnanmuotojen siirtäminen low poly -mallille

Viestinnän koulutusohjelma
3D-animaatio ja -visualisointi
Opinnäytetyö
14.10.2010

Silja Ruohomaa

TIIVISTELMÄSIVU

Koulutusohjelma Viestinnän koulutusohjelma		Suuntautumisvaihtoehto 3D-animaatio ja -visualisointi
Tekijä Silja Ruohomaa		
Työn nimi Normal map – High poly -mallin pinnanmuotojen siirtäminen low poly -mallille		
Työn ohjaaja/ohjaajat Kristian Simolin		
Työn laji Opinnäytetyö	Aika 14.10.2010	Numeroidut sivut + liitteiden sivut 40
TIIVISTELMÄ Opinnäytetyö käsittelee 3D-mallien tekstuureita ja sitä, kuinka niillä voidaan luoda illuusio tarkemmin mallinnetusta kappaleesta. Normal map -tekstuurit, joihin työ erityisesti keskittyy, mahdollistavat yksityiskohtaisen mallin pinnanmuotojen näennäisen siirtämisen yksinkertaisemmalle kappaleelle. Opinnäytetyön tavoite on myös kartoittaa yleisesti reaaliaikaiseen renderöimiseen soveltuvia malleja ja tekstuureita. Aihetta lähestytään normaali karttojen ja niiden ominaisuuksien kautta. Opinnäytetyö koostuu tekstuureihin, erityisesti pelialalla käytettyihin, ja normaalikarttoihin tutustumisesta ja sekä kolmen esimerkkityön yksityiskohtaisesta projektikuvauksesta. Esimerkkityöt esittelevät kolme erilaista tapaa luoda normal map Autodesk 3Ds Max -ohjelmalla luodulle mallille. Opinnäytetyössä käsitellään myös normal mapin luomista yleisemmällä tasolla. Normaalikartat ovat tehokas menetelmä luoda yksityiskohtaisia malleja tekemättä niistä liian raskaita nopealle renderöimiselle. Ennen normaalikarttoja käytettiin bump mapping -tekniikkaa, joka jää menetelmiä verratessa normal mapien taakse tarkkuudessaan ja realistisuudessaan. Menetelmän tekee hyödylliseksi se, että se ei tee 3D-scenestä tai objektista niin raskasta kuin mitä oikea korkearesoluutioinen kappale olisi. Normaalikartta pyrkii säilyttämään tarkan mallin yksityiskohdat siirtämättä high poly -mallien heikkouksia, kuten pitkiä renderöimisaikoja ja scenen raskautta low poly -malliin. Metodi on erityisen hyödyllinen reaaliaikaisessa mediassa ja on laajasti käytössä pelialalla.		
Teos/Esitys/Produktio		
Säilytyspaikka Metropolia Ammattikorkeakoulu Kirjasto, Tikkurila		
Avainsanat 3Ds Max, normal map, normaali kartta, tekstuuri, reaaliaikainen 3D		

Degree Programme in Media		Specialisation 3D Animation and Visualisation
Author Silja Ruohomaa		
Title Normal maps – Assigning high poly model's topography on a low poly model		
Tutor(s) Kristian Simolin		
Type of Work Bachelor's Thesis	Date 14 Oct 2010	Number of pages + appendices 40
<p>The thesis focuses on different textures used on a 3D model and how they can create the illusion of a more detailed surface on a simpler mesh. The main subject consisted of normal maps and how they enable the assigning of a more complex model's surface normals for a simpler one. The objective was to examine different ways of creating normal maps and to understand the demands of realtime rendering towards textures and models.</p> <p>This study includes illustration of the background of different textures used in the 3D field, especially those used in medium that require realtime rendering. It includes detailed description of three different ways to create a normal map for use on a model created in Autodesk 3Ds Max and information about creating a normal map in general.</p> <p>Normal mapping is an efficient way of creating light but detailed models. Prior to normal mapping, bump mapping was the method used to accomplish a similar result. Comparison of these approaches shows the advantage of normal maps over bump maps.</p> <p>The benefit of using normal mapping is to obtain the appearance of a high poly model without having to deal with slow processing and long rendering times. Normal maps are commonly utilised in real-time media and these days are essential for video games.</p>		
Work / Performance / Project		
Place of Storage Helsinki Metropolia University of Applied Sciences, Campus Library, Tikkurila		
Keywords 3Ds Max, normal map, texture, realtime 3D		

Sisällys

Johdanto.....	2
1 Käsitteitä.....	3
2 Low poly / High poly.....	5
2.1 Low poly ja high poly -mallien käyttö.....	5
3 Tekstuurit.....	7
3.1 UVW-unwrap.....	9
3.2 Tekstuurityyppejä.....	10
3.3 Normal map vs. bump map	12
4 Pelialalla käytetyt tekstuurit.....	14
5 Normal mapin luominen.....	17
6 Esimerkkitoita.....	20
6.1 Kasvot.....	21
6.2 Rakennus.....	26
6.3 Nuudelitekstuuri.....	33
7 Yhteenveto.....	37
Lähteet.....	39

Johdanto

Opinnäytetyöni käsittelee yksityiskohtien lisäämistä low poly -malleihin tekstuureilla, erityisesti normal mapeilla. Käyn läpi peliteollisuuden ennen ja nyt käyttämiä keinoja tuoda lisää vaihtelevuutta ja yksityiskohtia prosessoinnin kannalta keveinä pidettäviin malleihin. Normal mapeilla tarkoitetaan 3D-mallin pinnalle lisättävää tekstuuria, joka luo illuusion siitä, että mallissa olisi enemmän syvyyttä ja yksityiskohtia kuin siinä todellisuudessa on. 3D-teollisuudessa tarvitaan aina vain tarkempia ja realistisempia malleja. Geometrian tarkkuutta ei kuitenkaan pystytä kasvattamaan loputtomiin tekemättä objekteista liian raskaita koneiden käsiteltäviksi tai käyttäjien hallittaviksi.

Noin 15 vuotta sitten alettiin toteuttaa työkalua, joka mahdollistaisi korkearesoluutioisten meshien pinnanmuotojen siirtämisen yksinkertaistettujen mallien päälle. Tämä mahdollisti huomattavasti yksityiskohtaisemmat 3D-mallit myös pelialan ja muiden reaaliaikaisten medioiden käyttöön. Näille aloille yksinkertaistetut mallit ovat olleet nopeiden prosessointiaikojensa takia välttämättömiä. Sitä ennen tämä on ollut esteenä peleissä käytettyjen mallien ulkonäön monimuotoisemmaksi kehittymiselle. Pelkkä normaalien siirtäminen mallilta toiselle ei lisää yksinkertaisemmalle geometrialle merkittävästi laskenta-aikaa renderöidessä, mutta luo silti realistisesti valojen vaihteluun reagoivan ja moninkertaisesti yksityiskohtaisemman ja mielenkiintoisemman pinnan.

Ennen normal mapeja peliteollisuudessa turvauduttiin bump mapeihin, jotka olivat alan ensimmäisiä tekstuureita, jotka reagoivat valaistukseen edes etäisesti kuten oikeat pinnanmuodot. Vaikka bump mapin valmistaminen on huomattavasti nopeampaa kuin normal mapin, sen tarkkuus ja realismi jää kauas taakse, kun tekstuureiden tark-

kuutta verrataan toisiinsa. Vertailen normal mapien ja bump mapien ominaisuuksia syvällisemmin seminaarityössäni. Ensin avaan joitakin työssäni käyttämiä termejä ja keskityn sen jälkeen tarkan ja epätarkan geometrian eroihin luvussa Low poly / High poly. Käsittelen erilaisia tekstuureita pintapuolisesti, ja keskityn sitten tarkemmin peliteollisuudessa käytettyihin tekstuureihin ja niiden kehitykseen vuosien varrella.

Viimeisenä käyn kohta kohdalta läpi kolme erilaista tapaa toteuttaa normaalikartta 3Ds Maxilla mallinnetun low poly -mallin päälle. Nämä kolme esimerkkityötä ovat vanhan miehen kasvot, kerrostalo sekä nuudeliharkosta valokuvaamalla luotu tekstuuri. Valitsin nämä esimerkit, koska halusin toteuttaa sekä orgaanista että elotonta pintaa ja tuottaa normal mapeja sekä mallintamalla että valokuvaamalla ja kuvankäsittelyohjelmaa käyttäen. Vaikka tekniikka on tarkoitettu erityisesti liikkuvan kuvan ominaisuuksien parantamiseen, yritän lopputyössäni esitellä tekstuurien monimuotoisuutta ja ominaisuuksia kuvin ja selityksin.

Valitsin aiheen, koska olen kiinnostunut pelialasta. Halusin oppia peliteollisuudessa käytetyistä keinoista, jotka auttavat pitämään mallit tarpeeksi keveinä reaaliaikaiseen renderöimiseen samalla, kun niiden ulkonäkö hipoo täydellistä realistisuutta, tarpeen niin vaatiessa. Kiinnostukseni heräsi pelikonsolien kuten Sony Playstation 3 ja Microsoft Xbox 360 julkistamisen jälkeen, kun huomasin pelien muuttuneen pienessä ajassa graafisilta ominaisuuksiltaan mielestäni huomattavasti aiempaa vaikuttavammiksi.

Haluan työlläni valaista yleisesti reaaliaikaisessa mediassa käytettyjä keinoja luoda ulkonäöllisesti toivotunlainen lopputulos ja koota yhteen tietoa normaalikarttojen käyttötarkoituksesta ja luomisesta.

Toteutin työn pääosin Autodesk 3Ds Max 10:llä sekä Autodesk Mudbox 10:llä. Tekstuurien käyttöön käytin Adobe Photoshop CS4:ää sekä Gimp 2.6:ta.

1 Käsitteitä

Aluksi avaan joitain termeistä, joita tulen käyttämään työssäni.

Polygoni on mallin yksi pinta. Se on yleisimmin neliön muotoinen. Janat, jotka kiertä-

vät yhtä polygonia, ovat tässä työssä suomentamatta **edgejä**. Polygonit muodostavat **meshin**, eli mallin, jolla on geometriaa. Jokainen kahden tai useamman edgen muodostama haarakohta on verteksi.

Modifierillä tarkoitetaan erilaisia muokkaimia, joilla muokataan geometristä mallia. Esimerkiksi turbosmooth on modifier, joka jakaa mallin tiheämmäksi verkoksi laske-malla olemassa olevien edgejen keskiarvot ja lisäämällä sitten niiden väliin uuden edgen. Tämä pyöristää ja pehmentää mallia kauttaaltaan muuttamatta mallia kuiten-kaan pysyvästi, ellei modifieriä **collapsoida**, eli liitetä malliin pysyväksi osaksi.

Extrude on työkalu, joka lisää valitulle pinnalle syvyyden.

Tekstuuri on mallin päälle lisättävä 2D-kuva, joka yhdessä geometrian ja valaistuksen kanssa määrittelee mallin lopullisen ulkonäön. Tekstuuri määrittää muun muassa mallin värin. **Texture map** on kuvatiedosto, joka sisältää mallin tekstuurin. Sitä voidaan käsitellä periaatteessa missä tahansa kuvankäsittelyohjelmassa. Bittikartan sisältämä tekstuuri tarvitsee tiedon siitä, miten se asettuu kolmiulotteisen pinnan päälle ja näitä tietoja sanotaan **pinnoituskoordinaateiksi** (Lehtovirta & Nuutinen 2000). Yksi tapa määrittellä nämä on **UVW-unwrap**, jota käytettäessä malli "avataan" litteäksi pinnaksi, jolle voi osoittaa oikeanmuotoisen värikartan kuvankäsittelyohjelmassa.

Scene on 3D-ohjelman kolmiulotteinen työtila, jossa mallia muokataan. Työtilassa tietokoneen laskentatehojen takia valaistus ja tekstuurit toistuvat yleensä yksinkertais-tettuina. Yksinkertaistus on tarpeen, jotta työskentely olisi mahdollisimman nopeaa raskaitakin malleja käsitellessä. Ohjelman asetuksista voi määrittää scenen näkymän tarkkuuden.

Renderöiminen tarkoittaa uloslaskemista, eli tässä tapauksessa rasteroimista. Scenen ominaisuudet lasketaan yhteen ja lopputuloksena on pikseleistä muodostuva rasteri-kuva. Koska scene ei nykyresurssien takia pysty toistamaan mallia täysilaatuisena, val-miit kuvat tai animaatiot on renderöitävä erikseen ohjelmasta ulos. aikana tietokone laskee tekstuurit ja valaistukset ja muut efektit mallin lisäksi ja muodostaa kokonaisuudesta valmiin kuvatiedoston.

Sculptaus on nimensä mukaisesti 3D-ohjelmalla tapahtuvaa "kuvanveistoa", jossa tiuhageometristä pintaa muokataan erilaisilla työkaluilla.

2 Low poly / High poly

Low polyistä ja high polyistä puhuessa tarkoitetaan 3D-malleja, joiden tarkkuus vaihtelee matalan eli yksinkertaisen sekä korkean eli yksityiskohtaisemman geometrian välillä. Kuten nimestä voi päätellä, high polyllä tarkoitetaan sitä että mallissa on suuri määrä polygoneja. Mitä enemmän geometriaa kappaleella on, sitä tarkempi se on. Esimerkiksi tavallisessa kuutiossa näitä polyjä on 6. Low polyssä on niin vähän polygoneja kuin mahdollista, ilman että mallin muodosta poistetaan kuitenkaan mitään oleellista.

Ei ole mitään yleistä sääntöä sille, milloin malli on yksiselitteisesti joko low tai high poly. Toisille 3D-grafiikan tuottajille ja heidän puutteellisemmille resursseilleen high polyn raja on huomattavasti alempana kuin suurimpien animaatioelokuvien tuottajille, esimerkiksi Dream Worksille. Yhtenä low polyn määritelmänä voi pitää sitä, onko toivottuja yksityiskohtia jouduttu hieman karsimaan jo suunnitteluvaiheessa mallin geometriasta, että se olisi tarpeeksi kevyt renderöitäväksi siihen tarkoitukseen, mihin sitä aiotaan käyttää. Usein still-kuvia varten mallin voi tehdä juuri niin monimutkaiseksi kuin haluaa ja jos scene on liian raskas, lopullisen kuvan voi koota eri osissa renderöidystä mallista.

2.1 Low poly ja high poly -mallien käyttö

Minkä takia kaikkia malleja ei sitten tehdä niin yksityiskohtaisiksi kuin mallinnusohjelmilla on mahdollista? Kokonaisuutta rakentaessa tulee ottaa huomioon käytettävän laitteiston käsittelykyky: liian raskasta mallia rakentaessa työ voi kaatua siihen, ettei tietokone enää pysty laskemaan ja päivittämään työtilaa niin nopeasti kuin työskentely sitä vaatisi. Liian yksinkertaistetut mallit taas eivät nykyaikana tahdo kestää tarkempaa tarkastelua. Kun 3D-grafiikan alkutaipaleella kaikki mallit olivat yksinkertaisempia, kuluttajat hyväksyivät sen, että hahmot ja rakennelmat olivat pelkistettyjä, koska

tekniikka oli vielä uutta ja ihmeellistä. Tietokonepelejä on ollut olemassa jo kymmeniä vuosia, mutta edes etäisesti realistisia pelimaailmoja on saatu luotua 3D:llä vasta viimeisen 15 vuoden aikana. (Puhakka 2008, 24.)

Low polyjä käytetään pääosin reaaliaikaisiin sovelluksiin kuten videopelihin, koska tietokoneen täytyy pystyä laskemaan ja muodostamaan kuvaa reaaliajassa. Ensimmäiset 3D-ohjelmat pystyivät käsittelemään vain hyvin yksinkertaisia malleja ja mallinnukset olivat sekä muodoltaan että tekstuureiltaan hyvin pelkistettyjä nykyisiin malleihin verrattuna. Kehitys on ollut nopeaa, ja varsinkin pelien 3D-maailmat ja -hahmot ovat muuttuneet monta kertaa yksityiskohtaisemmiksi. 10 vuotta sitten grafiikoiltaan erinomaisiksi lasketuissa peleissä ei haitannut, vaikka päähenkilöllä ei olisi ollut geometristä nenää, vain litteä naama, johon oli lisätty tekstuurilla kasvot. Nykyaikana malleissa pyritään täydelliseen realismisuuteen, ja usein tässä onnistutaankin hyvin pieniä ongelmia lukuun ottamatta.

Mallin raskauden lisäksi liikkuvaa kuvaa varten luotavissa kappaleissa on otettava huomioon se, riittääkö mallin tarkkuus toivottujen liikkeiden toteuttamiseen. Jos hahmojen käsien halutaan taipuvan, kyynärpäiden kohdalle on luotava tiuhempaa geometriaa. Tämä estää sen, ettei malli rypistyisi tai menisi kasaan vääristä kohdista ja liian radikaalisti. Tarkempi geometria rajaa liikkeen aiheuttaman muutoksen pienemmälle alueelle, ja koska polygonit eivät taivu, tarvitaan niitä liikkuvien osien välille enemmän. Jos halutaan tehdä taite, pitää luoda vähintään kaksi polygonia, joiden välille taite voi tulla. Tämä tekee tasapainottelusta kevyen ja raskaan mallin välillä vieläkin monimutkaisempaa.

Vaikka todella yksityiskohtaisten työtilojen, eli mallinnettujen kokonaisuuksien, renderöiminen on hidasta, todella raskaitakin malleja käytetään aina vain enemmän ja enemmän. Esimerkiksi visualisointeihin ja still-taidekuviin voidaan mallien polygonimääriä kasvattaa huimiin lukemiin, koska rasteroiminen tulee suorittaa näitä medioita varten vain kerran. Still-kuvaa varten mallintaminen on muiltakin osin hyvin erilaista kuin liikkuvaa kuvaa varten työskentely. Still-kuvassa näkyvien kappaleiden ei tarvitse näyttää täydellisiltä kuin yhdestä suunnasta: siitä kuvakulmasta, mistä kuva aiotaan renderöidä. Kunhan varjot ja heijastukset toimivat, ei ole väliä, onko mallinnetun auton takana vain tyhjää, vai auton realistisesti mallinnettu toinen puoli.

Mallien raskauden lisäksi myös tekstuureiden suhteen voi optimoinnin kanssa olla hieman suurpiirteisempi, kun kyse on still-kuvasta. Peleissä tekstuureita on kaiken muun lisäksi pidettävä keveinä ja pieninä, kun taas still-kuvassa tekstuuritiedostot voivat olla suurempia. Tekstuureita käsitellään pidemmin seuraavassa luvussa. Tuottavan yhtiön resursseista riippuen animaatioitakin toteutetaan yhä useammin malleilla, joihin on käytetty hyvinkin korkeita polygonimääriä. Jos yhtiö on tarpeeksi suuri, sillä on varaa renderifarmeihin ja uusimpiin tietokoneisiin, jotka pystyvät pyörittämään high poly -malleja reaaliajassa. Renderifarmeilla tarkoitetaan pelkästään renderöimiselle omistettuja tietokoneita. Suurilla firmoilla on varaa ylläpitää useaa tietokonetta pelkkää laskemista varten. Jotkut yritykset taas vuokraavat renderöintiapua aiheeseen erikoistuneilta yhtiöiltä. Uutuuspelien välianimaatiot ja koko illan animaatioelokuvat onkin usein toteutettu hyvin tiuhageometrisillä malleilla niiden raskaudesta huolimatta.

Reaaliaikaisissa sovelluksissa, kuten peleissä sekä animaatioissa, ongelmaksi muodostuvat laskemisajat. Jos jokaisen kuvan laskemiseen, menee puoli tuntia tai edes puoli minuuttia, on peliä mahdotonta pyörittää millään olemassa olevalla kotikoneella. Tästä syystä low poly -mallit ovat välttämättömiä. Animaatiot ovat hieman anteeksi-antavampia, koska niitä ei rasteroida reaaliajassa. Kun peleissä koneen on pystyttävä näyttämään koko ajan päivittyvä kolmiulotteinen maailma tekstuureineen ja valoineen, animaatioissa valmis tuotos renderöidään ulos periaatteessa vain kerran. Tämän jälkeen animaation liikkuvaa kuvaa käsitellään tavallisena kuvasarjana, eikä 3D-tiedostona. Myös animaatioita tuotettaessa on pysyttävä maltillisena – pidettävä mallit yksinkertaisina ja laskenta-ajat pieninä. Jos kone joutuu laskemaan yhtä kuvaa lähes puoli tuntia ja koko tuotos on noin kahden tunnin mittainen, voi animaation renderöiminen viedä lähemmäs 2 000 000 laskentatuntia.

3 Tekstuurit

Tekstuureilla, eli mallin pinnalle heijastettavilla kuvilla, voidaan usein toteuttaa se, mihin pelkkä yksityiskohtainenkaan malli ei pysty. Tekstuuri-nimitys juontaa juurensa kuvien alkuperäisestä käyttötarkoituksesta: kuvilla pyrittiin luomaan illuusio, että mallin pinnassa oli hienorakennetta eli tekstuuria. Koska kyseessä on kuitenkin vain kaksiulotteinen kuva, oikeampi nimitys olisi image mapping tai color mapping. Suomeksi voitaisiin puhua värikartoista. Sana on kuitenkin jo vakiintunut käyttäjien keskuudessa,

eivätkä muut termit todennäköisesti korvaa sitä tulevaisuudessakaan. (Puhakka 2008, 206.)

Tekstuureilla lisätään malleihin väriä, yksityiskohtia, heijastuksia ja läpinäkyvyyttä, ja oikein toteutettuna niillä voi luoda illuusion tarkemmin mallinnetusta kappaleesta. Tekstuuri voi olla mikä tahansa 3D-ohjelman tukema kuvatiedosto. Mitä monimutkaisempi malli on, sitä vaikeampi tekstuuri on asetella sen pinnalle; 2D-kuvat kun eivät kolmiulotteisuutta tue. Tekstuureissa on otettava huomioon samoja perussääntöjä raskauden suhteen kuin geometriassa. Tekstuurina käytettävän kuvan koon kasvaessa myös laskenta-aika kasvaa. Jos kuva taas on liian pieni, sen tarkkuus ei riitä toivottuun lopputulokseen ja kuva voi pikselöityä rumasti mallin päällä.

Tekstuureita tulee optimoida sen mukaan, mihin tarkoitukseen ne tulevat. Niiden tulee olla tarkkuudeltaan yhteneväisiä, koska muuten mallit, joissa on muita epätarkemmat tekstuurit, erottuvat joukosta negatiivisessa mielessä ja voivat näyttää keskeneräisiltä tai huonosti toteutetuilta. Joissain tapauksissa tosin samaan työtilaan voi ujuttaa hyvinkin eritasoisia kuvatiedostoja. Ne on kuitenkin aseteltava hyvin huolellisesti ympäristöön niin, että niitä ei pääse tarkastelemaan liian läheltä.

Hyvänä sääntönä voi pitää sitä, että mitä kauempana ja mitä vähemmän aikaa tekstuuri näkyy, sitä epätarkempi se voi olla, ja toisin päin. Kohteiden, joita tarkastellaan läheltä ja pitkiäkin aikoja, on oltava kaikin puolin toimivia ja huolellisesti toteutettuja. Sama sääntö pätee myös mallintamisen tarkkuuteen: jos malli on vain pieni osa suuressa kokonaisuudessa, sitä ei kannata toteuttaa kovin tarkkaan, koska yksityiskohtia ei ehdi nähdä. (Lehtovirta, Nuutinen 2000.)

Jos mallia näytetään vuoroin kaukaa ja läheltä, mallin ollessa kaukana, kuluisi turhaa prosessointiaikaa siihen, että kaukana horisontissa olevalla kappaleella on sama yksityiskohtainen tekstuurikartta käytössä kuin silloin, kun sitä tarkastellaan pienen matkan päästä. Tätä varten reaaliaikaisissa sovelluksissa käytetään karttaa nimeltä mipmap. Tämä tekstuuritiedosto vie enemmän tilaa kuin tavallinen värikartta, koska siihen on tallennettu erikokoisia ja eri tarkkuudella toteutettuja kuvia käytettäväksi sitä mukaa, kuinka lähellä kameraa kappale on. Lähellä kameraa mipmapin suurin ja yksityiskohtaisin tekstuuri on käytössä, mutta mitä kauempaa kappaletta näytetään, mipmap vaihtuu portaittain aina vain pienemmäksi ja pelkistetyimmäksi kuvaksi (Wikipedia.org

2010b).

Kaikkein kevyimmät tekstuurit saadaan luomalla tilettävä, eli rajattomasti toistuva bittikartta. Sellaista käytettäessä kuvatiedoston tarvitsee olla pienehkö ja siten kevyemmin laskettava. 3D-ohjelmilla on myös useita matemaattisesti laskettavia materiaaleja, joissa on enemmän vaihtelua kuin tilettävissä kuvissa, mutta ne toistuvat silti tasalaa-tuisina mallin pinnalla. Matemaattisesti laskettava tekstuuri on laskennan kannalta usein bittikarttoja kevyempi tekstuurivaihtoehto, kunhan siinä ei ole muita ominaisuuksia kuin väritys. Keveys riippuu tietysti myös laskettavan materiaalin monimutkaisuudesta. Valmiit materiaalit, joissa on heijastuksia tai läpinäkyvyyttä, ovat laskennallisesti hyvin raskaita. Tavallisesti 3D-malleissa ja niiden tekstuureissa on ongelmana se, että ne ovat liian puhtaan ja täydellisen näköisiä. Varsinkin samanlaisena toistuvassa kuvassa on vaarana, että se luo liian tasaisen lopputuloksen. Luonnossa harvat asiat ovat täysin symmetrisiä, säröttömiä ja tasaisen värisiä (Lehtovirta, Nuutinen 2000). Malleihin lisätään epätäydellisyyttä ja rosoisuutta, jotta yleisö hyväksyisi näkemänsä.

3.1 UVW-unwrap

Miten kaksiulotteinen kuva sitten saadaan toistumaan toivotulla tavalla mallin pinnasta? Tätä varten on erilaisia työkaluja, joiden tehtävänä on unwrapata, eli levittää mallin pinta 2D-kuvaksi. Esimerkiksi modifier UVW-unwrap avaa mallin toivotuista kohdista ja levittää sen kuvatiedostoksi muuntamista varten litteäksi pinnaksi. UVW-nimitys tulee tekstuurikordinaateista u,v,w (vertaa 3D-työtilann x,y,z -kordinaatteihin).

Avaamisen voi tehdä parilla eri tavalla. Käytän itse box-mappingia toisessa esimerkkityössäni, koska mallin muoto on lähellä laatikkoa. Tällä asetuksella unwrappaus kuvaa sivun kerrallaan edestä, takaa, vasemmalta, oikealta ja ylhäältä ja alhaalta ja asettelee ne UVW-pohjalle. Se tallennetaan kuvatiedostona, jonka jälkeen se on valmis muokattavaksi missä tahansa kuvankäsittelyohjelmassa. Samalla malliin tallentuu tieto siitä, mihin mikäkin kohta tästä kuvatiedostosta asettuu: koordinaatit kertovat, mikä kuvatiedoston kohta asettuu minkäkin verteksin päälle. Tekstuurin voi määrätä mallin päälle myös avaamatta kuvaa erilliselle UVW-templatelle. Tekstuurin voi käskeä kiertämään mallia kuin palloa, laatikkoa tai vaikkapa litteää, neliskanttista pintaa. Tekstuurin asettelu näin on yksinkertaisempaa, mutta sen tarkkuus ei useinkaan ole riittävä ja malli on

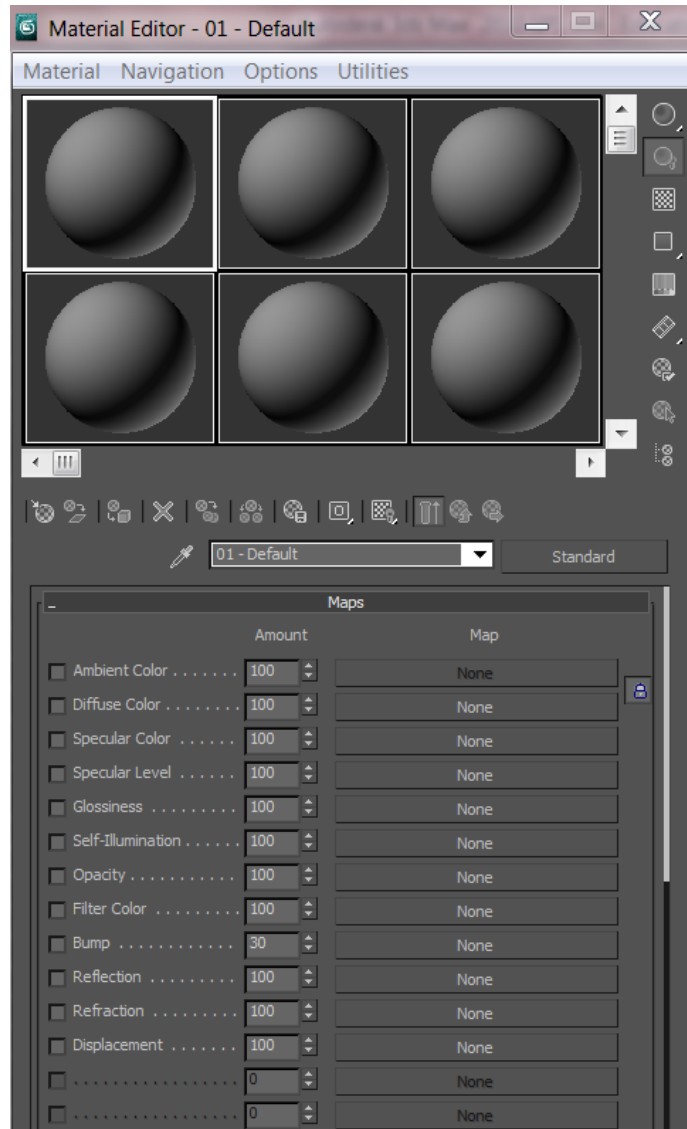
avattava teksturointia varten.

Toista työtä, eli ihmiskasvoja, teksturoidessani käytin pelt mapia, joka toimii paremmin orgaanisilla muodoilla, joilla ei ole selkeitä sivuja, pohjaa tai muita tahkoja. Pelt mappingissa mallia venytetään ja kutistetaan eri kohdista niin kauan, että sen kaikki polygonit on saatu vierekkäin kaksiulotteiseen kuvaan. Lisäämällä saumakohtia, joiden kohdalta pinnan annetaan katketa, malli pyritään avaamaan niin pienellä venyttämällä kuin mahdollista. Kova venyttäminen tai kutistaminen on pahasta, koska mitä enemmän polygoni on tietyssä kohdassa joutunut venymään tai kutistumaan verrattuna mallissa olevaan alkuperäiseen polygoniin, sitä enemmän tekstuurikin tulee venymään tai kutistumaan, kun kuva siirretään mallin päälle.

Viimeisessä työssä mallia ei tarvinnut unwrapata laisinkaan, koska malli on niin yksinkertainen. Pelkkä neliskulmainen plane ei usein tarvitse modifieriä määrittämään, miten tekstuurin tulee sen pinnalla toistua. Jos kuva ei kuitenkaan toistu toivotusti, planar UVW-mappingilla tekstuuri heijastetaan suoraan tasaisen levyn pintaan.

3.2 Tekstuurityyppejä

Pelkän värikartan lisäksi malliin voi lisätä useita erilaisia materiaaleja, jotka määrittävät muun muassa pinnan kiiltoa tai epätasaisuutta. Avaan tässä kohta kohdalta joitain Maxin tarjoamia materiaalivaihtoehtoja, jotta normaalikarttojen tarjoama hyöty olisi helpompi ymmärtää ja että niitä olisi helpompi verrata muihin tarjolla oleviin vaihtoehtoihin (Kuva 1).



Kuva 1: 3Ds Maxin Material editorissa voi määrittää mallille erilaisia materiaaleja ja muokata niiden ominaisuuksia.

Ambient color määrittää materiaalin värin silloin, kun siihen ei osu lainkaan suoraa valoa. Diffuse colorilla tarkoitetaan mallin pääväriä. Diffuse mapina käytetään kuvatiedostoa, jossa ei ole muuta tietoa kuin se, minkälainen värikartta mallin päälle tulee. Se toistuu työtilassa, kuten muutkin tekstuurit, hieman pelkistettyinä, mutta rasteroidessa sen pitäisi olla lähellä alkuperäistä kuvatiedostoa väriykseltään ja muuten ulkoisesti. Värikyseen vaikuttaa radikaalisti scenen valaistus. Jos valo on väriltään muuta kuin valkoista, muuttuvat tekstuurin värit – kuten reaali maailmassakin.

Specular color määrää mallin värin kohdissa, jotka ovat kirkkaassa, suorassa valossa. Specular level ja glossiness määrittelevät tekstuurin kiillon ominaisuuksia.

Self illuminationilla tarkoitetaan mallin itsensä luomaa valoa. Jos mallin halutaan kiiltävän pimeässä, siihen asetetaan self illumination, joka kattaa koko mallin, tai sitten self illumination map, joka kertoo ohjelmalle, mitkä kohdat mallista sen tulee valaista ja mitkä ei. Max käyttää tähän tarkoitukseen mustavalkokuvia, eli valkoiset kohdat self illumination mapissa loistavat kirkkaimmin ja mustat eivät tuota valoa ollenkaan. Näiden kahden välille jäävät kaikki harmaan eri sävyt edustamaan erisuuruista valovoimaa.

Kuten muissakin tekstuureissa, opacity voi kattaa koko mallin tai vain sen tiettyjä osia. Opacity map määrää mallissa läpinäkyvät ja läpinäkymättömät kohdat. Kuten self illumination map, opacity map koostuu mustavalkokuvasta, jonka ohjelma muuttaa kappaleeseen laitettaessa erivahvuiseksi läpinäkyvyyksiksi. Scanline-renderöijää käytettäessä valkoinen ei päästä valoa läpi, mustan ollessa sen sijaan täysin läpinäkyvä. Bump map muuntaa harmaasävykuvan pinnanmuodoiksi lisäten geometriaan yksityiskohtia, jotka reagoivat valoon kuten oikeat korkeudenvaihtelut. Käytännössä efekti ei ole kuitenkaan kovin tarkka tai realistinen. Bump mapiä käsitellään yksityiskohtaisemmin seuraavassa luvussa.

Reflection map saa ympäröivät pinnat heijastumaan mallista. Täysillä arvoilla sen tuottama pinta on peilimäinen. Refraction määrittelee, miten mallin läpi kulkeva valo taittuu. Tähän toimintoon tarvitaan siis lisäksi opacityä, läpinäkyvyyttä. Refractionin avulla esimerkiksi läpinäkyvän pallon saa vääristämään sen läpi näkyvää taustaa samalla tavalla kuin lasipallo vääristäisi asioita luonnossa. Displacement map muuttaa harmaasävykuvan oikeaksi geometriaksi mallin pinnalle. Se tekee näennäisesti saman asian kuin normal map, mutta käytännössä se tekee mallista huomattavasti raskaamman käsitellä.

3.3 Normal map vs. bump map

Normaali karttoja käytetään esimerkiksi peligrafiikoissa, joissa mallien on oltava mahdollisimman kevyitä ja reaaliajassa renderöityviä, ja joiden on kuitenkin nykyaikana oltava hyvin yksityiskohtaisia ja mahdollisimman realistisia. Samalla kun vaatimukset pelien grafiikoista kovenevat kovaa vauhtia, eivät tietokoneet ja konsolit pysy

tehoiltaan kehityksen perässä. Kun esimerkiksi pelihahmojen ihohuokokset ja ryypt tulee näkyä reaaliajassa, tulisivat yksityiskohtaiset mallit liian raskaiksi koneille, jos kaikki ulokkeet ja kuopat toteutettaisiin mallintamalla ne kappaleen pinnalle oikeasti.

Normal map saa nimensä normaaleista, joiden perusteella se laskee, mitä se heijastaa katsomatta heijastuspintoja suoraan luodusta geometriasta. Normaali on vektori, joka on kohtisuorassa tasoon nähden. (Puhakka 2008, 41) Kun yksinkertaiselle pinnalle luodaan normal mapia varten lisää geometriaa, 3D-ohjelma tallentaa yksityiskohtaisemman mallin normaalien tiedon ja liittää sen pelkistetyn mallin pintaan. Eli tallentamatta kaikkea raskaamman mallin geometriaa ohjelma pystyy ottamaan pelkästään sen tiedon, kuinka paljon valoa ja missä kulmassa mallin mikin osa heijastaa kameraa kohti. Tästä syntyy vaikutelma, että kyseessä on todellisuutta yksityiskohtaisempi kappale. Kun pinnan valoisuuksilla on enemmän vaihtelua, malli näyttää monimutkaisemmalta.

Bump map on Maxin oletustekstuurityökaluista lähimpänä normaali karttaa, mutta sen toimintaperiaate on hyvin paljon yksinkertaisempi. Bump map luo 2D-kuvan avulla illuusion sileän geometrian pintaan siitä, että siinä olisi enemmän epätasaisuutta. Toisin kuin normal map, bump map ei oikeasti sisällä mitään tietoa eri syvyyksistä, vaan se luo illuusion rend harmaasävykuvan värimuutokset pieninä korkeuseroina. Normal map sisältää informaatiota pinnanmuodoista ja se reagoi valon kanssa lähes täysin kuten oikeat geometriset muodot. Bump map luodaan, kuten monet muutkin tekstuurit, harmaasävykuvan avulla. Se on normal mapiin verrattuna paljon nopeampi ja yksinkertaisempi toteuttaa, joten epätarkempien pinnanvaihtelujen riittäessä bump map on hyvinkin vartenotettava vaihtoehto. Bump mapia käytettäessä 3D-ohjelma muuttaa sille osoitetun mustavalkoisen kuvan mukaan mustat osiot kuopiksi ja valkoiset osiot kohoumiksi, niiden välille jäävien harmaan eri sävyjen jäädessä ääripäiden välisiksi siirtymiksi. Bump mapit erottelevat muodot vain korkeussuunnassa ja ovat siksi epärealistisemmän näköisiä kuin normal mapeilla luodut pinnanmuodot, jotka ottavat huomioon ulottuvuuksissa y-ulottuvuuden lisäksi myös x:n ja z:n. Normal map on huomattavasti hitaampi toteuttaa kuin bump map, mutta renderöintiajoissa ei ole merkittäviä eroja ja normal mapista saatava hyöty korvannee moninkertaisesti siihen tuhraantuneen ajan.

Vain pikselien väliset erot muuttavat pinnan ulkonäköä bump mapia käytettäessä, toisin

kuin normal mapeissa. Bump mapin kompastuskivi on sen kaikesta huolimatta liian litteä lopputulos: vaikka valo reagoi eri pintoihin eri tavalla, lopputulos oli silti hyvin tasainen. Ongelmana on se, että yksittäinen pikseli ei anna tietokoneelle tarpeeksi tietoa siitä, mitä sen pitäisi seuraavaksi renderöidä. Bump mapin harmaa, musta tai valkoinen alue itsessään ei aiheuta mitään efektiä, jos sen ympärillä ei ole huomattavasti tummempaa tai vaaleampaa väriä. Yksityiskohdat ovat riippuvaisia ympäröivästä tekstuurista ja rasteroimisesta tulee näin huomattavasti epätarkempaa kuin silloin, kun jokainen pikseli sisältää tiedon pinnanmuodoista huolimatta siitä, mitä informaatiota viereisestä pikselistä löytyy.

4 Pelialalla käytetyt tekstuurit

Ensimmäinen "3D"-tietokonepeli, 3D Monster Maze, jonka maailmaa ei voi hyvällä tahdollakaan kutsua realistiseksi, julkaistiin vuonna 1982. Sen kolmivärinen labyrinttimaailma piti yksinkertaisuudestaan huolimatta pelaajat tiukasti otteessaan. Peliä ei oltu toteutettu 3D-ohjelmalla, mutta se oli ensimmäinen peli, jossa pelikentällä liikuttiin sivu- ja syvyys suunnassa (Kuva 2). Nykyaikana, kun 3D-grafiikka pystyy niin paljon huikeampiin suorituksiin, ei pelkistetty grafiikka enää myy niin hyvin kuin ennen, joitain poikkaustapauksia lukuunottamatta.



Kuva 2: 3D Monster Maze (New Generation Software) oli ensimmäinen tietokonepeli, jossa seikkailtiin kolmiulotteisessa maailmassa, joskaan peli ei vielä hyödyntänyt 3D-grafiikkaa (Wikipedia 2010).

Varhaisissa aidosti kolmiulotteisissa peleissä tekstuureilta käytettiin vain värikarttoina.

Läpinäkyvyyksiä, heijastuksia ja valaistukseen reagoivia rosoisuuksia ei vielä pystytty toteuttamaan reaaliaikaisesti. Valot, varjot ja monimutkaiset pinnanmuodot yritettiin toteuttaa yhdellä värikartalla. Ensimmäinen kehitysaskel oli renderöidä valot tekstureihin niin, että lopputulos näytti jo hieman realistisemmalta. Se lisäsi tasaisiin pintoihin varjoja ja vaaleampia kohtia tuomaan eloa. Suuri puute tässä metodissa oli se, että tuli valo mistä suunnasta tahansa, tekstuuriin yhdistetty valaistus ei muuttunut miksiäkään. Kun pelitekstuureihin pystyttiin vähän ajan päästä lisäämään bump map, eli käytössä oli samaan aikaan sekä värikartta että informaatiota pinnanmuodoista, pelimaailmoiden tarkkuus otti kunnan askeleen kohti realistisempaa lopputulosta.

Reaaliaikaisessa renderöimisessä joudutaan huijaamaan enemmän ja kiertämään fysiikan ja 3D:n lakeja toivotun lopputuloksen aikaansaamiseksi verrattuna medioihin, joissa renderöintiajan ei tarvitse pysyä minimissään. Siinä, missä still-kuvaa varten pystytään renderöimään oikeasti luotu high poly malli, jonka kiiltävällä pinnalla on heijastuksia ympäristöstä, reaaliaikaisessa maailmassa monimutkainen pintamateriaali täytyy renderöidä tekstuuriksi ja heijastukset lisätään etukäteen renderöimällä. Pinnasta ei heijastu mitään reaaliajassa, vaan kyseessä on vain efekti joka luo tällaisen illuusion. Vaikka 3D:ssä useat ominaisuudet eivät vastaa reaali maailman vastaavia, videopeleissä fysiikan lakien seuraaminen on vieläkin vaikeampaa ja laskenta-ajan säästämiseksi perusasioita kuten valo ei pystytä simuloimaan realistisesti vaan käytetään keinoja, jotka saavat lopputuloksen vain näyttämään siltä, että asiat toimivat kuten oikeassa maailmassa.

Kuten aiemmin mainitsin, bump mapien luoma illuusio lisägeometriasta on hyvin suurpiirteinen. Käytännössä tämä vähentää rasteroidun kuvan tarkkuutta huomattavasti ja kaikkein pienimmät yksityiskohdat katoavat tyystin. Bump map tunnistaa vain korkeussuunnassa tapahtuvan muuntelun, mutta normal mapiin tallentuu informaatio myös kahdesta muusta ulottuvuudesta. Normal mapin tarkkuus on usein tarpeeksi korkea siihen, ettei sitä erota oikeasta geometriasta. Vain mallin siluetti paljastaa huijauksen pysyessään muuttumattomana. (Hastings-Threw 2007) Bump map antoi reaaliaikaisille 3D-sovelluksille mahdollisuuden ottaa käyttöön reaaliaikaisen valaistuksen. Normal map sen sijaan teki valojen käytöstä huomattavasti kannattavampaa. Se myös mahdollisti realistisemmän ja vaihtelevamman peliympäristön luomisen, samalla kun työtila pysyi silti tarpeeksi nopeasti laskettavana.



Kuva 3: Pelimoottoreiden ja sen myötä grafiikan kehitystä vuosien varrelta: Unreal Tournament (1999), Unreal Tournament 2004 (2004), and Unreal Tournament 3 (2007). Unreal Tournament on Epic Gamesin julkaisema (Gossipgamers 2009).

Ideana normal map esiteltiin ensimmäisiä kertoja Siggraphissa, vuosittain järjestettävässä tietokonegrafiikkakonferenssissa vuonna 1996. Venkat Krishnamurthy ja Marc Levoy julkaisi "Fitting Smooth Surfaces to Dense Polygon Meshes" pohti ajatusta, miten keveitä malleja voisi monimutkaistaa lisäämättä niihin oikeaa geometriaa. 1998 Jonathan Cohen ja Cigoni julkaisivat kaksi tutkimusta, "Appearance-Preserving Simplification" ja "A general method for recovering attributes on simplified meshes". Tutkimuksissa käsiteltiin sitä, kuinka yksityiskohtia voisi rikastuttaa siirtämällä high poly -mallin normaalit low poly -mallille. Erona metodeihin, joita työssäni käsittelen, mainittakoon se, että Cigoni ja Cohen pohtivat monimutkaisen mallin yksinkertaistamisesta samalla, kun yksityiskohdat yritetään säilyttää. Tekniikat, joita itse käytän, koostuvat sen sijaan vain mallien muokkaamisesta asteittain monimutkaisemmiksi low poly -mallin pysyessä muuttumattomana.

Kaupallisissa peleissä normaali karttoja käytettiin ensimmäisen kerran vuonna 2003, ja vapaiden lähdekoodien pelit niukempine resursseineen ja rahoituksineen seurasivat

perässä joitain vuosia myöhemmin. Kun normal mapeja alettiin hyödyntää peliteollisuudessa, suosituimmista ja edistyneimmistä pelikonsoleista ainoastaan Playstation 2 ei tukenut normal mapeja. Nykykonsolit, kuten Microsoftin Xbox 360 ja Playstation 3, sen sijaan tukeutuvat voimakkaasti normal mapien tuomiin hyötyihin.

5 Normal mapin luominen

Normal mapin voi luoda monella eri tavalla. Sen voi renderöidä mallinnetusta tai sculpatusta pinnasta. Sen voi maalata käsin tai muokkaamalla valokuvia. Normaali kartan voi tehdä myös käyttämällä kuvankäsittelyohjelmaan ladattavia lisäosia, jotka muuntavat värikuvan normaali kartaksi. Näiden lisäksi on olemassa ohjelmia, jotka ovat erikoistuneet yksinomaan kuvatiedostojen muuntamiseen eri tekstuureiksi.

Malli, johon aiotaan käyttää normaali karttaa yksityiskohtien lisäämiseksi, luodaan samoilla periaatteilla kuin mikä tahansa muu siisti ja matalaresoluutioinen 3D-malli. Mallintaminen on hyvä aloittaa kuutiosta tai nelikulmiosta. Polygonit kannattaa pitää kaikki neliskanttisina, koska tekstuurityökälu käsittelevät polygoneja automaattisesti nelikulmaisina. Jos seassa on erimuotoisia polygoneja, tekstuurit eivät välttämättä toistu halu-tunlaisesti. Malli on hyvä pitää muutenkin niin siistinä kuin mahdollista ja noudattaa periaatetta, että edgejen muodostama ristikko mallin pinnassa olisi mahdollisimman symmetrinen ja että ne muodostavat linjoja jotka kulkevat mahdollisimman suoraan.

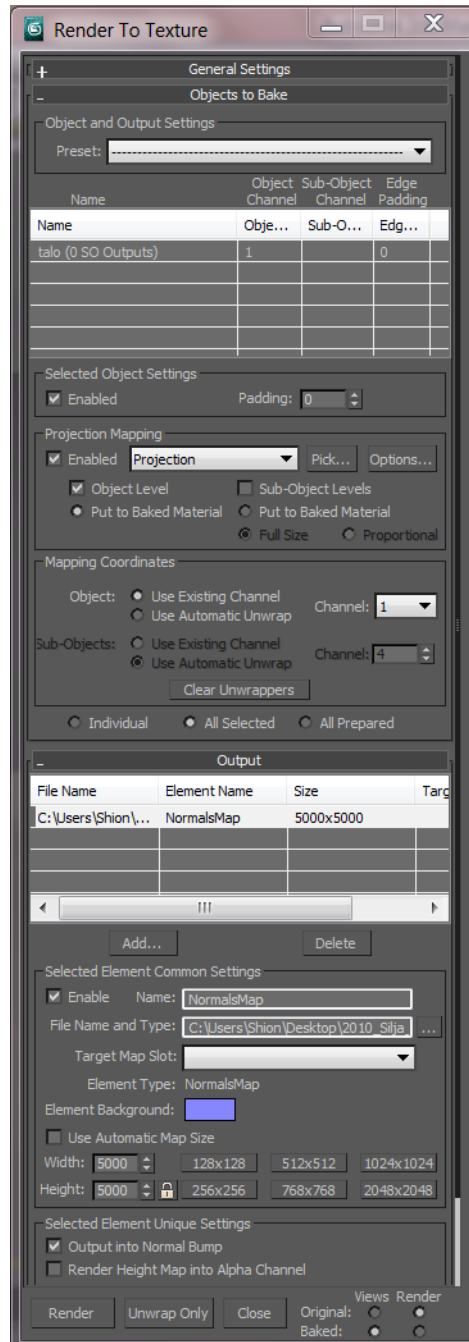
Sen lisäksi, että tekstuurit toimivat paremmin siistissä mallissa, on mallinnusta helpompi hallita, kun linjat pysyvät mahdollisimman suorina. Jos jossain kohdassa näkyy renderöidessä virhe, se on helppo paikantaa tasaisen verkon keskeltä, kun taas pinnassa, jossa polygoneja on päällekkäin ja edget menevät miten sattuu, yksittäisen virheen löytäminen ja korjaaminen voi olla lähes mahdotonta poistamatta ja luomatta osioita kokonaan uudelleen.

Jos normal map toteutetaan mallintamalla, workflow on pääpiirteissään seuraava: ensin mallinnetaan haluttu low poly -malli, johon normal map halutaan liittää. Low polyllle tehdään UVW-unwrap, jotta valmis tekstuuri saadaan järkevästi asetettua paikalleen. Malli kopioidaan ja kopio sijoitetaan alkuperäisen mallin kanssa kohdakkain

ja siihen lisätään mallintamalla geometriaa niin paljon kuin nähdään tarpeelliseksi. High poly -version valmistuttua sen pinnanmuodoista renderöidään tekstuuri eli normal map.

Toinen käytetty menetelmä on nimeltään retopo (re-topology), jossa työjärjestys on osin päinvastainen. Ensin mallinnetaan high poly -versio, jonka perusteella luodaan low poly -malli, joka olisi pinnanmuodoiltaan mahdollisimman lähellä alkuperäistä yksityiskohtaista geometriaa.

Tämän jälkeen kuva on valmis käytettäväksi lisäämään kevyeen malliin vaihtelevuutta ja realismia, jotka reagoivat oikean high polyn tavoin esimerkiksi valaistuksen muuttumiseen.



Kuva 4: Render to texture -ikkuna. Normal mapin voi renderöidä huoletta hieman suurempanakin tiedostona ilman, että mallin hidastuu merkittävästi.

Jos kuvaa ei toteuteta mallintamalla, low polyn mallintamisen ja unwrappaamisen jälkeen tekstuuri toteutetaan hyväksikäyttämällä eri värikanavia tallentamaan eri ulottuvuuksia. Työvaiheita käsitellään tarkemmin seuraavassa osiossa. Kuten muitakin tekstuureja, normal mapia voi käsitellä kuten mitä tahansa kuvatiedostoa. Jos virheitä tai muita epätoivottuja kohtia näkyy, niitä voi varovaisesti korjata kuvankäsittelyohjelmalla.

6 Esimerkkitoita

Käytän lopputyössäni pääasiallisena 3D-ohjelmana 3Ds Maxia ja kuvailen, miten sillä saa luotua normal mapin ja miten muulla ohjelmalla tehdyn normal mapin saa toimimaan Maxissa. Toiminto ei ole Maxin oma, eli sitä voi käyttää muissakin ohjelmissa, joskin työmenetelmät ovat luonnollisestikin erilaisia.

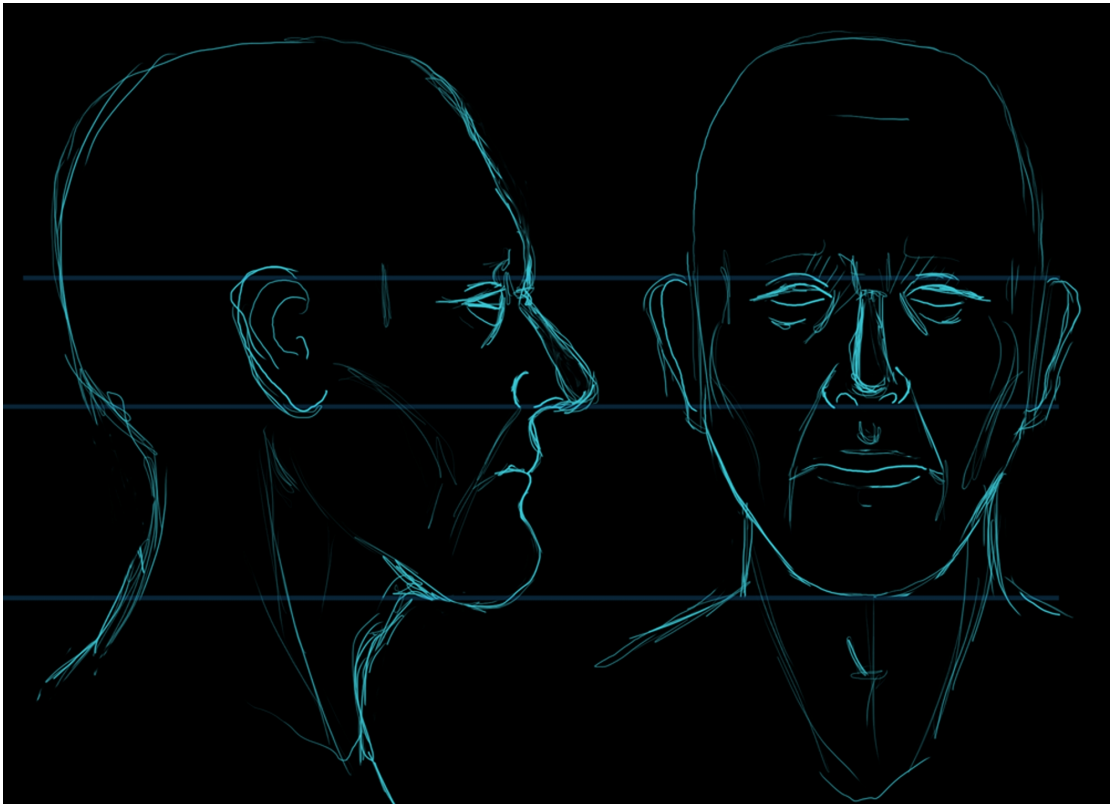
Päädyin valitsemaan esimerkkitoiksi kolme toisistaan mahdollisimman paljon poikkeavaa työtä. Halusin toteuttaa kolme eri tavalla mallinnettavaa ja normal mapattavaa työtä oppiakseni eri tapoja työskennellä erilaisten tekstuurien parissa. Valitsin orgaaniseksi malliksi vanhan miehen pään, koska erityisesti pelihahmojen aina vain yksityiskohtaisemmat kasvot ovat olleet hienona esimerkkinä siitä, kuinka grafiikat ovat kehittyneet huimasti vuosien varrella. Yksityiskohtia löytyy nykypeleistä loputtomiin, pienistä kauneusvirheistä lähtien. Vanha mies tuntui luontevalta valinnalta, koska ryppyjen ja muiden ihon epätäydellisyyksien toteuttaminen sculptaamalla tuo normal mapin kyvyt esiin mahdollisimman hyvin.

Toisena työnä päätin toteuttaa rakennuksen. Halusin mallintaa low poly -talon, johon lisäsin runsaasti yksityiskohtia, koska olen kiinnittänyt erityisesti huomiota pelikenttien taustoihin. Pelien latausajat eivät ole kasvaneet merkittävästi ja silti yksityiskohtaista kaupunkia tai viidakkoa on tutkittavana uutuuspeleissä niin pitkälle kuin päähenkilön silmä kantaa. Halusin selvittää, miten niin yksityiskohtaisia ja laajoja alueita voidaan toteuttaa nykytekniikalla. Käytännössä toteutin vain yhden osan kuvitteellisesta kaupungista, mutta tekniikka on sama kuin silloin, jos olisin tehnyt korttelillisen taloja. Talon kanssa työskennellessäni huomasin normal mapaamisessa ongelmia, kun toteuttavat yksityiskohdat ovat liian kulmikkaita. Esimerkiksi suoria ja kapeita linjoja on vaikea saada toistumaan toivotunlaisesti high poly -mallista renderöidessä.

Kolmas työ on malliltaan aiempia huomattavasti yksinkertaisempi ja myös tekstuurin teko poikkeaa edellisistä suuresti. Tällä kertaa en luonut mallia ollenkaan, koska tekniikka oli tässä tapauksessa pääosassa. Valokuvasin nuudeleita ja tein niistä normal map -tekstuurin.

6.1 Kasvot

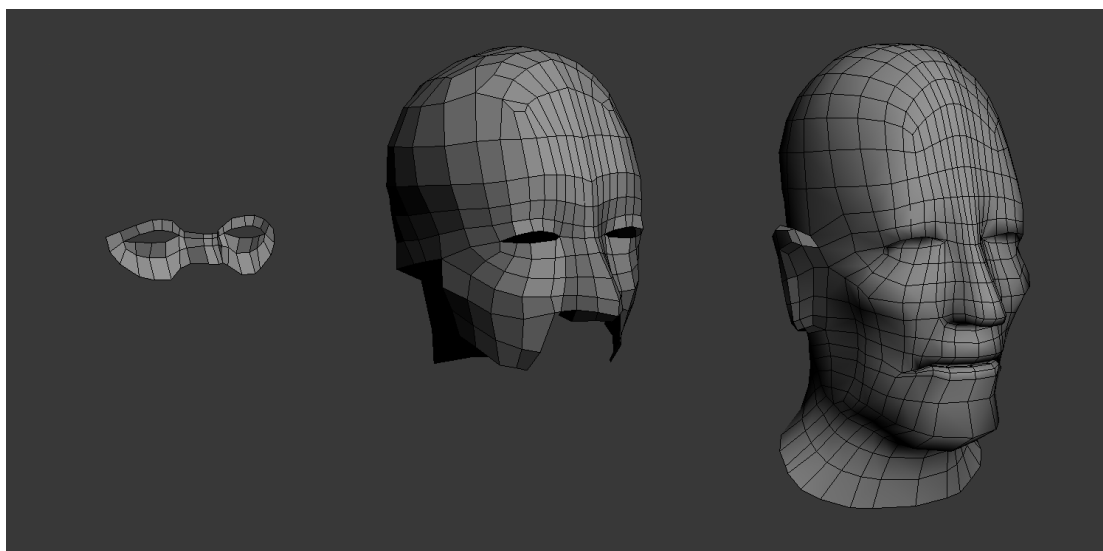
Aloitin ihmiskasvojen työstämisen piirtämällä kaksi mallikuvaa ja asettelemalla ne työtilaan kohtisuoraan toisiinsa nähden. (Kuva 5) Mallintamisen ensimmäinen vaihe oli luoda nelikulmio, josta muotoilin extrude-työkalulla silmänympäryksen. Kun olin suunnitellut jo valmiiksi pääosin sen, miten kasvojen verteksien tulisi sijoittua, geometrian paikoilleen asettelu ei ollut enää kovin monimutkaista. Vaikeinta tässä olisi ollut se, ettei suunnitelmaa olisi ollut ja mallintaessa olisi usein törmännyt viisikulmisiin tai kolmikulmisiin polygoneihin. Symmetryllä säästää paljon aikaa ja vaivaa, koska sitä käyttäessä tarvitsee luoda ainoastaan toinen puolisko kasvoista ja Max peilaa sen valittuun suuntaan halutun matkan päähän mallin toisesta puoliskosta.



Kuva 5: Mallinnuksessa apuna käytetty hahmotelma vanhan miehen kasvoista

Kun kasvot olivat muodoltaan valmiit (Kuva 6), kävin tarkkaan läpi, ettei malliin ollut karannut vääränmuotoisia polygoneja tai muita virheitä, kuten väärin päin olevia normaaleja. Jouduin asettelemaan mallin pivotin, eli keskikohdan uudelleen, koska se oli karannut symmetryn takia pois mallin keskeltä. Low poly -kasvoille tarvitsi tehdä vielä UVW-unwrap, joka tulisi varmistamaan, että Mudboxista importoitu normal map aset-

tuisi mallin päälle oikein. Tein sauman, joka kulki mallin takaraivosta pääläelle ja toisen sauman, joka kulki mallin otsan poikki. Pelt-työkalun avulla sain mallin kasvot levitettyä 2D-tekstuuritiedostoa varten. Collapsoituani symmetryn malli oli valmis exporttavaksi Mudboxiin yksityiskohtien lisäämistä varten. On huomattavaa, että mallin tulee olla Maxin työtilassa origossa, jotta Mudboxin symmetry toimisi kunnolla.



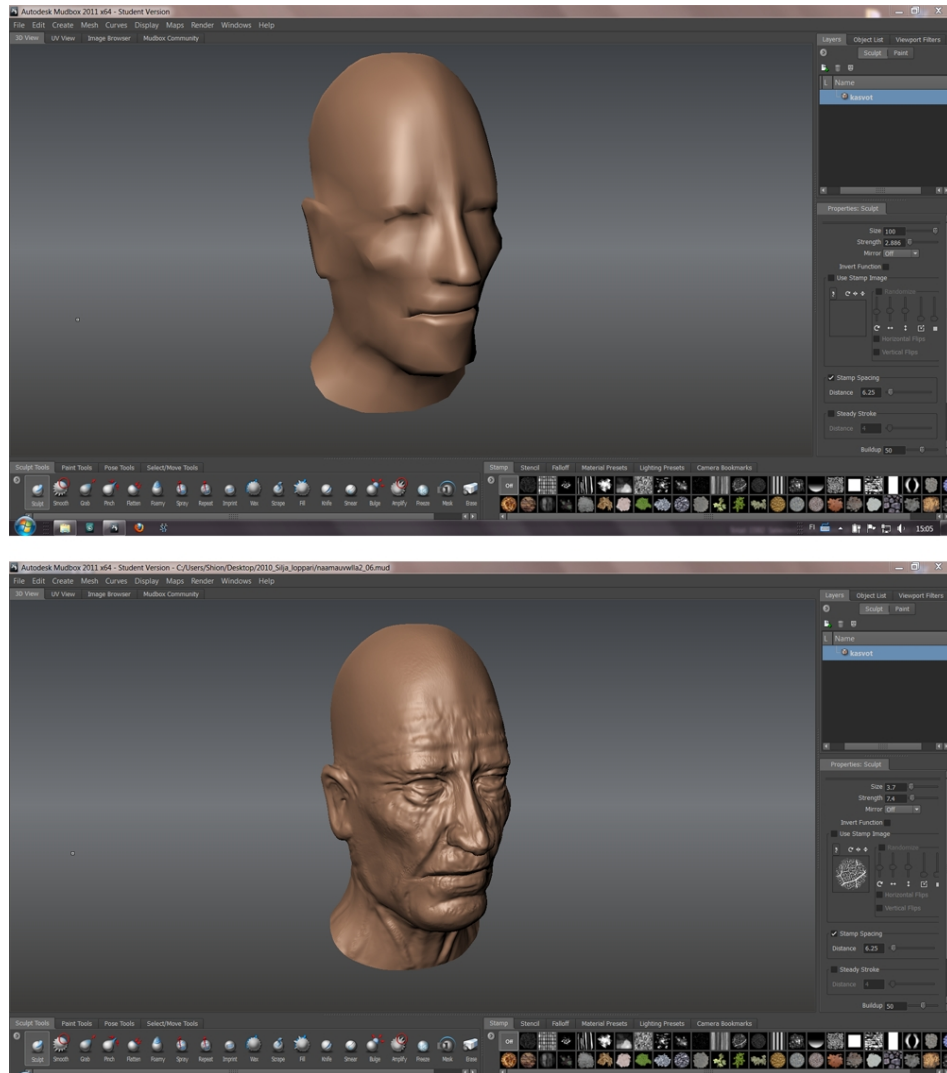
Kuva 6: Kasvojen mallintamisen eteneminen. Oikealla valmis low poly -malli.

Importoin Maxista exportoidun objektin veistettäväksi Mudboxiin. Sculptauksessa on tavalliseen mallintamiseen verrattuna se ero, että geometriaa tarvitaan automaattisesti huomattavasti enemmän jo lähtötilanteessa. Mudboxia käsketään luomaan tiuhempaa verkkoa ja se lisää polygoneja tietyllä kertoimella. Geometria tiuhentuu eksponentiaalisesti, eli jo parin subdivision levelin lisäämisen jälkeen polygonimäärä on voinut helposti ylittää sadantuhannen rajan. Sculptatessa mallin päälle "maalataan" ulkonevia osia tai leikataan työkaluilla, kuten puukolla, siihen syvennyksiä. Sculptaaminen on vähemmän teknistä kuin polygoniverkon pyörittely esimerkiksi Maxissa. Nimensä mukaan se on kuin kuvanveistoa, missä käsien ja työkalujen sijaan käytetään kursoria.



Kuva 7: Esimerkkikuvia kasvojen rypyistä sculptaamista varten. (Telegraph, Wordpress 2007)

Sculptaamiseen käytin Wacomin piirtopöytää, koska hiirellä geometrian maalailu olisi ollut toivottoman hidasta eikä rasitusvammojen saaminen olisi ollut kaukana. Etsin referenssikuvia vahoista ihmisistä, kun aloin toteuttaa rypyjä ja muita ihon epätasaisuuksia Mudboxilla (Kuva 7). Lähdin laajentamaan mallia ulospäin tekemättä siihen alkuperäisen geometrian ulointa kerrosta syvempiä kuoppia. Kuopat, joiden takia high poly -malli menisi alkuperäisen geometrian sisään, eivät toistu mapatessa tehtyinä kuoppina vaan mapiin tulee pahimmissa tapauksessa reikä, josta low poly -mallin pinta paistaa läpi. Eli sculptatessa laajensin vain ulospäin.



Kuva 8: Yllä malli importattuna Mudboxiin ja alla valmiina renderöitäväksi tekstuuriksi.

Kun sain kasvot haluamani malliseksi (Kuva 8), valitsin ylävalikosta Maps → Extract texture map → Normal Map. Tarkistin, että Target modelin kohdalla oli level 0, eli low poly -malli, jonka exporttasin Maxista, ja Source moden kohdalla level 4, eli malli johon olin lisännyt neljä subdivision leveliä geometriaa tarkentamaan.

Normal map -kasvoja ja high poly -kasvojen teknisiä tietoja vertaillen erot ovat huimia. Low poly -kasvoissa on 1592 kolmiota, kun taas valmiissa sculptatussa päässä niitä on 408 576. High poly -malli on hieman aidomman näköinen: varjot ovat pehmeämpiä ja valokohdat kirkkaampia. Mallin profiili on pyöreämpi, koska low poly -mallissa geometrian siluetista saattaa nähdä joitain polygoneja, vaikka mallissa onkin smoothing groups -asetus käytössä.



Kuva 9: Low poly, high poly ja low poly normal mapilla. Vaikka osa yksityiskohdista katoaa normaali karttaa käyttäessä, low poly -malli näyttää tekstuurin lisäyksen jälkeen merkittävästi yksityiskohtaisemmalta.

Liikkuvassa kuvassa näihin asioihin ehtii kiinnittää huomiota huomattavasti vähemmän aikaa kuin still-kuvissa. Jotkut mallin epätäydellisyydet katoavat automaattisesti hahmon liikkuesssa, koska joko katsojan silmä ei ehdi mukaan, tai liike sumentaa yksittäistä framea tarpeeksi piilottaakseen vian kokonaan. Yksi kuva normal map -kasvoista renderöityi kolmessa sekuntissa, kun taas high poly -päästä sai renderöidä täysin ilman tekstuureitakin 8 minuuttia 12 sekuntia. Jouduin lopulta renderöimään normal mapiksi rasteroidun high poly -mallin kuvat kevyempänä eli "vain" 103 736 kolmion tarkkuudella ja täysin ilman tekstuureita, koska sculptattu malli oli yksinkertaisesti liian raskas laskettavaksi tai UVW-unwrapattavaksi Maxissa. Tässä näkyikin, kuinka tärkeää työtilan kevyenä pitäminen on. Halusin kuitenkin saada samanlaisen esimerkkikuvan high poly -mallista, jonka tein low poly -mallille, vaikka en pystynyt käyttämään samaa diffuse mapia molemmissa. Geometrioiden erot näkee high poly -mallin eduksi, mutta normal mapin hyödyt voittavat sen heikkoudet (Kuva 9 ja 10).



Kuva 10: Valmis pää diffuse - ja normal mapilla. Renderöiminen kesti vain 4 sekuntia per kuva, kun taas high polyn laskeminen ilman diffuse mapia kesti useita minuutteja.

6.2 Rakennus

Lähdin mallintamaan kerrostaloa kuutiosta esimerkkikuvan perusteella (Kuva 11). En seurannut kuvaa orjallisesti enkä liittänyt sitä työtilaan, koska tässä tapauksessa pienet kokosuhteiden vaihtelut eivät mielestäni olleet oleellisia. Jos olisin aikonut tehdä täsmälleen samanlaisen rakennuksen kuin mallini, olisin ottanut siitä kuvia eri suunnista ja tehnyt niiden perusteella tarkemmat piirustukset. Tein ensin yhden kerroksen ikkunoineen ja parvekkeen ovineen. Tein ikkunankarmit ja syvennetyn ikkunalasin boolean-työkalulla ja liitin seinään. Koska leikkuri- ja yhdistämistoiminto harvoin tekee täysin toivottua tulosta, jouduin poistamaan syntyneitä ylimääräisiä verteksejä ja suoristamaan mutkalle menneitä edgejä. Sen jälkeen poistin valmista seinänpätkää ympäröivän seinän ja monistin valmiin kerroksen kokonaiseksi kerrostalon julkisivuksi. Näin minun ei tarvinnut tehdä kuin pari ikkunaa ja parvekkeen ovi alusta loppuun. Kopioitujen kerrosten yhteen liittäminen on huomattavasti yksinkertaisempaa ja nopeampaa kuin se, että olisin siistinyt boolean-työkalun jäljiltä useita ikkunoita ja ovia.



Kuva 11: Esimerkinä mallinnuksessa käytetty valokuva.

Unwrappasin mallin box mapping -asetuksella. Kopioin valmiin low poly -mallin (2684 polygonia), asetin kopion kohdakkain alkuperäisen kanssa align-työkalulla ja piilotin alkuperäisen saadakseni työskennellä pelkästään high poly -mallin kanssa. Toisin kuin kasvoja mallintaessa, taloon ei tullut orgaanisia muotoja, joten toteutin high poly -mallin täysin Maxin sisällä. Kokemukseni perusteella kulmikkaita ja mekaanisia muotoja on mielekkäämpää tehdä Maxilla kuin esimerkiksi sculptausohjelmilla. Koska unwrappasin talon tässä vaiheessa, minun piti jatkossa muistaa tekstuurin renderöityvän kanavalle 1. Jos olisin sen sijaan käyttänyt Render to texture -ikkunan automaattista unwrappausta, se olisi tallentanut tekstuurin tiedot kanaville 3 ja 4. Tässä vaiheessa tiedolla ei vielä ollut väliä, mutta lisättäessä normal map tekstuuriksi kanava tulee asettaa oikeaksi material editorissa, koska muutoin tekstuuri ei tule näkymään mallin pinnalla.

Halusin toistaa mallissani niin paljon esimerkkikuvastani löytyviä yksityiskohtia kuin mahdollista. Aloitin yksityiskohtien luomisen rakennuksen pintamateriaaleista. Tein ensimmäisen laudan kerrostalon katonrajaan, ja poistin sen ja seinän välissä olevan polygonin turhana. Asettelin laudan huolellisesti linjaan seinän kanssa, otin

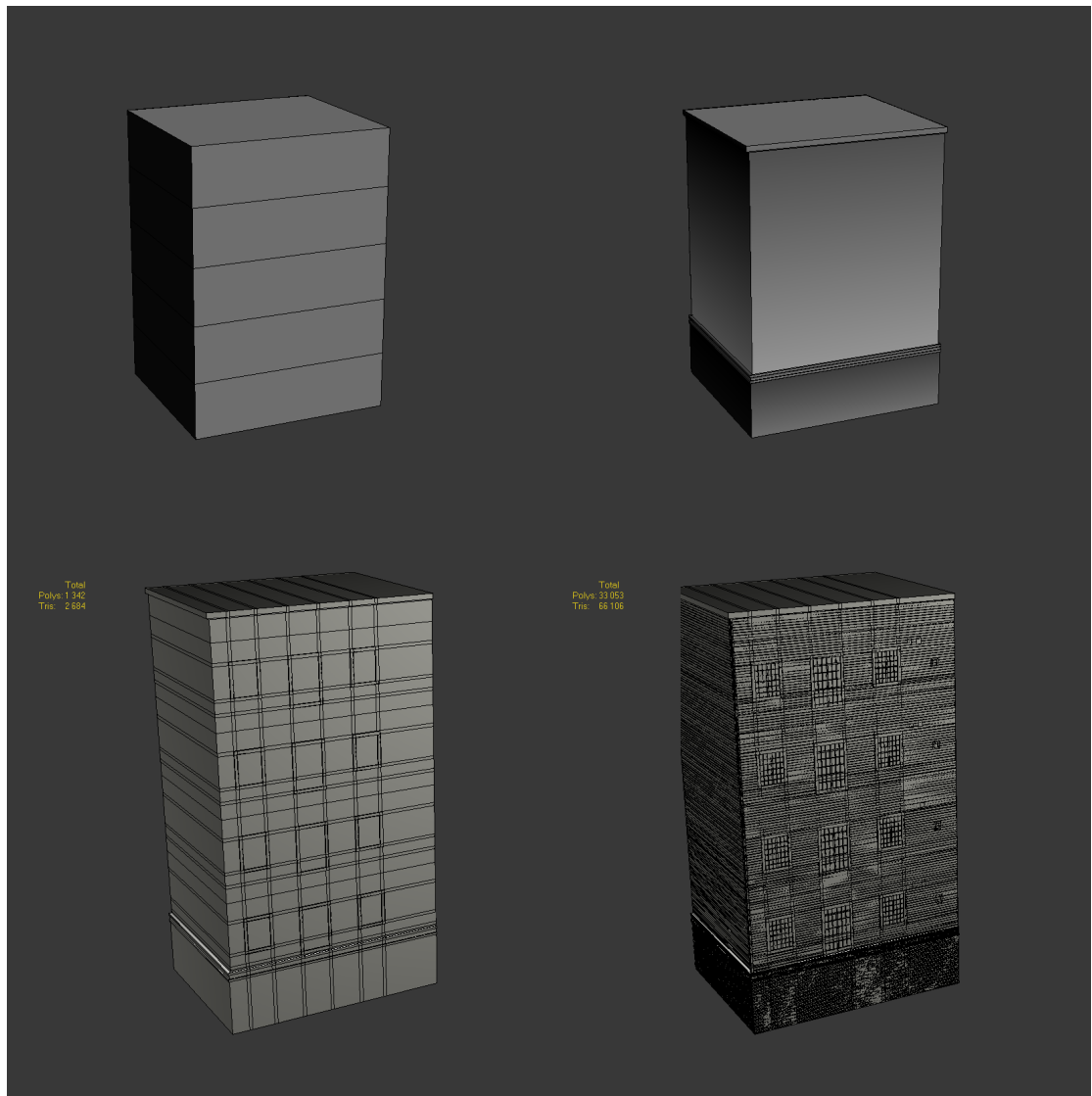
kopiointityökalun ja siirsin ensimmäisen kopion hieman ensimmäistä alemmas. Number of copies -kohdalla määrittelin lautojen määrän sopivaksi, jotta ne peittäisivät koko seinän. Tässä vaiheessa törmäsin yllättävään ongelmaan: jostain syystä kopiointityökalu ei tehnyt lautojen väleistä täsmälleen saman mittaisia, vaan joidenkin lautojen välissä oli pidemmät raot kuin toisten. Renderöidessä tämä näkyi kiusallisina raitoina seinässä: toiset raot olivat liian leveitä, kun taas toisen lautojen väliin ei jäänyt tilaa lainkaan. Ongelma korjaantui itsekseen, kun avasin työtilan toisella koneella ja toistin työvaiheen.

Kun kappaleiden määrä kasvaa tämän kokoiseksi tai suuremmaksi, kaikenlainen automatisointi on kultaakin kalliimpaa. Jos lautojen välit olisivat jääneet epätasaisiksi, käsin korjaaminen olisi ollut uskomattoman hidasta ja epätarkkaa puuhaa. Tässä vaiheessa myös on tärkeä tarkistaa, että ensimmäinen lautaobjekti oli varmasti nimetty joksikin muuksi kuin oletusnimiseksi, koska pian työtilassa olisi muuten tuhansia square-nimisiä objekteja, eikä oikeiden paikantaminen listasta ole enää mahdollista. Jos ensimmäinen kappale on nimetty, kopiot ovat siististi lauta 1, lauta 2, lauta 3 ja niin edelleen, eikä koko seinällisen paikantaminen valintalistasta ole vaikeaa.

Kun alkujaan lähdin tekemään talon tiiliverhottua helmaa, törmäsin samaan ongelmaan kuin laudoituksen kanssa. Tein ensin yhden pystyrivin tiiliä ja valitsin ne. Kun kopioin ne sivusuunnassa, seinän toiseen päähän mennessä tiilien välit olivat kadonneet olemattomiin. Tarkistin uudemman kerran, että unit setup -valikossa olin varmasti valinnut metrijärjestelmän ja senttimetrit molempiin kohtiin. Olin tehnyt näin, enkä tiedä, mistä moinen ongelma johtui. Onneksi tämä ongelma korjaantui samalla kun lautojen epätasaiset välit jäivät historiaan.

Koska halusin tiilistä hieman orgaanisemman muotoisia, eikä high poly -mallin kanssa tarvitse tässä tapauksessa huolehtia liian korkeista polygonimääristä, tein tiilistä hieman yksityiskohtaisempia kuin pelkästään sileitä kuutioita. Lisäsin geometriaa ensimmäiseen tiileen ja siirtelin hieman yksittäisiä verteksejä antaakseni sille hieman epätasaisuutta. Kopioidessa tätä samaa orgaanisemman muotoista objektiä seinästä näki aluksi hieman liian helposti varjoista ja valoisista kohdista, että sama tiili toistuu. Tämän ratkaisin valitsemalla ja yhdistämällä ne yhdeksi kappaleeksi attach-työkalulla. Sitten laitoin soft selectionin päälle, eli valitessani yhden verteksin Max valitsi tietyltä säteeltä muutkin verteksit ja toisti itse valitulle pisteelle tehdyt muutokset muihin ver-

tekseihin sitä heikompana mitä kauempana ne ovat tästä päävalinnasta. Valitsin summittaisesti joitakin verteksejä ja vedin niitä hieman seinästä ulos, ja toistin tämän pariin otteeseen siirtämällä aina noin kymmentä verteksiä hieman sisään tai ulos seinästä luodakseni sopivasti hajontaa liian symmetrisenä toistuvalla pinnalla.



Kuva 12: Talon mallintamisen eteneminen. Alla valmis low poly -malli sekä valmis high poly.

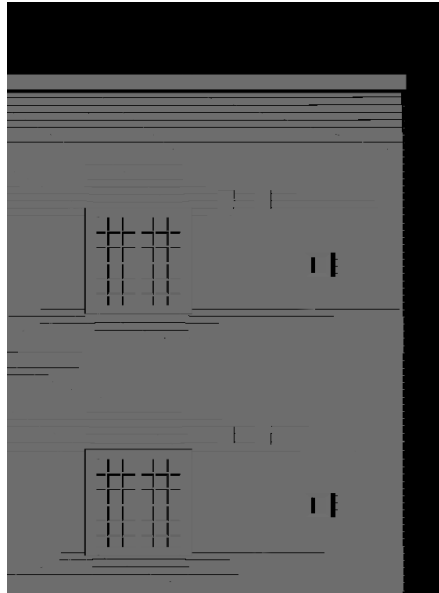
Pintamateriaalien jälkeen lähdin toteuttamaan muita yksityiskohtia, kuten tuuletusristikoiden ja parvekkeiden muttereita. Loin karkeasti mallin pintaan uusia kappaleita ja muotoilin ne haluamani näköisiksi. Pitääkseni irtonaisten esineiden määrän mahdollisimman vähäisenä liitin samankaltaiset osat toisiinsa attach-työkalulla.

Tämän jälkeen low poly -talolle lisättiin Projection modifier, ja projectionin kohteeksi

valittiin high poly -versio talosta. Ensimmäisellä kerralla tein mallia ympäröivästä cagesta liian suuren, ja tekstuuriini tuli paljon punaisia kohtia merkitsemään alueita, joissa normaalit eivät jostain syystä tallentuneet kunnolla. Punaisia jälkiä saa aikaan esimerkiksi jättämällä mallia ympäröivän cagen niin pieneksi, etteivät kaikkein eniten ulkonevat osiot mahdu sen sisään. Kun poistin modifierin ja tein maltillisemman häkin, kuva renderöityi ilman virheitä. Cagea saa kasvatettua kohdasta "Push".

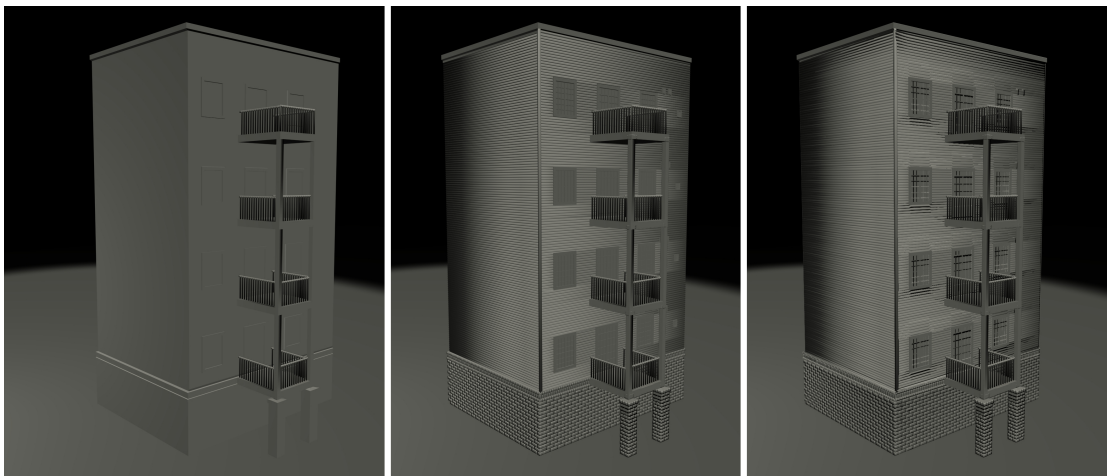
Tekstuuria renderöidessä valaisin mallin tasaisesti ja laitoin sille harmaan oletusmateriaalin, koska aikaisemmin olen törmännyt siihen, että osa tekstuurista toistuu normal map -tiedostossa pilaten sen. Valot sen sijaan muuttavat sitä, miten normaalikartta reagoi valoihin scenessä. Olin renderöinyt kuvan, jossa oli toispuolinen valo, ja renderöityäni normal mapin huomasi, että tätä varjopuolta ei saanut valaistuksi enää jälkeenpäin millään, vaan varjoalue oli renderöitynyt mustana.

Ylävalikosta avasin Render to Texture -ikkunan, josta klikkasin aktiiviseksi Enable Projectionin, ja Outputiksi valitsin NormalsMapin. Ensimmäinen testilaskeminen osoitti, että tekstuurin koko on aivan liian pieni näyttämään kaikkia pieniä yksityiskohtia talon seinässä. 1000 x 1000 pikseliä ei riittänyt, koska lautojen välit ovat niin kapeita, että näillä mitoilla ne katosivat tekstuurista: ne olivat tässä kuvasuhteessa alle pikselin levyisiä (Kuva 13). Kokeilin heti huomattavasti isompaa, 5000 x 5000 pikselin -tekstuuria, joka oli tarkkuudeltaan huomattavasti parempi, mutta teki scenestä heti raskaamman renderöidä.



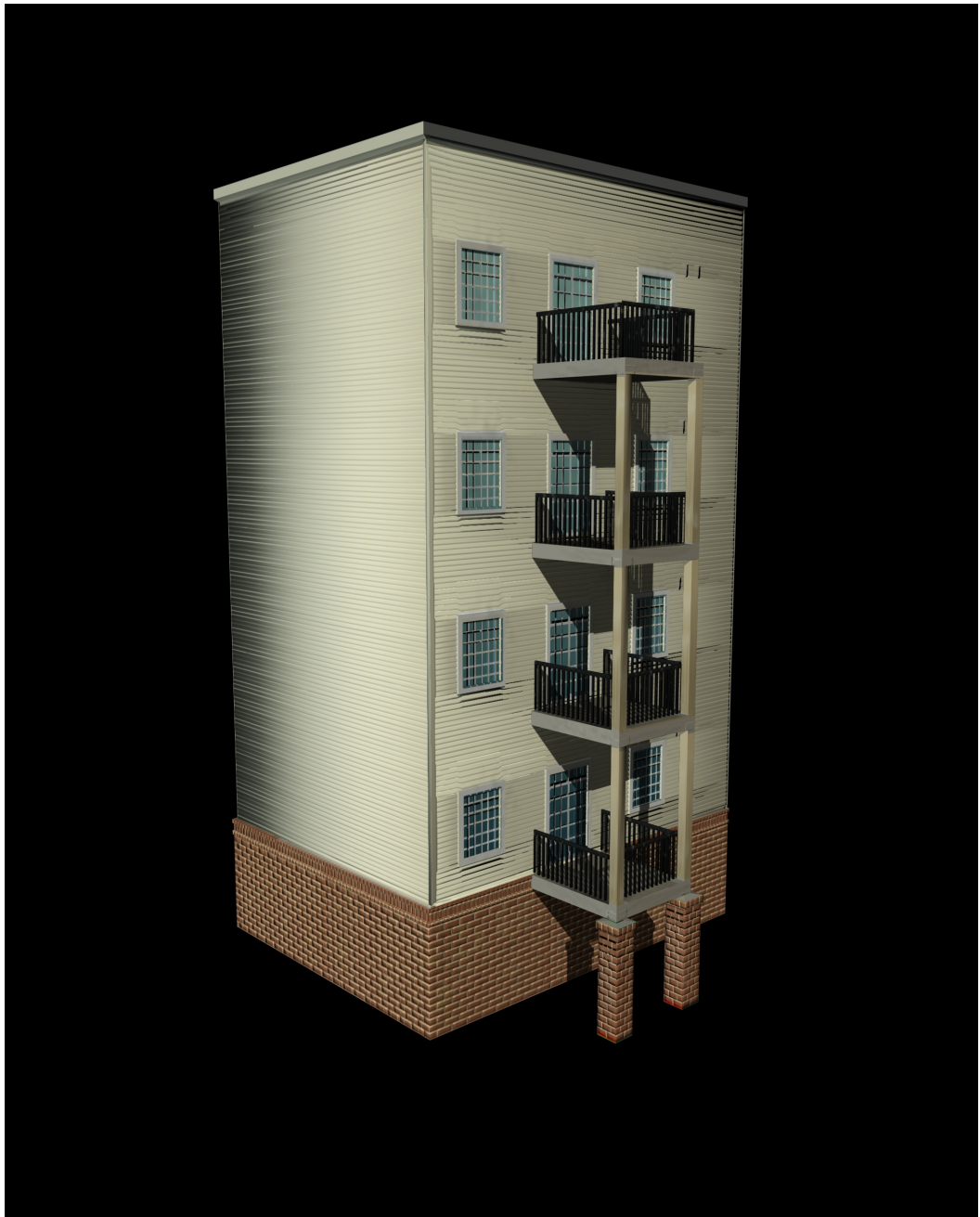
Kuva 13: Normal map renderöityi liian pienenä (1000 x 1000 pxl.) ja osa yksityiskohdista ei tallentunut ollenkaan, kuten esimerkiksi lautojen raot, jotka jäivät alle pikselin levyisiksi. Tekstuurin asetin mallille avaamalla Material Editorin ja luomalla uuden materiaalin.

Kohtaan Bump Map valitsin Normal Bump ja siihen ulos renderöidyn targa-kuvan. Tämän lisäksi Bump Map -kohtaan voi valita yhden toisen kuvatiedoston lisäämään pinnalle epätasaisuutta bump mapin muodossa. High poly -mallin piilottamisen jälkeen tekstuuri lisätään low poly -mallille. Kun bump map luodaan, sen voimakkuus on automaattisesti vain 30 prosenttia, joten kannattaa muistaa lisätä sen intensiivisyyttä Material Editorista, jos yksityiskohtien tuoma efekti on liian heikko.



Kuva 14: Low poly, high poly sekä low poly -malli normal mapilla. Geometriset muodot eivät toistu normal mapilla yhtä uskottavasti kuin pehmeät muodot, koska silmän on helpompi erottaa suorista viivoista ja symmetrisistä muodoista virheitä, mutta esimerkiksi pehmeälinjaisemmat tiilet toistuivat lähes virheettömästi.

Huomasin ikäviä ongelmia Render to Texture -toiminnon kanssa. Koska normal mapin renderöinti on välillä hyvinkin hidasta, olisi tärkeää, että renderöimisen saisi keskeytettyä, jos huomaa puoleessa välissä, että tekstuurissa on jotain vikaa. Render to texture -ikkunassa ei Max 10:ssä kuitenkaan ole Cancel-nappia, joten jos renderöimisen aloittaa, sitä ei saa keskeytettyä millään. Huomasin myös, kuinka Maxin Render Setup -valikon jotkut asetukset vaikuttavat tekstuureiden renderöimiseen. Olin esimerkiksi viimeksi renderöinyt pätkän animaatiota ja kun siirryin siitä suoraan Render to Texture -valikkoon, päädyin laskemaan samaa tekstuuria sen saman 40 framea, mitä animaatio oli kestänyt. Tekstuurin renderöimisen ei tulisi reagoida animaation laskemiseen, koska täsmälleen samaa kuvaa ei ole missään tilanteessa tarpeellista renderöidä uudelleen ja täten kuvasarjan luominen tekstuurista ei ole järkevää. Varsinkin peruutustoiminnon puuttuessa on ikävää huomata, kun pitkän renderöimisen kuuluisi olla ohi, että kone alkaakin laskea jo seuraavaa kuvaa.



Kuva 15: Normal map toistaa laudoituksen suorat linjat osin hieman liian jyrkinä, mutta esimerkiksi tiiliseinän elävämpi pinta toistuu kauniin pehmeänä.

6.3 Nuudelitekstuuri

Viimeinen menetelmä normal mapin luomiseksi eroaa huomattavasti kahdesta aikaisemmasta. Tällä kertaa luon kolmiulotteisuutta planen pintaan valokuvaamalla tekstuurin ja muokkaamalla valokuvat normal mapiksi Ryan Clarkin verkkotutorialin

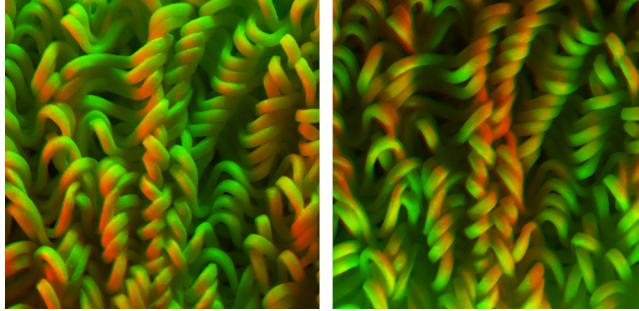
mukaan (zarria.net 2004). Tällä keinolla on kätevää luoda erilaisia pintamateriaaleja tai vaik-kapa siirtää olemassa olevia yksinkertaisia kappaleita ja pinnanmuotoja 3D-ohjelmaan ilman kalliita 3D-skannereita. Kuvasin opiskelijan lempiruuan, eli nuudelit, ja tein siitä tekstuurin. Keskityn viimeisessä osiossa vain itse tekstuurin tekemiseen, enkä mallin-tamiseen tai muihin työvaiheisiin, koska jos valitsee tämän työtavan, kaikki oleellinen tapahtuu valokuvatessa ja kuvankäsittelyohjelman parissa.

Ensimmäinen vaihe oli järjestää kuvauspaikka. Asetin kameran tripodiin nuudeleiden yläpuolelle, koska tarvitsin neljä kuvaa täsmälleen samasta kuvakulmasta. Himmensin taustavalot ja järjestin paikalle yhden kirkkaan ja siirrettävän valon. Nuudeleista piti saada neljä erilaista kuvaa samasta kuvakulmasta. Ainoa asia, mikä liikkui, oli valo. Kohde valaistiin vuorotellen vasemmalta, oikealta, alapuolelta ja yläpuolelta. Valmiit kuvat siirsin Photoshopiin muokattaviksi.



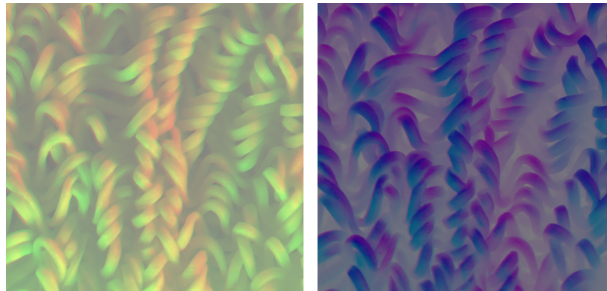
Kuva 16: Nuudeleista 4 kuvaa samasta kuvakulmasta, mutta valaistuna eri suunnista.

Rajasin kuvat mieleisekseni ja muutin ne harmaasävykuviksi. Loin kaksi uutta kuvatiedostoa. Toiseen liitin mustavalkoisen kuvan ylhäältä valaistusta nuudeleista vihreään värikanavaan ja vasemmalta valaistun kuvan punaiseen värikanavaan. Toiseen kuvaan liitin alhaalta valaistun kuvan vihreään kanavaan ja oikealta valaistun punaiseen kanavaan. Tässä vaiheessa on huomattava, että sinisten kanavien tulee olla mustia, koska muuten ne muuttavat lopullisen kuvan värimaailmaa niin, ettei normal map ole oikean värinen 3D-ohjelman luettavaksi.



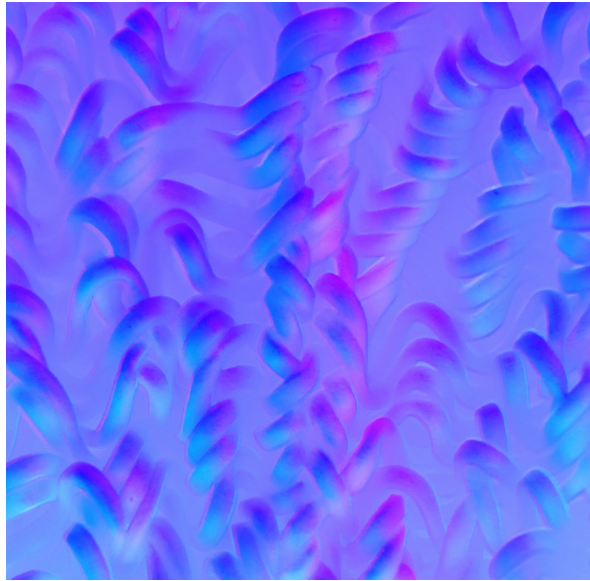
Kuva 17: Ylhäältä ja vasemmalta (ylävasen) valaistut kuvat vihreään ja punaiseen kanavaan liitettyinä sekä edestä ja oikealta (alaoikea) valaistut kuvat samoilla kanavilla sinisen kanavan ollessa musta.

Alaoikeaksi nimetyn kuvan kohdalla valitsin ylävalikosta Image → Adjustments → Levels. Output levels -kohtaan kirjoitin arvot 127 ja 0. Ylävasemmaksi nimetty kuva taas sai samoiksi arvoikseen 128 ja 255. Lopputulos näytti seuraavalta (Kuva 18).

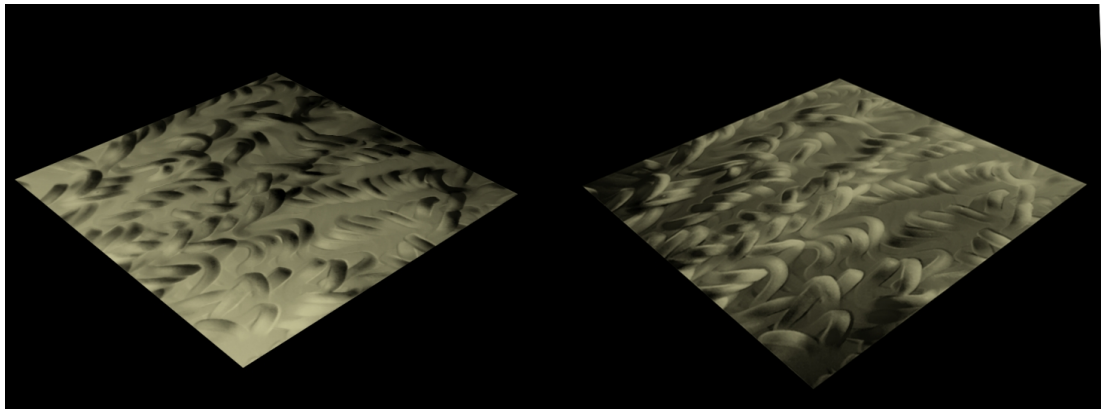


Kuva 18: Yhdistetyt punavihreät kuvat levels-asetusten jälkeen.

Kopioin alaoikean kuvan ylävasemman kuvan päälle, ja Blending modeksi valitsin Overlay. Layereiden yhdistämisen jälkeen normal map oli valmis käytettäväksi. (Kuva 19) Maxissa valitsin Material editorissa Bump map → Normals bump → Bitmap ja tähän tiedostosta juuri luodun normal mapin. Diffuse mapiksi käytin vain nuudeleille ominaista kellertävää väriä. Vaikka objekti, jolle materiaalin laitoin on sileä, kuvista näkee, kuinka nuudelit reagoivat valoon lähes oikean geometrian tavoin (Kuva 20).



Kuva 19: Overlaylla yhdistetyt kuvat muodostavat valmiin normal mapin.



Kuva 20: Nuudelitekstuuri planella. Valoa liikutellessa normal map muodostaa varjoja ja kirkkaita kohtia aivan kuin planen pinnassa olisi oikeaa geometriaa.

Tämä metodi ei ole kovin käytännöllinen kokonaisen normaalikartan luomisessa, mutta avaa valmistusprosessillaan sitä, millä tavalla tietokone tällaisen kolmiulotteisen tekstuurin valmistaa. Eri ulottuvuuksia koskeva tieto tallennetaan eri värikanaville ja niiden muodostama lopullinen rgb-kuva on 3D-ohjelman luettavassa muodossa valmiina muunnettavaksi huijausnormaaleiksi.

7 Yhteenveto

3D-mallien monimutkaistuessa ja tullessa raskaammiksi koneet ja tekijät alkavat kärsiä taakan alla. Erilaisilla tekstuureilla saadaan vähennettyä käsiteltävän geometrian määrää menettämättä merkittävästi yksityiskohtia. Kun värikartat ja bump mapit eivät enää tarkkuudeltaan riitä huijaamaan vaativien kuluttajien yksityiskohtien nälkää, mallien tarkkuutta voidaan nostaa esimerkiksi normal mapeilla. Normal mapin luominen ja renderöiminen vie aikaa, mutta reaaliaikaisessa mediassa ja animaatioissa työtilan renderöimissajan keventäminen edes kymmenellä sekunnilla toistuu ja kasvattaa eksponentiaalisesti loppujen lopuksi säästetyn ajan määrää. Peleissä scenen keventäminen tarkoittaa käytännössä sitä, että peli toimii tehokkaasti myös hieman tehottomammillakin alustoilla, mikä taas mahdollistaa pelin myymisen suuremmalle kuluttajamäärälle. Säästetyn ajan voi myös sijoittaa muihin asioihin, esimerkiksi tekemällä työtilasta yhtä raskaan, mutta vielä yksityiskohtaisemman tai muuten monimutkaisemman.

Halusin tämän aiheen valitessani oppia mahdollisimman paljon low poly -mallintamisesta ja tekstuureista, erityisesti normal mapeista. Näistä tekstuureista oli kuitenkin yllättävän vaikeaa löytää tietoa virallisia reittejä. Internet on pullollaan käyttäjien tekemiä tutoriaaleja, mutta esimerkiksi Maxin helpistä ei löytynyt muun muassa "normal map" tai "normalsmap" hakusanoilla mitään, mistä olisi ollut normal mapeihin tutustuessa mitään hyötyä. Opiskelu ja lopputyön tekeminen meni hieman enemmän kepillä jäätä kokeilemiseen kuin aiheesta valmiin tekstin lukemiseen kuin olin alun perin luullut. Puolessa välissä seminaarityön työstöä tilanne oli hyvin pitkälti se, etten löytänyt joko mitään mainintaa normal mapeista tai sitten löysin aiheesta vaikeaselkoisia tieteellisiä julkaisuja. Käytännössä tämä tarkoitti sitä, että joko löytämäni tiedot olivat itses-täänselvyyksiä tai niin monimutkaista tutkimustekstiä, ettei tällä osaamisen tasolla ole mahdollista ymmärtää tekstistä kuin hyvällä tuurilla puolet. Oli vaikeaa poimia itselleni liian matemaattisen tekstin seasta tiedonmurusia, joita pystyin käyttämään työssäni luottaen siihen, että olin ymmärtänyt materiaalin kaikin puolin oikein.

Aihe, johon keskityin oli mallintamisen ja 3D:n parissa työskentelyn kannalta suppea, mutta avaa silti mielestäni hyvällä tavalla hyvän 3D-grafiikan luomisen monimutkaisuutta reaaliaikaisille alustoille ja muutenkin. Evoluutio, jonka yksi tekstuurityökalu on käynyt läpi kuvastaa sitä valtavaa työmäärää, mitä yksittäisten efektien toistaminen

esimerkiksi videopeleissä on pitänyt sisällään. Tutustumalla syvällisemmin yhteen materiaalityyppiin, eli normaalikarttoihin, olen saanut monipuolisemman käsityksen siitä, mitä reaaliaikaisen 3D:n parissa työskentely ja sen eteenpäin kehittäminen karkeasti pitää sisällään.

Minulla oli aloittaessani kolme erilaista päätavoitetta työlleni. Tärkein niistä oli tutustua normaali karttoihin ja muuten teksturoimiseen syvällisemmin kuin mitä lähtötasoni oli. Saavutin tavoitteeni ja olen tyytyväinen rutiiniin, jonka esimerkkitoita tehdessä tekstuurien luomiseen sain.

Halusin myös toteuttaa mahdollisimman selkeän ohjeen sille, miten normaalikarttoja voi luoda ja käyttää. Halusin, että opiskelijat, jotka eivät vielä osaa niitä tehdä, voisivat käyttää työtani apuvälineenä uusien tekstuurien käyttämiseen. Ongelmaksi muodostui saamani palaute, jonka mukaan työ pitäisi suunnata pois tutorialmaisuudesta. Noudatettuani saamiani ohjeita, minusta tuntuu, ettei tekstini perusteella pysty oppimaan tekstuurien valmistamista käytännössä. Tavoitteeni jäi siis tältä osin saavuttamatta.

Kolmas tavoitteeni oli saada koottua tietoa reaaliaikaisessa mediassa käytetyistä tekstuureista yhdeksi selkokieliseksi kokonaisuudeksi sen takia, koska itse olen kokenut aiheesta tiedon löytämisen hankalaksi. Pyrin siihen, että työni perusteella lukijalle muodostuisi yleinen kuva siitä, millä tavalla reaaliaikaisiin sovelluksiin tuotetaan sisältöä, joka on laskennan kannalta tarpeeksi kevyttä ja samalla täyttää nykyaikaiset ulkonäkövaatimukset.

Asioita, joita tekisin eri tavalla, on muunmuassa työni rajaaminen. Jos olisin alusta asti ollut tekemässä työtä selvästi teorian pohjalta, enkä käytännönläheisesti, tutorialmaisuus ongelmaa kanssa ei olisi varmaankaan ollut.

Lähteet

Clark, Ryan 2004. Normal Map Photography. [verkkotutorial] Zarría.net. Mipmap.
<<http://zarría.net/nrmphoto/nrmphoto.html>> (luettu 3.9.2010)

Hastings-Threw, James 2007. Creating Normal Maps with Cinema 4d. [verkkodokumentti].
Planetpixmapemporium.com. <<http://planetpixmapemporium.com/tutorialpages/normal.html>>
(luettu 5.9.2010)

Illikainen, Kimmo 2002. 3D Studio Max 4.0. Jyväskylä: Docendo Finland Oy.

Krishnamurthy & Levoy 1996. Fitting Smooth Surfaces to Dense Polygon Meshes.
[verkkodokumentti] Siggraph 1996 <<http://www-graphics.stanford.edu/papers/surfacefitting/>>
(luettu 4.9.2010)

Lehtovirta, Pekka, Nuutinen, Kari 2000. 3D. Jyväskylä: Docendo Finland Oy.

Puhakka, Antti 2008. 3D-grafiikka. Helsinki: Talentum Media.

Wikipedia 2010a. Normal mapping. [verkkodokumentti]. Wikipedia.org.
<http://en.wikipedia.org/wiki/Normal_mapping> (luettu 5.9.2010)

Wikipedia 2010b. Mipmap. [verkkodokumentti]. Wikipedia.org.
<<http://en.wikipedia.org/wiki/Mipmap>> (luettu 25.9.2010)

Autodesk Max Help Center 2010. [Autodesk 3Ds Max 2010 avustaja]

Kuvat:

Kuva 2: <<http://en.wikipedia.org/wiki/File:3D-monster-maze-T-rex-2-steps-away.png>> (lainattu 26.9.2010)

Kuva 3: <http://www.gossipgamers.com/wp-content/uploads/2009/04/unreal_engine_comparison.jpg> (lainattu 26.9.2010)

Kuva 7: <<http://brianmpei.files.wordpress.com/2007/06/old-man.jpg>> (lainattu 26.9.2010)

<http://i.telegraph.co.uk/telegraph/multimedia/archive/00999/old-man_999439i.jpg>
(lainattu 26.9.2010)

<http://i.telegraph.co.uk/telegraph/multimedia/archive/00999/old-man_999439i.jpg>
(lainattu 26.9.2010)