



Kaupunkietiketti

Animaatioprojektin valaisu ja renderöinti Blenderiä käyttäen

Viestinnän koulutusohjelma
3D-animaatio ja visualisointi
Opinnäytetyö
25.5.2009

Peter Schulman

Koulutusohjelma Viestinnän koulutusohjelma		Suuntautumisvaihtoehto 3D-animaatio ja -visualisointi	
Tekijä Schulman Peter			
Työn nimi Kaupunkietiketti – animaatioprojektin valaisu ja renderöinti Blenderiä käyttäen			
Työn ohjaaja/ohjaajat Simolin Kristian			
Työn laji Opinnäytetyö		Aika 25.5.2009	Numeroidut sivut + liitteiden sivut 27+5
TIIVISTELMÄ <p>Opinnäytetyö on toiminnallinen opinnäytetyö. Työn tekijä toteutti osana kolmen hengen tiimiä, yhteistyössä Nelonen Median kanssa, sarjan kanavatunnuksia televisiokanava Neloselle. Toteutustapa oli 3D-hahmoanimaatio, ja tunnusten yhteinen teema oli hyvä käytös kaupunkiympäristössä. Tekijän pääasialliset vastuualueet olivat ulkoasun suunnittelu ja toteutus materiaalien, valojen, renderöinnin ja jälkikäsittelyn kautta, 3D-ohjelma Blenderiä käyttäen, ja kirjallisessa osiossa näitä asioita selvitetään perusteellisemmin.</p> <p>Blender on vapaan ohjelmiston periaatteisiin perustuva 3D-grafiikan ja animaation tekoon käytetty ohjelma. Siitä huolimatta, että se on kenen tahansa ladattavissa internetistä, ja vapasti käytettävissä sekä harrastuspohjaisiin että kaupallisiin projekteihin, se on teknisesti kehittynyt, ja monessa suhteessa samoissa tasoissa vastaavien kaupallisten ohjelmien kanssa. Varsinkin laajempien animaatioprojektien toteuttamiseen Blender tarjoaa runsaasti työntekoa helpottavia ominaisuuksia. Blenderissa on esimerkiksi mahdollista linkittää dataa tiedostosta toiseen, jolloin projektin eri osia on mahdollista pitää omissa tiedostoissaan, eikä muutoksia tarvitse tehdä kuin yhdessä paikassa.</p> <p>Materiaaleja ja valaistusta suunnitellessa visualisointityöhön ja muihin fotorealismia lähenteleviin tarkoituksiin, Blenderin sisäisen renderöintimoottorin rajoitukset tulevat helposti vastaan. Sen sijaan animaation, ja etenkin tyyllitellyn sellaisen, valaistuksen työstämiseen ja renderöintiin Blenderistä löytyy kattava määrä ominaisuuksia.</p> <p>Erialaisten valo- ja varjotyyppien ja monipuolisen materiaalijärjestelmän lisäksi erikoismaininnan ansaitsevat sisäänrakennettu noodikomposiittori, jonka avulla kuvan alustavaa värikorjailua on mahdollista suorittaa valaisun kanssa samanaikaisesti, sekä approximate ambient occlusion, jonka avulla voi tuottaa luonnollisen näköisiä varjoja nopeasti ja ilman kohinaa.</p> <p>Kirjallisessa osiossa selostetaan työn aikana eteen tulevia haasteita, sekä niihin löytyneitä ratkaisuja. Pääasiallinen näkökulma on animaation valaiseminen Blenderiä käyttäen, mutta siinä käsitellään myös asioita jotka ovat yleisluontoisia ja pätevät käytetystä ohjelmasta riippumatta.</p>			
Teos/Esitys/Produktio Kanavatunnuksia, 2 x 20 x 5 sekuntia hahmoanimaatiota			
Säilytyspaikka Metropolia Ammattikorkeakoulu kirjasto, Tikkurila			
Avainsanat Blender, animaatio, valaisu, renderöinti, ambient occlusion, linear workflow			

Degree Programme in Media		Specialisation 3D animation and visualisation
Author Schulman Peter		
Title Urban Etiquette – The lighting and rendering of an animation project using Blender		
Tutor(s) Simolin Kristian		
Type of Work Bachelor's Thesis	Date 25.5.2009	Number of pages + appendices 27+5
<p>ABSTRACT</p> <p>This thesis describes the work the author undertook shading, lighting, rendering and post-producing a series of channel idents for Nelonen Media / Channel Four Finland. The unifying theme for the idents was Urban Etiquette, or considerate behaviour in a city environment. The author worked as part of a three-person team, producing in total approximately three minutes of character animation, divided into 40 (20 pairs) of five seconds each.</p> <p>The main tool used for the project was the 3D modelling and animation software Blender. Blender is built around the principles of free software, and can be downloaded, used for both non-commercial and commercial projects, as well as modified to suit the user's need, at no charge. Despite this, Blender is technically capable, and in many areas on par with big-name commercial 3D graphics software.</p> <p>While Blender's internal rendering system lacks the physically correct shading and lighting models necessary for realistic visualisation work, character animation is commonly more dependant on flexibility and tools that allow for visually pleasing results with fast render times. In addition to a diverse selection of lights, shadows and material shading options, Blender provides several unique features, such as approximated ambient occlusion, which allows for quick and noise-free simulation of the effects of ambient light in the environment, and the built-in node compositor, which lets the lighting artist perform preliminary colour correction and contrast adjustment while setting up the scene lighting.</p> <p>This thesis aims to function as an introduction to shading, lighting and rendering animation using Blender, by describing the challenges faced by the author, and the solutions found. Additionally, it touches on many issues pertaining to these that are not specific to Blender, and may thus be of use to artists working in other software as well.</p>		
Work / Performance / Project Channel idents, 2 x 20 x 5 seconds character animation		
Place of Storage Metropolia University of Applied Sciences library, Tikkurila		
Keywords Blender, animation, lighting, rendering, ambient occlusion, linear workflow		

SISÄLLYS

1 JOHDANTO.....	2
2 DATABLOKIT JA LINKITYKSET.....	5
3 MATERIAALIT JA VALAISU.....	7
3.1 HAHMOJEN MATERIAALIT.....	7
3.2 MUUT MATERIAALIT.....	8
3.3 ANIMAATIOIDEN VALAISTUS.....	10
3.4 AMBIENT OCCLUSION -VARJOSTUS.....	13
4 KOMPOSITOINTI.....	17
4.1 LIIKE-EPÄTERÄVYYS JA VECTOR BLUR -NOODI.....	18
4.2 GAMMA-KORJATTU TYÖSKENTELY.....	22
5 POHDINTAA LOPUKSI.....	24
LÄHTEET.....	27
LIITE A	
LIITE B	

1 JOHDANTO

Kaupunkietiketti on nimitys ja teema animaatioprojektille, jonka allekirjoittanut suunnitteli ja toteutti yhdessä kahden kanssaopiskelijan, Jere Virran ja Jussi Saarelman, kanssa, syksyn 2008 ja kevään 2009 aikana. Kyseessä on yhteistyössä Nelonen Median kanssa luotu sarja mainoskatkojen yhteydessä näytettäviä kanavatunnuksia televisiokanava Neloselle.

Idea, jonka ympärille rakensimme animaatiomme oli hyvä käytös kaupunkiympäristössä. Ajatuksemme oli samalla viihdyttää katsojia, mutta myös saada heidät ajattelemaan omaa käytöstään.

Kanavatunnusformaattiin tämä sopisi sikäli, että mainoskatkolle mentäessä voitaisiin näyttää väärää käytöstä, ja takaisin tultaessa oikeaa. Visuaalinen ilme olisi informaatiografiikan tavoin pelkistetty, ja hahmot sympaattisen oloisia pallopäitä, jotka olisivat helposti animoitavissa ja joista saisi paljon komiikkaa irti. Tämän



Kuva 1: Animaatioiden luonnoksia

ajatuksen pohjalta kehitimme ideat 21 parille viiden sekunnin pätkiä, jotka löytyvät tämän työn liitteestä A. Asiakkaidemme kanssa olimme sopineet 20 parista, joten mielestämme heikoin, eli numero 11, "Kaupan risteyksessä", jäi pois.

Valitsimme pääasialliseksi työkaluksemme 3D-mallinnus ja -animaatio-ohjelma Blenderin. Blenderin tekee 3D-grafiikkaohjelmien joukossa erikoiseksi sen, että se on täydellinen paketti animaation ja stilligrafiikan tekoon, joka on saatavilla vapaan GPL (General Public Licence) -lisenssin alla. Sen lisäksi että ohjelma on vapaasti ladattavissa internetistä ja käytettävissä niin harrastepohjaisia kuin kaupallisia projekteja varten, myös sen lähdekoodi on vapaasti saatavilla, ja kenen tahansa on mahdollista osallistua kehitystyöhön. (blender.org – GPL for artists.) Virallisen julkaisun lisäksi onkin ladattavissa lukematon määrä kokeellisia versioita milloin milläkin kehittäellä olevalla lisällä, ja Blenderin uusia ominaisuuksia on kullakin hetkellä kehittäellä aina paremmista mallinnustyökaluista lähtien ominaisuuksiin, joiden avulla Blenderiä on mahdollista käyttää teollisuusrobottien ohjaukseen.

Me kolme Kaupunkietiketin tekijää tutustuimme Blenderiin ensimmäistä kertaa parin vuoden takaisessa työharjoittelussamme, jonka puitteissa kesän 2007 aikana itseopiskelimme Blenderin käytön, ja toteutimme sitä käyttäen Night of the Living Dead Pixels -lyhytanimaation. Blender teki meihin jo silloin vaikutuksen erittäin pätevänä ohjelmana, jonka aluksi hämmentävänoloisen käyttöliittymän alta paljastui mahdollisimman tehokkaaseen ja nopean työskentelyyn suunniteltu tehokas työkalu 3D-grafiikan tekoon. Blenderin poikkeavuus muista 3D-ohjelmista tuotti aluksi hämmenystä, mutta peruseriaatteet sisäistettyään huomasin, että Blender oli selvästi suunniteltu heille jotka ovat valmiita uhraamaan aikaa sen oppimiseen ja sisäistämiseen. Vastalahjana se tarjoaa muun muassa nopeat mallinnus- ja teksturointityökalut, monipuoliset ja työtä selkeyttävät työkalut hahmoanimaation tekoon, sisäänrakennetut toiminnot renderöidyn kuvan jälkikäsittelyyn ja leikkaukseen, sekä ryhmätyöskentelyä tukevan rakenteen.

Tykästyimme mainitun projektin aikana myös Blenderin käyttäjäyhteisöön, joka paitsi tarjoaa yksittäiselle käyttäjälle avuliaan ja innostuneen teknisen tuen erilaisten internetin keskustelupalstojen, sähköpostilistojen ja IRC-kanavien muodossa, myös

laajan ja teknisesti taitavan pohjan ohjelman tuleville versioille ja uusille ominaisuuksille. Näiden seikkojen varjossa Blender oli itsestäänselvä valinta meille Kaupunkietikettiä varten.

Valaistuksesta ja renderöinnistä kiinnostuneelle Blender on tällä hetkellä mielenkiintoinen, hieman ristiriitainen tapaus. Fotorealistiseen stilligrafiikkaan, kuten tuotevisualisointiin, Blender ja sen Blender Internal -nimellä yleensä kulkeva renderöintimoottori on jossain määrin rajoittunut. Se ei, ainakaan toistaiseksi, tarjoa esimerkiksi kehittyntä, todellista global illumination -ominaisuutta (3D Model Resources 2009), eikä fysikaalisesti korrekkeja materiaaleja (Hernández 2009). Blenderin kanssa on mahdollista käyttää näitä asioita mahdollistavia ulkoisia renderöintimoottoreita, kuten vapaat Yafaray ja LuxRender (blender.org – External Renderers). Nämä eivät kuitenkaan integroidu saumattomasti Blenderiin, ja esimerkiksi Blenderin renderöintitasot (Render Layers) tai sisäinen noodikomposiittori ei toimi niiden kanssa suoraan.

Animaatiota, ja varsinkin ilmeeltään tyyliä animatiota, tehdessä, realismia olennaisempaa on yleensä kuitenkin visuaalisesti näyttävä jälki ja alhaiset renderöintiajat. Näitä tavoitellessa Blender tarjoaa valaisun ja renderöinnin suhteen runsaasti mahdollisuuksia. Mainitun komposiittorin lisäksi esimerkiksi tästä ovat Approximate ambient occlusion -nimen alla kulkeva nopea tapa simuloida ympäristön yleisvalaistuksen vaikutusta, sekä Vector Blur -ominaisuus, jolla on nopeaa ja helppoa saada aikaan liikkeen epäterävyyttä. Näiden Blenderille erikoisten ominaisuuksien lisäksi Blenderissä on vahva tuki 3D-valaisun perusominaisuuksille, kuten erilaisille valoille ja pintamateriaaleille, sekä runsaasti muita animaatioprojektin toteuttamista helpottavia ominaisuuksia.

Tarkoitukseni on tässä opinnäytteeni kirjallisessa osiossa esitellä hieman näitä ominaisuuksia. On tietenkin mahdotonta antaa kaikissa tilanteissa päteviä ohjeita, saati sitten antaa ohjeita jokaiseen animatiota tehdessä eteen tulevaan tilanteeseen. Toivon kuitenkin, että esittelemällä Kaupunkietiketti-projektin aikana etemme tulleita haasteita, sekä niihin löytämiämme ratkaisuja on mahdollista antaa lukijalle kuvan siitä, millaisia ongelmia saattaa tulla vastaan, ja miten niistä on mahdollista selvittää Blenderin

tarjoamia työkaluja käyttäen. Uskon, että työstäni on hyötyä ensinnäkin heille, jotka käyttävät Blenderia, tai suunnittelevat käyttävänsä sitä, etenkin animaatioiden toteuttamiseen, mutta eivät varsinaisesti ole perehtyneet valaisuun ja renderöintiin. Toivon myös että mahdolliset 3D-grafiikasta jo kokemusta omaavat, ja varsinkin valaisusta, renderöinnistä ja jälkikäsittelystä kiinnostuneet, lukijat, kiinnostuisivat työni myötä Blenderistä ja sen tarjoamista mahdollisuuksista. Etenkin Blenderin erikoisemmat ominaisuudet uskon olevan tutustumisen arvoisia myös pääasiallisesti muiden ohjelmien parissa työskenteleville.

2 DATABLOKIT JA LINKITYKSET

Yksi Blenderin kätevimmistä toiminnoista, varsinkin laajemmissa projekteissa, joissa useampi henkilö joutuu työskentelemään samanaikaisesti jonkun tietyn asian parissa, on tiedostojen väliset linkitykset. Linkitykset ovat oikeasti vain tervetullut jatke tavalle jolla Blender käsittelee .blend-tiedostojen sisältämää dataa. Tämä perustuu niin sanottuihin datablokkeihin (datablock). Yksi datablokki sisältää kokonaisuuden tietyn tyyppistä dataa, ja datablokkit muodostavat keskenäisiä yhteyksiä. Tämä on ehkä helpoiten havainnollistettavissa tarkastelemalla lähemmin tapaa jolla Blender käsittelee 3D-maailmassa esiintyviä objekteja.

Blenderissä on joukko eri objektityyppejä, jotka ovat luultavasti enemmän tai vähemmän tuttuja muiden 3D-grafiikkaohjelmien parissa työskennelleille. Esimerkkejä näistä ovat mesh-objekti, kamera-objekti ja valo-objekti. Objektityypistä hieman riippuen jokaisella objektilla voi olla erilaisia ominaisuuksia, joista yhteisiä kaikille ovat nimi, sijainti (position), rotaatio (rotation) sekä skaalauskerroin (scale). Tyhjä Empty-objektia lukuunottamatta, jokaiseen objektiin kuuluu sen tyyppin mukainen datablokki. Esimerkiksi mesh-objekti on siten yhteydessä mesh-datablokkiin, joka pitää sisällään tiedot varsinaisesta polygonimeshistä. Mesh-datablokkiin saattaa puolestaan olla linkitetty esimerkiksi materiaali-datablokki, johon puolestaan tekstuuri-datablokki, ja niin edelleen. Alkuperäinen objekti-datablokki on puolestaan linkitetty siihen skene (scene) -datablokkiin, jonka edustamassa skenessä se esiintyy. Käytännössä kaikki Blenderin käsittelemä 3D-maailmaan liittyvä data sisältyy jonkin tyyppin datablokkiin.

(Mullen 2007, 18-22.)

Jokaisella datablokilla on ainutlaatuinen nimi oman datablokkityyppiryhmänsä sisällä, mutta esimerkiksi Blenderiä käynnistäessä oletusarvoisesti näkyvä kuutio on objekti nimeltä Cube, ja yhteydessä samannimiseen meshiin. Blenderin sisällä erityyppisiä datablokkeja erottaa toisistaan nimien edessä yleensä esiintyvistä tyyppimerkinnöistä, kuten OB (objekti), ME (mesh), LA ("lamp", valo), ja niin edelleen. Samalla datablokilla voi olla useampi käyttäjä, jolloin esimerkiksi suuri määrä samannäköisiä mesh-objekteja on helpointa toteuttaa linkittämällä ne samaan mesh-datablockiin. Tämä tarkoittaa myös sitä, että mikäli haluaa muuttaa jokaista, ei tarvittavia muokkauksia joudu tekemään kuin kerran, ja kaikki samaa dataa käyttävät objektit päivittyvät. Se saattaa toisaalta aiheuttaa välillä hämmennystä, varsinkin mikäli tämä järjestelmä ei ole tuttu, sillä objektia monistaessa kopioimalla siihen suoraan linkitetty datablokki, esimerkiksi mesh, monistuu myös, mutta tämä linkitetty datablokki käyttää edelleen samaa tietoa kuin se josta se on monistettu. Ellei tätä huomioi, sattuu helposti, että erehdyksessä muuttaa useamman datablokin jakamaa tietoa, kuten materiaaleja tai animaatiota.

Blenderin tallentamat .blend-tiedostot näkyvät käyttäjälle kokoelmana datablokkeja. Sen lisäksi että tiedoston datablokkit ovat yhteydessä toisiinsa tiedoston sisällä, datablokki saattaa olla yhteydessä muissa kuin omassa tiedostossa sijaitseviin blokkeihin. Blokkien linkittäminen tai lisääminen toisesta tiedostosta avattuun tiedostoon tapahtuu Blenderin File-valikon Append or Link -valintaa käyttäen. (Mullen 2007, 24)

Kaupunkietiketissä käytimme tätä ominaisuutta hyödyksemme ennen kaikkea siinä että hahmot ja animaatioissa käytettävä rekvisiitta oli mahdollista pitää omassa tiedostoissaan, erillään varsinaisesta animaatiosta. Pyrimme myös huomioimaan linkitysten tarjoamia hyötyjä materiaaleja ja valaistusta suunnitellessa, ja päädyimme pitämään käytännössä jokaisessa pätkässä käytettävää valaistusasetelmaa kirjastotiedostossa, josta sitä oli helppo linkittää animaatiotiedostoihin. Pidimme myös monimutkaisempia materiaaleja omassa omassa tiedostossaan, josta linkitimme ne edelleen ensin rekvisiitta-kirjastotiedostoihin, ja sieltä edelleen rekvisiitan mukana

varsinaisiin animaatiotiedostoihin. Materiaalien linkityksen suhteen olisi kylläkin ollut parantamisen varaa – olisi saattanut olla järkevämpää pitää alusta saakka kaikkia materiaaleja keskitetysti yhdessä tiedostossa.

3 MATERIAALIT JA VALAISU

3. 1 HAHMOJEN MATERIAALIT

Inspiraationa Kaupunkietiketin hahmojen ulkomuodolle toimi vinyylifiguurit, esimerkiksi Newyorkilaisen Kidrobot-yhtiön Munny- ja Dunny-hahmot. Tätä jatkaen haimme hahmoille hieman läpikuultavan kumimaista ilmettä. Helppo tapa saada Blenderissä aikaan läpikuultavuuden tunnetta on nostaa materiaalin Translucency-arvoa.

Translucency lisää materiaalin valoisuutta valonlähteestä päinvastaisella puolella, eli käytännössä vaalentaa tummimpia kohtia, jäljentäen valon kulkemista suoraan objektin läpi (Doc:Tutorials/Textures/Using Textures – BlenderWiki).

Läpikuultavuutta saa edelleen vahvistettua tekstuurinoodijärjestelmän avulla. Blenderissä ei ole, kuten joissakin muissa 3D-grafiikkaohjelmissa, täysin noodeihin pohjautuvaa materiaalijärjestelmää, vaan noodit ovat työkalu eri materiaalien yhdistämiseen ja muunteluun, ja monet materiaalien osa-alueet, kuten esimerkiksi raytrace-toiminnot, ovat ainakin toistaiseksi noodien ulottumattomissa.

Järjestelmä on toimiva, vaikkakin käyttöliitymältään hieman kankea – klikatessa materiaalinäkymästä noodinappia, sen vieressä näkyvän datablokin tyyppi muuttuu tyyppistä MA (material) tyyppiksi NT (node tree), ja alkuperäinen materiaalidatablokki ilmestyy uuteen valikkoon sen alle. Tämä materiaali toimii noodipuun input-arvona, ja sen arvoja muutetaan, kuten mitä tahansa materiaalia, materiaalinäkymän kontrolleilla. NT-noodi on noodipuun output-noodille syötetty materiaali, ja on se joka annetaan sitä materiaalinaan käyttävälle objektille. Noodipuuhun voi samanaikaisesti olla liitettyä useampi eri materiaaliblokki, ja NT-blokkia monistaessa siihen liitetyt materiaaliblokit eivät monistu mukana, eli niistä joutuu erikseen tekemään kopiot, mikäli sille on tarvetta.

Blenderin noodijärjestelmässä on kolmenlaisia liitoksia, joihin muita noodeja on

mahdollista kytkeä. Nämä ovat keltaiset pikseliliitokset, harmaat skalaariarvoliitokset, ja siniset vektoriliitokset. Erilaisten kokeilujen jälkeen päädyimme verrattain yksinkertaiseen materiaalinoodipuuhun. Input-materiaalista otimme vektoriliitoksen kautta materiaalin normaaliarvon, ja liitimme sen Normal-noodiin. Normal-noodin avulla on mahdollista muuttaa syötettyä normaaliarvoa, ja saada muutettu tulos joko vektorina tai skalaariarvona. Skalaari on mahdollista liittää ColorRamp-noodiin, jolle annoimme väriliun vaaleasta sävystä mustaan. ColorRampin tulos, joka on pikselityyppiä, yhdistettiin Add-tilaan asetetun Mix-noodin avulla alkuperäiseen input-materiaalin diffuusiarvoon, ja siitä edelleen output-noodiin. Tulos oli helppo tapa saada aikaan takaa tai sivusta tulevaa valoa, ilman että skeneen tarvitsee lisätä valoja sitä varten. Hahmojen materiaaleissa käytettiin tätä lisäämään alhaaltapäin tulevaa valoisuutta, jolla haettiin maasta heijastuvan valon ja lisää läpikuultavuuden vaikutelma.

Mix-noodin tarkkaa arvoa säädettiin sen myötä kun lopullinen ilme ja ympäristöjen valaistus varmistui, ja lopullinen arvo oli verrattain alhainen – noin 0.1. Syy tähän oli suurten arvojen luonnottomuus – tällä tavalla toteutettu valaistus lisätään jo varjostetun materiaalin päälle, eikä reagoi mitenkään valonlähteisiin tai niiden puutteeseen. Hienovaraisena tehosteena se oli kuitenkin käytännöllinen.

3.2 MUUT MATERIAALIT

Ympäristöjen objektien materiaaleista pyrimme tekemään mahdollisimman yksinkertaisia, mikä sen lisäksi että se istui pelkistettyyn yleisilmeeseen, oli myös aikataulun kannalta järkevää. Useimmissa tapauksissa käytimme oletusarvoisia Lambert-mallin diffuusivarjostinta ja Cook-Torrence spekulaarivarjostinta. Realistisemmän jäljen aikaansaamiseksi diffuusivarjostimen on Blenderissä mahdollista vaihtaa tyyppiä Oren-Nayar, jossa pinnan karheus otetaan varjostuksessa huomioon (CAVE | Projects: Oren-Nayar Reflectance Model).

Suurempaa, pinnan näkyvää variaatiota ja karheuden tunnetta varten, käytimme valituissa paikoissa DistortedNoise-proseduraalitekstuuria, joka asetettiin vaikuttamaan pintanormaaliin, eli se mikä on monelle 3D-grafiikkaa harrastaneelle tuttua nimellä

bump-mapping. Kannattaa huomioida että Noise-nimellä kulkeva tekstuurityyppi lasketaan joka kuvaruudussa uudelleen, joten se tuottaa animaatioissa kohinaa, DistortedNoise taas ei. Jälkimmäinen vääristää yhtä kohinatekstuuria toisella, mutta asettamalle vääristyksen arvoksi nolla sillä on mahdollista saada samaa jälkeä kuin Noisella.

Kätevä internetsivusto materiaaleja suunnitellessa on osoitteessa <http://matrep.parastudios.de/> toimiva Blender Open Material Repository, josta löytyy laaja valikoima eri kategorioihin jäsenneltäviä, vapaasti käytettäviä materiaaleja, jotka ovat usein käyttökelpoisia joko suoraan tai pienellä muokkauksella. Kaupunkietikettiä varten löysimme sieltä toimivat pohjat asfaltille, ruoholle sekä kudotulle villalle.

Metallimateriaaleja varten toimivaksi tavaksi saada realistisempaa tunnetta todettiin heijastustekstuurien käyttö. Blenderissä on mahdollista asettaa tekstuurikordinaateiksi heijastus (Refl), jolloin tekstuuria voi käyttää niin sanottuna environment mappina, eli ympäristön heijastuksena. Tämä on suunniteltu käytettäväksi etenkin Blenderin oman EnvMap -tekstuurityypin kanssa, joka luo skenen ympäristöstä heijastustekstuurin objektille, mutta sitä on mahdollista käyttää myös muiden tekstuurien kanssa.

Tekstuurin on oltava kuutiokartta (cubemap) -tyyppiä, eli objektin heijastama ympäristö on sen ympärille asetettu kuutio, ja jokainen kuution sivuista saa tekstuurinsa yhdestä kuutiokartan osasta. Emme löytäneet Blenderin dokumentaatiosta missä järjestyksessä sivujen tulisi olla, mutta EnvMap-tekstuurilla tehtyjen kokeilujen avulla saimme sen selvitettyä. Sen perusteella oli mahdollista luoda oikeanlainen kuutiokartta, jossa ympäristön ylempi puolisko oli vaalea ja alempi tumma, simuloiden kirkasta taivasta. Kun tämän tekstuurin asetti vaikuttamaan materiaalin väri (Col) ja valoisuus (Emit) -ominaisuuksiin sopivan alhaisilla arvoilla, saatiin aikaan metallissa näkyvien heijastuksia muistuttava hienovarainen tehoste, joka lisäsi materiaalin tunnistettavuutta poikkeamatta yleisilmeestä liikaa. Koska raytrace-menetelmään perustuvia heijastuksia tai vastaavaa ei käytetty, tämä ei lisännyt renderöintiäikoja käytännössä yhtään.

Näitä heijastuksia käyttävä materiaali lisättiin materiaalikirjastotiedostoon, josta sen pystyi linkittämään rekvisiittakirjastoihin, joissa sitä käytettiin. Perusmetallia käytettiin

sellaisenaan useassa paikassa, ja sellaisissa tapauksissa, joissa oli tarve muutoksille esimerkiksi värisävyn suhteen, siitä tehtiin kulloinkin kyseessä olevaan rekvisiittatiedostoon lokaalikopio, ja muutettiin siinä.

Yleisilmeen selkeyden ja ajankäytön takia bittikarttatekstuureja ei edellämäinittua lukuunottamatta Kaupunkietiketissä juuri käytetty. Joitakin poikkeuksia tähän kuitenkin oli, kuten esimerkiksi muutamassa klipissä näkyvä bussipysäkki, johon lisättiin linja-auton kuvalla varustettu kyltti. Näissä tapauksissa oli tarpeellista käyttää Blenderin UV unwrap -työkaluja tekstuurin saamiseksi oikeaan paikkaan.

3.3 ANIMAATIOIDEN VALAISTUS

Vaikka kaikkien animaatioiden yhteenlaskettu pituus ei ollut muutamaa minuuttia pitempi, niiden suuri määrä yhdistettynä tiukkaan aikatauluun tarkoitti, ettei valaistusta ollut mahdollista suunnitella joka kerta alusta alkaen. Myös yhtenäisyyden kannalta tuntui järkevältä, että olisi kirjastotiedostossa muutama perusvalo, joita lisättäisiin jokaiseen animaatiotiedostoon. Kohtauskohtaisesti olisi sitten mahdollista muokata näitä perusvaloja, ja tarpeen mukaan luoda lisää valonlähteitä.

Nämä "perusvalot" koostuivat meillä kolmesta valosta. Päävalo oli ylhäältä suunnattu spottivalo, etupäässä koska Spot-tyypin valo on ainoa Blenderin valotyyppi joka tukee puskurivarjoja (buffer shadow, myös depth map shadow). Puskurivarjot ovat yleisin suuren mittakaavan 3D-animaatio ja elokuvatuotannossa käytetty varjotyyppi, koska ovat renderöinnin kannalta verrattain nopeita laskea. Ne perustuvat siihen, että ennen renderöintiä luodaan jokaiselle varjolle oma varjokartta (shadow map), johon tallennetaan matka valosta jokaiseen pisteeseen johon se osuu. Lopullista kuvaa renderöidessä tarkistetaan, onko renderöitävä piste sama, joka on tallennettu varjokarttaan, ja ellei, sen päätetään olevan kyseiseltä valolta varjossa. Varjokartan tarkkuus on muutettavissa, jolloin on mahdollista poistaa liian epätarkan kartan luomia artefakteja kartan viemän muistimäärän kasvun kustannuksella. Puskurivarjojen hyöty on, että ne ovat verrattain nopeita laskea. (Birn 2006, 55-56.)

Yksi puskurivarjojen ongelma on osittain läpinäkyvät objektit, jotka eivät heitä

luonnollisen näkyviä, läpinäkyviä varjoja. Joissakin renderöintijärjestelmissä on mahdollista käyttää niin sanottuja syviä varjokarttoja (deep shadow map), joihin tallentuu pisteen varjo- ja valoisuuden lisäksi myös tieto siitä, miten paljon jokin piste varjostaa. Nämä soveltuvat osittain läpinäkyvien ja läpikuultavien materiaalien lisäksi myös esimerkiksi ruohon ja hiusten tai karvojen valaisemiseen. (Birn 2006, 61.) Blender ei kuitenkaan, ainakaan toistaiseksi, niitä tue.

Toinen 3D-valaistuksessa käytetty varjotyyppi ovat raytrace-menetelmään perustuvat varjot, jotka ovat usein hitaampia laskea, varsinkin monen ison studion käyttämässä RenderMan -renderöintimoottorissa. Ne perustuvat siihen, että jokaisen renderöitävän pisteen kohdalla tarkistetaan, onko sen ja valonlähteen välissä varjoa heittävä objekti, ja mikäli näin on, pisteen päätetään olevan varjossa. Raytrace-pohjaiset varjot ovat oletusarvoisesti aina täysin teräväreunaisia, mutta heittämällä suurta määrää varjoja lähekkäisistä sijainneista, on niiden avulla mahdollista luoda luonnollisen näköisiä varjoja, jotka ovat varjoa heittävän esineen lähellä teräviä, ja muuttuvat matkan myötä sumuisemmiksi. (Birn 2006, 62-68.)

Päätimme Kaupunkietiketissä keskittyä käyttämään puskurivarjoja nopeuden takia, ja koska muuten haalea värimaailma kaipaisi epäteräviä varjoja. Päätös osoittautui oikeaksi myös siitä syystä, että kääntämällä materiaalin raytrace-näkyvyyden päälle ja pois, sen vaikutusta ambient occlusion -varjostukseen, on mahdollista kontrolloida. Mikäli raytrace-pohjaisia varjoja ei käytä, tällä ei ole vaikutusta valojen aiheuttamiin varjoihin.

Puskurivarjoja käyttäessä saattaa ilmetä ongelmallisia artefakteja, kuten viivoja varjoalueiden reunoilla, tai valovuotoja, jotka yleensä ovat korjattavissa varjokartan Bias-arvoa muuttamalla. Birn (2006, 56-60) selittää tätä tarkemmin Digital Lighting and Rendering -kirjassaan. Spottivaloja käyttäessä tulee myös varoa, että valon muodostaman kartion reuna ei aiheuta ongelmia esimerkiksi leikkautumalla häiritsevästi kuva-alueen sisällä.

Päävalon lisäksi perusvaloihin kuului kaksi täytevaloa, jotka olivat Blenderin Hemi-tyyppiä. Hemi-tyypin valot tuottavat valoa puolipallon muotoiselta alueelta, heittämättä

varjoa, ja soveltuvat siksi hyvin tällaiseen tarkoitukseen. Toinen täytevaloista sijoitettiin suoraan hahmojen ja ympäristöjen yläpuolelle, ja toinen, hieman pienemmälle teholle säädetty, suoraan alapuolelle. Koska hemivalot eivät heitä varjoja, alapuolelle sijoitettu, ylöspäin käännetty valo paistaa suoraan mahdollisen maapinnan läpi, ja yhdessä molemmat täytevalot vaalensivat skeneä kauttaaltaan, ja teki varjoista vähemmän jyrkkiä. Molempien täytevalojen ominaisuuksista oli kytketty valon spekuarikomponentti pois päältä, mikä teki niiden vaikutuksesta tasaisen.

Blenderissä on tasojen (Layers) avulla mahdollista määrittellä tarkasti mihin objekteihin tietyn valon vaikutus yletty. Aktivoimalla Layers-napin valon ominaisuuksista se valaisee ainoastaan samalla tasolla sijaitsevia objekteja. Blenderissä sama objekti voi samanaikaisesti sijaita useammalla tasolla, joten yhden valon valaisemien objektien ei tarvitse sijaita keskenään samalla tasolla, vaan valon voi sijoittaa usealle tasolle, jolloin se vaikuttaa niihin kaikkiin. Blenderin versiosta 2.48 lähtien on myös mahdollista määrittää ainoastaan valon kanssa samalla tasolla sijaitsevat objektit heittämään varjoja. Kaupunkietiketin valaistuksessa käytimme valon vaikutuksen rajaamista hyödyksi muun muassa siinä, että alapuolelle sijoitettu täytevalo ei vaikuttanut hahmojen silmiin, jotka olisivat muuten hohtaneet luonnottoman valkoisina.

Näitä jokaiseen kohtaukseen lisättyjä perusvaloja piti muuttaa jonkin verran perusasetelmasta lähes jokaisessa tapauksessa. Päävalon tarkkaa paikkaa siirrettiin jonkin verran riippuen kohtauksen ympäristöstä ja kamerakulmasta, jotta oudon näköisiä tai häiritseviä varjoja syntyisi mahdollisimman vähän. Myös täytevalojen voimakkuuksia säädettiin jonkin verran, riippuen siitä oliko kyse ulkoilmaan vai sisälle sijoittuva kohtaus. Valon paikan muuttaminen edellytti sitä että valon objekti-datablokista teki lokaalin, ja valon voimakkuuden tai muun ominaisuuden muuttamiseksi myös Lamp (LA) -data piti olla paikallista. Tämän jälkeen ne eivät tietenkään enää reagoi kirjastotiedostossa tehtyihin muutoksiin, mutta vastapainoksi yhdestä animaatiotiedostosta on mahdollista linkittää tai kopioida valot toiseen, jossa käytetään samanlaista valaistusta, mikä monesti säästi aikaa.

Päävalon ja kahden lisävalon lisäksi saatettiin kohtauksiin lisätä tarpeen mukaan lisävaloja. Perusasetelman päävalo oli sijoitettu verrattain suoraan ylhäältäpäin

vaikuttavaksi, ja kauas, jotta se muistuttaisi mahdollisimman paljon auringonvaloa eikä tuottaisi häiritsevästi eri suuntiin meneviä varjoja. Tämän takia sen tuottamat spekulariheijastukset jäivät usein huomaamattomiksi, ja jotta joidenkin objektien pintoihin tulisi tarpeeksi vaihtelua, oli joissain tapauksissa tarpeellista lisätä skeneen yksittäisiä, varjoja heittämättömiä ja ainoastaan spekulariheijastukseen vaikuttavia, valoja. Jossain ulkoilmakohtauksissa tällaista käytettiin myös maapinnan alapuolella lisäämään vaikutelmaa maasta kirkkaalla auringonvalolla esimerkiksi hahmoihin kohdistuvaa heijastuvaa valoa.

Koska täytevalot eivät heittäneet minkäänlaisia varjoja, ne valaisivat luonnottomasti myös sellaisia kohtia jotka olisivat reaali maailmassa olleet täysin varjossa. Kaupunkietiketin voimakkaasti tyylitellyn ulkonäön takia tämä ei yleensä ollut suuri ongelma, mutta joissakin kohdissa koimme tarpeelliseksi lisätä tummentavia, eli vaikutukseltaan negatiivisia valoja. Blenderissä valosta voi tehdä tummentava aktivoimalla Negative-kohdan sen ominaisuuksista. Tämän jälkeen on usein tarpeellista määritellä sen vaikutusalueita Dist- ja Falloff -ominaisuuksia käyttäen. Molemmat yllä selitetyt valot, sekä spekulareja lisäävät että tummentavat, toteutettiin pistevaloilla, jotka Blenderissä kulkevat nimellä Lamp, ja muissa 3D-grafiikkaohjelmissa nimillä kuten point light tai omni light.

3.4 AMBIENT OCCLUSION -VARJOSTUS

Ambient occlusion (AO) -nimellä kulkeva ominaisuus on varsinkin liikkuvassa 3D-grafiikassa usein käytetty tapa simuloida ympäristöstä kantautuvaa yleisvaloa, kuten esimerkiksi taivaalta joka suunnasta tulevaa valoisuutta, tai ympäristöstä heijastuvaa valoa. Perinteisesti AO toimii tarkastelemalla raytrace-menetelmällä jokaisen pisteen läheisyyden muihin objekteihin, eli ampumalla pisteestä säteitä eri suuntiin tietyn matkan verran, ja tarkistamalla miten moni säteistä osuu johonkin objektiin. Mitä useampi säde osuu objektiin, sitä enemmän "varjossa" piste oletetaan olevan, ja sitä enemmän sitä tummennetaan. (Birn 2006, 73.)

Blenderissä on versiosta 2.46 lähtien ollut raytrace-pohjaisen AO:n lisäksi myös nimellä

Approximate ambient occlusion (AAO) kulkeva vaihtoehtoinen menetelmä. Tämä, joskus myös nimellä point-based AO tunnettu, menetelmä perustuu siihen, että jokainen verteksi oletetaan tummentavan tietyn kokoisen alueen ympärillään. Laskemalla miten monta tällaista aluetta vaikuttaa tietyn renderöitävän pisteen kohdalla on mahdollista laskea sen oletettu varjoisuus. Raytrace-pohjaisen AO:n tarkkuus kasvaa ja lopputuloksen kohina vähenee sen myötä kun sen näytteenottotaajuutta kasvatetaan, mutta samalla kasvavat myös laskenta-aika. AAO sen sijaan on lähtökohtaisesti kohinaton. (van Lommel 2008.)

AAO:ssa on kokemuksistamme päätellen kuitenkin myös omat ongelmansa. Sitä voi joskus olla vaikea hallita, ja varsinkin tiheän geometrian alueet saattavat tummentua liikaa. Blenderin, mahdollisesti verrattain huonosti optimoitu, tapa laskea se alkaa myös hidastelemaan kun polygonien määrä nousee satoihin tuhansiin. Tätä välttääksemme jouduimme käyttämään AAO:ta Kaupunkietiketissä mahdollisimman harkitusti. Päädyimme siihen, että sen aiheuttama tummennus oli tehokeino, jota käytettäisiin hahmojen vaikutuksessa ympäristöön sekä joissakin muissa, harkituissa paikoissa. Pyrimme myös siihen, että kyseessä olisi nimenomaan tehokeino, jota käytettäisiin mahdollisimman hienovaraisesti.

Materiaaliasetuksissa on Traceable-valinta, jonka avulla on mahdollista kontrolloida mikäli kyseinen materiaali tuottaa raytrace-varjoja. Harhaanjohtavasti tämä on merkitty "Makes material detectable by ray tracing", mutta valinta ei vaikuta mitenkään raytrace-pohjaisiin heijastuksiin, ja tässä tapauksessa "ray tracing" koskee myös AAO:ta. Kytkemällä Traceable pois päältä meidän oli siis mahdollista estää geometrialtaan monimutkaisia objekteja vaikuttamasta renderöijän AAO-laskelmiin ja siten hidastamasta renderöintiä liikaa. Koska käytimme valoissa ainoastaan puskurivarjoja, objektien heittovarjot eivät kuitenkaan jääneet pois.

Loppujen lopuksi Traceable oli päällä käytännössä ainoastaan hahmojen materiaaleissa, sekä erikoisessa "occlusion"-materiaalissa, joka oli muuten näkymätön, mutta tuotti ainoastaan raytrace-varjoja. Viimeksimainittu on mahdollista kytkemällä materiaalista päälle OnlyCast, jolloin siitä tulee kameralle näkymätön, ja tuottaa ainoastaan varjoja, ja Shadbuf pois päältä, jolloin materiaali ei rekisteröidy puskurivarjoihin. Tätä

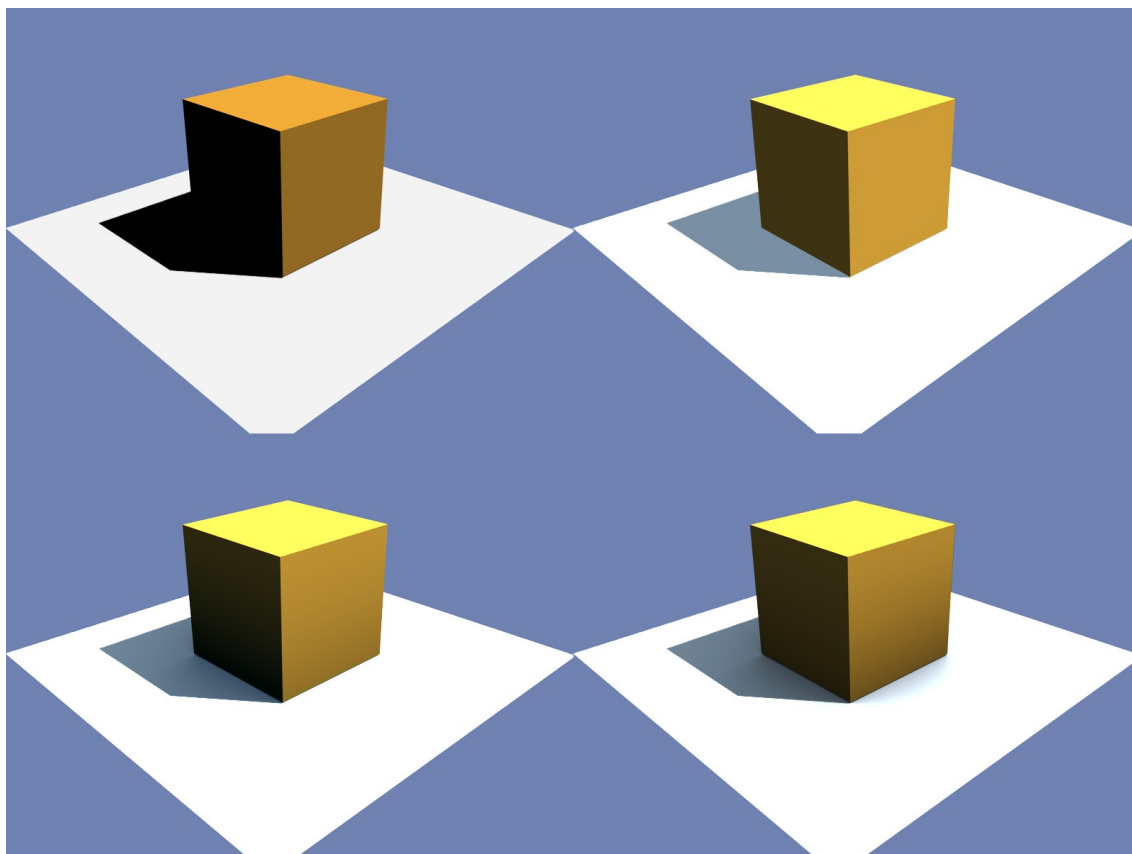
occlusion-materiaalia annettiin esimerkiksi kuutioille tai muulle yksinkertaiselle muodolle, jota pystyi sijoittamaan paikkaan jota toivoi tummentuvan. Näin AO:n vaikutusta sai kohdistettua tarkasti, ilman että sen laskeminen vaikutti merkittävästi renderöinti-aikoihin.

AO:n varjostuksen voimakkuutta on mahdollista määritellä joko globaalisti, AO:n asetuksista, tai materiaalikohtaisesti, materiaaliasetusten Amb-säätimellä. Amb on oletusarvoisesti asetettu verrattain korkealle, arvoon 0,5 nolasta yhteen ulottuvalla skaalalla. Useiden materiaalien kohdalla vähensimme tätä 0,1 - 0,2 välille, jottei materiaali tummenisi liikaa. Myös hahmoissa suunnittelimme alunperin näkyvän AO:n vaikutusta, mutta täysin musta tummentuminen ei sopinut yhteen hahmojen kylläisten värien kanssa. Jätimme sen hahmojen perusmateriaaleista pois, ja lisäsimme sitä sen sijaan erikseen sopiviin kohtiin, esimerkiksi hattupäisen hahmon päähän, ja kohtaan jossa suuri joukko hahmoja ovat tiiviisti pakkautuneet pieneen tilaan.

Kuten nimikin kertoo, ambient occlusion -varjostus on suunniteltu toimimaan yhdessä valaistuksen ambient-komponentin kanssa (zap's mental ray tips 2008). Ambientilla tarkoitetaan sitä ympäristön perusvaloa, johon muun valoisuus lisätään. Tämä on yleensä lähtökohtaisesti mustaa eli olematonta, kuten myöskin Blenderissä. Ambient-valaistuksen väriä voi tarkistaa 3D-grafiikkaohjelmassa luomalla skenen jossa on yksi varjoa heittävä valo, sekä jokin varjoa heittävä objekti. Yleensä varjojen väri on tässä tapauksessa täysin musta, mikä tarkoittaa että myös ambient-valo on mustaa.

Muuttamalla ambient-valon väriä (Blenderissä Shading-valintojen alla kohdassa World) on mahdollista muuttaa varjojen väriä ja lisätä niihin valoisuutta, mutta näin syntyvä varjojen vaalentuminen on täydessä tasaisuudessaan kovin luonnotonta, ja sitä neuvotaan yleensä välttämään, ja käyttämään sen sijasta täytevaloja. Liian tasaisen varjojen valoisuuden ongelma saattaa ilmetä myös täytevaloja käytettäessä, mikäli täytevalot eivät heitä varjoja. Mikäli tarkastelee varjoja ympärillään huomaa, ettei varjot ole ihan täysin tasaisia, vaan tummuus vaihtelee riippuen esimerkiksi siitä miten lähellä varjostava esine on varjostettua pintaa. Tätä on mahdollista simuloida asettamalla täytevalot heittämään varjoja, jolloin saattaa syntyä häiritseviä

päällekkäisiä varjoja. Toinen mahdollisuus on hyödyntää tähän tarkoitukseen ambient occlusion -varjostusta. Blenderissä voi AO:n asettaa joko vaalentamaan ambient-valoa, jolloin se ajaa suoraan täytevalojen virkaa (asetuksista Add), tummentamaan sitä (asetuksista Sub), tai sekä tummentamaan että vaalentamaan (Both).



Kuva 2: Ambient Occlusion -varjostus. Ylävasemmasta myötäpäivään: yksinomaan päävalo, päävalo ja täytevalo, AO kompositoitu jälkepäin (väärä tapa) , AO:n vaikutus näkyy vain varjoissa (oikea tapa)

AO:ta näkee usein käytettävän niin, että se renderöidään erillisenä kuvana ulos, jolloin sen vaikutus on kaikkialla yhtä tasainen, ja kompositoidaan jälkepäin renderöidyn kuvan päälle multiply-operaatiolla. Tämä johtaa kuitenkin siihen, että AO:n vaikutus näkyy kaikkialla yhtä voimakkaana, eikä ota muuta valoisuutta huomioon. Usein kuulee sanottavan, että vaikutelma on "likainen". (zap's mental ray tips 2008.)

Oikeampi tapa on pyrkiä rajoittamaan vaikutusta niihin alueisiin joissa ambient-komponentin, eli ympäristöstä tulevan valon, vaikutus olisi voimakkaimmillaan, eli varjoihin (zap's mental ray tips 2008). Blenderissä ei ole suoraa tapaa rajoittaa AO:n

vaikutusta vain esimerkiksi jonkin tietyn valon valaistuksen ulkopuolelle, mutta tarkalla valojen voimakkuuksien ja materiaalien Amb-arvojen säädöllä se on mahdollista. AO tummentaa silloin myös muualla, mutta vahvan päävalon ansiosta sen vaikutus jää käytännössä olemattomaksi varjojen ulkopuolella.

4 KOMPOSITOINTI

Yksi Blenderin käytettävimmistä ominaisuuksista valaisutyöskentelyn kannalta on sisäänrakennettu noodikomposiittori. Sen ansiosta on mahdollista tehdä värikorjailuja, säätää renderöidyn kuvan valoisuutta ja kontrastia, ja jopa yhdistää erillään renderöitäviä kuvia, samanaikaisesti varsinaisen 3D-työskentelyn kanssa. Vaikka lopullinen kompositointi tapahtuisikin erillisessä ohjelmassa, voi tällä tavalla saada kuvan siitä millaiset korjaustoimenpiteet ovat mahdollisia ja järkeviä tehdä renderöidylle kuvalle jälkikäteen, ja mitkä vaativat skenen valaisun tai materiaalien muuttamista ja kuvan uudelleenrenderöintiä.

Komposiittori on käytettävissä suoraan avaamalla Node Editor -iikkuna, valitsemalla siitä komposiittinoodit ja klikkaamalla "Use nodes". Jotta komposiittorin tulos näkyisi myös renderöidyn kuvan ikkunassa, pitää renderöintivalikosta vielä valita Do composite, jolloin Composite-nimiselle noodille syötetty kuva näytetään. Tuloksen tarkasteluun ovat käytännöllisiä myös Viewer ja SplitViewer -noodit. Viewerillä voi tarkastella sille syötettyä kuvaa, ja SplitViewerillä vertailla kahta kuvaa vierekkäin. Aktivoimalla noodieditoriikkunasta valinnan "Backdrop" saa viimeksi valitun Viewer-noodin kuvan näkymään noodinäkymän taustalla. Lopuksi löytyy vielä File Output -noodi, jolla kuvan voi tallentaa haluamaansa paikkaan ja tiedostomuotoon, jolloin esimerkiksi erillistä kompositointiohjelmaa käytettäessä, eri tasoilta renderöidyt kuvat on mahdollista tallentaa erikseen ja yhdistää jälkeensä.

Vaikka Blenderin komposiittori on integraationsa ansiosta erittäin käytännöllinen työkalu, siinä on kuitenkin omat rajoituksensa, minkä takia erillisen kompositointiohjelman käyttö saattaa olla tarpeen. Se ei tue minkäänlaista keyframe -animaatiota, joten stattiisten muutosten aikaansaaminen on helppoa, mutta ajan kanssa muuttuvien asetusten käyttö on hankalampaa. Periaatteessa ne ovat

toteutettavissa Time -noodin, joka antaa käyrän määrittelemän arvon nollan ja yhden väliltä ajan funktiona, sekä erilaisten matemaattisten operaatioiden, avulla, mutta tämä ei ole kovin käyttäjäystävällistä.

Tämä oli suurin syy miksi päädyimme myös Kaupunkietiketissä tekemään lopullisen jälkityöstön Adoben After Effects -ohjelmassa. Kaupunkietiketin jokainen pätkä vaatisi loppuun sekunnin mittaisen pysähdyskuvan ja oikein- tai väärinmerkin, mikä olisi huomattavasti vaivattomampaa toteuttaa After Effectsin aikajanapohjaisessa järjestelmässä. Tämän lisäksi jokaiseen sisältyisi osia Nelosen kanavailmeestä, jotka oli toimitettu meille After Effects -projektitiedostoina. Tämä ei kuitenkaan tarkoittanut että Blenderin komposiittori olisi jäänyt täysin käyttämättä, ja se tarjosi edellä mainittujen värikorjauksen esikatselumahdollisuuksien lisäksi myös helpon ja nopean tavan saada aikaan myös motion blur - tunnettua liike-epäterävyttä.

4.1 LIIKE-EPÄTERÄVYYS JA VECTOR BLUR -NOODI

Liike-epäterävyys on se, joka kameralla kuvatessa syntyy kohteen liikkuaessa filmin valoituksen aikana. Fotorealismisuutta tavoitellessa sen jäljittäminen on 3D-grafiikassa välttämätöntä, mutta myös tyyliteltyä animaatiota tehdessä se on käytännöllinen keino saada aikaan pehmeämpiä liikkeitä.

Yksinkertaisin tapa epäterävöittää liikkeitä Blenderissä on aktivoida renderöintiasetuksista MBLUR. Blender renderöi silloin jokaisen kuvaruudun yhteydessä myös edeltävät ja seuraavat kuvaruudut, ja luovat näiden perusteella epäterävyyttä renderöitävään kuvaan. Epäterävyyden määrää voi valita MBLUR-napin alapuolella olevalla asetuksella. Perinteisissä elokuvakameroissa suljin on toteutettu filmin edessä pyörivällä metallilevyllä, joka on osittain avoin. Aukon asteissa ilmoitettu suuruus kertoo miten paljon filmiä valoitetaan – 360 asteen aukko on jatkuvasti avoin. Blenderin epäterävyyden oletusarvo 0,5 vastaa 180 asteen aukkoa, mikä antaa luonnollisen näköisen lopputuloksen. (Birn 2006, 165-167.)

Edelläselitetyn menetelmän ongelma on sen hitaus. Jokaisen renderöidyn kuvan eteen joudutaan renderöimään useamman, lähes samanlaisen kuvan, ja renderöintiajat nousevat moninkertaisiksi. Kompositointiohjelmassa, kuten myös Blenderin noodikomposiittorissa ja Adobe Photoshopissa, on tavallisesti Motion blur -nimen alla kulkeva toiminto, joka epäterävöittää kuvaa valittuun suuntaan, mutta joitakin poikkeustapauksia lukuunottamatta se ei ole käytännöllinen tähän tarkoitukseen. Tällainen toiminto ei ota huomioon liikkuvan kappaleen eikä liikkeen kolmiulotteisuutta, vaan epäterävöittää tasaisesti yhteen suuntaan, valitun määrän mukaisesti.

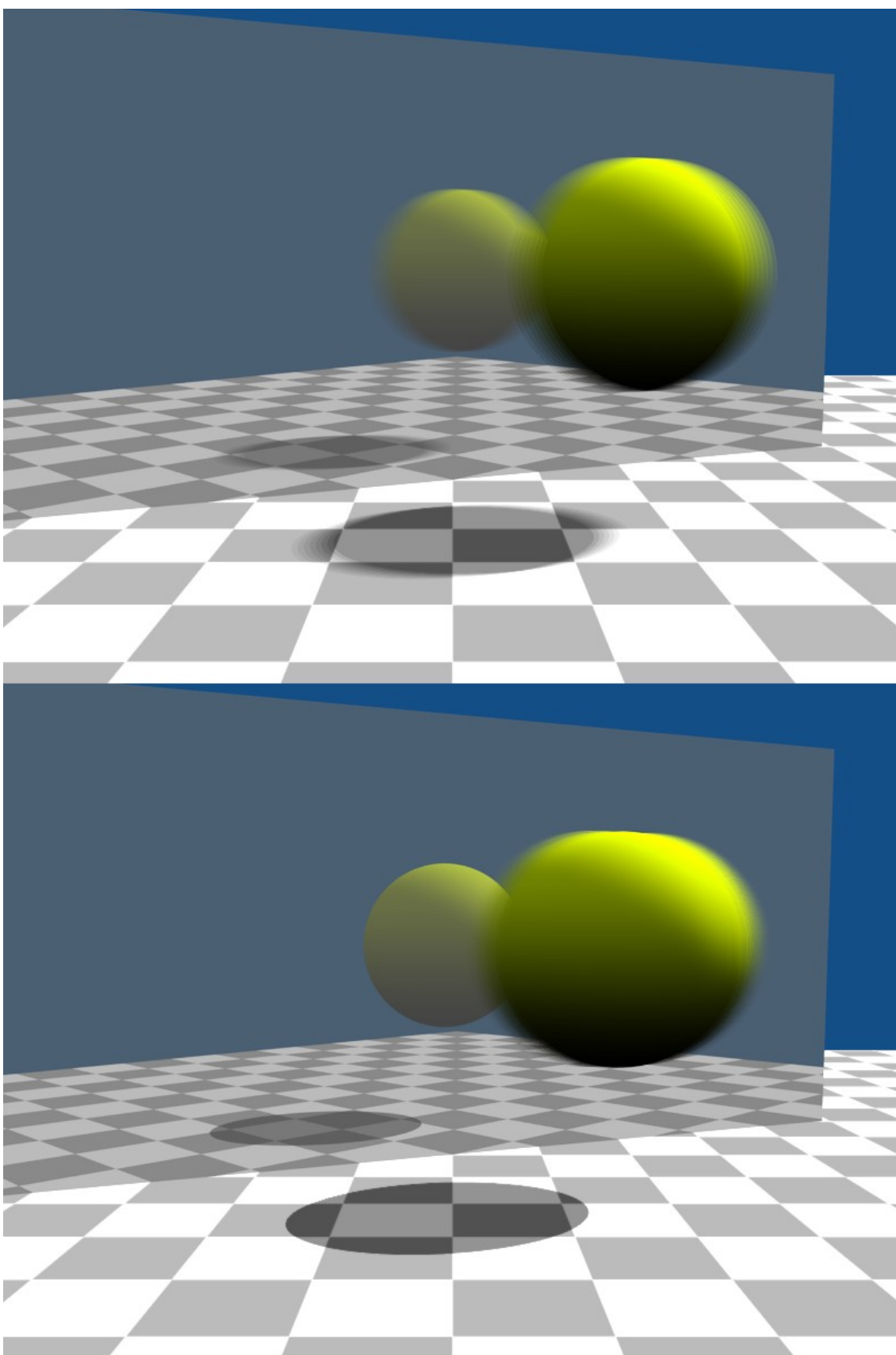
Blender tarjoaa kuitenkin Vector Blur -nimisen filterinoodin muodossa kolmannen tavan, joka tuottaa kohtalaisen realistisen näköistä jälkeä, ilman merkittävästi pidempiä renderöintiaikoja. Vector Blur -noodille syötetään käsiteltävän kuvan lisäksi Z-pass, eli tietoa kuvan syvyydestä, siitä mikä on jokaisen kuvan pikselin ja kameran välinen etäisyys, sekä renderöintilayerieissä oletusarvoisesti poiskytketty Speed- tai Vector pass, joka sisältää tietoa siitä mihin suuntaan ja millä nopeudella kukin kuvan pikseleistä sillä hetkellä liikkuu. Vector pass lasketaan aina ensimmäisenä ennen varsinaisen kuvaruudun renderöinnin alkamista, ja prosessi vie joitakin sekunteja.

Vector Blur -filteri aloittaa luomalla käsiteltävästä kuvasta kolmiulotteisen mallin, jossa jokaisesta kuvan pikselistä tehdään polygoni, ja järjestetään ne keskenään syvyyden perusteella. Renderöidessä laskettujen vektoreiden perusteella filteri epäterävöittää jokaista jokaista polygonia erikseen, laskien ensin oikean suunnan ja määrän epäterävöitykselle kolmiulotteisen vektorin sekä polygonin syvyyden pohjalta. Tämän jälkeen filteröidyt polygonit koostetaan uudestaan kuvaksi. (blender.org – Vector Blur.)

Vector Blur -noodi tarjoaa käyttäjälle mahdollisuuden muuttaa filterin tarkkuutta sekä epäterävöityksen määrää. Tarkkuuden oletusarvo 32 saattaa joissain tapauksissa, esimerkiksi objektin liikkeessa kameran läheltä, tuntua liian alhaiselta, ja sen nostaminen esimerkiksi 64:ään tuottaa pehmeämpää jälkeä ilman että filteröinnin viemä aika juurikaan kasvaisi. Mikäli filteri epäterävöittää liikaa tai liian vähän, sen vaikutusta on mahdollista hienosäätää MinSpeed, MaxSpeed ja BlurFac -asetuksilla. Koska filteröinti tehdään jälkeempäin jo renderöityyn kuvaan, tämä ei vaadi kuvan uudelleenrenderöintiä.

Vector Blurrilla on kuitenkin oman heikkoutensa, jotka ovat hyvä tiedostaa. Koska sumentaminen tapahtuu objektin liikkeen perusteella, liikkuvien objektien tuottamat varjot ja heijastukset pysyvät ennallaan, elleivät objektit joihin ne osuvat ole itsessään liikkeessä suhteessa kameraan. Vector Blur on myös riippuvainen Blenderin tuottamasta vektoridatasta, joten noodikomposiittorin useimmista muista toiminnoista poiketen sitä ei ole mahdollista käyttää Blenderin ulkopuolelta tuodun kuvadatan kanssa.

Kaupunkietiketissä käytimme Vector Blur -filtteriä kaikissa animaatioissa niin, että se oli ainoa Blenderin komposiittorissa tehty toimenpide joka sisällytettiin tallennettuihin renderöinteihin, kun jälkityöstötoimenpiteet muuten keskittyivät After Effectsin puolelle.



Kuva 3: Liike-epäterävyys. Ylempi kuva: tavallinen liike-epäterävyys (Blenderissä MBLUR-valinta) antaa luonnollisen lopputuloksen (renderöintiäika: 2 min 48 s). Alempi kuva: Vector Blur -noodi on huomattavasti nopeampi, mutta ei vaikuta heittovarjoihin tai heijastuksiin (renderöintiäika 18 s).

4.2 GAMMA-KORJATTU TYÖSKENTELY

Niin sanottu "linear workflow", eli gamma-korjattu työskentely, on aihe, joka on viimeisen parin vuoden aikana herättänyt runsaasti keskustelua internetin 3D-aiheisilla keskustelupalstoilla. Kyse on siitä, että pyritään minimoimaan erilaisten gamma-arvojen aiheuttamaa sekaannusta 3D-grafiikkaa työstäessä.

Gamma-arvot ja gammakorjaukset ovat perua ensimmäisten kuvaputkinäyttöjen ajoilta. Kuvaputket eivät reagoi lineaarisesti niille syötettyyn virtaan, niin että viidenkymmenen prosentin virran voimakkuus vastaisi viidenkymmenen prosentin valoisuutta, vaan valoisuus lisääntyy alhaisilla virroilla nopeammin (Poynton 2002). Ratkaisuna tähän käytetään niin sanottua gammakorjausta. Useimmissa Windows-koneissa kuvaa korjataan ennen näyttämistä gamma-arvolla joka on jotakuinkin 2,2. (de Zwart 2005.)

Tämä on kuitenkin ongelma 3D-grafiikkaa tehdessä, sillä renderöinti tapahtuu niin sanotussa lineaarisessa väriavaruudessa, eli ilman gammakorjausta, jossa skenen valon voiman lisääminen tietyllä määrällä lisää valoisuutta. Gammakorjauksen jälkeen tämä ei kuitenkaan enää päde, ja yrittämällä kumota sen tummentavan vaikutuksen valojen tehoja lisäämällä lopputuloksesta ei tule korrekti valojen ja varjojen välisen kontrastin, ja esimerkiksi valon "falloffin", eli vähenemisen kuljetun matkan myötä, suhteen. (de Zwart 2005.)

Oikea ratkaisu on sen sijaan päinvastaisen gammakorjauksen tekeminen renderöinnin lopuksi. Mikäli näytön gamma-arvo on 2,2, päinvastainen korjaus on $1 / 2,2$ eli noin 0,45. Lopullisesta gamma-arvosta tulee tällöin $2,2 * 0,45$, eli jotakuinkin 1, toisin sanoen lineaarinen gamma. (Bardak 2007.)

Monessa 3D-ohjelmassa, kuten esimerkiksi Autodesk 3ds maxissa ja Blenderin kanssa yhdessä toimivassa ulkoisessa renderöintimoottori Yafarayssa, on valmis asetus renderöidyn kuvan gammakorjauksen arvolle. Blenderissä tätä ei suoraan ole, mutta noodikomposiittori korvaa tämän puutteen.

Pitää kuitenkin myös ottaa huomioon, että bittikarttatekstuureja muokatessa tämä tapahtuu gammakorjatussa väriavaruudessa, ja jotta tekstuuri näkyisi lineaariseen gammaan korjatussa kuvassa samalta, sille pitää muokkauksen lopuksi antaa gammakorjaus joka vastaa näytön gammaa (Bardak 2007). Joissakin ohjelmissa myös tämä on tehtävissä kerralla kaikille tekstuureille ohjelman asetuksista, ja Blenderissä tekstuurien gammakorjaus on versiosta 2.49 tehtävissä uusien tekstuurinoodien avulla. Toinen toimiva tapa on tehdä se kuvankäsittelyohjelmassa jossa tekstuuria muokataan, tai vaihtoehtoisesti tallentaa tekstuuri tiedostomuotoon joka lähtökohtaisesti käyttää lineaarista gammaa, kuten esimerkiksi OpenEXR (Johnson 2008).

Kaupunkietiketin valaisutyöskentelyä varten aloitimme kalibroimalla näyttöjä mahdollisimman lähelle gamma 2,2:ta. Näytön kalibrointi ei ole ikinä täysin tarkkaa, etenkin ilman erikoislaitteistoa, mutta toisaalta televisiografiikkaa tehdessä ei ikinä pysty täysin ennakoimaan lopullisen katseluun käytettävän laitteen näyttämiä värejä, joten koimme että käyttämämme tapa oli tarkoituksiimme riittävä. PC-koneilla käytimme Nvidian näytönohjainten ajureiden mukana tulevaa kalibrointiohjelmaa, Mac-koneiden OS X -käyttöjärjestelmän mukana tulee valmiina ColorSync-niminen työkalu. Molemmat perustuvat siihen että käyttäjä silmäämäärisesti arvioi koneen näyttämiä kuvia, ja näytön kuva säädetään sen perusteella mahdollisimman neutraaliksi. ColorSync antaa sen lisäksi lopuksi mahdollisuuden valita näytön gamma-arvon numeroskaalalta. PC-koneiden näytönohjainajureista löytyi gamma-asetus, mutta se ei näyttänyt eri arvoja suoraan. Ratkaisu oli internetistä ladattava kalibrointikuva, jossa harmaasävyä arvioimalla pysty asettamaan gammaa haluttuun arvoon. Tällaista käyttäessä kannattaa kuitenkin varoa, ettei kuvaa tarkastele siitä mahdollisesti löytyviä väriprofiileja huomioon ottavassa ohjelmassa, kuten Photoshop, tai yli kolmosversion Mozilla Firefox -selain. Käytimme tähän tarkoitukseen Internet Explorer 7 -selainta.

Valoja muokatessa Blenderissä oli ensimmäinen toimenpide aina kompositointinoodien aktivoiminen ja Gamma-noodin lisääminen. Gamma-noodille annettiin arvo 0,45, ja se pidettiin aina viimesimpänä noodina ennen Composite ja Viewer -noodeja, jotta kaikki kuvan mahdolliset muokkaukset tehtäisiin lineaarisessa väriavaruudessa.

Gammakorjauksen jälkeinen kuva oli miltei aina hieman haalea, jota korjattiin Gamma-

noodin eteen asetetulla RGB Curves-noodilla. Valaistus tehtiin niin, että se näiden noodien läpi näytti oikealta. Lopullinen kuva tallennettiin File Output -noodeilla niin, että RGB Curves ja Gamma -noodien vaikutus ei tallentunut niihin.

Renderöintien tallentamiseen valitsimme OpenEXR -formaatin, koska se on avoin, sekä Blenderin että After Effectsin hyvin tukema 32-bittinen HDR -kuvaformaatti. HDR (High Dynamic Range) tarkoittaa, että värikanavat ovat tallennettu siihen perinteisen kahdeksan bitin kanavakohtaisen kokonaislukuarvon sijasta 32-bittisenä liukulukuna, mikä tarkoittaa että kuvan valoitusta ja kontrastia on mahdollista muuttaa jälkeempään ilman että kuvanlaatu siitä kärsii (Birn 2006, 241-242).

After Effectsin avattuna OpenEXR-kuvasarjat näyttivät samalta kun gammakorjattu kuva Blenderissä, minkä perusteella oletimme että After Effects tunnistaa kuvan gamman, ja muuttaa sen automaattisesti 2.2:een, emmekä enää muttaneet kuvan valoitusta tai gamma-arvoa. After Effectsissä suoritettu värikorjailu tapahtui lähinnä Curve-tehosteen avulla. Toteutimme klippien loppujen pysähdyskuvat jokaisen kohdalla monistamalla viimeiseksi renderöidyn kuvan, lisäsimme Photoshopissa tehdyt Oikein- ja väärin -merkit sekä Nelosen kanavailmeen elementit, ja tallensimme ulos lopulliset animaatiot meille annetuilla asetuksilla.

5 POHDINTAA LOPUKSI

Jälkiviisaana on monta asiaa joita olisi Kaupunkietiketin tuotannon aikana voitu tehdä paremmin. Valaisu- ja renderöintityön joutui suorittamaan melko lailla liukuhihnaperiaatteella, vaikka allekirjoittanut olisi mielellään uppoutunut niihin vielä enemmän. Paremmalla tekijätiimin sisäisellä työnjaolla ja aikataulutuksella tämä olisi luultavasti ollut mahdollista.

Toinen huomattava ongelma on se, että emme missään vaiheessa projektia tarkastaneet miltä kuva näytti televisioruudulta katsottuna. Sopivan laitteiston puutteessa jouduimme turvautumaan siihen, miltä kuva näytti tietokoneen ruudulta, sekä valvomaan After Effectsin väripipetillä, etteivät kirkkaiden värien arvot nousseet

liian korkeiksi.

Vaikka Blender ei varsinaisesti tunnu rajoittuneelta, se ei tarkoita että materiaali- ja valaisutyökaluissa olisi parantamisen varaa. Materiaalinoodijärjestelmä on nyky muodossaan kohtalaisen sekava, mikä on kylläkin enemmän käyttöliittymäongelma kuin tekninen rajoite. Suurempi ongelma on se, että yksittäiset materiaaliasetukset ovat usein vaikeaselkoisia, ja ohjaavat samanaikaisesti useita ominaisuuksia. Esimerkkinä Traceable-valinta kontrolloi samanaikaisesti materiaalin näkyvyyttä AO:lle, raytrace-pohjaisille varjoille ja raytrace-pohjaisille heijastuksille. Huomattavasti käytännöllisempää olisi jos nämä valinnat olisivat erillisiä, ja selvästi merkittyjä. Mahdolliset monimutkaisemmat ratkaisut, kuten esimerkiksi objekti, joka näkyy heijastuksissa, mutta ei näy suoraan kameralle eikä heitä varjoja, vaatisi eri elementtien renderöinnin erikseen, ja niiden yhdistämisen jälkeenpäin, joka Blenderin komposiittorissa tai jossain muussa ohjelmassa.

Yhteistä Blenderin kaikille valoille on, että niistä voi kytkeä vaikutuksen materiaalien diffuusi- ja spekuhaarivarjostimiin päälle ja pois. Ei kuitenkaan ole mahdollista kontrolloida valon eri komponentteja tarkemmin. Erikoisemman valon, kuten esimerkiksi sellaisen, jonka diffuusikomponentti on voimakas, mutta spekuhaarikomponentti heikko, ja jonka komponentit ovat eri väriä, ei ole mahdollista luoda käyttämättä kahta eri valoa, joista toinen tuottaa pelkkää diffuusivaikutusta ja toinen pelkkää spekuhaarivaikutusta. Valoille ei myöskään ole Blenderissä noodijärjestelmää, jolla kahden valon asetuksia saisi yhdistettyä, kuten materiaaleissa.

Ambient-valoon ei ole Blenderissä mahdollista vaikuttaa muulla tavalla kuin globaalin asetuksen kautta. Joissain tapauksissa saattaisi olla hyödyllistä vaikuttaa siihen paikallisesti, valo-objektin avulla. Tämä voisi olla joko oma valotyyppinsä, tai nykyisissä valotyypeissä voisi tarkemmin määriteltävien diffuusi- ja spekuhaarikomponenttien lisäksi olla ambient-komponentti, jonka vaikutus olisi oletuksena asetettu olemattomaksi. Olisi myös kätevää, mikäli ambient occlusion -varjostuksen vaikutus olisi mahdollista sitoa valoihin, niin että se ei vaikuttaisi ollenkaan esimerkiksi tietyn valon vaikutusalueilla.

Tällaiset ominaisuudet eivät vastaisi valon luonnollista käyttäytymistä, mutta koska

Blender ei nykyiselläänkään pyri mallintamaan valon käyttäytymistä tieteellisen korrektilla tavalla, tämä on huono perustelu niiden poisjättämiselle. Sen sijaan tulisi pyrkiä antamaan graafikolle mahdollisimman monipuolisia työkaluja. Nykyisellään Blender on ongelmallisesti sijoittunut kahden maailman – luonnollisuuden ja mahdollisimman monipuolisten mahdollisuuksien – väliin.

Kannattaa kuitenkin huomioida, että näissä heikkouksissa on kyse asioista, jotka sijoittuvat jo huomattavasti välttämättömien perusominaisuuksien ulkopuolelle, ja että Blender on jo nyky muodossaan täysin toimiva Kaupunkietiketin kaltaisen projektin pääasialliseksi 3D-työkaluksi.

LÄHTEET

3D Model Resources 2009. Carrara vs Blender 3D. 3D Model Resources 9.4.2009. [Verkkodokumentti] <<http://www.ideacosmo.com/3d-model/useful-information/carrara-vs-blender-3d/>> (luettu 22.5.2009).

Bardak, Harry 2007. Colorspaces in XSI. XSIBlog 6.1.2007. [Verkkodokumentti] <<http://www.xsi-blog.com/archives/133>> (luettu 17.5.2009).

Birn, Jeremy 2006. Digital Lighting & Rendering. Toinen painos. Berkeley: New Riders.

blender.org – External Renderers. [Verkkodokumentti] <<http://www.blender.org/download/get-blender/external-renderers/>> (luettu 22.5.2009).

blender.org – GPL for artists. [Verkkodokumentti] <<http://www.blender.org/education-help/faq/gpl-for-artists/>> (luettu 22.5.2009).

blender.org – Vector Blur. [Verkkodokumentti] <<http://www.blender.org/development/release-logs/blender-242/vector-blur/>> (luettu 23.5.2009).

CAVE | Projects: Oren-Nayar Reflectance Model. Computer Vision and Graphics Center, Department of Computer Science, Columbia University. [Verkkodokumentti] <<http://www1.cs.columbia.edu/CAVE/projects/oren/>> (luettu 22.5.2009).

Doc:Tutorials/Textures/Using Textures – BlenderWiki. [Verkkodokumentti] <http://wiki.blender.org/index.php/Doc:Tutorials/Textures/Using_Textures> (luettu 22.5.2009).

Hernández, Raúl Fernández 2009. Blender Internal + unbiased rendering. True Volumetrics for Blender 4.2.2009. [Verkkodokumentti] <<http://farsthary.wordpress.com/2009/02/04/blender-internal-unbiased-rendering/>> (luettu 22.5.2009)

Johnson, David 2008. linear workflow and gamma. djax blog 13.9.2008. [Verkkodokumentti] <<http://www.djax.com.au/blog/2008/09/13/linear-workflow-and-gamma/>> (luettu 17.5.2009).

van Lommel, Brecht 2008. Approximate Ambient Occlusion. Big Buck Bunny Blog Archive 11.1.2008. [Verkkodokumentti] <<http://www.bigbuckbunny.org/index.php/approximate-ambient-occlusion/>> (luettu 22.5.2009).

Mullen, Tony 2007. Introducing Character Animation with Blender. Indianapolis: Wiley Publishing, Inc.

Poynton, Charles 2002. Gamma FAQ – Frequently Asked Questions about Gamma. <http://www.poynton.com/notes/colour_and_gamma/GammaFAQ.html> (luettu 17.5.2009).

zap's mental ray tips 2008. The Joy of a little "Ambience". [Verkkodokumentti] <<http://mentalraytips.blogspot.com/2008/11/joy-of-little-ambience.html>> (luettu 17.5.2009).

de Zwart, Gijs 2005. Linear workflow 'reloaded'. <<http://www.gijsdezwart.nl/tutorials.php>> (luettu 17.5.2009).

LIITE A

KAUPUNKIETIKETIN ANIMAATIOIDEN SYNOPSISIT

1. Elokvateatterissa h l tys

V  RIN: Hahmot istuvat elokvateatterissa. Osa hahmoista h l tt  tauotta ep selv ll   n nell  (sanoista ei saa selv  ). Muut vaikuttavat n rk st neilt . Taustalla kuuluu elokuvan  niraita.
(+ Valon vaihtelut)

OIKEIN: Hahmot istuvat elokvateatterissa. Kaikki ovat hiljaa. Taustalla elokuvan  ni.

2. Metro, ovien edess  odottaminen

V  RIN: Metro pys htyy. Laiturilla odottavat hahmot ovat suoraan ovien edess . Metrosta pois j  v t eiv t p  se ulos.

OIKEIN: Metro pys htyy. Laiturilla odottavat hahmot ovat ryhmittyneet ovien ym rille j tt en selke n kulkureitin. Metrosta pois j  v t k velev t reitist  ulos, sitten muu massa siirtyy sis  n.

3. Rullaportaat, vasemmalla seisokelu

V  RIN: Hahmot seisovat paikallaan rullaportaiden oikealla puolella. Yksi hahmoista k velee yl s vasenta puolta ja pys htyy  kisti paikalleen. Takaa tulevat muut vasemman puolen kulkijat t rm  v t seisovaan hahmoon ja pakkautuvat massaksi.

OIKEIN: Hahmot seisovat paikallaan rullaportaiden oikealla puolella. Muut hahmot kulkevat ohi vasemmalta.

4. Koiranpaskan ker  minen

V  RIN: Kuvassa n kyy koiranpaskap ntt . Hahmo seisoo kuvassa koiran kanssa. Koira ulostaa maahan. Koira ja hahmo k velev t pois.

OIKEIN: Hahmo ja koira seisovat kuvassa. (Koira ulostaa maahan). Hahmo ker   paskan vieress  olevaan p ntt  n.

5. Pullojen rikkominen

VÄÄRIN: Hieman päihtyneen oloinen hahmo seisoo kuvassa. Taustalla näkyy ruohoa. Hahmo heittää pullon olkansa yli, ääni kertoo sen menevän rikki. Hahmo heiluu päihtyneenä.

OIKEIN: Hieman päihtynyt hahmo heittää pullon taakseen nurmikolle, josta toinen hahmo äkkiä kerää sen pussiin. Päihtynyt hahmo jatkaa heilumista paikallaan.

6. Rullaportaat, uloskäynnin eteen jääminen

VÄÄRIN: Kolme hahmoa saapuu rullaportaiden yläpään. Pois jäätyään he jäävät keskustelemaan aivan uloskäynnin kohdalle. Takana tulevat pakkautuvat ihmismassaksi heidän taakseen.

OIKEIN: Kolme hahmoa saapuu rullaportaiden yläpään ja kävelee pois. Takana tulevat tekevät samoin.

7. Kännykällä musiikin kuuntelu

VÄÄRIN: Hahmoja istuu junassa. Yksi hahmoista kuuntelee kovaäänistä ärsyttävää jytettä kännykästään. Muut hahmot vaikuttavat ärsyyntyneiltä, pitelevät korviaan tms. (Kännykän päällä myös ilmestyy ja katoaa erilaisia nuotteja)

OIKEIN: Kännykkää kuunteleva hahmo laittaa kuulokkeet korvilleen, jumputus laantuu hädintuskin kuultavalle tasolle. Muut hahmot ovat selkeästi helpottuneita.

8. Junan ovet aukeavat molempiin suuntiin

VÄÄRIN: Hahmoja odottelee junan eteisessä. Juna pysähtyy asemalle. Osastosta ulos yrittävä hahmo yrittää väkisin avata oven ahtaaseen eteiseen päin koputtaen sillä toistuvasti edessä seisovaa matkustajaa.

OIKEIN: Juna pysähtyy asemalle. Osastosta ulos yrittävä huomaa ettei ovella ole tilaa aueta eteiseen, ja avaa sen osastoon päin ja jää sitten pois.

9. Istuminen junan penkeillä

VÄÄRIN: Hahmo röhnöttää täydessä junassa vieden kahden penkin tilan. Viereen saapuu toinen hahmo, joka jää seisomaan koska penkillä ei ole tilaa.

OIKEIN: Hahmo röhnöttää täydessä junassa. Huomatessaan kanssamatkustajan hahmo korjaa asentonsa. Toinen hahmo istuu alas.

10. Elokvateatteri, kännykän valo

VÄÄRIN: Hahmot istuvat elokuvateatterissa. Yhtäkkiä yksi hahmoista kaivaa kännykän esiin. Kännykän valo valaisee teatterin. Muut vaikuttavat närkästyneiltä.

OIKEIN: Hahmot istuvat elokuvateatterissa.

11. Kaupan risteyksessä

VÄÄRIN: Kaupan "risteykseen" saapuu muutama hahmo kärryjen kanssa. Hahmot pysähtyvät ja jäävät kälättämään keskelle risteystä. Taakse muodostuu valtava ruuhka.

OIKEIN: Hahmot jäävät kälättämään käytävien sivuille. Muut hahmot pääsevät ohi mainiosti.

12. Ostoskärryillä vierekkäin puskeminen

VÄÄRIN: Kaksi hahmoa liikkuu kaupassa ostoskärryjen kanssa vierekkäin. Takana tulee muita hahmoja jotka kuikuilevat kahden muun yli ja etsivät ohituspaikkaa, mutta eivät löydä.

OIKEIN: Kaksi hahmoa liikkuu kaupassa ostoskärryjen kanssa peräkkäin. Muut hahmot pääsevät vaivatta ohi vierestä.

13. Kuseksiminen ympäriinsä

VÄÄRIN: Joukko hahmoja katselee hieman hätäisinä ympärilleen kaupunkiympäristössä. Hetken päästä kaikki alkavat kusemaan ympäriinsä.

OIKEIN: Etualalla silinteripäinen hahmo painaa nappia jolloin ympäristöön nousee useita bajamajoja. Hahmot jouksevat niihin sisään.

14. Invapaikan käyttö

VÄÄRIN: Julkisen välineen invapaikalla istuva ei ymmärrä antaa paikkaansa vanhukselle.

OIKEIN: Julkisen välineen invapaikalla istuva hahmo ymmärtää antaa paikkansa vanhukselle.

15. Invapaikan käyttö vol 2

VÄÄRIN: Invapaikalla istuvan (musiikkia kuuntelevan) hahmon selän takana oleva vanhus tyytyy vain huokailemaan äänekkäästi.

OIKEIN: Vanhus koputtaa hahmoa olkapäähän, jolloin tämä luovuttaa paikkansa.

16. Baaritiskillä notkuminen juoman saannin jälkeen

VÄÄRIN: Etualalla pari ihmistä saa juomansa, ja jäävät niille sijoilleen hörppimään. Takana ihmiset yrittävät kuikuilla heidän selkensä takaa.

OIKEIN: Etualan hahmot saavat juomansa ja siirtyvät seuraavien tieltä pois. Jono rullaa jouhevasti.

17. Rullakassin kasaus

VÄÄRIN: Hahmo jää metrosta pois. Astuttuaan ulos hahmo nostaa kassin sentin päähän ovista ja alkaa kasaamaan sitä. Takana tulevat eivät pääse ulos.

OIKEIN: Hahmo jää metrosta pois. Astuttuaan ulos hahmo kantaa rullakassin ovien viereen pois kulkuväylältä ja alkaa kasata sitä. Muut pääsevät rauhassa ulos.

18. Metroon kiirehtiminen

VÄÄRIN: Hahmo juoksee juuri lähtevään metroon. Matkalla hahmo kaataa metrosta ulos tulevia hahmoja, eikä lopuksi ehdi metroon.

OIKEIN: Hahmo kävelee metrolaiturille josta juuri lähtee metro. Hahmo odottaa noin sekunnin, jolloin paikalle saapuu toinen metro.

19. Matkatavarat hattuhyllylle

VÄÄRIN: Hahmo istuu junassa rinka vierellään käytävällä. Toiset hahmot eivät pääse ohi, vaan pakkautuvat kasaan ja kaatuilevat rinkan yli.

OIKEIN: Hahmon rinka on hattuhyllyllä. Muut matkustajat kävelevät ohi.

20. Pysäkitupakointi

VÄÄRIN: Hahmo polttaa pysäkin sisällä. Toiset hahmot köhivät.

OIKEIN: Hahmo polttaa pysäkin ulkopuolella. Toiset hahmot seisoskelevat rauhassa sisäpuolella.

21. Pysäkitupakointi vol 2

VÄÄRIN: Hahmo polttaa pysäkin ulkopuolella. Pysäkin sisältä kävelee hahmo ulkopuolelle ja alkaa köhimään.

OIKEIN: Hahmo polttaa pysäkin ulkopuolella. Toiset hahmot seisoskelevat rauhassa sisäpuolella.

LIITE B

VALMIIT ANIMAATIOT JA TYÖTIEDOSTOT