

KARELIA-AMMATTIKORKEAKOULU  
Viestinnän koulutusohjelma

Harri Pietiläinen

FOTOREALISMI BLENDERISSÄ JA BLEND4WEBISSÄ:  
Karelia-amk:n CHP-voimalan 3D-visualisointi

Opinnäytetyö  
Syyskuu 2016



**OPINNÄYTETYÖ**  
**Syyskuu 2016**  
**Viestinnän koulutusohjelma**

Länsikatu 15  
80110 JOENSUU  
p. 013 260 600

**Tekijä**  
Harri Pietiläinen

**Nimeke**  
Fotorealismi Blenderissä ja Blend4Webissä: Karelia-amk:n CHP-voimalan  
3D-visualisointi  
Toimeksiantaja  
Karelia-ammattikorkeakoulu, Sirkkalan Energiapuisto / Ville Kuittinen

**Tiivistelmä**

Tämän toiminnallisen opinnäytetyön tavoitteena oli toteuttaa selainpohjainen 3D-visualisointi Karelia-ammattikorkeakoulun opetus- ja tutkimuskäytössä olevasta CHP-voimalasta. Toteutusvälineiksi valikoituivat Blender- ja Blend4Web CE -ohjelmat, jotka ovat kumpikin GPL-lisensioituja, avoimeen lähdekoodiin pohjautuvia sovelluksia.

Opinnäytetyön raportointi koostuu kolme osasta. Ensimmäisessä osassa selvitetään CHP-laitoksen rakennetta ja toimintaa. Toisessa osassa tutkitaan sitä, mitä fotorealismi tarkoittaa ja miten materiaalien fyysisiin ominaisuuksiin ja valon käyttäytymiseen perustuvat PBR-materiaalit määritellään teoriatasolla. Kolmas osa keskittyy siihen, miten edellä olevia periaatteita sovelletaan käytännössä niin Blenderissä kuin Blend4Webissä. Työ päättyy projektin ja työvälineiden analyysiin ja pohdintaan.

Työn aikana ilmeni, että valon fysiikkaan perustuvien fotorealististen PBR-materiaalien mallintaminen vaatii syvällistä perehtymistä asiaan. Käytetyt ohjelmistot tarjoavat kuitenkin melko hyvät ja toimivat välineet materiaalien toteuttamiseen 3D-ympäristöissä.

Opinnäytetyön tuloksena syntyi 3D-visualisoinnista raakaversio, jonka avulla CHP-laitoksen rakenteeseen voi tutustua nettiselainta käyttäen. Sovellusta kehitetään muun muassa käyttöliittymää hiomalla ja lisätoimintojen avulla.

**Kieli**  
suomi

Sivuja 92  
Liitteet  
Liitesivumäärä

**Asiasanat**  
3D-visualisointi, fotorealismi, PBR-materiaalit, Blender, Blend4Web



**THESIS**  
**September 2016**  
**Degree Programme in Media**  
Länsikatu 15  
FI 80110 JOENSUU  
FINLAND  
Tel. +358 13 260 600

Author  
Harri Pietiläinen

Title  
Photorealism in Blender and Blend4Web: 3D visualization of the Karelia UAS's CHP plant  
Commissioned by  
Karelia University of Applied Sciences, Sirkkala Energy Park / Ville Kuittinen

Abstract

The nature of this thesis is functional, with the object of implementing a browser-based 3D visualization of the CHP plant that is used in Karelia UAS for teaching and research purposes. The visualization was implemented using Blender and Blend4Web CE software, both of which are open source and GPL licensed.

The report is divided into three parts. The first part deals with the CHP plant itself, its structure and how the plant actually works. The second part studies photorealism and how PBR materials (physically based rendering) are defined on a theoretical level. The third part focuses on implementing the theoretical principals in practice, using Blender and Blend4Web. The thesis ends with the analysis and discussion of the project and implementation tools.

During the project it became clear that implementing photorealistic PBR materials requires a rather deep knowledge of how light behaves on different materials. However, Blender and Blend4Web provide practical and handy tools for implementing the materials in 3D environments.

The result of this thesis was a fairly rough version of the web-based 3D visualization that can be used for exploring the CHP plant. The visualization will be finalized after the thesis project by enhancing the interface and adding extra functionality.

Language  
Finnish

Pages 92  
Appendices  
Pages of Appendices

Keywords

3D visualization, photorealism, PBR materials, Blender, Blend4Web

## Sisällysluettelo

1	Johdanto.....	8
2	CHP-laitoksen rakenne ja toiminta .....	9
2.1	Yleiskuvaus .....	9
2.2	Sähköntuotanto .....	10
2.3	Lämmöntuotanto.....	11
2.4	Visualisointi.....	13
3	Fotorealistinen visualisointi .....	13
3.1	Määritelmä.....	13
3.2	3D-mallintamisesta lyhyesti .....	14
3.3	Fotorealistiset materiaalit.....	16
3.4	Valon fysiikasta ja materiaalien ominaisuuksista.....	17
3.4.1	Valo ilmiönä .....	17
3.4.2	Läpinäkyvyys ja läpikuultavuus .....	18
3.4.3	Kiinteät materiaalit.....	19
3.4.4	Pinnan mikrotason rakenne.....	21
3.4.5	Valon taittuminen .....	22
3.4.6	Kaustiikka .....	23
3.5	Materiaalien mallintaminen.....	24
3.5.1	PBR-materiaalien perusteita.....	24
3.5.2	Diffuusiväri.....	25
3.5.3	Läpinäkyvät tekstuurit.....	26
3.5.4	Bittikarttojen heikkoudet .....	27
3.5.5	Proseduraaliset tekstuurit .....	27
3.5.6	Kiiltävyys.....	28
3.5.7	Normaalikartat .....	29
3.5.8	Kohoumakartat .....	31
3.5.9	Poikkeutuskartat.....	31
3.5.10	Ympäristövarjostuskartat .....	32
3.6	Muut yksityiskohdat .....	32
4	Blender.....	33
4.1	Blenderin perusteita.....	33
4.2	Cycles.....	34
4.2.1	Taustaa lyhyesti.....	34
4.2.2	Cycles-materiaalien määrittäminen .....	35
4.2.3	Node Editorista lyhyesti .....	36
4.2.4	PBR-eriste .....	38
4.2.5	PBR-metalli.....	45
4.2.6	PBR-lasi.....	52
4.2.7	Tekstuurien kehittäminen Cyclesissa.....	54
4.3	Blender Internal .....	58
4.3.1	Taustaa lyhyesti.....	58
4.3.2	BI – Materiaalit .....	59
5	Blend4Web / WebGL.....	68
5.1	Lyhyt esittely.....	68
5.2	Materiaalien määrittäminen .....	69
5.3	Materiaalikirjasto.....	75

## Sisällysluettelo, jatkoa

6	Materiaalien analysointi.....	78
6.1	CHP-laitoksen pintamateriaalit.....	78
6.2	Blend4Web ja visualisointi.....	80
6.2.1	Kiiltävä teräspinta.....	80
6.2.2	Messinki.....	82
6.2.3	Valurauta.....	83
6.2.4	Maalatut metallipinnat.....	84
6.2.5	Muovi.....	84
6.2.6	Lasi.....	86
6.2.7	Kumi.....	87
7	Pohdinta.....	87
7.1	Projektin tulokset.....	87
7.2	Blend4Web työvälineenä.....	88
7.3	Blend4Webin tulevaisuudennäkymät.....	90
	Lähteet.....	91

## Sanasto

albedo	albedokartta: määrittelee kappaleen perus- eli mattaväriin, voidaan tehdä tavallisesta valokuvatekstuurista poistamalla valoheijastukset ja varjot (Wilson 2016a).
ambient occlusion (AO)	AO-kartta / ympäristövarjostus (Joensuu 2016): erityisen laskentakaavan mukaan luotu kartta, määrittelee ympäristö- eli hajavalon perusteella kohdat joita lähellä olevat kappaleet tai kappale itse varjostavat, kuten kappaleiden alapuoliset osat, nurkat jne. (Birn 2006).
baking	kehittäminen (Joensuu 2016) / leipominen (Ilama 2014): laskentaprosessi jossa voidaan luoda esimerkiksi kartta- tai tekstuuri suoraan kappaleen perusteella (AO- ja normaalikartat) tai yhdistää varjo- tai valokuvioita suoraan kappaleen pintatekstuuriin (Joensuu 2016).
bump map	kohouma- tai kuhmukartta (Puhakka 2008): luo illuusion siitä, että kappaleen pinnassa on epätasaisuutta, kuten naarmuja, halkeamia, tai kohoumia, mutta ei muuta kappaleen geometriaa (Birn 2006).
BSDF	matemaattinen funktio, jonka avulla voidaan määrittellä valon hajaantuminen kappaleen pinnassa (Wikipedia 2016), käytetään muun muassa Blenderin Cycles-renderöintimoottorin sävyttimissä (Blender Foundation 2016).
caustics	kaustiikka (Kontu 2014): valon heijastumisen ja/tai taittumisen aikaansaamat valoilmiot esimerkiksi peileistä, lasiesineistä ja muista erityisen kiiltävistä pinnoista tai läpinäkyvistä kappaleista (Birn 2006).
cavity map	kolokartta: käytetään joskus lisäämään pieniä yksityiskohtia kappaleen pintaan muuttamatta kappaleen rakennetta, vrt. kouhouskartta (Wilson 2016a).
decals	siirto- tai tarrakuva: osittain läpinäkyvä, valokuvaan perustuva kuvatekstuuri, jota käytetään lisäämään kappaleen pintaan esimerkiksi tekstejä, ruostetta, valumajälkiä ym. yksityiskohtia (Birn 2006).
diffuse color	diffuusiväri: se osa valosta, jonka kappaleen materiaali luonnostaan oman rakenteensa vuoksi heijastaa takaisin (Birn 2006).
displacement map	poikkeutuskartta (Puhakka 2008): kuten kohouskartta, mutta muuttaa kappaleen geometriaa ja edellyttää siksi että kappaleen määrittävä monikulmioverkko on tiheämpi (ks. high-poly model) (Birn 2006).
gloss map	karkeus- tai sileyskartta: määrittää miten karkea kappaleen pinta on eri kohdissa, mikä vaikuttaa siihen miten terävänä heijastus nähdään (Birn 2006).
high-poly model	high poly -malli (Joensuu 2016), tarkasti mallinnettu 3D-kappale, jossa yksityiskohtia on tuotettu tiheämmän monikulmioverkon avulla. Näiden avulla voidaan mm. tuottaa tarkkoja normaalikarttoja low-poly -malleja varten (Birn 2006).
image / color map	valokuvaan tai piirroksen perustuva värikartta: määrittää kappaleen värin bittikartan avulla (jpg, png...) (Birn 2006).
index of refraction (IOR)	taitekerroin: määrittää materiaalin optisen tiheyden eli sen, miten paljon materiaali (valo-opissa väliaine) (Wikipedia 2016b).

low-poly model	low-poly -malli (Joensuu 2016), karkeasti mallinnettu 3D-kappale, jossa monikulmioverkko on harva ja malli on siksi kulmikas. Karkeaan malliin lisätään usein yksityiskohtia mm. normaalikarttojen avulla. Tällaisia malleja käytetään usein esimerkiksi peleissä, koska niiden käsittely on nopeampaa ja laskentatehon kannalta kevyempää kuin high-poly -malleilla (Birn 2006).
normal map	normaalikartta: määrittää kappaleen pintojen jokaiselle pisteelle normaalin eli suunnan, joka on pinnan pistettä vastaan kohtisuorassa. Voidaan käyttää lisäämään 3D-malliin yksityiskohtia, joita ei tehdä mallintamalla (Birn 2006).
PBR / physically based rendering	materiaalien fyysisiin ominaisuuksiin ja valo-oppiin perustuvat renderöintitekniikat, joiden mukaan määritellyt pintamateriaalit käyttäytyvät aitojen materiaalien tyyppisesti erilaisissa valaistusolosuhteissa (Russell 2016).
procedural textures	proseduraaliset tekstuurit: matemaattisiin algoritmeihin perustuvia tekstuureja joiden avulla voi peittää laajojakin alueita ilman toistuvia yksityiskohtia (Birn 2006).
shader	sävytin / varjostin: ohjelmarutiini joka määrittelee miten 3D-kappale renderöidään (Puhakka 2008; Joensuu 2016).
specular map	kiiltokartta: määrittää miten kappaleen pinta heijastaa valoa eri kohdissa (Birn 2006).
UV mapping	UV-kartoitus tai -kaavoitus: kappaleen purkaminen tai levittäminen kaksiulotteiseksi kaava-arkiksi, jonka avulla kuvatekstuurit kohdistetaan ja skaalataan käytettäväksi kappaleen pinnassa (Birn 2006; Joensuu 2016).

# 1 Johdanto

Opinnäytetyöni aiheena oli Karelia-ammattikorkeakoulun CHP-voimalan 3D-visualisointi interaktiivisena 3D-selainsovelluksena. Sovellus toteutettiin pääosin avoimen lähdekoodin sovelluksilla eli Blender 3D-ohjelmistolla sekä sen rinnalle erikseen kehitetyllä Blend4Web-renderöintityökalulla.

Koska kohteena oli todellinen toimiva voimalaitos ja visualisoinnin kohderyhmänä oli ensisijaisesti laitoksesta kiinnostuneet yrittäjät ja laajemmin muu yleisö, oli visualisoinnissa tarkoitus pyrkiä mahdollisimman realistiseen ulkoasuun. Siksi visualisointi toteutettiin siten, että erilaiset rakennemateriaalit erottuvat selkeästi toisistaan ja ovat helposti tunnistettavissa.

Työn keskeisiksi tutkimuskysymyksiksi valikoitui lopulta kaksi aihetta: mitä keinoja fotorealististen 3D-visualisointien tekemisessä yleensä käytetään ja miten näitä keinoja on mahdollisuus hyödyntää Blenderissä ja Blend4Webissä. Sen sijaan kaikki mallintamiseen, valaisemiseen, jälkikäsitteilyyn sekä käyttöliittymän suunnitteluun liittyvät asiat rajattiin työn ulkopuolelle.

Työn alussa tutkitaan kohteena olevan CHP-voimalan toimintaa ja rakennetta, mikä luo pohjan varsinaisen visualisoinnin toteuttamiselle. Koska visualisoinnin on tarkoitus olla mahdollisimman realistinen ja aidon oloinen, tuotantoprosessin ymmärtäminen pääpiirteissään auttaa varmistamaan, että kaikki olennaiset osat on mallinnettu ja toteutettu todellisuutta vastaavalla tavalla.

Työn painopisteenä oli kuitenkin tutkia sitä, miten realistisia fysikaalisiin ominaisuuksiin perustuvien PBR-materiaalien (Physically Based Rendering) teoriaa ja miten teoriaa sovelletaan käytännössä. PBR-materiaalien teoria pohjaa vahvasti siihen, miten valo käyttäytyy osuessaan eri materiaaleihin. Siksi myös niiden mallintaminen edellyttää teorian ymmärtämistä. Työssä tutkitaan tämän lisäksi myös erilaisten tekstuurien käyttämistä realistisen lopputuloksen saavuttamisessa.



Työn loppuosassa tarkastellaan vielä työvälineitä eli miten edellä mainittuja menetelmiä ja tekniikoita voidaan hyödyntää Blenderissä ja Blend4Webissä. Blenderin osalta tutustutaan siihen, miten PBR-materiaaleja määritellään uudemmassa PBR-valmiuksin varustetussa Cycles-renderöintimoottorissa sekä vanhemmassa, perinteistä tyyppiä olevassa Blender Internal -moottorissa. Jälkimmäinen on mukana siksi, että Blend4Webissä materiaalit määritellään pitkälti samalla tavalla. Koska Blend4Web tuottaa Blender-malleista web-grafiikkaa eli html5/WebGL-koodia, varsinkin web-grafiikan asettamia rajoituksia tutkitaan tarkemmin.



Kuva 1: Karelia-ammattikorkeakoulun opetus- ja tutkimuskäytössä oleva CHP-voimala tuottaa Sirkkalan Energiapuistossa osan koulukiinteistön sähkö- ja lämpöenergiasta (Kuva: Harri Pietiläinen).

## 2 CHP-laitoksen rakenne ja toiminta

### 2.1 Yleiskuvaus

Karelia-ammattikorkeakoululla on Volter 40 -omasähkölaitos (kuva 1), joka on konttirakenteinen sähkön ja lämmön yhteistuotantolaitos (CHP, Combined Heat

and Power). Laitosta käytetään uusiutuvan energian painoalan tutkimus- ja kehityspalveluiden tuottamiseen sekä energia- ja ympäristötekniikan opetukseen. Laitos sijaitsee Sirkkalan Energiapuistossa, mutta se on siirrettävissä myös muualle ja se kykenee toimimaan sekä valtakunnan verkkoon kytkettynä että tarvittaessa myös täysin itsenäisesti. Energiapuistossa laitos tuottaa Sirkkalan kampusrakenuksen kuluttamasta sähkö- ja lämpöenergiasta noin 30 prosenttia. (Kuittinen 2016; Volter Oy 2015a.)

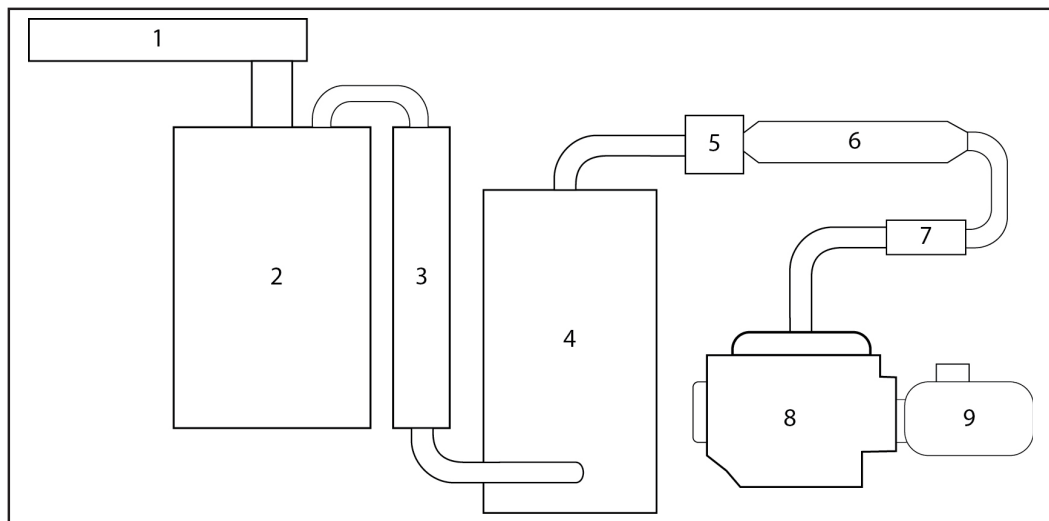
CHP-laitos toimii paikallisella puuhakkeella, josta se tuottaa puukaasugeneraattorin avulla sähköä sekä jäähdytyskierron sivutuotteena lämpöä. Tuotantoprosessin aikana puukaasusta, pakokaasuista sekä itse polttomoottorista otetaan jäähdytyskierron aikana lämmönvaihtimien avulla talteen lämpöenergiaa, joka voidaan edelleen johtaa muualle hyötykäyttöön, tai purkaa lauhdutinkkennojen kautta ulkoilmaan. (Volter Oy 2015b, 5.)

## **2.2 Sähköntuotanto**

CHP-laitos on rakennettu siirrettävissä olevaan konttiin samoin kuin sen yhteydessä oleva hakesiilo. Siilossa oleva puuhake siirretään ketju- ja ruuvikuljettimilla voimalaan (kuva 2), missä pintavahti (mikroaaltoanturi) huolehtii siitä, että haketta on poltossa kaasuttimessa aina oikea määrä (Kuittinen 2016). Kaasuttimessa haketta kaasutetaan noin 1 000 °C:n lämpötilassa, mikä tuottaa palavia kaasuja eli häkää (CO), vetyä (H<sub>2</sub>) sekä metaania (CH<sub>4</sub>) (Volter Oy 2015b, 5).

Kaasuttimessa syntynyt puukaasu ohjataan seuraavaksi ensiölämmönvaihtimeen, joka ottaa kaasusta talteen lämpöenergiaa. Jäähdytetty puukaasu johdetaan seuraavaksi suodattimeen, joka poistaa kaasusta kiinteää ainesta, kuten muun muassa tuhkaa. Kaasuttimen ja suodattimen alla kulkeva tuhkaruuvi siirtää kerätyn tuhkan ja hiilen pois laitoksesta. Suodattimen jälkeen puukaasu jatkaa

matkaansa ensin toisiolämmönvaihtimeen ja edelleen toiseen suodattimeen, joka toimii varmistuksena pääsuodattimelle. (Kuittinen 2016).



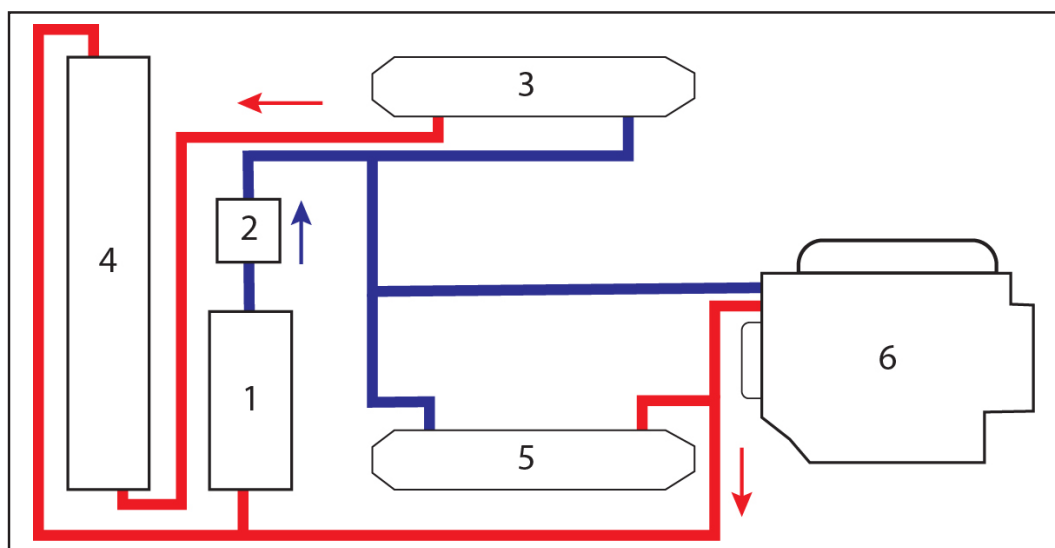
Kuva 2: CHP-voimalan sähköntuotantoprosessin vaiheet. 1. Ruuvikuljetin 2. Kaasutin 3. Ensiolämmönvaihdin 4. Suodatin 5. Varmistussuodatin 6. Toisiolämmönvaihdin 7. Ilmansekoitusventtiili 8. Polttomoottori 9. Sähkögeneraattori (Kuva: Harri Pietiläinen).

Puhdistettuun ja jäädytettyyn puukaasuun lisätään ilmansekoitusventtiiliin kautta ilmaa ennen kuin kaasuseos johdetaan polttomoottoriin. Moottori on Agco Sisun valmistama dieselmoottori, joka on muunnettu kaasumoottoriksi muun muassa lisäämällä siihen sytytystulpat. Moottori käyttää siis kaasua polttoaineenaan ja pyörittää samalla siihen kytkettyä sähkögeneraattoria, joka tuottaa tuotantoprosessin lopputuloksena sähköenergiaa. Generaattorin tuottama sähkö siirretään akuston ja verkkoyksikön kautta edelleen joko sitä hyödyntävään kiinteistöön tai laitoksen ylläpitämään saarekeverkkoon. (Kuittinen 2016; Volter 2015a, 3.)

### 2.3 Lämmöntuotanto

CHP-laitos tuottaa lämpöä sähköntuotannon sivutuotteena. Lämpöenergia otetaan prosessin aikana talteen useassa eri vaiheessa, ja sitä voidaan hyödyntää edelleen laitokseen naapurissa olevan kiinteistön lämmityksessä. Saarekekäytössä, jolloin lämpöenergiaa ei tarvita, laitoksen tuottama ylimääräinen lämpö ohjataan lauhdutinkennoston kautta ulkoilmaan. (Kuittinen 2016.)

Laitoksen jäähdytyskierto on suljettu eli sama glykoli-vesiseos kiertää laitoksen eri osissa keräten talteen prosessin tuottamaa lämpöä (kuva 3). Jäähdytysneste ohjataan kolmeen eri kohteeseen, joista ensimmäisenä on ensiölämmönvaihdin heti kaasuttimen jälkeen. Ensiölämmönvaihdin on rakenteeltaan ruuvikierre, jossa puukaasu kiertää luovuttaen samalla lämpöä kierrettä ympäröivään jäähdytysnestevaippaan. Suurin osa lämpöenergiasta saadaan talteen jo tässä vaiheessa. Kaasusuodattimen jälkeen puukaasusta kerätään lämpöä vielä toistamiseen toisiolämmönvaihtimessa. (Kuittinen 2016.)



Kuva 3: CHP-voimalan lämmöntuotantoprosessin vaiheet. 1. Päälämmönvaihdin 2. Kiertovesipumppu 3. Toisiolämmönvaihdin 4. Ensiölämmönvaihdin 5. Pakokaasulämmönvaihdin 6. Polttomoottori (Kuva: Harri Pietiläinen).

Toinen osa jäähdytysnesteestä ohjataan polttomoottoriin, josta kerätään talteen itse moottorin palamisreaktion tuottamaa lämpöenergiaa. Kolmas osa jäähdytysnesteestä ottaa talteen lämpöä vielä polttomoottorin pakokaasuista. Nämä kolme eri haaraa yhdistetään jälleen lämmönvaihtimien jälkeen yhdeksi ja kuuma jäähdytysneste ohjataan edelleen levylämmönvaihtimeen. Siinä jäähdytysnesteen lämpöenergiaa siirretään kiinteistön kiertoveteen kennostossa, joka erottaa nämä kaksi nestekiertoa toisistaan. (Kuittinen 2016.)

Mikäli lämpöenergiaa ei hyödynnetä kiinteistön lämmittämiseen, tai jos jäähdytysnesteen lämpötila ei ole muutoin laskenut riittävästi levylämmönvaihtimessa,

jäähdytysneste ohjataan edelleen laitoskontin ulkoseinään asennettuun lauhdutinkennostoon. Siellä jäähdytysnesteestä puretaan pois ylimääräistä lämpöä, minkä jälkeen jäähdytetty jäähdytysneste ohjataan uudelleen lämmönvaihtimille. (Kuittinen 2016.)

## **2.4 Visualisointi**

CHP-laitoksesta on tehty tämän opinnäytetyön puitteissa selainpohjainen 3D-visualisointi, jonka avulla voidaan tutustua laitoksen rakenteeseen ja toimintaan. Mallinnuksen pohjana käytettiin olemassa olevia CAD-malleja, joita jouduttiin kuitenkin muokkaamaan ja täydentämään suurelta osin.

Tässä vaiheessa visualisointi on kuitenkin vielä melko yksinkertainen ja mahdollistaa lähinnä mallin tarkastelun eri puolilta, läheltä ja kauempaa. Laitos sisältää monenlaisia pinta- ja rakennusmateriaaleja, jotka on pyritty mallintamaan mahdollisimman pitkälti todellisuutta vastaaviksi. Pintamateriaaleja tarkastellaan työn lopussa.

## **3 Fotorealistinen visualisointi**

### **3.1 Määritelmä**

Yksi 3D-grafiikan viime vuosien tärkeimmistä suuntauksista on ollut fotorealismi, jossa 3D-malleista pyritään tekemään niin yksityiskohtaisia ja realistisia, että niitä on käytännössä lähes mahdotonta erottaa valokuvista. Fotorealistista 3D-grafiikkaa ja -efektejä käytetään nykyisin hyvin laajalti muun muassa tietokonepeleissä, elokuvissa ja tv-sarjoissa, mainoksissa, animaatioissa sekä erilaisissa visuaali-

soinneissa. Esimerkiksi huonekalujätti Ikean vuosittaisissa kuvastojen tuotekuvista jopa 75 % on nykyisin 3D-grafiikkaa. (Parkin 2014; Price 2015.)

Jotta kuvasta saataisiin mahdollisimman realistisen ja aidon tuntuinen, monen asian tulee loksahda kohdalleen jo mallinnusvaiheessa. Yleensä renderöidyt kuvat vaativat vielä monenlaista jälkikäsittelyä, jota renderöinnin aikana ei ole mahdollista tai järkevää tehdä. Esimerkiksi pitkään 3D-mallinnuksen parissa työskennellyt ja BlenderGuru-sivuston perustanut Price on todennut, että foto-realistisuuteen vaaditaan neljän elementin täydellistä toteuttamista, jotka ovat mallintaminen, valaistus, materiaalit sekä renderöinti (Price 2012).

### **3.2 3D-mallintamisesta lyhyesti**

Monet 3D-alan ammattilaiset korostavat referenssikuvien tutkimisen tärkeyttä jo ennen varsinaisen mallintamisen aloittamista. Price mainitsee, että kuvia tutkimalla voi kerätä tietoa paitsi mallinnettavien kohteiden muodoista, rakenteesta ja mittasuhteista, myös niiden väreistä, materiaaleista ja pinnanmuodoista. Myös oikeanlaiset valaistusolosuhteet sekä mahdolliset kameraoptiikan aiheuttamat virheet ja vääristymät lisäävät realistisuutta. (Price 2012.)

Arkkitehtuurivisualisoineistaan ja erityisesti lyhytelokuvastaan *The Third and the Seventh* tunnettu Alex Roman julkaisi vuonna 2013 kirjan nimeltään *From Bits to the Lens*. Kirja sisältää runsaasti laadukkaita visualisointikuvia sekä suuren määrän tietoa 3D-grafiikasta, aina esituotannosta ja mallintamisesta jälkikäsittelyyn asti. Renderöintiin liittyvä osuus käsittelee Romanin käyttämää kaupallista V-Ray-renderöintimoottoria, ja siksi siitä ei ole varsinaisesti hyötyä omassa projektissani. Kirjasta on tehty vain pieni painos eikä sitä ole saatavana e-kirjana ja painettunakin ainoastaan suoraan tekijältä, joten päätin tältä osin turvautua tässä työssä toisen käden tietoon. (Roman 2016.)

Price on poiminut podcastiinsa Romanin kirjasta muutamia olennaisia vinkkejä, joita voi hyödyntää yleisemminkin 3D-mallinnuksessa. Ensimmäinen Pricen poimima neuvo on muistaa malleissa myös sellaiset yksityiskohdat, joihin ei tavallisesti tule kiinnittäneeksi edes huomiota, mutta joiden puuttuminen saa mallin vaikuttamaan oudolta tai puutteelliselta. Price mainitsee esimerkkinä oven, jonka harjatusta metallista tehdyssä kahvassa on koristeuurteita ja jonka saranoiden maalipinta on hieman rapistunut kulumisen takia. (Price 2014.)

Toinen Pricen esiin nostama huomio on se, että maailma on lopulta melko kaootinen paikka ja sen vuoksi myös 3D-malleihin on syytä lisätä epäjärjestyä ja poikkeamia. Esimerkkinä hän käyttää kirjahyllyä, jonka kirjat ovat hieman erikoisia ja -paksuisia, eivätkä ne kaiken lisäksi ole välttämättä edes kovin hyvässä järjestyksessä. Tällainen epäjärjestys on normaalielämässä varsin tavanomaista ja siksi epäjärjestyksen täydellinen puuttuminen saa asetelman tuntumaan teennäiseltä. (Price 2014.)

Pixar-animaatiostudiolla pitkään työskennellyt Jeremy Birn on perehtynyt erityisesti 3D-grafiikkaan liittyviin valaistusasetuksiin. Hän mainitsee pari seikkaa, joilla 3D-kappaleisiin saa pienillä mallintamiskikoilla lisättyä realistista tuntua. Ensimmäiseksi hän kehottaa lähes järjestelmällisesti pyöristämään mallien kaikki kulmat ja reunat. Pyöristämisen idea on siinä, että täysin teräviä kulmia tai reunoja ei käytännössä ole normaalisti olemassa juuri muualla, kun veitsenterissä tai muissa vastaavissa kappaleissa, sillä ne on aina pyöristetty vaaratilanteiden ehkäisemiseksi ja käytön helpottamiseksi. Niinpä myös 3D-malleissa reunoja tulisi aina hieman pyöristää. Niillä on myös valaistuksen kannalta olennainen merkitys, sillä valo korostaa kappaleiden reunamuotoja osuessaan niihin. (Birn 2013.)

Oletusarvoisesti 3D-mallinnusohjelmat piirtävät tasot tai kappaleiden pinnat äärimmäisen ohuina, jolloin valoa voi päästä pilkistämään odottamattomista kohdista. Birn suosittelee, että esimerkiksi talojen seinät mallinnettaisiin realistisen paksuisina, mikä auttaa vähentämään tällaisia vuotoja (Birn 2013).

### 3.3 Fotorealistiset materiaalit

Fyysisessä maailmassa eri materiaalit näyttävät erilaisilta, olipa kyse sitten luonnollisista tai ihmisen tuottamista keinotekoisista materiaaleista. Jotkut materiaalit heijastavat valoa paremmin kuin toiset, ja jotkut taas hajottavat valonsäteitä tai päästävät ne kulkemaan lävitseen. Myös erilaiset pinnanmuodot näkyvät usein vasta sitten, kun valo osuu materiaalin pintaan ja saa aikaan varjoja. Kaikki nämä ilmiöt johtuvat materiaalien fysikaalisista ominaisuuksista ja siitä, miten valo käyttäytyy osuessaan kappaleisiin.

Jotta nämä asiat osaisi ottaa huomioon, materiaalin ja valaistuksen tutkimiseen kannattaa todella panostaa. Andrew Price lainaa podcastissaan Alex Romania, jonka mukaan jopa 90 % fotorealismista saadaan aikaan oikeanlaisilla materiaaleilla ja valaistuksella (Price 2015). Yhtenä ongelmana tosin on se, että jokaisella renderöintimoottorilla eli ohjelmalla, joka tuottaa 3D-malleista kaksiulotteisia kuvia, on omat toteutustapansa materiaaleille ja valaistukselle. Sen vuoksi fyysisen maailman materiaalien lisäksi on pakko tuntea erittäin tarkasti myös se, miten kussakin projektissa käytettävä renderöintimoottori tuottaa vastaavia materiaaleja mallinnuksen tarpeisiin. Tätä korostaa myös Pricen referoima Roman (Price 2015).

Kun 3D-grafiikalla halutaan tuottaa fotorealistisia, valokuvaan verrattavissa olevia kuvia, turvaudutaan usein fysikaalisiin ominaisuuksiin perustuvaan renderöintiin (physically based rendering, PBR). Russell toteaa, että PBR:n määritelmä ei ole täysin vakiintunut, mutta yleensä sillä tarkoitetaan matemaattisia malleja, jotka pyrkivät kuvaamaan valon käyttäytymistä eri materiaaleissa mahdollisimman tarkasti. Mallien avulla voidaan ottaa huomioon paitsi materiaalien ominaisuudet (valon sironta / diffuusio, heijastavuus, pinnan karkeus, jne.), myös valaistusolosuhteet (suora valo, heijastunut valo, ympäristövalaistus, valon väri) ja niiden vaikutus kappaleeseen. (Russell 2016; Wilson 2016a.)



PBR ei kuitenkaan automaattisesta tarkoita sitä, että renderöinnistä tulisi (foto-)realistinen, vaan että valon kulku ja käyttäytyminen pystytään mallintamaan aiempaa tarkemmin. Aihetta opinnäytetyössään tutkinut Joonas Sairiala Tampereen ammattikorkeakoulusta toteaa, että PBR-malleja hyödyntävät renderöintimoottorit kykenevät tarjoamaan hyvin laajan kirjon erilaisia vaihtoehtoja myös muunlaiselle taiteelliselle ilmaisulle kuin fotorealismille. Esimerkkinä tästä Sairiala mainitsee, että muun muassa Pixar-elokuvastudio käyttää omassa animaatioelokuviinsa PBR-työnkulkua, mikä mahdollistaa paljon realistisemmän ja yksityiskohtaisemman lopputuloksen. (Sairiala 2015, 13.)

### **3.4 Valon fysiikasta ja materiaalien ominaisuuksista**

#### **3.4.1 Valo ilmiönä**

Todellisilla materiaaleilla on erilaisia fyysisestä rakenteesta riippuvia ominaisuuksia, jotka vaikuttavat siihen, millaisina materiaalit nähdään ja miten valo käyttäytyy niihin osuessaan. Valon fysiikka on itse asiassa matemaattisina malleina ilmaistuna melko monimutkaista. Vaikka nämä matemaattiset mallit ole sinänsä olennaisia tämän työn kannalta, niiden kuvaamat ilmiöt luovat perustan materiaalien mallintamiselle 3D-ympäristöissä.

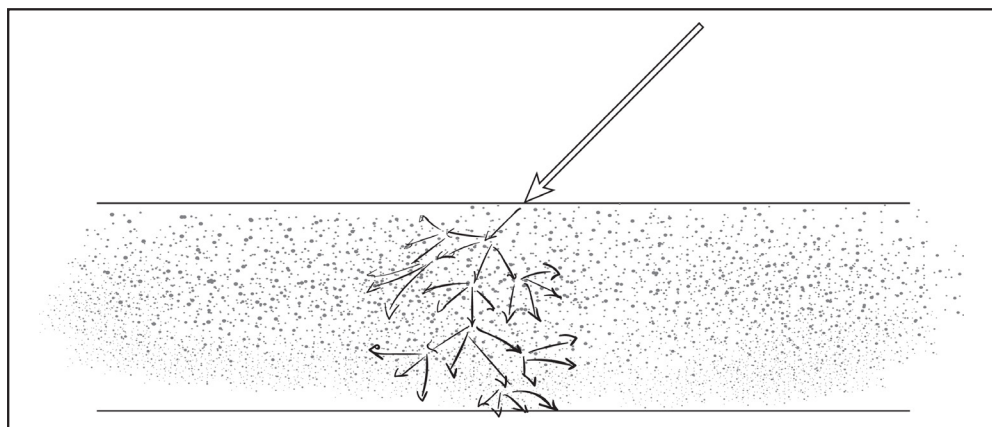
Näkyvä valkoinen valo koostuu monista eri aallonpituuksista, jotka yhdessä näyttävät valkoiselta. Vastaavasti värillisessä valossa on mukana vain tiettyjä aallonpituuksia. Jos valkoinen valo ohjataan esimerkiksi prismaan, eri aallonpituudet taittuvat eri tavalla ja heijastuksessa näkyy koko näkyvän valon spektri. Sama ilmiö näkyy sateenkaressa eri aallonpituuksien taittuessa vesipisaroista eri tavalla. Eri aallonpituudet käyttäytyvät usein eri tavoin myös osuessaan kappaleisiin.

Yksinkertaistaen voidaan ajatella, että osuessaan aineeseen tai kappaleeseen, valo voi joko heijastua, siroutua, imeytyä, taittua tai kulkea kappaleen tai aineen läpi. Useimmiten todellisuus on kuitenkin jonkinlainen yhdistelmä edellä mai-

nituisista, sillä kunkin materiaalin fyysinen rakenne vaikuttaa siihen, mitä valolle tapahtuu. Kappaleen materiaali on siis tunnettava, jos se halutaan renderöidä mahdollisimman realistisesti.

### 3.4.2 Läpinäkyvyys ja läpikuultavuus

Helpoin ja suoraviivaisin tapaus on läpinäkyvä ja kirkas homogeeninen väliaine tai materiaali, jonka läpi valo pääsee kulkemaan esteettä. Hoffman käyttää esimerkkinä puhdasta vettä, jonka valo läpäisee kulkusuunnan muuttamatta. Puhtaaseen veteen valo ei juurikaan imeydy eli absorboidu ainakaan lyhyellä matkalla, joten myös valon intensiteetti (energia) säilyy kutakuinkin samana. Pidemmällä matkalla imeytyminen näkyy kuitenkin selvemmin esimerkiksi uima-altaassa tai merivedessä (tosin kummassakin ainakin osittain jo pelkästään niissä olevien epäpuhtauksien ja pienhiukkasten vaikutuksesta), joissa valon intensiteetti heikkenee matkan kasvaessa. Kirkas neste tai aine, kuten esimerkiksi kupillinen teetä, voi myös muuttaa matkalla valon väriä, mikäli aine absorboi vain tiettyjä valon aallonpituuksia. (Hoffman 2013.)



Kuva 4: Läpikuultavissa materiaaleissa valo siroaa eri suuntiin osuessaan aineessa oleviin hiukkasiin tai molekyyliin ja samalla sen nopeus hidastuu. Mitä enemmän törmäyksiä tapahtuu, sitä laajemmalle valo hajaantuu (Kuva: Harri Pietiläinen).

Heterogeeniset väliaineet sisältävät puolestaan erilaisia hiukkasia, joihin törmätessään valo hajaantuu eli siroaa (scattering) eri suuntiin (kuva 4). Mitä useam-

piin hiukkasiin valo osuu, sitä laajemmalle valo siroaa ja sitä läpinäkymättömämpi aine on (Hoffman 2013). Sama ilmiö koskee niin nesteitä kuin kiinteitäkin materiaaleja, ja se selittää myös sen, miksi sama materiaali voi olla ohuena läpinäkyvä tai läpikuultava, mutta paksumpana täysin läpinäkymätön. Mitä pidemmän matkan valo kulkee materiaalin sisällä, sitä varmemmin se törmää hiukkasiin yhä uudestaan ja lopulta siroaa kaoottisesti eri suuntiin eikä enää läpäise materiaalia lainkaan.

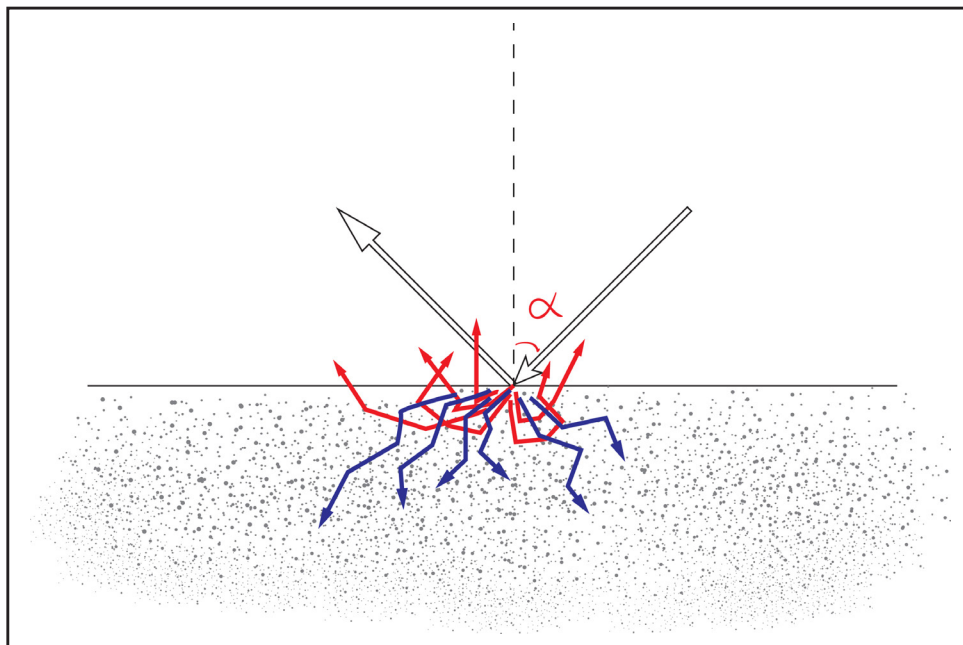
### 3.4.3 Kiinteät materiaalit

Kiinteät, läpinäkymättömät materiaalit voidaan karkeasti ottaen jakaa kahteen ryhmään: johteisiin (enimmäkseen metalleja) ja eristeisiin (ei-metallit). Metallit heijastavat jopa 60–90 % niihin osuvasta valosta, kun eristeissä heijastavuus on paljon pienempi (Russell 2016). Metalleissa on eristeistä poiketen myös vapaita elektroneja, jotka absorboivat välittömästi sen osan valosta, joka tunkeutuu materiaalin sisään, eikä valoa pääse siroutumaan takaisin ulospäin (Hoffman 2013). Juuri nämä samat vapait elektronit saavat myös metallin johtamaan sähköä. Metallilla tarkoitetaan tässä puhdasta, pinnoittamatonta metallia. Jos metalli on esimerkiksi maalattu, maalipinta käyttäytyy eristeen tavoin, ja sama koskee myös pinnassa olevia epäpuhtauksia (Price 2016b).

Eristeissä puolestaan materiaalin pintakerrokseen tunkeutuneesta valosta vain osa absorboituu (usein vain tietyt valon aallonpituudet) ja loppuosa valosta siroaa takaisin kappaleesta satunnaisesti kaikkiin suuntiin (kuva 5). Jos vain tietyt aallonpituudet siroavat ympäristöön, materiaali saa sille ominaisen perusvärin. Punainen muovi näyttää siis punaiselta siksi, että kyseinen muovilaatu absorboi itseensä kaikki muut valon aallonpituudet. (Hoffman 2013.)

Materiaali voi myös absorboida tasaisesti kaikkia aallonpituuksia, jolloin kappaleen väri määräytyy harmaasävyasteikolla. Mikäli kappale absorboi kaiken valon,

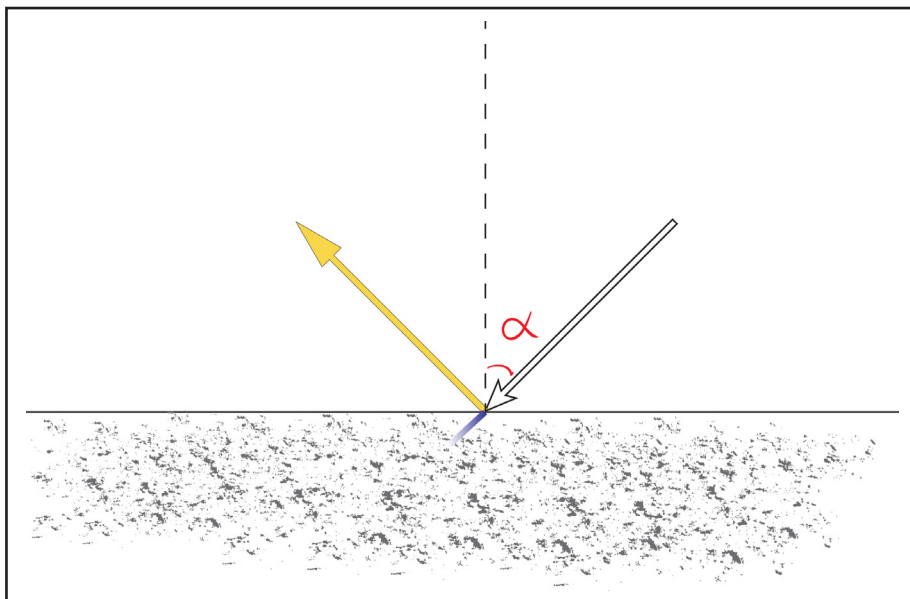
kappale näyttää mustalta ja vastaavasti kaiken valon takaisin heijastava kappale näyttää valkoiselta. Käytännössä täydellisen heijastamattomat pinnat ovat erittäin harvinaisia. (Russell 2016.)



Kuva 5: Eristeissä osa valosta heijastuu suoraan kappaleen pinnasta (valkoinen nuoli), osa siroutuu kappaleen pintakerroksista takaisin (punaiset nuolet) ja osa imeytyy kappaleeseen (siniset nuolet) (Kuva: Harri Pietiläinen).

Kiilto-ominaisuus johtuu siitä, että osa valonsäteistä heijastuu materiaalin pinnasta suoraan eteenpäin, metalleissa siis huomattavasti voimakkaammin kuin eristeissä. Eristeisiin pätee sääntö, että mitä kiiltävämpi kappale on, sitä vähemmän kappaleen oma perusväri näkyy. Valon heijastuminen ja siroutuminen ovat toisensa poissulkevia ilmiöitä. (Russell 2016.)

Koska metallit heijastavat suurimman osan niihin osuvasta valosta ja loput näkyvästä valosta absorboituu pintakerroksissa, useimmat metallit näyttävät paitsi kiiltäviltä, myös melko värittömiltä. Poikkeuksen tekevät esimerkiksi kulta ja kupari sekä metalliseokset, joissa edellä mainittuja on käytetty. Niistä heijastunut valo on värillistä, koska kulta ja kupari absorboivat kemiallisen rakenteensa takia tiettyjä näkyvän valon aallonpituuksia enemmän kuin toisia. Esimerkiksi kulta absorboi eniten sinistä valoa, ja siksi kulta näyttää valkoisessa valossa keltaiselta, joka on sinisen vastaväri (kuva 6). (Wikipedia 2016a.)



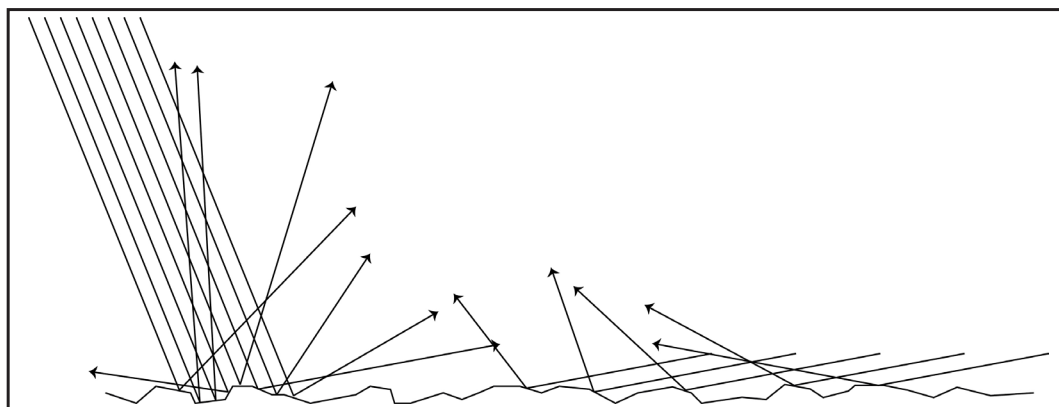
Kuva 6: Metalleissa ei valon siroutumista tapahdu lainkaan. Osa valosta imeytyy kappaleeseen ja valtaosa heijastuu suoraan eteenpäin. Esimerkiksi kullan kohdalla vain siniset valoaalot imeytyvät kappaleeseen, mikä antaa kullalle sen keltaisen sävyn (Kuva: Harri Pietiläinen).

#### 3.4.4 Pinnan mikrotason rakenne

Myös materiaalin mikrotason pintarakenne vaikuttaa sen kiilto-ominaisuuksiin. Peilin kaltaiseen täysin sileään pintaan osuessaan valo heijastuu eteenpäin pinnan normaaliin nähden täsmälleen samassa kulmassa kuin se osui kappaleeseen. Todellisuudessa lähes kaikissa pinnoissa on mikroskooppista epätasaisuutta tai karkeutta, mikä saa heijastuvan valon muuttamaan hieman suuntaansa. Mitä karkeampi pinta on, sitä enemmän valo hajaantuu heijastuessaan ja sitä sumeammalta ja epätarkemmalta heijastus näyttää. (Russell 2016.)

Yksi heijastuksiin vaikuttava tekijä on niin sanottu Fresnel-ilmiö, joka johtuu edellä mainituista mikroskooppisista epätasaisuuksista. Kun valonsäde osuu lähes kohtisuoraan pinnan mikroviisteeseen, valosta suurin osa imeytyy materiaaliin tai siroutuu ja pienempi osa heijastuu eri suuntiin (kuva 7). Mitä loivemmassa kulmassa valo osuu mikroviisteeseen, sitä suurempi osa siitä heijastuu eteenpäin ja sitä vähemmän valosta imeytyy materiaaliin tai siroutuu. Yksinkertaisimmillaan ilmiön voi havaita esimerkiksi tynnen veden äärellä, jossa rantavedessä näkee

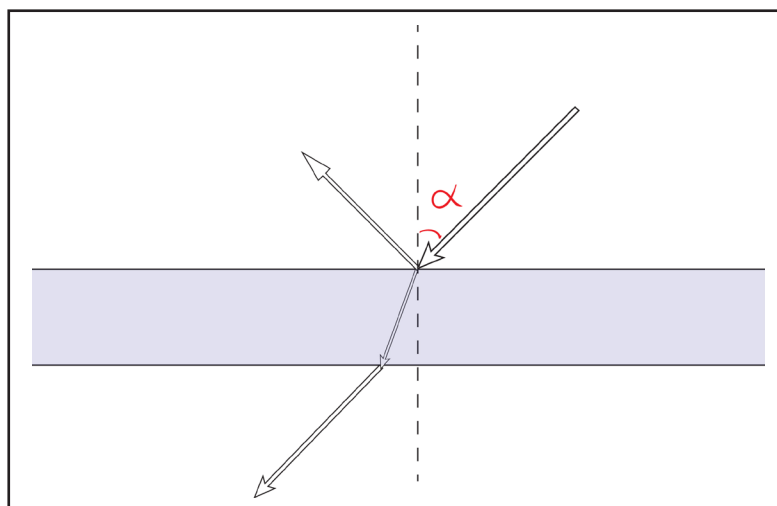
helposti pohjassa olevat kivet, mutta kauempana vedenpinta muistuttaa jo peiliä. (Russell 2016.)



Kuva 7: Fresnel-ilmiö selittyy materiaalien mikrotason epätasaisuuksilla. Mitä loivempi valon tulokulma on pintaan nähden, sitä todennäköisemmin valo osuu vain huippukohtiin ja heijastuu eteenpäin (Kuva: Harri Pietiläinen).

### 3.4.5 Valon taittuminen

Kun valo siirtyy väliaineesta toiseen, se yleensä taittuu eli muuttaa kulkusuuntaansa jonkin verran. Tämän voi helposti todeta vaikka laittamalla lusikan vaajaaseen vesilasiin: tietyssä asennossa vedessä oleva osa on eri linjassa vedenpinnan yläpuolella osan kanssa. Kullakin aineella tai materiaalilla on taitekerroin (index of refraction, IOR), joka kertoo sen optisen tiheyden: mitä suurempi taitekerroin, sitä tiheämpää materiaali on optisesti ja sitä enemmän se paitsi taittaa myös hidastaa valon kulkua (kuva 8). (Wikipedia 2016b.)



Kuva 8: Kun valo kulkee lasin läpi, se taittuu kummassakin ilman ja lasin välisessä rajapinnassa (Kuva: Harri Pietiläinen).

Kahden väliaineen rajapinnassa tapahtuva taittuminen riippuu kyseisten aineiden tai materiaalien taitekerrointen eroista. Ilman taitekerroin on noin 1, lasin noin 1.5 ja vedellä 1.33. Koska edellisessä esimerkissä jokaisella aineella (ilma, lasi, vesi) on erilainen taitekerroin, valo taittuu itse asiassa jokaisessa kahden aineen rajapinnassa (ilma->lasi->vesi->lasi->ilma). (Wikipedia 2016b.)

### 3.4.6 Kaustiikka

Yksi kiinnostava valon aikaansaama piirre on niin sanottu kaustiikka. Termillä tarkoitetaan Birnin mukaan ”epäsuoraa valoa, joka ei hajaannu epämääräisesti eri suuntiin vaan säilyy tarkkaan rajattuna”. Birn mainitsee esimerkkeinä valon taittumisen linseissä ja prismoissa ja heijastukset peileistä, vedenpinnasta tai muista kiiltävistä esineistä. Kaustiikaksi lasketaan myös tavallisten lasiesineiden läpi kulkeneen valon muodostamat valokuviot, sekä vastaavat valoilmiot esimerkiksi uima-altaan tai järven pohjassa. (Birn 2006, 116–117.)

Birn toteaa, että kaustiikka edellyttää yleensä varsin kirkasta valaistusta ja suu-  
rehkoa kontrastia valon ja hämärän välillä. Siksi heijastuksia syntyy useimmiten auringonvalon osuessa suoraan johonkin kiiltävään esineeseen tai vaikkapa peiliin tai lasimaljakkoon. Koska tällaiset valoilmiot ovat hyvin silmään pistäviä ja useimmille meistä siksi tuttuja, niiden lisääminen renderöintiin saa tilan tuntu-  
maan aidommalta ja realistisemmalta. (Birn 2006, 116–118.)

Kaustiikkaa hyödynnetään myös tietoisesti valaistuksessa, mistä Birn mainitsee esimerkkeinä salamavalot ja autojen ajovalot. Näiden lisäksi monissa moderneissa valaisimissa käytetään erilaisia tekniikoita erikoisten valaistuskuvioiden aikaansaamiseksi. Birnin mielestä tällaiset valokuviot kannattaa kuitenkin toteuttaa 3D-malleissa yleensä helpommin ja nopeammin valoihin liitettävien tekstuurien avulla, kuten niin sanotuissa IES-valoissa (Illuminating Engineering Society) on tehty. (Birn 2006, 116–118.)

### 3.5 Materiaalien mallintaminen

#### 3.5.1 PBR-materiaalien perusteita

PRB-materiaalien mallintaminen riippuu täysin käytettävästä renderöintimoottorista. Kaikki moottorit eivät kykene tai ole edes tarkoitettuja tuottamaan realistisia renderöintejä ja nekin jotka ovat, voivat poiketa toteutustavoiltaan hyvin paljon toisistaan. Eroja löytyy paitsi käytetyssä terminologiassa, myös siinä miten renderöintiasetukset valitaan ja miten itse moottori tuottaa kuvan annettujen parametrien ja toteutetun 3D-mallin perusteella. Moottoreilla voi olla myös erilaisia painotuksia ja piirteitä, joita toisista moottoreista ei löydy lainkaan. Kun käytettävän moottorin tarjoamat mahdollisuudet ja erilaiset asetukset tuntee riittävän hyvin, voi renderöinnistä saada halutun laisen.

Itse materiaalien määrittelyyn on useita eri lähestymistapoja, ja siihen vaikuttaa renderöintimoottorin lisäksi myös se, missä tuotoksia on tarkoitus käyttää. Esimerkiksi Sairiala on omassa työssään rakentanut PRB-sävyttimen Wilsonin esittelemän työnkulun mukaan. Tällaista työkulkua käytetään muun muassa pelimaailmassa, missä mallit pyritään pitämään mahdollisimman yksinkertaisina ja realismia tavoitellaan erilaisten tekstuurien ja karttojen avulla. Siksi asioita tarkastellaan hyvin eri lähtökohdista kuin vaikkapa animaatioelokuviissa tai arkkitehtuurivisualisoinneissa. (Wilson 2016b.)

Wilsonin mukaan PBR tarkoittaa pohjimmiltaan sitä, että hienostuneiden sävyttimien (shaders) ja tekstuurien avulla saadaan aikaan todellisuutta vastaava materiaali, joka noudattaa valon ja materian fysiikkaa ja näyttää siksi hyvältä kaikissa valaistusolosuhteissa (Wilson 2016b). Edellä mainitut sävyttimet ovat renderöintimoottoreissa niitä keskeisiä ohjelmarutiineja, jotka annettujen parametrien perusteella määrittelevät kappaleen ulkoasun. Esimerkiksi Blenderin Cycles-renderöintimoottorissa omat sävyttimet voidaan määritellä muun muassa diffuusivärille, kiiltävyydelle, läpinäkyvyydelle, lasille ja valaisulle (Blender Foundation 2016: Cy-



cles Render Engine / Shader Nodes). Cyclesissa sävyttimiä voi yhdistellä hyvin vapaasti ja monipuolisesti toisiinsa, joten niillä voidaan tuottaa lukematon määrä erilaisia materiaaleja.

### 3.5.2 Diffuusiväri

Yksi tapa aloittaa materiaalin koostaminen on määrittää kappaleelle perus- eli diffuusiväri, joka on pelkistetyimmillään koko kappaleen peittävä tasainen värisävy. Sitä voi muokata mielenkiintoisemmaksi esimerkiksi käyttämällä diffuusivärinä bittikarttaa, joka voi olla valokuva, skannattu kuva tai piirrosohjelmalla (Photoshop, Gimp, tms.) digitaalisesti maalattu kuvatekstuuri. Entistä mielenkiintoisempia tekstuureja saadaan aikaan, kun edellä mainittuja yhdistellään toisiinsa. Birn sanoo monien ammattilaisten käyttävän valokuvia ja skannattuja tekstuureja osana maalattuja tekstuureja, koska ne lisäävät tekstuurien realismia (Birn 2013). Tätä kirjoitettaessa skannattujen tekstuurien käyttö on kovaa vauhtia yleistymässä, kun tekninen kehitys on tehnyt 3D-skannauksesta helpompaa ja edullisempää.

Bittikarttoja voi käyttää joko sellaisenaan tai yhdistämällä niitä toisten bittikarttojen ja/tai diffuusivärien kanssa renderöinnin aikana. Bittikartoilla pintaan voidaan lisätä yksityiskohtia, joita ei kannata tehdä kappaleeseen mallintamalla. Käytetyt bittikartat ovat yleensä jpeg- tai png-tyyppisiä, ja näistä viimeksi mainittua käytetään erityisesti silloin, kun kuva on osittain läpinäkyvä.

Tavallisten valokuvista tehtyjen diffuusivärikarttojen ohella värien määrittelyssä käytetään myös niin sanottuja albedokarttoja, joissa kirkkauseroja on tasoitettu poistamalla tekstuurista suuntavalojen vaikutukset eli varjot ja valoheijastukset. Wilsonin mukaan perinteiset diffuusikartat voivat näyttää tietyissä valaistusolosuhteissa vääränlaisilta (esimerkiksi varjot voivat olla väärässä paikassa), koska renderöinnissä valoisat kohdat ja varjot tuotetaan valaistuksella. Siksi albedokartta voi olla perinteistä diffuusikarttaa parempi vaihtoehto. (Wilson 2016a.)

### 3.5.3 Läpinäkyvät tekstuurit

Läpinäkyviä tekstuureja käytetään usein silloin, kun halutaan yhdistellä tekstuureja toisiinsa. Tästä hyvä esimerkki on erilaiset yksityiskohtat, kuten valumajäljet, tahrat, naarmut tai vaikkapa graffitit seinissä tai laitteiden kyljissä. Näistä yksityiskohtia lisäävistä bittikartoista käytetään englanninkielistä termiä decals, joka tarkoittaa tarraa tai siirtokuvaa. Se kuvaakin aika hyvin niiden käyttötarkoitusta.

Kun näitä "siirtokuvia" lisätään malleihin Birnin mukaan kannattaa kysyä, miksi kuluma tai valumajälki on juuri tietyssä kohdassa. Esimerkiksi matot kuluvat siellä missä yleisimmin kuljetaan, naarmuja syntyy sinne missä ovi avautuessaan hankaavat lattiaan ja valumajälkiä näkyy siellä minne vesi luonnostaan tippuu tai valuu. Tällaisia yksityiskohtia ei siis tule lisätä materiaaliin sattumanvaraisesti, vaan ainoastaan tarkkaan harkittuihin paikkoihin. (Birn 2013.)

Läpinäkyvyyskartoille on tarvetta myös silloin, kun halutaan tehdä itse mallista osittain läpinäkyvä. Tällaisia ovat muun muassa ruohonkorret ja puiden lehdet. Jos kokonainen puu rakennetaan 3D-mallina, sen renderöiminen on hyvin raskasta lehtien suuren määrän takia. Mallinnettujen lehtien sijaan voidaan käyttää yksinkertaisia suorakaidepintoja, joille määritellään läpinäkyvyysasetukset sisältävä lehtitekstuuri. Läpinäkyvyystieto voi olla itse tekstuuritiedostossa mukana alpha-kanavalla (png) tai erillisessä mustavalkoisessa maskitiedostossa, jonka avulla varsinainen lehtitekstuuri erotetaan suorakaiteen muotoisesta kuvasta. Samaa tekniikkaa voi käyttää myös muunlaisissa kappaleissa, joista halutaan osaksi läpinäkyviä. Birn tosin huomauttaa, että parhaiten tämä tekniikka toimii taustalla olevissa kappaleissa, sillä lähikuvissa tällaiset tekstuurit eivät välttämättä näytä kovin vakuuttavilta (Birn 2006, 289).

### 3.5.4 Bittikarttojen heikkoudet

Bittikarttojen heikkoutena on niiden rajallinen tarkkuus eli resoluutio ja huono skaalautuvuus. Toisaalta mitä tarkempia kuvia renderöinnissä käytetään, sitä enemmän tarvitaan työmuistia, mikä edelleen hidastaa prosessia. Siksi Birn kehottaa pohtimaan, miten kutakin tekstuuria käytetään renderöinnissä. Jos kyseessä on vain pieni kappale tai sellainen joka näkyy vain kaukana, ei kuvatekstuurin tarvitse olla kovin tarkka, mutta läheltä tarkasteltavien kappaleiden kohdalla tarkkuuden tulee olla niin hyvä, että yksityiskohdat erottuvat selvästi. (Birn 2013.)

Jos bittikarttaa käytetään hyvin laajoilla pinnoilla kuten rakennusten seinissä tai lattiassa, täytyy bittikartan olla joko suurempi kooltaan tai siinä tulee olla vain vähän silmään osuvia yksityiskohtia. Koska laajat pinnat peitetään yleensä toistuvilla bittikartoilla laatoituksen tai tapetin tapaan, selkeästi erottuvat tai toistuvat yksityiskohdat ja toisaalta reunojen saumakohdat antavat tekstuurille epärealistisen vaikutelman. Vaikka tekstuurit näyttäisivät sinällään liian selkeästi toistuvilta, Birn neuvoo silti kokeilemaan niitä käytännössä, sillä renderöityinä tekstuurit voivat näyttää varjoista ja pintoja peittävästä muista kappaleista johtuen toimivammalta (Birn 2013).

### 3.5.5 Proseduraaliset tekstuurit

Bittikartoille on olemassa myös kevyempi vaihtoehto, mutta sitä ei voi käyttää joka tilanteessa. Kyseessä ovat proseduraaliset eli matemaattisiin algoritmeihin perustuvat tekstuurit (procedural textures). Birn toteaa, että parhaiten ne toimivat laajoilla pinnoilla, sillä ne voivat peittää rajattoman suuria alueita saumattomasti ja ilman toistuvia yksityiskohtia, ja ne skaalautuvat yleensä erinomaisesti eri koihin, lähelle ja kauas. Lähikuvissa näkyvät tekstuurit Birn kehottaa kuitenkin

testaamaan huolellisesti, sillä proseduraaliset tekstuurit eivät käyttäydy kaikissa tilanteissa aivan oletetusti. (Birn 2013.)

Proseduraalisia tekstuureja on yleensä ainakin jonkin verran valmiina renderöintimoottoreissa, mutta niitä voi tehdä itsekin. Se tosin edellyttää joko kokeilua yrityksen ja erehdyksen kautta, tai hieman syvällisempää ymmärrystä matematiikasta ja tekstuureissa käytetyistä algoritmeista. Myös proseduraalisia tekstuureja voi ja kannattaa yhdistellä esimerkiksi valokuvatekstuurien kanssa.

### **3.5.6 Kiiltävyys**

Koska todellisuudessa käytännössä kaikki materiaalit kiiltävät ainakin jonkin verran valaistuksesta ja katselukulmasta riippuen, myös mallinnetuille materiaaleille täytyy määritellä kiiltävyysasetukset. Yleensä renderöintimoottoreissa kiiltävyys voidaan määritellä yleisesti koko kappaleelle, mutta tämä on harvoin paras vaihtoehto, sillä tosielämässä muun muassa pintakäsittely, pöly, lika tai vaikkapa sormenjäljet vaikuttavat pinnan kiiltoon. Paremman lopputuloksen voikin saavuttaa käyttämällä kiiltokarttoja (specular maps).

Mustavalkoisia kiiltokarttoja tulkitaan yleensä siten, että mitä valkoisempi alue on, sitä enemmän se kiiltää. Birn huomauttaa, että kartta saa tietyt alueet kiiltämään vasta silloin, kun niihin osuu valoa. Kyseisissä kohdissa siis olisi muutenkin valoheijasteita, mutta kartan avulla heijasteita voidaan muokata yksityiskohtaisemmiksi. Vastaavasti kartan tummia alueita hyödyntämällä voidaan jokin alue määritellä lähes mattaväriseksi. (Birn 2006, 286.)

Kiiltokartan ohella heijasteisiin voidaan vaikuttaa karkeuskartalla (gloss map), sillä myös pinnan mikrotason epätasaisuudet eli pinnan karkeus vaikuttaa kiiltävyyteen. Russell toteaa, että mitä karkeampi pinta on, sitä enemmän valo hajaantuu heijastuessaan pinnasta ja sitä epätarkemmalta heijastus näyttää. Mustavalkoi-

sella karkeuskartalla määritetään pinnan karkeutta siten, että vaaleammat sävyt tarkoittavat karkeampaa pintaa. (Russell 2016.)

Materiaalien kiiltävyys, karkeus ja diffuusiväriin voimakkuus ovat todellisuudessa sidoksissa toisiinsa energian säilyvyyden pohjalta. Kun kappaleeseen osuu valoa, osa siitä imeytyy, osa siroaa ja osa heijastuu. Imeytyneen, siroutuneen ja heijastuneen valon määrä on yhtä suuri kuin kappaleeseen osuneen valon määrä. Kappale ei siis voi ikinä heijastaa enempää valoa kuin siihen osuu (Russell 2016).

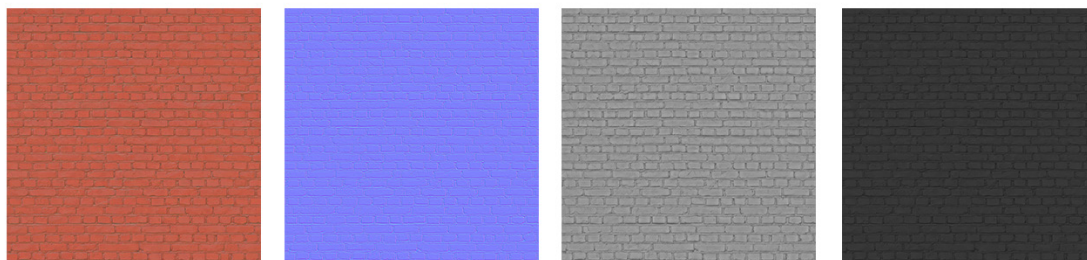
Tästä seuraa että kiiltävyys ja diffuusiväri (mattaväri) ovat toisensa poissulkevia. Mitä kiiltävämpi kappale siis on, sitä vähemmän sen perusväri näkyy ja vastavasti päinvastoin. Myös pinnan karkeus toimii samalla periaatteella: mitä karkeampi pinta on, sitä laajemmalle heijastus levittäytyy kappaleen pinnalla ja sitä heikommaksi heijastuneen valon intensiteetti käy. Toisaalta hyvin sileässä pinnassa heijastus on pienempi mutta voimakkaampi. (Russell 2016.)

Energian säilyvyys on ominaisuus, jonka voi toteuttaa PBR-moottoreilla, mutta se ei välttämättä ole automaattisesti mukana. Esimerkiksi Blenderin Cycles-moottorissa energian säilyvyys on mahdollista huomioida sekoittamalla diffuusisävytin kiiltosävyttimeen siten, että toisen osuuden kasvaessa toinen vähenee samassa suhteessa. Sävyttimiä voi toki tarkoituksella yhdistellä toisiinsa myös muilla tavoilla energian säilymisestä piittaamatta, mutta silloin ei ole enää tarkalleen ottaen kyse PBR-materiaalista. Cycles kuitenkin jättää mahdollisuuden tällaisille taiteellisille valinnoille.

### **3.5.7 Normaalikartat**

3D-malleihin ei kannata mallintaa kaikkia yksityiskohtia tarkasti, koska yksityiskohdat lisäävät mallissa tarvittavaa geometriaa, mikä puolestaan lisää tarvittavan muistin määrää ja hidastaa renderöintiä. Peligrfiikassa ja myös interaktiivisissa

visualisoinneissa mallit kannattaa pitää suhteellisen yksinkertaisina, koska tarvittavia yksityiskohtia voidaan lisätä kevyemmin pintatekstuureilla. Näitä ovat normaalikartat (normal maps), kohoumakartat (bump maps) sekä poikkeutuskartat (displacement maps) (kuva 9).



Kuva 9: Esimerkkejä erilaisista tekstuurikartoista. Vas. diffuusikartta (albedo), normaalikartta, poikkeutuskartta ja kiiltokartta (julkaistu Poliigon.comin luvalla).

Normaalikartat ovat yleensä enimmäkseen sinisävyisiä karttoja, jotka määrittävät pintojen normaalin eli pintaa vastaan kohtisuoran suunnan 3D-avaruudessa. Jo itse 3D-malleissa on määritelty jokaiselle pinnalle normaali eli suunta, jonne pinnan taso osoittaa. Pinnan normaalin perusteella renderöintimoottori osaa laskeasen, miten valo käyttäytyy (pinnan valaistus, heijastukset, varjot) osuessaan kyseiseen pintaan. Normaalikartoilla nämä normaalit voidaan määritellä koko kappaleelle jopa yksittäisten pisteiden tasolla.

Birnin mukaan normaalikartat ovat kätevä ja varsin yleinen tapa muuttaa kappaleen sävytystä mallin geometriaa muuttamatta. Tyypillisesti niiden avulla saadaan karkeammalla geometrialla ja vähemmällä monikulmioilla rakennettu malli (low-poly model) näyttämään tarkemmin mallinnetulta kappaleelta (high-poly model). Usein tällaiset normaalikartat itse asiassa kehitetään (baking) tarkemmista malleista renderöintimoottorin avulla, minkä jälkeen tarkan mallin karttaa käytetään karkeamman mallin kanssa. Vaikka malli voi näyttää hyvin yksityiskohtaiselta, kulmikkaasta silhuetista voi usein kuitenkin päätellä, että siinä on käytetty karkeampaa geometriaa. (Birn 2006: 293.)

### 3.5.8 Kohoumakartat

Kohoumakarttojen avulla voidaan luoda illuusio kappaleen pinnassa olevista kohoumista, kuopista tai muista vastaavista epätasaisuuksista. Birn sanoo, että kohoumakartat ovat yleensä harmaasävykarttoja, joissa vaaleat sävyt tulkitaan kohoumiksi ja tummat sävyt taas kuopiksi. Kohoumakartat eivät siis muuta itse mallin geometrista rakennetta, vaan muuttavat normaalikarttojen tapaan, joskin suoraviivaisemmin, pinnan normaaleja. (Birn 2006, 291.)

Birnin mukaan erona normaalikarttoihin on se, että kun normaalikartoissa jokaiselle pinnan pisteelle on määritelty normaali, kohoumakartoissa vertaillaan vierekkäisten kuvapisteiden korkeustietoa ja niiden perusteella lasketaan yksittäisten pisteiden normaalit. Ja kuten normaalikarttojen kohdalla, myös kohoumakarttoja käytettäessä mallin silhuetti ei muutu, mikä voi paljastaa käytetyn tekniikan. Sen vuoksi myös tällaisten kappaleiden heittämät varjot mukailevat mallien alkuperäistä geometriaa. (Birn 2006, 292.)

### 3.5.9 Poikkeutuskartat

Kappaleen geometriaa voidaan muuttaa tekstuurin avulla, jos käytetään poikkeutuskarttaa. Birn selittää, että mitä vaaleampi sävy kartassa on, sitä kauemmas pinnasta vastaavat pisteet siirretään. Yleensä musta on kartan perustaso, josta vaaleampia kohtia siirretään ulommas. Joissain moottoreissa on kuitenkin mahdollista säätää offset-arvoa eli sitä, mistä korkotasosta musta väri lasketaan mallille. Mallin lopullinen muoto määräytyy kuitenkin vasta sitten, kun se lasketaan poikkeutuskartan avulla renderöinnin yhteydessä. (Birn 2006, 290.)

Toisin kuin edellä mainitut kartat poikkeutuskartta edellyttää mallilta tiheämpää geometriaa, ja yleensä tulos on sitä parempi mitä enemmän mallissa on yksittäisiä verteksejä eli geometriapisteitä. Mitä yksityiskohtaisemmasta kartasta on

kyse, sitä enemmän mallissa tulee ole monikulmiopintoja. Muutoin lopputulos voi olla kulmikas ja epätarkka. Samalla tietenkin vaadittavan muistin määrä kasvaa ja renderöinti hidastuu, joten poikkeuskarttaa kannattaa käyttää harkiten. Jos kappaleen mallintaminen käsin olisi erittäin monimutkaista, voi sopivan poikkeuskartan käyttäminen olla perusteltua. (Birn 2006, 290.)

### **3.5.10 Ympäristövarjostuskartat**

Kappaleiden pinnanmuotoja voidaan korostaa myös ympäristövarjostuskarttojen (AO-kartta / ambient occlusion map) ja joissain moottoreissa myös kolokarttojen (cavity map) avulla. Birnin mukaan AO-algoritmi luo kappaleisiin pehmeitä varjoja sellaisiin kohtiin, joihin hajavaloa ei pääse kappaleen itsensä (urat, kolot, kappaleen alle jäävät osat) tai muiden kappaleiden varjostuksen takia (nurkat, kulmat) (Birn 2006, 73-74). Vaikka AO-kartat eivät tarkalleen ottaen ole fysiikan lakien mukaisia, niitä käytetään silti toisinaan täydentämässä yksityiskohtia. Wilson mainitsee, että AO-kartat kehitetään tyypillisesti renderöintimoottorin avulla suoraan malleista ja niitä käytetään yleensä yhdessä diffuusi- tai albedokarttojen kanssa (Wilson 2016a).

## **3.6 Muut yksityiskohdat**

Kun vertaa aloittelevien tekijöiden ja 3D-grafiikan huippuammattilaisten fotorealistisia visualisointeja, silmiinpistävin ero näkyy yleensä runsaissa yksityiskohdissa ja huolellisessa viimeistelyssä. Vaikka jälki on toki muutoinkin usein näyttävämpää kuin vasta-alkajilla, yksityiskohtien huomioiminen nostaa pitkään alalla toimineiden visualisoinnit aivan omalle tasolleen. Vaikka ihmissilmä on tottunut sivuuttamaan monet näistä pienistä visuaalisista tekijöistä epäolennaisina tai ikään kuin normitilanteena, ne lisäävät kuitenkin oman mausteensa kokonaisuuteen.



Yksi tavanomaisimmista virheistä, joihin aloittelevien tekijöiden töissä törmää, on liiallinen siisteys. Todellinen maailma on kaikkea muuta kuin siisti ja puhdas. Likaa löytyy kaikkialta, ja kuten Roman on todennut (Pricen referoimana), käytännössä kaikissa kappaleissa on muun muassa pölyä, kulumisjälkiä, roiskeita, sormenjälkiä tai valumajälkiä (Price 2015). Jos ajatellaan työn kohteena olevaa CHP-laitosta, on selvää, että laitoksen rakenteista löytyy varmuudella juuri pölyä ja nokea, eikä kolhuilta ja muilta kulumisjäljiltäkään voi varmasti välttyä. Visualisointi vaikuttaa heti autenttisemmalta kun siihen lisätään muun muassa varoitustarroja ja ohjekylttejä, joita on eri puolilla laitosta. Kuten aiemmin materiaalien kohdalla oli jo puhetta, suurimman osan liasta, sormenjäljistä, kulumajäljistä ja tarroista voi toteuttaa yhdistämällä osittain läpinäkyviä kuvakarttoja toisiin karttoihin tai mallien perussävyttimiin.

## **4 Blender**

### **4.1 Blenderin perusteita**

Blender on avoimeen lähdekoodiin perustuva ilmainen 3D-mallinnus- ja animointiohjelmisto, joka sisältää kaikki olennaiset työkalut muun muassa 3D-mallintamiseen, renderöintiin, teksturointiin ja kompositointiin (Blender Foundation 2016). Päädyin käyttämään tässä projektissa Blenderiä siksi, että se on minulle kaikkein tutuin väline ja toisaalta siksi, että sitä voi käyttää GPL-lisensoinnin ansiosta maksutta ja täysin vapaasti myös kaupallisiin tarkoituksiin. GPL-lisenssin ansiosta tässä työssä on voitu käyttää ilman erillistä lupaa myös Blenderistä otettuja ruutukaappauksia.

Blenderissä on nykyisin vakiona kaksi eri renderöintimoottoria: alkuperäinen Blender Render Engine (tunnetaan myös nimellä Blender Internal, BI) sekä uudempi Cycles. BI-moottorikin kykenee varsin realistiselta näyttäviin renderöin-

teihin, mutta se käyttää siinä monenlaisia kikkoja ja siksi sen heikkous on valon käsittelyssä ja erityisesti varjoissa tai valoheijastuksissa. Cycles käyttää säteenjäljitystä ja muita fysikaalisiin malleihin perustuvia tekniikoita, ja siksi sen tuottama jälki on selvästi laadukkaampaa esimerkiksi silloin, kun valo siroaa lasissa tai läpikuultavissa materiaaleissa. (Simonds 2013, 183.)

## 4.2 Cycles

### 4.2.1 Taustaa lyhyesti

Koska Cycles-renderöintimoottorilla on mahdollista mallintaa tarkasti fysiikan lakeja noudattavia PBR-materiaaleja, on ehkä parempi tutustua ensin siihen ja katsoa sen jälkeen, miten samoja periaatteita sovelletaan BI- ja Blend4Web-moottoreissa. Cyclesissa mallinnettuja materiaaleja voi tarvittaessa ”leipoa” (bake) tekstuureiksi, joita sitten edelleen hyödynnetään BI- ja Blend4Web-materiaaleissa. Vaikka Cyclesin toimintojen ja erilaisten valintojen perinpohjainen hallitseminen vaatii paljon opettelua, se tarjoaa silti melko selkeät ja visuaalisesti helpotajaiset mekanismit materiaalien määrittämiseen. Siksi sen avulla on helpompi ymmärtää eri säätöjen vaikutus materiaalin ulkonäköön ja miten PBR-teoria yhdistyy käytäntöön.

Sairiala tutki omassa opinnäytetyössään Cycles-renderöintimoottorin PBR-työnkulkua. Sairialan mukaan Cycles ei sinänsä ole PBR-renderöintimoottori, mutta se tarjoaa mahdollisuuden myös PBR-renderöintiin (Sairiala 2015, 33). Cyclesin ongelmana on tosin se, että se tarjoaa tällä hetkellä (syyskuu 2016) ainoastaan perusvälineet (sävyttimet) materiaalien tuottamiseen, toisin kuin monet kaupalliset sovellukset. Cycles-moottoria kehitetään kuitenkin edelleen, ja GPL-lisenssin ansiosta tavalliset käyttäjät tuottavat Blenderiin uusia materiaaleja ja muita ominaisuuksia jatkuvasti.

Kuten Sairialakin toteaa, että kyseisiä sävyttimiä yhdistelemällä voidaan luoda rajattomasti erilaisia materiaaleja (Sairiala 2015, 37–38). Se ei ole kuitenkaan varsinkaan aloittelevalle tekijälle mikään ihan helppo tehtävä. Oman kokemukseni perusteella erilaisten materiaalien kehittelyyn ja niiden ominaisuuksien ymmärtämiseen ja hiomiseen menee tavattoman paljon aikaa, vaikka periaatteet on helppo omaksua. Opeteltavaa on paljon ja joskus jokin aivan pieni yksityiskohta voi olla ratkaisevan tärkeä renderöinnin laadun parantamisessa.

#### 4.2.2 Cycles-materiaalien määrittäminen

3D-grafiikassa materiaalit määritellään sävyttimien avulla. Sävyttimet on renderöintimoottoreissa ohjelmarutiineja, jotka antavat kappaleille niiden lopullisen ulkoasun. Cyclesissa näitä sävyttimiä on neljää tyyppiä: BSDF (tai pinta, surface), emissio (tai säteily, emission), tilavuus (volume) sekä tausta (background).

Ensimmäinen tyyppi eli BSDF (Bidirectional Scattering Distribution Function) määrittää miten valo heijastuu, taittuu tai imeytyy kappaleen pinnasta, ja tätä tyyppiä käytetään yleisimmin tavallisissa kiinteissä kappaleissa. Emissiosävytin määrittää valosäteilyn kappaleen pinnasta tai tilavuudesta ja sitä tarvitaan erityisesti itsevalaisevissa kappaleissa, mutta sitä voidaan käyttää myös lamppuissa (tosin lamppuja ei yleensä tarvitse määritellä solmujen avulla). Tilavuussävyttimet määrittävät valon imeytymisen tai siroamisen kappaleen sisällä (tai tilavuudessa), ja siksi niitä käytetään valon kulun määrittämiseen esimerkiksi savupilven tai nesteen sisällä. Taustasävytin on tarkoitettu lähinnä vain ympäristövalaistuksen määrittämiseen solmujen avulla. (Blender Foundation 2016: Cycles Render Engine / Nodes / Introduction.)

Koska tässä opinnäytetyössä toteutettavassa visualisoinnissa on tarkoitus käyttää lähinnä vain fotorealistisia PBR-materiaaleja (tai niitä mahdollisimman pitkälti vastaavia), tässä yhteydessä käsitellään ainoastaan BSDF-sävyttimien avulla

määriteltäviä materiaaleja. Koska laitoksen rakenteissa on käytetty enimmäkseen terästä sellaisenaan ja maalattuna, messinkiä sekä muovia, painopiste on PBR-metallien ja -eristeiden mallintamisessa. Lasia on käytetty vain harvoissa kohdissa ja se poikkeaa muista materiaaleista läpinäkyvyytensä vuoksi. Sille löytyy onneksi valmis sävytin Cyclesista.

### 4.2.3 Node Editorista lyhyesti

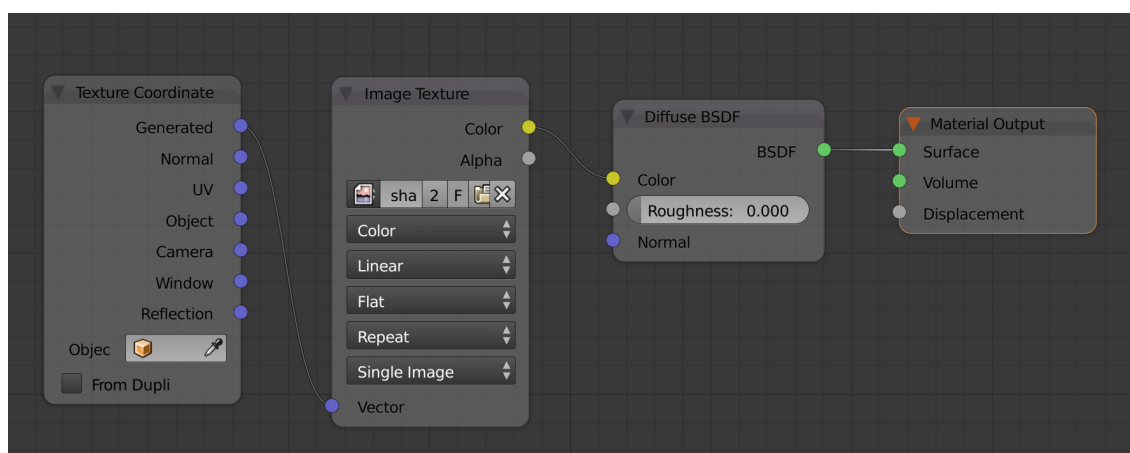
Vaikka materiaalit voi määrittellä Cyclesissa suoraan materiaalipaneelin kautta, monimutkaisemmissa materiaaleissa paneelin käyttö on todella monimutkaista, ellei jopa täysin mahdotonta. Materiaalit on paljon selkeämpää määrittellä solmujen käyttöön perustuvalla Node Editorilla, joka on kiinteä osa Blenderiä. Vastavanlainen editori löytyy myös monesta muusta renderöintimoottorista.

Node Editorin perusideana on se, että materiaalit kasataan erilaisista ominaisuuksista, jotka on määritelty solmuina (node). Kukin solmu lisää materiaaliin jonkin ominaisuuden, ja erilaisia solmuja yhdistelmällä materiaali saadaan näyttämään lopulta halutulta. Koska solmuja voi yhdistellä toisiinsa tiettyjen sääntöjen puitteissa lähes mielivaltaisesti, on erilaisia yhdistelmiä lukematon määrä.

Solmut on ryhmitelty Cyclesissa toiminnan perusteella eri tyypeiksi, jotta oikean solmun lisääminen olisi helpompaa ja nopeampaa. Esimerkiksi syötesolmut (input) ovat solmuja, joiden avulla materiaalille voidaan määrittellä erilaisia vakioarvoja ja tai vaikkapa se, miten käytettävä tekstuuri kytketään valittuun kappaleeseen. Sävytinsolmujen (shader) avulla taas määritellään materiaalissa käytettävät erilaiset sävyttimet (Diffuse BSDF, Glossy BSDF, Glass BSDF jne.). Sävytinsolmuihin kuuluvat myös edellä mainittuja yhdistelevät sekoitus- (Mix Shader) ja lisäyssolmut (Add shader). Kaikissa tässä työssä määriteltävissä materiaaleissa käytetään aina myös materiaalisolmua (Material Output), joka lopulta kokoaa yh-

teen kaikki määrittelyssä tarvittavat solmut. (Blender Foundation 2016: Cycles Render Engine / Nodes / Types.)

Solmut kytetään toisiinsa linkittämällä yhden solmun tulosteportti toisen solmun syöteporttiin (kuva 10). Yhden tulosteportin voi kytkeä usean eri solmun syöteporttiin, mutta kuhunkin syöteporttiin voi kytkeä vain yhden solmun. Syötesolmuissa on vain tulosteportteja, ja materiaalisolmussa ainoastaan syöteportteja. Muissa solmuissa on portteja sekä syötteitä että tulosteita varten. Portit on myös värikoodattu datatyypin mukaan, mikä helpottaa solmujen yhdistämistä oikeiden porttien avulla. Koska Node Editorissa solmut järjestetään aina vasemmalta oikealle, tulosteportit ovat solmujen vasemmassa reunassa ja syöteportit oikeassa. Viimeisin solmu oikealla on aina materiaalisolmu, joka on pakollinen materiaaleja määriteltäessä.



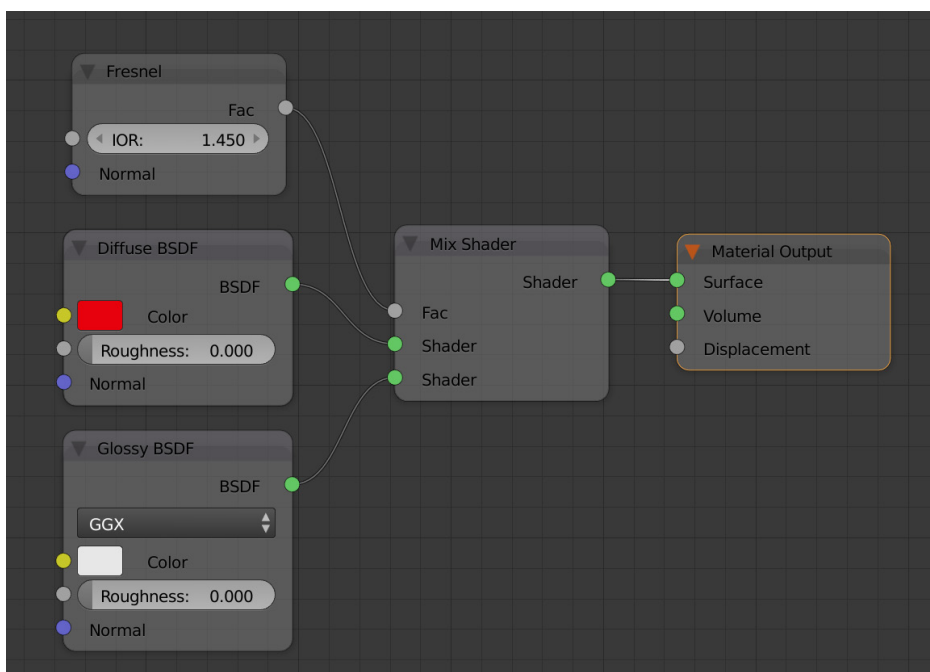
Kuva 10: Node Editorissa käytettäviä solmuja. Solmuissa syöteportit ovat vasemmalla ja tulosteportit oikealla, ja lisäksi kaikki portit on värikoodattu yhdistämisen helpottamiseksi (Kuvakaappaus: Harri Pietiläinen).

Solmuja voidaan tarvittaessa myös yhdistää ryhmittämällä, mikä voi selkeyttää varsinkin monimutkaisempia materiaaleja. Tällöin yhdeksi ryhmäksi määritellyt solmut näkyvät ja käyttäytyvät ikään kuin yhtenä solmuna, jolla on muiden solmujen tapaan omat syöte- ja tulosteportit.

#### 4.2.4 PBR-eriste

Eristeissä valo käyttäytyy siten, että osa valosta heijastuu suoraan kappaleen pinnasta, osa imeytyy pintakerroksissa ja osa siroaa joka puolelle ympäristöön antaen kappaleelle sen ominaisvärin (diffuusiväri). Kappaleen pinnan mikrorakenne taas määrittelee sen, miten heijastunut valo jatkaa matkaansa eli miten peilimäinen heijastus on. Lisäksi Fresnel-ilmiö lisää pinnan heijastavuutta, kun katselukulma lähenee pinnantasoa. Näiden tietojen pohjalta voidaan lähteä rakentamaan eristeelle (esim. muovi) sopivaa materiaalia.

Koska toteutustapoja on useita eikä niitä kaikkia voi käydä läpi tässä työssä, PBR-materiaalit toteutetaan Pricen ehdottoman mallin pohjalta. Sairiala rakentaa omassa työssään vastaavat materiaalit eri lähtökohdista, mutta Pricen malliin verrattuna Sairialan toteutus on hieman vaikeaselkoisempi.

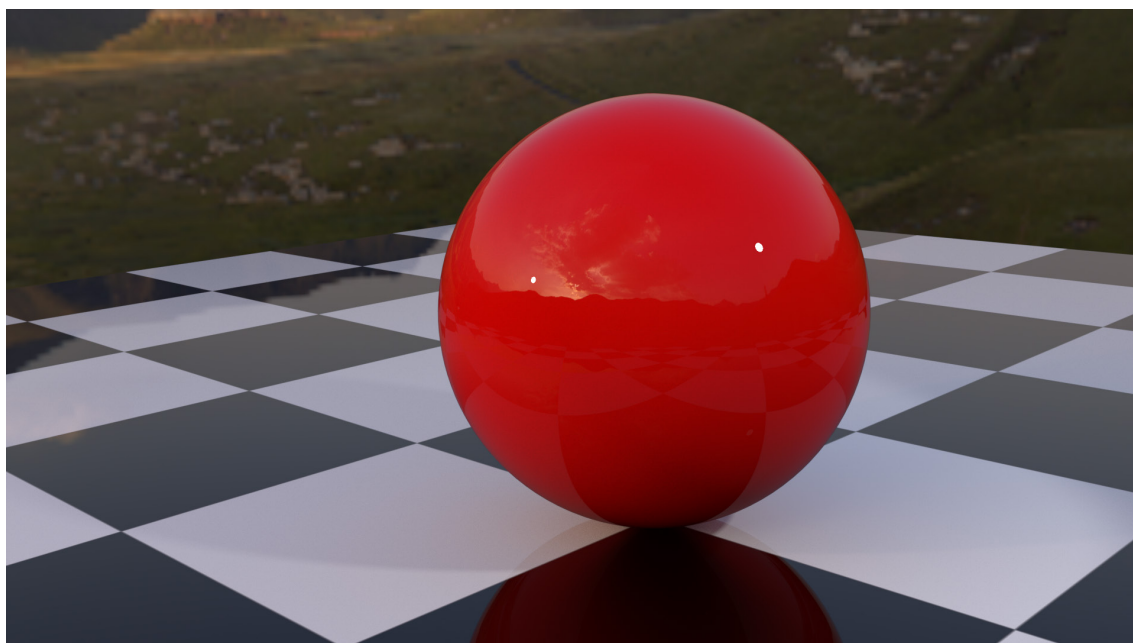


Kuva 11: PBR-eristeen määrittely yksinkertaisimmillaan. Diffuusiväri ja kiiltävyys sekoitetaan yhteen Fresnel-solmun tuottaman maskin perusteella (Kuvakaappaus: Harri Pietiläinen).

Cyclesissa on olemassa sävyttimet diffuusiväriä ja kiiltävyyttä varten, joten periaatteessa riittää, kun ne sekoitetaan Node Editorissa yhteen Mix Shader -solmulla.

Solmun liikusäädin (Fac) määrittää sävyttimien sekoitussuhteen tasaisesti koko kappaleelle. Sävyttimien sekoitussuhde voidaan määrittää myös Fresnel-solmun kappaleen perusteella tuottaman harmaasävymskin avulla. Kuvassa 11 sekoitetaan diffuusi- ja kiiltosävyttimet toisiinsa. Mitä tummempi jokin maskin alue on, sitä voimakkaammin vastaava kohta kappaleesta värjätään diffuusivärillä. Vastaavasti vaaleammassa kohdissa kiiltävyys korostuu.

Fresnel-solmussa on IOR-liikusäädin, jolla voi muuttaa Fresnelin taitekerrointa (Index Of Refraction, IOR) materiaalin mukaan, mutta yleensä oletusarvokin (1.450) on riittävän tarkka useimmille materiaaleille. Mitä suurempi arvo on, sitä laajemmalle heijastava alue levittäytyy. Toisaalta alle 1 menevien arvoja ei kannata käyttää (esimerkiksi kullan taitekerroin on 0.470), koska Blenderin Fresnel-toteutuksessa nämä arvot toimivat tulkitaan erikoisesti.

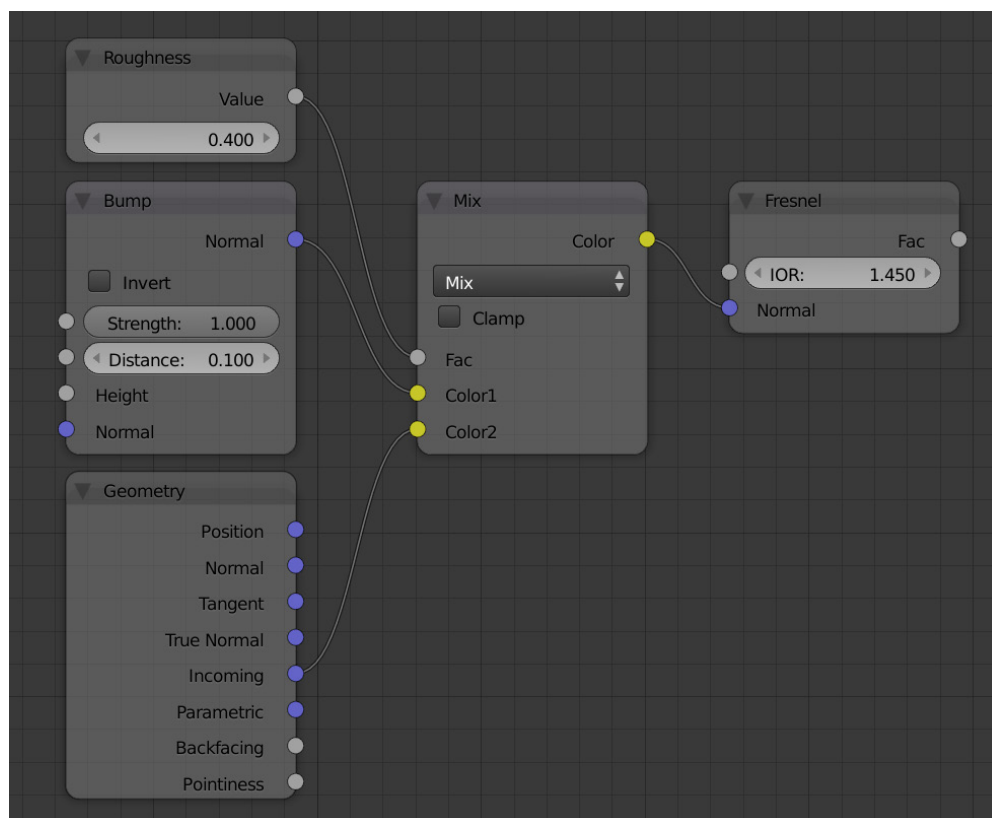


Kuva 12: Edellä kuvatun solmuasetelman vaikutus pallon pintaan. Reunoilla pallo näyttää kiiltävämmältä, mutta keskemmällä kiiltävyys vähenee. Diffuusiväri on kuitenkin vallitseva (Kuva: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

Price huomauttaa, että edellä kuvattu solmu yhdistelmä rajoittaa kuitenkin mahdollisuutta määrittellä kappaleen kiiltävyyttä, koska Fresnel-solmu on kytketty diffuusivärin ja kiiltävyyden suhdetta määrittävään säätimeen ja siksi kappale kiiltää vain siellä, missä Fresnel vaikuttaa (kuva 12). Toinen ongelma tässä on

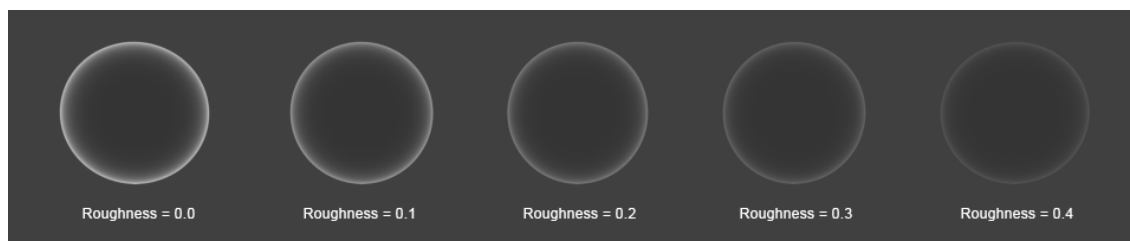
Pricen mukaan se, että todellisuudessa pinnan karkeus vaikuttaa suoraan Fresnel-ilmiön käyttäytymiseen: mitä karkeampi pinta, sitä heikompi Fresnel. (Price 2016a.)

Jälkimmäiseen ongelmaan Price esittää korjausta, joka saa Fresnelin muuttumaan syötteenä annettavan karkeusarvon mukaan (kuva 13). Kyseinen syöte kytketään myöhemmin myös Glossy-solmun karkeussäätimeen (Roughness), jolloin karkeuden lisääntyessä Fresnelin vaikutus heikkenee. Luonnostaan Fresnelin pienin arvo (F0) on kappaleen niissä osissa, missä pinnan normaali on suoraan kohti kameraa tai katsojaa. Tyypillisesti tuollaisessakin kohdassa pinnan heijastavuus on vielä 2–5 prosenttia. Ideana tässä ratkaisussa on sekoittaa Fresnelin vakioarvo yhteen Fresnel-minimin kanssa karkeussyötteen määrittämässä suhteessa. (Price 2016a.)



Kuva 13: Vaikka nykyisellään Fresnel ei heikkene pinnan karkeuden lisääntyessä, ongelma voidaan korjata pienellä virityksellä (Kuvakaappaus: Harri Pietiläinen).





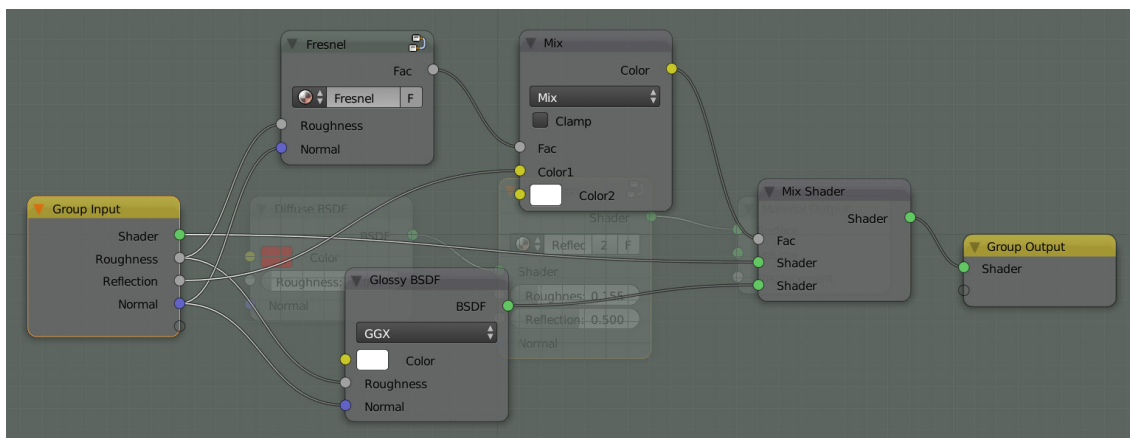
Kuva 14: Yllä esitetyn korjauksen avulla Fresnel heikkenee karkeuden kasvaessa. Mitä vaaleampi Fresnel-alue on, sitä voimakkaammin se heijastaa valoa (Kuva: Harri Pietiläinen).

Pricen mukaan edellä mainitut Fresnel-arvot voidaan tuottaa sekoittamalla yhteen Bump-solmun Normal-arvo ja Geometry-solmun Incoming-arvo. Näistä ensimmäinen antaa syötteenä kappaleen pinnan normaaleita eli käytännössä se ei vaikuta Fresnel-arvoon mitenkään. Incoming-arvo puolestaan tarkoittaa suoraan kameraa kohti olevan pinnan arvoa eli sen avulla Fresnel-solmu saa miniminsä F0. Kun nämä sitten sekoitetaan värisekoitussolmulla ja sekoitussuhteena käytetään karkeussyötettä, lopputulos on juuri sellainen kuin pitääkin (kuva 14). Värisekoitussolmun (MixRGB) käyttäminen on pientä kikkailua, koska tässä yhdistetään normaalivektoreja eikä värejä, mutta parempaakaan vaihtoehtoa ei Blenderissä tällä hetkellä ole. Selkeyden vuoksi nämä solmut yhdistetään Fresnel-ryhmäksi. (Price 2016a.)

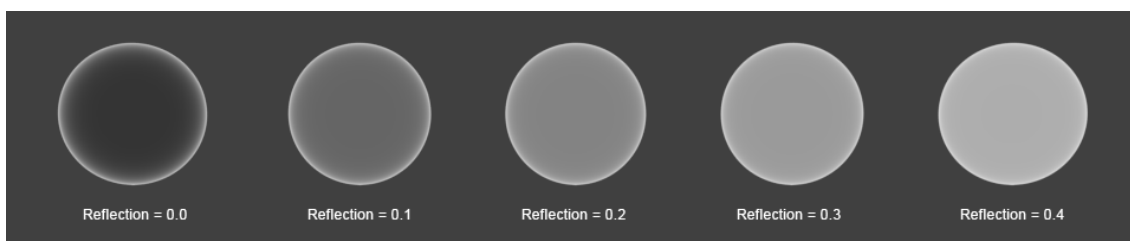
Toisen ongelman ratkaisu eli kiiltävyysasetuksen palauttaminen on hieman monimutkaisempi. Tätä varten Price rakentaa uuden Fresnel-ryhmän sekä Glossy-solmun pohjalta uuden Reflection-ryhmän, jonka avulla materiaalin kiiltoa voidaan lisätä Fresnelistä huolimatta. Edellä näimme miten Fresnelin vaikutus ilmenee vahvimmin pallon reunoilla, missä katselukulma pinnan normaaliin nähden on pienimmillään, ja heikkenee keskustaa kohti tultaessa. (Price 2016a.)

Uudessa Reflection-ryhmässä korjatun Fresnel-solmun jälkeen on lisätty MixRGB-solmu, jonka avulla koko pallon kiiltävyyttä voidaan lisätä vaalentamalla sekoitussuhdetta määrittävää maskia (kuva 15). Toinen kyseisen MixRGB-solmun väreistä lisätään ryhmän syötteenä ja se näkyy ulospäin liikusäätimenä. Sen avulla maskin vaaleutta voidaan säätää portaattomasti. Kuvassa 16 Fresnel-arvo

pysyy koko ajan samana, mutta Reflection-ryhmän Reflection-säätimen arvoa muutetaan pykälä kerrallaan. (Price 2016a.)



Kuva 15: Ryhmäsolmun editointitilassa.nähdään, että uuteen Reflection-ryhmään on lisätty MixRGB-solmu, jonka avulla koko kappaleen heijastavuutta voidaan säätää Fresnelistä huolimatta (Kuvakaappaus: Harri Pietiläinen).

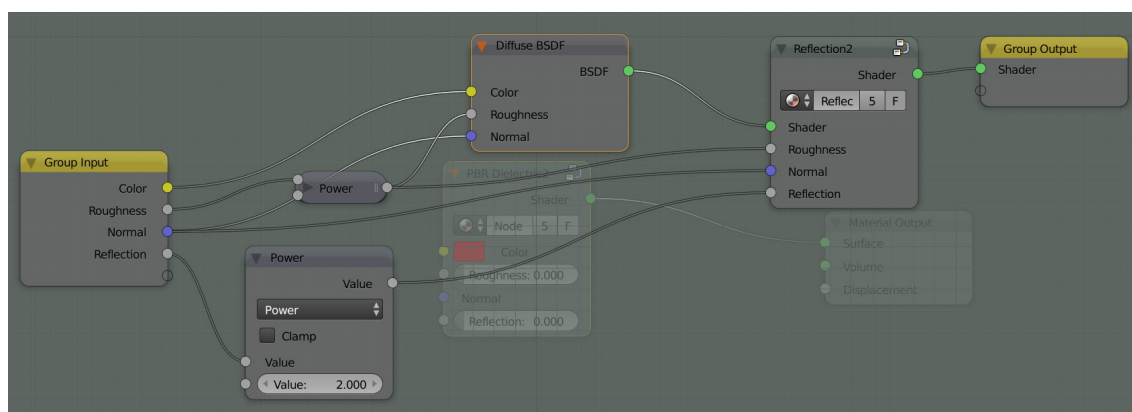


Kuva 16: Uuden Reflection-ryhmän Reflection-arvoa säätämällä voidaan muuttaa koko pallon kiiltävyyttä Fresnel-arvon säilyessä ennallaan (Kuva: Harri Pietiläinen).

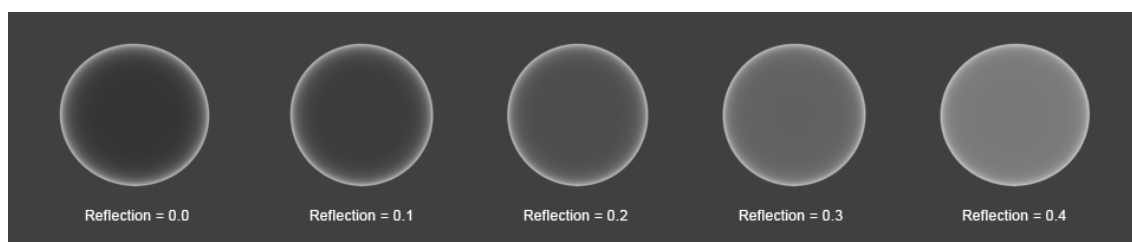
Kuten kuvasta 16 huomataan, ero ensimmäisen ja toisen arvon välillä on melko näkyvä, vaikka muutos on lukuarvona vain 0.1. Seuraavien välillä ero on jo huomattavasti pienempi. Tämä johtuu siitä, Blenderin toteutuksessa kiiltävyys- ja karkeusarvot eivät muutu lineaarisesti, vaan gammakorjauskäyrän mukaan. Gammakorjausta käytetään tasaamaan kuvien väritasapainoa, kun niitä näytetään tietokoneen ruudulla, ja ilmeisesti Blenderin toteutuksessa on haettu juuri tätä vastaavuutta kiiltävyys- ja karkeusarvojen kohdalla (edes Blenderin online-manuaalista ei löydy tarkempaa tietoa aiheesta). (RealLifePudding 2016.)

RealLifePudding-blogin mallin pohjalta Price kehottaa lisäämään Math-solmun sekä Roughness- että Reflection-arvoille. Näin karkeus- ja kiiltävyysarvojen muutos on tasaisempi ja helpommin säädettävissä koko arvoalueella. Price yhdistää

ensin Diffuse BSDF -solmun ja Reflection-ryhmän uudeksi materiaalityhmäksi, jolle annetaan nimeksi PBR Dielectric (eli PBR-eriste). Sen jälkeen lisätään Math-solmut, joissa kummassakin jälkimmäiseksi lukuarvoksi asetetaan 2 ja operaatioksi Power (kuva 17). Säätimen arvot korotetaan siis toiseen potenssiin, mikä muokkaa arvojen todellista vaikutusta ominaisuuksiin nähden aiempaa lineaarisemmiksi, kuten voidaan todeta kuvasta 18. (RealLifePudding 2016; Price 2016.)



Kuva 17: Lopullinen PBR Dielectric -ryhmä, joka siis toteuttaa PBR-eristemateriaalin. Mukaan lisätyt Math-solmut muokkaavat karkeus- ja kiiltävyyssäätimien arvoja lineaarisemmaksi ja siten helpommin hallittaviksi koko arvoalueella 0.0–1.0 (Kuvakaappaus: Harri Pietiläinen).

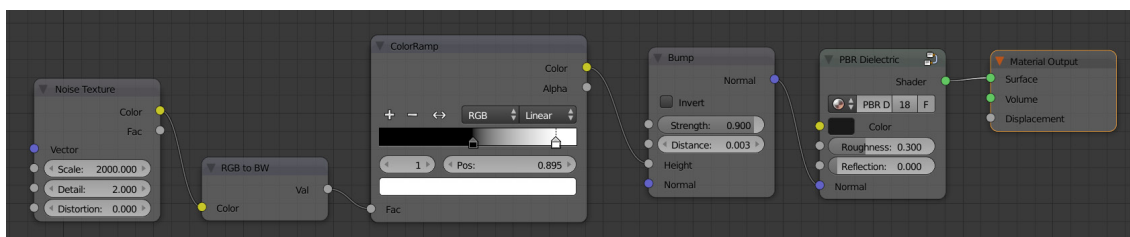


Kuva 18: Kun säätimen arvot korotetaan Math-solmun avulla toiseen potenssiin, arvojen vaikutus lopputulokseen – tässä tapauksessa Reflection-arvoon – muuttuu vähitellen ja tasaisemmin kuin aiemmassa esimerkissä. Fresnel-arvo pysyy tässäkin samana (Kuva: Harri Pietiläinen).

PBR-materiaalia voidaan hyödyntää lähes sellaisenaan monissa eri yhteyksissä, sillä eristeiksi lasketaan paitsi kaikki puu-, kumi-, muovi- ja lasipinnat, myös maalatut metallipinnat. Maali on nimittäin eriste, vaikka metalli itsessään on johde. Maalipinta ei välttämättä vaadi tekstuureja, paitsi silloin kun kyse on valetusta metallista. Valetut pinnat ovat usein karheita ja siksi maalipintakin on hieman

epätasainen. Cyclesissa voidaan pintaa ”karhentaa” yhdistämällä PBR-eristemateriaaliin Cyclesin omalla Noise-tekstuurilla aikaansaatu kohoumakartta. Noise-tekstuuria voidaan säätää siten, että sen tuottama kuvio on hyvin pienijakoista, aivan kuten todellisessa valetussa pinnassa.

Noise-teksturi muokataan edelleen mustavalkoiseksi RGB to BW -solmulla ja kuvion kontrastia ”tiivistetty” ColorRamp-solmulla, ja lopuksi teksturi yhdistetään Bump-solmun kautta PBR-materiaalin Normal-syöteporttiin. Bump-solmu kertoo Cyclesille, että kyse on kohoumakartasta, jolloin pinnasta tulee renderöinnissä karheen näköinen. Samalla itse PBR-materiaalin karheutta lisätään säätämällä Roughness-arvoa suuremmaksi. Näin valoheijastukset leviävät hieman laajemmalle eivätkä ne ole enää teräviä ja peilimäisiä (kuva 19).



Kuva 19: Esimerkiksi valuraudalle käytettävän PBR-eristemateriaalin solmurakenne. Valetussa pinnassa näkyvä karheus tuotetaan Cyclesin Noise-tekstuurin avulla (Kuvakaappaus: Harri Pietiläinen).

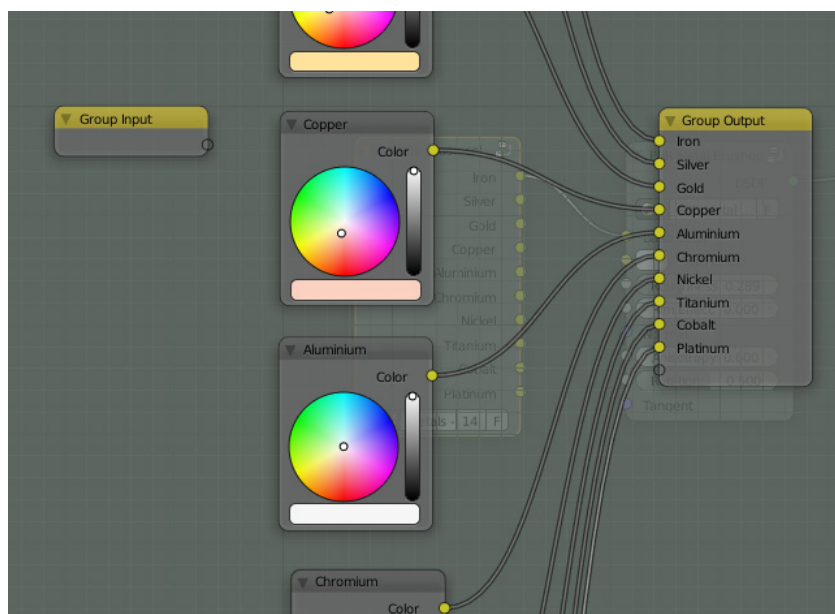
Kun samaa PBR-materiaalia käytetään sitten muissa pintamateriaaleissa, esimerkiksi muovin ja kumin kohdalla Roughness-arvon tulee olla vielä suurempi. Näin siksi, että näissä materiaaleissa pinnan mikrorakenne on paljon karheampi, vaikka ne tuntuisivat ja näyttäisivät sileämmiltä. Varsinkin kumipinnoissa heijasteet ovat yleensä laajempia, heikompia ja usein hyvin pehmeitä.

Sairiala huomauttaa, että käytännössä lähes kaikilla pinnoilla on todellisuudessa usein poikkeamia ja epäpuhtauksia, jotka voivat vaikuttaa pintojen heijastavuuteen ja jopa värjätä heijastuksia (Sairiala 2015, 37). CHP-laitoksessa epäpuhtauksina ilmenevät noki ja pöly voidaan lisätä materiaaliin Noise-tekstuurin avulla. Valumajäljet puolestaan on ehkä helpointa lisätä osittain läpinäkyvinä siirtokuva-tekstuureina (decals), sillä aidot kuvatekstuurit tuottavat yleensä paremman lop-

putuloksen kuin muilla tavoilla toteutetut ratkaisut. Kummassakin tapauksessa nämä lisäykset yhdistetään PBR-materiaaliin väri-informaationa. Sen sijaan eri paikoissa ympäri laitosta näkyvät varoitustarrat on näppärintä laittaa paikoilleen taso-objekteina, joihin varsinaiset kuvat on kytketty. Niitä ei ole kovin järkevää yhdistää suoraan materiaaliin, koska näiden kuvien kohdistaminen juuri oikeaan paikkaan on paljon hankalampaa kuin taso-objektien avulla.

#### 4.2.5 PBR-metalli

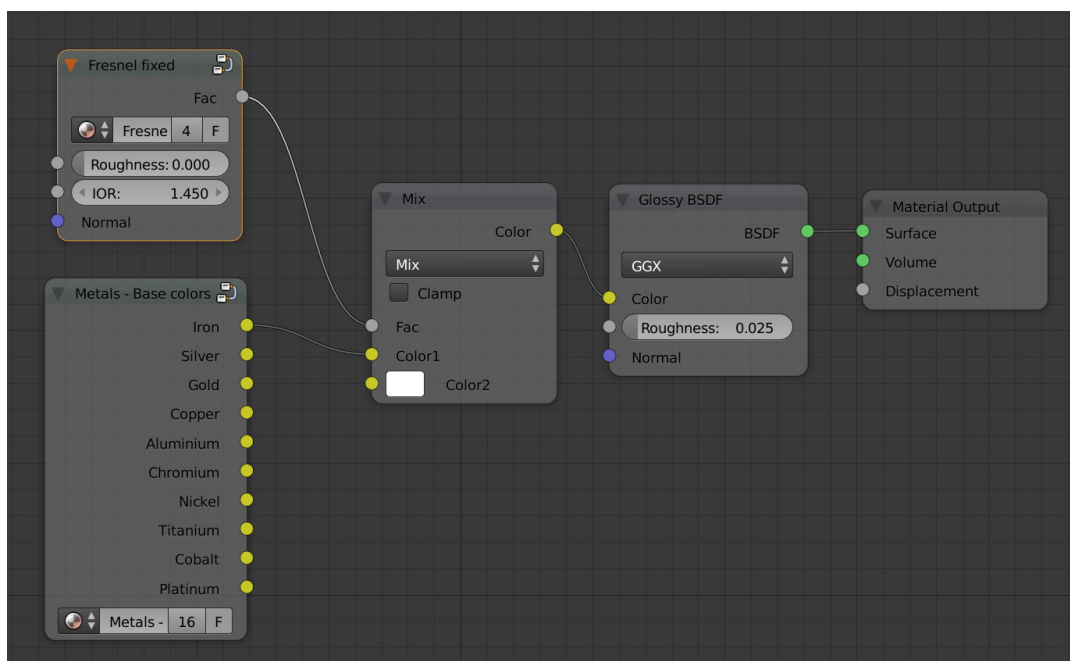
Valo käyttäytyy johteissa eri tavalla kuin eristeissä. Johteet, jotka tässä yhteydessä on selkeyden vuoksi rajattu vain metalleihin, heijastavat hyvin suuren osan valosta ja loppuosa imeytyy käytännössä välittömästi metallin pintarakenteeseen siroutumatta. Näin käsittelemättömien metallien pinta näyttää yleensä hyvin kirkkaalta ja melko värittömältä (pois lukien muun muassa kulta ja kupari), koska diffuusi- tai mattaväri puuttuu kokonaan toisin kuin eristeillä. Metallien heijastukset kuitenkin saavat kuitenkin pienen sävyn, joka antaa niille ominaisen kiillon.



Kuva 20: Metallivärit voi koostaa valmiiksi värikartaksi yhdistämällä oikeat värisävyt sisältävät RGB-solmut ryhmäksi (Kuvakaappaus: Harri Pietiläinen).

Edellä mainitusta seuraa, että PBR-metalli rakennetaan Cyclesissa pelkän Glossy BSDF -solmun pohjalta, eli PBR-metalli on puhtaasti kiiltävä materiaali. Vaikka metallit ovat lähes värittömiä ja harmaasävyisiä, eri metallien välillä on pieniä sävyeroja tummuusasteikolla: hopea on selvästi vaaleampaa tai kirkkaampaa kuin rauta tai titaani. Metallien väriarvoista kannattaa rakentaa värikartan yhdistämällä kutakin väriä varten säädetyt RGB-solmut yhdeksi ryhmäksi. Kun tulosteportit nimitetään vielä kunkin metallin mukaan, saadaan oikea sävy kytkettyä Glossy-solmuun hetkessä (kuva 20). (Price 2016b.)

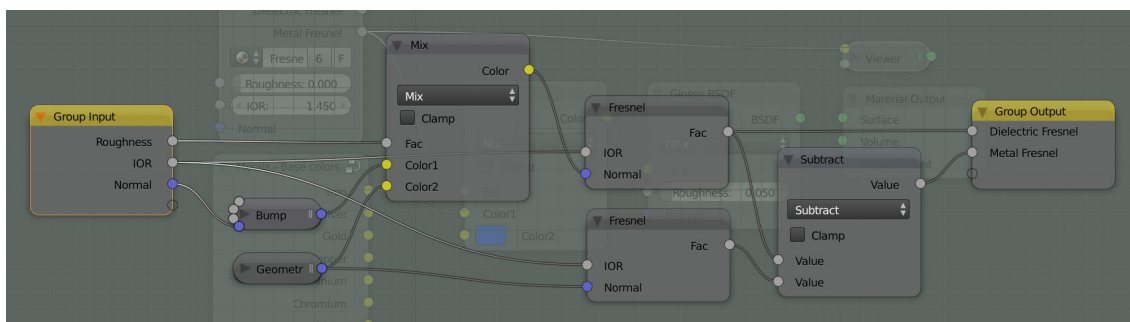
PBR-metallissa Price hyödyntää osaksi samaa rakennetta kuin edellä PBR-eristeen kohdalla, koska esimerkiksi Fresnel-ilmiö vaikuttaa myös metalleihin. Lisäksi metallien heijastukset voivat värjäytyä, mikä otetaan kyllä myöhemmin huomioon PBR-materiaalissa. Ensiksi kuitenkin valitun metallin sävy sekoitetaan Fresnel-maskin suhteessa valkoiseen väriin MixRGB-solmun avulla. Näin metallin perussävy näkyy materiaalissa kauttaaltaan, mutta Fresnel-maskiin yhdistetty valkoinen väri tekee pinnasta peilimäisen siellä, missä Fresnel vaikuttaa eli käytännössä kappaleen reunoilla (kuva 21). (Price 2016b.)



Kuva 21: Aiemmin luotua uutta Fresnel-ryhmää hyödynnetään osana PBR-metallimateriaalia. MixRGB-solmu määrittää Fresnelin vaikutuksen kappaleen reunoilla, joista Glossy-solmu tekee valkoisen sekoitusvärin perusteella peilimäisiä (Kuvakaappaus: Harri Pietiläinen).

Price kuitenkin huomauttaa, että aiemmin muodostettu Fresnel-ryhmä toimii kyllä mainiosti eristeiden kanssa, mutta metallien kohdalla se aiheuttaa hieman ongelmia. Normaalistihan ne pinnat, jotka ovat suoraan kohti katsojaa, näkyvät Fresnel-maskissa harmaina, mutta metallien kohdalla maskin liukuväriin täytyy jatkua aina mustaan saakka. Silloin perussävyyn ei sekoitu mukaan väritöntä kiiltoa ja sävy säilyy puhtaana. (Price 2016b.)

Price korjaa ongelman Fresnel-ryhmään siten, että siellä oleva Fresnel-solmu kopioidaan ja kopion Normal-porttiin kytketään Geometry-solmun Incoming-portti, jota käytettiin jo aiemmin toisessa yhteydessä. Näin Fresnel-kopio tuottaa maskin, joka on kauttaaltaan tasaisen harmaa, kuin sellainen pinta, jota katsotaan suoraan edestä. Kun alkuperäisestä maskista vähennetään Math-solmun avulla kopion tuottama tasaharmaa maski, lopputuloksena on alkuperäisen kaltainen maski, joka tummenee etualalla mustaksi (kuva 22). (Price 2016b.)



Kuva 22: Fresnel-ryhmää korjataan siten, että se tuottaa sopivan maskin myös metalleja varten. Metall-Fresnelin maskia tummennetaan vähentämällä alkuperäisestä maskista tasaharmaa maskin tuottama kopio (Kuvakaappaus: Harri Pietiläinen).

Koska samaa Fresnel-ryhmää hyödynnetään sekä eristeessä että johteessa, Price muuttaa Fresnel-ryhmän tulosteportteja siten, että eriste-Fresnelille ja metalli-Fresnelille on omat porttinsa. Ensimmäiseen kytketään alkuperäinen Fresnel-solmu ja jälkimmäiseen Fresnelien vähennyslaskuun käytetty Math-solmu. Portit myös nimetään selkeästi käytön helpottamiseksi, ja näin Fresnel-ryhmä sopii molempiin materiaaleihin. (Price 2016b.)

PBR-metalli kaipaa vielä heijastesävyä, joka on metalleilla yleensä hieman perussävyä vaaleampi. Sitä varten Price lisää materiaaliin vielä toisen MixRGB-solmun, joka kytketään metallivärikarttasolmun ja aiemman MixRGB-solmun väliin. Tällä solmulla metallin perussävy sekoitetaan yhteen heijastesävyn (tai reuna-sävyn, kuten Price sanoo) kanssa ja tämä yhdistelmä siis syötetään edelleen aiempaan MixRGB-solmuun. Perussävyn ja heijastesävyn sekoitussuhdemaskia varten Price korjailee jälleen Fresnel-ryhmää hieman paremmaksi. (Price 2016b.)

Fresnel-ryhmään lisätään vielä pari uutta solmua, joista ensimmäinen on Layer Weight. Tämä solmu voi tuottaa Fresnel- ja Facing-maskeja. Fresnel-maski on erilainen kuin varsinaisen Fresnel-solmun tuottama, ja koska solmusta puuttuu IOR-arvo, ei tämä Fresnel ole aivan täsmällisesti fysiikan lakien mukainen. Price toteaa, että jotkut käyttävät sitä silti, koska Blend-säätimen avulla maskia voi säätää helpommin kuin varsinaista Fresneliä. (Price 2016b.)

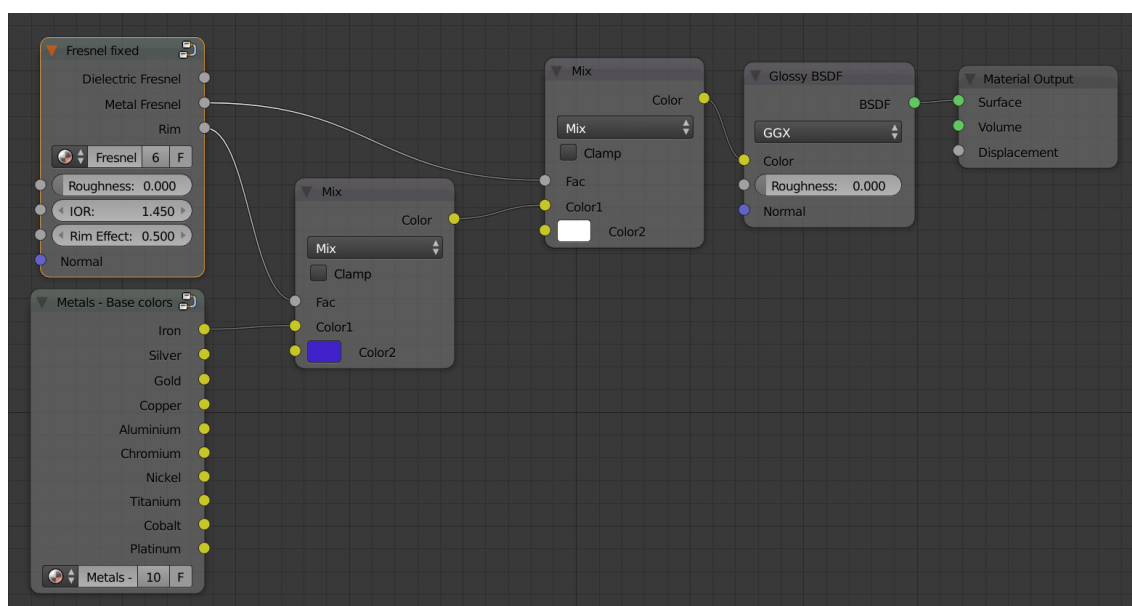
Tässä yhteydessä ei kuitenkaan käytetä tätä toisenlaista Fresnel-maskia, vaan nimenomaan Facing-maskia. Facing-maskissa kameraa tai katsojaa kohti olevat alueet ovat mustia ja mitä kauemmas kamerasuunnasta liikutaan, sitä vaaleammaksi maski muuttuu (kuva 23). Price lisää Facing-maskin Fresnel-ryhmään siksi, että sen avulla metallimateriaalin heijastesävy levittäytyy laajemmalle alueelle, mutta suoraan edessä olevissa kohdissa metallin perussävy näkyy sellaisenaan. (Price 2016b.)



Kuva 23: Maskien vertailua. Layer Weight -solmun Fresnel käyttäytyy lähes identtisesti aidon kanssa normaalitilanteessa, mutta vähennyslaskutulosta sillä ei aivan pysty tuottamaan. Facing-maski taas poikkeaa selvästi Fresnel-maskeista (Kuva: Harri Pietiläinen).



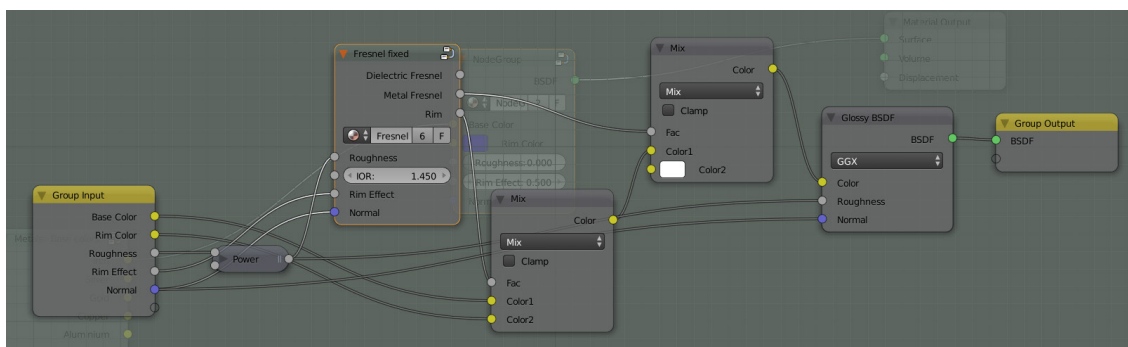
Koska Facing-maskin muutos Blend-arvosta 0.0 arvoon 0.1 on varsin huomatta-  
va, Price lisää sen perään aiempaan tapaan Math-solmun, joka korottaa maskin  
laskennalliset arvot toiseen potenssiin ja tasaa siten muutosta koko arvoalueella.  
Blend-säätimen Price ulkoistaa ryhmän syötteeksi, jolloin heijastesävyn laajuutta  
voidaan helposti säätää suoraan ryhmäsolmusta. Säätimen nimeksi annetaan  
Rim Effect. Fresnel-ryhmään lisätään myös uusi tulosteportti Rim, jonka kaut-  
ta Fresnel-ryhmä kytketään sekoitustekijäksi heijastussävyn määrittävään Mix-  
RGB-solmuun (kuva 24). (Price 2016b.)



Kuva 24: Fresnel-ryhmän uusi Rim-portti kytketään ensimmäiseen MixRGB-solmuun, jonka Color1-porttiin kytketään metallin perussävy, ja Color2 puolestaan määrittää heijastesävyn. Fresnel-ryhmän Rim Effect -arvolla määritetään, miten laajalle alueelle heijastussävy sekoitetaan (Kuvakaappaus: Harri Pietiläinen).

Lopuksi aletaan kasata PBR-metallimateriaalia yhdeksi ryhmäksi. Ryhmään sijoitetaan paranneltu Fresnel-ryhmä, MixRGB-solmut sekä Glossy-solmu (kuva 25). Jotta uutta PBR-materiaalia voisi säätää helposti, ryhmästä ulkoistetaan useita arvoja ryhmän syötteiksi. Price antaa syönteille nimet Base Color (eli metallin perussävy), Rim Color (heijastesävy), Roughness (pinnan karkeus, joka kytketään sekä Fresneliin että Glossy-solmuun), Rim Effect (heijastesävyn laajuus) sekä Normal (pinnan normaalit, kytketään sekä Fresneliin että Glossy-solmuun). Kuten edellä PBR-eristeen kohdalla myös tässä Roughness-arvoa tasoitetaan

Math-solmulla korottamalla arvot toiseen potenssiin. Lopuksi ryhmälle annetaan nimi PBR Metal. (Price 2016b.)



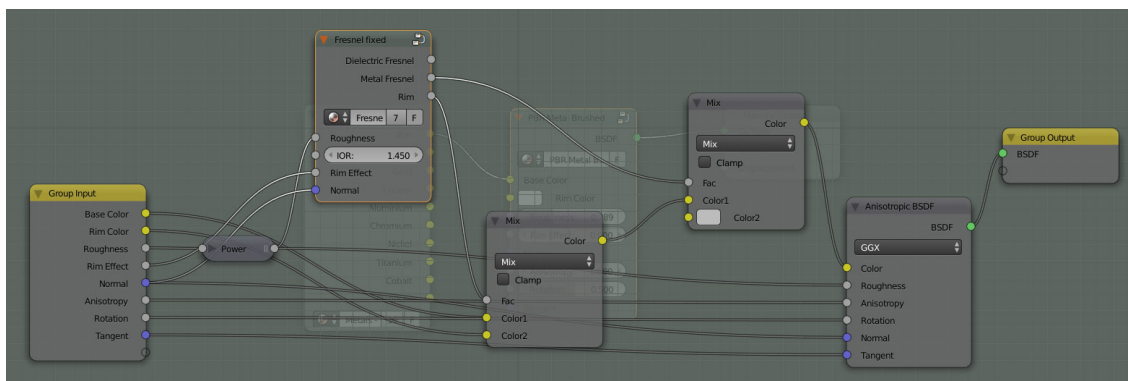
Kuva 25: Lopullinen PBR Metal -materiaali, josta ulkoisesta syötteiksi tietyt säätimet ja portit (Kuvakaappaus: Harri Pietiläinen).

PBR-metallimateriaalia voidaan käyttää kaikissa sellaisissa kappaleissa, jotka ovat käsittelemätöntä metallia. Materiaali sopii siis hyvin esimerkiksi teräkseen ja messinkiin, mutta valetun messingin kohdalla voidaan hyödyntää samaa pinta-kuviota kuin aiemman maalatun valuraudan kohdalla. Sopivaa Noise-tekstuuria käyttäen pinta näyttää karhealta ja heijastukset epämääräisemmiltä.

Rosterista eli harjatusta teräksestä valmistetut kohteet käyttäytyvät kuitenkin eri tavalla kuin tavallinen metalli. Tällaisissa pinnoissa on mikroskooppisen pieniä yhdensuuntaisia uria, joista valo heijastuu anisotrooppisesti eli valon tulokulma ja toisaalta katselukulma vaikuttavat valon heijastumiseen. Uraa vasten poikittain osuva valonsäde heijastuu eri tavalla kuin uran suunnassa kulkeva valo. Siksi tällaisiin kappaleisiin ei voi käyttää suoraan tätä PBR-metallia, mutta materiaalista voidaan helposti tehdä uusi toimiva versio.

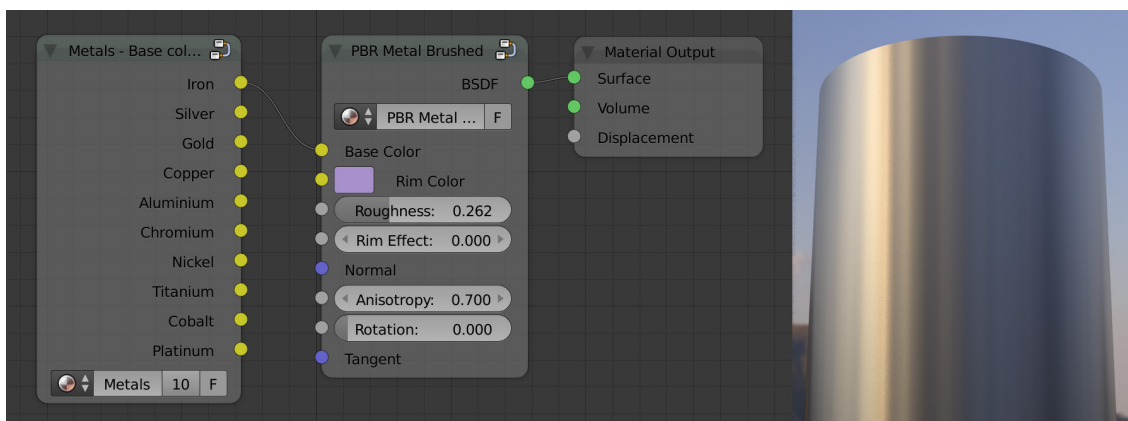
PBR-metallista tehdään uusi kopio, jolle annetaan nimeksi PBR Metal Brushed (kuva 26). Uudesta materiaalista vaihdetaan Glossy-solmu Anisotropic-solmuun, joka voidaan määrittellä toimimaan rosteripinnan lailla. Anisotropic-solmussa on samat perussäädöt kuin Glossy-solmussa, mutta lisäksi siinä on kolme muutakin. Anisotropy määrittelee sen, miten voimakkaasti pinnan mikrotasolla olevat urat

vaikuttavat valon heijastumiseen. Rotation taas määrittää urien kulkusuunnan. (Blender Foundation 2016.)



Kuva 26: PBR-metallin Glossy-solmu korvataan Anisotropic-solmulla, joka käyttäytyy kuin rosteripinta. Uuteen solmuun liittyvät säätimet ulkoistetaan ryhmän syötteiksi (Kuvakaappaus: Harri Pietiläinen).

Kolmas lisäys Glossy-solmuun verrattuna on Tangent-syöteportti, jonka kautta solmulle voidaan haluttaessa määrittellä erityisellä Tangent-solmulla, minkä akselin (x, y, z) ympäri anisotropia kappaleessa ilmenee. Tangent-solmulla voi myös määrittellä monimutkaisemman, kappaleen UV-karttaan pohjautuvan heijastumistavan. Jos Tangent-solmua ei ole määritetty, käytetään oletusarvoa (kappaleen z-akseli). (Blender Foundation 2016.)



Kuva 27: PBR-rosterin asetukset sekä niiden vaikutus sylinterimäiseen kappaleeseen, kun anisotropia-arvo on 0.7 ja pinnan karkeus on 0.262 (Kuvakaappaus: Harri Pietiläinen).

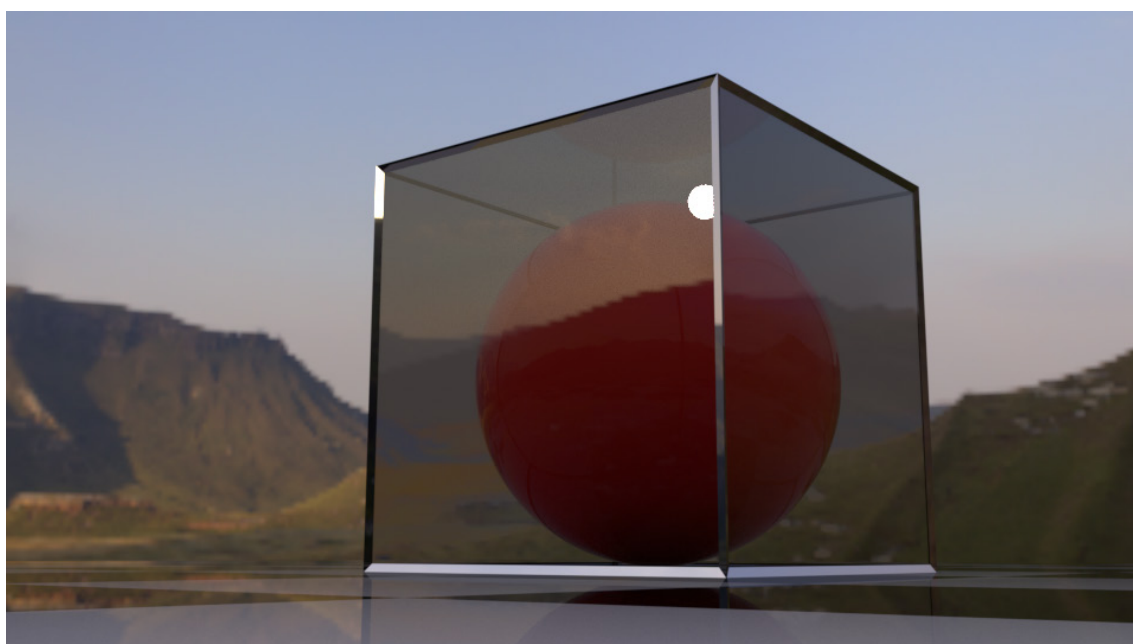
Anisotropia ilmenee kappaleen pinnassa vain, jos pinnan karkeusarvo on nollaa suurempi: täysin sileässä pinnassa kun ei voi olla myöskään yhdensuuntaisia uria. Kuvasta 27 huomataan, että anisotropia ilmenee selkeimmin, kun kirkas

valo osuu kappaleen pintaan, sillä karkeassa pinnassa muunlaiset heikommat heijastukset eivät juurikaan erotu.

Todellisuudessa kaikissa metallipinnoissa on lähes poikkeuksetta jonkin verran likaa, nokea, valumis- ja kulumisjälkiä sekä mahdollisesti jopa ruostetta. Eristeiden tapaan metallipintoja saadaan aidomman oloiseksi sillä, että niihin yhdistetään siirtokuvatekstuureja, mikä toimii hyvin varsinkin valumisjälkien ja ruosteen kohdalla erinomaisesti. Nokea taas voi lisätä myös proseduraalisilla tekstuureilla.

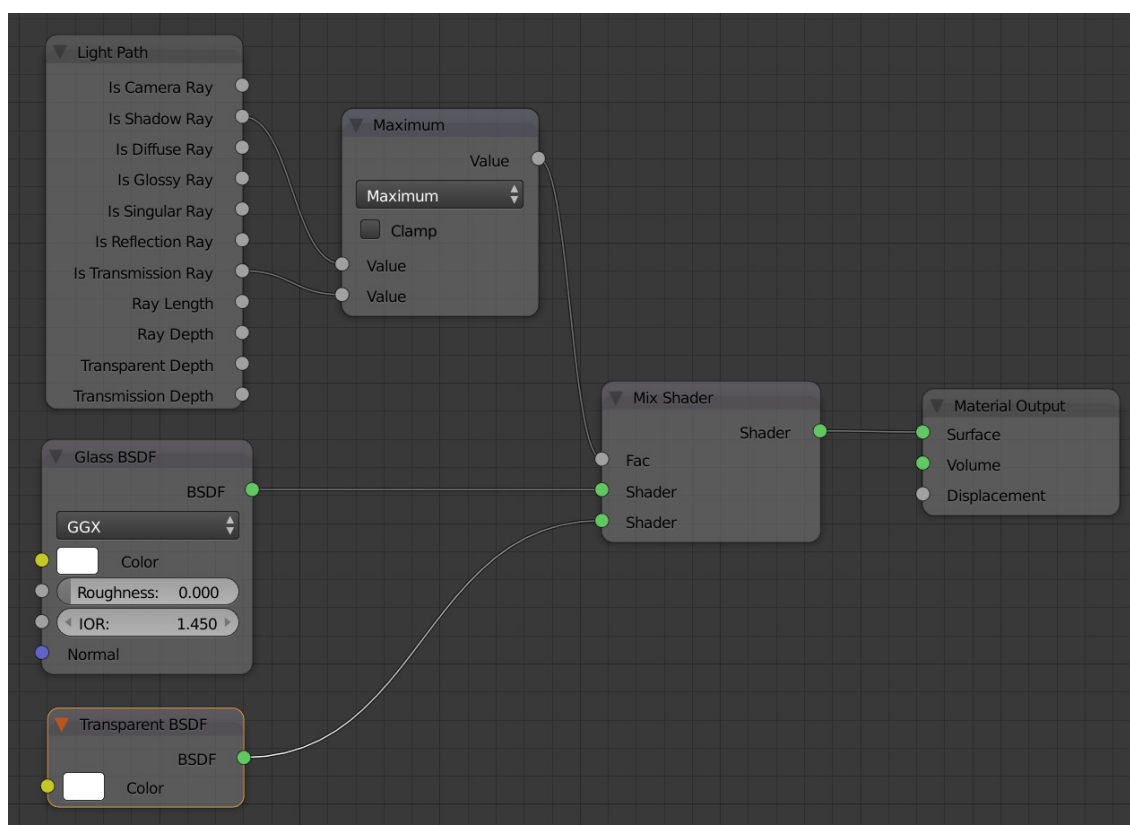
#### 4.2.6 PBR-lasi

Lasi on materiaalina siinä mielessä helpohko, että sille löytyy Cyclesista valmis sävytinsolmu (Glass BSDF). Cyclesin lasimateriaalissa on tosin tietyt heikkoutensa, sillä sitä ei ole suunniteltu läpäisemään valoa (kuva 28). Näin ollen erilaiset valoheijastukset ja -kuviot (kaustiikka) toimivat Cyclesissa huonosti. Siksi kaustiikkaa vaativat visualisoinnit kannattaa yleensä renderöidä jollain toisella moottorilla, joka soveltuu tällaiseen paremmin.



Kuva 28: Cyclesin lasimateriaali ei oletusarvoisesti läpäise valoa, ja siksi lasilaatikon sisällä oleva kappale näyttää tummemmalta kuin pitäisi (Kuva: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

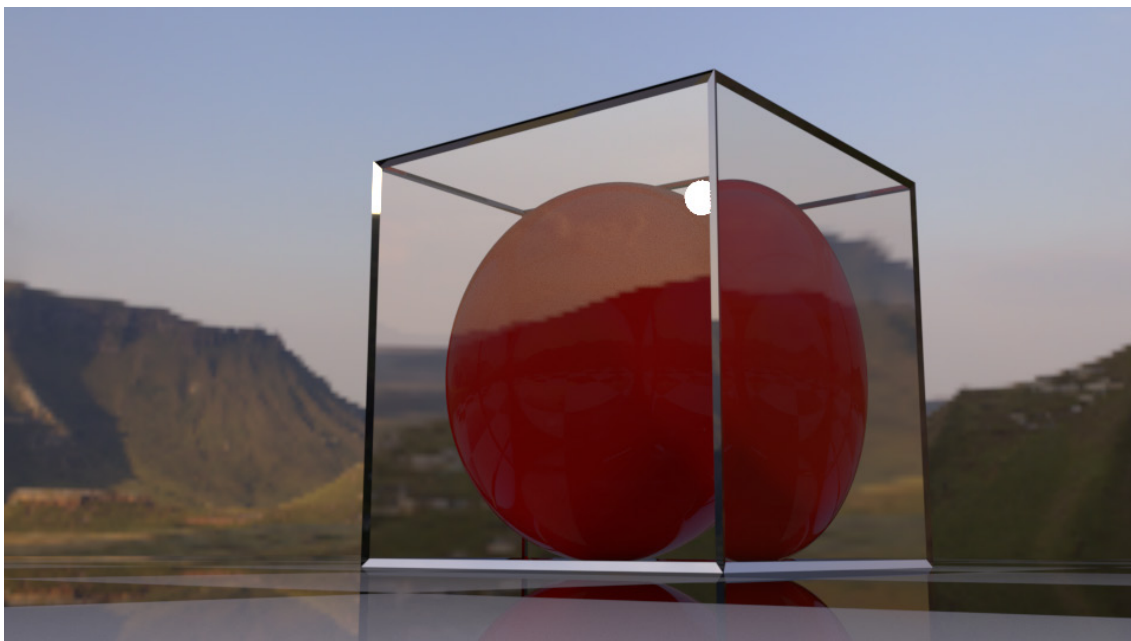
Jos visualisoinnissa tarvitaan lasia, jonka läpi valo kulkee luonnollisesti, on Cycles-lasia muokattava hieman paremmaksi. Cyclesista löytyy Light Path -syötesolmu, jonka avulla on mahdollista erotella Cyclesin säteenjäljitystekniikassa eri tavoilla kulkevat valonsäteet tyypin perusteella. Sen avulla voidaan erotella eri säteistä erotella kaksi tyyppiä: Shadow Ray (varjoja tuottava säde) sekä Transmission Ray (kappaleen läpi kulkeva säde). (Blender Foundation 2016: Nodes / Input / Light Path.)



Kuva 29: Valo saadaan kulkemaan Cycles-lasin läpi kappaleen sisään, kun tiettyjen säteenjäljityksessä käytettävien valonsäteiden kohdalla sävyttimiksi vaihdetaan lasimateriaalin sijaan läpinäkyvä materiaali (Kuvakaappaus: Harri Pietiläinen).

Näiden säteiden perusteella lasimateriaalin sävytinsolmuksi valitaan joko normaali Glass BSDF tai täysin läpinäkyvä Transparent BSDF (kuva 29). Kyseiset solmut yhdistetään Mix Shader -solmulla ja sekoitussuhde pitää määritellä siten, että vain jompikumpi on käytössä. Light Path -solmu antaa kunkin sädetyypin toteutuessa arvon 1 ja muutoin arvon 0, joten kun edellä mainittujen sädetyypien totuusarvoista otetaan maksimi Math-solmulla, arvo on aina joko 0 (kumpikaan ei pidä paikkaansa) tai 1 (jompikumpi tai molemmat pitävät paikkansa). Kun

Math-solmu kytketään edelleen Mix Shader -solmuun, jonka ylemmässä Shader-portissa on Glass ja alemmassa Transparent, kyseisten säteiden kohdalla lasi määritelläänkin siis läpinäkyväksi. Näin lasi näyttää luonnollisemmalta, kun valo pääsee kulkemaan materiaalin läpi esteettä (kuva 30). (Blender Foundation 2016: Nodes / Shaders / Mix.)



Kuva 30: Määritellyllä solmuasetelmalla valo pääsee lasilaatikon sisään ja muutoksen seurauksena laatikko myös taittaa valoa eri tavalla kuin ennen (Kuva: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

#### 4.2.7 Tekstuuriin kehittäminen Cyclesissa

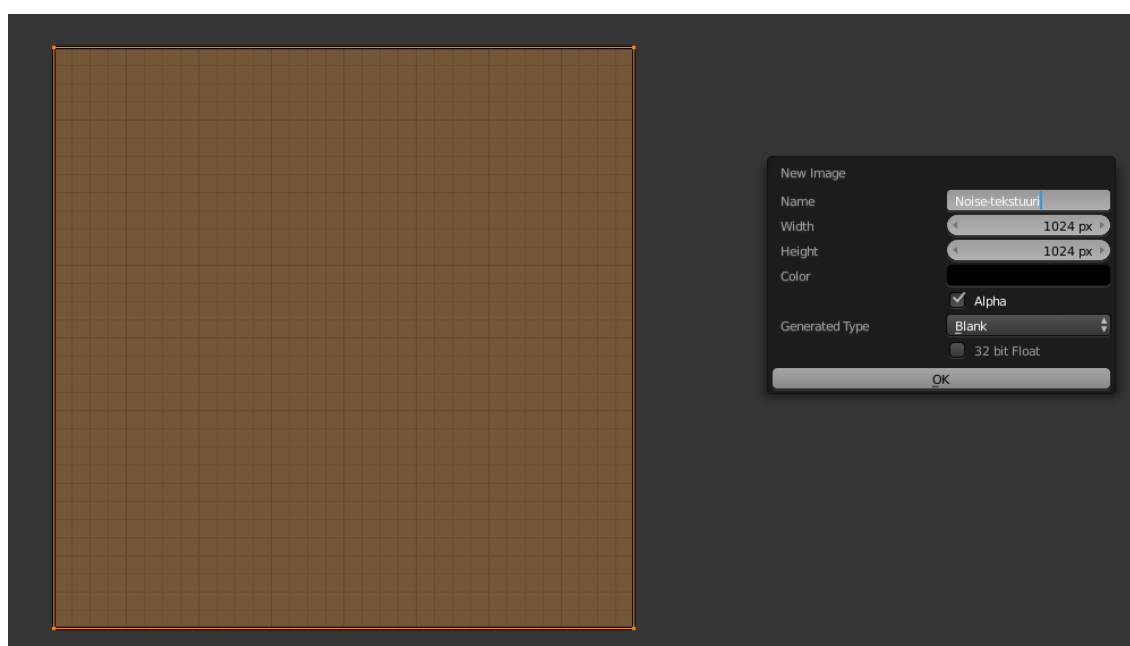
Koska Cyclesilla voi rakentaa varsin uskottavan näköisiä PBR-materiaaleja, sitä kannattaa hyödyntää myös BI- ja Blend4Web-moottorien yhteydessä. Cyclesilla voi nimittäin kehittää (bake) tekstuureita sellaisistakin materiaaleista, joiden määrittämiseen ei BI:ssä tai Blend4Webissä ole valmiita tai helppoja välineitä. Kehittäminen tarkoittaa sitä, että kappaleen pintarakenteen tai esimerkiksi ympäristön perusteella renderöidään tekstuureja, joka voidaan sitten määrittää kappaleen pintamateriaaliksi.

Samalla tavalla voidaan kehittää tavallisten kuvatekstuuriin lisäksi myös esimerkiksi normaalikarttoja, valo- ja varjokarttoja (valoheijastukset ja varjot suoraan

kuvatekstuurissa), kiiltokarttoja (Glossy) ja ympäristövarjostuskarttoja (AO / Ambient Occlusion) (Blender Foundation 2016: Baking). Näiden avulla voidaan Bl:n ja Blend4Webin materiaaleja parannella lähemmäs aitoja PBR-materiaaleja.

Koska Blend4Webin nykyinen versio (16.7.0) ei vielä tue esimerkiksi proseduraalisia tekstuureita, Cyclesissa voidaan kehittää vaikkapa aiemmin moottorilohkon pinnassa käytetystä Noise-tekstuurista sopiva normaalikartta tai kohoumakartta käytettäväksi Blend4Webin puolella. Se toimikoon tässä yhtenä esimerkkinä tekstuurien kehittämisestä, sillä kaikkea tekstuurien kehittämiseen liittyviä asioita ei kannata tämän työn puitteissa käydä läpi.

Teksturien kehittäminen edellyttää, että kappaleesta tehdään ensin UV-kartta, mikä tarkoittaa kappaleen purkamista kaksiulotteiseksi osakaavaksi. Varsinkin monimutkaisempia kappaleita joutuu hieman käsittelemään, jotta niistä saisi toimivia UV-karttoja, muun muassa määrittelemällä jotkut reunat saumoiksi. Tällöin saumoilla erotellut osat leikellään UV-kartassa erilleen saumojen mukaan – samaan tapaan kuin vaatteiden osat kaava-arkille. UV-karttoja käsitellään Blenderin omassa UV-editorissa.

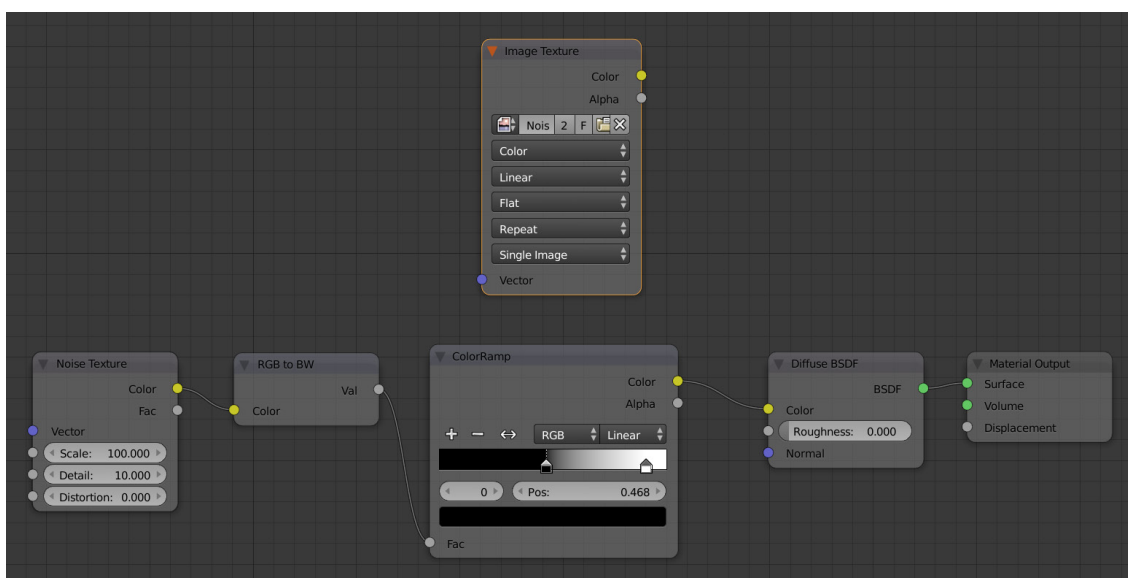


Kuva 31: Yksinkertainen taso on purettu UV-kartaksi UV-editoriin. Seuraavaksi sille luodaan uusi teksturi, jolle annetaan nimeksi Noise-teksturi, ja kooksi määritellään 1024x1024 pikseliä (Kuvakaappaus: Harri Pietiläinen).



Kun kappaleen UV-kartta on valmis, UV-kartan pohjalta luodaan uusi tekstuuri, joka edelleen talletetaan koneen kovalevylle (kuva 31). Tekstuurille tulee määrittellä koko pikseleissä, joka määritellään yleensä kahden potenssina (256x256, 512x512, 1024x1024 jne.). Tällöin näytönohjain (grafiikkakortti/GPU) pystyy käsittelemään tekstuuria mahdollisimman tehokkaasti. Tekstuurin koko valitaan käyttötarkoituksen mukaan. Lähikuvissa tarvitaan yleensä tarkempia tekstureja kuin taustakappaleissa. Käytännön maksimikoko on 4096x4096, sillä sitä suuremmat vievät nykykorteilla jo liikaa muistia ja ovat siksi raskaita ja hitaita käsitellä.

Kun tekstuuri on tallennettu levylle, siirrytään Node Editoriin. Kappaleelle määritellyn PBR-materiaalin rinnalle lisätään uusi kuvatekstuuri (Image Texture), johon ladataan edellä luotu tekstuuripohja. Tämä solmu jätetään irralliseksi Node Editoriin, mutta sen tulee olla aktiivisena solmuna (kuva 32). Seuraavaksi Cycle-sin renderöintiasetusten Bake-osiossa määritellään se, mitä halutaan kehittää.

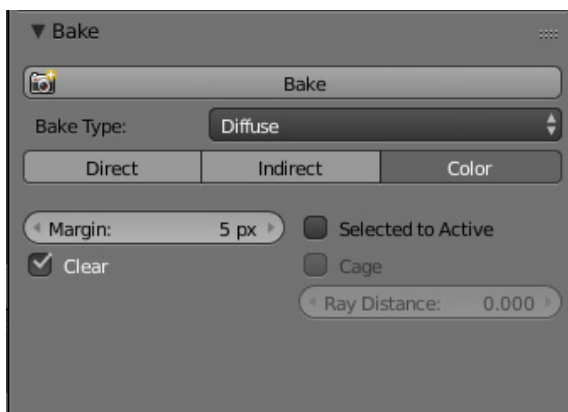


Kuva 32: Node Editorissa käytetään pohjana Noise-tekstuuria, joka muokataan ensin mustavalkoiseksi ja sen jälkeen kontrastia lisätään ColorRamp-solmulla (Kuvakaappaus: Harri Pietiläinen).

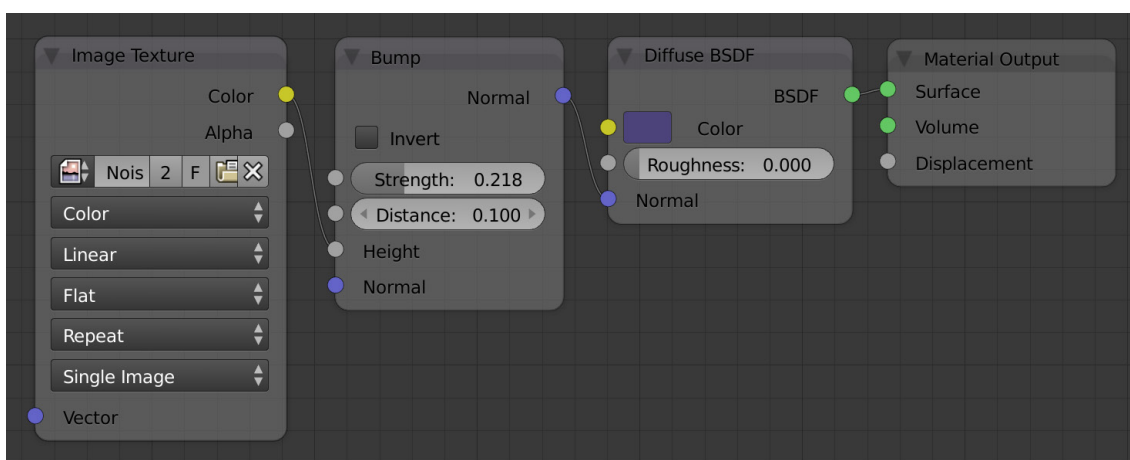
Bake Type on oletuksena Combined, jolloin valintalaatikon alapuolelta voidaan valita rastittamalla ne tekijät, jotka halutaan tekstuuriin mukaan. Jos esimerkiksi halutaan tehdä mustavalkoinen kohoumakartta, riittää kun rasti jää kohtaan Diffuse, tai valita suoraan Bake Typeksi Diffuse (kuva 33). Jälkimmäisen kohdal-



la riittää kun tarkenteeksi valitaan vielä Color (Direct ja Indirect kannattaa ottaa pois). Kun asetukset ovat valmiina, painetaan Bake-painiketta ja jäädään odottamaan tekstuurin valmistumista. Mitä suurempi ja monimutkaisempi kartta on, sitä pidempään sen valmistuminen kestää. Valmis kartta näkyy lopulta UV-editorissa, mistä se täytyy kehittämisen jälkeen tallentaa uudelleen levyille. (Blender Foundation 2016: Baking.)

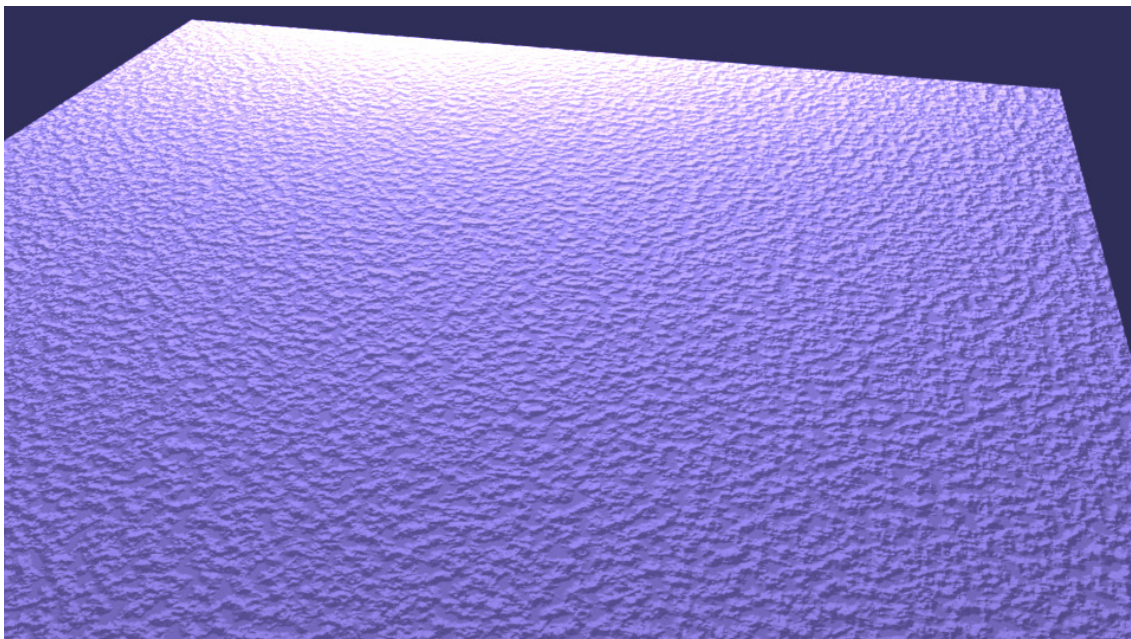


Kuva 33: Mustavalkoista kohoumakaartaa kehitettäessä renderointiasetusten Bake-osiossa valitaan Bake Typeksi Diffuse ja sen alla olevista tarkenteista Direct ja Indirect otetaan pois päältä (vaaleamman harmaat) (Kuvakaappaus: Harri Pietiläinen).



Kuva 34: Uutta Noise-tekstuuria käytetään kohoumakaartana kytkemällä se Bump-solmun Height-porttiin, sillä mustavalkoväriä käytetään korkeustietona. Bump-solmu kytketään edelleen Diffuse-solmun Normal-porttiin, jolloin Cycles osaa muokata pinnan normaaleja tekstuurin perusteella (Kuva: Harri Pietiläinen).

Luodun tekstuurin voi testata Cyclesissa helposti siten, että se ladataan Image Texture -solmuun joka kytketään Bump-solmuun ja tämä edelleen esimerkiksi Diffuse-solmun Normal-porttiin (kuva 34). Tällä kerrotaan Cyclesille, että tekstuuria käytetään kohoumakaartana ja se vaikuttaa pinnan normaaleihin. Renderöinnissä pinnan pitäisi olla tekstuurin mukaisesti epätasainen (kuva 35).



Kuva 35: Edellä kuvatuilla asetuksilla renderöity pinta näyttää epätasaiselta, eli tekstuuri toimii kutakuinkin halutulla tavalla. Tekstuurin voimakkuutta voi säätää helposti haluttaessa Bump-solmusta. (Kuva: Harri Pietiläinen).

### 4.3 Blender Internal

#### 4.3.1 Taustaa lyhyesti

Blenderin alkuperäinen renderöintimoottori Blender Internal (BI) on perinteistä tyyppiä oleva moottori, joka toimii samaan tapaan kuin monet pelimoottorit aiemmin. Simonds toteaa, että BI:ssa tavoitteena on ollut renderöinnin nopeus ja joustavuus eikä fyysisten ominaisuuksien tarkka mallintaminen, mutta sillä pystyy silti tuottamaan hyvin realistisen näköisiä malleja (Simonds 2013, 183). Koska realismi saadaan aikaan monenlaisilla tempuilla ja nikseillä, materiaaleja voi joutua muokkaamaan valaistusolosuhteiden mukaan.

Sen jälkeen kun Blenderin uudempi renderöintimoottori Cycles julkaistiin, vanhemman moottorin käyttö näytti yhteisössä hiipuvan. Tämä näkyy muun muassa siinä, että esimerkiksi YouTubesta löytyy runsaasti opetusvideoita Cyclesin käytöstä, mutta hyvin vähän enää BI:sta ja nekin ovat enimmäkseen jo vuosien takaa. Ehkä juuri siksi Blender Foundation on ollut aikeissa korvata BI-moottorin ja Blenderin oman pelimoottorin Blender Game Enginen (BGE) kokonaan uudella

uutta PBR-laatusella moottorilla. Toteutusta suunniteltiin täksi vuodeksi (2016), mutta tilanne on tätä kirjoittaessa vielä auki. (Blender Foundation 2015.)

Vaikka BI onkin Blenderin kahdesta renderöintimoottorista vanhempi, se on tämän työn kannalta olennainen, koska Blend4Webissä materiaalit määritellään osittain samalla tavalla. BI:ssä materiaalit on määritelty tavallisesti materiaalipaneelin kautta, mutta käytettävissä on nykyisin myös materiaalisolmuja. Cyclesiin verrattuna BI:n solmuvalikoima on kuitenkin huomattavasti suppeampi. Blend4Webissä näitä materiaalisolmuja hyödynnetään paljon enemmän, ja siihen on myös lisätty uusia erikoissolmuja monipuolisempien materiaalien tarpeisiin.

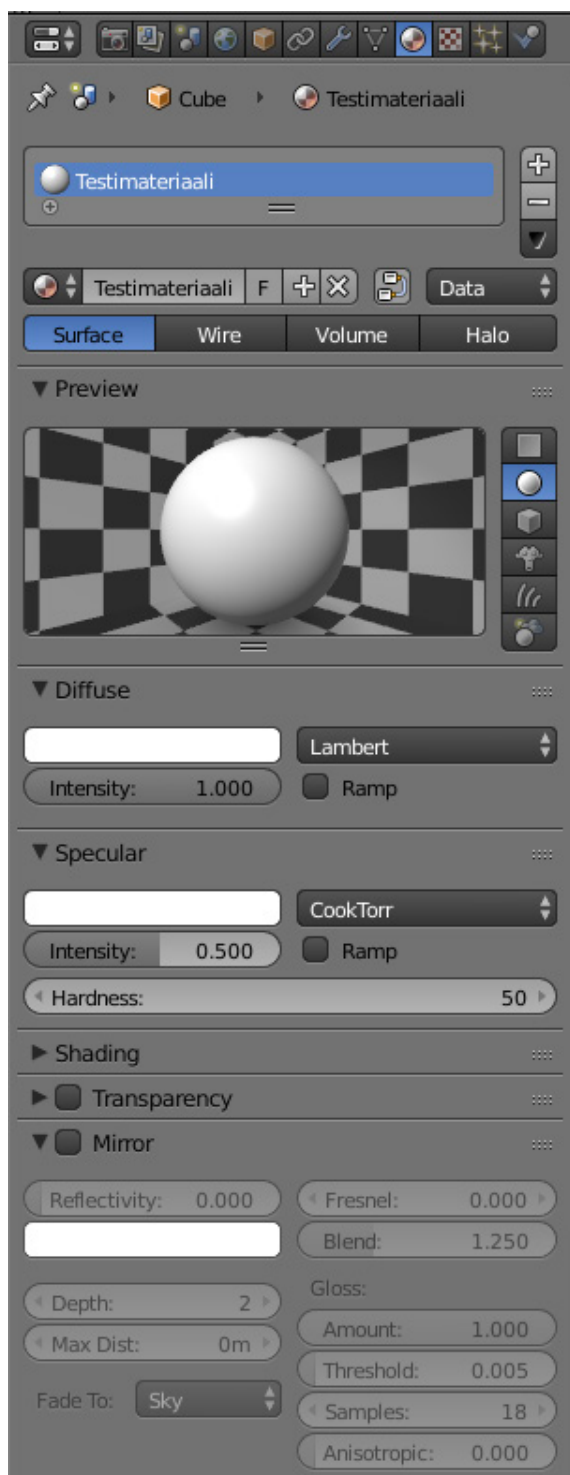
#### **4.3.2 BI – Materiaalit**

Blender Internal -renderöintimoottorissa materiaalit voi määritellä joko materiaalipaneelin kautta tai Cyclesista tutulla tavalla Node Editorissa solmujen avulla. BI:n solmuvalikoima tosin on selkeästi suppeampi kuin Cyclesissa, eikä materiaalien määrittämäinen paneelien kautta ole aivan yhtä helppoa ja sujuvaa kuin Cyclesin solmupohjainen lähestymistapa. Kaikkia säätöjä ei voi myöskään tehdä materiaalisolmuista käsin, vaan ne on säädettävä paneelistä.

PBR-eristeissä keskeisiä ominaisuuksia ovat diffuusiväri (sironta) ja kiiltävyys (heijastuminen). Näiden tulee olla toisensa poissulkevia eli kiiltävyyden lisääntyessä diffuusivärin osuus vähenee. Toisaalta mitä sileämpää materiaali on mikrotasolla, sitä tarkempina ja pienemmällä alueella valoheijastukset näkyvät ja sitä voimakkaampia ne ovat intensiteetiltään. Valon kokonaisintensiteetin täytyy siis pysyä heijastusalueella vakiona. PBR-materiaaleissa noudatetaan energian säilyvyyslakia ja siksi edellä mainittujen ominaisuuksien tulisi muuttua aina suhteessa toisiinsa.

BI:ssa tilanne on kuitenkin sikäli mielenkiintoinen, että kaikki edellä mainitut säädöt kyllä löytyvät siitä, mutta toimivat täysin toisistaan riippumatta (kuva 36).

Esimerkiksi valoheijasteet (diffuusiheijasteet, BI-terminologiassa Specular) on erotettu peilimäisistä heijastuksista (Mirror). Lisäksi valoheijasteiden laajuutta säädellään CookTorr-sävytintä käytettäessä kovuusarvolla (Hardness), ja peiliheijasteissa taas säädetään pinnan karkeutta ja samalla heijastusten terävyyttä Gloss-säätimellä. Kaikkien osatekijöiden saaminen kohdilleen voi olla melko työlästä verrattuna Cyclesiin.

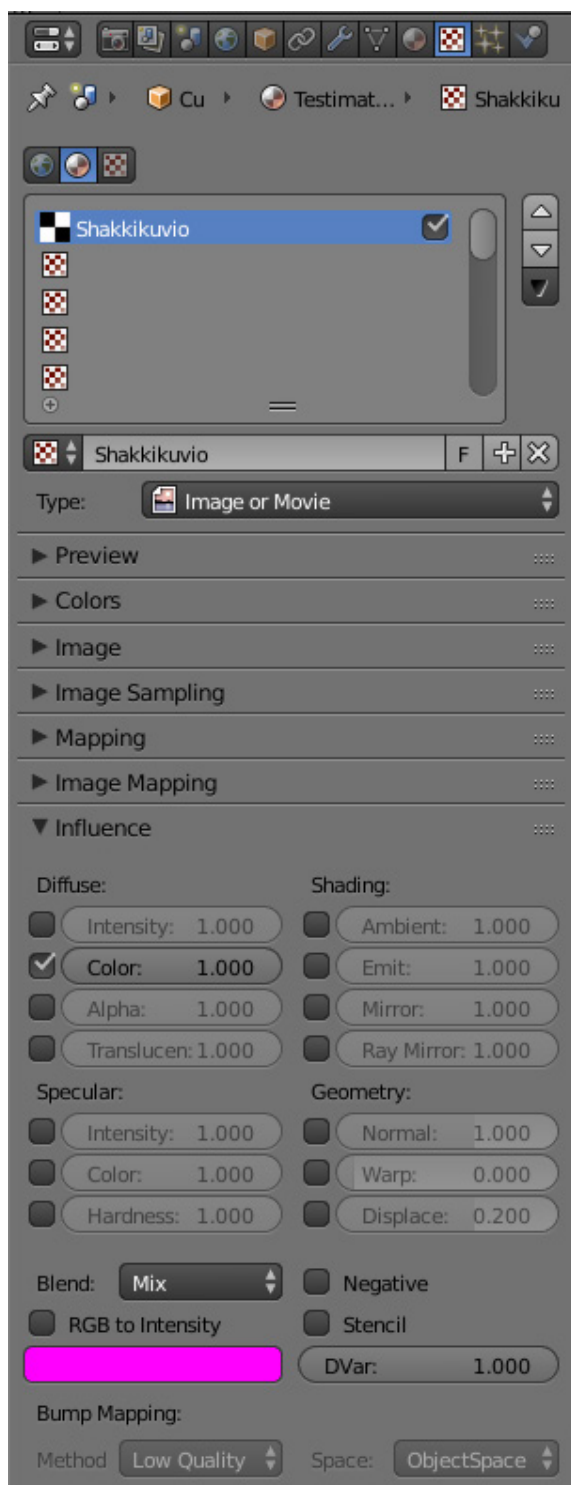


Säätöjen erillisyydestä johtuen kapale voi periaatteessa heijastaa ympäristönsä muutoin täydellisesti, mutta ilman valoheijasteita, mikä ei ole lainkaan ominaista PBR-materiaaleille. Toisaalta Simonds huomauttaa, että monesti pelkkä valoheijastusten renderöinti voi riittää, mikäli kappaleen ei ole edes tarkoitus heijastaa ympäristöään – peiliheijastukset kun vaativat monimutkaisempaa säteenjäljitystekniikkaa, mikä puolestaan hidastaa renderöintiä (Simonds 2013, 184).

BI:lla on mahdollista rakentaa melko uskottavia materiaaleja, kun muistaa muutaman olennaisen asian. Diffuu-

Kuva 36: Blender Internal -moottorin materiaalipaneelista määritellään materiaalien perusominaisuudet kuten diffuusiväri ja kiiltävyys. Säädöt ovat täysin riippumattomia toisistaan (Kuvakaappaus: Harri Pietiläinen).

sivärin määrää tulee säätää käsin pienemmäksi, mikäli valoheijastuksia lisätään, ja toisaalta valoheijastusten intensiteettiä täytyy lisätä, jos materiaalin kovuus lisääntyy ja heijastusten koko pienenee. Simonds kehottaa pitämään huolta myös siitä, että diffuusivärin ja valoheijasteiden osuuksien summa on maksimissaan 1.0, mieluiten hieman alle (osa valosta imeytyy kappaleeseen), jotta kappale näyttäisi realistisemmalta (Simonds 2013, 209).



Blender Internal -moottorissa tekstuurit määritellään erikseen omissa asetuksissaan (kuva 37). Kappaleelle määritellään ensin materiaali, johon tekstuurit sitten kytketään. Samalle kappaleelle voi määritellä monia erilaisia tekstuureja ja kullekin tekstuurille edelleen sen, millä tavalla se kappaleeseen vaikuttaa. Tekstuuripaneelin Influence-osiossa tekstuurin vaikutus valitaan rastittamalla halutut kohdat ja säätämällä sitten vaikutuksen voimakkuutta liikusäätimellä. (Blender Foundation 2016: Blender\_render / Textures / Influence / Material.)

Kuva 37: Tekstuuripaneelissa määritellään materiaaliin kytkettävät kuvatekstuurit sekä niiden vaikutus materiaalin ulkonäköön tai kappaleen muotoon (Kuvakaappaus: Harri Pietiläinen).



Vaikutustyyppejä on neljä: diffuusiväri (Diffuse), valoheijasteet (Specular), sävytys (Shading) sekä geometria (Geometry). Näistä kolme ensimmäistä ryhmää vaikuttavat samoihin materiaalin perussäätöihin, jotka löytyvät myös materiaali-paneelistä. Viimeinen taas muuttaa kappaleen geometriaa eli normaali-, kohouma- ja poikkeutuskartat käyttävät näitä viimeksi mainitun ryhmän säätimiä. (Blender Foundation 2016: Blender\_render / Textures / Influence / Material.)

Koska Blend4Web käyttää hyvin pitkälti samoja asetuksia ja säätimiä materiaalin määrittelyyn kuin Blender Internal -moottori, materiaalien määrittelyä tarkastellaan lähemmin seuraavassa Blend4Webiä käsittelevässä luvussa. Tässä on kuitenkin pari esimerkkiä BI:n materiaaleista.

Tarkastellaan ensin kiiltävää terästä. Cyclesissa teräsmateriaali koottiin puhtaasti Glossy-solmun pohjalta, koska metalleissa valo ei siroa eikä niillä siten ole diffuusiväriä lainkaan. Tämän perusteella BI:n määrittelyssä voidaan kokeilla vastaavaa ja asettaa diffuusivärin voimakkuus (intensity) kokonaan nolulle (kuva 38).



Glossy-solmua BI:n materiaaliasetuksissa vastaa lähimmin Mirror, joka säätää peilimäisiä heijastuksia. Kun Glossy-solmun Roughness-arvo säädetään nolleen, sekin tuottaa peilimäisen lopputuloksen. Mirror-kohdan laitetaan siis rasti ja heijastavuus (Reflectivity) arvoon 1. Väriarvonnaksi määritellään sama RGB-väri, jota Cyclesissa käytettiin raudalle.

Kuva 38: Kiiltäväpintainen teräsmateriaali määritellään BI-moottorissa samoilla periaatteilla kuin Cyclesin materiaalit (Kuvakaappaus: Harri Pietiläinen).

Fresnel-arvoksi laitetaan sama 1.45, jota käytettiin myös Cyclesissa. Bl:n puolella Fresnelin vaikutus näyttää olevan selvästi voimakkaampi kuin Cyclesissa, mutta valittu arvo tuottaa aika hyvän lopputuloksen. Blend määrittää sen, miten Fresnelin reunoja kirkastava vaikutus sekoittuu perusväriin. Kokeilujen perusteella päädyin käyttämään varsin pieniä arvoja, jotka näyttivät paremmalta. Tähän valitsin arvon 1.15.

Lopuksi määritellään vielä pinnan karkeus Gloss-säätimellä. Se vaikuttaa suoraan kappaleen pinnassa näkyvien ympäristöheijastusten selkeyteen. Mitä pienempi arvo, sitä sumeammaksi heijastukset muuttuvat. Gloss-arvo 1.0 tarkoittaa täysin puhdasta peiliä, arvot siis toimivat juuri päinvastoin kuin Cyclesissa. Koska kyse on teräksestä, Gloss-arvoa on syytä pudottaa hieman alemmas, sillä kirkaspintainen teräskään ei ole täysin peilimäinen. Arvo 0.95 näyttää aika hyvältä.

Viimeisenä määritellään vielä valoheijasteet (Specular), joka vastannee Cyclesissa luodun PBR-metallin reunaväriä. Väriksi valitaan hieman Mirror-osiossa määriteltyä perussävyä vaaleampi harmaa. Koska diffuusiväriä ei ole määritelty, Intensity-arvo saa olla lähellä yhtä, olkoon se vaikkapa 0.915. Teräs on materiaalina kovaa ja siksi valoheijastuksetkin ovat pieniä ja teräviä. Niinpä Hardness-arvoa voi nostaa reippaasti, tässä päädyin arvoon 400. Lopuksi vaihdan vielä oletussävyttimen (CookTorr) toiseen eli Blinniin.

Blender-manuaalin perusteella Blinn on ehkä lähinnä PBR-sävyttintä, sillä se tarjoaa myös IOR-säätimen, jota muissa vaihtoehdoissa ei ole. IOR-arvoksi kokeilin ensin samaa arvoa, jota käytettiin Fresnel-arvona, mutta se heikentää kohdevaloheijastusta liikaa. Suurempi arvo kirkasti kohdevalopistettä, joten arvoksi valikoitui lopulta teräksen IOR 2.95. Renderöitynä materiaali näyttää melko kelvolliselta (kuva 39). Jos teräksestä haluaa hieman mattapintaisempi, Mirror-osion Glossy-arvon voi pudottaa esimerkiksi arvoon 0.85. Tällöin pinta tosin näyttää renderöitynä suhteellisen karhealta verrattuna Cyclesin vastaavaan. (Blender Foundation 2016: Blender\_render / Materials / Specular Shaders.)



Kuva 39: Blender Internal -moottorille määritelty teräsmateriaali näyttää renderöitynä melko aidolta (Kuva: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

Kokonaan toisenlainen lähestymistapa materiaalien määrittämiseen on olemassa, ja se tapahtuu hyödyntämällä solmueditoria sekä valeheijastuksia. Tässä esimerkissä CG Cookie -sivuston kouluttaja Kent Trammell esittelee, miten heijastus saadaan aikaiseksi käyttämällä sopivia kuvatekstuureita.

Aluksi kappaleelle lisätään uusi materiaali, jolle annetaan nimeksi heijastus\_terävä, ja sille luodaan tekstuuripaneelissa samanniminen tekstuuri. Trammell huomauttaa, että BI:ssa kappaleet eivät kykene heijastamaan ympäristötekstuuria samalla tavalla kuin Cyclesin puolella, mutta materiaalitekstuurin avulla päästään lähes samalta näyttävään lopputulokseen. Kun sopiva kuvatekstuuri on valittu, tekstuuripaneelin Mapping-osiosta Coordinates- arvoksi vaihdetaan oletusarvo UV:n sijaan Reflection. Influence-osiosta tekstuurin vaikutus saa jäädä oletusarvoon Color. (Trammell 2015.)

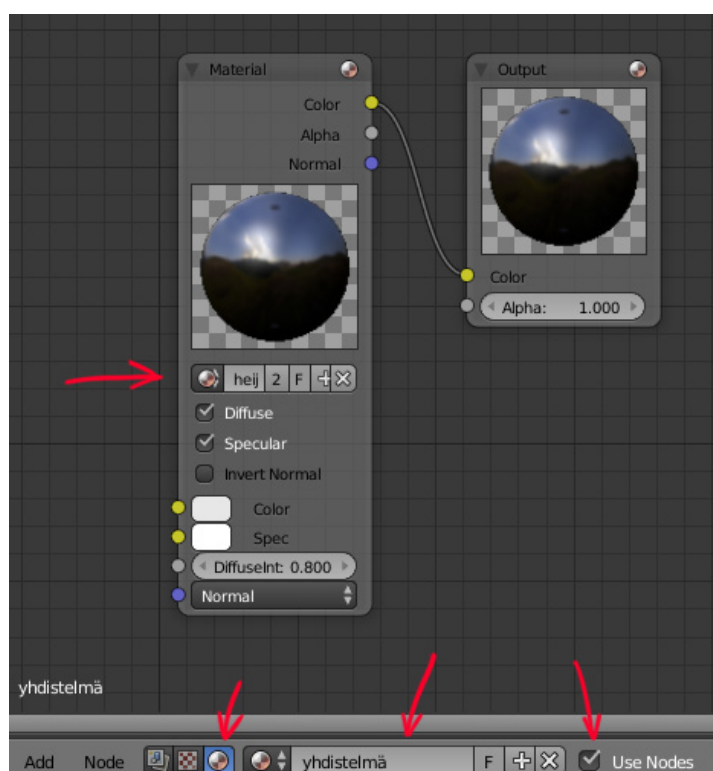
Kappaleelle määritellään tämän jälkeen toinen materiaali, heijastus\_sumea, ja sille vastaava tekstuuri. Tällä kertaa kuvatekstuuriksi valitaan edellä käytetystä kuvatiedostosta tehty sumennettu versio, mutta muutoin asetukset ovat täsmälleen samat. Lopullista ”PBR-tyyppistä” materiaalia varten on tarpeen luoda vielä yksi materiaali, jossa heijastukset yhdistetään. Uusi materiaali, nimeltään yhdis-



telmä, lisätään materiaalipaneelissa pinon ylimmäksi materiaaliksi ennen äskeisiä heijastusmateriaaleja. Tämä siksi, että Blenderissä materiaaleja käytetään aina järjestyksessä ylhäältä alaspäin. (Trammell 2015.)

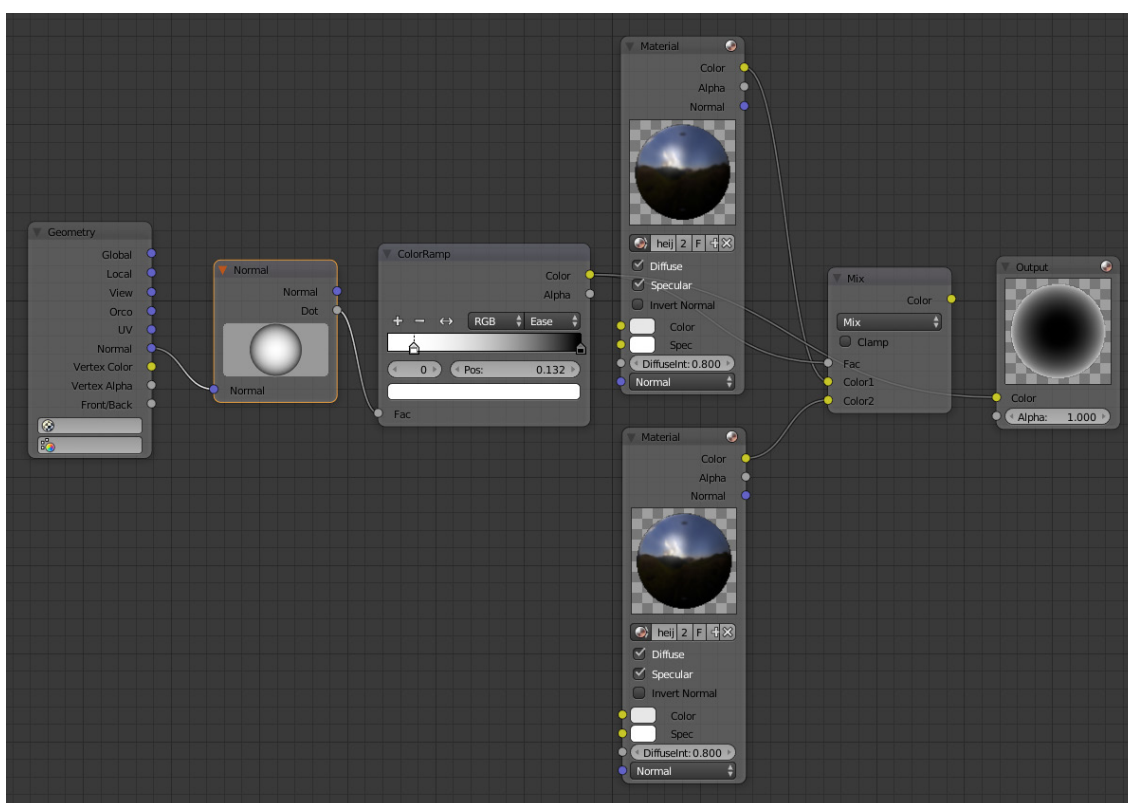
Vaikka materiaalipinossa järjestystä voi muuttaa laatikon oikealla puolella olevilla ylä- ja alanuolilla, näin muutettu järjestys ei näytä vaikuttavan käytettävään materiaaliin. Järjestystä on siis muutettava siten, että valitaan materiaalilaatikosta ensimmäinen materiaali ja vaihdetaan kyseisen rivin materiaali laatikon alla olevasta pudotusvalikosta. Sama operaatio suoritetaan muillekin materiaaleille, kunnes järjestys on halutun lainen.

Seuraavaksi siirrytään sitten Cyclesista tuttuun Node Editoriin, missä heijastusmateriaalit kootaan yhdistelmä-materiaaliksi (). Ensiksi materiaalipaneelista valitaan yhdistelmä-materiaali aktiiviseksi, Node Editorin alareunasta valitaan solmupuun tyyppi (ikonit) Shader Nodes ja lopuksi asetetaan rasti kohtaan Use Nodes. Näin syntyneen Material-solmuun linkitetään materiaaliksi aiemmin luotu heijastus\_sumea. Material-solmu kopioidaan ja kopioon linkitetään materiaaliksi heijastus\_terävä. (Trammell 2015.)



Kuva 40: Kun yhdistelmä-materiaali on valittuna (alh. kesk), Node Editorissa vaihdetaan solmupuutyyppi (alh. vas.) sävyttimet ja laitetaan rasti kohtaan Use Nodes. Uuteen materiaalisolmuun linkitetään sumeampi heijastusmateriaali (Kuvakaappaus: Harri Pietiläinen).

Materiaalisolmut on tarkoitus yhdistää Cyclesin tapaan siten, että heijastusten sekoitustekijänä käytetään Fresneliä. Koska BI:n puolella ei ole Fresnel-solmua, joudutaan taas turvautumaan pieneen kikkailuun. Sekoitusmaskin tuottamiseksi tarvitaan Geometry- ja Normal-solmut, joiden Normal-portit kytketään toisiinsa (kuva 41). Normal-solmun Dot-portti tuottaa lähes samanlaisen maskin kuin Fresnel. Normal-solmun perään lisätään vielä ColorRamp-solmu, johon Dot-portti kytketään. ColorRampin interpolointityypiksi vaihdetaan Ease oletusarvon Linear sijaan, ja lisäksi värien suunta käännetään siten, että valkoinen on vasemmalla ja musta oikealla, jolloin maskissa valkoinen väri jää reunoille ja musta on koh- ti kameraa. ColorRampin väriliukujen avulla voidaan maskia säätää halutuksi. (Trammell 2015.)

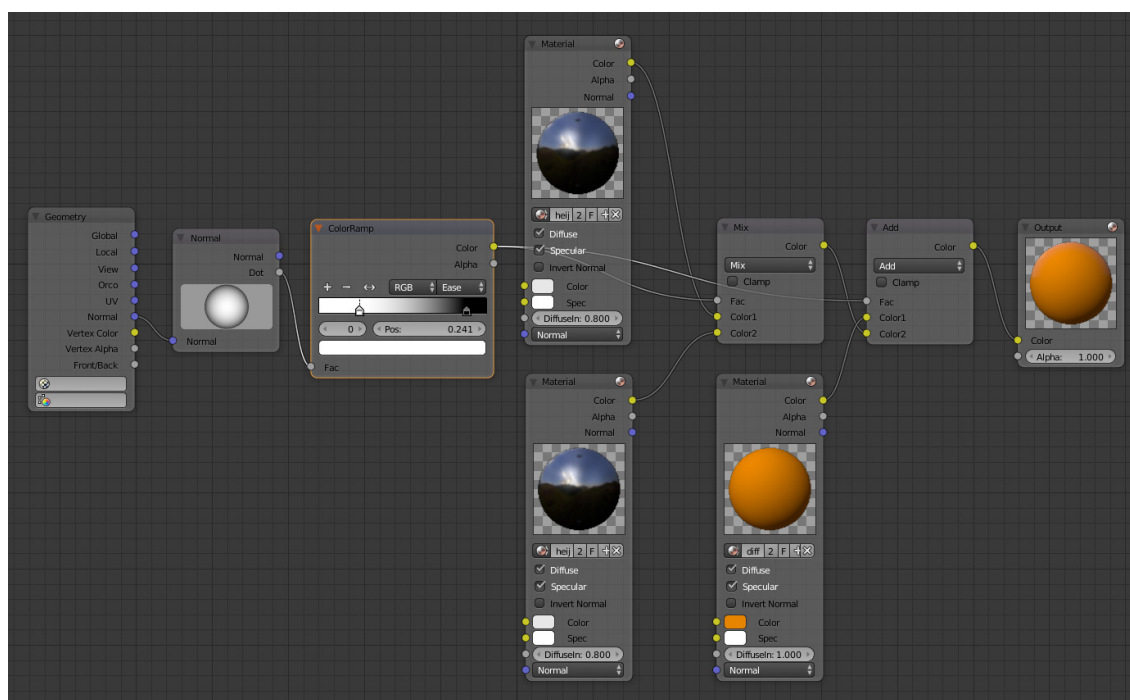


Kuva 41: Koska BI:ssa ei ole Fresnel-solmua, heijastusmateriaaleja varten luodaan vastaava sekoitusmaski Geometry-, Normal- ja ColorRamp-solmujen avulla. Kun niiden lopputulos kytketään suoraan Output-solmuun, nähdään että maski on hyvin samankaltainen kuin Cyclesin Fresnel-solmun tuottama maski aiemmin (Kuvakaappaus: Harri Pietiläinen).

Nyt materiaalisolmut yhdistetään toisiinsa lisäämällä mukaan MixRGB-solmu, jonka ylempään Color-porttiin kytketään sumeampi heijastusmateriaali ja alem-

paan terävä heijastusmateriaali. Sekoitustekijäksi Fac-porttiin kytketään Color-Ramp-solmu. Näin kappaleessa näkyy sumeampi heijastus kohti kameraa ja reunoilla heijastukset ovat terävämpiä. (Trammell 2015.)

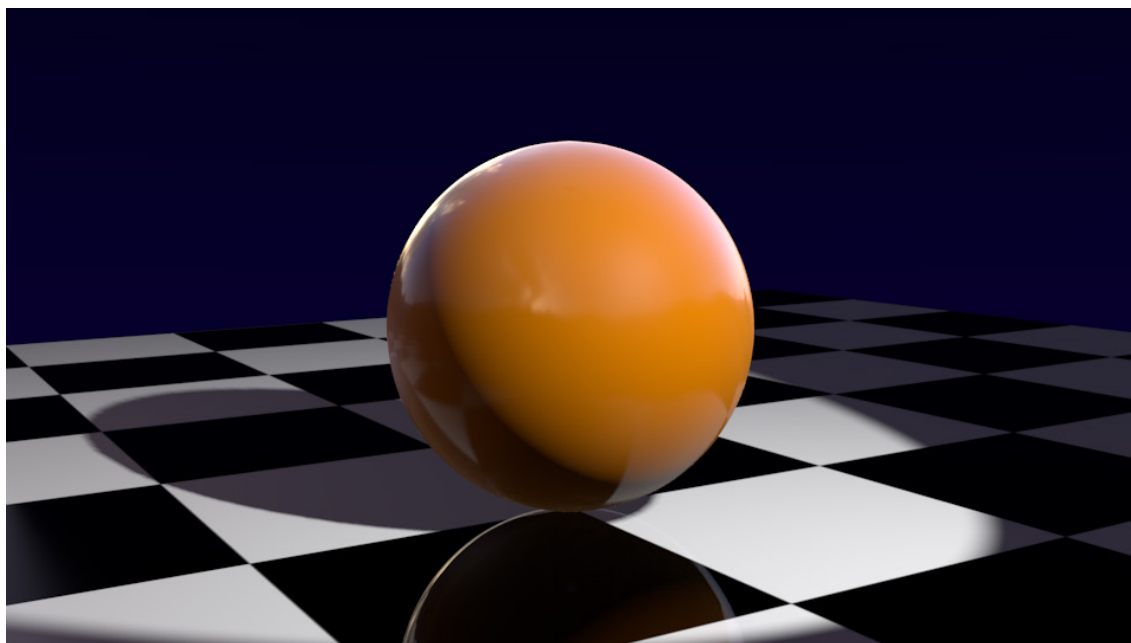
Trammell lisää materiaaliin vielä diffuusiväriä, joka toimii kappaleen pohjaväriä heijastusten alla (kuva 42). Tämä vaatii ensin uuden materiaalin, joka lisätään materiaalipaneeliin ja sen nimeksi annetaan diffuusi. Materiaalille määritellään haluttu diffuusiväri, jonka intensiteetiksi asetetaan 1.0, ja samalla Specular-osion intensiteetti tiputetaan nolnaan. Näin väri on puhdas diffuusiväri ilman valoheijastuksia. Tämän jälkeen Node Editorissa kopioidaan toinen materiaalisolmuista ja diffuusimateriaali linkitetään siihen. (Trammell 2015.)



Kuva 42: Lopulliseen yhdistelmämalliin on lisätty vielä pohjaväriksi diffuusimateriaali. Kun se yhdistetään MixRGB-solmulla aiempiin heijastuksiin, diffuusiväri on vallitsevana etualalla ja heijastukset näkyvät kappaleen reunoilla (Kuvakaappaus: Harri Pietiläinen).

Lopuksi MixRGB-solmu kopioidaan ja kytketään alkuperäisen ja Output-solmun väliin. Uuden solmun sekoitustyyppiä vaihdetaan Add (oletus Mix). Ylempään Color-porttiin kytketään diffuusimateriaali ja alempaan ensimmäinen MixRGB-solmu. Näin pohjaväri on vallitseva ja heijastukset näkyvät selvemmin kappaleen reunoilla. (Trammell 2015.)

Yksi ongelma materiaalissa on kuitenkin ilmeinen, ja se johtuu BI:n rajoituksista. Heijastus pysyy nimittäin koko ajan paikallaan, vaikka kappaletta kierrettäisiin ympäri. Se näyttää siis joka suunnasta täsmälleen samalta eikä tekstuuri siis kulje kappaleen mukana. Muutoin materiaali näyttää melko hyvältä, vaikka ei ”aito” PBR-materiaali olekaan (kuva 43).



Kuva 43: esitellyllä solmuasetelmalla aikaansaatu materiaali näyttää heijastuksen ansiosta melko kelvolliselta muovimateriaalilta, vaikka onkin tuotettu BI:ssa pienten kikkailujen avulla (Kuva: Harri Pietiläinen).

## 5 Blend4Web / WebGL

### 5.1 Lyhyt esittely

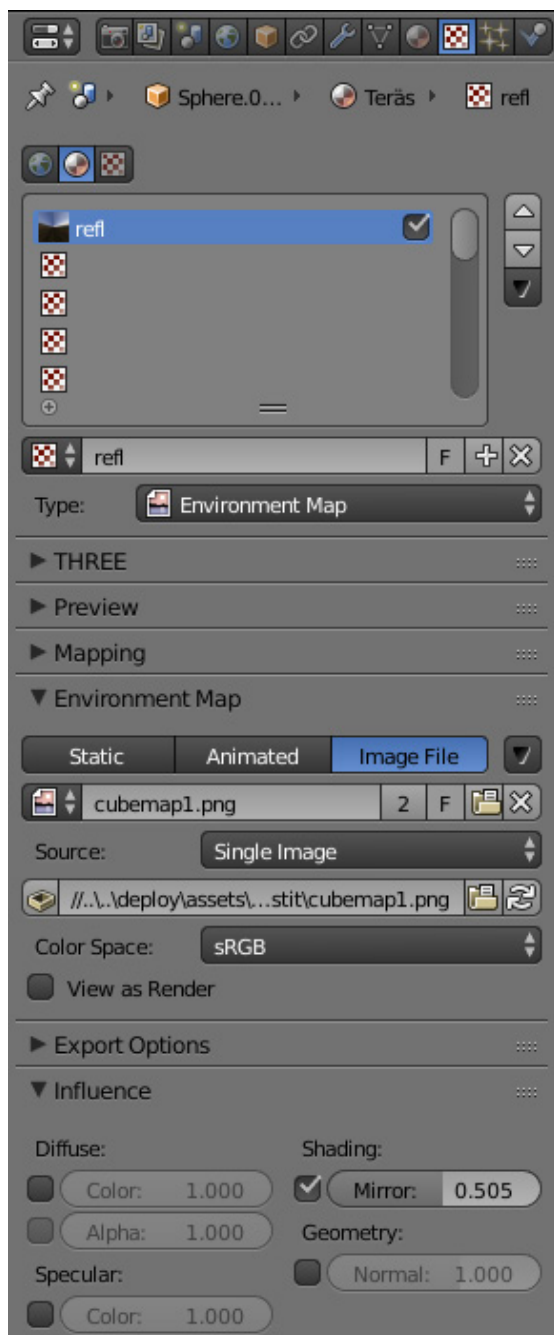
Blend4Web (B4W) on Blender 3D -ohjelmiston rinnalle ja siihen läheisesti integroitu 3D-moottori, jonka kykenee esittämään 3D-grafiikkaa selainpohjaisesti. Sen avulla Blenderissä luoduille 3D-malleille voidaan määrittää värit, materiaalit ja pintatekstuurit, sekä erilaisia erikoisefektejä vaikkapa partikkelijärjestelmiä käyttäen (tuli, savu, ruoho, lehdet) ennen mallien renderöintia WebGL-selaingrafiikaksi. Blend4Webillä voidaan selaingrafiikkaan lisätä myös ääntä ja animaatioita sekä interaktiivisia toimintoja. (Triumph LLC 2016a.)

Blend4Webistä on olemassa kaksi eri tavoin lisensoitua versiota. Community Edition (CE) on vapaasti ladattavissa GPL-lisenssillä omaan käyttöön ja sen käyttö on ilmaista. Koska GPL-lisenssi asettaa omat ehtonsa omien tuotosten jakelulle, on ohjelmistosta olemassa myös kaupallinen versio (Blend4Web Pro), jonka avulla tekijä voi suojata oman ohjelmakoodinsa, mikä voisi esimerkiksi tässä tapauksessa tarkoittaa visualisointiin lisättyä toiminnallisuutta. Tässä opinnäytetyössä on käytetty GPL-lisensoitua Blend4Web CE -versiota, ja siksi ohjelmasta on voinut ottaa ruutukaappauksia kuvitukseksi ilman erillistä lupaa. (Triumph LLC 2016b.)

Koska Blend4Web sisältää paljon erilaisia toimintoja ja asetuksia, ei niitä kaikkia ole järkevää käydä läpi tämän työn puitteissa. Siksi tässä keskitytään pelkästään siihen, miten sen avulla määritellään sellaisia materiaaleja, jotka vastaavat mahdollisimman pitkälti aiemmin Cyclesin kohdalla määritetyjä PBR-materiaaleja. Blend4Web ei kykene tuottamaan tällä hetkellä (versio 16.07) aitoja PBR-materiaaleja, mikä johtunee osaksi myös WebGL:n rajoituksista.

## **5.2 Materiaalien määrittäminen**

Blend4Webissä voi tehdä perusmateriaalisäädöt samaan tapaan materiaalipaneelista kuin Blender Internal -moottorissa, mutta säätimiä on jonkin verran vähemmän kuin BI:ssa. Mirror-osion peilaavuussäätimet ovat paljon suppeammat ja esimerkiksi Gloss-arvo puuttuu B4W:stä kokonaan. Myös tekstuurien määrittäminen materiaaleille tekstuuripaneelissa on hyvin samankaltainen kuin BI:ssa, mutta esimerkiksi Influence-osiossa säätimiä on vain muutama. Pikaisen testin pohjalta kävin nopeasti ilmi, että Blenderissä mallinnettu teräsmateriaali ei toiminut sellaisenaan B4W:ssä. Selaingrafiikkana se ei näyttänyt lainkaan teräkseltä. Erityisesti heijastukset jäivät puuttumaan.



Kuva 44: Blend4Webissä heijastukset voidaan tuottaa staattisina kuvatekstuurin avulla. Tällöin kuva määritellään ympäristötekstuuriksi, joka asetetaan heijastamaan kappaleen pinnasta (Kuvakaappaus: Harri Pietiläinen).

Materiaalille voi edellä mainitun peilaavuusasetuksen (Mirror) lisäksi määrittellä heijastusominaisuudet joko staattisena tai dynaamisena. Staattinen heijastus saadaan aikaan kuvatekstuurilla, joka kytketään materiaaliin tekstuuripaneelissa (kuva 44). Kuvan tyypiksi määritetään ympäristökartta (Environment Map) ja kun haluttu kuva on avattu paneelin Environment Map -osiossa, Influence-osiosta asetetaan Mirror-asetus päälle. Heijastuksen voimakkuutta voidaan säätää liuku-säätimellä. Samalla otetaan oletusarvona päällä oleva Diffuse-asetus pois. Staattisen heijastuksen ongelmana on se, että se ei heijasta lainkaan ympäristöä, vaan ainoastaan sen mitä näkyy tekstuurissa, eikä se liioin muutu tilanteen mukaan. (Triumph LLC 2016a: Textures.)

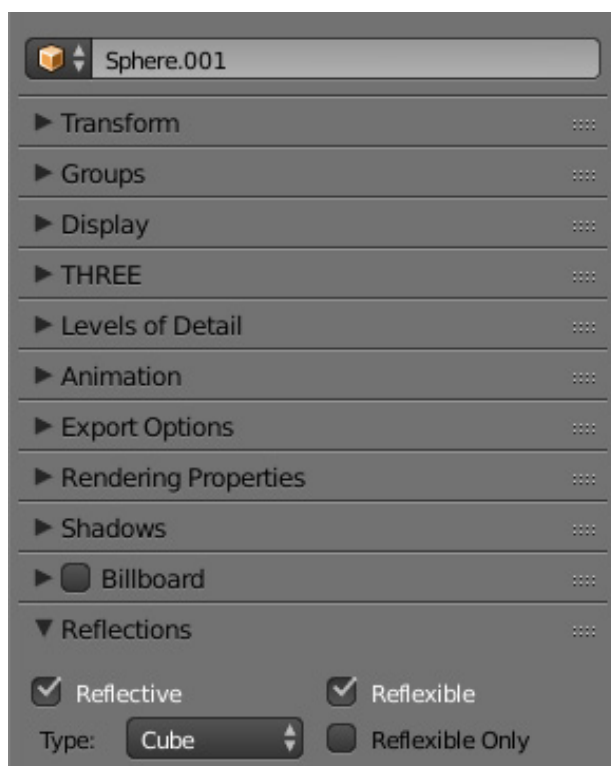
Dynaaminen heijastus säädetään sen sijaan erikseen kahdesta eri paikasta. Dynaamiset heijastukset asetetaan ensin päälle renderöintipaneelin Reflections and Refractions -osiosta (kuva 45). Oletusarvoisesti Reflections-arvo onkin pääl-

lä. Samalla voidaan valita heijastusten laatu (Quality), vaihtoehtoina ovat Low, Medium ja High. (Triumph LLC 2016a: Materials.)



Kuva 45: Renderöintipaneelin voidaan heijastukset asettaa päälle sekä valita, minkä laatusina heijastukset näytetään. (Kuvakaappaus: Harri Pietiläinen)

Tämän jälkeen heijastavuus täytyy määritellä kullekin kappaleelle erikseen Object-paneelin Reflections-osiosta (kuva 46). Kappaleen voi määrittää heijastavaksi asettamalla Reflective-asetus päälle. Heijastustyyppiä voidaan valita joko Plane tai Cube. Plane eli tasomainen heijastus toimii kuin normaali peilipinta ja toimii siten tasoilla kuten lattiassa. Cube-asetuksen avulla monimutkaisemmissa kappaleissa, muodosta riippumatta, voidaan käyttää heijastusten synnyttämiseen kuvakuutiota (cube map), jollaisia on perinteisesti käytetty ympäristötekstuureina. (Triumph LLC 2016a: Materials.)

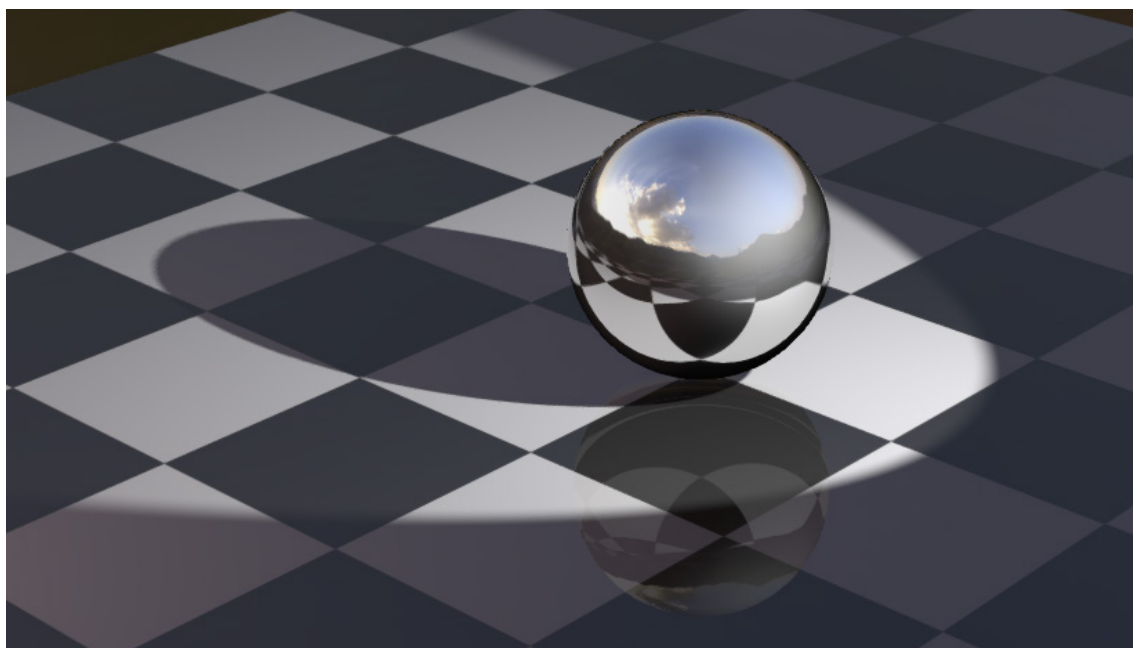


Kuva 46: Heijastusominaisuudet on määritettävä kullekin kappaleelle erikseen objektipaneelistä. Cube-tyyppi määrittelee, että heijastuksissa käytetään ympäristötekstuuria. (Kuvakaappaus: Harri Pietiläinen.)



Reflexible-asetus puolestaan määrittää sen, heijastuuko kappale toisista kappaleista vai ei, eli sekin kannattaa laittaa päälle. Kolmas asetus, Reflexible Only, on varsin kätevä silloin, jos halutaan kappaleen näkyvän heijastuksena muissa kappaleissa, mutta itse kappaletta ei renderöidä. (Triumph LLC 2016a: Materials.)

Edellä kuvatuilla perusasetuksilla saadaan jo melko hyvää jälkeä aikaan. Web-grafiikaksi renderöity testimalli näyttää suhteellisen aidolta, vaikka siinä ei olekaan käytetty vielä kaikkia Blend4Webin tarjoamia välineitä hyväksi (kuva 47).



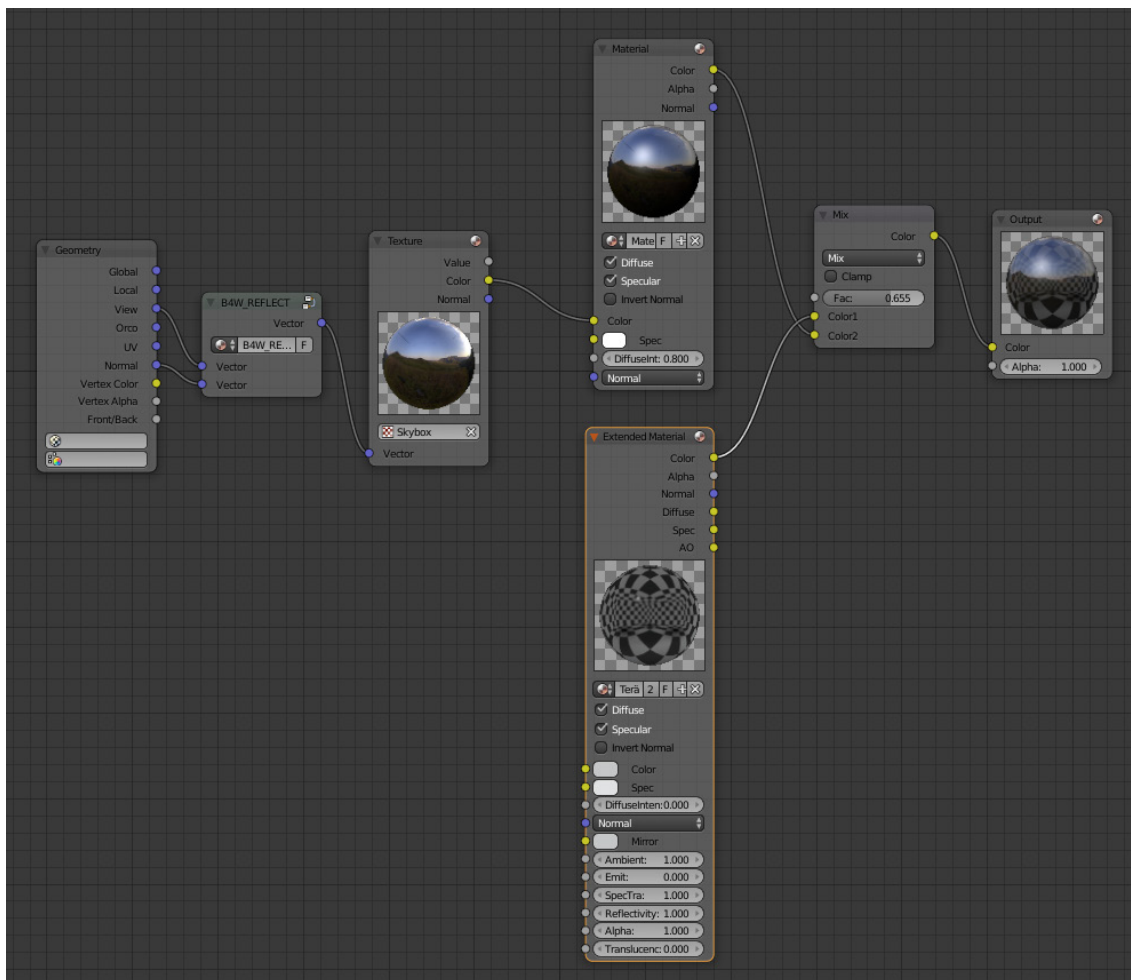
Kuva 47: Blend4Webillä tuotetussa selaingrafiikassa teräspallo näyttää heijastuksineen melko aidolta, joskin pientä säätöä materiaalin ulkoasuun voi vielä varmasti tehdä (Kuva: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

Blend4Webissä materiaaleja voi määrittää BI:n tapaan solmujen avulla Node Editorissa. B4W:n perussolmut ovat samoja kuin BI:ssa, mutta näiden lisäksi B4W:n mukana tulee joukko erikoissolmuja, jotka tarjoavat materiaalien määrittämiseen lisätyökaluja. Solmujen avulla edellinen metallimateriaali voidaan toteuttaa toisella tavalla ja toimivammin.

Tähän toteutukseen hain inspiraatiota BI:n yhteydessä esitellystä Trammellin mallista. Kappaleelle lisätään materiaalipaneelissa uusi materiaali, olkoon nimenä vaikka Teräs + heijasteet. Uusi materiaali siirretään materiaalipinon päällim-



mäiseksi ja siitä tehdään samalla solmumateriaali laittamalla materiaalin solmumäärityspainike päälle.

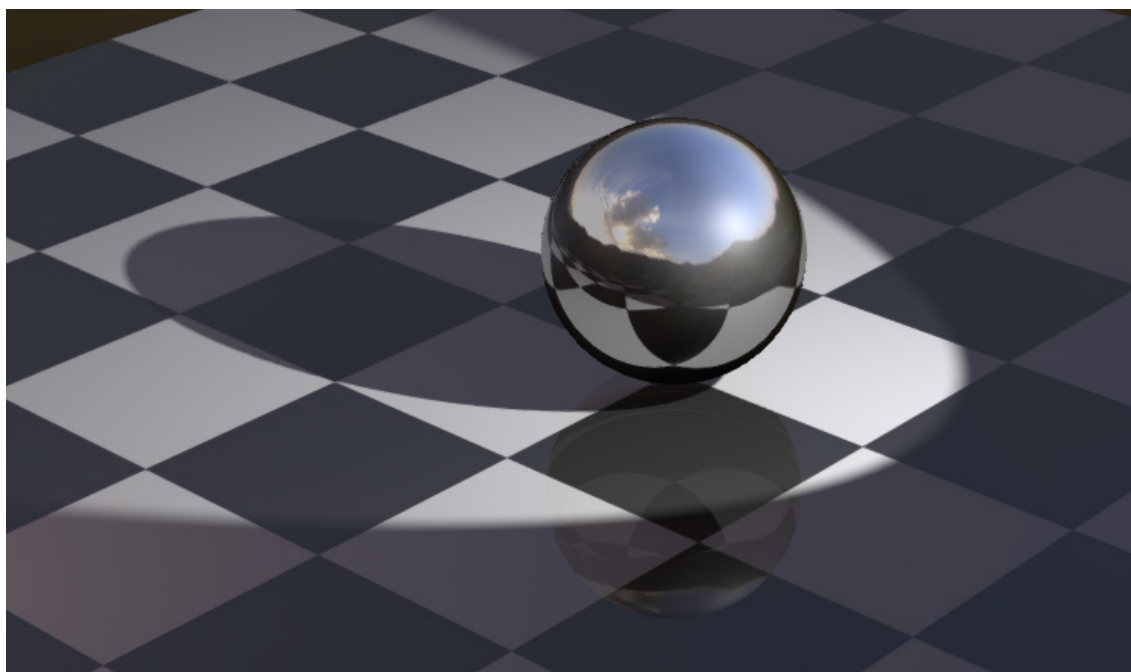


Kuva 48: Edellistä teräsmateriaalia on paranneltu lisäämällä siihen ympäristöheijastukset. Blend4Webin Reflect-solmun avulla heijastus näkyy kappaleen pinnassa oikein. Alkuperäinen teräsmateriaali sekoitetaan yhteen heijastusmateriaalin kanssa MixRGB-solmussa (Kuvakaappaus: Harri Pietiläinen).

Seuraavaksi siirrytään Node Editoriin, missä Material-solmulle luodaan uusi materiaali – ei siis linkitetä olemassa olevaa kuten aiemmin BI:ssa (kuva 48). Sen sijaan tälle uudelle solmumateriaalille rakennetaan heijastus ja se edellyttää useita lisäsolmuja. Ensimmäiseksi tarvitaan jälleen Geometry-solmu ja sen rinnalle B4W:n erikoissolmuista B4W\_Reflect. Nämä kaksi kytketään toisiinsa siten, että B4W\_Reflect-solmun ylempään Vector-porttiin liitetään Geometry-solmun View-portti, ja alempaan taas Normal-porttiin. Ideana tässä on se, että B4W\_Reflect laskee katselukulman ja kappaleen pintojen normaalien perusteella sen, miten heijastus näkyy kappaleen pinnassa. Heijastusta varten tosin tarvitaan vielä

Texture-solmu, johon ladataan tekstuuriksi sama kuvakuutiotekstuuri, jota käytetään tässä esimerkissä myös ympäristötekstuurina. Lopuksi Texture-solmun Color-portista vedetään yhteys Material-solmun Color-porttiin, ja heijastusmateriaali on valmis. (Triumph LLC 2016a: Node materials.)

Nyt uusi heijastusmateriaali täytyy yhdistää olemassa olevaan teräsmateriaaliin. Tämä tapahtuu siten, että asetelmaan lisätään uusi Extended Material -solmu, johon teräsmateriaali linkitetään. Tässä tilanteessa ei voi käyttää tavallista Material-solmua, koska sitä käytettäessä materiaali ei heijasta alla olevaa shakkikuvioista alustaa lainkaan, toisin kuin Extended Material -solmu. Materiaalisolmut yhdistetään MixRGB-solmulla siten, että teräsmateriaali tulee ylempään Color-porttiin ja heijastusmateriaali alempaan. Erilaisista sekoitusoperaatioista oletusarvo Mix näyttäisi toimivan parhaiten, ja sekoitussuhdetta säädetään kunnes tulos on kelvallinen. Tässä esimerkissä heijastusta korostetaan hieman teräsmateriaalin kustannuksella arvolla 0.655. Näin kappale heijastaa alaosassa shakkilaudan ja sen ulkopuolella näkyy heijastus ympäristöstä (kuva 49). BI-esimerkin tyyliin tästä metallimateriaalista voisi pienin muutoksin mallintaa melko kelvollisen muovimateriaalinkin.

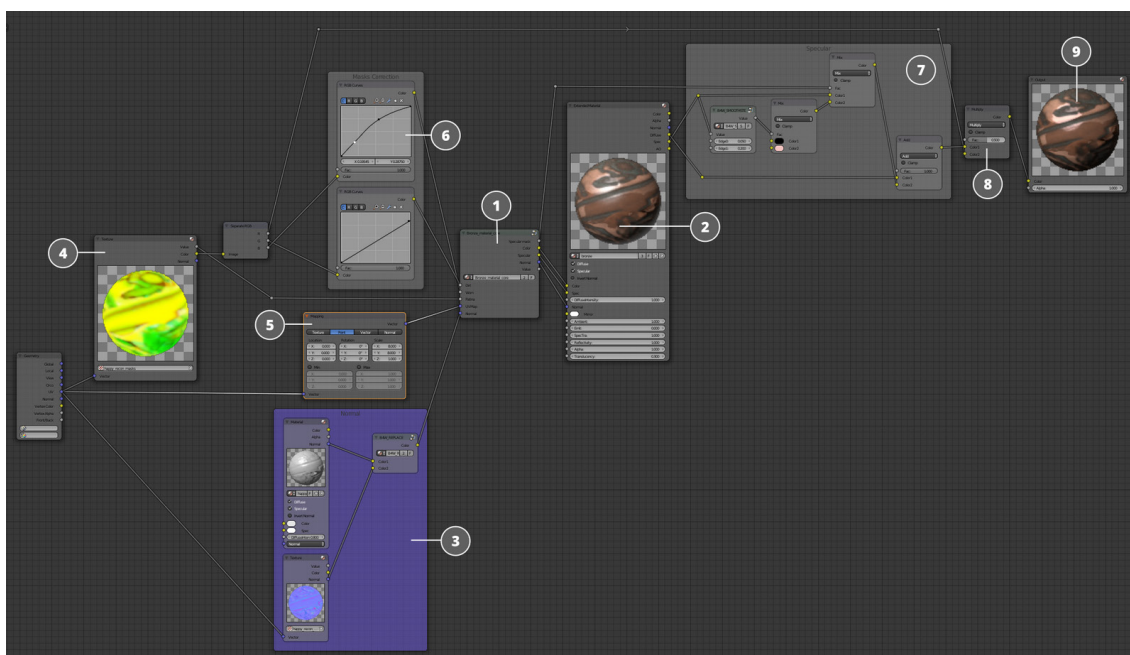


Kuva 49: Alkuperäiseen teräsmateriaaliin verrattuna pinta näyttää nyt kiiltävämmältä ja heijastukset ovat hieman kirkkaammat (Kuva: Harri Pietiläinen).

### 5.3 Materiaalikirjasto

Blend4Web CE:n asennuspaketti sisältää nykyisin myös pienen materiaalikirjaston, jossa on muutamia valmiiksi mallinnettuja materiaaleja vapaasti käytettäväksi omissa projekteissa. Näitä ovat muun muassa kumi, muovi, lasi, pronssi ja kromi. Maksullisessa Pro-versiossa materiaalikirjasto on laajempi.

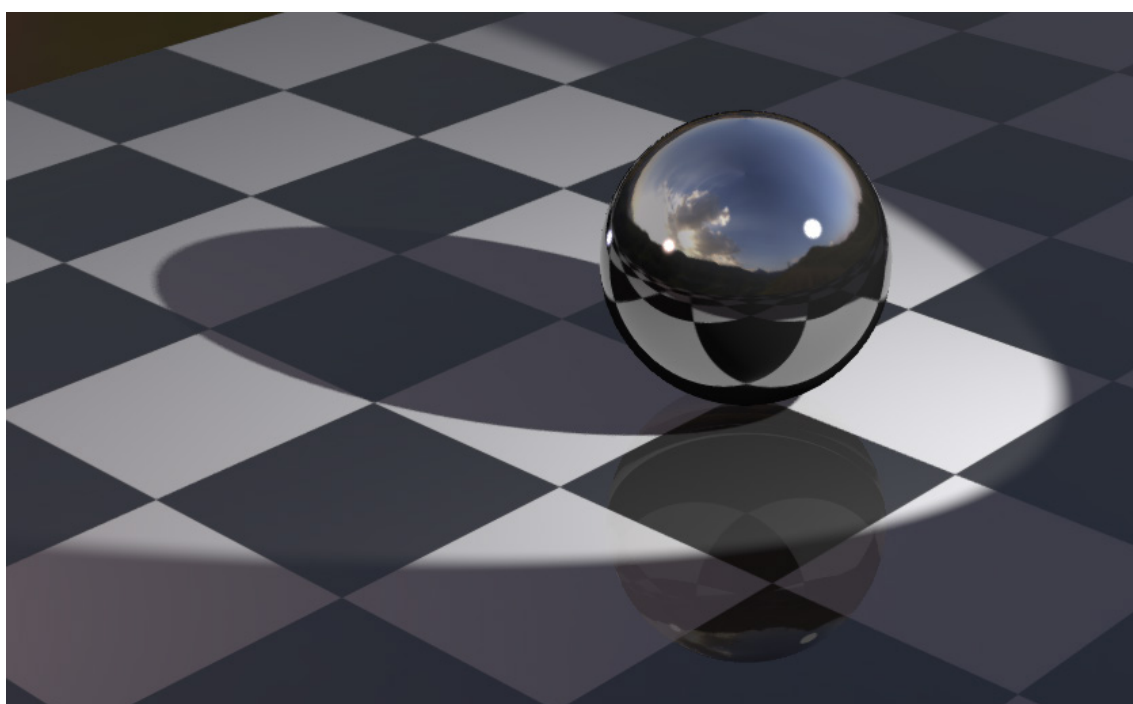
Kullekin materiaalille on määritelty niin sanottu ydinmateriaali, jota voidaan käyttää omien materiaalien pohjana. Ydinmateriaalin ympärille voi kuitenkin lisätä paljon muunlaisia asetuksia, jotka tekevät siitä uskottavamman. Näitä ovat erilaiset pintatekstuurit, normaalikartat, heijastuskartat jne. Niinpä materiaali voi olla lopulta jo melko monimutkaisen näköinen, vaikka yksittäiset osat ja niiden toiminnot ovat sinänsä aika helposti ymmärrettäviä.



Kuva 50: Blend4Webin käyttöohjeessa esitellään materiaalin solmurakennetta. Materiaalin ydinryhmän (1) ja Material-solmun lisäksi voidaan tarvita esimerkiksi tekstuureja (4) ja heijastekarttoja (7) (Kuvakaappaus: Harri Pietiläinen).

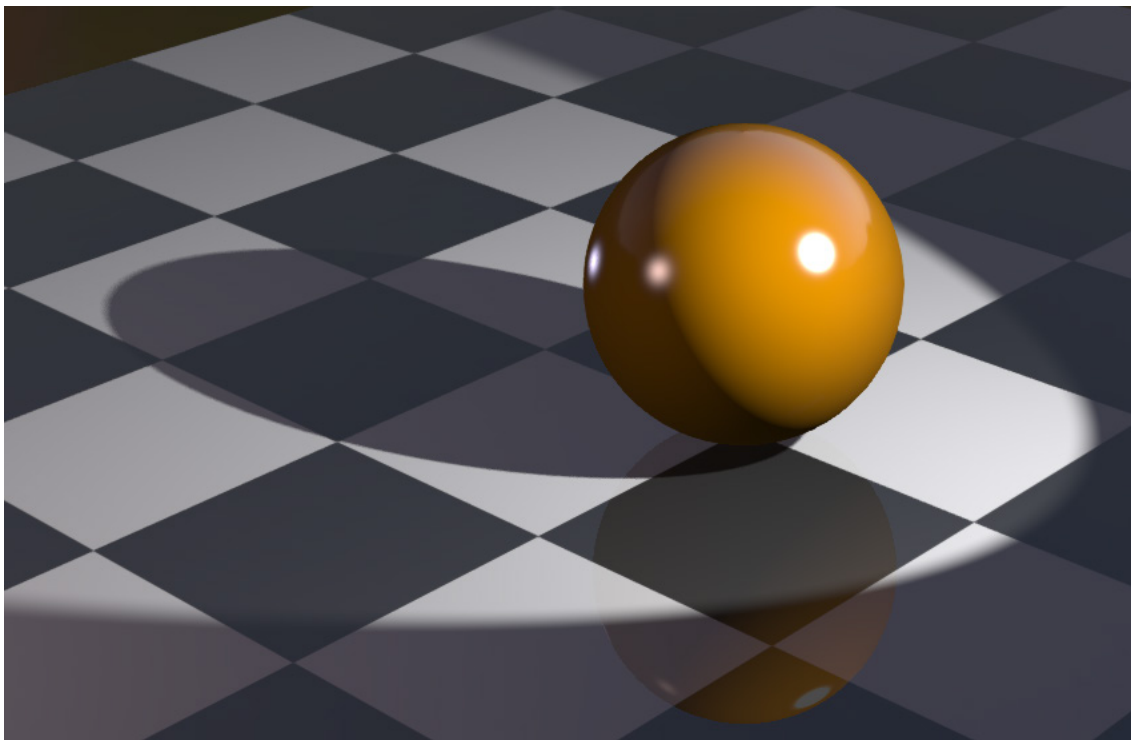
Materiaalikirjaston kromi on materiaalina melko lähellä kiiltävää terästä, paitsi että kromi on yleensä paljon kiiltävämpää. Muutoin sävyero näiden kahden välillä on hyvin minimaalinen. Siksi kromia voi harkita teräsmateriaalin pohjaksi, mikäli

kiiltoa hieman vähennetään. Sellaisenaan kromi ei sovellu aiemman teräsmateriaalin korvaajaksi, sillä siitä puuttuu lähiympäristön heijastukset (esimerkissä shakkilauta). Asian voi korjata sillä, että Material-solmun tilalle vaihdetaan Extended Material -solmu, joka kykenee näitä heijastuksia toistamaan (kuva 51). Tällöin täytyy kuitenkin pakko säätää materiaalin arvoja materiaalipaneelistä. Itse tehtyihin materiaaleihin verrattuna kromi on varsin monimutkainen. Mikäli kappaleeseen halutaan lisätä yksityiskohtia normaalikarttojen avulla, silloin kromimateriaalin kaltainen solmuasetelma on jo tarpeen.



Kuva 51: Korjattu kromimateriaali näyttää vielä kiiltävämmältä kuin teräs edellä. Myös shakkilauta heijastuu oikein, kun materiaalisolmuksi on vaihdettu Extended Material (Kuva: Harri Pietiläinen).

Myös materiaalikirjastosta löytyvä kiiltävä muovi näyttää melko vakuuttavalta sen jälkeen kun heijastuskartat on vaihdettu sopivampiin (Kuva 52). Teräksen tapaan tässäkin materiaalissa ei ole lähiympäristön heijastuksia, joten ne pitää lisätä samalla tavalla kuin kromin kohdalla. Varsinkin muovissa heijastukset näyttävät tällöin liiankin selviltä, säätöjä siis tarvitaan tässäkin. Tavallisesti muovin heijastukset ovat kuitenkin niin vähäisiä, ettei niillä ole suurta merkitystä kokonaisuuden kannalta. Jos muovin kiiltoa tarvitsee säätää, se onnistuu helposti materiaaliin määritellyllä liukusäätimellä.



Kuva 52: Materiaalikirjaston muovimateriaali vaikuttaa pienten säätöjen jälkeen aika uskottavalta, ja kiiltoakin voi tarvittaessa säätää pienemmäksi (Kuva: Harri Pietiläinen).

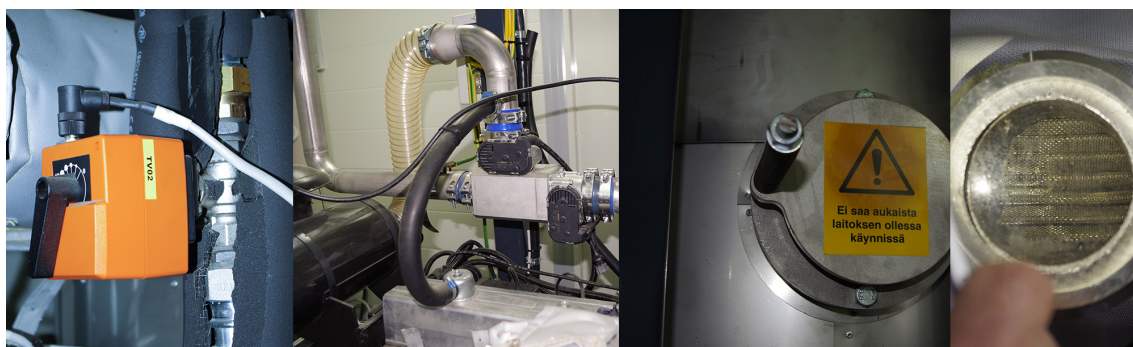
Läpikuultavat muovilaadut sen sijaan tuottavat tälle muovimateriaalille ongelmia. Koska Material-solmu ei sisällä läpinäkyvyystietoa, se on tässäkin tapauksessa vaihdettava Extended Material -solmuun. Silloin materiaalipaneelissa tehtävät läpinäkyvyysäädit vaikuttavat myös solmupohjaiseen materiaaliin. Transparency-osiossa läpinäkyvyys asetetaan ensin päälle kohdasta Show Transparency, minkä jälkeen Alpha-liukusäätimellä määritellään kappaleen läpinäkyvyyden tai läpikuultavuuden aste. Arvo 0.5 on aika lähellä sopivaa. Lopuksi Type-kohtaan valitaan Alpha Sort, jotta kappale renderöidään oikein kiinteiden kappaleiden keskelle. (Triumph LLC 2016a: Materials.)

Lasi on läpinäkyvyytensä takia selkeästi hankalin mallinnettava Blend4Webissä. Lasin ongelmana on se, että väritöntä lasia ei juurikaan näe, ellei se heijasta tai taita valoa. Valmis materiaali vaikuttaa sinällään suhteellisen toimivalta, mutta sopivuus täytyy testata erikseen tapauskohtaisesti.

## 6 Materiaalien analysointi

### 6.1 CHP-laitoksen pintamateriaalit

CHP-laitoksen fotorealistisessa visualisoinnissa on tärkeää, että pintamateriaalit ovat ominaisuuksiltaan todellisuutta vastaavia. Harjattu teräs, maalattu metalli ja muovi käyttäytyvät eri valaistusolosuhteissa eri tavalla ja siksi myös laitoksen rakennemateriaalit täytyy visualisoinnissakin erottua selkeästi toisistaan. Koska muita materiaaleja on käytetty melko vähän, on materiaalien mallinnuksessa keskitytty vain näihin. Kuvassa 53 on esimerkkejä materiaaleista.



Kuva 53: CHP-laitoksessa muovia on käytetty muun muassa erilaisissa kytkimissä ja läpikuultavana ilma- ja syöttöputkissa. Terästä on käsittelemättöinä putkistoissa, maalattuna kontin seinä- ja tukirakenteissa ja rosteripintana kaasuttimessa ja suodattimessa. Myös kumiletkuja on tarvittu. Lasia on laitoksessa vähän, muun muassa varmistussuodattimen tähytysikkunassa (Kuvat: Harri Pietiläinen).

Laitoksessa on käytetty terästä lähinnä kuumuutta tai mekaanista rasitusta kestävässä kohdissa. Näitä ovat kaasutin, suodatin, lämmönvaihtimet, polttomoottori, sähkögeneraattori, soihdpuhaltimen moottori, sekä jäähdytys- ja kaasuputket. Lähes kaikki kuumat putkistot on suurelta osin eristetty lämmön talteen ottamiseksi ja palovammojen ehkäisemiseksi, joten niitä ei todellisuudessa näy kuin pieneltä osin paljaina. Myös kontin ja hakesiilon runkorakenteet ja seinät sekä kuljettimet ja tuhkaruuvi ovat terästä, samoin kuin ulkoseinän lauhdutinkeno sekä tuuletusaukkojen ritilät.



Terästä esiintyy laitoksessa eri muodoissa: harjattuna (rosteri), maalattuna ja selaisenaan. Rosteria on käytetty vain kaasuttimessa ja suodattimessa. Maalattuja pintoja esiintyy polttomoottorissa, sähkögeneraattorissa, sulkusyöttimessä sekä lisäksi monissa ulkoilman armoilla olevissa kohteissa, kuten kontin rungossa, tukirakenteissa ja seinissä, hakesiilossa sekä kuljettimissa ja niiden moottoreissa. Osassa näistä on käytetty valettua metallia ja siksi pintarakenne on karkea. Näitä ovat lähes kaikkien moottorien rungot. Lisäksi lauhdutinkenko, ohjaus- ja sähkökaapit sekä akuston suojakuori on maalattu. Jäähdytys- ja kaasuputket, lämmönvaihtimet ja sisällä olevat pakoputkiston osat ovat sen sijaan käsittelemätöntä terästä, mutta ne on todellisuudessa suojattu eristeillä. Visualisoinnissa suojaeristeet on kuitenkin jätetty pois.

Messinkiä on käytetty siellä täällä jäähdytyskierron putkiston liitoskappaleissa ja venttiileissä, mutatakuissa sekä muun muassa sähkökeskuksen ovenkahvoissa. Suurin osa messingistä lienee valettua ja siksi pintarakenteeltaan hieman epätasaista. Ovenkahvat puolestaan ovat sileää ja kiiltävää materiaalia.

Muovia on käytetty laitoksessa tarvittavassa elektroniikassa, kytkimissä ja erilaisissa antureissa, sekä esimerkiksi kiertovesipumpussa. Muutamissa paikoissa muovi on läpinäkyvää. Lisäksi muovia on käytetty tulipalon varalta hakekuljettimien liitosputkissa, tuhkaruuvien tyhjennysputkissa sekä ilmansuodattimen ja kaasunseikoittajan yhdistävässä ilmaputkessa. Nämä viimeksi mainitut ovat samantyyppistä ohutta ja läpikuultavaa muovia, jonka on tarkoitus sulaa nopeasti tulipalon sattuessa ja näin estää palon leviäminen eteenpäin.

Laitoksen jäähdytysputkistossa on käytetty muutamissa kohdissa mustaa kumi-putkea, lähinnä polttomoottoriin liittyvissä kohdissa. Nämä ovat paikkoja, joissa jäähdytysvesi tai ilma eivät ole kumin kestävyyskannalta liian korkeita.

Lasia esiintyy vain muutamissa kohdissa, ja niistäkin osa näkymättömissä rakenteiden sisällä. Selkeimmin esillä on kaasuputken varmistussuodattimen tarkistusikkuna, josta voi tarkistaa varasuodattimen tilanteen. Lasia (tai läpinäkyvää

pleksiä / muovilasia) on käytetty myös ohjauskeskuksen ovesa olevassa kosketusnäytössä.

## 6.2 Blend4Web ja visualisointi

### 6.2.1 Kiiltävä teräspinta

CHP-voimalan 3D-visualisoinnissa pintamateriaalit oli määritettävä huolella, jotta käyttäjälle syntyisi selvä käsitys laitoksessa käytetyistä todellisista rakennusmateriaaleista (kuva 54). Esimerkiksi kirkkaat, kiiltävät teräspinnat näyttävät Blend4Webillä tuotetussa web-grafiikassa aidon oloisilta vain silloin, kun ympäristö heijastuu niistä luonnollisesti.



Kuva 54: Blend4Webillä toteutettu 3D-visualisointi jäi opinnäytetyön puitteissa vielä keskeneräiseksi, mutta jo tästä versiosta voi päätellä, millaista jälkeä B4W:llä on mahdollista tuottaa (Kuvakaappaus: Harri Pietiläinen, HDRi-taustakuva: Greg Zaal (CC-BY)).

Blend4Webissä ympäristötekstuuri voi olla täysin irrallinen heijastuksista eli kapale voi heijastaa jotain muuta kuin mitä visualisoinnissa näkyy taustalla. B4W:ssä voidaan määritellä useita ympäristötekstuureita, joista yhtä käytetään ympä-



ristövalaistuksen tai laitoksen ulkopuolisen ympäristön määrittämiseen ja muilla voidaan määrittää yksilöllisesti kunkin kappaleen heijastukset. Koska voimalan rakenne ei muutu eivätkä rakenteet liiku visualisoinnin aikana, voidaan kappaleiden materiaaleissa käyttää staattisia heijastuskarttoja aiemmin kuvatulla tavalla. Dynaamisia heijastuksia sen sijaan ei kannata käyttää, koska tällöin kappaleet heijastaisivat myös laitoksen ulkopuolista ympäristöä mikä ei ole visualisoinnissa toivottavaa (kuva 55).



Kuva 55: Tässä vaiheessa kuvakuutioiden perustuvia heijastuskarttoja ei ole vielä toteutettu ja siksi etualan kirkkaat teräspinnat eivät vielä heijasta laitoksen sisätiloja. Mattapintaiset teräsputket puolestaan näyttävät kovin vaisuilta, joten materiaali kaipaa vielä hiomista (Kuvakaappaus: Harri Pietiläinen).

Yksilöllisten heijastusten tuottamiseksi kullekin kappaleelle luodaan oma kuvakuutiotekstuuri, jonka avulla voidaan määrittää heijastukset kappaleen pinnassa. Luonnollisin vaihtoehto on, että kiiltävät teräsmateriaalit heijastavat voimalan sisä rakenteita (seinät, katto, ovet jne.). Tällöin kullakin kappaleella on omanlaisensa näkökulma kyseiseen tilaan ja ympäröiviin kappaleisiin ja siksi kuvakuutiotekstuureidenkin pitäisi olla erilaisia. Onneksi voimalassa on vain muutama kappale (mm. kaasutin, suodatin, kondenssivesisäiliö), joissa kiiltävää terästä (tai rosteria) tarvitaan, joten kuvakuutioiden luominen ei välttämättä ole liian työlästä.

Ympäristötekstuuri voidaan kehittää (bake) Blender Internalin tai Cyclesin puolella kappaleen omasta näkökulmasta ja tallentaa sitten kuvakuutioksi. BI:ssa ympäristötekstuurin tekeminen on ainakin tällä hetkellä huomattavasti helpompaa kuin Cyclesissa, mutta vastaavasti Cyclesissa materiaalit ovat selvästi parempilaatuisia kuin BI:ssa. Kuvakuutiota käytetään B4W:ssä kappaleen materiaaliin kytkettynä tekstuurina staattisena heijastuskarttana. Näin materiaali näyttää eri

kulmista katseltuna heijastavan esimerkiksi laitoksen sisärakenteita, mistä syntyy vaikutelma kiiltävästä ja sileäpintaisesta materiaalista. (Blend4Web 2016.)

Mattapintaisissa teräspinnoissa voidaan käyttää periaatteessa samaa teräsmateriaalia ilman heijastuksia. Niiden sijaan pinnan elävöittämiseksi materiaaliin voi lisätä jonkinlaisen pintatekstuurin, jonka avulla esimerkiksi teräsputket näyttävät hieman käsitellyiltä, likaisemmilta ja siten aidommilta. Todellisuudessa suurin osa kaasu- ja vesiputkista on suojattu eristeillä, mutta joitakin putkia on näkyvillä paljainakin. Niistä voi siis ottaa mallia visualisoinnin toteutukseen. Mikäli pintatekstuureita tarvitaan, on kyseiset kappaleet purettava tai kaavoitettava kaksiulotteisiksi UV-kartoiksi.

### 6.2.2 Messinki

Kiiltävää teräsmateriaalia voi käyttää myös messinkisissä kappaleissa. Sähkökeskuksen ulko-oven kahva on kirkasta messinkiä, mutta koska se sijaitsee laitoksen ulkoseinällä, siinä voi suoraan hyödyntää materiaalin Mirror-ominaisuuksia. Kahva voi heijastaa ympäristöään sellaisenaan, toisin kuin laitoksen sisällä olevat kirkkaat teräspinnat, eikä teräsmateriaalin kuvakuutioheijasteita tarvita. Hieman yksinkertaisempi ja mahdollisesti jopa pelkästään perusmateriaalimäärittäyksillä ilman solmuja toteutettu ratkaisu siis voi riittää mainiosti.



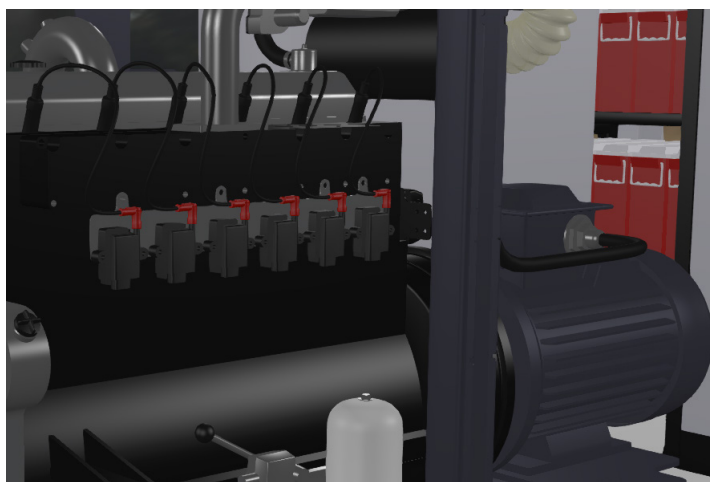
Kuva 56: Jäähdytysputkiston liitoskappaleet ja muut messingistä tehdyt osat eivät nekään ole vielä kovin vakuuttavia läheltä tarkasteltuna. Näistä puuttuu myös valetun messingin epätasaisuus, jonka voi lisätä normaalikartan avulla (Kuvakaappaus: Harri Pietiläinen).

Mattapintaisiin teräskappaleisiin käytettävää teräsmateriaalia voidaan puolestaan hyödyntää valetusta messingistä valmistetuissa kappaleissa eli lähinnä vesiputkistossa käytetyissä venttiileissä, liitoksissa jne. Valetut kappaleet ovat kuitenkin pinnaltaan hieman röpelöisiä, joten aitouden lisäämiseksi valettuun messinkimateriaaliin voi lisätä pintatekstuuriksi vaikkapa jonkin sopivan normaalikartan (kuva 56).

Myös kohoumakartta voisi tulla kyseeseen, sillä lopputulos kummallakin on kuktakin samanlainen. Yksityiskohdat erottuvat vain hyvin läheltä katsottaessa, mutta tekevät pinnasta epätasaisen näköisen ja siten hieman elävemmän. Valetut kappaleet eivät peilaa ympäristöään, vaan heijastukset ovat hyvin epämääräisiä juuri pinnan epätasaisuuden vuoksi. Myös normaali- tai kohoumakartan käyttäminen edellyttää kappaleiden UV-kartoitusta.

### 6.2.3 Valurauta

Röpelöpintaista messinkimateriaalia voi hyödyntää melko helposti valurautaisten kappaleiden (sähkögeneraattori ja -moottorit, sulkusyötin) materiaalina, vaikka nämä ovatkin kaikki maalipintaisia ja siten käyttäytyvät hieman eri lailla kuin käsittelemättömät metallipinnat (kuva 57). Valuraudan pintarakenne muistuttaa kuitenkin pitkälti valetun messingin pintaa, joten messingissä hyödynnetyt pintatekstuurit voidaan todennäköisesti hyödyntää sellaisenaan. Myös heijastukset ovat



Kuva 57: Polttomoottorin valurautainen lohko näyttää ilman normaalikarttaa liian sileältä, samoin kuin moottorin jatkona oleva sähkögeneraattorikin. Molemmat materiaalit kaipaavat siis vielä lisätyötä (Kuvakaappaus: Harri Pietiläinen).

valuraudassa samankaltaisia eli hyvin epämääräisiä. Kuten messingin kohdalla, myös valurautakappaleet tulee UV-kartoittaa, mikäli niissä käytetään pintatekstuureita.

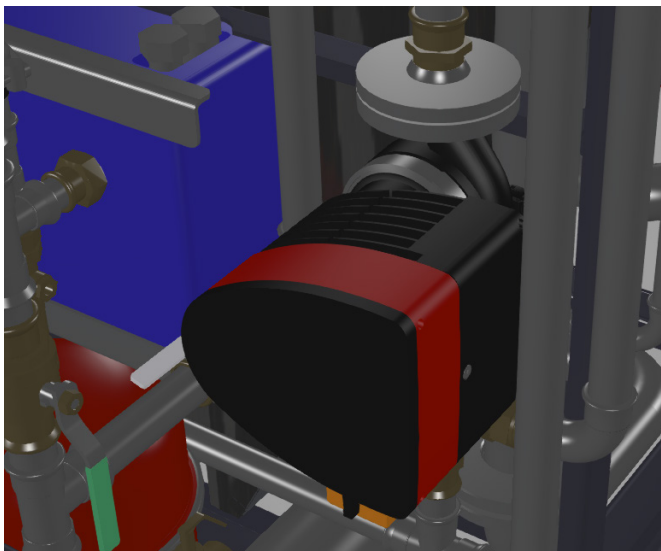
#### **6.2.4 Maalatut metallipinnat**

Maalatuille metallipinnoille voidaan käyttää valuraudan materiaalia ilman valetun pinnan pintatekstuureita. Kuitenkin esimerkiksi laitoksen ulkoseinissä on käytetty teräspeltiä, jossa kulkee poikkisuuntaisia uria säännöllisin välein. Näitä uria ei tietenkään kannata lisätä 3D-malliin, vaan ne voi halutessa toteuttaa normaali-kartan avulla. Urat ovat aika pieni ja epäolennainen yksityiskohta koko voimalan kannalta mutta ne elävöittävät pintaa ja lisäävät hivenen todenmukaisuutta. Myös likaa ja valumajälkiä voidaan lisätä sinne tänne osittain läpinäkyvien siirtokuvatekstuurien (decals) avulla. Myös siirtokuvat vaativat kappaleista UV-karttoja.

Muissa maalatuissa metallipinnoissa (runkorakenteet, kuljettimet, ohjaus- ja sähkökeskukset jne.) voidaan käyttää perusmetallimateriaalia ilman normaalikarttoja. Myös näihin pintoihin voidaan tosin lisätä tarvittaessa likaa, nokea jne. siirtokuvien avulla.

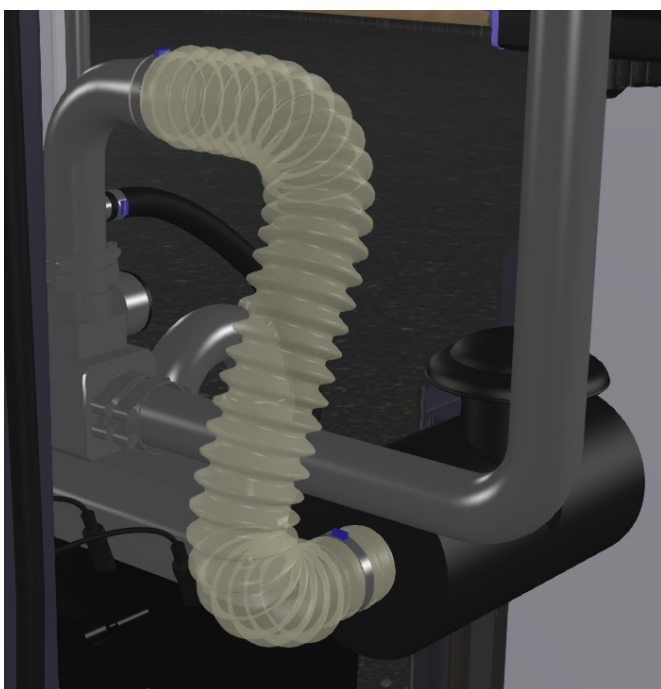
#### **6.2.5 Muovi**

Useimmat laitoksessa näkyvät muovipinnat ovat sileitä ja mattapintaisia. Ne ovat kooltaan pieniä, erilaisia säätimiä tai laitteiden osia (kuva 58). Niissä voi käyttää materiaalikirjaston valmista sileää muovimateriaalia, joka näyttää sellaisenaan varsin hyvältä visualisoinnissa. Tavallisesti kappaleita ei edes tarvitse välttämättä purkaa tai kaavoittaa kaksiulotteiseksi UV-kartaksi (UV unwrap), ellei niiden yhteydessä käytetä mitään pinta- tai muita kuvatekstuureja.



Kuva 58: Kiertovesipumpun suojakuoret ovat muovia ja ne näyttävät jo perusmateriaalia käyttäen melko hyvältä. Muovi onkin yksi helpoiten määriteltävistä materiaaleista, joita visualisoinnissa on tarvittu (Kuvakaappaus: Harri Pietiläinen).

Voimalassa on myös läpikuultavasta muovista valmistettuja putkia muutamassa paikassa, ja varsinkin ilmansuodattimesta kaasunsekoittimeen kulkeva ilmaputki on melko iso ja näkyvä kappale (kuva 59). Samaa muovilaatua on käytetty myös kaasuttimen päällä olevassa syöttöputkessa sekä ulkona tuhkanpoistoputkissa tuhkaruuvien ja tynnyreiden välissä. Siksi materiaali täytyy mallintaa huolella, jotta se renderöitynä näyttäisi mahdollisimman hyvältä.



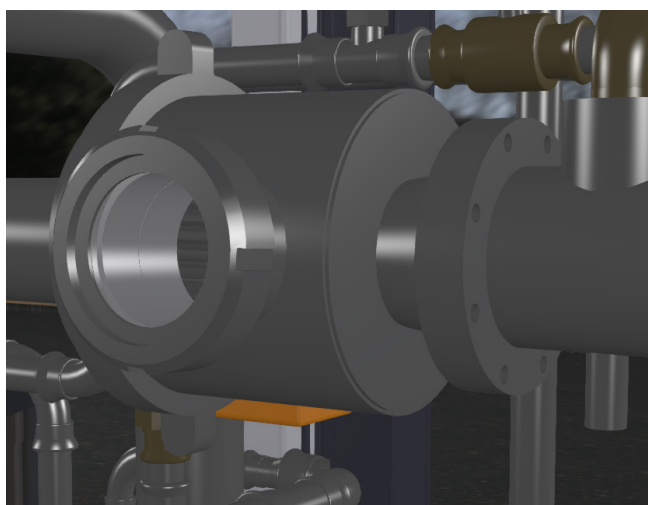
Kuva 59: Läpikuultavuuden toteuttaminen tuottaa web-grafiikassa hieman ongelmia ja sopivien asetusten löytäminen vaatii kokeilua. Läpikuultavaa muovia on käytetty muun muassa tässä ilmaputkessa. Tätäkin materiaalia täytyy vielä parantella (Kuvakaappaus: Harri Pietiläinen).

Läpikuultavilla materiaaleilla on kuitenkin yksi piirre, joka täytyy jotenkin huomioida: ne eivät nimittäin Alpha Sort -tilassa luo lainkaan varjoja. Yksi (ja tällä

hetkellä ainoa) ratkaisu tähän on kopioida kappale ja jättää kopio täsmälleen samaan kohtaan alkuperäisen kanssa. Kopiolle ei anneta lainkaan materiaalia, ja sen varjoasetukset määritetään objektipaneelin Shadows-osiossa siten, että Cast Shadows sekä Cast Only ovat molemmat päällä. Näin alkuperäinen kappale renderöidään läpikuultavana ja kopiokappale tuottaa ainoastaan varjon, muutoin se jää näkymättömiin. Varjo on tosin liian tumma ollakseen läpikuultavan kappaleen aiheuttama, mutta siihen on pakko tällä erää tyytyä (kuva 54). (Blend4Web 2015.)

### 6.2.6 Lasi

Lasin osalta tilanne on siinä mielessä hieman hankala, että kirkasta lasia on voimassa hyvin harvoissa ja pienissä kohteissa ja niissäkin lasi jää aika huomaamattomaksi (kuva 60). Jotta lasi ei materiaalina jäisi täysin turhaksi, voisi lasin pintaan lisätä pölyä, nokea tai vaikkapa kulumajälkiä. Tällöin pinta heijastaisi valoa epätasaisemmin ja lasi erottuisi materiaalina selkeämmin. Muiden materiaalien siirtokuvatekstuureja voi siis hyödyntää tässäkin, ja onneksi lasikappaleiden kohdalla jälkiä tarvitsee käytännössä vain ulkopinnalle, joten kaavoittaminen UV-kartaksi on triviaalia.



Kuva 60: Kaasulinjan varmistussuodatimen tarkastusikkunan lasi jää tällä hetkellä täysin näkymättömiin. Se ei näytä heijastavan valoa eikä sen ulkopinnassa ole mitään paljastavia yksityiskohtia. Sisällä oleva suodatinkasetti näkyykin paljon selkeämmin kuin todellisuudessa (Kuva-kaappaus: Harri Pietiläinen).

## 6.2.7 Kumi

Kumi on sinänsä helppo materiaali, että sitä on käytetty vain muutamissa paikoissa, ja lisäksi sitä varten löytyy valmiiksi mallinnettu materiaali Blend4Webin materiaalikirjastosta (kuva 61). Koska laitoksen kumiputket ovat pinnaltaan melko sileitä eikä niissä ole mitään poikkeavaa pintarakennetta, valmis kumimateriaali toimii sellaisenaan ihan kelvollisesti. Mikäli kumin pintaan halutaan lisätä likaa tai pölyä siirtokuvatekstuurilla, nämäkin kappaleet tulee kaavoittaa UV-kartaksi.



Kuva 61: Voimalan näkyvimät kumikapaleet ovat jäähdytysputkiston kumiputket, joiden kautta jäähdytysvesi kulkee polttomoottoriin ja sieltä pois. Perusmuovin tavoin kumi on melko yksinkertainen materiaali. Blend4Webin materiaalikirjastosta löytyy myös hieman paranneltu versio, jonka avulla kumin pintaan voi lisätä tekstuureilla esimerkiksi kulumaa tai likaa (Kuvakaappaus: Harri Pietiläinen).

## 7 Pohdinta

### 7.1 Projektin tulokset

Vaikka visualisointi jäikin opinnäytetyön puitteissa kesken, se ehti osoittaa miten paljon työtä ja aikaa tällaisen toteuttamiseen vaaditaan. Myös tiedonhankinta Blenderissä ja Blend4Webissä käytettävistä menetelmistä vei aikaa, sillä tieto on usein hajallaan pieninä palasina ja niiden etsiminen ja löytäminen oli välillä aika hankalaa. Tiedon koostamisesta yhteen sekä sen hyödyntämistä visualisoinnis-



sa on silti ollut ja tulee olemaan paljon hyötyä, sillä ne lisäävät ammattitaitoa ja osaamista varsin monipuolisesti.

Blend4Web oli ennen tätä projektia aivan uusi tuttavuus, ja tuntuu että näinkin laajan työn aikana siihen on ehtinyt perehtyä vain hyvin pintapuolisesti. Edes kaikkia materiaalien määrittelyyn liittyviä yksityiskohtia ei ole ollut mahdollisuutta selvittää tässä ajassa. Olisin toivonut, että perehtymiseen olisi ollut käytettävissä enemmän aikaa, mutta itse visualisoinnin aikataulu asetti opinnäytetyölle omat rajoitteensa. Siksi lopputulos ei ollut vielä tässä vaiheessa aivan niin hyvä kuin mihin pyrin.

Koska visualisointi on tarkoitus viedä loppuun ja viimeistellä opinnäytetyön valmistumisen jälkeen, tulee B4W varmasti entistäkin tutummaksi välineeksi ja lopputuloksen laatukin varmasti paranee. On täysin mahdollista, että tulen käyttämään B4W:tä jatkossa muissakin vastaavissa projekteissa, mikäli sopivia kohteita tulee vastaan. Ainakin omia pienempiä kokeiluja aion jatkaa edelleen. Oma tulevaisuuteni on vielä täysin auki enkä siksi osaa sanoa, millaisia työhaasteita elämä seuraavaksi eteeni asettaa.

## **7.2 Blend4Web työvälineenä**

Blend4Webistä on kehittymässä varsin monipuolinen väline mitä erilaisimpiin tarkoituksiin. Vaikka se ei olekaan vielä täysin loppuun asti hiottu tuote, sitä käytetään jo selainpohjaisiin visualisointeihin, animaatioihin, tuote-esittelyihin ja jopa peleihin. Onpa joku älynnyt tehdä sillä jopa interaktiivisen portfolionkin itselleen. Näyttää siltä, että uusia sovellutuksia tulee koko ajan lisää, ja vain mielikuvitus on rajana sille, mitä Blend4Webillä voi ajan kuluessa tehdä.

B4W:llä toteutetut sovellukset toimivat periaatteessa millä tahansa selaimella (IE, Firefox, Chrome, Safari, Opera jne.), alustalla (Windows, Mac, Linux jne.) ja laitteella (tietokone, tabletti, älypuhelin), joita käytämme tavallisesti päivittäin.



B4W-sovellukset toimivat selaimessa sellaisenaan ilman että käyttäjän tarvitsisi asentaa omalle laitteelleen mitään selainlaajennuksia (toisin kuin esimerkiksi Unityn kohdalla). Selain-, alusta- ja laiteriippumattomuuden takia sovelluskehittäjän tarvitsee parhaassa tapauksessa tehdä sovelluksesta vain yksi versio, jota voidaan käyttää monissa eri ympäristöissä samalla tavalla. Toki tällöin sovelluksessa täytyy ottaa huomioon responsiivisuus eli sovelluksen tulee skaalautua käytettävän laitteen näyttöresoluution mukaan.

Blend4Web on myös käytettävissä täysin ilmaiseksi avoimen lähdekoodin CE-versiona (Community Edition), mikä varmasti innostaa monia tutustumaan tähän välineeseen. Tarvittaessa sovelluksissa voi hyödyntää myös kaupallista lisenssiä, mikäli ei halua jakaa avoimen lähdekoodin periaatteiden mukaisesti kaikkea tuottamaansa koodia vapaasti eteenpäin. Itselleni nimenomaan vapaa käyttö ilman rajoituksia oli yksi niistä syistä, miksi halusin tutustua B4W-sovellusten tekemiseen, koska samalla periaatteella toimiva Blender oli jo entuudestaan tuttu väline.

Blend4Webissä on hyvää myös se, että se tarjoaa varsin laajan skaalan erilaisia tekniikoita ja välineitä omien sovellusten toteuttamiseksi, ja uusia ominaisuuksia julkaistaan vuoden mittaan useasti. Kehitystyö on siis jatkuvaa ja käyttäjätuki melko toimivaa ainakin omien kokemusteni perusteella. Koska kyseessä on melko uusi väline, on tarvittavan tiedon tai esimerkkien löytyminen kuitenkin joskus hieman hankalaa. Onneksi B4W-keskustelufoorumien kautta voi tarvittaessa esittää kysymyksiä suoraan kehittäjille.

Tämän visualisointiprojektin perusteella on noussut esiin kuitenkin joitain miinuspuoliakin, joita B4W:iin liittyy. Laajemmista visualisoinneista tulee helposti aika raskaita sovelluksia, joiden lataus ja käyttö voi olla laitteesta ja nettiyhteyden nopeudesta riippuen hidasta. Välillä sovellukset voivat jumittua kokonaan tai lataus katketa kesken. Osa ongelmista johtuu varmasti siitäkin, ettei visualisoinnissa käytettyä 3D-mallia ole optimoitu tarpeeksi hyvin, vaan se on edelleen liian

tarkasti toteutettu. Mitä enemmän visualisointi sisältää dataa, sitä hitaammaksi lataus ja käyttö luonnollisesti käyvät.

Tällä hetkellä B4W ei myöskään tue vielä Cyclesin kaltaisia PBR-materiaaleja, joten materiaalit täytyy tuottaa hieman turhankin monimutkaisesti. Kyse ei ole WebGL:n / Html5:n rajoituksista, sillä esimerkiksi WebGL-toteutuksissa usein käytettävässä Three.js-kirjastossa on jo toteutettu PRB-sävyttimiä. B4W:iin on kuitenkin jo lupailtu lähitulevaisuudessa PBR-tukea, mutta aikataulu on vielä tätä kirjoitettaessa avoin.

### **7.3 Blend4Webin tulevaisuudennäkymät**

Koska B4W:n kehitystyö on jatkuvaa ja uusia ominaisuuksia lisätään koko ajan, on aika vaikea arvata mihin suuntaan kehitys on viemässä. Yksi mielenkiintoinen uudehko piirre B4W:ssä on tuki stereokuville ja virtuaalitodellisuudelle (VR). Tämän myötä B4W:llä voidaan toteuttaa myös sovelluksia myös VR-laseille, mikä avaa taas kokonaan uusia mahdollisuuksia sovelluskehittäjille. Ja voisi kuvitella, että samalla ovet avautuvat myös lisätyn todellisuuden (AR, augmented reality) sovelluksille, joskaan tätä tukea B4W ei vielä suoraan tarjoa.

Päätelaitteiden ja nettiyhteyksien tehostuessa ja nopeutuessa myös monet aiemmin liian raskaiksi ja hitaiksi todetut sovellutukset saattavat hyvinkin toteutua Blend4Webin avulla. Esimerkiksi selainpohjaisia nettipelejä on markkinoilla ollut jo pitkään, joten siinä ei ole sinänsä mitään uutta tai erikoista. B4W voi kuitenkin tarjota toisenlaisen tavan toteuttaa pelejä, varsinkin kun pelien pelaaminen ei vaadi selainlaajennuksia. Kun B4W on lisäksi integroitu toimimaan suoraan toisen avoimen lähdekoodin ohjelman eli Blenderin kanssa, nämä kaksi muodostavat varsin käyttökelpoisen kokonaisuuden sovelluskehittäjälle. Blenderin ja Blend4Webin kehitys kulkee varmasti jatkossakin tiukasti käsi kädessä, ja uudet piirteet ja ominaisuudet Blenderissä voivat siten tuottaa jotain aivan uutta myös B4W:ssä ja päinvastoin.

## Lähteet

- Birn, J. 2013. Top Ten Tips for More Convincing Lighting and Rendering. <http://www.peachpit.com/articles/article.aspx?p=2165641>. 16.3.2016.
- Birn, J. 2006. Digital lighting and rendering (second edition). Berkley, USA: New Riders.
- Blend4Web. 2015. How to: material with png alpha. <https://www.blend4web.com/en/forums/topic/266/?page=1>. 2.9.2016.
- Blend4Web. 2016. Cubemap Textures: Creating and Using. 21.7.2016. <https://www.youtube.com/watch?v=ZjZRBfajApw>. 26.4.2016.
- Blender Foundation. 2015. Blender 2.8 – the Workflow release. 20.7.2015. <https://code.blender.org/2015/07/blender-2-8-the-workflow-release/>. 25.8.2016.
- Blender Foundation. 2016. Blender reference manual (Blender v. 2.77). <https://www.blender.org/manual/>. 17.3.2016.
- Hoffman, N. 2013. Background: Physics and Math of Shading. [http://blog.selfshadown.com/publications/s2013-shading-course/hoffman/s2013\\_pbs\\_physics\\_math\\_notes.pdf](http://blog.selfshadown.com/publications/s2013-shading-course/hoffman/s2013_pbs_physics_math_notes.pdf). 22.7.2016.
- Ilama, J. 2014. Dynaaminen teksturointi reaaliaikaisessa 3D-grafiikassa. Metropolia Ammattikorkeakoulu. Viestinnän koulutusohjelma. Opinnäytetyö.
- Joensuu, J. 2016. 3D-alan sanasto. Kajaanin ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö.
- Kontu, J. 2014. 3D-valaistuksen tekniikat. Tampereen ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö.
- Kuittinen, V. 2016. Painoalaprojektipäällikkö. Karelia-ammattikorkeakoulu. Haastattelu. 18.3.2016.
- Parkin, K. 2014. Building 3D with Ikea. [http://www.cgsociety.org/index.php/cgs-features/cgsfeaturespecial/building\\_3d\\_with\\_ikea](http://www.cgsociety.org/index.php/cgs-features/cgsfeaturespecial/building_3d_with_ikea). 19.3.2016.
- Price, A. 2012. Achieving photorealism in Blender. <http://www.blenderguru.com/articles/achieving-photorealism-in-blender/>. 19.3.2016.
- Price, A. 2014. 7 lessons on photorealism we can learn from Alex Roman. <http://www.blenderguru.com/podcasts/podcast-7-lessons-on-photorealism-we-can-learn-from-alex-roman/>. Podcast. 20.3.2016.
- Price, A. 2015. Why you should be striving for photorealism. <http://www.blenderguru.com/articles/photorealism-important/>. 19.3.2016.
- Price, A. 2016a. How to make photorealistic PBR materials – Part 1. <https://www.youtube.com/watch?v=V3wghbZ-Vh4>. 28.6.2016.
- Price, A. 2016b. How to make photorealistic PBR materials – Part 2: Metal. <https://www.youtube.com/watch?v=m1PkSViBi-M>. 13.7.2016.
- Puhakka, A. 2008. 3D-grafiikka. Helsinki: Talentum.
- RealLifePudding, 2016. Correcting Blenders Roughness Value. <https://tastypudding.wordpress.com/2014/12/19/simple-pbr-shader-for-blender-2-72-cycles/>. 14.8.2016.
- Roman, A. 2016. The art from The Third and the Seventh. <http://thirdseventh-book.com/the-book/>. 20.3.2016.

- Russell, J. 2016. Basic Theory of Physically-Based Rendering. <http://www.marmoset.co/toolbag/learn/pbr-theory>. 16.3.2016.
- Sairiala, J. 2015. PBR workflows in Cycles Render Engine. Tampereen ammattikorkeakoulu. Degree Programme in Media. Opinnäytetyö.
- Simonds, B. 2013. Blender Master Class. A Hands-on Guide to Modeling, Sculpting, Materials and Rendering. San Fransisco, USA: No Starch Press, Inc.
- Trammell, K. 2015. "PBR" in Blender's Viewport. 20.5.2015. <https://cgcookie.com/2015/05/20/pbr-in-blenders-viewport/>. 25.8.2016.
- Triumph LLC. 2016a. Blend4Web user manual. <https://www.blend4web.com/doc/en/index.html>. 17.3.2016.
- Triumph LLC. 2016b. Frequently asked questions. <https://www.blend4web.com/en/faq/> 17.3.2016.
- Volter Oy. 2015a. Volter 40 – teknisiä tietoja (pdf).
- Volter Oy. 2015b. Käyttöohjekirja – Volter 40 omasähkölaitos (SN31082016-15) (pdf). Kempele.
- Volter Oy. 2016. Esittelyvideo. <https://www.youtube.com/watch?v=sdqh5nENK-HY>. 24.8.2016.
- Wikipedia. 2016a. Relativistic Quantum Chemistry. [https://en.wikipedia.org/wiki/Relativistic\\_quantum\\_chemistry#Color\\_of\\_gold\\_and\\_caesium](https://en.wikipedia.org/wiki/Relativistic_quantum_chemistry#Color_of_gold_and_caesium). 23.8.2016.
- Wikipedia. 2016b. Refractive Index. [https://en.wikipedia.org/wiki/Refractive\\_index](https://en.wikipedia.org/wiki/Refractive_index). 23.8.2016.
- Wikipedia. 2016c. Bidirectional scattering distribution function. [https://en.wikipedia.org/wiki/Bidirectional\\_scattering\\_distribution\\_function](https://en.wikipedia.org/wiki/Bidirectional_scattering_distribution_function). 17.9.2016.
- Wilson, J. 2016a. Tutorial: Physically Based Rendering, And You Can Too! <http://www.marmoset.co/toolbag/learn/pbr-practice>. 16.3.2016.
- Wilson, J. 2016b. Tutorial: Tutorial: PBR Texture Conversion <http://www.marmoset.co/toolbag/learn/pbr-conversion>. 16.3.2016.