

Sini Haapanen

Peliobjektin teksturointi PBR-materiaaleilla käyttäen Substance Designer -ohjelmaa

Metropolia Ammattikorkeakoulu

Medianomi

Viestinnän Koulutusohjelma

Opinnäytetyö

21.5.2017

Tekijä(t) Otsikko	Sini Haapanen Peliobjektin teksturointi PBR-materiaaleilla
Sivumäärä Aika	40 sivua + 6 liitettä 21.5.2017
Tutkinto	Medianomi
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	3D-animaatio ja -visualisointi
Ohjaaja(t)	Lehtori Kristian Simolin
<p>Tämä opinnäytetyö käsittelee peliympäristöön soveltuvan 3D-mallin teksturointia PBR- eli Physically Based Rendering -materiaaleilla käyttäen Substance Designer -nimistä ohjelmaa. PBR:n käyttö on muodostunut standardiksi monella 3D-grafiikkaa hyödyntävällä alalla, ja varsinkin pelinkehityksessä se on otettu monessa yrityksessä osaksi tuotantoa. Työn tavoitteena oli tutkia PBR-materiaalien keskeisiä piirteitä ja soveltaa opittua henkilökohtaisessa projektissa. Tarkoituksena oli avata aihetta 3D:stä perusteet osaaville henkilöille, jotka ovat kiinnostuneet teksturoinnista, sekä havainnollistaa itse teksturointiprosessia.</p> <p>Opinnäytetyön teoriaosuudessa käydään läpi perinteisen ja PBR-teksturoinnin eroja, tutkitaan PBR:n taustalla toimivaa valon fysiikkaa sekä verrataan kahta yleisintä PBR-menetelmää. Lisäksi esitellään toiminnallisessa osuudessa käytettyä Substance Designeria ja sen ominaisuuksia. Työn toiminnallisessa osuudessa teksturoitiin projektia varten mallinnettu 3D kilpi. Kilpeä varten tehtiin kolme erilaista materiaalia: puu, metalli ja kristalli. Osuudessa käydään läpi koko teksturointiprosessi referenssien hakemisesta materiaalien yhdistämiseen. Tämän jälkeen tuotosta testattiin Unity-pelimoottorissa, jonka myötä havaittiin, että materiaalit eivät näytä samoilta Unityssa ja Substance Designerissa.</p> <p>Pohdintaluvussa todettiin, että PBR:n ymmärtäminen ja sen mukaan työskentely on monimutkaista, muttei kuitenkaan vaikeaa. Tekstuuriartisteilta on jo ennen PBR:n yleistymistä vaadittu materiaalien ymmärtämistä, joten PBR vaatii käytännössä vain toimintatapojen päivittämistä ja mahdollisesti uusien ohjelmien opiskelua. Lisäksi toiminnallisen osuuden pohjalta todettiin Substance Designerin olevan varteenotettava työkalu pelien parissa työskenteleville tekstuuriartisteille sen proseduraalisuuden ja tekstuuriin muokattavuuden ansiosta. Lopuksi mietittiin mahdollisia muita käyttötarkoituksia PBR:lle, joihin lukeutuu käytännössä kaikki 3D-grafiikkaa hyödyntävät alat ja tuotteet.</p>	
Avainsanat	PBR, teksturointi, peligrafiikka, Substance Designer

Author(s) Title	Sini Haapanen Texturing A Game Object With PBR-Materials
Number of Pages Date	40 pages + 6 appendices 21 May 2017
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	3D Animation and Visualization
Instructor(s)	Kristian Simolin, Senior Lecturer
<p>This thesis studies the texturing of game 3D models using PBR or Physically Based Rendering materials, using a software called Substance Designer. Using PBR has become a standard in many fields utilizing 3d graphics, and especially in the game industry it has been incorporated in many companies as a part of the production pipeline. The objective of this thesis was to research the core features of PBR materials and apply the knowledge gained in a personal project. The aim of the thesis was to shed light on the subject for those who are interested in texturing and who possess a basic understanding of 3D graphics. Also another goal is to demonstrate the texturing process itself.</p> <p>The theory of the thesis goes through the differences in traditional and PBR texturing, the physical phenomena underlying PBR and the two most common PBR methods and their differences. Also the thesis describes the features and the functionality of Substance Designer, which is used in the practical part. During the practical part of the thesis, a 3D shield made for this project was textured. Three different materials were done for the shield: wood, metal and crystal. The thesis covers the process from finding references to combining the created materials. Afterwards, the result was tested in Unity, which revealed that the materials looked vastly different between Unity and Substance Designer.</p> <p>One of the conclusions made from the study is that, despite understanding PBR and working with it is complicated, it isn't difficult. Texture artists have needed to understand materials even before PBR became commonly used, which means the artists only need to update their workflows and potentially adapt new tools. From the practical part it was learned that Substance Designer suits well texture artists working in the game industry due to the procedural workflow and easily modified textures. Lastly it is concluded that PBR can be used in many other fields, as every field using 3D graphics can benefit from it.</p>	
Keywords	PBR, texturing, game graphics, Substance Designer

Sisällys

1	Johdanto	1
2	Teksturointi ennen ja nyt	3
3	Valon fysiikkaa teksturiartistin näkökulmasta	4
3.1	Valon absorbointi ja sironta	5
3.2	Diffuusi- ja spekulariheijastukset	6
3.3	Mikropinta ja makropinta	8
3.4	BRDF ja Fresnel	8
3.5	Valon aallonpituudet eli värit	9
3.6	Johtavat ja eristävät materiaalit	9
4	PBR-menetelmät	10
4.1	Metalli/rosisuus-menetelmä	10
4.2	Spekulari/kiiltävyys-menetelmä	12
4.3	Yhteiset tekstuurikartat	13
4.4	Tekstuurin resoluutio ja texelien tiheys	15
4.5	Tekstuurit ja gammakorjaus	16
5	Substance Designer	18
5.1	Substanssit	18
5.2	Käyttöliittymä	19
6	Propin teksturointi	20
6.1	Proseduraalisten materiaalien tekeminen	21
6.1.1	Puu	22
6.1.2	Metalli	25
6.1.3	Hohtava kristalli	26
6.2	Leivotut tekstuurikartat	29
6.3	Lopullisen materiaalin kokoaminen ja testaus	30
7	Yhteenveto	34
	Lähteet	35

Liitteet

Liite 1. Puumateriaalin graafin kuvakaappaukset

Liite 2. Metallimateriaalin graafin kuvakaappaukset

Liite 3. Kristallimateriaalin graafin kuvakaappaukset

Liite 4. Scene Information Baking -ikkuna

Liite 5. Lopullisen graafin kuvakaappaukset

Liite 6. Linkki lähdetiedostoihin

1 Johdanto

Physically Based Rendering (*PBR*), eli fysikaalisiin ominaisuuksiin pohjautuva renderöinti, on ollut tiedossa jo vuosia. Kuitenkin vasta viime vuosina sen soveltaminen on levinnyt niin laajalle, että siitä on muodostunut standardi 3D-grafiikan tuotannossa. PBR:n taustalla vaikuttavat kehittyneemmät algoritmit, jotka sallivat realististen valon ja materian vuorovaikutuksesta johtuvien ilmiöiden simuloinnin. Lisäksi PBR tuo uudenlaisia työtapoja, jotka ovat sekä nopeampia että helpompia perinteisiin menetelmiin nähden. Tämä tekee PBR:stä hyödyllisen lähes kaikilla aloilla, jotka käyttävät 3D:tä tuotannossa. Monet ohjelmistojen ja pelimoottoreiden tarjoajat ovat jo tehneet siirtonsa ja päätyneet tukemaan PBR:ää.

Videopelien 3D-grafiikan tuotannossa tekstuureilla on suuri merkitys pelien ulkoasuun. Siksi 3D-mallien teksturointiin käytetään paljon aikaa. Teknologia ei kuitenkaan ole aina ollut sitä mitä se on nykyään. Tekstuuriartistien on täytynyt pitää tekstuurien lukumäärä mahdollisen alhaalla ja samaan aikaan pitää huoli niiden toimivuudesta kaikissa valaistuksissa. Tämä on lisännyt työn määrää ja siten vaikuttanut oleellisesti tekstuurien tekemiseen käytettyyn aikaan. PBR tuo mukanaan vanhasta poikkeavia menetelmiä teksturointiin. PBR:ää noudattavat tekstuurit ovat yksinkertaisempia, ja ne toimivat eri valaistuksissa ilman manuaalista säätöä, mikä helpottaa artistien työtä.

Tässä opinnäytetyössä tullaan tarkastelemaan PBR:n ominaisuuksia ja sitä, miten 3D-mallin teksturointi PBR-materiaaleilla tapahtuu pelinkehityksen näkökulmasta. Tavoitteena on tutkia PBR-materiaalien keskeisiä piirteitä ja soveltaa opittua peliympäristöön soveltuvan 3D-mallin teksturoinnissa. Työkalut rajataan PBR-materiaaleihin erikoistuneeseen Substance Designer-ohjelmaan. Tarkoituksena on avata aihetta 3D:stä perusteet osaaville henkilöille, jotka ovat kiinnostuneet peliobjektien teksturoinnista ja PBR-materiaaleista.

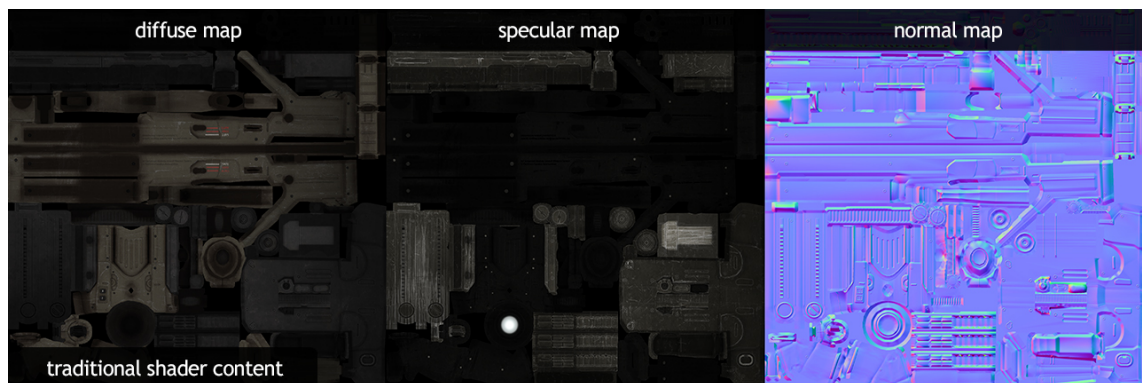
Teoriaosuudessa kerrotaan lyhyesti aiemmista teksturointimenetelmistä ja PBR:stä sekä näiden kahden eroavaisuuksista. Tämän jälkeen siirrytään esittelemään PBR:än taustalla toimivia fysikaalisia ilmiöitä. Tästä jatketaan PBR-materiaalien eri työkulkuihin ja vertaillaan näiden eroja ja yhteneväisyyksiä, minkä jälkeen käydään läpi toiminnallisessa osuudessa käytettävän teksturointiohjelman Substance Designerin ominaisuuksia.

Toiminnallisessa osuudessa toteutettiin henkilökohtaisena projektina fantasiahenkinen peliympäristöön soveltuva kilven 3D-malli, jolle luotiin kolme erilaista materiaalia. Opinnäytetyössä käydään läpi materiaalien tekemisen eri vaiheet suunnitelmasta aina valmiin teksturoidun mallin testaamiseen saakka Unity-pelimoottorissa.

2 Teksturointi ennen ja nyt

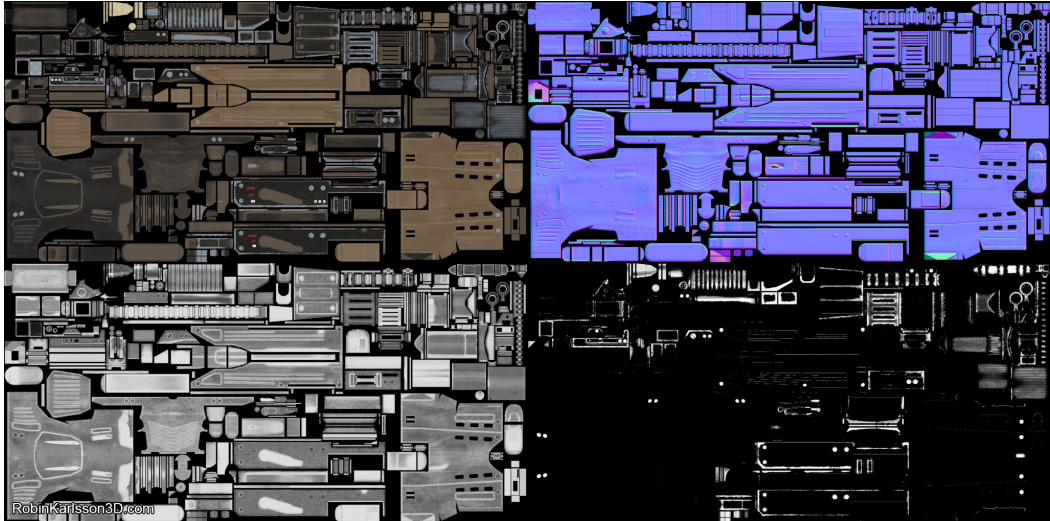
Teksturoinnissa artistin tehtävänä on tuottaa 3D-objekteille tekstuurikarttoja (*texture map*), jotka määrittelevät objektien varsinaisen ulkonäön. Tekstuurit itsessään eivät kuitenkaan huomioi objektin fyysisiä ominaisuuksia, kuten esimerkiksi heijastavuutta (Scolastici & Nolte 2013, 82). Tätä varten on olemassa pieni ohjelma nimeltä varjostin (*shader*). Sen avulla lasketaan valon intensiteetti kussakin kohdassa 3D-mallin pinnalla. (Hearn & Baker 1997, 252–256.) Tekstuurien, käytetyn varjostimen sekä mahdollisen muun datan muodostamaa lopullista kokonaisuutta kutsutaan materiaaliksi (Scolastici & Nolte 2013, 82).

Perinteisesti 3D-malleille tehdään kolme tekstuurikarttaa: diffuusi- eli värikartta, speulaarikartta ja normaalikartta (kuvio 1). Vanhemman teknologian rajallisten kykyjen vuoksi tekstuureihin lisättiin manuaalisesti varjoja ja muita vallaistukseen liittyviä tehosteita. Lisäksi monimutkaisempiin materiaaleihin tehtiin tarpeen mukaan useita erillisiä karttoja määrittelemään eri valon ominaisuuksia, kuten heijastuksia tai speulaarin voimakkuutta. (Lagarde 2011.) Eri valaistustilanteet, kuten päivä ja yö, saattoivat myös vaatia omat versionsa tehtävistä kartoista (Burke 2015).



Kuvio 1. Perinteisesti toteutettuja tekstuurikarttoja (Wilson 2015).

Nyt standardiksi muodostunut PBR yksinkertaistaa tätä prosessia tehden monista aiemmin tehdyistä tekstuureista tarpeettomia. PBR on sisällöntuotantomenetelmä, joka on yhdistelmä kehittyneempiä varjostimia ja niille tehtyjä tietynlaisia tekstuureja. PBR:n myötä digitaalisten ympäristöjen valaistus on kehittynyt niin pitkälle, ettei valoinformaatiota tarvitse sisällyttää itse tekstuureihin. Muuten ne ovat lähes samat PBR:ssä kuin vanhemmissa menetelmissä, mutta joukkoon on lisätty erillinen pinnan rosoisuutta kuvaava kartta (kuvio 2). (Wilson 2015.)



Kuvio 2. Esimerkki PBR-tekstuureista (Karlsson 2014).

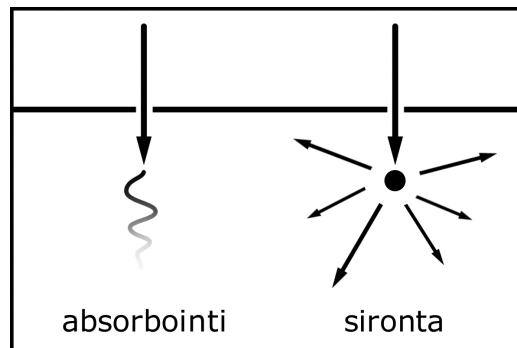
Uniikkien piirteidensä vuoksi PBR vaatii kuitenkin jonkin verran totuttelua artistilta, sillä fysikaalisesti uskottavan datan käyttäminen on tärkeää tällä tuotantomenetelmällä. Koska PBR-varjostimet matkivat todellisia valon ja pintojen ominaisuuksia, on mahdollista ja jopa suotavaa käyttää todellisten materiaalien ominaisuuksia kuvaavia arvoja. Niitä ei kuitenkaan ole helppo löytää, minkä lisäksi artistin pitää tottua siihen, ettei hän voi kokonaan hallita spekuloinnin toimintaa. Opittuaan tämän työtavan artistin on kuitenkin helpompi luoda asetteja, jotka toimivat eri valaistustilanteissa, mikä tehostaa tuotantoa. Lisäksi visuaalinen laatu on parempi kuin aiempia varjostusmalleja käyttämällä. (Lagarde 2011.)

3 Valon fysiikkaa tekstuuriartistin näkökulmasta

Varjostus on yksi tärkeimmistä ja ehkä monimutkaisemmista asioista reaaliaikaisen grafiikan renderöinnissä. Koska tekstuuriartistien vastuulla on luoda 3D-ympäristöön uskottavia materiaaleja, on heidän otettava käyttöön varjostusmalli huomioon. Sikäli kuin varjostimien tarkoitus on matkia oikean maailman valaistuksen ja pintojen fysikaalisia ilmiöitä, on tekstuureita tekeville artisteille äärimmäisen tärkeää ymmärtää, miten valo käyttäytyy materiaalin pinnalla. (McDermott n.d. a, 2.) Tähän liittyvät matemaattiset kaavat eivät kuitenkaan ole oleellisia ilmiöiden ymmärtämisen kannalta, joten ne on tässä työssä rajattu pois.

3.1 Valon absorbointi ja sironta

Valo on sähkömagneettinen aalto, joka voi imeytyä eli *absorboitua*, tai hajota eli *sirota* kulkiessaan erilaisten väliaineiden läpi (kuvio 3). Väliaineeksi kutsutaan sitä ainetta, jonka sisällä valo kulkee (Suomisanakirja n.d.). Absorbointi ja sironta tapahtuvat silloin, kun valonsäde osuu väliaineessa sijaitseviin hiukkasiin (Hiukkastieto n.d.). Mikäli väliaineen absorbointikyky ylittää tietyn pisteen, osa sen läpi kulkevasta valosta absorboituu muuttuen toiseen energian muotoon. Tämän seurauksena valon intensiteetti heikkenee ja väri voi muuttua. Sironta taas tarkoittaa valon pirstaloitumista useiksi eri suuntiin liikkuviin säteisiin. Tähän vaikuttaa väliaineen *taitekerroin*. (Hoffman 2010, 2–7.)



Kuvio 3. Absorbointi ja sironta havainnollistettuna.

Taitekerroin (*IOR*) kuvastaa väliaineen vaikutusta valon nopeuteen ja intensiivisyyteen. Homogeenisissa väliaineissa taitekerroin on yhtenäinen, joten valo kulkee suoraan muuttamatta suuntaansa. Jos väliaineen absorboivuus on lisäksi alhainen, valo pääsee kulkemaan sen läpi juurikaan muuttumatta, jolloin väliaine näyttäytyy läpinäkyvänä. Heterogeenisissä väliaineissa taitekerroin vaihtelee. Jos taitekerroin muuttuu lyhyellä etäisyydellä tarpeeksi nopeasti, se siroaa. Sekä absorboinnille että sironnalle on oleellista väliaineessa olevien hiukkasten tiheys sekä kyseessä olevan väliaineen skaala. Mitä enemmän hiukkasia, ja mitä pidemmän matkan valo kulkee, sitä enemmän valoa absorboituu ja siroaa. (Hoffman 2010, 2–7). Tämä tapahtuu myös kirkkaaksi miellettyissä materiaaleissa, mikä voidaan havaita kuviosta 4.



Kuvio 4. Veden sumeus johtuu valon absorboinnista ja sironnasta (Schau 2015).

Veden absorbointikyky on heikkoa, joten lähellä sijaitsevat objektit näkyvät selkeästi. Etäisyyden kasvaessa vedessä olevat hiukkaset absorboivat ja sirottavat valoa, mikä saa veden näyttämään sumealta.

3.2 Diffuusi- ja spekulariheijastukset

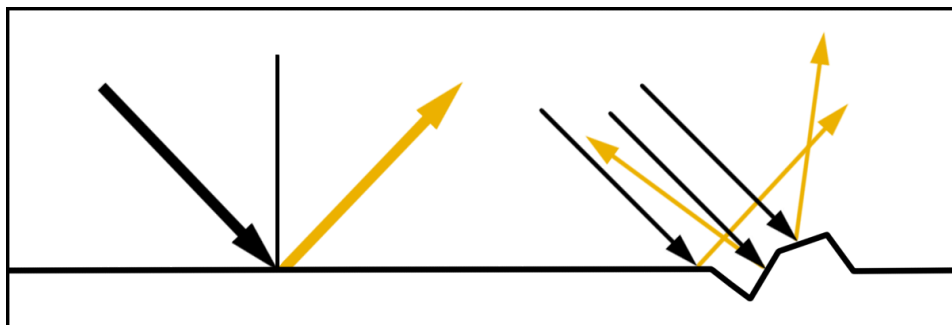
Heijastusta kutsutaan diffuusiksi tai spekulariksi riippuen siitä, onko valo käynyt kappaleen sisällä, eli onko se taittunut. Diffuusiheijastuksissa valo on läpäissyt objektin pinnan, sironnut useaan kertaan ja lopuksi palannut edelliseen väliaineeseen suurin piirtein samasta kohtaa kuin se oli tullutkin (kuvio 5). Mikäli väliaine on hyvin absorboivaa, voi olla, että valo ei pääse palaamaan, jolloin diffuusiheijastusta ei synny. (McDermott n.d. a, 4.) Näin käy muun muassa metalleissa, jotka absorboivat kaiken valon heti. Jos objektin absorbointikyky on heikko mutta sironna voimakasta, valonsäteet palaavat suurella todennäköisyydellä pinnan yläpuolelle. Joskus tämä näkyy objektin hehkumisena. Tämänlaisia materiaaleja ovat muun muassa iho ja vaha (kuvio 5). Palaavat valonsäteet eivät yleensä tule ulos samasta kohtaa, joten

esimerkiksi ihon realistista renderöintiä varten on käytettävä niin kutsuttuja SSS- eli Subsurface Scattering -varjostimia, jotka simuloivat tätä ilmiötä. (Hoffman 2010, 8.)



Kuvio 5. Kynttilän vaha hehkuu diffuusiheijastusten ansiosta (Zahradníček 2016).

Spekulaariheijastukset syntyvät, kun valo kimpoaa suoraan objektin pinnalta läpäisemättä sitä. Mikäli kyseessä on täysin tasainen pinta, niin valo noudattaa heijastumisen lakia, eli sen heijastuskulma on sama kuin tulokulma (kuvio 6). Mutta koska pinnat ovat harvemmin tasaisia, valonsäteiden heijastuskulmat vaihtelevat pinnan epätasaisuuden mukaan. Siksi rosoisemmilla pinnoilla heijastukset näyttävät isompina ja sumeampina, kun taas sileämmillä pinnoilla valonsäteet heijastuvat enemmän samansuuntaisesti, mistä seuraa tarkemmat heijastukset. Mutta vaikka toisenlainen heijastus saattaa näyttää heikommalta, niin heijastuneen valon määrä on kuitenkin sama riippumatta pinnan tasaisuudesta. (McDermott n.d. a, 3–4.)



Kuvio 6. Valon heijastuminen tasaisella ja epätasaisella pinnalla.

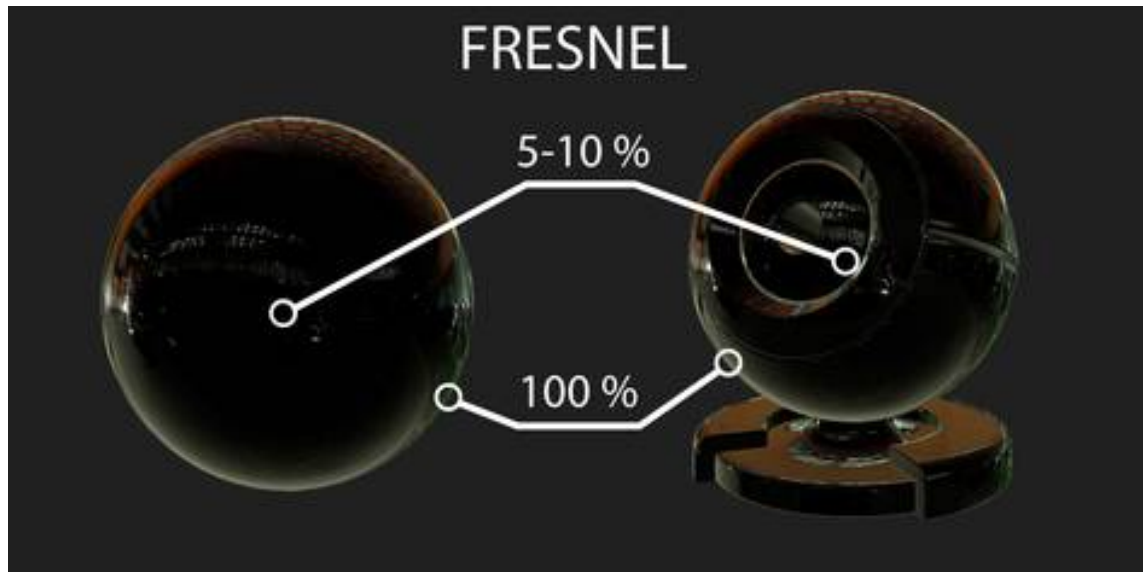
3.3 Mikropinta ja makropinta

Mikropinnasta (*microsurface*) puhuttaessa viitataan pinnan mikroskooppisella tasolla ilmeneviin epätasaisuuksiin. Teoriassa nämä vaikuttavat sekä spekulari-että diffuusiheijastuksiin, mutta diffuusiheijastuksissa ilmenevän sironnan takia sen vaikutuksen huomaa käytännössä vain spekulariheijastuksissa. (McDermott n.d., 5.) Mikropinnan katsotaan toimivan kuin iso määrä täysin tasaisia ja eri suuntiin osoittavia pintoja. Havaittu heijastus koostuu kaikista mikropinnasta heijastuneista säteistä, joten mitä samansuuntaisempia nämä ovat, sitä tarkempia heijastukset ovat. (Hoffman 2010, 6.) Koska kaikkien mikropinnan yksityiskohtien varjostus vaatisi paljon resursseja, käsitellään pintaa kokonaisuudessaan makrotasolla. Pinnan rosoisuudelle määritellään yleisluontoinen arvo, joka voidaan määrittellä joko tekstuurilla tai numeraalisena arvona. (Russell n.d.)

3.4 BRDF ja Fresnel

BRDF (*Bidirectional Reflectance Distribution Function*) on funktio, joka kuvastaa pinnan heijastusominaisuuksia, kuten heijastuvan valon määrää. BRDF-malleja on monenlaisia, mutta kaikki niistä eivät ole fysikaalisesti uskottavia. Uskottavan funktion tulee täyttää kaksi kriteeriä. Ensinnäkin sen tulee noudattaa *energian säilymisen lakia*, jonka mukaan energian määrä pysyy samana: heijastuvan valon määrä ei siis voi olla suurempi kuin sisääntulevan valon. Toiseksi funktion tulee olla symmetrinen eli isotrooppinen, eli tulevan ja lähtevän säteen kulman tulee olla sama. (McDermott n.d. a, 6–7; Kwast n.d., 2-3; Hoffman 2010, 9–13.)

Fresnel on termi, jota usein käytetään BRDF-funktioista puhuttaessa. Se ilmaisee tietystä kuvakulmasta nähdyn heijastuneen valon määrän (kuvio 7). Fresnel merkitään numeraalisesti välillä 0 ja 1 tai prosentteina välillä 0 %–100 %. (McDermott n.d. a, 7–8.) Materiaalille tyypillinen spekulari merkitään arvolla $F(0)$. Se saadaan laskemalla kohtisuoraan tulevan valon heijastama valon määrä. Lisäksi joissakin materiaaleissa, kuten metalleissa, Fresnel koostuu erillisistä RGB-arvoista, joten $F(0)$ arvoa voi nimittää myös spekularin väriksi. (Hoffman 2010, 13.)



Kuvio 7. Fresnel ilmaisee heijastuneen valon määrän (3D Coat n.d.).

Kuvio 7 havainnollistaa Fresnelin ja heijastusten suhdetta toisiinsa. Keskellä objektiä vain pieni osa valon säteistä heijastuu kohti katsojaa, kun taas reunoilta säteitä heijastuu enemmän, minkä vuoksi objektin reunat näyttävät vaaleammilta.

3.5 Valon aallonpituudet eli värit

Pinnan väriin vaikuttaa useampi tekijä: se mitä aallonpituuksia se heijastaa, mitä se absorboi sekä valonlähteen säteilemän valon väri. Tämä koskee sekä diffuusi- että spekulariheijastuksia, mutta riippuu materiaalista kumpi vaikuttaa havaittuun väriin. Esimerkiksi metalleilla ei ole diffuusiväriä, koska niillä ei tapahdu sirontaa, vaan niiden väri koostuu spekularista (Hoffman 2010, 14). Muilla materiaaleilla taas spekulari toimii lähes itsenäisesti valon aallonpituudesta. (McDermott n.d. a, 6.)

3.6 Johtavat ja eristävät materiaalit

Koska valo koostuu elektronisista ja magneettisista aalloista, vaikuttaa materiaalin sähkönjohtavuus valon reagointiin sen pinnalla. Siksi materiaalin tyyppi tulee ottaa huomioon tekstuureja tehdessä.

Yksinkertaisimmillaan materiaalit voi jakaa johtaviin metalleihin ja muihin, eristäviin materiaaleihin. Metallit heijastavat suurimman osan, noin 70%-100%, niiden pinnalle saapuvista valonsäteistä. Loput valosta katoaa metallin sisälle. Metallin ominainen väri

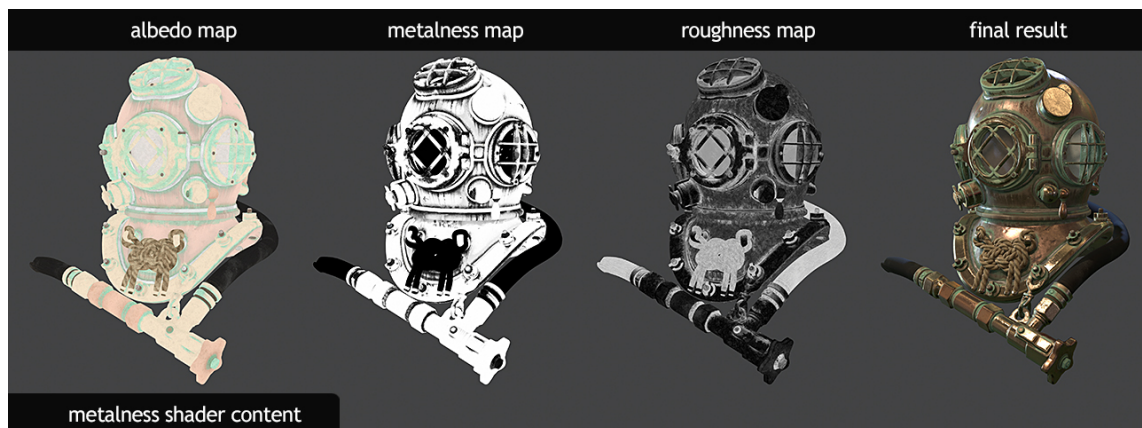
riippuu siitä, mitkä aallonpituudet jäävät heijastuksille. Tämä kuitenkin koskee vain puhdasta metallia, ruosteinen tai jonkin muun materiaalin peittämä metalli toimii kuten eristävä materiaali. Eristävät materiaalit heijastavat metalleja vähemmän valoa, mutta niiden sisälle päätyneestä valosta osa palautuu takaisin. (McDermott n.d. a, 8–9.)

4 PBR-menetelmät

Yleisimmin käytetyt PBR-menetelmät ovat metalli/rosoisuus (*metal/roughness*) sekä spekulaaari/kiiltävyys (*specular/glossiness*). Valittu menetelmä määrittelee sen, minkälaiset tekstuurikartat syötetään varjostimelle. Molemmilla on myös hyvät ja huonot puolensa, mutta tärkeintä on tehdä niiden työnkulkuun kuuluvat tekstuurikartat PBR:n ohjeistuksen mukaan. (McDermott n.d. b, 2.)

4.1 Metalli/rosoisuus-menetelmä

Metalli/rosoisuus-menetelmä on hyvin suoraviivainen, sillä siinä keskitytään itse materiaalin tyyppiin, eli onko materiaali metallia vai ei. Sen avulla voi myös välttyä joiltakin virheiltä realismin suhteen, koska heijastusten kontrolli on rajallista. Menetelmälle ominaiset tekstuurikartat ovat pohjaväri, metalli ja rosisuus (kuvio 8). (McDermott n.d. b, 5–6.)



Kuvio 8. Metallirosoisuus-menetelmälle ominaiset kartat ja niiden lopputulos (Wilson 2015).

Pohjavärikartta (*base color*) on RGB-kuva, joka sisältää tiedon eristävien objektien heijastamista väreistä eli albedosta. Siksi karttaa saatetaan kutsua myös albedoksi. Sen värit ovat hyvin tasaiset. Lisäksi ne vastaavat kirkkaudeltaan täydessä valossa nähtyä pintaa, minkä vuoksi ne saattavat näyttää kirkkaammilta kuin uskoisi. Karttaan

ei tulisi lisätä varjoja tai muutakaan valaistukseen liittyvää informaatiota, sillä kyseisiä ominaisuuksia varten on omat karttansa. (Maximov 2014; McDermott n.d. b, 5–6.) Mitä värien kirkkauteen tulee, niiden tulisi pysyä vähintään 30 ja 240 sRGB:n välillä, sillä täysin mustaa tai valkoista ei yleensä esiinny todellisuudessa. Mikäli materiaali sisältää metallisia kohtia, lisätään karttaan näihin kohtiin metallin heijastuksia määrittävä RGB-arvo. Yleensä tämä osuu RGB-skaalassa välille 180–255, mikä vastaa 70 %–100 % heijastavuutta. Joillakin metalleilla, kuten kullalla, tämä väri voi olla sävytetty. (McDermott n.d. b, 5–6, 8.)

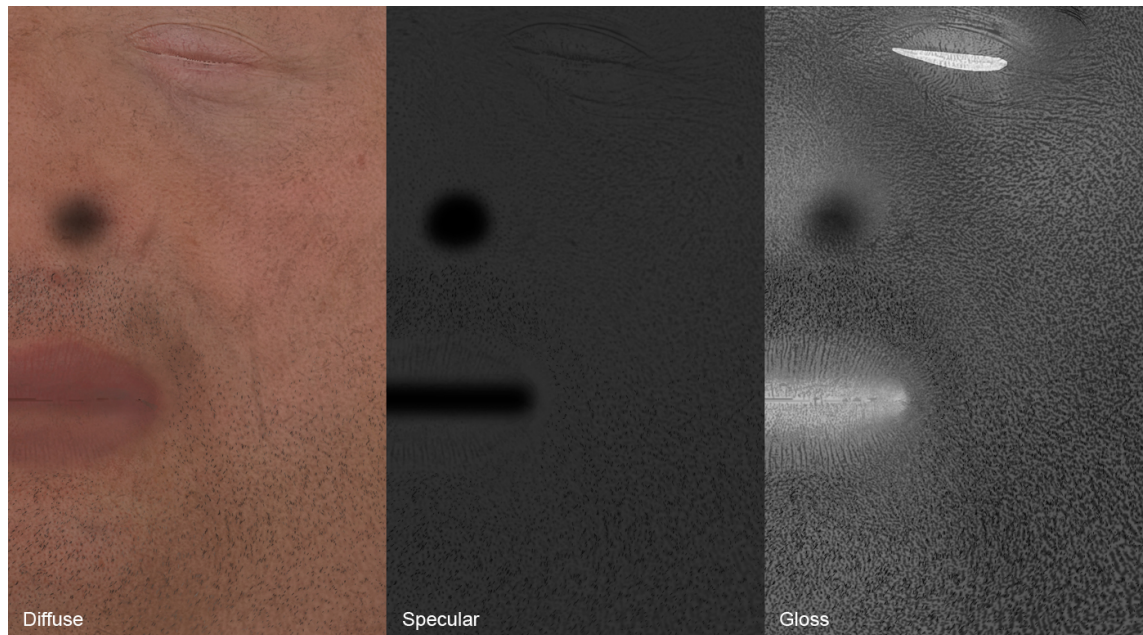
Metallikartta on yksinkertainen mustavalkoinen kuva, joka määrittää metalliset ja eristävät osat. Valkoinen tarkoittaa metallia ja musta muita materiaaleja. (Maximov 2014; McDermott n.d. b, 7.) Harmaasävyjä käytetään välitapauksissa, kuten likaisen metallin kanssa (Maximov 2014). Numeraalisesti sävyt merkitään mustan ja valkoisen välillä arvoilla 0 ja 1, tai RGB skaalan mukaan 0 ja 255. Metallikartta on tiiviissä suhteessa albedoon. Se kertoo mitkä kohdat albedosta määrittävät metallien heijastuksia ja mitkä muiden materiaalien väriä. Eristäville materiaaleille annetaan automaattisesti ennalta määrätty $F(0)$ arvo, mutta sen voi ohjelmasta riippuen vaihtaa. Jos tavoiteltu metalli ei ole puhdas, todennäköisesti sen väriä pitää tummentaa. Tämä pitää tehdä sekä metallikartassa että albedossa. Lisäksi jos metallia peittää esimerkiksi maalikerros, niin sitä ei enää lasketa metalliksi vaan eristäväksi materiaaliksi. (McDermott n.d. b, 4–5, 7, 9.) Jos metallia ei koko tekstuurista löydy, tai teksturoitava objekti koostuu pelkästään puhtaasta metallista, niin metallikartan voi korvata pelkällä numeraalisella arvolla (Maximov 2014).

Kaikista metallimenetelmän tekstuurikartoista rosoisuuskartta (*roughness map*) antaa eniten sijaa luovuudelle. Se on metallikartan tavoin mustavalkoinen, mutta sen avulla määritetään se, miten sileältä tai karhealta pinta näyttää. Tämä näkyy heijastusten tarkkuudessa. Musta vastaa tässä kartassa sileää pintaa ja valkoinen rosoista, eikä kartan tekemiseen ole oikeastaan sen tarkempia sääntöjä. Sen avulla voi myös simuloida naarmuja tai muita kulumisen merkkejä. (Maximov, 2014; McDermott n.d. b, 10.) Lisäksi kartan voi halutessaan kertoa syvennyksiä tummentavalla ambient occlusion -kartalla, sillä normaalisti nurkat ja syvennykset eivät kulu samalla tavalla kuin ulospäin pistävät osat (Maximov 2014).

4.2 Spekulaari/kiiltävyys–menetelmä

Spekulaarimenetelmässä kartat määrittävät samoja ominaisuuksia kuten metallimenetelmässä, mutta hieman eri tavalla. Diffuusi-, spekulaari- ja kiiltävyyskartat ovat ominaisia tälle menetelmälle (kuvio 9). Nimet saattavat muistuttaa vanhoissa menetelmissä käytettyjä karttoja, mutta PBR:ssä ne toimivat eri tavalla. (McDermott n.d. b, 13.) Spekulaarimenetelmässä keskeisin ominaisuus on spekulaarin täysi kontrolli. Tämä tuo tosin riskejä, sillä se antaa mahdollisuuden virheille.

Diffuusikartta (*diffuse map*) vastaa metallimenetelmän pohjaväriä, ja sitä voidaan myös kutsua albedoksi, mikä saattaa aiheuttaa hämmennystä. Se sisältää diffuusin värit eli albedon, mutta metallikohdat jätetään täysin mustiksi. Diffuusikartta toimii yhteistyössä spekulaarikartan (*specular map*) kanssa. Spekulaarikartta on värillinen RGB-kuva, johon sijoitetaan metallien heijastusarvot sekä eristävien materiaalien $F(0)$ arvot. Tämän ansiosta on mahdollista määritellä eri materiaaleille eri arvot. Koska eristävien materiaalien $F(0)$ arvot ovat yleensä hyvin alhaisia, minkä lisäksi niiden heijastukset ovat neutraalin värisiä, käytetään näiden materiaalien yhteydessä tummia harmaasävyjä. Artistin tulee kuitenkin ottaa huomioon diffuusikartan albedo, sillä se lasketaan yhteen spekulaarikartan datan kanssa. Näiden summa ei saisi ylittää arvoa 1, mikä vastaa 100 % tulevasta valosta. Siksi on suositeltavaa pyrkiä käyttämään mitattuja arvoja virheiltä välttymiseksi. Eristäville materiaaleille $F(0)$ arvo osuu tyypillisesti välille 40–75 sRGB-skaalassa. Metalleilla spekulaari on hyvin voimakasta, ja metallimenetelmän tavoin niiden arvot näkyvät hyvin vaaleina alueina kartassa. Lisäksi spekulaarin väri voi metalleilla olla sävytetty. (Maximov, 2014; McDermott n.d. b, 13–15.)



Kuvio 9. Spekulaarimenetelmälle tyypilliset tekstuurikartat (Cryengine 2014).

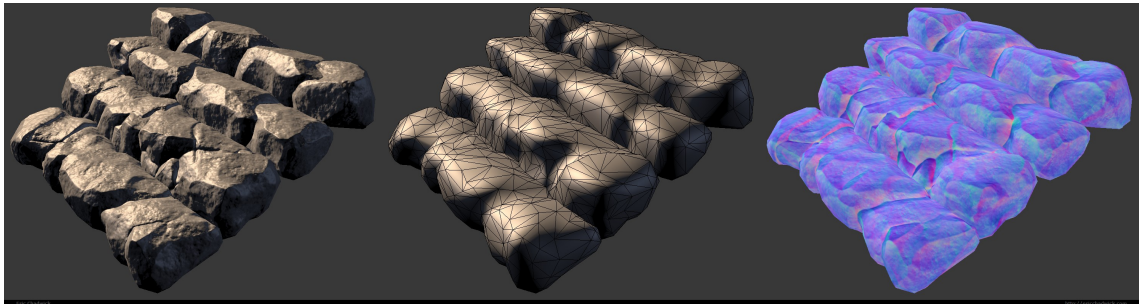
Kiiltävyyskartta (*glossiness map*) muistuttaa erehdyttävästi rosoisuuskarttaa, ja sen tarkoitus on myös sama; se määrittää pinnan kiiltävyyden vaikuttamalla sen heijastusten tarkkuuteen. Tärkein ero näiden kahden välillä on värien päinvastainen tarkoitus. Kiiltävyyskartassa valkoinen vastaa kiiltävää ja musta rosoista pintaa. (Maximov, 2014; McDermott n.d. b, 16.)

4.3 Yhteiset tekstuurikartat

Riippumatta valitusta PBR-menetelmästä on tekstuurikarttoja, joita voi käyttää missä tahansa materiaalissa. Eri tilanteisiin suunniteltuja tekstuurityyppejä on lukuisia. Tietyt niistä ovat kuitenkin onnistuneet vakiinnuttamaan asemansa yleisimmin käytettyjen tekstuurien joukossa.

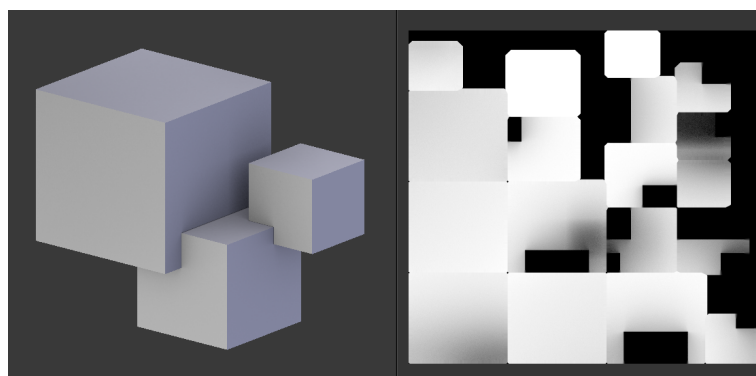
Bump mapit ovat tekstureja, joiden avulla luodaan illuusio yksityiskohdista 3D-mallin pinnalla. Vanhan mallinen bump map on mustavalkoinen kuva, jossa vaaleat värit vastaavat kohotettuja ja tummat syvennettyjä kohtia. (Russell 2014) Normaalikartta (*normal map*) on monimutkaisempi (kuvio 10). Se on RGB-kuva, jonka värikanavat vastaavat kutakin kolmesta suunta-akselista 3D-ympäristössä. Kun tavallinen bump map sisältää tiedon pinnan korkeusarvoista, normaalikartta kertoo pinnan suuntien eli normaalien vaihtelut jokaisella kuvan pikselillä. (Polycount n.d. a; Russell 2014) Normaalikarttoja on myös kahdenlaisia: tangent space ja object space. Tangent space-

kartat ovat hyviä malleissa, joissa geometria muuttuu animaation myötä. Siksi niitä käytetään muun muassa hahmoihin. Object space -kartat taas käyvät staattisissa malleissa, eivätkä ne vie aivan yhtä paljon resursseja. Normaalikarttoja on kuitenkin hankala tehdä käsin, lukuun ottamatta tiettyjä sitä helpottavia ohjelmia. Siksi normaalikartat tehdään yleensä leipomalla data suoraan korkearesoluutioisesta 3D-mallista. (Russell 2014) Normaalikartat ja height mapit eivät kuitenkaan muokkaa itse geometriaa, joten sivusta katsottuna mallin silhuetti pysyy samana.



Kuvio 10. Esimerkki normaalikartan käytöstä yksityiskohtien luomisessa (Chadwick 2015).

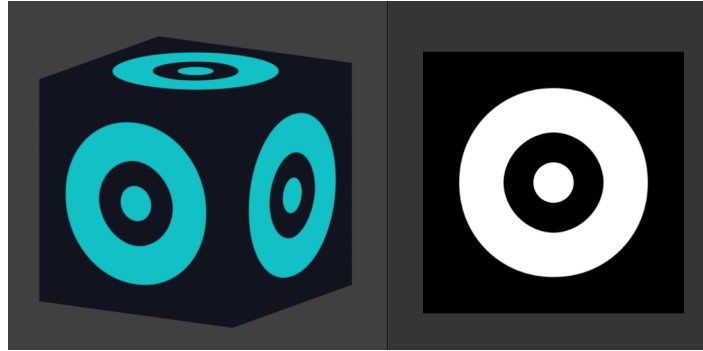
Ambient Occlusion, lyhyesti AO, on yleisesti käytetty kartta (kuvio 13). Se on mustavalkoinen, ja sen tummat osat luovat varjoja alueille, joihin ympäröivän valon on hankalaa päästä (McDermott n.d. b, 18; Polycount n.d. b; Wilson n.d.). AO yleensä leivotaan 3D-ohjelmassa käyttämällä simuloitua valaistusta, ja se saatetaan yhdistää jonkin toisen kartan kanssa joko tehosteeksi tai muistin säästämiseksi. Se ei kuitenkaan saisi peittää spekulaarivalaistusta. (Polycount n.d. b; Wilson n.d.)



Kuvio 11. Esimerkki Ambient Occlusion-kartasta.

Yksi erikoistilanteisiin luotu karttatyppi on emissio (*emissive map*). Sen erikoisominaisuus on hehkuvien osien simulointi (kuvio 12). Niissä kohdissa, joissa emissiokartta vaikuttaa, materiaali ei vastaanota mitään valoja tai varjoja. Tämä saa

pikselit näkymään täydellä kirkkaudella riippumatta ympäristön valoisuudesta, mikä luo illuusion hehkuvasta objektista. (Polycount n.d. c.) Emissiokartta voidaan toteuttaa useammalla tavalla, mutta yleisimmin käytetään erillistä mustavalkoista karttaa, jossa valkoinen vastaa hehkuvia osia. Haluttu hehkuvan osan väri tulee itse värikartasta.

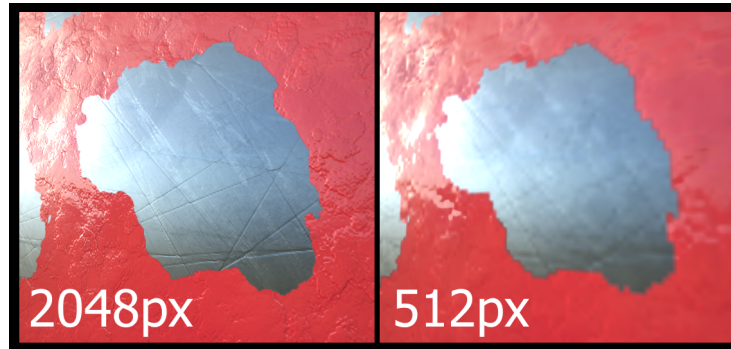


Kuvio 12. Esimerkki emissiokartasta ja sen käytöstä 3D-mallissa.

Kuviossa 12 kuution sivuilla näkyy emissiokartan vaikutus. Mustavalkoisen kuvan valkoinen kuvio näkyy täydellä kirkkaudella 3D-mallin päällä. Musta alue ei vaikuta käytettyyn materiaaliin, joten myös väri näkyy vain emissiokartan valkoisella alueella.

4.4 Tekstuurin resoluutio ja texelien tiheys

Tekstuurien laatuun vaikuttaa sekä käytetty resoluutio että tekselitiheys (*texel density*). Resoluutiolla viitataan tekstuurin kokoon, joka noudattaa yleensä kahden potenssia. Tekselillä taas tarkoitetaan tekstuurin pikseliä. Jotta lopputuloksena tekstuurit näyttäisivät tarkoilta ja mittasuhteiltaan oikeilta, tulee mallin ja tekstuurin olla oikean kokoisia suhteessa toisiinsa. Mitä enemmän tekstuurin pikseleitä 3D-mallin UV-kartta peittää, sitä suurempi tekselitiheys. Jos mallin UV-kartta on väärän mallinen suhteessa tekstuuriin tai tekstuuri on liian pieni, voi tekstuuri näyttää 3D-mallin päällä sumealta tai venyneeltä. (Diaz 2013.) Tämän vaikutuksen voi nähdä kuvioista 13, jossa pienemmän resoluution tekstuurit saavat materiaalit näyttämään epätarkoilta.



Kuvio 13. Esimerkki tekstuurien resoluution vaikutuksesta materiaaleihin.

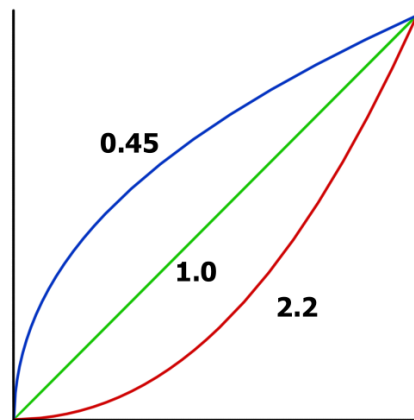
PBR-menetelmistä puhuttaessa huono resoluutio näyttäytyy myös ikävinä artefakteina eli visuaalisina virheinä. Virhe esiintyy metallisten ja ei-metallisten alueiden rajalla, johon ilmestyy silmännähtävä reuna. Tämä näkyy sekä metalli- että spekulaarimenetelmissä, vaikkakin pienellä erolla. Metallimenetelmässä reuna on valkoinen, kun taas spekulaarimenetelmässä se on musta. Tämä johtuu siitä, kun pohjavärikartan metalliset alueet interpoloivat muun tekstuurin kanssa. Paras tapa välttää tältä on tarpeeksi suuren resoluution ja tekselitiheyden käyttö tekstuureissa. (McDermott n.d. b, 11.)

4.5 Tekstuurit ja gammakorjaus

Tekstuurit ovat käytännössä ruudulla nähtäviä kuvia, minkä vuoksi niitä koskevat samat säännöt kuin kuvankäsittelyssä yleisesti. Varsinkin PBR-menetelmien yleistyttyä lineaarinen työnkulku (*linear workflow*) on tullut yhä tärkeämmäksi pelinkehityksessä. Lineaarisella työnkululla viitataan työtapaan, jossa kuvan gamma otetaan huomioon kuvaa käsiteltäessä (Pedersen n.d.). Tämä tapa juontuu vanhojen CRT-monitorien ajalta, koska kyseiset laitteet vääristävät kuvien kirkkautta (Sewell 2016). Lineaarisella työnkululla varmistetaan, että lopullinen kuva renderöidään ruudulle halutunlaisena.

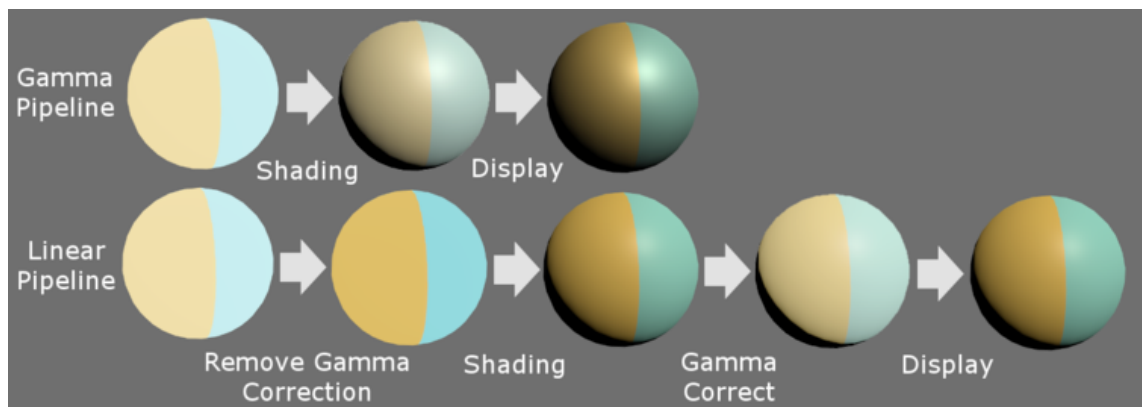
Yksinkertaistaen gamma on funktio, joka määrittelee kuvan numeerisen datan ja kuvan kirkkauden suhteen. Gamma-arvo ilmaistaan desimaalilukuna, ja sen suhdetta kirkkauteen kuvataan usein käyrällä. Arvolla 1.0 gamma esitetään suorana viivana, ja sitä kutsutaan lineaariseksi gammaksi. Tämä vastaa valon toimintaa reaali maailmassa, missä valon intensiteetti vastaa valon kirkkautta. (McHugh n.d.) Monitorit eivät kuitenkaan toimi näin, vaan ne muokkaavat kuvan gammaa. Yleensä monitorin gamma on 2.2, mikä tekee kuvasta tummemman. Siksi lineaariset kuvat täytyy tallentaa

gammakorjattuun formaattiin, kuten sRGB:hen, jonka gamma on päinvastainen. Tämän ansiosta kuva näkyy ruudulla lineaarisena. (Sewell 2016.)



Kuvio 14. Gammakäyrät ja niiden vastaavat numeeriset arvot.

Gammakorjattuja kuvia ei kuitenkaan voi käyttää sellaisenaan tietokoneohjelmissa, sillä tietokoneen laskutoimitukset tapahtuvat lineaarisessa avaruudessa. Gammakorjatut kuvat täytyy siis muuntaa takaisin lineaarisiksi laskutoimitusten ajaksi (kuvio 15). (Pedersen n.d.)



Kuvio 15. Vertailu yksinkertaisen ja lineaarisen työnkulun välillä (Sewell 2016).

Kuvio 15 havainnollistaa eroa gamman säilyttävän ja lineaarisen työnkulun välillä. Ilman gammakorjauksen poistamista ennen kuvan muokkaamista tulee lopullisesta kuvasta paljon tummempi kuin lineaarista työnkulkua noudattaen.

5 Substance Designer

Substance Designerin taustalla on Allegorithmic-niminen ranskalainen yritys, jonka perusti matemaatikko Sébastien Deguy vuonna 2003. Yrityksen tavoitteena oli luoda korkealaatuisia teksturointityökaluja, joiden pohjalla olisi Deguyn opiskeluaikana kehittämät matemaattiset kaavat. Vuosien aikana yritys kehitti useita teksturointiin liittyviä sovelluksia, ja vuonna 2010 se julkaisi ensimmäisen version Substance Designerista.

Viime vuosien sisällä Substance Designerista onkin tullut laajasti käytetty standardi PBR-materiaalien tekemiseen. Monet suurista peliyrityksistä on ottanut sen käyttöönsä projekteissaan, ja ohjelman käyttäjäkunta kasvaa edelleen. Ohjelman suosion taustalla saattaa olla sen vahva erikoistuminen juuri tekstuurien tekemiseen. Substance Designer onkin täynnä ominaisuuksia ja työkaluja, jotka helpottavat ja nopeuttavat tätä prosessia.

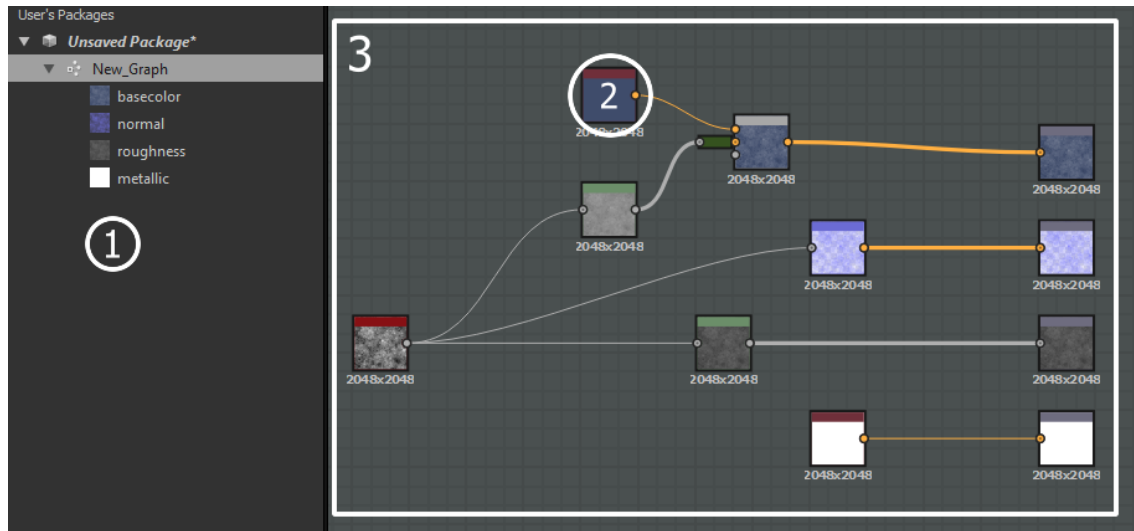
5.1 Substanssit

Erään määritelmän mukaan substanssi on ”älykäs tekstuurigeneraattori, jota voi käyttää sellaisenaan monissa ulkoisissa sovelluksissa” (Allegorithmic Documentation n.d, a). Se on oma tiedostomuotonsa, joka poikkeaa monellakin tapaa tavallisista tekstuureista. Ominaisin piirre sille on proseduraalisuus, mikä mahdollistaa sen muokkaamisen missä tahansa vaiheessa ilman, että aiemmin tehty työ valuisi hukkaan.

Substanssilla on oikeastaan kaksi eri tiedostomuotoa. Muokattavan tiedoston päätte on .SBS. Kun substanssi halutaan viedä johonkin muuhun ohjelmaan tai esimerkiksi pelimoottoriin, täytyy substanssista tallentaa .SBSAR-tiedosto. (Allegorithmic Documentation n.d. b.)

Substance Designerissa tarkasteltuna SBS-tiedosto sisältää yhden tai useamman graafin (*graph*) sekä näiden resurssit (kuvio 16). Graafit itsessään ovat solmuista (*node*) ja niiden välisistä linkeistä koostuvia verkkomaisia kokonaisuuksia. Solmujakin on erilaisia. Atomiset solmut (*atomic nodes*) ovat tiettyihin toimintoihin luotuja komponentteja. Instanssi (*instance*) on yksinkertaisesti graafi, joka on tuotu toiseen graafiin. Instanssia kutsutaan myös alagraafiksi (*sub-graph*). (Allegorithmic Documentation n.d. c.) Solmut ovat käytännössä se komponentti, mikä tekee

substanssin. Erilaisia solmuja valitsemalla ja yhdistelemällä voi saada aikaan hyvinkin erilaisia tekstuureja. Viimeisenä graafien hierarkiassa on yksi tai useampi tulossolmu (*output node*), johon graafin lopputulos liitetään jotta sitä voi käyttää graafin ulkopuolella.



Kuvio 16. Graafin resurssit, solmu ja itse graafi esiteltynä.

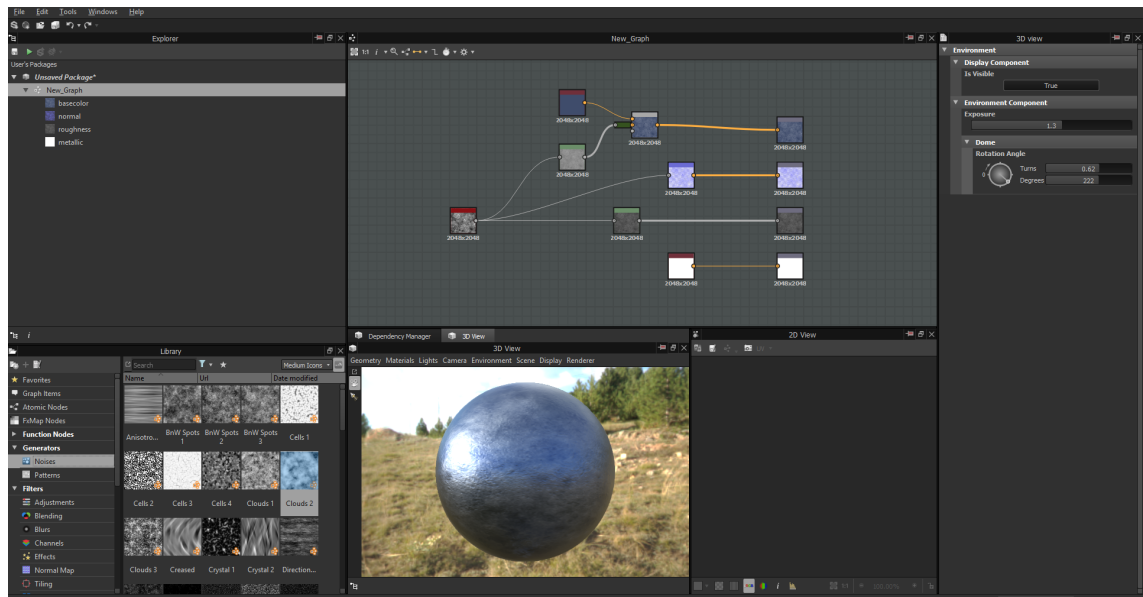
SBSAR-tiedostot sisältävät vain välttämättömimmän datan substanssien renderöintiin, ja tekstuurien sijaan ne pitävät sisällään pikselidataa. Niitä ei tämän vuoksi voi muokata esimerkiksi kuvankäsittelyohjelmissä. Lisäksi ne ovat sitä kevyempiä, mitä proseduraalisempia niiden sisältö on. (Allegorithmic Documentation n.d. c.)

Yksi tärkeimmistä piirteistä substansseissa on niiden tekijän mahdollisuus ”paljastaa” (*expose*) substanssin ominaisuuksia. Paljastettuja ominaisuuksia pystyy muokkaamaan lopullisessa sovelluksessa, jossa substanssia käytetään. Loppukäyttäjä voi näitä ominaisuuksia säätämällä tehdä erilaisia versioita alkuperäisestä substanssista, joten useampien tekstuurien tekeminen eri variaatioita varten ei ole tarpeellista. Paljastaminen tulisi jättää viimeiseksi tehtäväksi substanssin tekemisessä, sillä sen jälkeen ominaisuus lukittuu. (Allegorithmic Documentation n.d. d.)

5.2 Käyttöliittymä

Käyttöliittymä saattaa tuntua uudesta käyttäjältä sekavalta, mutta todellisuudessa se on hyvin organisoitu, minkä lisäksi sitä voi muokata halutessaan. Substance designerin oletusulkoasu muodostuu kuudesta ikkunasta: resurssinäkö (Explorer), kirjasto

(Library), graafinäkymä (Graph view), 3D-näkymä (3D View), 2D-näkymä (2D View) sekä parametrit (Parameters) (kuvio 17).



Kuvio 17. Substance Designerin käyttöliittymä.

Keskellä kuviota 17 näkyy ylempänä graafinäkymä, jossa substansseja varsinaisesti rakennetaan. Tähän näkymään tuodaan tarvittavat solmut ja linkitetään ne yhteen verkkomaisiksi rakenteiksi. Vasemmalla näkyvät resurssit ja kirjasto, kun taas oikealla puolella sijaitsee parametrien ikkuna.

6 Propin teksturointi

Tämä luku käsittelee toiminnallisessa osuudessa tehtyä henkilökohtaista projektia, sen varrella opittuja asioita sekä itse lopputulosta. Projektin tavoitteena oli teksturoida peliin soveltuva 3D-malli PBR-materiaaleilla ja sen avulla havainnollistaa PBR-tekstuurien tekemisprosessia.

Koska mahdollisia materiaaleja on monenlaisia, päätettiin projektissa tehdä useampaa erilaista materiaalia soveltuva objekti, jossa kukin materiaali tulisi nähtävästi esille. Tarkoituksena oli pystyä tutkimaan ja esittämään yhteneväisyyksiä ja eroja erilaisten materiaalien tekotavoissa. Projektia varten suunniteltiin ja toteutettiin fantasiahenkinen peliympäristöön soveltuva 3D-kilpi, jolle luotiin puu-, metalli- ja kristallimateriaalit (kuvio 18).

Työkalut rajattiin Substance Designer 5 -ohjelmaan, mikä on yksi yleisimmin käytetyistä tuotteista tekstuurien luomiseen. Lisäksi testaukseen käytettiin Unity-pelimoottoria. Opinnäytetyön ulkopuolelle jätettiin itse objektin mallinnus ja siihen liittyvät osuudet, sillä tavoitteena oli keskittyä ainoastaan teksturointiin.



Kuvio 18. Luonnos kilvestä ja valmiit low poly ja high poly -mallit.

Kuviossa 18 on esiteltyä alkuperäinen piirros toteutettavasta kilvestä. Suunnitelmasta näkyy materiaalien värit ja sijoittelu. Luonnoksen vieressä on valmis teksturoitava low poly -malli, josta voi havaita joidenkin yksityiskohtien puuttumisen. Näitä varten tehtiin erikseen high poly -malli, mikä näkyy kuviossa oikealla.

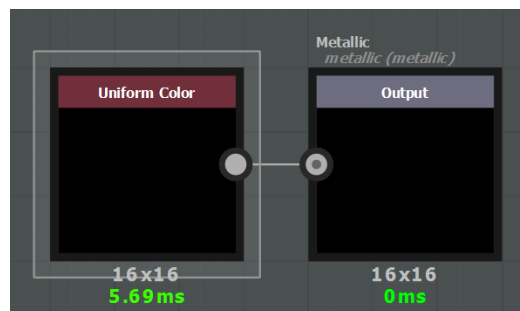
6.1 Proseduraalisten materiaalien tekeminen

Projektissa sovellettiin materiaalien osalta metalli/rosisuus-menetelmää sen yksinkertaisuuden vuoksi. Kyseisessä menetelmässä varjostin pitää huolen heijastuksista ja spekulaarista, joten lopputulos noudattaisi reaalimaailman fysiikkaa (McDermott n.d. b, 4). Tästä huolimatta materiaalien tekemisessä piti pyrkiä löytämään oikeita arvoja kuvaamaan materiaalien ominaisuuksia, tässä tapauksessa Fresnel-arvojen sijaan sRGB-lukuja. Aiheesta löytyykin erilaisia kaavioita ja taulukkoja, mutta moni niistä ei ole tarpeeksi kattava ja jotkut niistä on tehty tietyissä ohjelmissa käytettäväksi, joten ne eivät välttämättä toimi valitussa sovelluksessa. Projektin aikana löydettyä informaatiota käytettiin lähinnä suuntaa antavana, ja sen ohella käytettiin

muiden tekemiä materiaaleja ja valokuvia referenssinä. Muun muassa Allegorithmicin perustama Substance Share-verkkosivusto sisälsi paljon analysoitavaa materiaalia. Tärkeintä oli materiaalin uskottavuus ja esteettisesti hyvä lopputulos.

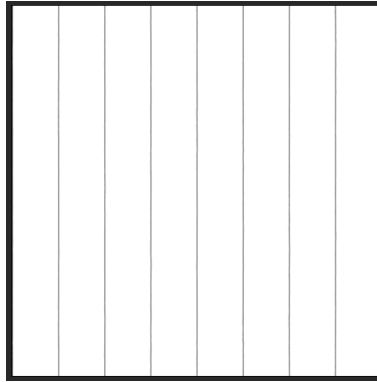
6.1.1 Puu

Puu on eristävä, matta materiaali, joten sen metallisuusarvo määriteltiin kauttaaltaan nollassa. Tämä tehtiin asettamalla Metallic-tulossolmun väri mustaksi liittämällä siihen yksivärinen Uniform Color-solmu (kuvio 19). Koska kyseessä oli yksivärinen kuva, pystyi sen määrittelemään erittäin pieneksi, mikä vähentää sen vaatimaa muistia. Resoluutioksi valittiin 16 X 16 pikseliä Allegorithmicin ohjeiden mukaisesti (Allegorithmic Documentation n.d. a).



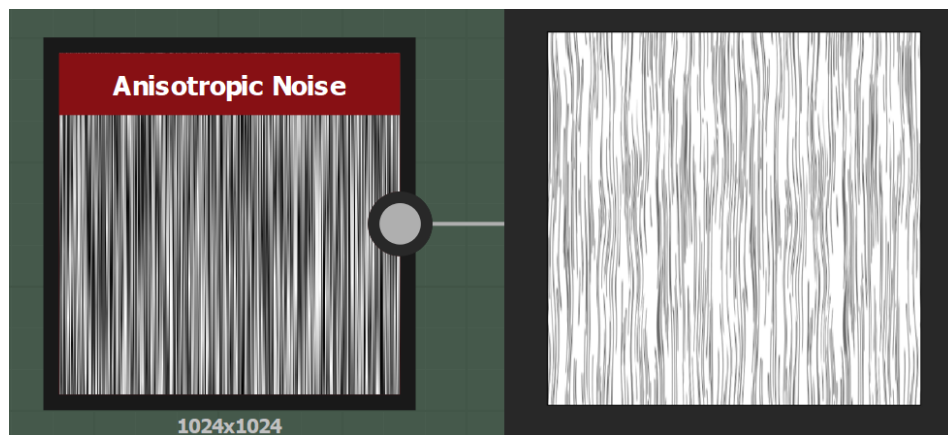
Kuvio 19. Metallic-kartan Output-solmuun liitetty musta Uniform Color -värisolmu.

Puumateriaalin oli tarkoitus simuloida vierekkäisiä lankkuja, joten niiden välisiä uurteita varten graafiin lisättiin viiva *Stripes*-solmulla (ks. liite 1). Viivasta muokattiin värimuokkauksen jälkeen pystysuora viivarivistö *Transformation 2D*-solmulla kääntämällä kuvaa 90 astetta ja pienentämällä kuvaa. Proseduraalisuuden ansiosta kuva monistaa itseään ulospäin. Koska kyseessä oli orgaaninen ja pehmeä materiaali, niin lopputulosta muokattiin vielä *Warp* nimisellä solmulla, joka sekoittaa annetun kuvan pikseleitä toisen kuvan datan pohjalta. Näin uurteista saatiin luonnollisen epätasaiset (kuvio 20).



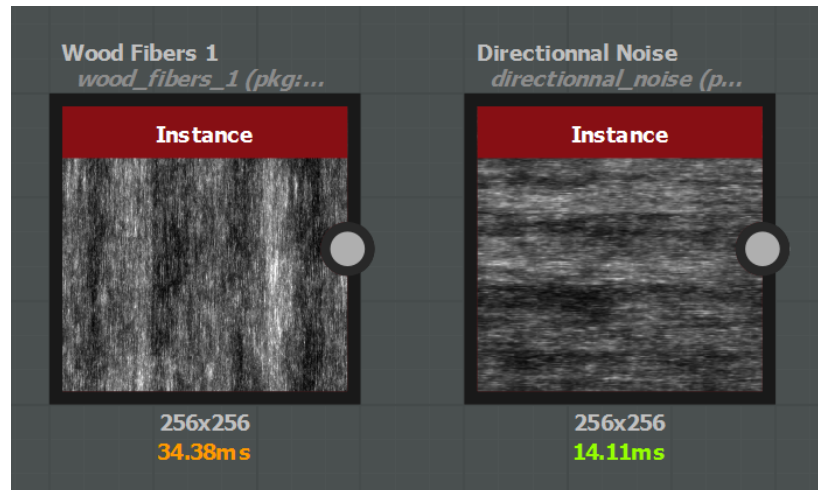
Kuvio 20. Lankkujen kuvio.

Puun aaltomainen pintakuviointi tehtiin muokkaamalla Anisotropic Noise -solmua (kuvio 21). Kyseisen solmun viivakuviointi on valmiiksi epätasaisen paksua, mikä soveltui hyvin orgaaniseen materiaaliin. Ensin tekstuurin kontrastia kasvatettiin Levels-solmulla, minkä jälkeen kuviolle tyypillinen aaltomaisuus saatiin aikaan toisella Warp-solmulla. Kuvan kokoa pienennettiin sivusuunnassa ja väriä vaalennettiin ennen yhdistämistä lankku-kuvion kanssa, johon käytettiin Blend-solmua (ks. liite 1).



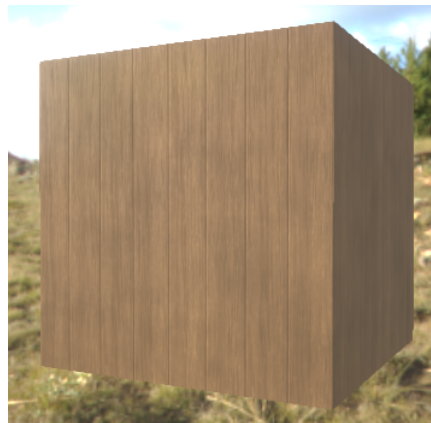
Kuvio 21. Anisotropic Noise -solmu ja siitä johdettu aaltomainen kuvio.

Puulle on ominaista kuituisuus, ja Designerista löytyikin erikseen Wood Fibers -solmu. Kyseinen solmu oli kuitenkin raskaampi kuin hyvin samankaltainen Directional Noise, joten materiaalissa päädyttiin käyttämään jälkimmäistä (kuvio 22). Kiertämällä kuvio pystyasentoon, pienentämällä sitä ja muokkaamalla sen kontrastia teki solmusta sopivan tarkoitukseensa.



Kuvio 22. Wood Fibers 1 -solmun prosessointi kestää kauemmin kuin Directional Noisen.

Tämän jälkeen kaikki kolme tehtyä mustavalkoista tekstuuria yhdistettiin kahdella Multiply-tilan Blend-solmulla. Lankkujen kuviota vaalennettiin ensin Levels-solmulla ennen Blend-solmuun liittämistä (ks. liite 1).



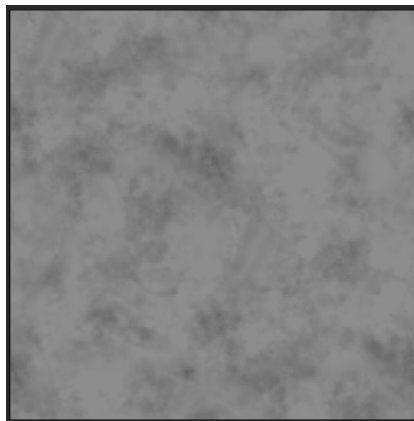
Kuvio 23. Valmis puumateriaali tarkasteltuna kuution pinnalta.

Yhdistetystä mustavalkoisesta tekstuurista johdettiin rosoisuus- ja normaalikartat, minkä jälkeen tehtiin albedo sekoittamalla sama mustavalkoinen tekstuuri erillisen värillisen kuvan kanssa. Koska materiaalit ovat harvemmin tasavärisiä, tehtiin värillinen kuva yhdistämällä Dirt- ja Clouds-solmuista sotkuinen mustavalkokuva, joka väritettiin Gradient Map -solmulla (ks. liite 1). Yhdessä tarkasteltuna tekstuurikartat muodostivat valmiin puumateriaalin (kuvio 23).

6.1.2 Metalli

Metallimateriaalin tekemisessä oli vain pari oleellista eroa puuhun verrattuna. Ensinnäkin sen pohjaväriksi riitti tasainen väri, sillä yksityiskohdat ja värierot tulisivat muista tekstuurikartoista. Toiseksi tämä materiaali vaati metallikartan määrittävän metalliset alueet, joten yksivärinen kartta ei tässä tilanteessa käynyt. Värin määrittämiseksi käytettiin harmaata Uniform Color -solmua.

Metallin haluttiin näyttävän kuluneelta, joten sitä varten tehtiin harmaasävyinen kuva (kuvio 24). Ensin yhdisteltiin erilaisia generaatiosolmuja keskenään, kuten Clouds ja Liquid, joita myös muokattiin Levels- ja Transformation 2D -solmuilla. Tässä käytettiin enemmän luovaa kuin suunnitelmallista lähestymistapaa. Lopputulos piti kuitenkin muokata tilettäväksi, koska kuvion tuli olla monistettavissa ilman näkyviä reunoja jotta materiaali olisi proseduraalinen. Designerista löytyy valokuvia varten suunniteltu *Make It Tile Photo*- solmu, jonka avulla kuvioista saatiin tilettävä (ks. liite 2).



Kuvio 24. Metallimateriaalin alku.

Jotta materiaali olisi vieläkin mielenkiintoisempi, päätettiin siihen lisätä vielä naarmuja. Designerista löytyy sitäkin varten oma solmunsa nimeltä Scratches Generator. Kuvan värit piti kuitenkin kääntää Invert-solmulla siten, että naarmut olisivat mustia. Levels-solmun avulla naarmuja muokattiin pienemmiksi, kun taas Blur-solmulla pehmennettiin tästä seurannut liiallinen terävyys. Tämä kuva yhdistettiin edellisen graafin kanssa Multiply Blend -solmulla. Naarmujen reunoja korostamaan tehtiin vielä oma kuva lisäämällä graafiin Highpass Grayscale -solmu, josta saatiin kontrastia muokkaamalla reunojen heijastukset. Naarmujen kuvio yhdistettiin aiemmin tehtyyn tekstuurikuvioon käyttäen Add (Linear Dodge) Blend -solmua (ks. liite 2).



Kuvio 25. Lopullinen metallimateriaali.

Tästä viimeisestä Blend-solmusta johdettiin lopulliset tekstuurit käyttäen Levels-solmuja. Kuvan tekstuuri ja epätasainen väritys piti huolen siitä, että lopullinen materiaali näytti luonnollisen kuluneelta (kuvio 25).

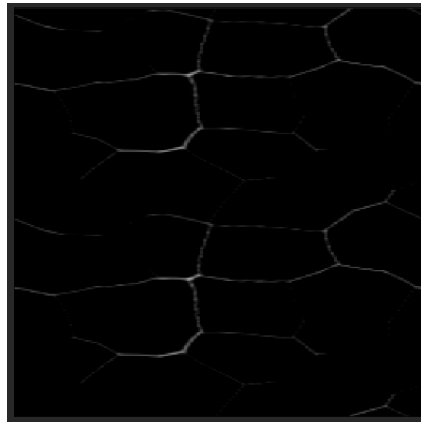
6.1.3 Hohtava kristalli

Kolmas materiaali oli edellisiin verrattuna hieman erilainen, sillä sen piti esittää kristallimaista pintaa. Kristalli on kiiltävää ja lasin tavoin läpinäkyvää. Kuitenkin peleissä käytetään läpinäkyvyyttä harkiten, sillä läpinäkyvät objektit tarkoittavat pikselien täyttämistä useampaan kertaan ja siten vaativat enemmän tehoja laitteelta (Pruett 2014). Siksi tästä materiaalista jätettiin läpinäkyvyys pois ja keskityttiin muihin piirteisiin. Mallina käytettiin muun muassa erilaisia jalokiviä (kuvio 26).



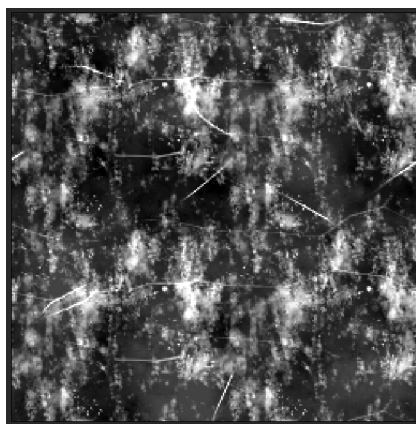
Kuvio 26. Kuva referenssinä käytetystä ametystistä (Ochel 2016)

Materiaalin tekeminen aloitettiin pinnan epätasaisuuksista. Nähin lukeutuivat kristallimaisille materiaalille tyypilliset halkeamat. Graafin ensimmäiseksi solmuksi valittiin Cells 3 -solmu, jonka kuviota muokattiin Levels- ja Blur HQ Grayscale -solmuilla (ks. liite 3). Graafia jatkettiin Warp-solmulla, jonka avulla kuvion viivoista tehtiin kaarevampia, ja Histogram Scan- solmulla luotiin satunnaista kontrastin vaihtelua. Lopuksi kuvan värit käännettiin toisin päin Invert-solmulla niin että halkeamista saatiin valkoisia, sillä lasimaisissa objekteissa halkeamien reunojen heijastukset saavat ne usein näyttämään vaaleilta (kuvio 27).



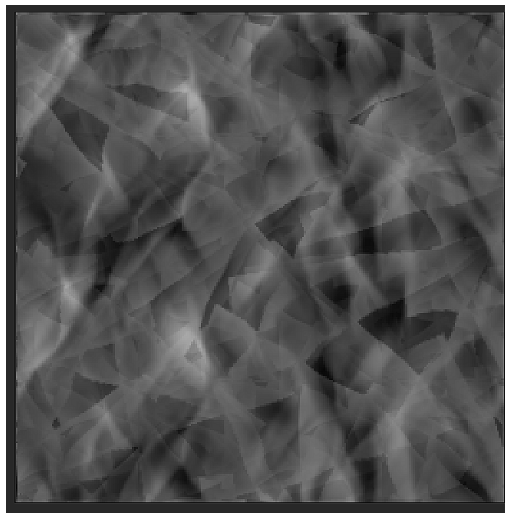
Kuvio 27. Halkeamien kuvio.

Muita yksityiskohtia varten sekoitettiin Grunge Map-, Scratches Generator- ja Moisture Noise -solmut keskenään Blend-solmuilla, minkä jälkeen halkeamat liitettiin osaksi tätä graafia. Tästä syntyneestä kuvasta saatiin Levels-solmuilla muokattua normaali- ja rosoisuuskartoille sopivat versiot (kuvio 28).



Kuvio 28. Kulumia varten tehty mustavalkoinen tekstuuri.

Koska kristalli on läpikuultavaa materiaalia, niin sen pinnan alla olevan kuvioinnin ja heijastukset voi nähdä. Referenssejä tarkastellessa kristallimaisten materiaalien kuviointi näytti olevan terävää ja siksakkimaista, ja Substance Sharesta löydettyistä valmiista substansseista sai hyvin vinkkejä tämänlaisen kuvion tekemiseen. Ensin Gradient Linear-solmusta tehtiin Splatter-solmulla siksakkimainen kuvio, minkä jälkeen graafiin lisättiin Crystal-solmu, jota muokattiin Transformation 2D ja Levels solmuilla (ks. liite 3). Näille kahdelle kuviolle annettiin enemmän aaltomaisuutta Vector Warp-solmulla käyttämällä Perlin Noisea määrittelevänä karttana. Siksakkimaisesta kuviosta muokattiin tiheämpi versio, joka sekoitettiin edellisen kuvion kanssa lisäämään yksityiskohtaisuutta ennen yhdistämistä toisen kristallikuvion kanssa (kuvio 29).



Kuvio 29. Kristallin pinnanalaisen kuvionnin tekstuuri.

Väri lisättiin yhdistämällä kristallikuvio yksiväriseseen Uniform Color -solmuun Multiply-asetuksella. Tämän lisäksi Uniform Color -solmusta johdettiin uusi Output-solmu *Emissive*, joka tekee materiaalista hohtavan näköisen. Jotta sekä kristallin että sen emission väriä pystyisi säätämään graafin ulkopuolellakin, paljastettiin Uniform Color -solmun Background-asetus. Tämä lisäsi koko graafin parametrien listaan erillisen parametrin tälle värille. Asetuksiksi valittiin Float3 ja Color (RGB), jolloin kaikki RGB-värit ovat käytössä (ks. liite 3.)

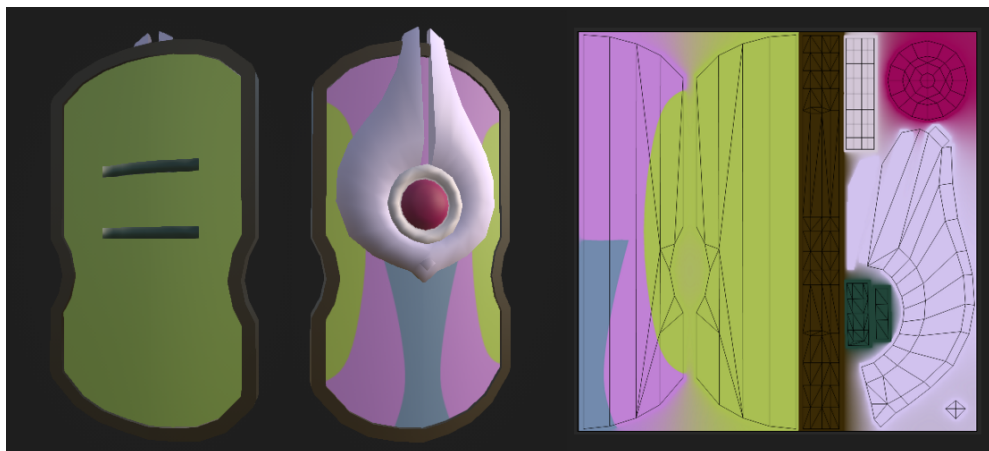


Kuvio 30. Lopullinen kristallimateriaali.

Koska kristalli on eristävä materiaali, niin metallikartta jätettiin mustaksi käyttäen Uniform Color -solmua. Kuviosta 30 voi nähdä, miltä lopullinen kristallimateriaali näyttää Substance Designerissa katseltuna.

6.2 Leivotut tekstuurikartat

Lopullinen materiaali tarvitsi vielä leivotut tekstuurit. Leivotuilla tekstuureilla low poly -kilven pinta saadaan muistuttamaan mudoiltaan ja yksityiskohdiltaan high poly -mallia. Leipomista varten luotiin uusi ja viimeinen SBS-tiedosto ja graafi, johon linkitettiin sekä high- että low poly -mallit. low poly -mallin toimintovalikosta avattiin Scene Information Baking-ikkuna, jonka Bakers-listaan lisättiin generoitavat kartat (ks. liite 4). Vaihtoehdoista valittiin Color ID eli väritunniste-, normaali- AO-kartat. Karttojen asetuksissa määriteltiin high-poly objekti ja tehtävän kartan resoluutio eli Output Size. Väritunnistekarttaa oli tarkoitus käyttää maskin tavoin, jotta low poly -mallin pinnan alueita pystyi muokkaamaan riippumatta geometriasta (kuvio 31).

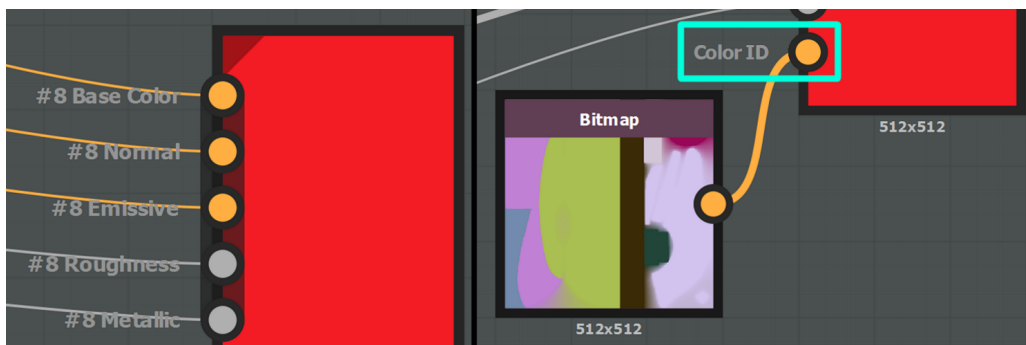


Kuvio 31. Värikartta tarkasteltuna sekä low poly -mallin päällä että UV-kartan kanssa.

Väritunnistekartan asetuksissa määriteltiin värin lähteeksi Submesh ID, jolloin jokainen erillinen high-polyn osa tarkoittaisi eri värejä. Asetusten jälkeen tarvitsi vain painaa OK-painiketta ja Substance Designer generoi määritetyt kuvat.

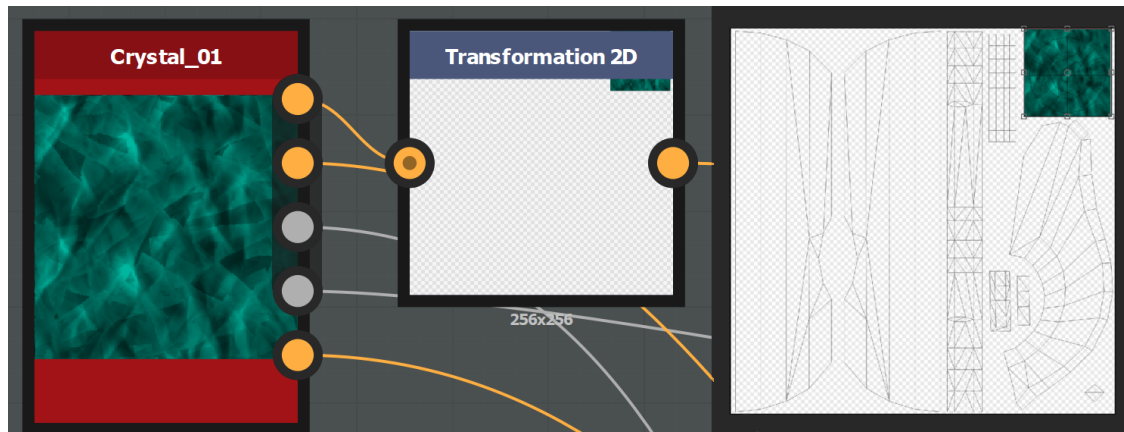
6.3 Lopullisen materiaalin kokoaminen ja testaus

Lopullisen materiaalin kokoamista varten kaikki aiemmin tehdyt substanssit tuotiin viimeksi tehtyyn tiedostoon instansseina. Näiden yhdistämiseksi luotiin Multi-Material Blend -solmu, myöhemmin MMB-solmu, jonka vastaanottaviin kanaviin lisättiin emissio. Lisäksi tehtävällä objektilla oli useampia metallisia alueita, joissa metalli-materiaalin ominaisuuksia piti voida muokata erikseen, joten materiaalien kokonaisuudeksi määriteltiin kahdeksan. Alimpaan syötteeseen liitettiin leivottu väritunnistekartta, josta poimittiin värit MMB-solmun materiaalien maskeiksi (kuvio 32). Tämän jälkeen instanssien tekstuurikartat linkitettiin syötteisiin MMB-solmussa.



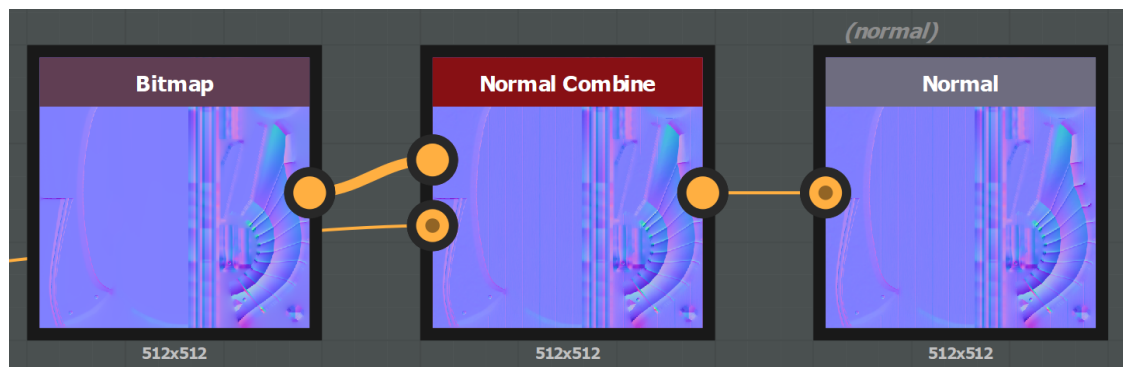
Kuvio 32. Lähikuva Multi-Material Blend-solmusta ja siihen liitetystä värikartasta.

Materiaali-instansseja piti muokata lisäämällä solmuja instanssin ja MMB-solmun välisiin linkkeihin (ks. liite 5). Kristallin tekstuurit pienennettiin ja sijoitettiin UV-kartan mukaisesti oikeaan kohtaan, sillä sellaisenaan materiaali levittäytyi koko UV-kartan kokoiseksi (kuvio 33). Metallista johdettiin vaaleampi versio tiheämmällä kuvioinnilla käyttämällä HSL- ja Transformation 2D -solmuja. Puun tekstureja siirrettiin sivusuunnassa niin, että lankkujen uurteet sijoittuivat paremmin mallin pinnalla. Tässä välissä koettiin tarpeelliseksi paljastaa myös puun väri mahdollista muokkaamista varten.



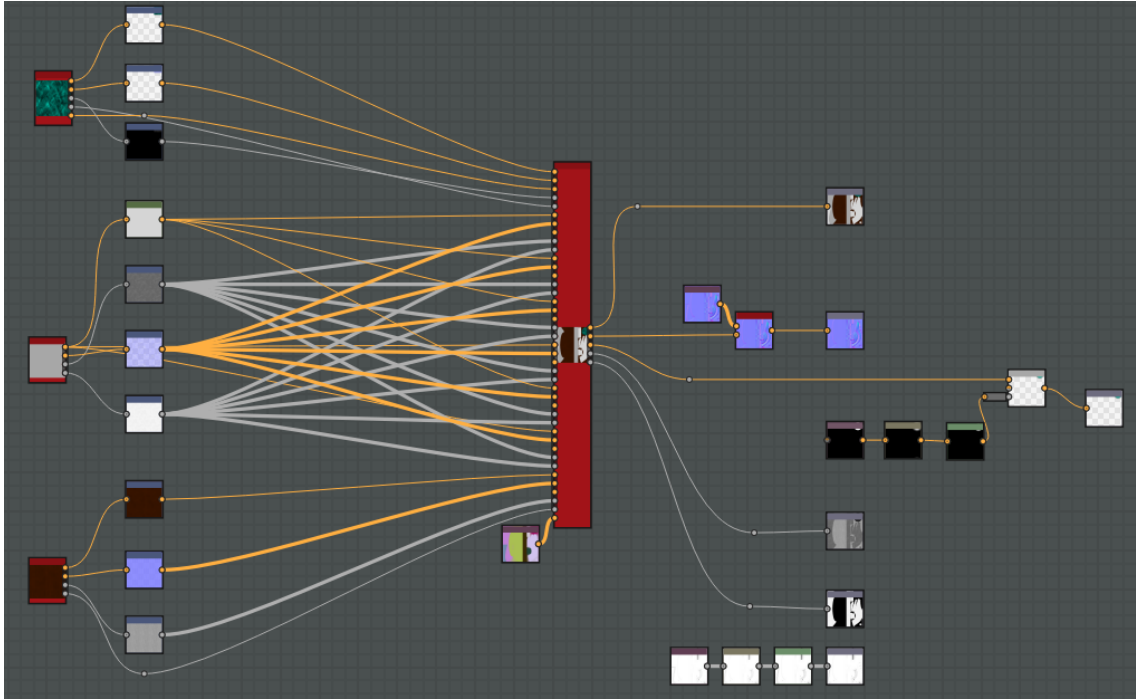
Kuvio 33. Kristallin materiaali sijoitettiin oikealle paikalle UV-kartassa.

Graafi alkoi jo olla melko monimutkainen, mutta lopulliset kartat piti vielä liittää Output-solmuihin. Väri-, rosoisuus- ja metallikartat liitettiin vastaavista MMB-solmun syötteistä suoraan Output-solmuihin ilman muokkauksia. Leivottu normaalikartta piti yhdistää materiaaleista koostettuun karttaan, joten ne piti yhdistää Normal Combine-solmulla, joka toimii samalla periaatteella kuin Blend-solmu (kuvio 34).



Kuvio 34. Normaalien yhdistäminen Normal Combinella.

Tämän lisäksi emission haluttiin hälvenevän ulospäin. Tämä saatiin maskaamalla kuva Blend-solmun ja siihen liitetyn mustavalkoisen kuvan avulla. Leivottua AO:ta muokattiin Blur- ja Levels-solmulla (ks. liite 5).



Kuvio 35. Lopullinen graafi sisältäen kaikki materiaalit ja tekstuurit.

Lopullinen substanssin graafi näyttää kuvion 35 mukaiselta. Jos kaikki materiaalit olisi tehty suoraan yhdessä tiedostossa instanssien käyttämisen sijaan, olisi graafista tullut paljon monimutkaisempi, eikä materiaaleja olisi voinut käyttää yksinään missään muussa projektissa. Substance Designerissa tarkasteltuna materiaalit näyttivät kilven päällä suunnitelman mukaisilta (kuvio 36).



Kuvio 36. Lopulliset tekstuurit kilven päällä Designerissa tarkasteltuna.

Substanssista tallennettiin SBSAR-versio, joka vietiin Unity-pelimoottoriin yhdessä kilpiobjektin kanssa tarkasteltavaksi. Materiaalien värit kuitenkin poikkesivat huomattavasti siitä, miltä ne näyttivät Designerissa (kuvio 48).



Kuvio 37. Materiaalit ensimmäisen kerran Unityssa tarkasteltuna.

Puun värin sai muutettua Unityssa sen paljastamisen ansiosta, mutta metallin väri piti muokata Designerissa. SBSAR-tiedostosta piti tämän vuoksi tehdä uusi versio. Unityssa testattiin myös kristallin värin muuttamista, mikä toimi moitteetta sen paljastamisen ansiosta. Nyt kilpi oli materiaaleineen käyttövalmis (kuvio 38).



Kuvio 38. Kilven lopullisen materiaalin tarkasteleminen Unityssa.

7 Yhteenveto

Opinnäytetyön tavoitteena oli luoda tekstuurit peliin soveltuvalle 3D-mallille noudattaen PBR:n työnkulkua. Tähän käytettiin PBR-materiaalien tekemiseen erikoistunutta ohjelmaa Substance Designeria. Lopputuloksena syntyi kolme eri materiaalia käyttävä 3D-kilpi, jota voi käyttää pelimoottorissa. PBR-materiaalien ansiosta objektin pinnat näyttävät uskottavilta, ja substanssin ominaisuuksien avulla tekstuurit ovat helposti muokattavissa.

Opinnäytetyössä opittuja asioita tarkastellessa PBR ja sen mukaan työskentely on monimutkaista, muttei kuitenkaan vaikeaa. Tekstuuriartistien on jo ennen PBR:ää täytynyt opiskella materiaalien ominaisuuksia pystyäkseen luomaan uskottavia objekteja digitaaliseen ympäristöön, mikä on vaatinut muun muassa tarvittavan fysiikan tuntemusta. Tästä näkökulmasta katsottuna PBR tarkoittaa artistin kannalta lähinnä jo olevien käytäntöjen päivittämistä. PBR:n soveltaminen vaatii kuitenkin materiaalien tuntemuksen lisäksi ymmärrystä käytettävistä työkaluista, mitkä saattavat Substance Designerin tavoin olla teknisempiä kuin perinteiset Photoshopin käyttöön pohjautuvat menetelmät.

Teknisyydestään huolimatta Substance Designer on osoittautunut erittäin monipuoliseksi ja pelinkehityksen kannalta hyödylliseksi välineeksi. Tekstuurien automaattinen generointi solmuilla ja niiden helppo muokattavuus tarkoittavat entisiin menetelmiin verrattuna nopeampaa tekstuurien tekemistä. Lisäksi tekstuurien proseduraalisuus ja ominaisuuksien paljastaminen mahdollistavat lukuisten eri versioiden tekemisen, ja pelisuunnittelija pystyy tekemään mahdolliset muutokset niihin suoraan pelimoottorissa sen sijaan, että hänen tarvitsee pyytää artistia tekemään ne.

Vaikka opinnäytetyön painopiste oli peligrafiikassa, soveltuvat siinä käytetyt tekniikat ja ohjelmat moneen muuhunkin. PBR:ää voi hyödyntää missä tahansa tuotannossa, joka käyttää 3D-grafiikkaa, kuten esimerkiksi arkkitehtuurivisualisoinnissa, mainoksissa ja elokuvissa. PBR:n hyödyt eivät kuitenkaan rajoitu teksturointiin. Se muun muassa nopeuttaa tuotantoprosessia, koska artistien tarvitsee tehdä vähemmän tekstuurikarttoja kuin vanhoilla menetelmillä. PBR ei kuitenkaan ole mikään ihmelääke, joka tekee materiaaleista automaattisesti realistisia, vaan se vaatii artistin, joka tietää mitä tekee.

Lähteet

Allegorithmic Documentation n.d. a. What is a "Substance" file?. [Verkkodokumentti].

<<https://support.allegorithmic.com/documentation/pages/viewpage.action?pagelId=129368143>>

(luettu 19.1.2017).

Allegorithmic Documentation n.d. b. Substance File Formats. [Verkkodokumentti].

<<https://support.allegorithmic.com/documentation/display/SD5/Substance+File+Formats>>

(luettu 19.1.2017).

Allegorithmic Documentation n.d. c. Nodes Reference. [Verkkodokumentti].

<<https://support.allegorithmic.com/documentation/display/SD5/Nodes+Reference>>

(luettu 19.1.2017.).

Allegorithmic Documentation n.d. d. Expose a Parameter. [Verkkodokumentti].

<<https://support.allegorithmic.com/documentation/display/SD5/Expose+a+Parameter>>

(luettu 19.1.2017).

Burke, Steve 2015. Defining PBR (Physically-Based Rendering) Graphics with Crytek Developers. [Verkkodokumentti].

<<http://www.gamersnexus.net/gg/1866-what-is-pbr-cryengine-star-citizen>>

(luettu 18.2.2016).

Diaz, Tim 2013. Texel Density for game art. [Verkkodokumentti].

<<https://www.3dtotal.com/tutorial/1813-texel-density-for-game-art-3ds-max-unreal-dev-kit-by-tim-diaz-udk-bricks-tutorial-environment>>

(luettu 17.4.2016).

Hearn, Donald & Baker M. Pauline 1997. Computer Graphics, C Version. 2. painos. Cambridge. Pearson.

Hiukkastiето n.d. Absorbtiosta. [Verkkodokumentti].

<<http://www.hiukkastiето.fi/en/taxonomy/term/37>>

(luettu 18.1.2016).

Hoffman, Naty 2010. Physically-Based Shading Models in Film and Game Production. [PDF-dokumentti].

<http://renderwonk.com/publications/s2010-shading-course/hoffman/s2010_physically_based_shading_hoffman_a_notes.pdf>

(luettu 17.01.2016.).

Kwast, Daniël nd. An Introduction to BRDF Models. [PDF-dokumentti].

<<http://hmi.ewi.utwente.nl/verslagen/capita-selecta/CS-Jimenez-Kwast-Daniel.pdf>>

(luettu 13.6.2016).

Lagarde, Sébastien 2011. Feeding a physically based shading model. [PDF-dokumentti]

<<https://seblagarde.wordpress.com/2011/08/17/feeding-a-physical-based-lighting-mode/>>

(luettu 14.2.2016).

Maximov, Andrew 2014. Physically Based Texturing for Artists. Art Is A Verb. [Verkkodokumentti].

<<http://www.artisaverb.info/PBT.html>>

(luettu 23.6.2016).

McDermott, Wes n.d a. The Comprehensive PBR Guide by Allegorithmic – vol. 1.[PDF-dokumentti].

<https://www.allegorithmic.com/system/files/software/download/build/PBR_Guide_Vol.1.pdf>

(luettu 17.1.2016.).

McDermott, Wes n.d b. The Comprehensive PBR Guide by Allegorithmic – vol. 2.[PDF-dokumentti].

<https://www.allegorithmic.com/system/files/software/download/build/PBR_Guide_Vol.2.pdf>

(luettu 17.6.2016.).

McHugh, Sean n.d. Understanding Gamma Correction. [Verkkodokumentti].

<<http://www.cambridgeincolour.com/tutorials/gamma-correction.htm>>

(luettu 11.11.2016.).

Pedersen, Leif n.d. Linear Workflow. [Verkkodokumentti].

<<https://renderman.pixar.com/view/LinearWorkflow>>

(luettu 6.11.2016.).

Polycount Wiki n.d. a. Normal Map. [Verkkodokumentti].

<http://wiki.polycount.com/wiki/Normal_map>

(luettu 17.2.2016).

Polycount Wiki n.d. b. Ambient occlusion map. [Verkkodokumentti].

<http://wiki.polycount.com/wiki/Ambient_occlusion_map>

(luettu 17.2.2016).

Polycount Wiki n.d. c. Emissive map. [Verkkodokumentti].

<http://wiki.polycount.com/wiki/Emissive_map>

(luettu 17.2.2016).

Pruett, Chris 2014. Performance Optimization For Mobile Devices. [Verkkodokumentti]

<<http://robotinvader.com/blog/?p=438>>

(luettu).

Russell, Eddie 2014. Eliminate Texture Confusion: Bump, Normal and Displacement Maps. [Verkkodokumentti].

<<https://www.pluralsight.com/blog/film-games/bump-normal-and-displacement-maps>>

(luettu 7.2.2016).

Russell, Jeff n.d. Basic Theory of Physically-Based Rendering. Marmoset. [Verkkodokumentti].

<<https://www.marmoset.co/posts/basic-theory-of-physically-based-rendering/>>

(luettu 17.2.2016).

Sewell, Scott 2016. Gamma and Linear Space - What They Are and How They Differ. [Verkkodokumentti].

<<http://www.kinematicsoup.com/news/2016/6/15/gamma-and-linear-space-what-they-are-how-they-differ>>

(luettu 2.12.2016.).

Suomisanakirja n.d. Väliaine. [Verkkodokumentti].

<<http://www.suomisanakirja.fi/v%C3%A4liaine>>

(luettu 18.1.2016).

Wilson, Joe 2015. PBR Texture Conversion. Marmoset. [Verkkodokumentti]
<<https://www.marmoset.co/posts/pbr-texture-conversion/#tradconv>>
(luettu 31.3.2017).

Kuvalähteet

Kuva 1. Wilson, Joe 2015. PBR Texture Conversion.

<<https://www.marmoset.co/wp-content/uploads/traditionalcontent01.jpg>>

(luettu 11.4.2017).

Kuva 2. Karlsson, Robin 2014.

<<http://robinkarlsson3d.com/index.php?page=portfolio&piece=sfmachinegun>>

(luettu 11.4.2017).

Kuva 4. Schau, Andreas 2015

<<https://pixabay.com/en/diving-underwater-sea-float-813028/>>

(luettu 11.4.2017).

Kuva 5. Zahradníček, Ivo 2016.

<<https://pixabay.com/en/candle-candles-flame-wax-1700855/>>

(luettu 11.4.2017).

Kuva 8. 3D Coat n.d. Physically Based Rendering.

<<http://3dcoat.com/pbr/>>

(luettu 11.4.2017).

Kuva 9. Bichler, Walter 2015

<<https://pixabay.com/en/key-old-close-rusted-iron-metal-846527/>>

(luettu 11.4.2017).

Kuva 10. Wilson, Joe 2015. PBR Texture Conversion.

<<https://www.marmoset.co/wp-content/uploads/divingmetal01.jpg>>

(luettu 11.4.2017).

Kuva 12. Chadwick, Eric 2015.

<http://wiki.polycount.com/wiki/File:Normalmap_stairs.jpg>

(luettu 11.4.2017).

Kuva 17. Sewell, Scott 2016. Gamma and Linear Space - What They Are and How They Differ.

<<https://static1.squarespace.com/static/525ed7ade4b0924d2f499223/t/575f42e327d4bdc48f2261e4/1465860851928/An+image+comparing+gamma+and+linear+pipelines?format=750w>>

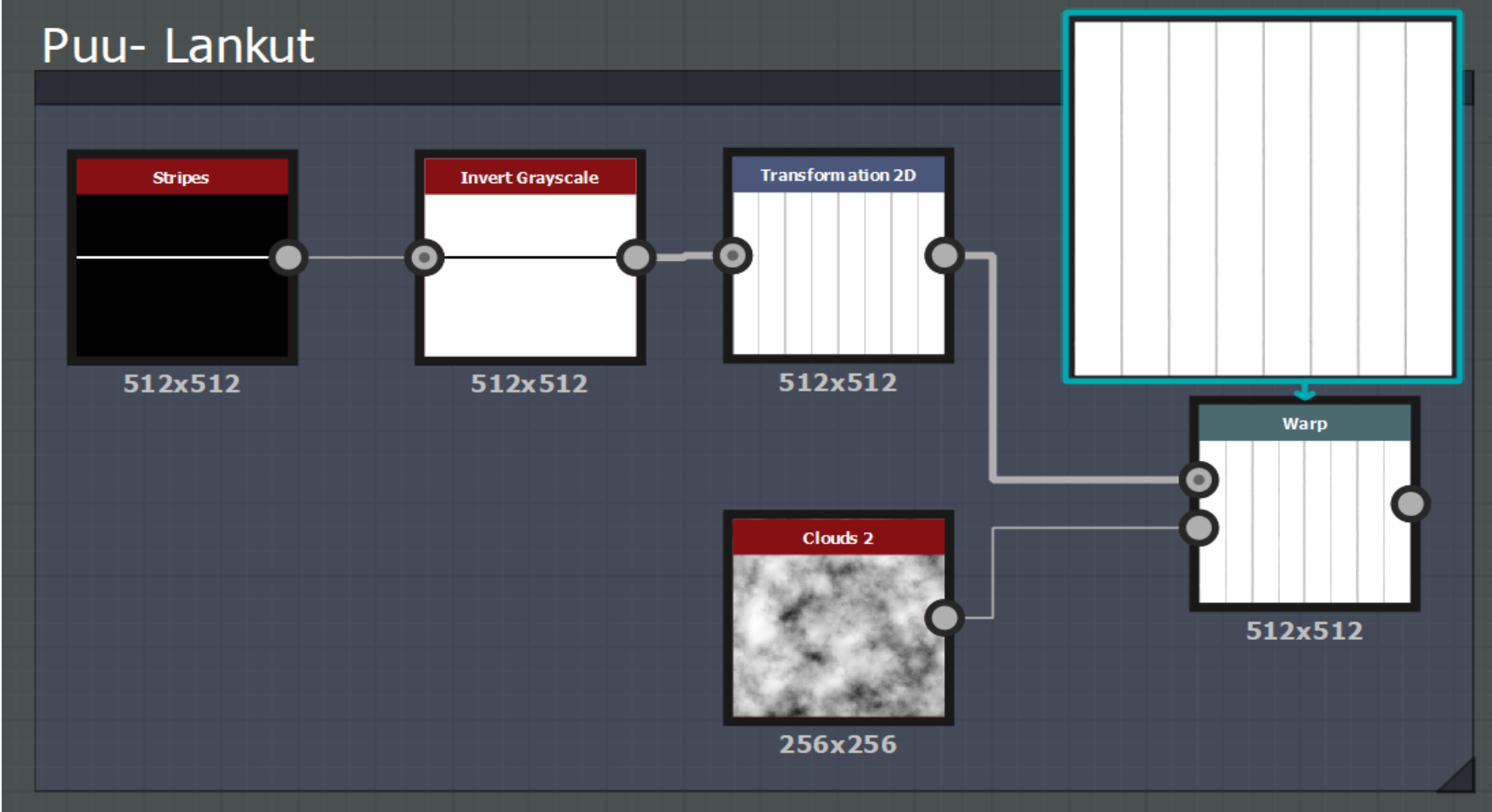
(luettu 11.4.2017).

Kuva 33. Ochel, Evita 2016.

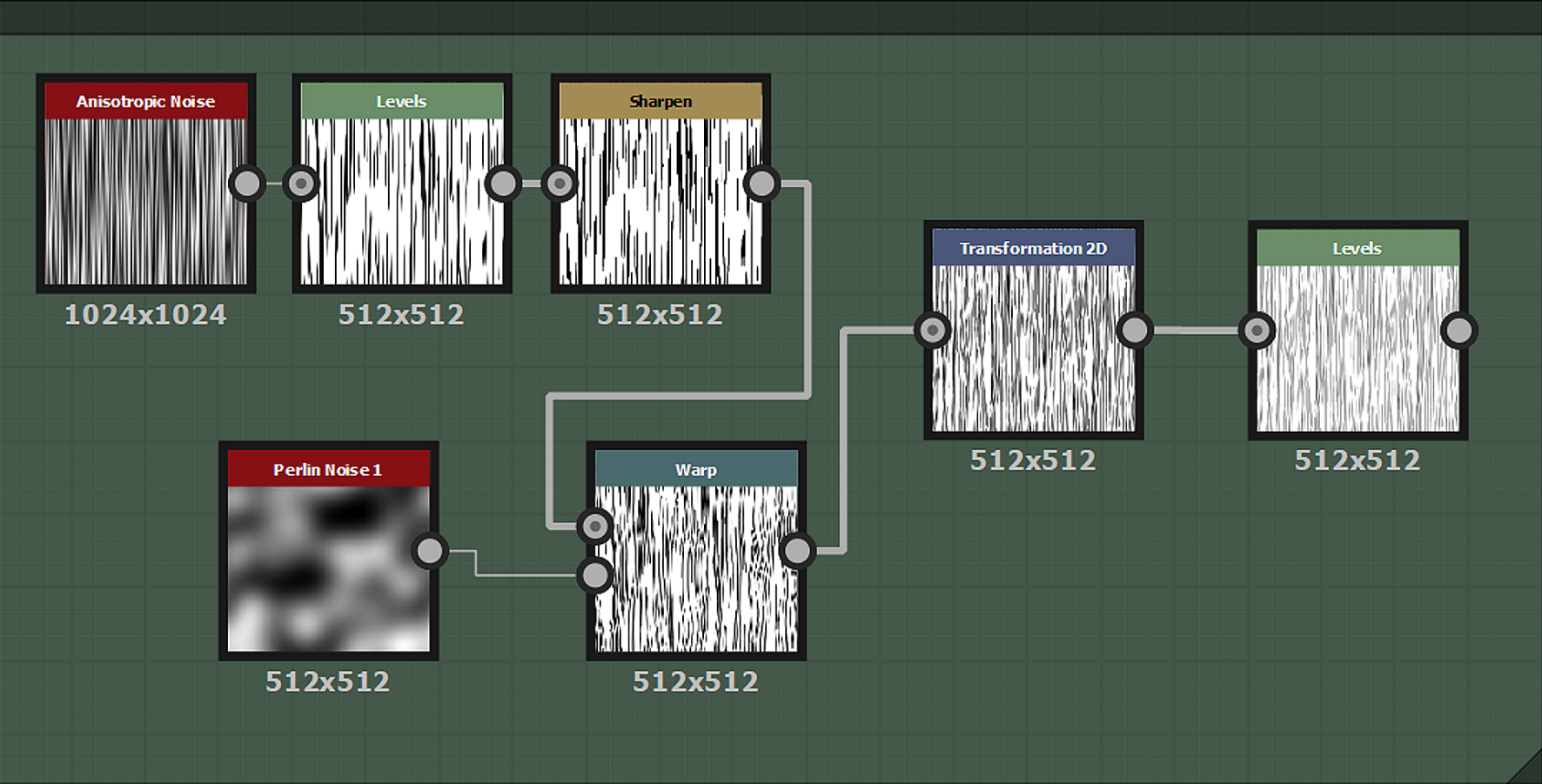
<<https://pixabay.com/en/crystal-amethyst-purple-mineral-1405272/>>

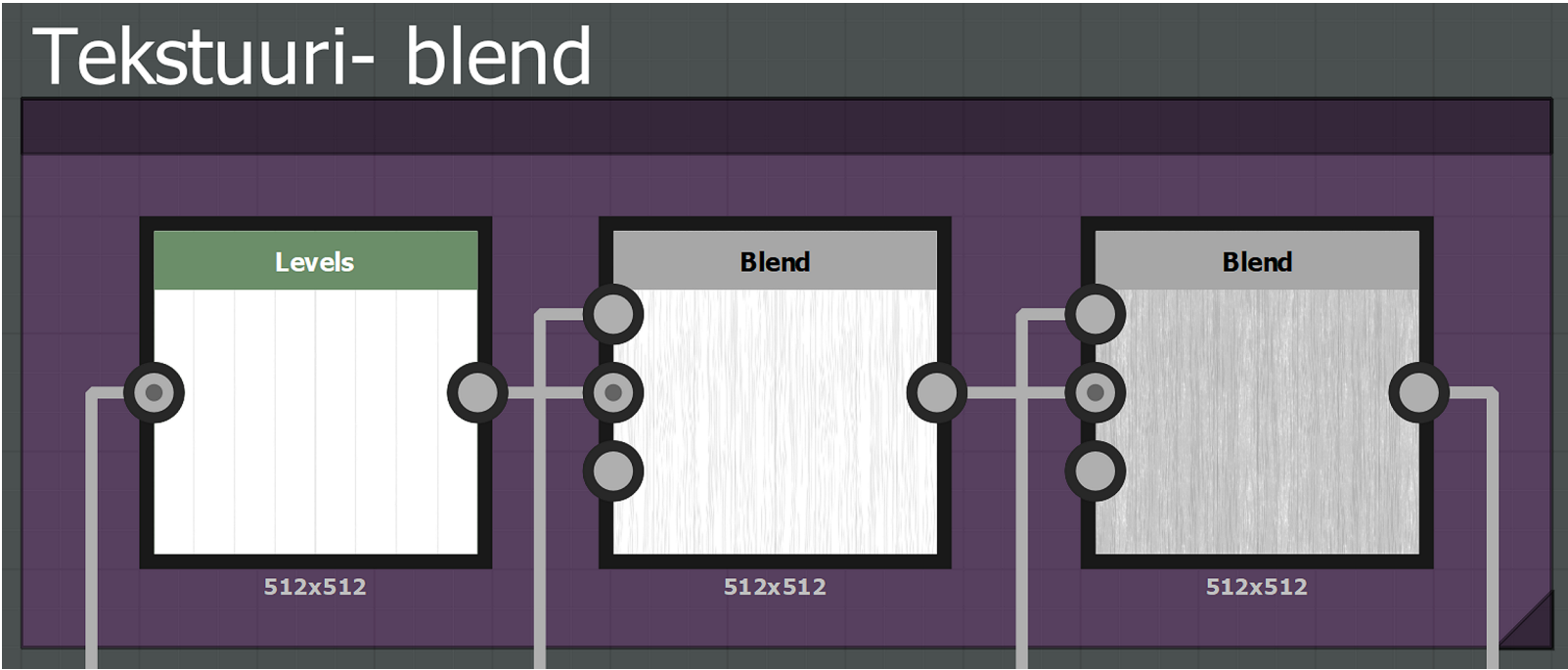
(luettu 17.4.2017).

Puumateriaalin graafin kuvakaappaukset

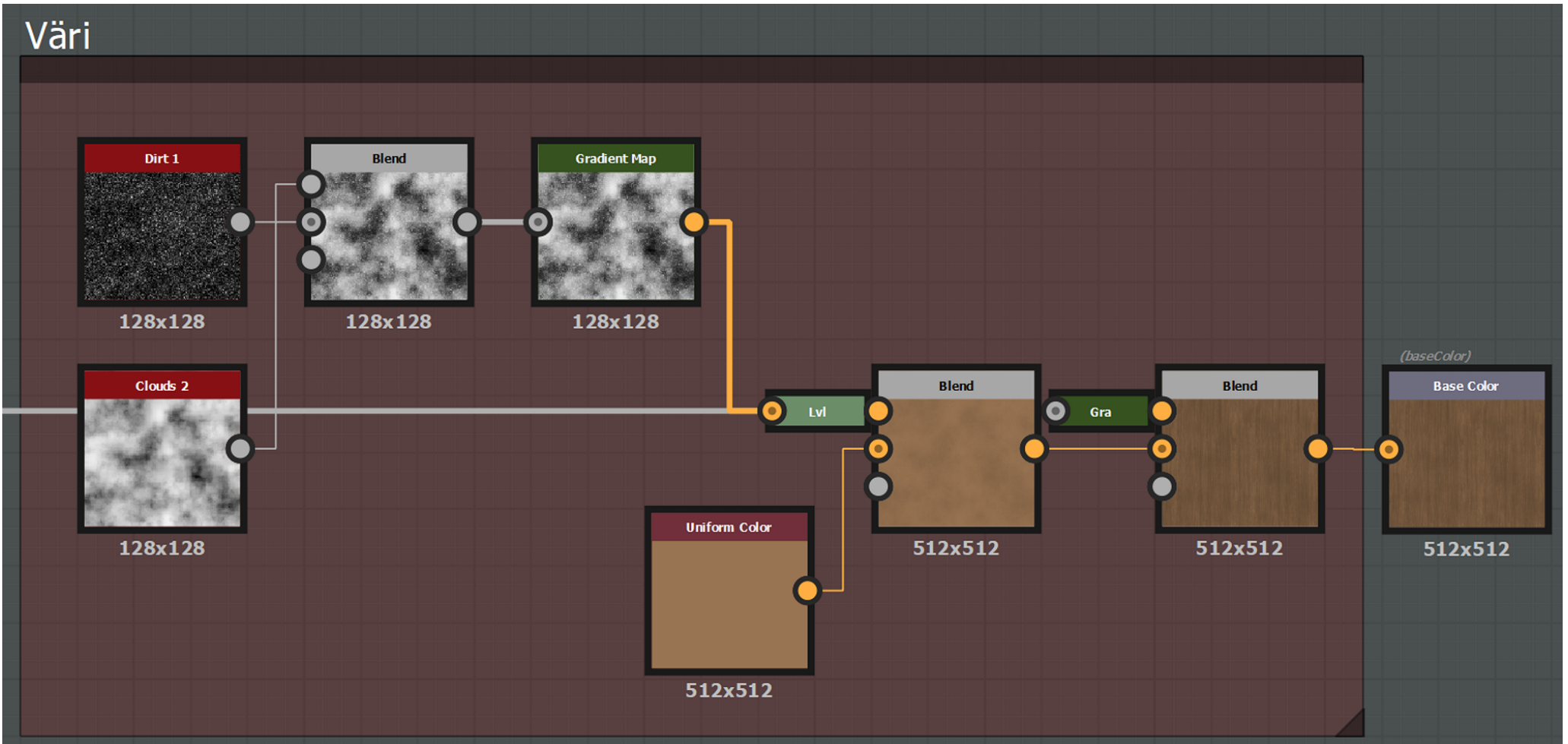


Puu- Tekstuuri

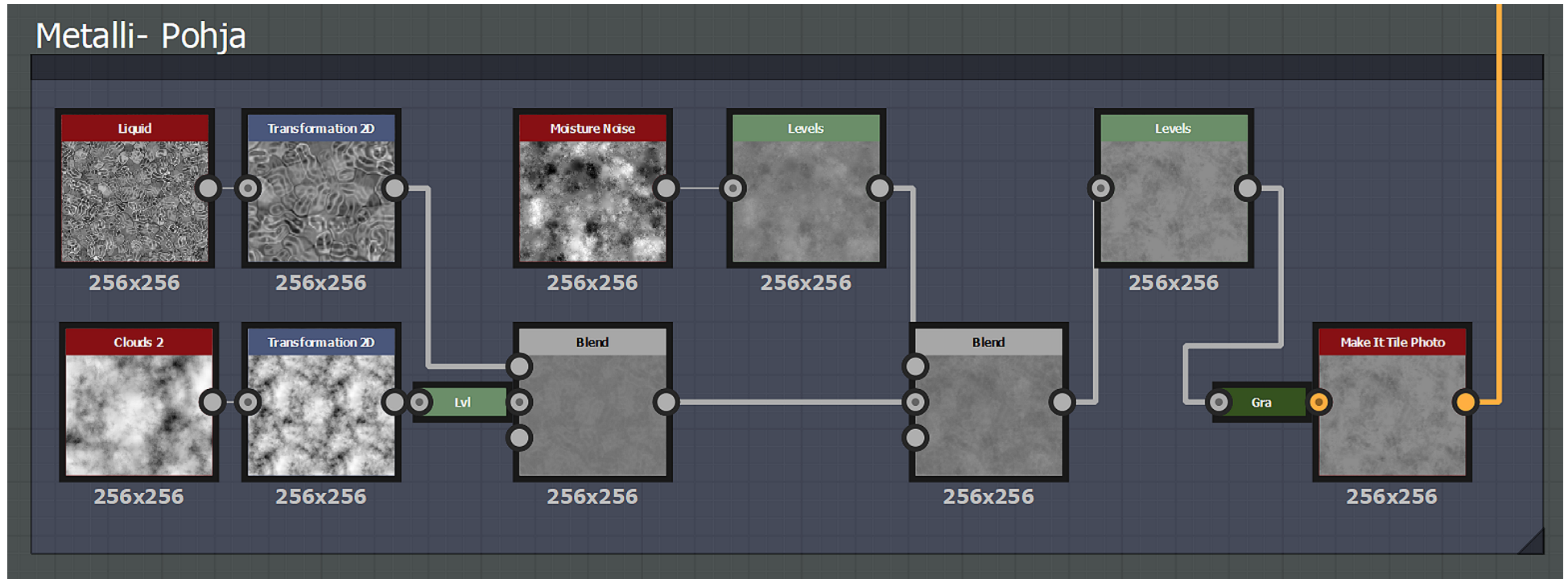




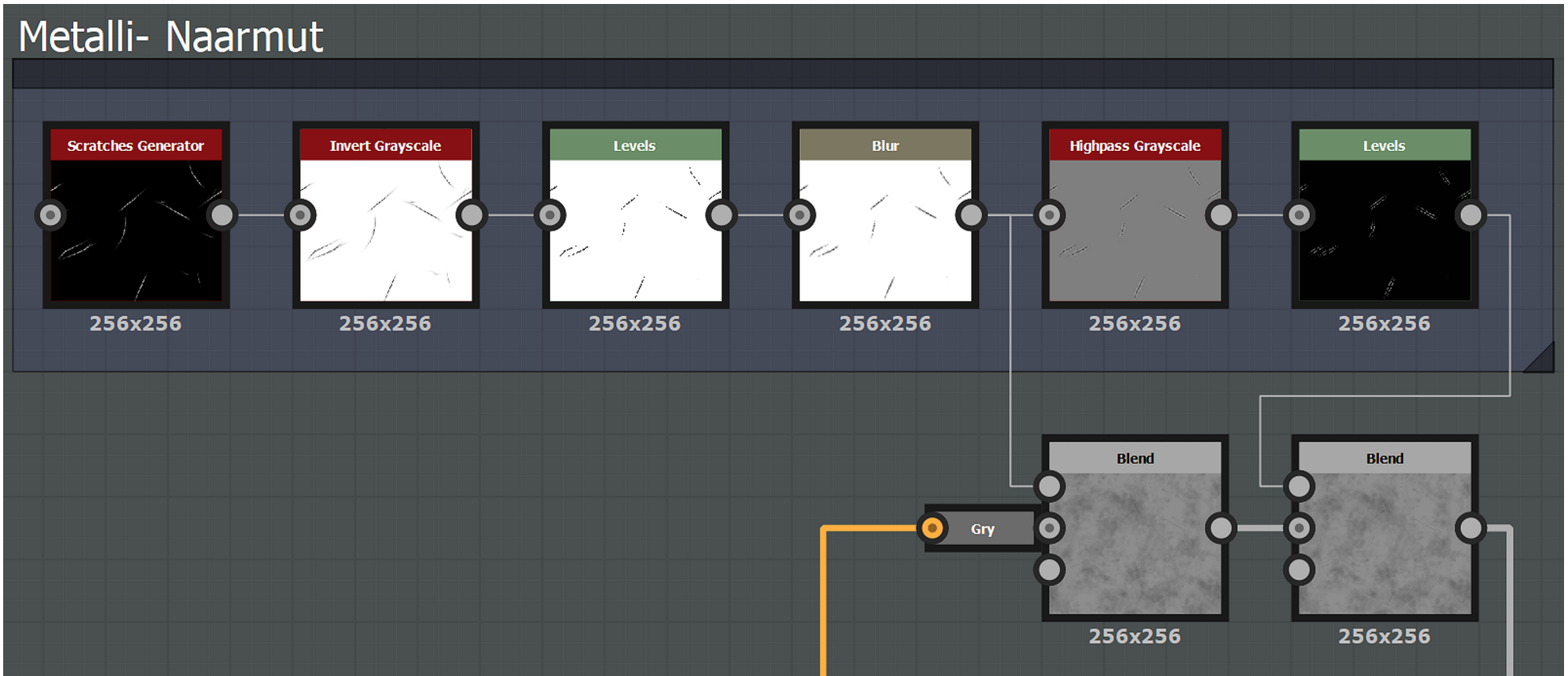
Väri



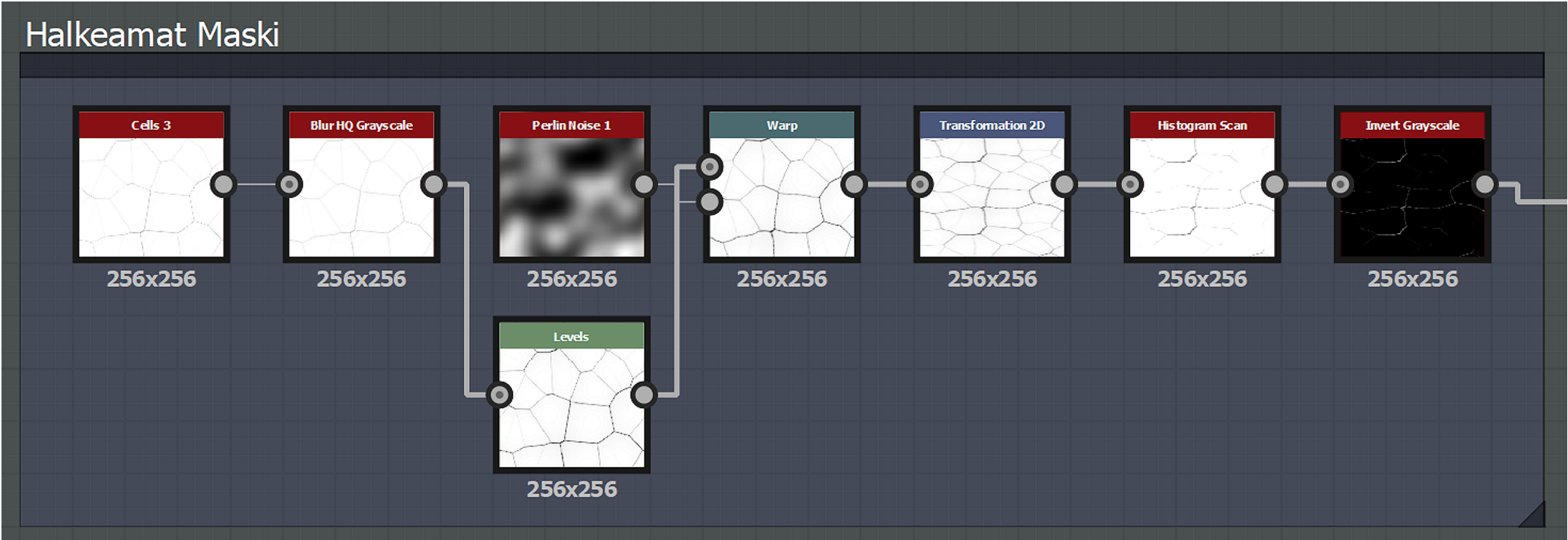
Metallimateriaalin graafin kuvakaappaukset



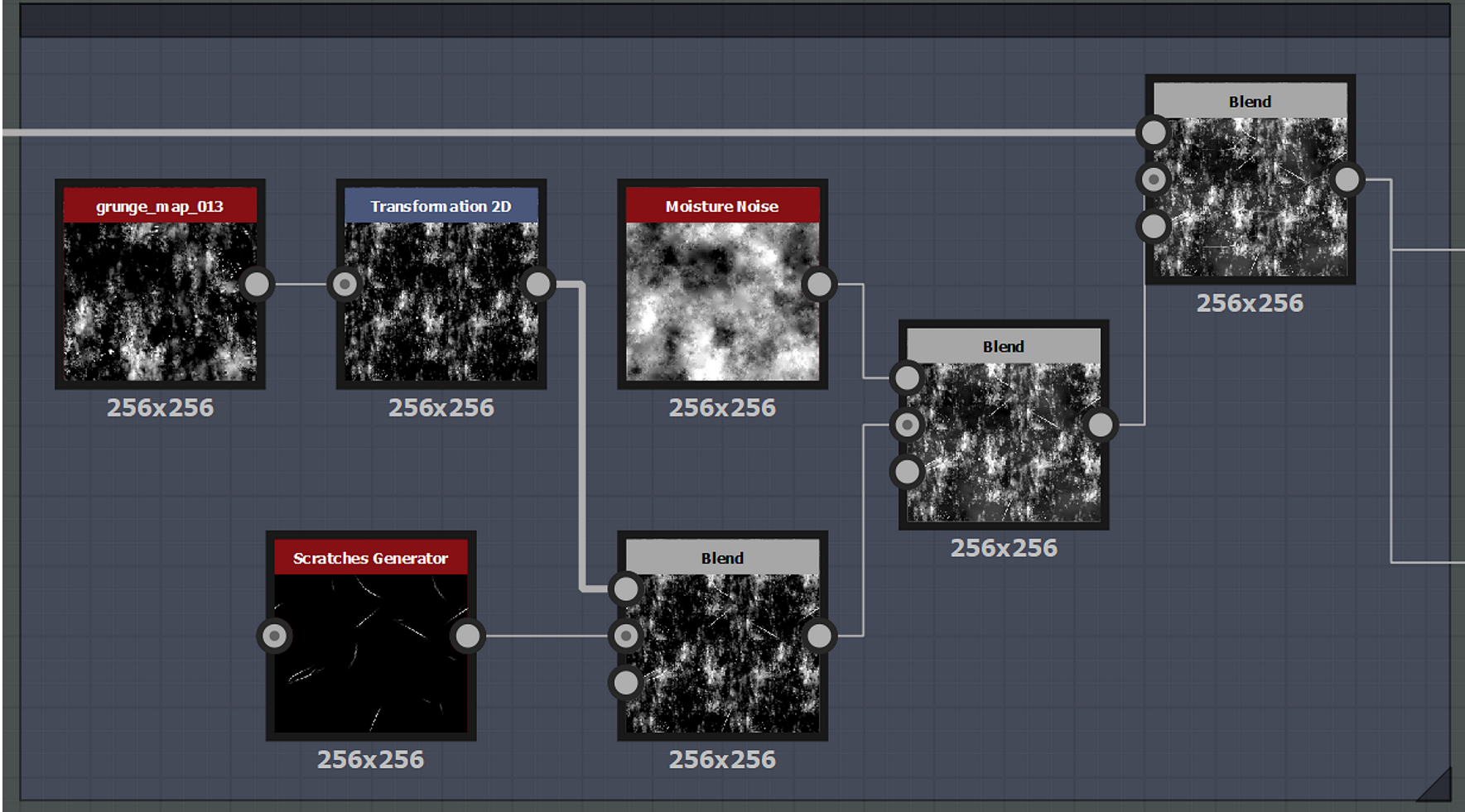
Metalli- Naarmut



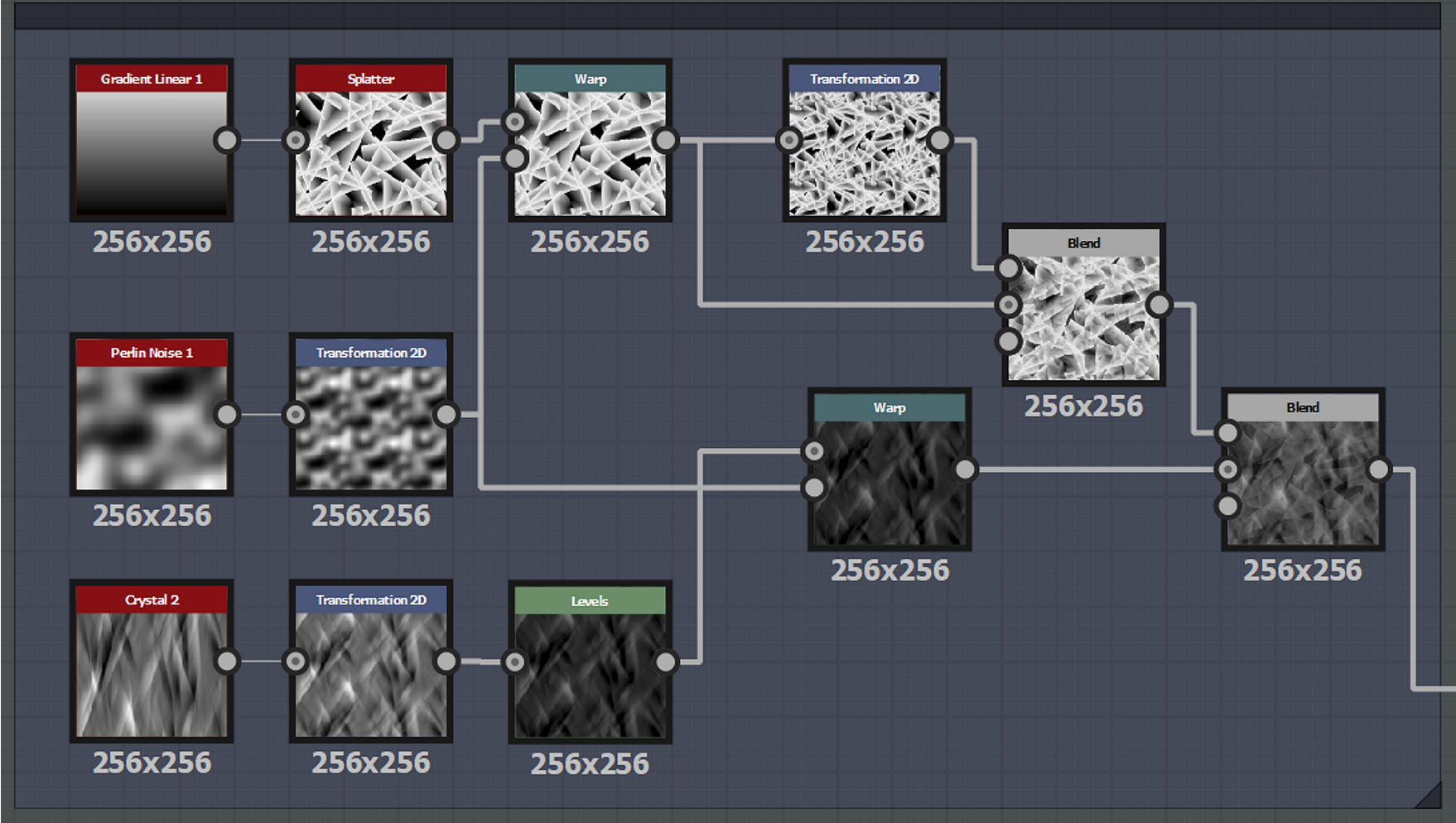
Kristallimateriaalin graafin kuvakaappaukset

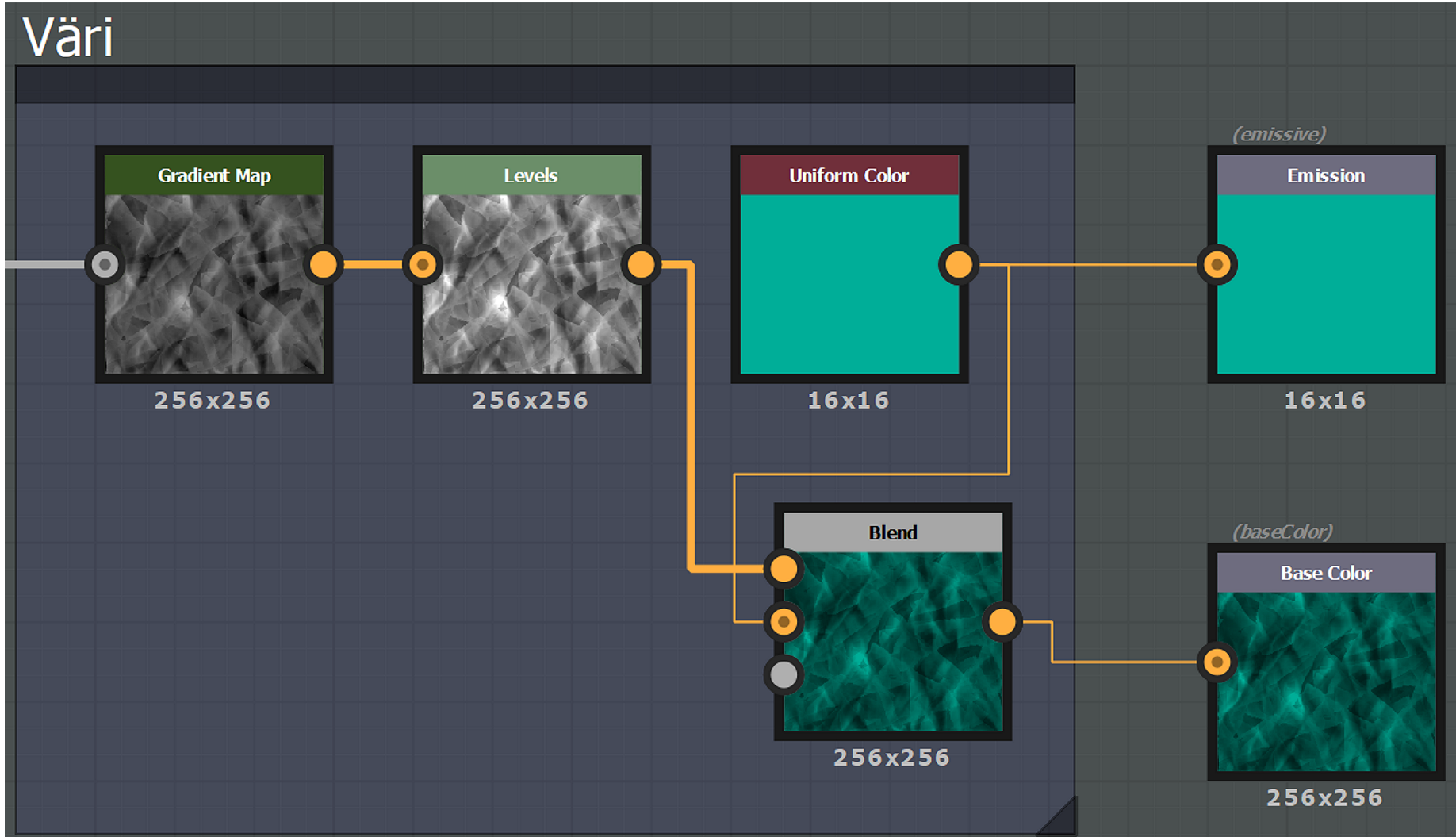


Kulumat



Pintakuviointi



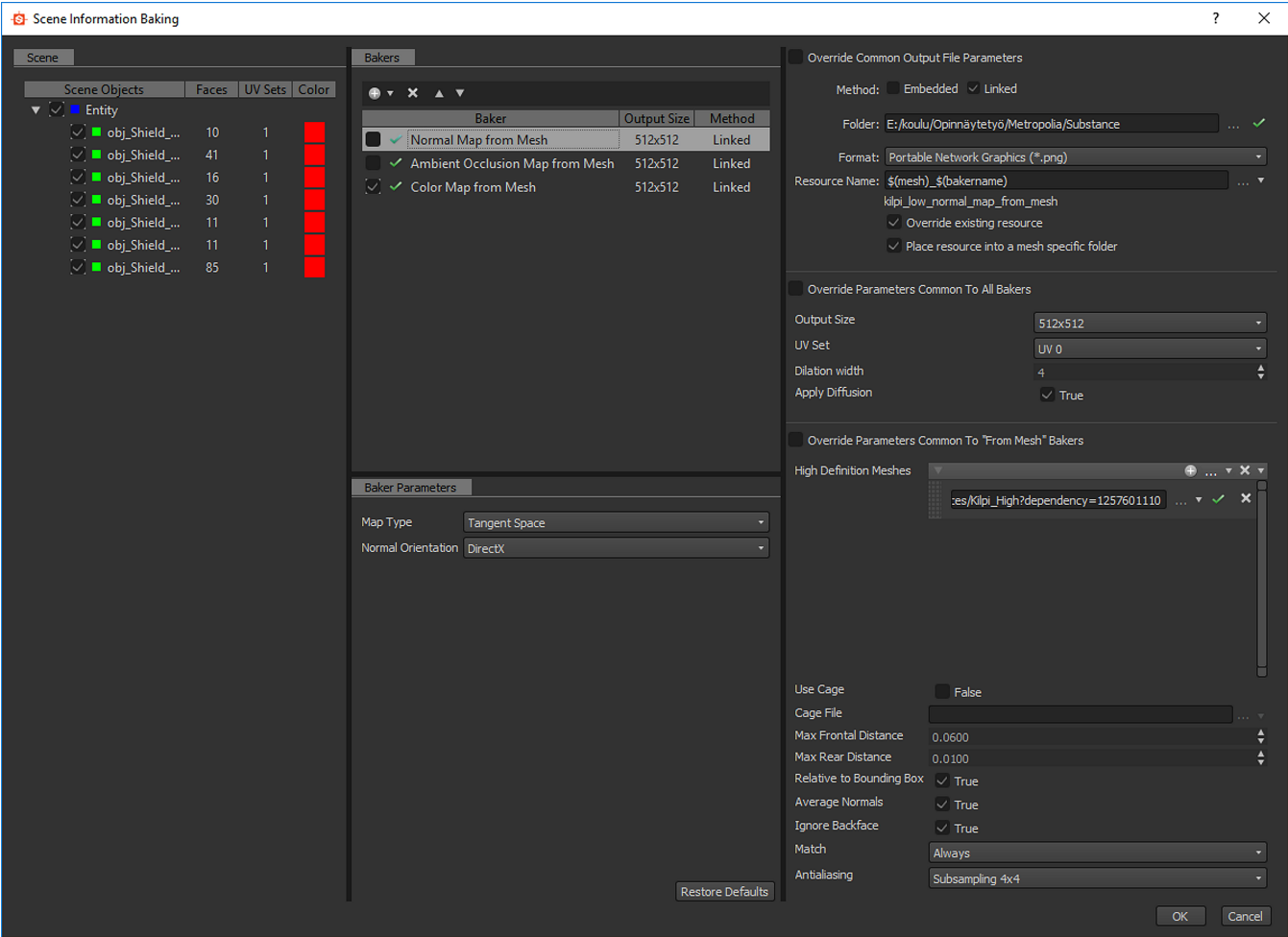


The image shows a software interface with two main panels. The left panel, titled "Uniform Color", features a dark red header with the text "Uniform Color" in white. Below the header is a large teal rectangular area. On the right side of this teal area, there is a circular orange button with a dark grey border. The right panel is a settings window titled "Input Parameters" with a close button (X) in the top right corner. It contains a section for "Crystalcolor (Float 4)" with the following fields:

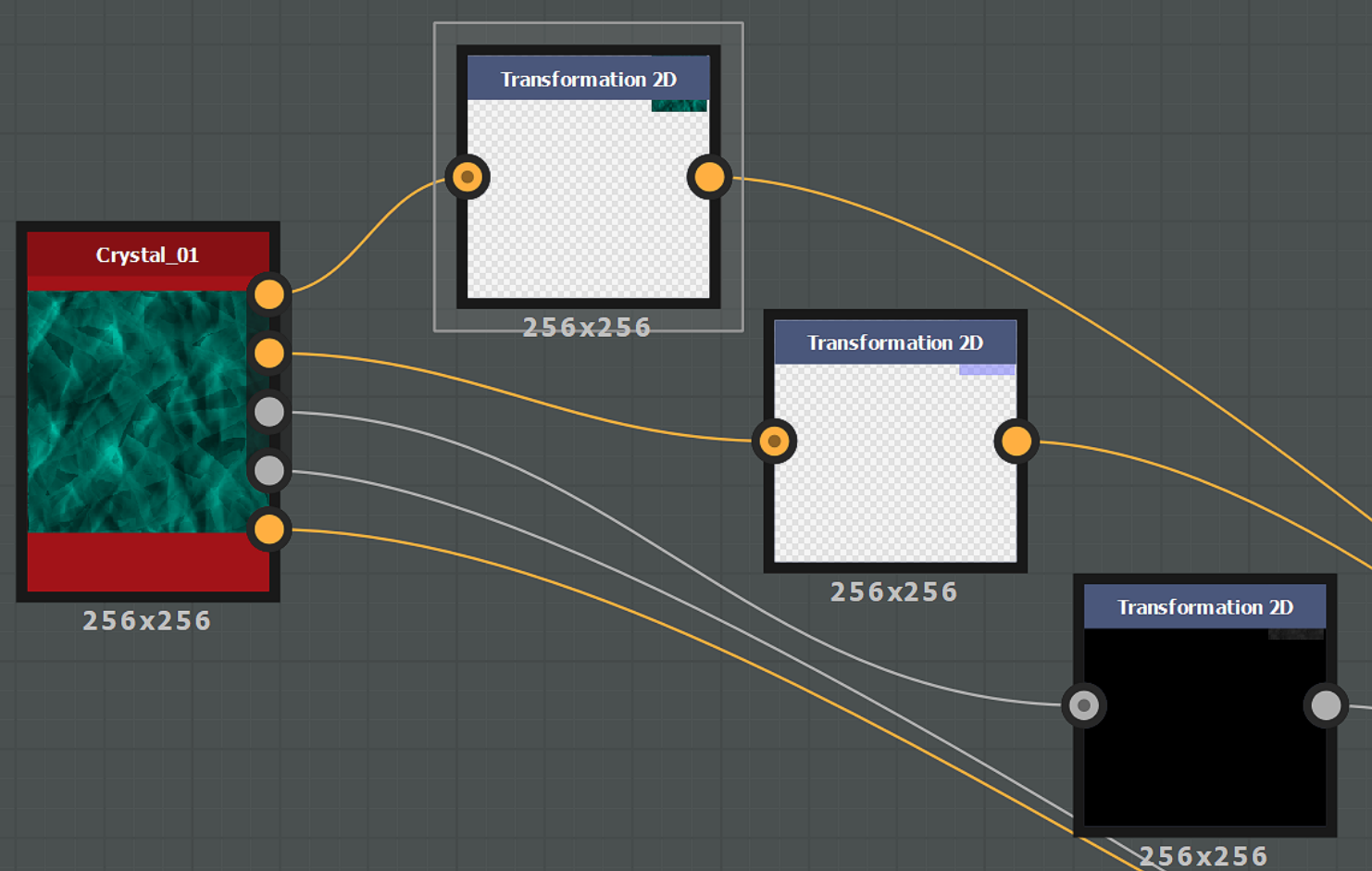
- Identifier: Crystalcolor
- Type/Editor: Float4 (dropdown), Color (RGBA) (dropdown)
- Description: (empty text area)
- Label: (empty text field)
- Group: (empty text field)
- Default: A color swatch showing a teal color next to a horizontal rainbow gradient bar.
- Color selection controls: "Pick" and "HSV" buttons, and input fields for H (172), S (255), V (174), and A (255).
- User Data: (empty text field)
- Visible If: (empty text field)

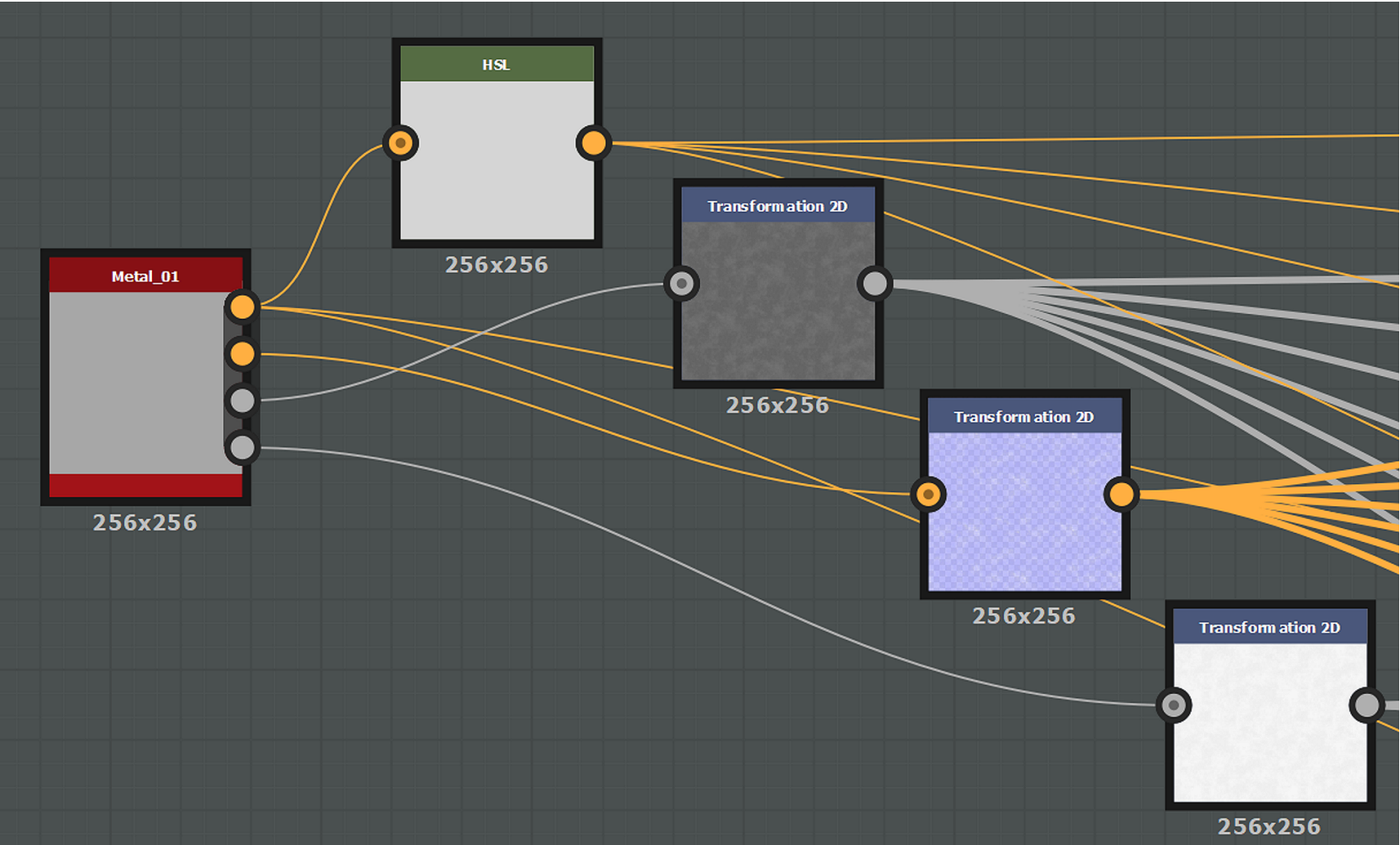
Below the "Input Parameters" section is a section titled "Input Images".

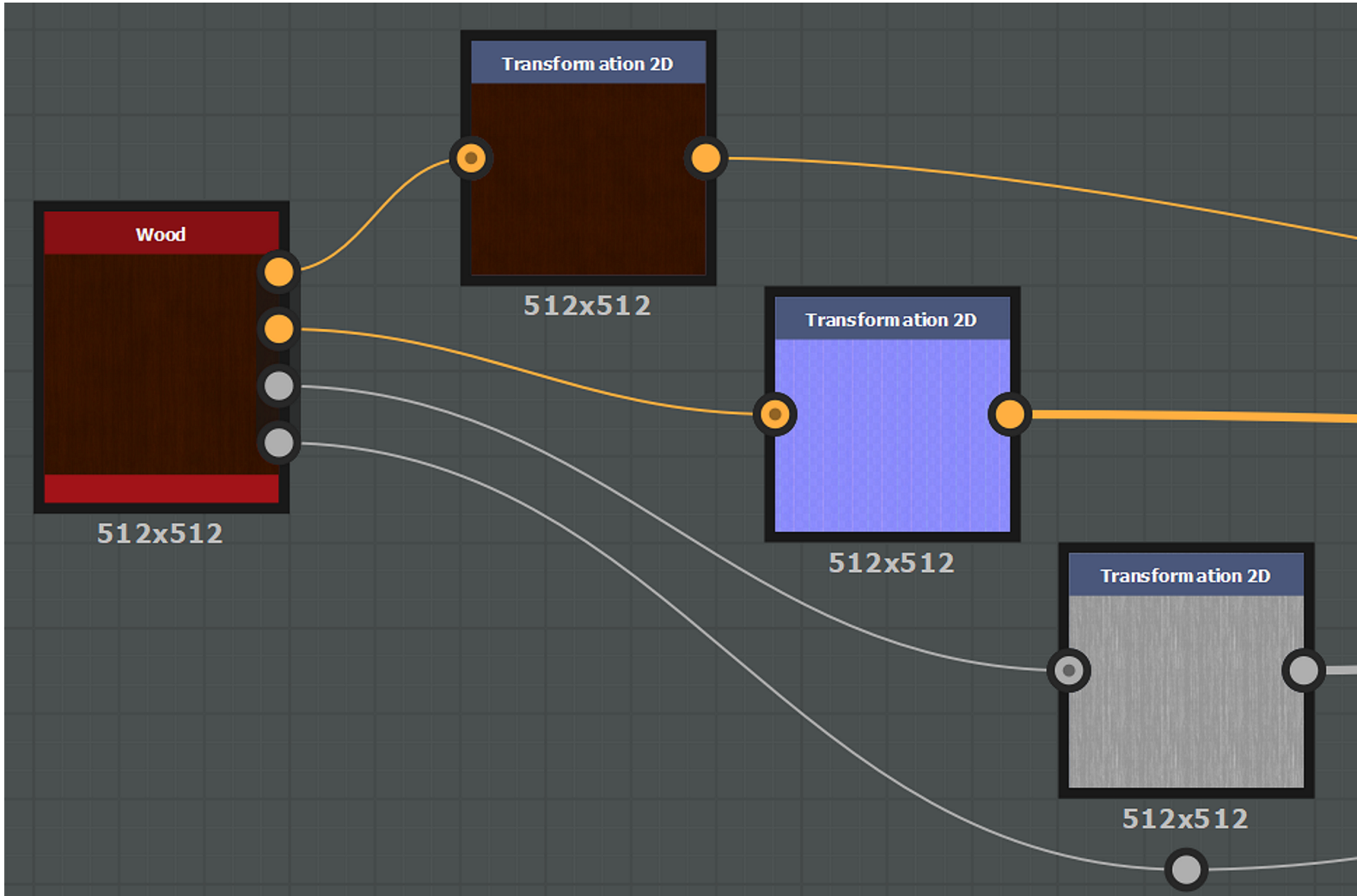
Scene Information Baking -ikkuna

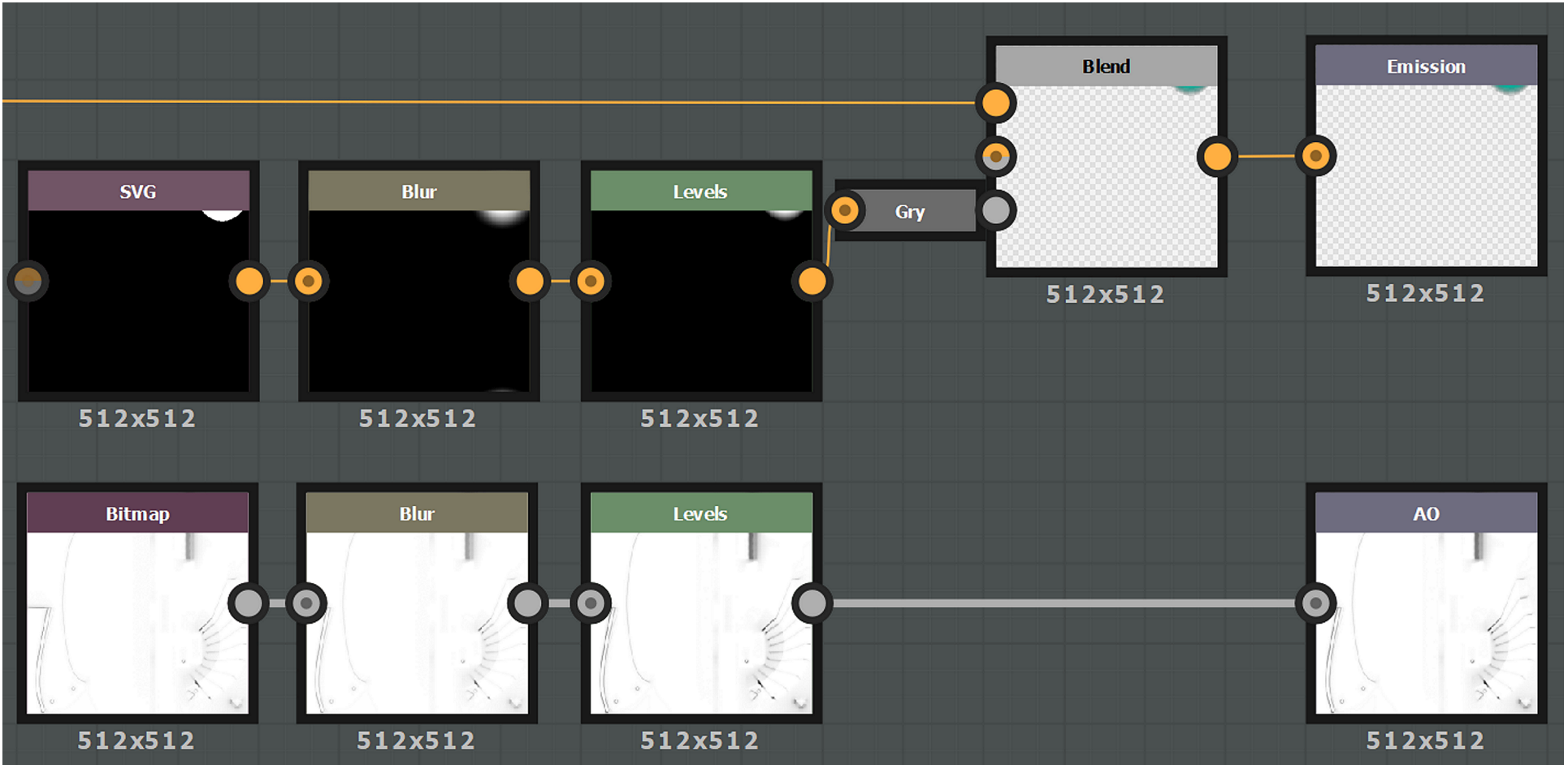


Lopullisen graafin kuvakaappaukset









Linkki lähdetiedostoihin

https://drive.google.com/file/d/0B5IL3_IkoZINZXNITnZReGlvNjg/view?usp=sharing