

Physically Based Renderer -ma- teriaali Substance Designer - ohjelmassa

Tomi Pulkkinen

Opinnäytetyö
Joulukuu 2020
Liiketalousala
Tietojenkäsittelyn koulutusohjelma
Tradenomi (AMK)

Tekijä(t) Pulkkinen, Tomi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 12.2020
	Sivumäärä 53	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Physically Based Renderer -materiaali Substance Designer -ohjelmassa		
Tutkinto-ohjelma Tietojenkäsittelyn koulutusohjelma, Jyväskylän Ammattikorkeakoulu		
Työn ohjaaja(t) Karhulahti, Mika		
Toimeksiantaja(t) Jyväskylän Ammattikorkeakoulu		
Tiivistelmä <p>Allegorithmic:n Substance Designer -ohjelmiston avulla tuotetaan korkealaatuisia tekstuurireita 3D-malleihin usealla eri alalla. Pelialalla useat, suuret kuin pienetkin pelistudiot käyttävät Allegorithmic:n ohjelmistoja tekstuurien valmistuksessa. Substance Designer mahdollistaa proseduraalisen työnkulun ja materiaalin vaatimat eri tekstuurikartat ilman erillisiä ohjelmia.</p> <p>Opinnäytetyössä tutkittiin, kuinka Substance Designer -ohjelmalla valmistetaan Physically Based Rendering -materiaali ja sitä millaisen ohjeistuksen kyseisen materiaalin luomiseen Jyväskylän Ammattikorkeakoulun Tietojenkäsittelyn opiskelijat tarvitsevat. Tutkimus toteutettiin kehittämistutkimuksena, jossa yhdistyvät myös kvalitatiiviset menetelmät. Tutkimuksen teoriaosuudessa käydään läpi PBR-tekniikan teoriaa ja millaisia siinä käytetyillä tekstuurikartoilla on. Teemahaastattelujen avulla selvitettiin JAMK:in tietojenkäsittelyn opiskelijoiden näkökulmaa, siitä millainen opas toimisi proseduraalisten PBR-tekstuurien valmistukseen.</p> <p>Haastattelujen tulosten perusteella saatiin selvitettyä, ettei oppaassa tulisi käydä niinkään ohjelman toimintoja läpi, vaan keskittyä siihen, kuinka työnkulku ohjelmassa toimii. Oppaan tulisi olla myös mahdollisimman visuaalinen, että sen avulla on helppoa seurata mitä opastetaan tekemään. Oppaassa tekstit tulisivat olla kuvien apuna selvittämässä mitä kuvissa tapahtuu.</p> <p>Tutkimuksen käytännön osassa valmistettiin Substance Designer -ohjelmalla proseduraalinen physically based renderer -materiaali. Kyseisen materiaalin valmistamisen jälkeen kirjoitettiin yksityiskohtainen opas Jyväskylän Ammattikorkeakoulun tietojenkäsittelyn oppilaiden käyttöön.</p>		
Avainsanat (asiasanat) Physically Based Rendering, Tekstuurit, Substance Designer, 3D-malli		
Muut tiedot (salassa pidettävät liitteet)		

Author(s) Pulkkinen, Tomi	Type of publication Bachelor's thesis	Date 12.2020
	Number of pages 53	Language of publication: Finnish
		Permission for web publication: x
Title of publication Physically Based Renderer -material in Substance Designer -Program		
Degree programme Business Information Technology degree programme		
Supervisor(s) Karhulahti, Mika		
Assigned by Business Information Technology degree programme, JAMK University of Applied sciences		
Abstract <p>Allegorithmic's Substance Designer software is used to produce high-quality textures for 3D models in a variety of fields. In the gaming industry, many game studios -both, - large and small use Allegorithmic's software to produce textures. Substance Designer enables the procedural workflow and different texture maps required by the material without separate programs.</p> <p>In the thesis, I studied how Substance Designer program is used to produce Physically Based Rendering material and what kind of guidelines the students of Jyväskylä University of Applied Sciences require to be able to create such material. The research was carried out as a development study, which also combines qualitative methods. In the theoretical part of the study reviews the theory of PBR technology and what kind of texture maps are used in it. The thematic interviews were used to find out the perspectives of JAMK's technology degree students on what kind of guide would work to produce procedural PBR textures.</p> <p>Based on the findings of the interviews, it was clarified that the guide should not go through the functions of the program. Instead, it should focus on how the workflow in the program works. The guide should also be as visual as possible to make it easy to follow. In the guide, the text should support the pictures.</p> <p>In the practical part of the study, procedural physically based renderer material was produced with the Substance Designer program. After the production of this material, a detailed guide was written for the use of technology degree students at Jyväskylä University of Applied Sciences.</p>		
Keywords (subjects) Physically Based Rendering, textures, Substance Designer, 3D-model		
Miscellaneous (Confidential information)		

1 Sisältö

1	Johdanto	4
2	Tutkimusasetelma	5
2.1	Tavoitteet ja rajaukset.....	5
2.2	Kehittämistutkimus -ja kysymykset.....	6
3	3D-mallinnus	7
3.1	3D-malli	7
3.2	Tasainen ja sileä varjostus.....	9
3.3	Teksturointi	9
4	Grafiikan piirtyminen ruudulle.....	10
4.1	Mallin ja maailman tila	11
4.2	Valot ja varjot	11
4.3	Projisointi.....	11
4.4	Rajaus ja kuvan kartoitus.....	12
4.5	Rasterisointi.....	12
4.6	Varjostimet, eli Shaderit.....	14
5	Physically Based Rendering.....	15
5.1	Fresnel-heijastus.....	17
5.2	Sähköä johtavat ja eristävät materiaalit	18
6	PBR-tekstuurikartat	19
6.1	Metalli- ja karheustyönkulku.....	19
6.2	Metallikartta	20
6.3	Karheuskartta	21
6.4	Spekulaari- ja kiiltotyönkulku	21
6.5	Spekulaari	22
6.6	Diffuusikartta.....	22

6.7	Kiiltävyysskartta	23
6.8	Normaalikartta	23
6.9	Ympäristön okklusio	23
6.10	Korkeuskartta	23
7	Tutkimustulokset.....	24
7.1	Alkukartoitus oppaan tarpeista	24
7.2	Oppaan käyttötarkoitus opiskelijoille	24
7.3	Oppaan valmistus	25
7.4	Oppaan testaus haastateltavilla	26
8	Johtopäätökset.....	26
9	Pohdinta.....	27
10	Lähdeluettelo	29
11	Kuvalähteet	30
12	Liitteet.....	31

Kuvat

Kuva 1.	3D-mallin mahdollistavat verteksi pisteet.	8
Kuva 2.	Grafiikkaputkilinjasto.....	10
Kuva 3.	Triangelit ennen ja jälkeen rajauksen.....	12
Kuva 4.	Verteksien matka pikseleiksi grafiikkaputkilinjastossa.	14
Kuva 5.	Valon säteiden sironna, heijastus ja imeytyminen materiaalissa.	16
Kuva 6.	Fresnel heijastus.....	17
Kuva 7.	Sähköä eristäviä materiaaleja.	18
Kuva 8.	Sähköä johtavia materiaaleja.	19
Kuva 9.	Metalli- ja karhauskartan tekstuurit.....	20

Kuva 10. Spekulaariryöngyön diffuusi- ja kiiltotekstuurikartat	22
--	----

1 Johdanto

Opinnäytetyössäni tutkin kuinka Substance Designer -ohjelmalla luodaan Physically Based Render-materiaali. Työn tarkoituksena on tutustua Substance Designer-ohjelman käyttöliittymään, ohjelman ominaisuuksiin ja millaisia tekstuurikarttoja PBR-materiaalit vaativat näyttääkseen aidoilta. Tutkimuksen tuotoksena on kuinka graafinen suunnittelija voi tuottaa Substance Designer -ohjelmalla proseduraalisen PBR-materiaalin.

Allegorithmic:n luomat Substance Painter ja Substance Designer -ohjelmat ovat monialaisia teksturointiohjelmaa, joita käytetään niin videopeli-, 3D-mallinnus- kuin elokuvateollisuudessa. Useat pelistudiot käyttävät Substancen ohjelmistoja luodakseen korkealaatuista sisältöä videopeleihin. Substance Designer onkin vakinaistunut asemansa Physically Based Rendering -tekstuurien valmistuksessa.

Substance Designerin etuna on tuottaa proseduraalinen työkulku tekstuurien valmistuksessa, jolloin luodut tekstuurit ja valmiit materiaalit ovat helposti muokattavissa ja uudelleenkäytettävissä.

Physically Based Rendering on lisääntynyt koneiden laskentatehojen mukana. Aiemmin PBR-materiaalit ovat olleet aiemmin valmiiksi renderöityjen mallien käytössä elokuva- ja mallinnusalalla, jolloin kuvaa ei tarvitse renderöidä reaaliaikaisesti. Nykyiset pelimoottorit pystyvät laskemaan käytettävien tekstuurikarttojen pinnoilta aidonmukaiset valojen heijastukset, jotka tuottavat pintamateriaalista aidon näköisen lopputuotoksen.

Opinnäytetyössäni keskityn tutkimaan, kuinka Substance Designerilla luodaan Physically based rendering -materiaali ja kuinka ohjelman proseduraalinen työkulku käytännössä toimii. Työn tuloksena on luoda ohjeistus tietojenkäsittelyn opiskelijoille, joille Substance Designer -ohjelma ei ole entuudestaan tuttu.

2 Tutkimusasetelma

2.1 Tavoitteet ja rajaukset

Toimeksiantajana kehittämistyössä toimii Jyväskylän ammattikorkeakoulu. Kehittämistyön tavoitteena on tuottaa uutta tietoa toimeksiantajalle, eli Jyväskylän ammattikorkeakoululle tekstuurien valmistamisesta ja työssä käytettävästä ohjelmasta. Työssä tutkitaan sisällöntuottajan näkökulmasta kuinka eri tekstuurikartat vaikuttavat materiaalin ulkonäköön ja kuinka tutkimuksessa käytettävän ohjelman ominaisuudet mahdollistavat niiden käytön. Lopputuotoksena on laatia dokumentaatio oppaan muodossa ohjelmasta, joka opastaa lukijalle ohjelmiston toimintoja ja kuinka valmistetaan physically based renderer-tekstuurit.

Tutkimuksen ulkopuolelle on rajattu ilmiöön liittyvät fysiikan ja matematiikan kaavat, sekä varjostustekniikan sisältämä ohjelmistokoodaus. Tutkimus tehdään teksturiartistin näkökulmasta.

Sisällöntuottajien tehtävänä on valmistaa 3D-malleja, tekstuureja, visuaalisia efektejä ja muita tyylittelyyn vaadittavia osa-alueita. Jyväskylän Ammattikorkeakoulun tietojenkäsittelyn opinto-ohjelman opiskelijoina on useita sisällöntuotannosta kiinnostuneita, jotka voivat osallistua Ticorporate-projektissa pelien kehitykseen.

Projektityöskentelyssä ongelmana on tehokas ajan käyttö. Yksilöllisten tekstuurien valmistaminen perinteisellä kuvankäsittelyohjelmalla on aikaa vievää ja eri tekstuurikarttojen luominen vaatii usein muita apuohjelmia. Käytettävät tekstuurit ovat harvoin ensimmäisen luonnoksen jälkeen valmiita, vaan niitä on myös muokattava tuotannon aikana.

Substance Designerilla proseduraalisten tekstuurien luominen mahdollistaa niiden muokkaamisen missä työvaiheessa tahansa rikkomatta niitä. Tällöin valmistettavaa tekstuuria on helppo muuttaa, tehdä eri versioita ja päivittää

tarvittaessa. Ohjelma mahdollistaa esimerkiksi normaali-, heijastus-, tai karheuskartat ilman erillisiä ohjelmia. Tekstuurit ovat esikatseltavana 3D-näkyvässä ilman erillistä renderöintiohjelmaa. Proseduraaliset tekstuurit ovat monikäyttöisiä ja muokattavissa yleisimmissä pelimoottoreissa kuten Unity ja Unreal Engine.

2.2 Kehittämistutkimus -ja kysymykset

Tutkimusmenetelmänä kehittämistutkimuksessa pyritään kehittämään ilmiötä, prosessia tai asiantilaa. Kehittämistutkimuksen tavoitteena on tuottaa tietoa yhteisölle tutkimusongelmien ratkaisussa. Ongelmien ratkaisussa on kyse uuden tiedon viemisestä toimintaympäristöön (Kananen 2012, 13.)

Kehittämistutkimus on joukko erilaisia tutkimusmenetelmiä, menetelmät määräytyvät kehittämiskohteen mukaan. Kehittämistutkimus on tutkimusotteena monimenetelmäinen ja se yhdistää kvalitatiivisen sekä kvantitatiivisen tutkimusmenetelmän. Tutkimuksen taustalla on teoriaa, johon tutkimuksessa nojataan. Itse kehittämistyö dokumentoidaan, joka tuottaa uutta tietoa (Mts, 21.)

Laadullisen tutkimuksen tiedonhankinnassa tutkittavien näkökulmaa saadaan selville havainnoinnin, teema- tai ryhmähaastatteluilla ja erilaisten dokumenttien analyysien avulla. Tutkittavien näkökulmat ovat ainulaatuisia, eikä niillä haeta yleistettävyyttä koko populaatioon (Bister 2019, 33.)

Kehittämisongelmaan- tai kohteeseen pyritään saamaan muutos toimenpiteillä, eli interventiolla. Interventiolla pyritään vaikuttamaan ongelmaan ja tuottamaan muutos lähtötilanteeseen. Kolmansien osapuolien vaikutusta interventiosta ei aina onnistuta poistamaan (Kananen 2012, 22.)

Tutkimus itsessään voi olla kohdistettu tutkittavan ilmiön osaan. Muutokseen kohdistuva tutkimus on oleellista, sillä epäonnistuessaan prosessin arviointi vaatii epäonnistumisen syiden etsimistä. Kehittämistyötä tehdessä voidaan prosessia arvioida, jolloin tuloksiin on mahdollista vaikuttaa itse kehittämistyön aikana (Mts, 23.)

Kehittämistyön tarkoituksena ei ole yleistää tutkimustuloksia, vaan niillä saadaan muutos kehittämisen kohteena olleeseen ilmiöön. Saadut tulokset ovat liitoksissa yksittäistapauksiin, mutta itse prosessi voidaan siirtää sellaisenaan vastaavaan kontekstiin (Mts 43.)

Kuinka luodaan proseduraalinen materiaali Substance Designer -ohjelmalla?

Tutkimuksessa havainnoidaan ja toteutetaan PBR-materiaali Substance Designer -ohjelmalla.

Miten eri tekstuurikartat toimivat 3D-grafiikan renderöinnissä?

Työssä tutkitaan eri tekstuurikarttojen ominaisuuksia ja kuinka ne muuttavat 3D-mallin ulkonäköä.

Kuinka luodaan opiskelijoille ohjeistus Substance Designer -ohjelman käyttöön?

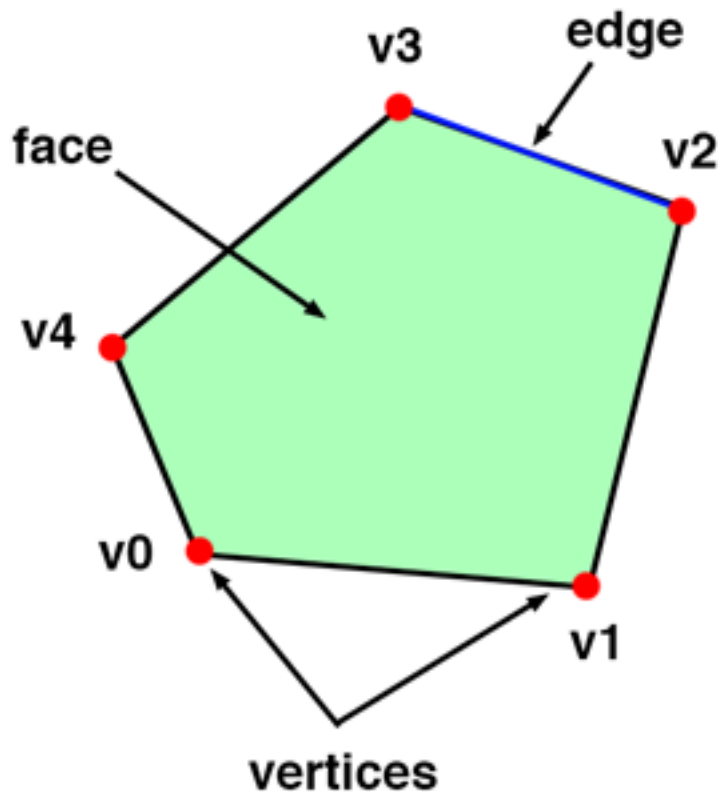
Tutkimuksessa selvitetään, kuinka valmistetaan ohjeistus tietojenkäsittelyn opiskelijoille Substance Designer -ohjelman käyttöön ja selvittäen edellä mainittuja tutkimuskysymyksiä.

3 3D-mallinnus

3.1 3D-malli

Tietokonegrafiikassa jokainen 3D-malli rakentuu polygoneista. Polygoni on yhdistelmä pisteitä, jotka yhdistyessä muodostavat renderöitävän 3D-muodon. Kyseisiä pisteitä kutsutaan vertekseiksi (Vertex), jotka sijaitsevat kolmiulotteisen koordinaatiston x, y ja z-akseleilla. 3D-ohjelmistoissa yleisesti käytettävä koordinaatiston x-akseli on leveys, y-akseli korkeus ja z-akseli on syvyys. Renderöitävässä polygonissa on vertekspisteiden välillä oltava yhdistävä

linja, jota kutsutaan reunaksi. Yksinkertaisimmillaan polygonin muodostaa kolmen yhdistävän vertekspisteen reunat, jolloin niihin muodostuu renderöitävä pinta. Kyseistä kolmen pisteen muotoa kutsutaan triangeliksi. (Paquette 2008, 5-7.)



Kuva 1. 3D-mallin mahdollistavat vertekspisteet. (Scratchapixel, n.d.)

Yleisin 3D-grafiikassa käytetty muoto on polygoniverkko (triangle mesh), joka on yhdistelmä joukkoa, jossa yhtenevät vertekspisteet, linjat ja pinnat. Polygoniverkossa yhdistyvät useat triangelin muodot, jossa jokaisella verteksillä on oltava yksi yhtenevä linja ja jokaisella linjalla on vähintään yksi yhtenevä pinta. (Hughes, Van Dam, Mcguire, Sklar, Foley, Feiner & Akeley 2014, 187.)

Polygonit sisältävät tiedon myös siitä, kuinka sen pintojen tulee heijastaa valonsäteitä. Jokaisen triangelin pinnan keskelle generoidaan ohjelmallisesti verteksille suunta, jonka myös kyseisen triangelin reunat ja verteksit perivät. Kyseinen suunta määrittää päästääkö se valonsäteen pinnan läpi vai tuottaako kyseinen pinta heijastuksen. Kyseistä ominaisuutta kutsutaan polygonin Vector Normal:ksi (useammin Normal:ksi). (Paquette 2008, 12-14.)

3.2 Tasainen ja sileä varjostus

3D-mallien valoheijastuksiin vaikuttavat käytettävät varjostustekniikat, joita ovat tasainen ja sileä varjostus (flat- ja smooth shading). Tasaisessa varjostustekniikassa triangelin pinnan heijastukseen vaikuttaa yksi avain verteksi, jonka värillä heijastuu koko pinta. Kyseinen varjostustekniikka soveltuu matalapolygonisiin malleihin. Matalapolygonisten mallin pinnat erottuvat selvästi kyseisellä varjostustekniikalla. Polygonien määrää nostamalla pintojen varjostus hieman tasoittuu, kuitenkin kokonaan poistumatta. Ongelman ratkaisuun on kehitetty sileä varjostustekniikka, jossa renderöitävän 3D-mallin pintojen kaarevuus lasketaan interpoloimalla vertekspisteiden väli. Tekniikka tuottaa mallin pintojen välien tasaisemman varjostuksen ilman polygonimäärän nostoa. Polygonin normalien avulla voidaan tuottaa 3D-malliin tasaisempaa varjostusta interpoloimalla kahden vertexin suunnan keskiarvo ilman, että polygoniin muotoon lisättäisi useampia vertex-pisteitä ja reunoja. Kyseistä toimintoa kutsutaan vertex normal:ksi. (Hughes ym. 2014, 128-129.)

3.3 Teksturointi

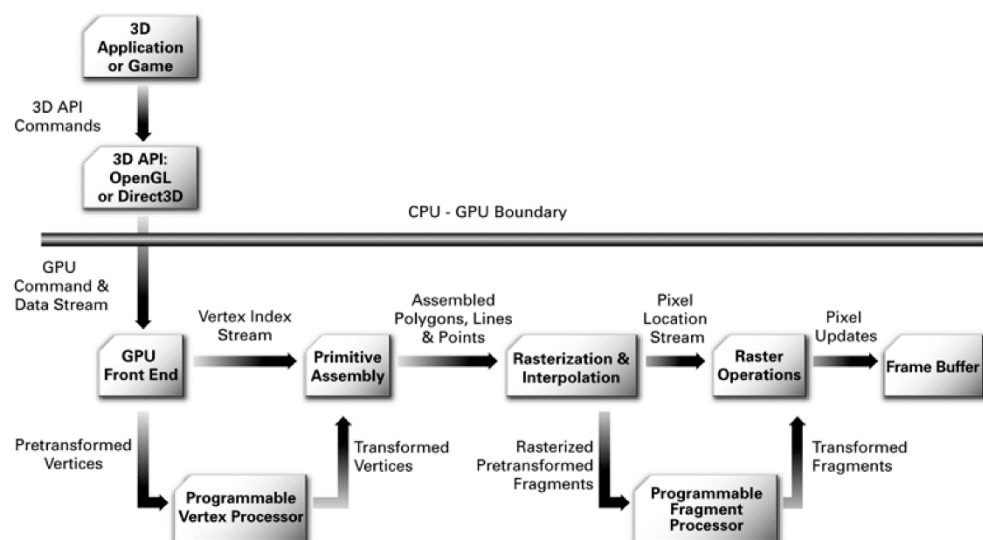
Erilaisilla tekstuurikartoilla lisätään 3D-malleihin yksityiskohtia ilman itse mallin geometrian muokkaamista. Tekstuurien avulla 3D-mallin päälle voidaan luoda ”nahka” päälle, joka näyttäytyy aidon oloisena materiaalina valon säteiden siihen osuessa. (Akenine-Möller & Haines 2002, 117.)

Tekstuurikartoituksessa 2D-kuva liimataan 3D-mallin päälle hyödyntäen polygonien vertekspisteitä. Tekstuurikartat sijaitsevat erillisellä 2D-ulottuvuudella U- ja V-akseleilla, jotka vastaavat mallin X- ja Y-akseleita. Kyseistä metodia kutsutaan UV-kartoitukseksi. (Mts, 118-119.)

4 Grafiikan piirtyminen ruudulle

Modernit tietokoneen grafiikkapiirit kykenevät tehokkaasti laskemaan tietokoneen ruudulle 3D-mallin geometriaa, tekstuureja ja varjostimia. Grafiikkapiirin apuna on application programming interface kirjaston sisältämän putkilinjasto, joita kutsutaan API:ksi. Kirjasto sisältää toimintoja, joiden avulla kolmiulotteiset mallit piirtyvät kaksiulotteiseksi kuvaksi ruudulle. (Shirley, Marschner, Ashikhmin, Gleicher, Hoffman, Johnson, Munzner, Reinhard, Sung, Thompson, Willemsen & Wyvill. 2009, 4-5.)

Reaaliaikaisen grafiikan muodostumiseen ruudulle voidaan jakaa kolmeen käsitteelliseen tilaan. Putkilinjastossa yhtäaikaistilat voidaan jakaa ohjelmistoon, geometriaan ja rasterointiin. Ohjelmistotilassa lasketaan piirrettävän mallin verteksipisteet, linjat ja triangelit. Näitä kutsutaan primitiiveiksi. Ohjelmistotilan primitiivien laskenta suoritetaan tietokoneen prosessorissa. Mallin primitiivien laskennan jälkeen tietokoneen prosessoritehot valjastetaan muille laskentatehoja vaativille operaatioille kuten törmäysten tunnistamiseen ja animaatioitten laskentaan. (Akenine-Möller & Haines 2002, 9-12.)



Kuva 2. Grafiikkaputkilinjasto. (CgUsersManual, 2005.)

4.1 Mallin ja maailman tila

Jokainen ruudulla näkyvä kolmiulotteinen malli käy läpi useampia eri tiloja tai koordinaatistoja ennen piirtymistä ruudulle. Mallien omaa koordinaatistoa kutsutaan mallin tilaksi, joka sisältää mallin verteksien ja normal-pisteiden muutokset. Yhdestä mallista voidaan tehdä erilaisia kopioita kolmiulotteiseen tilaan, jossa niitä voidaan suurentaa pienentää tai kääntää. Nämä muutokset tapahtuvat maailman tilassa. Jotta nämä mallit piirtyisivät ruudulle, maailman tilaan tarvitaan myös kamera. Kameran katselukulma määrittää kuinka mallit näkyvät ruudulle ja katselukulman ulkopuolelle jääviä malleja ei piirretä ollenkaan. Kameran näkymää kutsutaan näkymämuutokseksi. (Akenine-Möller & Haines 2002, 14-15.)

4.2 Valot ja varjot

Mallien renderöintiin maailman tila tarvitsee vähintään yhden valonlähteen. Reaaliaikaisessa renderöinnissä ruudulla näkyvien mallien pinnoilta lasketaan valon heijastukset ja kuinka pinnat imevät valonlähteen säteitä. Säteiden laskennassa simuloidaan kuinka valon fotonit kimpoavat tai imeytyisivät niiden osuessa erilaisiin pintoihin. Valonheijastukset lasketaan mallin vertekspisteiden ja normaalien suuntien mukaan. (Akenine-Möller & Haines 2002, 15-16.)

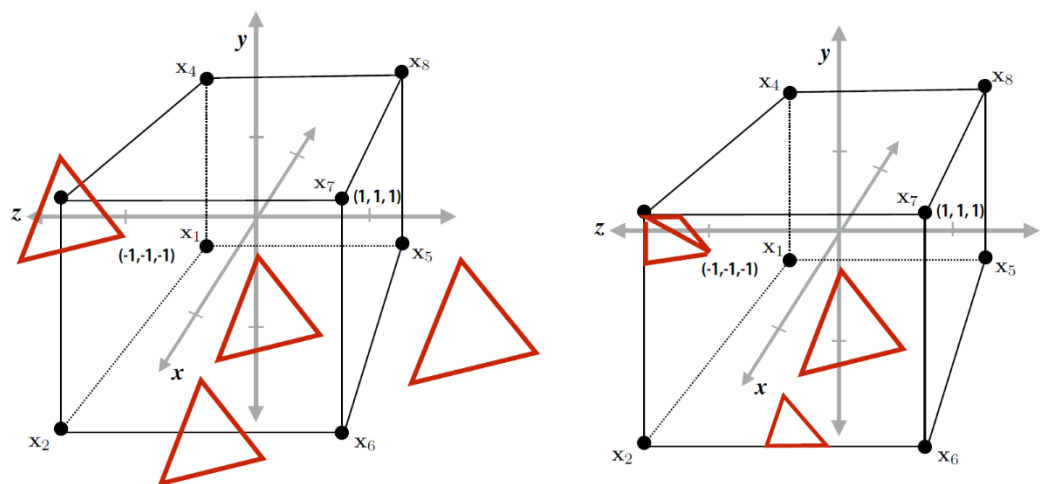
4.3 Projisointi

Kameralla, joka kuvaa kolmiulotteista maailmaa, on käytettävissä kaksi erilaista näkymää, ortografinen ja perspektiivinen. Ortografinen kuvakulma heijastaa maailman suorakulmionmuotoisesti, jolloin linjat näyttäytyvät rinnakkaisesti. Perspektiivinen kuva simuloi kuinka ihmissilmä näkee maailman. Kaukaisimmat objektit piirtyvät kauemmaksi. Kyseisessä perspektiivissä maailma kuvataan kartion mallisesti, jonka pohjalla on suorakulmio. Kartion pohjalla olevan suorakulmion ala on näkymä, jota kutsutaan frustrumiksi. (Akenine-Möller & Haines 2002, 17-18.)

4.4 Rajaus ja kuvan kartoitus

Mallit, jotka näkyvät ruudulla kokonaan tai osittain, piirretään ruudulle. Kuva-kulman ulkopuolella olevat mallit hylätään piirtonäkymästä. Ruudulla osittain näkyvien mallien ulkopuolella olevia verteksipisteitä ei lasketa vaan niille luodaan uusi verteksipiste renderöitävän kuvan sisäpuolen ja ulkopuolen rajaan. (Akenine-Möller & Haines 2002, 18.)

Geometrian viimeisenä tilana on kartoittaa kolmiulotteisessa maailmassa sijaitsevat mallit ruudulle 2D-näkymänä. Primitiivien sijainnit x-, y- ja z-akseleilta muutetaan kolmiulotteisesta maailmasta kaksiuulotteiseen yhdistäen kuvan ja ikkunan koordinaatiston. (Mts, 19.)



Kuva 3. Triangelit ennen ja jälkeen rajauksen. (Stanford University, 2018.)

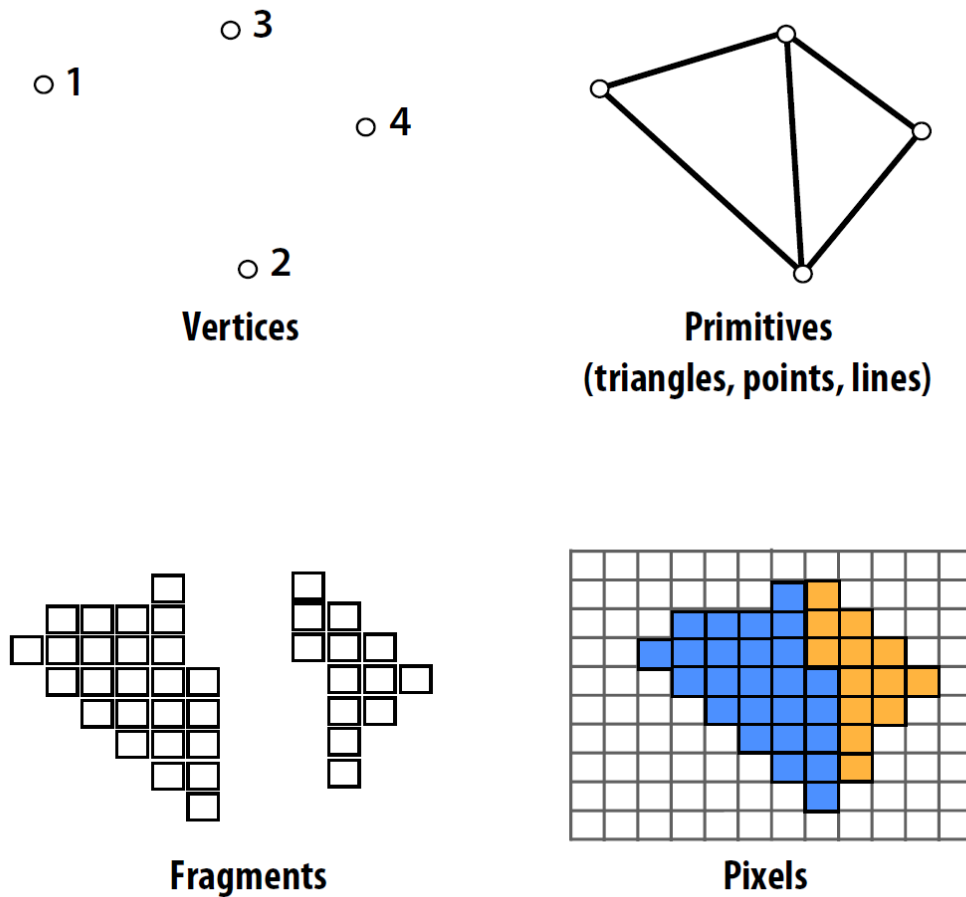
4.5 Rasterisointi

Geometriatilasta saapuneet polygonit niiden väri- ja tekstuuritiedot saavat rasterisoinnissa oikeat värit ja niiden ominaisuudet. Rasterisoinnissa verteksien ominaisuudet lasketaan pikseleinä ja geometriatilassa polygoneina. Pikselien ominaisuudet ovat sisällytettynä väripuskuriin (Color buffer), jonka RGB-skaala antaa jokaiselle pikselille niiden värimäärytykset. (Akenine-Möller & Haines 2002, 20.)

Näytöllä näkyvät pikselit saavat arvonsa taulukoista liukulukuina, jotka ovat 0-1 mustavalkoisille pikseleille tai 0-255 muoto jokaiselle RGB:n osavärille, eli punaiselle, vihreälle ja siniselle sävyille. (Shirley ym. 2009. 60-61.)

Rasterointi-tila pitää sisällään myös kaksi muuta puskuria. Z-puskurin (Z-buffer) ominaisuutena on laskea mallien pikselien näkyvyydet z-akselilla. Pikselien sijainnit ja etäisyydet lasketaan syvyysakselilla ja saavat näkyvyyden sen mukaan mikä malleista sijaitsee lähimpänä kameran näkymää. Kuvan renderöinnissä tuplapuskurin (Double-buffer) tehtävänä on piirtää kuvaa kahden eri kuvatiedon mukaan. Puskurin muistissa oleva kuva korvaa nykyisen kuvan, kun primitiivien seuraavat sijainnit ja väritiedot on saatu laskettua. (Akenine-Möller & Haines 2002, 20-21.)

Primitiivit, joille on määritetty yksi tai useampi tekstuuri, piirtyvät rasterointitilan lopussa ruudulle värillisinä pikseleinä määrättyjen värien mukaan. Pikselien piirtämisen jälkeen putkilinjaston toiminnot saavuttavat päätöksen ja aloittavat syklin uudestaan. (Mts, 21-23).



Kuva 4. Verteksin matka pikseleiksi grafiikkaputkilinjastossa. (Stanford University, 2018.)

4.6 Varjostimet, eli Shaderit

Varjostimet (Shader) ovat pieniä ohjelmia, joiden tehtävänä on kertoa kuinka 3D-mallit ja niiden pinnalla olevat tekstuurit näyttävät ruudulla. Varjostimien sisältämät koodit voivat suorittaa laskutoimituksia, jotka muuttavat 3D-mallin geometriaa ja niiden pikselit näyttävät ruudulla. Varjostimet sisältävät myös valaistusmallin, eli sen kuinka pintojen ja tekstuurien tulee käyttäytyä valon niihin osuessa. (Hughes ym. 2014, 927-930.)

5 Physically Based Rendering

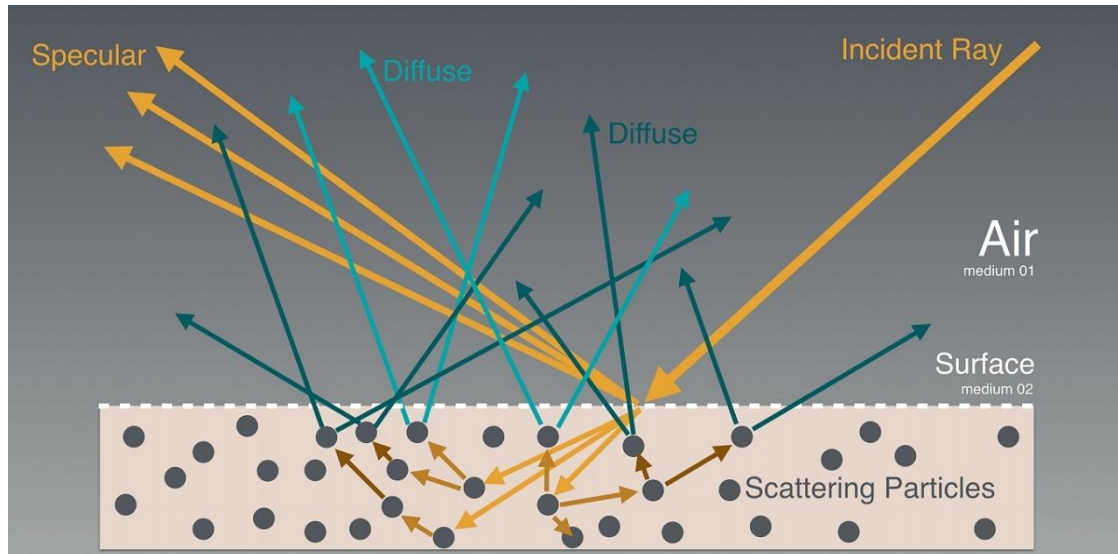
Physically Based Rendering (PBR) on renderointi ja varjostustapa, jolla voidaan toteuttaa kuinka materiaalit ja niiden pinnat heijastavat valoa. Fyysisesti realististen materiaalien pintojen heijastukset pohjautuvat reaalimaailman kaavoihin, siihen kuinka ne käyttäytyvät valon niihin osuessa. Näiden arvojen avulla on pinnoista mahdollista saada realististen näköisiä kaikissa eri valolosuhteissa. (McDermott 2018, 45.)

Valonsädemallissa valonsäteet käyttäytyvät ennakoitavasti erilaisten läpinäkyvien aineiden läpi kuten ilman tai veden. Mallin avulla havainnollistetaan valonsäteiden kulkeutuminen materiaalin pinnalle ja kuinka ne jatkavat matkaa niistä heijastuessa. Säteiden osuessa heijastavaan materiaalin pintaan, säteet heijastuvat ja jatkavat matkaa vastakkaisessa kulmassa tulokulmaan nähden. Valon kulkeutuessa läpikuultavien materioiden läpi valon säteet jatkavat suuntaansa kohtisuoraan tai taittuen. (Mts, 18-19.)

Valon kulkeutuessa läpikuultavien materiaalien läpi valon säteet imeytyvät tai voivat tehdä sirontaa. Valon imeytyessä valonteho heikkenee ja se muuttaa muotoaan energiaksi kuten lämmöksi. Valon säteiden aallonpituus vaikuttaa kuinka paljon valon väri muuttuu sen imeytyessä materiaaliin, kuitenkin niin, ettei säteiden suunta muutu. (Mts, 19.)

Säteiden sirottuessa ja taittuessa niiden suunta muuttuu materiaalin pinnan poikkeaman mukaan ilman valon tehon heikkenemistä. Valon tehon heikkenemiseen, taittumiseen ja sirontaan vaikuttaa kuinka paksun materiaalin läpi säteet kulkeutuvat. (Mts, 20.)

Useimmat pinnat ovat epätasaisia ja karheita, muuttaen satunnaisesti valonheijastusta pinnoiltaan. Epätasaisimmat kohdat karheissa pinnoissa näyttävät tummempina, toisin kuin heijastavat pinnat, jotka ilmenevät kirkaampia ja tuottavat valon heijastuksia. Vaikka pintojen heijastukset ovat intensiteetiltään erilaiset, heijastavat ne valoa pinnoiltaan yhtä paljon. (Mts, 22.)



Kuva 5. Valon säteiden sironta, heijastus ja imeytyminen materiaalissa. (Romanowski, 2016.)

Valon taittuessa materiaalista toiseen sen suunta ja vauhti muuttuu. Tätä kutsutaan taitekertoimeksi (Index of refraction, IOR). Kertoimen arvo määrittää valonsäteen taittumiskulman materiaalista. Taitekertoimen ilmiötä voidaan kuvata hyvin säteiden kulkeutuessa useamman läpinäkyvän materiaalin läpi kuten ilman, veden ja lasin läpi. (Mts, 23.)

Diffuusiheijastuksessa valonsäteet heijastuvat useaan kertaan läpäisten väliaineen jatkaen siitä toiseen. Säteiden siroamista tapahtuu useaan kertaan, läpäisten ensin materiaalin pinnan ja tuoden heijastuksia takaisin sen sisältä takaisin pintaan. Diffuusimateriaalit voivat imeä valonsäteet kokonaan itseensä, jos säteiden sironta matkaa kauan materiaalin sisällä. (Mts, 24.)

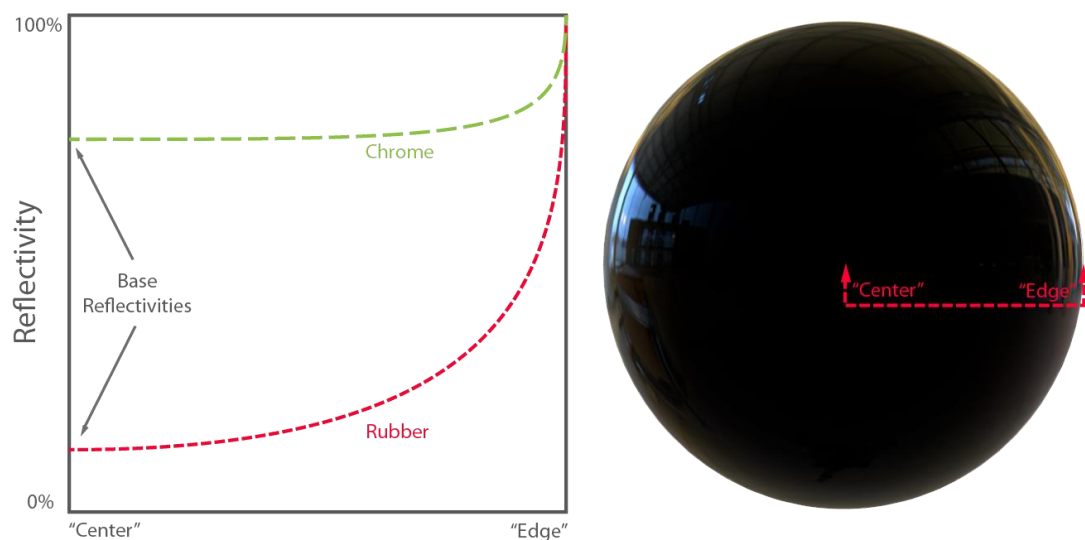
Mikropintateoriassa niin valoa taittavat kuin heijastavat materiaalit sisältävät pinnoillaan epätasaisuuksia. Tämä näyttäytyy reaali maailmassa siten, että valoa taittavassa materiaalissa pinnoilta heijastava valo on vähemmän näkyvää koska sirontaa tapahtuu materiaalin sisällä. Pintojen epätasaisuudet tuottavat sumean heijastuksen materiaalin pinnalta valonsäteiden hajonnan takia. PRB-materiaaleissa mikropinnat sisältyvät karheus- tai kiiltokarttaan riippuen valitusta työnkulusta. (Mts, 25.)

Värit, jotka näemme materiaalin pinnoilla, johtuvat valon aallonpituuksista, joita pintamateriaali heijastaa takaisin. Aallonpituudet imeytyvät materiaaliin ja

heijastuvat takaisin niin diffuusiivisesti kuin myös spekulaaristi. Punaisen omenan kuori heijastaa takaisin näkyvästi vain punaiset valonsäteet ja toisten sävyjen imeytyessä siihen. Korkeasti heijastavat pinnat tuottavat saman valon sävyn takaisin pinnastaan, koska ne johtavat huonosti sähköä. (Mts, 28.)

5.1 Fresnel-heijastus

Fresnel-heijastus käsittää sen, kuinka valon osat heijastuvat optisesti tasaiselta pinnalta, eli sisään tulevan valon kulman ja materiaalin pinnan heijastuksen. Fresnel-heijastuksessa arvot ovat 0 – 100%, koska pintojen heijastus ei voi olla alle 0% tai yli 100%. Sisään tulevan valon kulman ollessa 0-45 astetta pinnan heijastavuus pysyy lähes vakiona. Kulman kasvaessa 45-75 asteeseen heijastavuus kasvaa huomattavasti ja kulman ollessa 75-90 astetta materiaalin pinnan heijastus kasvattaa 100% heijastavuuden, joka näyttäytyy materiaalin pinnoilla valkoisena. Spekulaarisesti heijastavien materiaalien Fresnel arvona voidaan pitää $F(0^\circ)$ koska materiaalin heijastavuus pysyvät suurimmaksi osaksi tämän arvon sisällä. (Hoffman 2010, 13.)



Kuva 6. Fresnel heijastus. (Russell, N.D.)

PBR-tekstuurien valmistuksessa F0 heijastavuusarvot ilmaistaan epämetalleissa harmaasävyä arvoina 0.02-0.05 ja metalleissa RG arvoina 0.5-1.0. Sähköä johtamattoman tasaisen pinnan F0 arvossa pinnan valon heijastavuus on täten 2-5 % ja materiaalin reunoilla valon heijastavuus on 100%. (McDermott 2018, 32.)

5.2 Sähköä johtavat ja eristävät materiaalit

PBR-materiaalit voidaan luokitella pinnoiltaan metalleiksi, jotka johtavat sähköä ja ei-metalleiksi, jotka eristävät sähköä. Metallisten materiaalien pinnassa oleva sähkökenttä heijastaa valon elektroneja ja imee taittavat valonsäteet itseensä. Täysin kiiltävapintaisen metallin valonheijastavuusarvot voivat olla 70-100%. Koska metallit imevät eri valon aallonpituuksia, tulee niiden ominainen väri imeytyneestä valon aallonpituudesta. Metalliset materiaalit, joiden pinnoilla on korroosiota, likaa tai maalia, niitä ei voi pitää enää sähköä johtaviksi. Sähköä huonosti johtavissa materiaaleissa valonsäteiden heijastus on heikompaa, säteiden sirpaloituessa tai imeytyessä itse materiaalin sisälle. (Mts, 37.)



Kuva 7. Sähköä eristäviä materiaaleja. (Nichols, C. 2018.)



Kuva 8. Sähköä johtavia materiaaleja. (Nichols, C. 2018.)

6 PBR-tekstuurikartat

Fyysisesti realististen tekstuurien valmistuksessa artistien on ymmärrettävä kuinka eri työtavat ja tekstuurit vaikuttavat mallien lopputulokseen. Valmistettavan materiaalin luonne, eli onko tarkoituksena valmistaa karhea vai kiiltäväpintaista materiaalia vaikuttaa teksturienvalmistuksen työkulkuun. Mallinnusohjelmien ja pelimoottorien varjostinyksikön tehtävänä on laskennalliset tehtävät kuten valon heijastavuus materiaalin pinnoilta, mutta artistin on ymmärrettävä kuinka varjostinyksikölle syötetty tekstuurikartta vaikuttaa materiaalin ulkonäköön. (McDermott, 2018. 44-46.)

6.1 Metalli- ja karheustyönkulku

Artistin luodessa materiaalia, joka jäljittelee metallia, on otettava huomioon pohjavärikartassa (base color), ettei kyseisessä tekstuurissa käytetä liian tummia tai vaaleita sävyjä. Pohjavärikartan väri heijastaa materiaalissa valoa, eikä siihen tule lisätä varjostuksia. Värikarttaan voidaan lisätä mikrovarjostuksia,

niissä tapauksissa, joissa varjostimen AO-kartan ominaisuudet eivät riitä tuottamaan toivottua materiaalin ulkonäköä. Metallista materiaalia valmistettaessa pohjavärikartan tummat sävyt eivät saisi alittaa 30–50 sRGB:n arvoa ja kirkkaat sävyt ylittää 240 sRGB arvoa. Väriarvot, jotka alittavat tai ylittävät edellä mainitut lukemat eivät tuota realistisia heijastuksia valon säteiden niihin osuessa. Artistin luodessa pohjavärikarttaan tietoa metallissa olevasta liasta tai hapettumasta, kyseiset kohdat vähentävät metallin heijastavuutta ja tällöin materiaalia ei voi kutsua enää puhtaaksi metalliksi. Pohjaväriässä olevat hapettumat ja lian väritiedot on otettava myös huomioon metallivärikartassa. (Mts. 51-52.)



Kuva 9. Metall- ja karhauskartan tekstuurit. (McDermott, W. 2018.)

6.2 Metallikartta

Metallikartta määrittää onko materiaali metallia vai ei. Metallikartta toimii mustavalkoisena tekstuurina varjostimessa maskin tavoin, jossa musta (0.0 sRGB) väriarvo ei tuota metallin heijastusta ja puhdas valkoinen (255 sRGB) heijastaa puhtaan metallin tavoin. Varjostinyksikkö tarkistaa metallikartan mustavalkoiset sävyt ja yhdessä pohjavärikartan väritietojen kanssa tuottaa materiaalin pinnan heijastavuusarvot, jossa mustan sävyt eivät läpäise pohjakartan värejä

ja valkoisen sävyjen läpäisevän ne täysin. Puhtaan metallin pinnan heijastavuus arvot ovat 70-100 %. Tällöin mustan väriarvot ovat metallikartassa 235-255 sRGB ja pohjavärikartassa 180-255 sRGB. (Mts, 54-55.)

Metallikartan avulla vaikutetaan materiaalin ulkoisiin tekijöihin kuten likaan ja metallin hapettumaan. Lian ja hapettuman kohdilla metallisen materiaalin pinta on vähäisesti heijastava, jos ollenkaan. Kyseiset kohdat materiaaliin muodostetaan metallikartan ja pohjavärikartan väriarvojen avulla. Metallikartassa lika määritellään mustan eri sävyillä kuitenkin niin, etteivät ne saa ylittää 235 sRGB:n arvoja. Tapauksissa, joissa mustan sävyjen arvon ylittäessä on muokattava pohjavärikartan tuottamaa metallin heijastavuutta pienemmäksi. Kahden eri tekstuurikartan avulla materiaalin pintoihin saadaan korkeasti ja vähäisesti heijastavia pintoja. (Mts, 55-58.)

6.3 Karheuskartta

Kuten metallikartta karheuskartta käyttää mustan ja valkoisen sävyjä muodostaakseen materiaalin pinnasta valon heijastuksia. Karheat kohdat tuottavat epäsuoria valon heijastuksia pinnasta, näyttäytyen tummina sävyinä materiaalin pinnassa. Täysin sileät kohdat heijastavat niihin osuvan valon kirkaampana kuin iskeytyvä valonsäteet. Karheuskartassa musta tuottaa materiaalin pinnasta täysin sileän ja valkoinen sävy karhean pinnan. (Mts, 59-60.)

6.4 Spekuulaari- ja kiiltotyönkulku

Spekuulaari- ja kiiltokartta työnkulussa fyysisesti realististen karttojen valmistamiseen käytetään kolmea erilaista tekstuurikarttaa, diffuusi-, spekulariittiivi- ja kiiltotekstuurikartat. Kuten metalli- ja karheuskartta työnkulussa, erilaisten tekstuurien yhdistelmänä varjostin muodostaa aidon näköisen materiaalin. Spekuularityönkulussa artistilla on mahdollista tuottaa vääränlaisia heijastavuusarvoja ja ne voivat ylittää luonnollisen valon heijastuksen, jos diffuusi- ja spekuulaarikarttoina on arvoina puhdas valkoinen. (Mts, 64-65.)

6.5 Spekulaari

Spekulaarikartta määrittää heijastavuusarvot metallille ja epämetallille. Spekulaariryökyssä metallin pintaheijastukset ja heijastamattomuuden kohdat saadaan yhdellä RGB-kartalla. Puhtaan metallin pinnalle tehtävät hapettumat ja liat määritetään diffuusiokarttaan korkeammilla väriarvoilla ja laskemalla spekulaarikarttaan tummilla sävyillä. Heikosti heijastavien pintojen väriarvot kartassa ovat 40-75 sRGB ja korkeasti heijastavien pintojen 180-255 sRGB. (Mts, 67-70.)



Kuva 10. Spekulaariryökylun diffuusi- ja kiiltotekstuurikartat (McDermott, W. 2018).

6.6 Diffuusikartta

Diffuusikartta käsittää materiaalin värin ilman heijastavuusarvoja kuten metalli- ja karheuskarttojen pohjavärikartalla. Puhtaan metallin arvoina tekstuurikartassa käytetään 0.0 sRGB mustan arvoja. Tummiin sävyjen arvot eivät saa alittaa 30 sRGB, ellei kyseessä ole puhdas metalli. Vaaleat arvot eivät saa ylittää 240 sRGB arvoja. (Mts, 65-67.)

6.7 Kiiltävyysskartta

Kiiltävyysskartta (Glossiness map) määrittää tässä työkulussa mustavalkoisena värikarttana materiaalin pinnan kiiltävyyden ja karheuden. Päinvastoin karheuskartan arvoja tässä kartassa puhtaan valkoisen arvo tuo pintaan karheuden ja täysin musta on kiiltäväpintainen. (Mts, 70.)

6.8 Normaalikartta

Normaalikartta (Normal Map) määrittää materiaalin pinnan yksityiskohdat. Kyseinen RGB-värikartan ominaisuutena on muodostaa syvyyttä, halkeamia ja yksityiskohtia ilman, että yksityiskohdat lisättäisi itse 3D-malliin. Tekstuurikartan R, G, B arvot ovat samat kuin käytössä olevan polygonin X, Y ja Z-koordinaatit. (Denham n.d.)

6.9 Ympäristön okkluusio

Ympäristön okkluusio (Ambient Occlusion) kartta määrittää kuinka paljon ympäristön valaistus vaikuttaa pinnan korkeisiin ja syvimpiin pisteisiin. Kyseinen mustavalkoinen tekstuuri vaikuttaa diffuusikartan arvoihin ja voimistaa ympäristön valaistusta mallissa. Ympäristön okkluusio-tekstuurikartta on omana valinnaisena karttana. Tekstuurikartan ominaisuuksia ei myöskään PRB-materiaalia valmistettaessa saa lisätä muihin tekstuurikarttoihin. (McDermott 2018, 74.)

6.10 Korkeuskartta

Korkeuskartta (Height Map) lisää 3D-mallin polygonilukua lisäten tarkkuutta ja luoden mallin pinnalle syvyyseroja kuitenkin ilman, että mallia itsessään muokattaisiin. Korkeuskartan ominaisuudet lisäävät mallin pinnalle reaalisuutta korostaen normaalikartan ominaisuuksia. Kuten AO-kartta myös korkeuskartta on valinnainen tekstuuri PBR-materiaaleissa. (Mts, 75-76.)

7 Tutkimustulokset

7.1 Alkukartoitus oppaan tarpeista

Physically based renderer -tekniikan tarpeellisuudesta tietojenkäsittelyn opiskelijoille on aluksi havainnoitu tutkijan toimesta. Keskustelu opettajan kanssa oppaan tarpeellisuudesta toi tutkimuksen aiheeseen varmuutta. Aineiston keruu tapahtui teemahaastattelujen avulla. Haastateltaviksi valikoitui kolme tietojenkäsittelyn opiskelijaa, joilla oli aiempaa kokemusta tekstuurien valmistuksesta.

Teemahaastattelut tapahtuivat verkon yli ja kyseiset haastattelut tallennettiin. Haastattelujen jälkeen aineisto litteroitiin ja samankaltaisia vastauksia luokiteltiin.

7.2 Oppaan käyttötarkoitus opiskelijoille

Oppaan käyttötarkoituksesta alkuhaastattelussa ilmeni, ettei kaikkia ohjelman toimintoja tarvitse käydä läpi yksityiskohtaisesti, vaan opastaa kuinka sillä luodaan PBR-materiaali.

”Oppaan tulisi olla mahdollisimman visuaalinen, että näytetään ennemmin kuin kirjoitettaisi ja kerrotaisi toiminnoista.”

”Esimerkin kautta oppiminen auttaa ymmärtämään kokonaisuutta paremmin”

”Kun osaa enemmän voi hakea ja etsiä ratkaisuja tekstistä”.

”Kun on taitoa käyttää ohjelmaa, voi käyttää tekstiä tai wikikirjastoa, josta selviää kaikki ohjelman ominaisuudet”.

Haastateltavilta saaduissa palautteissa ilmeni usein, että oppaan tulisi olla myös mahdollisimman visuaalinen, jolloin oppaassa olevista kuvista voisi seu-

rata miten materiaalia valmistetaan. Haastattelun tuloksina selvisi, että oppaassa kuvien tulisi olla tekstiä suuremmassa roolissa. Ohjelmaa entuudetaan osaaville toimisi opas, jossa tekstit ovat kuvien sijasta suuremassa roolissa.

7.3 Oppaan valmistus

Physically Based Renderer -materiaalin luomiseen oppaaseen muodoksi valikoitui kivitietekstuuri, jonka ulkonäköä opasta seuraavan on helppo muokata mieleisekseen käyttötärpeen sopivaksi. Kyseisen tekstuurin valmistukseen tarvitaan myös ohjelman perustoimintoja ja toistoja, jolloin ohjelman toimintaperiaatteet tulevat selkeämmäksi. Ennen työnkulun opastusta oppaassa käydään läpi ohjelman perusikkunoita ja niiden käyttötarkoituksia.

Ohjelman proseduraalinen työnkulku selkenee hyvin nopeasti materiaalin pohjamuotoa tehdessä, jolloin kaikki kyseisen toiminnon arvot olivat vapaasti säädettävissä oman mielen mukaan ja tehtävä muoto päivittyy reaaliaikaisesti arvoja muutettaessa. Muotoon pystyi vapaasti lisäämään ulkonäköä muuttavia toimintoja silmukoiden avulla. Työnkulussa pystyi palaamaan takaisin jo tehtyihin arvoihin ja muuttamaan tai poistamaan niitä mieleisekseen ilman, että jo valmistettu muoto tuhoutuisi täysin ja materiaalin valmistaminen tulisi aloittaa alusta.

Substance Designer -ohjelmassa materiaalin muotoa valmistetaan pitkään pelkillä korkeus-, normaali- ja ympäristön okklusiokartoilla. Näiden karttojen avulla pystyy tekstuurin pinnoilla olevaa korkeutta havainnoimaan, lisäämään pinnalle yksityiskohtia ja näkemään selkeämmin muotojen korkeuserojen varjostusta. Aluksi vain kyseisiä tekstuurikarttoja käyttämällä tehtäviä muutoksia ja niiden vaikutuksia valmistettavan materiaaliin on helppo seurata. Pohjan muotojen valmistuksen jälkeen väritekstuurikartat tuovat tietoa siitä, mitä kukin elementti materiaalissa on. Muotojen värit yhdistetään ohjelmalla, jolloin kaikki materiaalissa olevat sävyt ovat yhdessä väritekstuurikartassa. Karheuskartan ominaisuutena on tuoda mustilla- ja vaaleilla materiaalin elementeille eri kohtaan karkeaa tai kiiltävää pintaa.

7.4 Oppaan testaus haastateltavilla

Oppaan valmistuttua haastattelujen avulla pyrittiin varmistamaan sen informatiivisuus ja selkeys. Haastateltavilta saatu palaute antoi tietoa, kuinka oppaan luettavuutta pystyisi parantamaan. Haastatteluissa nousi esille, että tekstissä olevia avainsanoja tulisi korostaa. Saatujen tulosten jälkeen opasta on muokattu palautteen mukaan.

”Tekstissä olevia avainsanoja voisi korostaa esimerkiksi lihavoimalla.”

”Tärkeintä osaa voisi alleviivata tai kehystää”.

”Kuvat ovat selkeitä, eikä mitään ole jätetty välistä pois”

”Kuvakaappaukset ovat selkeitä ja ne näyttävät noden asetukset, jolloin niistä voi tarkistaa mitä käytännössä tapahtuu”

Haastateltavista kaksi ennättivät testaamaan oppaan toimivuutta käytännössä, ja heiltä saatu palaute varmisti oppaan toimivan opastamaan PBR-materiaalin valmistamisessa. Saadusta palautteesta pystyi päättämään, ettei ohjelman käyttöliittymää ja sen toimintojen tarkempi läpikäyminen ole tarpeellista materiaalin valmistuksessa. Palautteen mukaan myös oppaan avulla valmistettavassa materiaalissa käytetään toistoja ja usein samoja toimintoja, jotka antavat ymmärrystä siitä, kuinka ohjelmassa työnkulku ja toiminnot toimivat.

8 Johtopäätökset

Haastatelluilla oppilailla ei ollut aiempaa käytännön tietoa PBR-materiaaleista ja millaisia tekstuurikarttoja ne vaativat. Haastatellut olivat työskennelleet aiemmin tekstuurikartoilla, jotka olivat värikartta- ja normaalikarttatekstuurit. Kyseisiä tekstuurikarttoja oppilaat olivat valmistaneet kuvankäsittelyohjelmalla, eikä Substance Designer ja sen proseduraalinen työnkulku ollut entuudestaan tuttu.

Kehittämistutkimuksen tuloksena syntynyt opas tuo sen käyttäjälle tietoa mil-laista hyötyä proseduraalinen työnkulku tuo sisällöntuottajille tekstuurien val-mistuksessa. Proseduraalisessa työnkulussa on artistin mahdollista muuttaa tekstuurien ulkonäköä ja niiden ominaisuuksia missä vaiheessa tahansa. Val-miista tekstuurista on mahdollista tehdä erilaisia variaatioita tai vaihtaa reso-luutiota käyttötärpeiden mukaan. Perinteisissä kuvankäsittelyohjelmissä nämä muutokset ovat hankalammin toteutettavissa ja vaatisivat useasti enemmän työtä, pahimmassa tapauksessa koko työn aloittamista alusta.

Physically based -materiaali tuottaa aidon näköisiä materiaalia kaikissa valai-suolosuhteissa. Tektuurikarttojen ominaisuuksia yhdistelemällä materiaalille ar-tisti voi luoda oman tarinan millainen ympäristö on muokannut pintoja sel-laiseksi kuin ne ovat. PBR-tekniikan avulla materiaali näyttää aidolta kaikissa mahdollisissa valaistusolosuhteissa.

Tietojenkäsittelyn opiskelijoilta vaaditaan paljon omaa oppimista ja tie-donetsintää. Oppaan tarkoituksena on antaa vaihtoehtoinen tapa saada tietoa ilmiöstä ja kehittää sen kautta omaa oppista. Haastatteluissa selkeni myös, että opetusvideoiden seuraaminen ei ole kaikille luonnollisin tapa oppia uutta.

9 Pohdinta

Tutkimustuloksissa ilmeni, ettei sisällöntuottajilla ollut aiempaa tietoa PBR-materiaaleista ja niiden toteutuksista, vaikka heillä oli aiempaa kokemusta tekstuurien valmistuksesta. Haastattelujen avulla useampi vastaaja kertoi oppaan ohjeistuksen tarpeellisuuden käsittelemään PBR-materiaalin valmista-mista, eikä niinkään pelkän ohjelman käyttöön. Oppaan testauksen jälkeen useampi haastateltava antoi parannusehdotuksia oppaaseen pyytämällä pie-niä muutoksia korostamaan tärkeimpien toimintojen korostamista ja tuomaan selkeämmin esille tekstin ja kuvien välistä yhteyttä. Useampi vastaaja sanoi opasta toimivaksi ja selkeästi seurattavaksi. Oppaan avulla ohjelmaa käyttä-neet antoivat varmistuksen siitä, että oppaan sisältämä ohjeistus on toimivaa.

Haastateltavilta saaduissa vastauksissa ilmeni samoja tutkimuksen tuloksia, eli saadussa aineistossa tapahtui kylläntymistä. Alkuhaastattelussa haastateltavilta Tutkimuksessa saadut tulokset eivät ole yleistettävissä, vaan ne kertovat pelkästään valmistetun oppaan tarpeellisuudesta Jyväskylän Ammattikorkeakoulun Tietojenkäsittelyn opiskelijoille, jotka valmistavat tekstuureita.

Oppaan rakentamisessa teoreettisen viitekehyksen tiedot auttoivat ymmärtämään paremmin millaista tietoa ja millaisessa väriavaruudessa eri tekstuurikartat tulevat pitämään sisällään. Physically based renderer -materiaalit pohjautuvat reaali maailman arvoihin, joten on myös olennaista ymmärtää kuinka valonsäteet käyttäytyvät erilaisilla pinnoilla. Tällöin artistin on mahdollista luoda aidon näköisiä materiaaleja.

Substance Designer –ohjelman käyttöönotto on tekstuuria aiemmin valmistaneelle helpompaa mutta ohjelman kaikkien ominaisuuksien ja toimintojen opettelu työläämpää. Työtapoja ja mahdollisuuksia ohjelman sisällä on useampia, eikä siinä ole aina vain tiettyä yhtä tapaa tuottaa tiettyä ulkonäöllistä muutosta. Yksi ohjelman suurimpia etuja on tuhoutumaton työnkulku, jolloin tekstuureita on helppo päivittää ja muokata tarpeitten mukaan. Ohjelmalla valmistetut tekstuurit, oikein tehtynä ovat ulkonäöltään ensiluokkaisia.

Oppaan avulla valmistettavassa materiaalissa käytetään karheita ja kiiltäviä pintoja tuomaan niihin oikeanlaista eloa. Metallisen materiaalien valmistukseen aika ei riittänyt, joten opasta voisi täydentää sähköä johtavan materiaalin opastukseen. Oppaan avulla valmistettava teksturi on helposti käytettävissä tasaiselle 3D-mallille. Opasta jatkokehitettäessä voidaan käydä läpi myös erilaisia muotoja tai tekstuurien valmistuksen tiettyyn 3D-malliin. Oppaan käyttämä formaatti myös toi taitollisia rajoituksia, jolloin kuvaan on saatava mahdollisimman paljon informaatiota ja kuitenkin niin, ettei kuvat vie sivujen koko pinta-alaa. Kuten haastatteluissakin ilmeni, ettei tutkimuksessa käytetty formaatti ole kaikille sopivin. Jatkokehityksessä videoformaatti voisi antaa myös erilaisia vapauksia opastamaan ohjelman käytössä.

10 Lähdeluettelo

Akenine-Möller, T., Haines, E. Real-Time Rendering, Second Edition. A K Peters, Ltd. 2002

Bister, T. 2019. Tietojenkäsittelyn opinnäytetyö, viittoja ja karttoja tutkimisen ja kehittämisen teille. Jyväskylän Ammatikorkeakoulu.

Denham, T. n.d. Texture Maps: The Ultimate Guide For 3D Artists. Viitattu 26.10.2020.

<https://conceptartempire.com/texture-maps/>

Hoffman, N. 2010. Physically-Based Shading Models in Film and Game Production. Viitattu 26.10.2020.

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

Hughes, J., Van Dam, A., Mcguire, M., Sklar, D., Foley, J., Feiner, S., Akeley, K. 2014. Computer Graphics Principles and Practice, Third Edition. Pearson Education, Inc.

Kananen, J. 2012. Kehittämistutkimus opinnäytetyönä, kehittämistutkimuksen kirjoittamisen käytännön opas, Jyväskylän ammattikorkeakoulun julkaisuja 134

McDermott, W. 2018. The PBR Guide

<https://academy.substance3d.com/courses/the-pbr-guide-part-1>

Paquette, A. 2008. Computer Graphics for Artists. Springer-Verlag London Limited.

Shirley, P., Marschner, S., Ashikhmin, M., Gleicher, M., Hoffman, N., Johnson, G., Munzner, T., Reinhard, E., Sung, K., Thompson, W., Willemssen, P., Wyvill, B. 2009. Fundamentals of Computer Graphics, Third Edition. A K Peters/CRC Press

11 Kuvalähteet

Nichols, C. 2018. Understanding metalness.
<https://www.chaosgroup.com/blog/understanding-metalness>

CgUsersManual. 2005.

Scratchapixel 2.0. N.D. Introduction to Polygon Meshes.
<https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-polygon-mesh>

The Real-Time 3D Graphics Pipeline Architecture. 2020. Visual Computing Systems Stanford University.
http://cs348k.stanford.edu/spring20content/lectures/13_gfxpipeline/13_gfx-pipeline_slides.pdf

Romanowski, J. 2016. Let's get physical.
<https://www.jeremyromanowski.com/blog/2015/11/20/lets-get-physical-part-1-of-3>

Russell, J. N.D. Basic Theory of Physically-Based Rendering.
<https://marmoset.co/posts/basic-theory-of-physically-based-rendering/>

12 Liitteet

Liite 1. Physically Based Rendering.

https://www.dropbox.com/s/cn0aylrব্যযেজ/PBR_Guide_Tomi_Pulkkinen.pdf?dl=0

Korkearesoluutio versio.



PHYSICALLY BASED RENDERING

TOMI PULKKINEN

JYVÄSKYLÄ UNIVERSITY OF APPLIED SCIENCES



INTRODUCTION

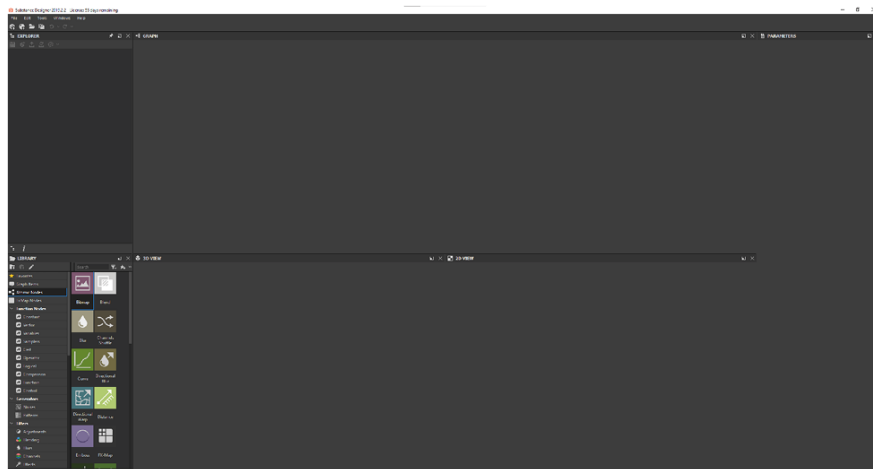
Purpose of this guide is to show one way to create physically based textures. It is intended for students who are not familiar with creating textures in Substance Designer. Substance Designer offers non-destructive way to create procedural textures, and there are multiple ways manipulate shapes and outcome of the final textures. Substance Designer works in node-based graph where you link nodes drawing the lines between them.

Substance offers good online documentation explaining different nodes and how it's parameters works:

<https://docs.substance3d.com/sddoc/substance-designer-102400008.html>

This guide will show how to combine different nodes, how to set up the workflow and how to create PBR-material.

Students are entitled to a free personal licence of Substance Designer.



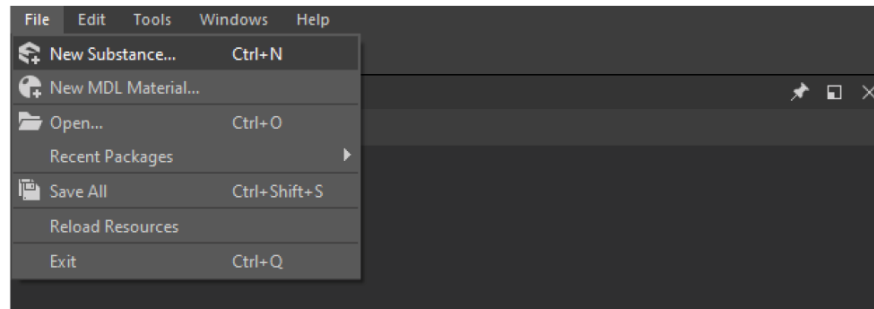
The basic screen offers information on Explorer, Graph, Library, 3D view, 2D view and parameters. Information windows can be undocked and combined to other windows based on user preference.

Library offers all functional nodes from the program.

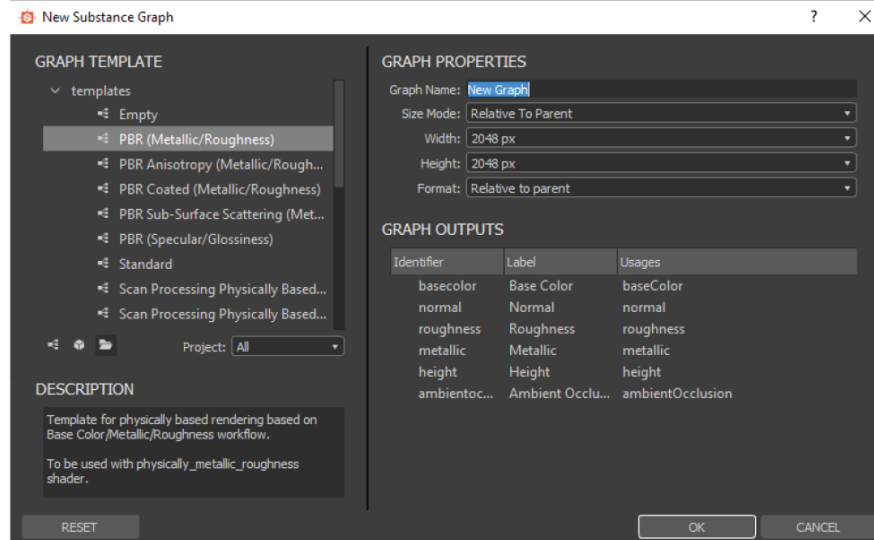
The Graph view shows nodes and outputs.

All outputs from the output nodes can be viewed from the 3D-window, where you can change shape of the 3D-model.

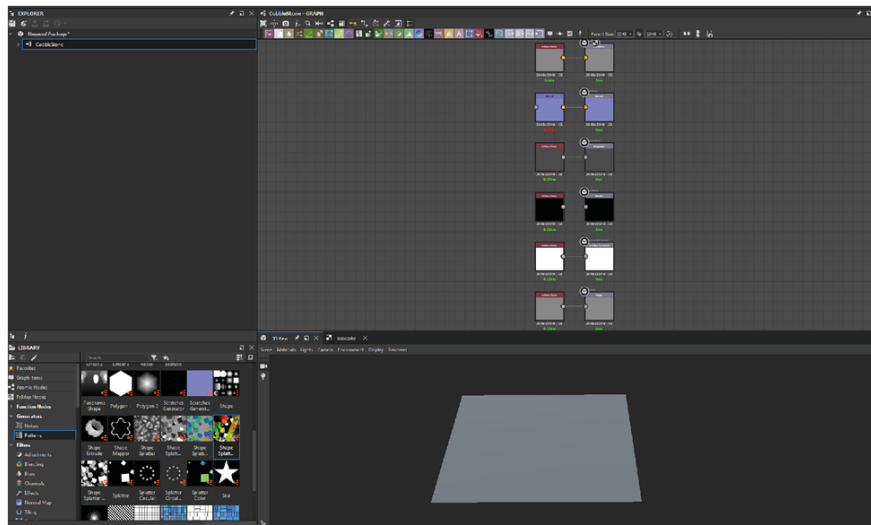
2D-window shows a 2D-image from the selected node.



To create new substance material select **File, New Substance**. In this tutorial, the workflow selected is Metallic/Roughness, as it is most used workflow when creating physically based materials.

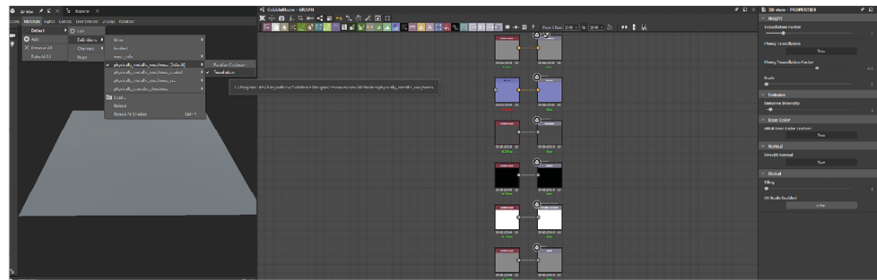


Graph properties show the outputs, graph name, size mode, texture sizes, and format. Size mode and format is set to Relative To Parent, as it is the safest way to control outputs from the nodes and graphs. If needed, all of these are options can be changed in node attributes. In this guide, the only change made is re-naming the Graph to CobbleStone.



Selecting Metallic/Roughness workflow all needed outputs come from template file. Outputs can be deleted or added as required. Graph-name can be seen in explorer under the Unsaved Package section. To save your package, go to **file and save all**.

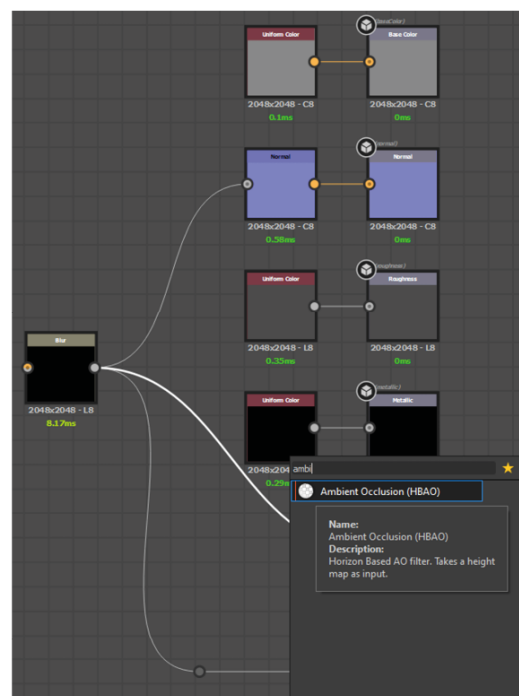
In Substance Designer, user windows can be arranged however the user prefers. In this guide, the 2D-view is docked with Explorer window. This way all changes in nodes can be seen in the 3D-model as it updates in realtime and the 2D-image is bigger. Combining the windows gives more space to see Graph, 3D and 2D-image.



From material selection tessellation scale input can be changed. Increasing Scale number height map tessellation can be seen more clearly.

All needed nodes and functions can be found in the library window or searching for them by pressing the spacebar. To start, **Blur node** is added to connect normal, height and Ambient Occlusion. Also adding **Ambient Occlusion node** gives more control to the height in AO-input. When attaching different nodes pressing ALT-key in node line gives option to reroute the lines. This way the graph becomes easier to read after adding more nodes and lines.

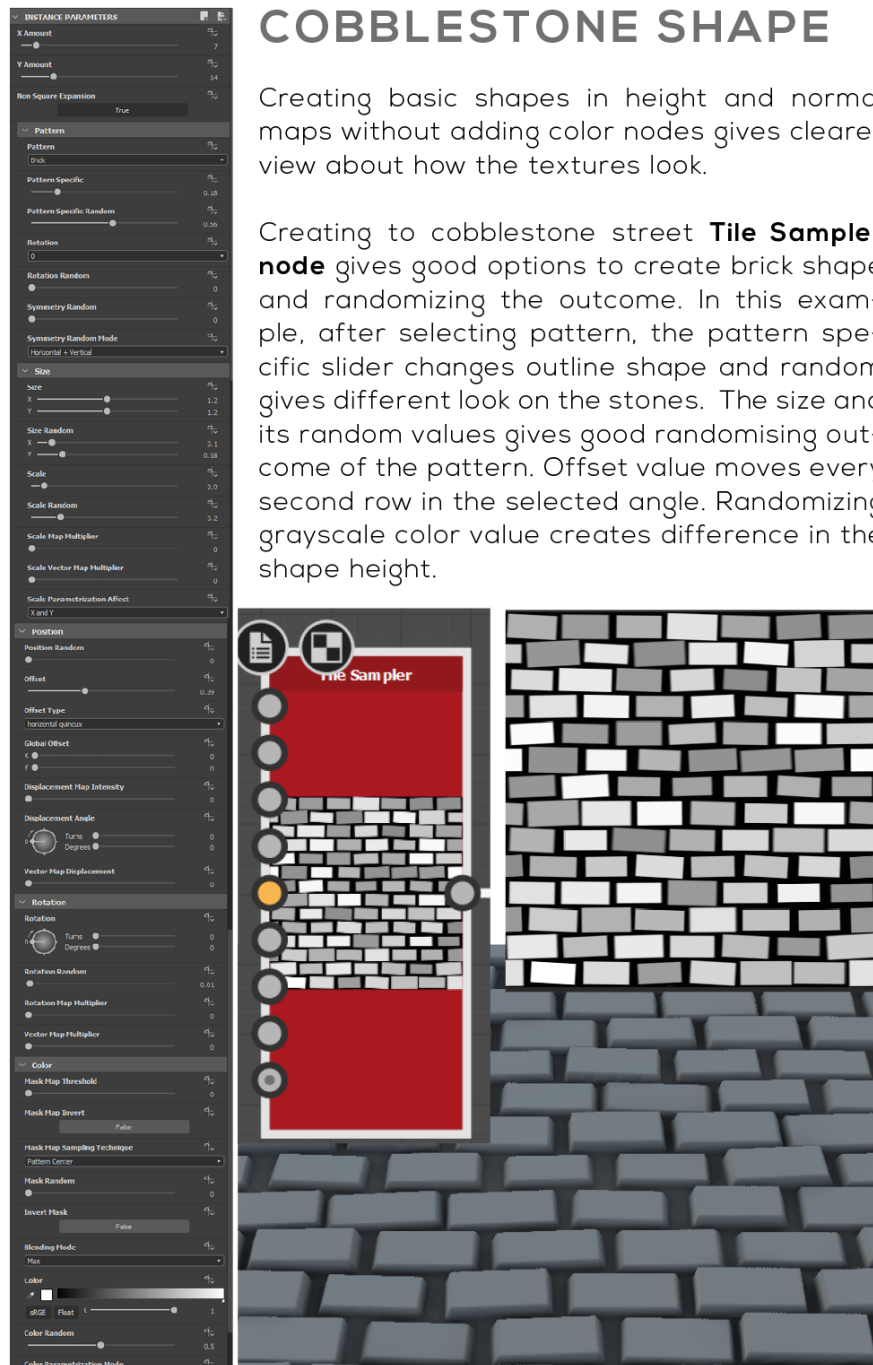
When creating textures in Substance Designer it is good practice to start creating shapes that are needed. Adding detail to shapes comes when progressing creating textures.

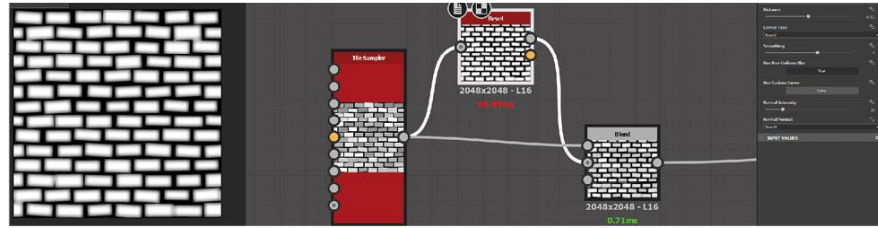


COBBLESTONE SHAPE

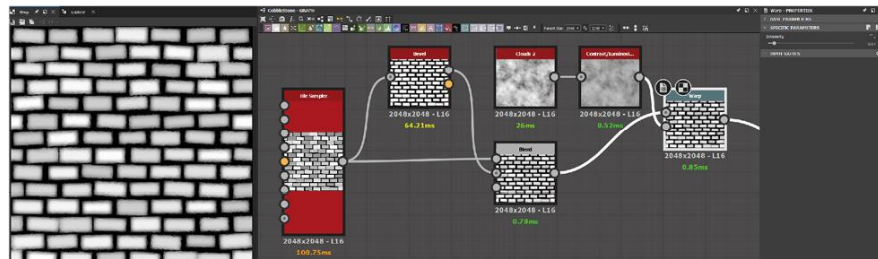
Creating basic shapes in height and normal maps without adding color nodes gives clearer view about how the textures look.

Creating to cobblestone street **Tile Sampler** node gives good options to create brick shape and randomizing the outcome. In this example, after selecting pattern, the pattern specific slider changes outline shape and random gives different look on the stones. The size and its random values gives good randomising outcome of the pattern. Offset value moves every second row in the selected angle. Randomizing grayscale color value creates difference in the shape height.

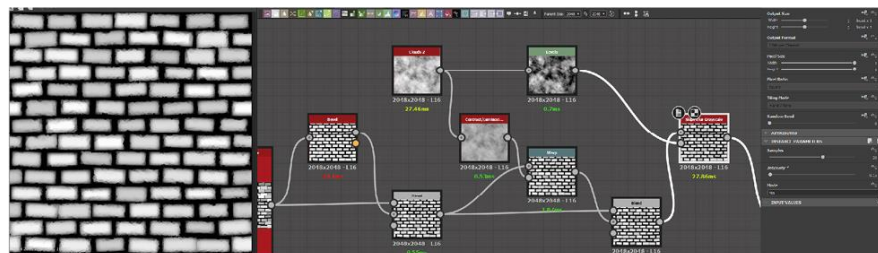




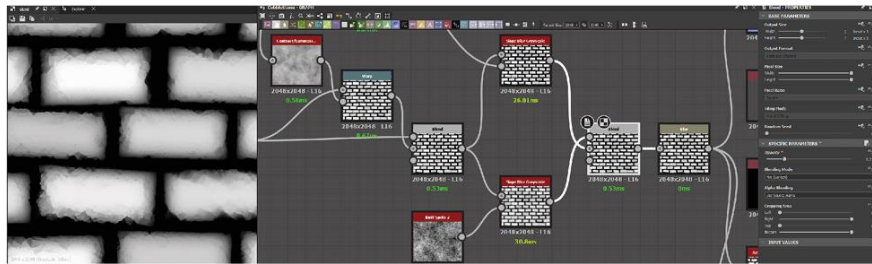
Adding a **bevel node** gives bricks a more rounded shape. Using the **blend node** enables combining of the image from bevel node to original image. Blending mode is set to Min (Darken), as it picks the lower value between the background and the foreground. The opacity is set to 0.5.



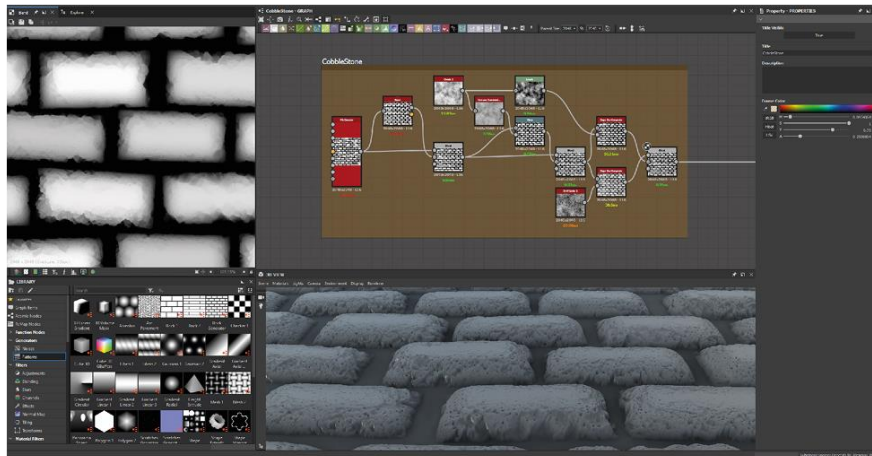
To create cracks to edge of stone **Clouds 2 node** gives irregular grayscale image which contrast can be tweaked combining it with **Contrast/Luminosity Grayscale node**. Smaller contrast variation is needed as the **Warp node** takes gradient values to its input to warp pixels.



Same nodes can be used multiple times. After **Clouds 2 node**, **Levels node** is attached to **Slope Blur node**. Slope blur enables the creation of more wearing effect to the edges as it is set to Min mode.

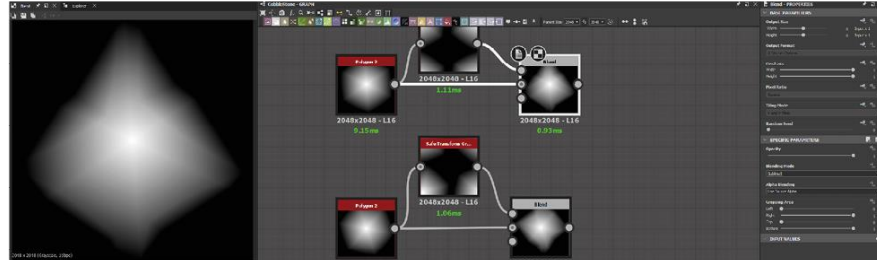


Nodes can be duplicated by pressing CTRL+D. This adds another slope blur, but now with **BnW** the spots node stones will have small holes in it. With blend node both slope blur effects can be blended.

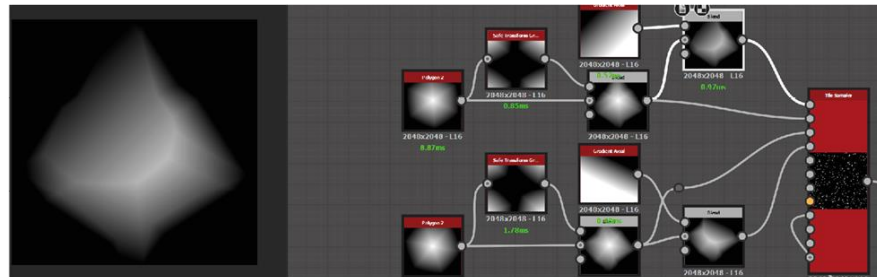


All nodes that are used to create shape of the stone street can be framed. In this way all nodes can be moved at once in the graph and it makes easier to read which nodes effects making the specific shape.

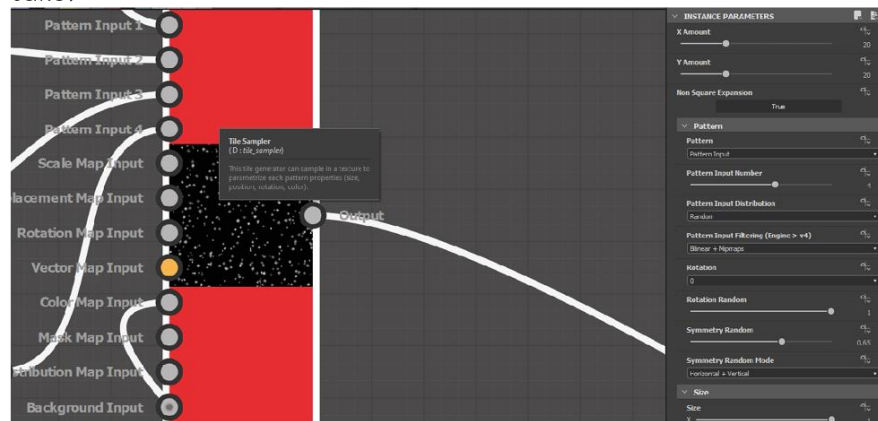
SMALLER STONES

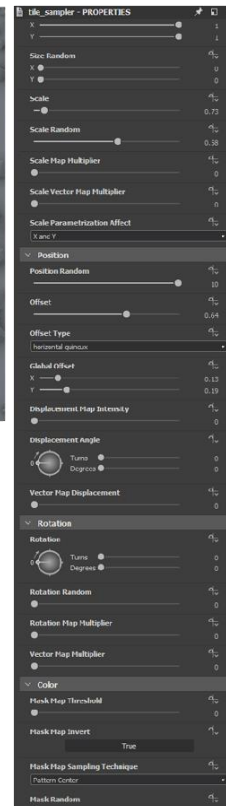
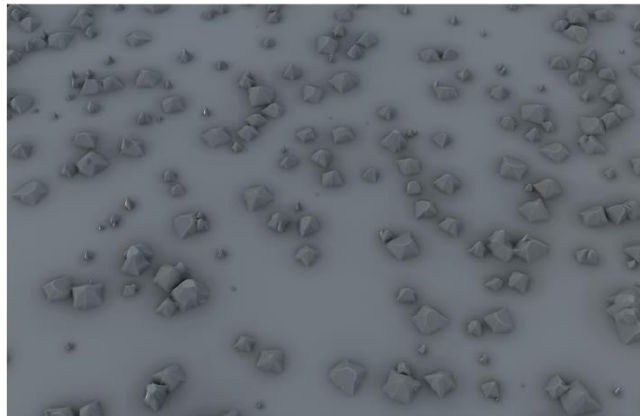


Polygon shape node is duplicated to give two different shapes. In **safe transform grayscale node** image information is rotated and moved to a different position. Blend node is used to subtract two images, and result is more irregular in shape.



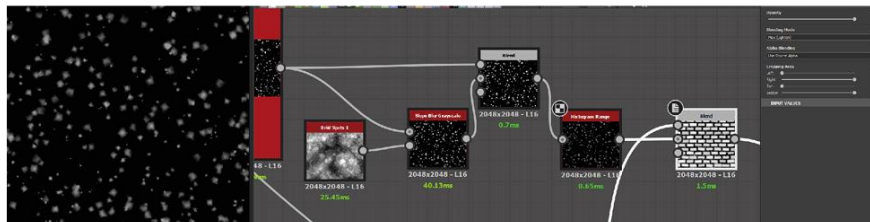
Adding two **gradient nodes** created stone shapes can be varied even more. Now we have four different stone shapes which can be attached to Tile sampler node inputs. Tile sampler pattern needs to be changed to pattern inputs and name how many it needs to take.

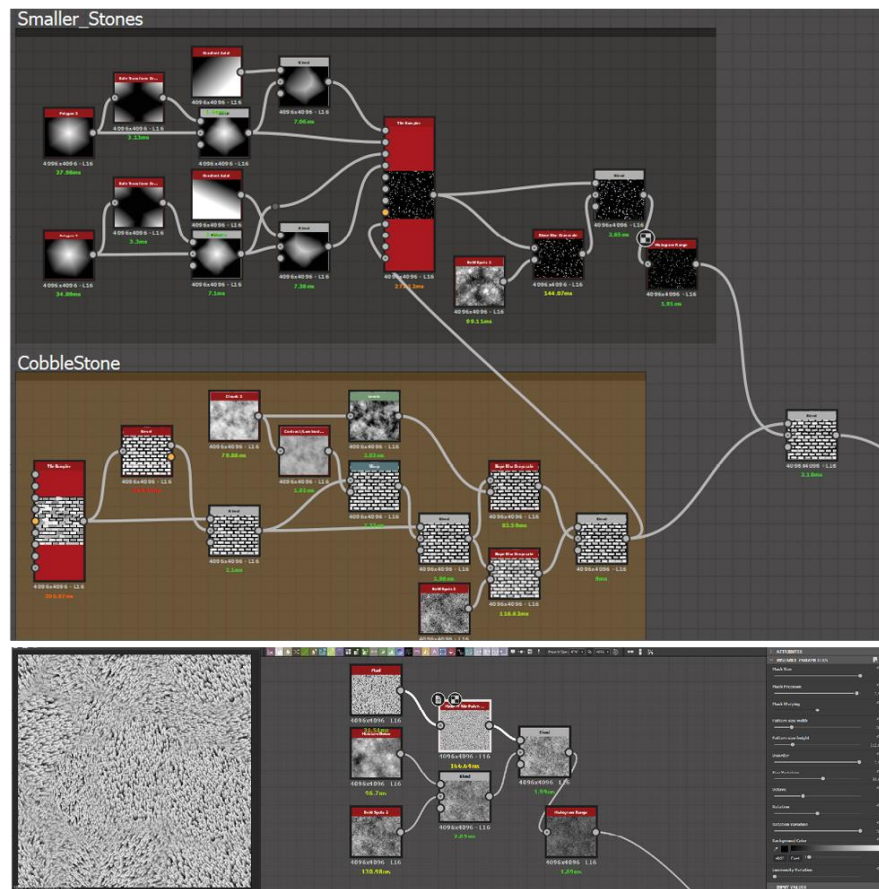




Tile sampler creates variation from four created stones with different sizes. Tile sampler also takes mask map information from created cobblestone shape node.

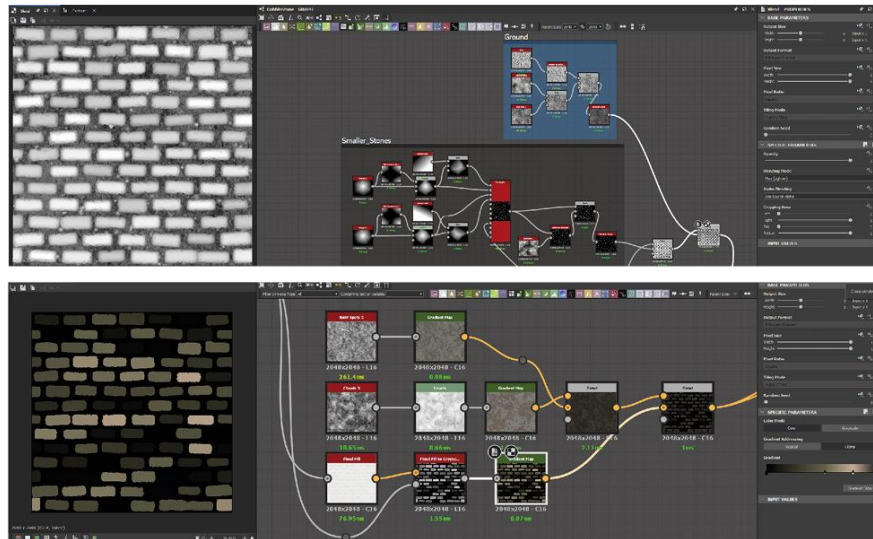
Black and white spots with **slope blur node** is to give rougher edges to the stones. After blending it from original stone pattern image histogram range comes handy to control height of the stones. Stones are intended to be in the cracks of the cobblestone.



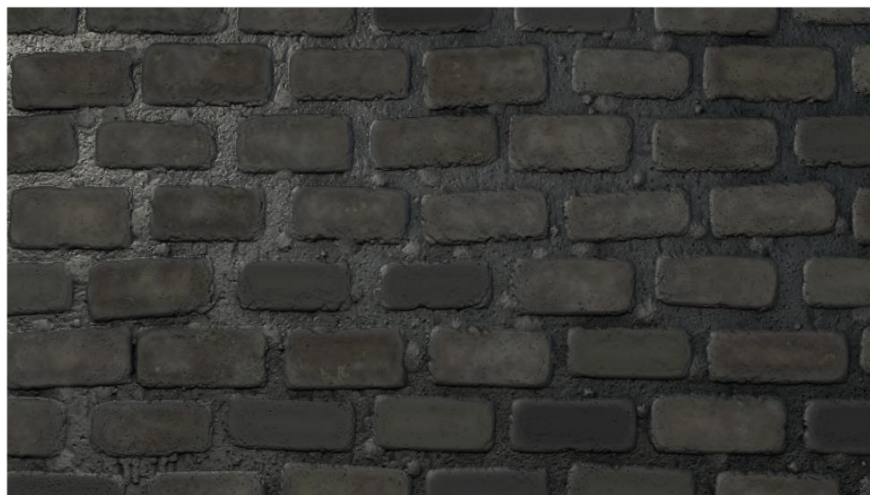


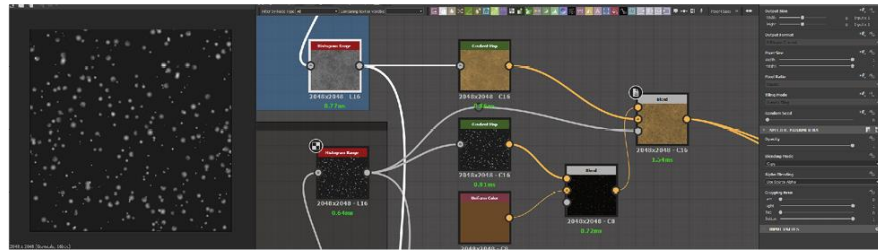
After framing smaller stones the ground shape can be created. Blending **Moisture noise** and **black and white spots nodes** creates a rough base to surface. With **Fluid node** attached to **Make It Tile Patch node** creates high range variation from the shape. After blending Patch node and ground base created Histogram Range node enables to control ground height.

CREATING COLORS

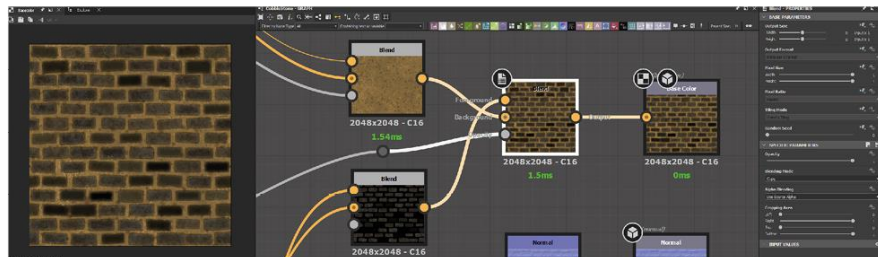


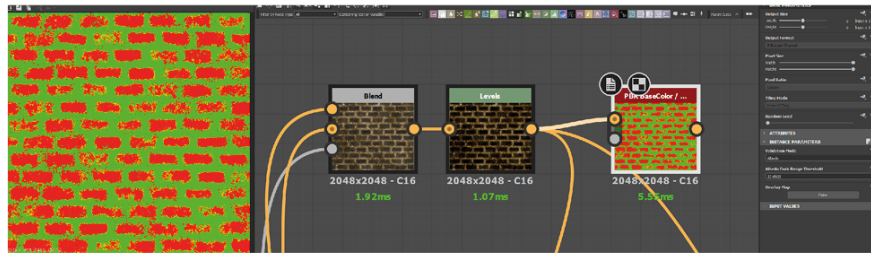
At this point the combined materials are ready to get color nodes. **Flood Fill node** takes Cobble Stone image to its input. This enables attaching Flood Fill to Grayscale where grayscale values are tweaked. Gradient map is used to give different colors to the stones. **Black and white spots** and **Clouds nodes** are connected to Gradient Maps and blended before blending them to the top of the stones Gradient Map.



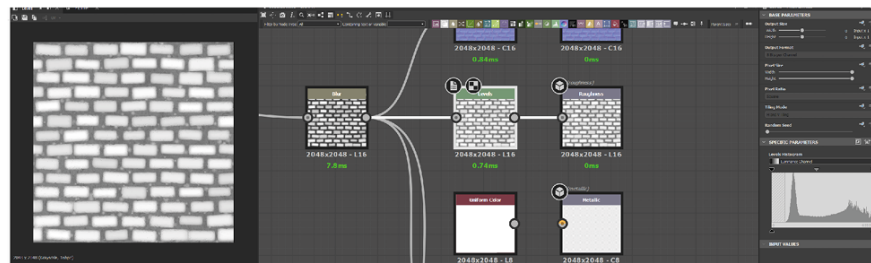


Color of the ground and smaller stones are done with a few nodes. Smaller stones are blended between brown **uniform color** and **gradient map**. When creating colors in Substance Designer **GRAY-SCALE COLORS MUST BE LINKED WITH GRADIENT MAP-NODE**, otherwise node functions will not work. Ground color is mapped with gradient and will take smaller rocks color top of it. Now cobblestones and ground can be blended. **Blend node** takes opacity input from cobblestones grayscale value.





NOTE: Creating metallic colors **PBR BaseColor valitade-node** checks if the dark colors are lower than 30 in sRGB range. Node shows those colors in red in current node. But in current material there is no need to worry about these colors.

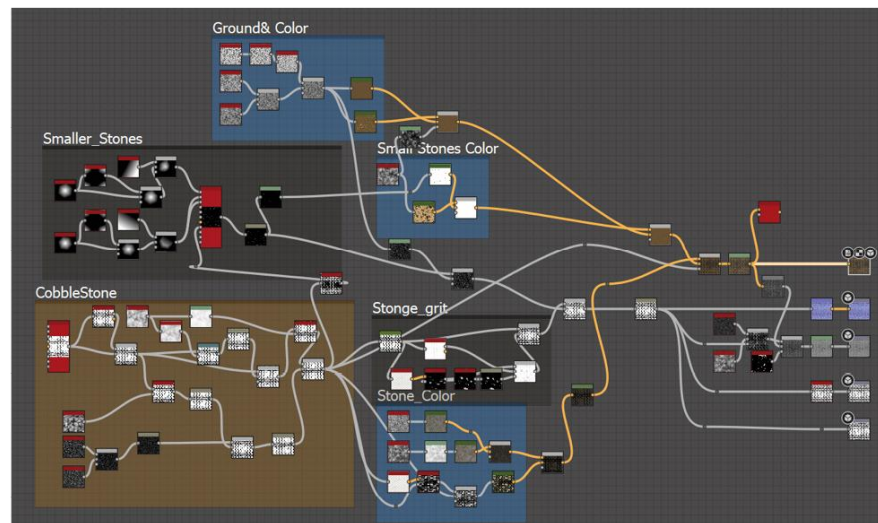


To have better control of the roughness map in **Level node** contrast in black and white values has been decreased to have rougher ground. In level node black and white can be inverted, so the rougher parts will be ground and the smoother parts are the

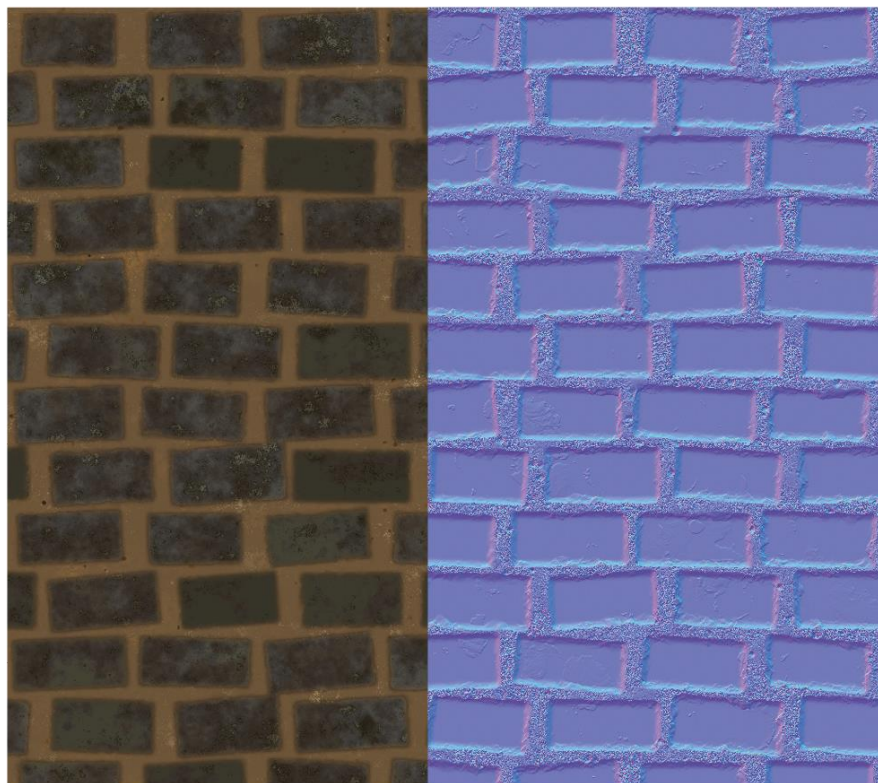


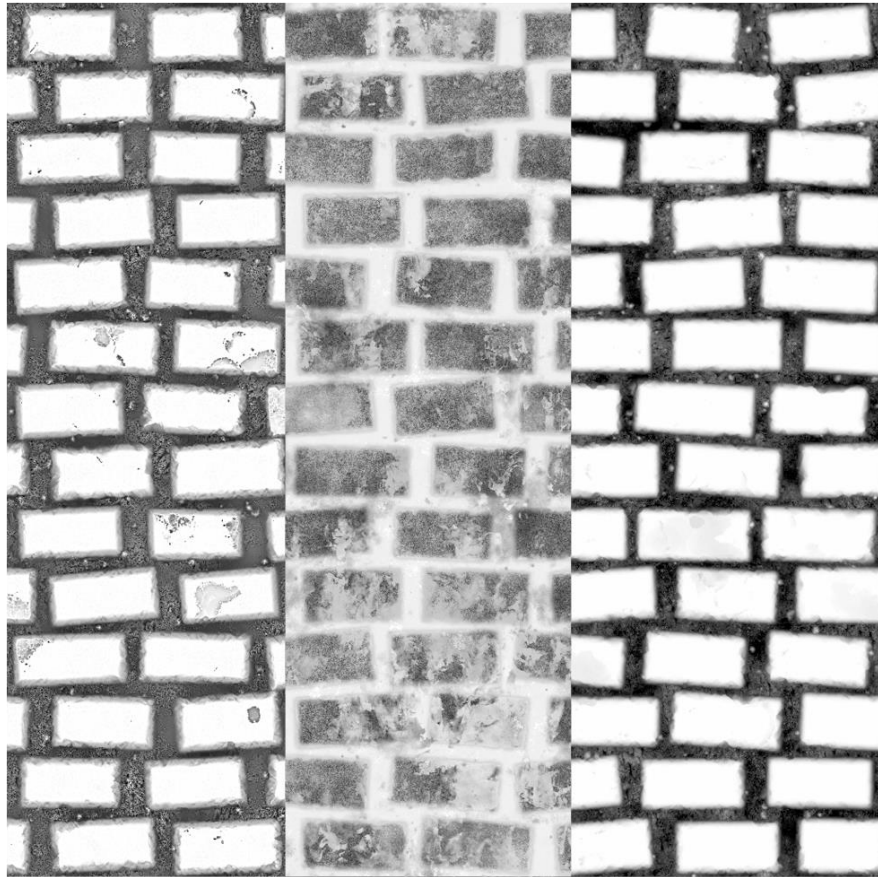
stones.

After coloring the shapes and creating the roughness the procedural workflow shows its power. All created nodes and values can be tweaked without the risk of destroying textures. After tweaking the values and adding more details with more nodes, all used texture maps can be exported by the wrench tool in the graph toolbar.



Final graph after tweaking and adding more nodes.





Texture maps:
Base Color
Normal map
Ambient Occlusion
Roughness Map
Height Map

RENDERS FROM THE MATERIAL



Iray render in Substance Designer before tweaking.



Iray render in Substance Designer after tweaking.