Izrada tekstura za 3D modele

Nakić, Jakov

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:602074

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-04-20**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jakov Nakić

IZRADA TEKSTURA ZA 3D MODELE

DIPLOMSKI RAD

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jakov Nakić

Matični broj: 44918/16-R

Studij: Informacijsko i programsko inženjerstvo

IZRADA TEKSTURA ZA 3D MODELE

DIPLOMSKI RAD

Mentorica:

Prof. dr. sc. Valentina Kirinić

Varaždin, rujan 2022.

Jakov Nakić

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Cilj je bio ovim radom istražiti teksturiranje kao jednu od najvažnijih faza produkcije u 3D računalnoj grafici. Razjašnjeni su pojmovi teksture, materijala i *shadera*, koje 3D artisti često poistovjećuju. Spomenute su proceduralne teksture kao potencijalna zamjena u budućnosti za klasične rasterizirane slike kao teksture. Nabrojane i objašnjene su različite vrste teksturnih mapa čijim kombinacijama 3D modeli dobivaju sva potrebna površinska svojstva materijala iz stvarnog svijeta. Novi popularni trend fizički zasnovanog renderiranja (eng. *physically based rendering*; *PBR*) pridonio je razvoju novih teksturnih mapa iz kojih su se razvila dva najpopularnija PBR teksturna tijeka rada: *metallic-roughness* i *specular-glossiness*. Postoje mnogi programski alati za teksturiranje, a u radu je objašnjeno nekoliko najpopularnijih, od kojih se najviše ističu Substance 3D Designer, Substance 3D Painter, Mari, i Mudbox. Budući da su dobre performanse neophodne za besprijekorno iskustvo u igrama i budući da se osim računala igre razvijaju i za jeftinije platforme, optimizacija je vrlo važna, pa je u radu spomenuto nekoliko tehnika efikasnog teksturiranja. Također je naglašen koncept trošenja materijala i njegova važna uloga u vizualnom pripovijedanju. Na kraju rada detaljno je objašnjen Substance 3D Designer kao odabrani programski alat za teksturiranje korišten u praktičnom primjeru. Cilj je bio kroz praktični primjer primijeniti teorijsko znanje o teksturama i svojstvima površina kako bi se napravila simulacija realističnog materijala iz stvarnog svijeta.

Ključne riječi: teksturiranje; UV mapiranje; fizički zasnovano renderiranje (PBR); teksturne mape; proceduralne teksture; programski alati za teksturiranje; Adobe Substance 3D Designer; igre; filmovi

Glosar

AAA igre – visokobudžetne igre razvijene od strane velikih i poznatih izdavača

Aliasing – pojava u računalnoj grafici gdje zbog pravokutne mrežne strukture piksela na ekranu kose linije izgledaju zupčasto

Antialiasing – radnje koje se poduzimaju za uklanjanje aliasinga

Bitmapa – drugi naziv za rasteriziranu sliku; pravokutna matrica piksela

Izvanmrežni renderer – (eng. *offline renderer*) programski alat koji služi za renderiranje 3D scene; često se radi o visokokvalitetnom renderiranju koje nije u stvarnom vremenu, zbog čega se i zove izvanmrežni renderer; obično se koristi za renderiranje 3D animacija u filmovima

Metode renderiranja – načini generiranja 2D slike na temelju virtualne 3D scene; postoji više metoda renderiranja od kojih su najpoznatije *ray tracing* i rasterizacija trokuta

Motor za razvoj igara – (eng. *game engine*) programski alat u kojem se razvijaju video igre

Poligoni – osnovni elementi 3D modela; bilo koji mnogokuti; najčešće trokuti i četverokuti

Renderiranje – računalni proces generiranja 2D slike na temelju virtualne 3D scene

Subdivizirana površina – površina 3D modela kojoj su osnovni poligoni podijeljeni na više manjih poligona

Teksel – (eng. *texel*; *texture element*) osnovni element teksture, jedan piksel na teksturi

Topologija – distribucija poligona i njihova struktura na 3D modelu

Sadržaj

Sadržaj iv
1. Uvod 1
2. Metode i tehnike rada 2
3. Uvod u područje tekstura i teksturiranja 3
3.1. UV mapiranje 4
3.2. Proceduralne teksture 7
3.2.1. Prednosti proceduralnih tekstura 8
3.2.2. Nedostaci proceduralnih tekstura 8
4. Fizički zasnovano renderiranje (PBR) 10
4.1. Tehničke osnove procesa fizički zasnovanog renderiranja 13
4.1.1. Difuzija i refleksija 14
4.1.2. Translucencija i prozirnost 15
4.1.3. Metali 15
4.1.4. Fresnel 16
4.1.5. Mikropovršina 17
4.1.6. Očuvanje energije 18
5. Vrste teksturnih mapa 20
5.1. *Diffuse* mapa 20
5.2. *Albedo* mapa 20
5.3. *Alpha* mapa 21
5.4. *Normal* mapa 22
5.5. *Height* mapa 23
5.6. *Bump* mapa 24
5.7. *Ambient occlusion* mapa 25
5.8. *Roughness* mapa 25
5.9. *Glossiness* mapa 26
5.10. *Metallic* mapa 27
5.11. *Specular* mapa 28
5.12. Ostale vrste teksturnih mapa i zaključak 28
6. Uobičajeni pristupi teksturiranju 30
6.1. Programski alati za teksturiranje 30
6.2. Tijek rada teksturiranja 32
6.3. Teksturiranje u igrama i teksturiranje u filmova 33
6.4. Korisni savjeti za artiste teksturiranja 35
6.5. Tehnike efikasnog teksturiranja 35
6.5.1. Teksturni atlasi 35
6.5.2. Popločavanje 36

6.5.3. Trake 37
6.5.4. Tehnika ultimativne trake 38
6.5.5. Naljepnice 39
6.5.6. Pakiranje kanala 41
6.5.7. Miješanje tekstura 42
6.6. Koncept trošenja materijala i njegova uloga u vizualnom pripovijedanju 44
7. Primjer izrade tekstura za 3D modele 47
7.1. Substance 3D Designer 47
7.1.1. Softver na bazi čvorova 47
7.1.2. Nelinearan softver 49
7.1.3. Nedestruktivan softver 49
7.1.4. Proces izrade materijala 49
7.1.5. Ugrađeni 3D objekti za testiranje materijala 49
7.1.6. Substance 3D Designer i proceduralni razvoj tekstura 50
7.2. Izrada tekstura u Substance 3D Designeru – praktičan primjer 52
7.2.1. Kratki pregled procesa izrade materijala 53
7.2.2. Izrada *height* mape 53
7.2.3. Izrada *ambient occlusion* mape 56
7.2.4. Izrada *roughness* mape 57
7.2.5. Izrada *metallic* mape 59
7.2.6. Izrada *base color* mape 60
7.2.7. Izrada *normal* mape 64
7.2.8. Kraj izrade materijala 65
8. Zaključak 68
Popis literature 70
Popis slika 74

1. Uvod

Teksturiranje je jedan od najvažnijih elemenata 3D računalne grafike. Ono prodaje uvjerljivost 3D virtualne scene odnosno računalno generirane slike. Teksturiranje obično uvijek ide zajedno s 3D računalnom grafikom i uspjeh mnogih industrija koje se bave istom ovisi upravo o kvaliteti teksturiranja. Neka od područja u kojima je teksturiranje od iznimne važnosti su filmovi, video igre, virtualna stvarnost, 3D skeniranje, 3D printanje, digitalna umjetnost, arhitektura, i komercijalno oglašavanje.

Na samim počecima razvoja 3D računalne grafike, teksturiranje je bilo iznimno teško. Teksture su se izrađivale ručno u programima za 2D obradu slike. Osim toga, ni metode renderiranja nisu davale realistične rezultate, zbog čega su se teksture modela morale neprestano prilagođavati ovisno o izgledu koji se želio postići.

Danas, proces teksturiranja je uvelike olakšan. Postoje 3D aplikacije specijalizirane upravo za teksturiranje pa je proces teksturiranja u njima intuitivan, brz i lagan. Osim toga, danas je vrlo popularan trend fizički zasnovanog renderiranja koji 3D modelima daje visoki stupanj realizma.

Najpoznatija područja 3D računalne grafike su filmovi i igre. To su područja u kojima teksturiranje doseže svoj vrhunac. Dosad su filmovi uvijek bili vizualno ljepši i realističniji od igara. To je zato što su filmovi unaprijed renderirani sadržaj pa za njih nije bitno koliko vremena je bilo potrebno za renderiranje neke 3D scene – gledateljima je serviran gotov rezultat renderiranja. Igre se, s druge strane, izvode u stvarnom vremenu i, s obzirom na hardverska ograničenja računala, jedini način da korisnik postigne besprijekorno iskustvo je da se one optimiziraju. To znači da je u igrama kvaliteta žrtvovana zbog boljih performansi.

Područje 3D računalne grafike se konstantno razvija. Hardver postaje snažniji, otkrivaju se nove metode optimizacije renderiranja, a realizam igara je sve veći. U travnju 2022. tvrtka Epic Games je lansirala svoj najnoviji motor za razvoj igara – Unreal Engine 5. Njegovim dolaskom predstavljene su dvije revolucionarne tehnologije: Lumen i Nanite. Ovim tehnologijama Epic Games je, može se reći, na neki način uklonio jaz između filmova i igara – cilj im je bio postići realizam filmske kvalitete u realnom vremenu.

Moja motivacija za odabir ove teme je ljepota 3D umjetnosti kojoj je granica samo nebo. Iako je 3D umjetnost veliko područje koje zahtjeva mnogo truda i rada te poznavanje raznih faza produkcije (od modeliranja, UV mapiranja i teksturiranja do animiranja, simuliranja i renderiranja), konačni rezultat je oku ugodan i, kao kod svake umjetnosti, nešto što nas čini ponosnima. Odabrani programski alat za teksturiranje u praktičnom primjeru je Adobe Substance 3D Designer, jer je već dug niz godina industrijski standard za izradu tekstura.

2. Metode i tehnike rada

Primarni izvori informacija za izradu teorijskog dijela rada su uglavnom bila recentna akademska istraživanja. Korištena je i najnovija stručna literatura. Sekundarni izvori informacija su bile razne web stranice, članci, objave na blogu, YouTube videozapisi. Za različite programske alate koji se spominju, uz ostalo, bila je korištena njihova službena web stranica i dokumentacija. Većina priloženih slika u teorijskom dijelu rada su autorski rad u programskom alatu, napravljene kao dokaz teorijskih činjenica.

Za izradu praktičnog primjera ovog rada korišten je programski alat Adobe Substance 3D Designer – industrijski standard za izradu tekstura. Za objašnjenja funkcionalnosti softvera korištena je njegova službena dokumentacija. Cijeli praktičan primjer je objašnjen uz priložene snimke zaslona radi lakšeg razumijevanja i čitanja rada.

3. Uvod u područje tekstura i teksturiranja

U stvarnom svijetu, tekstura nekog predmeta se odnosi na neposredan opipljiv podražaj njegove površine. U računalnoj grafici tekstura se odnosi na vizualni dojam nekog objekta sa slike ostvaren jednostavnim vizualnim podražajima kao što su njegova boja, intenzitet i detaljnost (Kumar, 2020, str. 35).

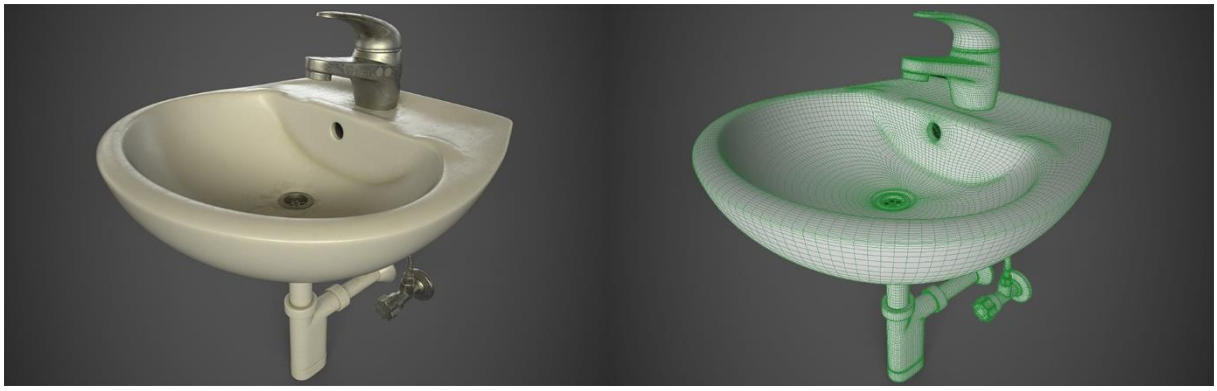
„U računalnoj grafici, tekstura je digitalna reprezentacija izgleda neke površine.“ (Kumar, 2020, str. 43).

Tekstura nekog 3D modela se može opisati kao fizički atributi njegove površine kao što su boja (eng. *color*), kvrgavost (eng. *bumpiness*), hrapavost (eng. *roughness*), reflektivnost (eng. *reflectivity*) i tako dalje (Kumar, 2020, str. 31)**.**

U računalnoj grafici pojam teksture nije strogo definiran. Ponekad artisti se odnose na cjelokupan izgled neke površine, a ponekad se odnosi na samo jedan njen segment odnosno na jednu od mnogih teksturnih mapa. Dakle, tekstura, također poznata kao bitmapa ili teksturna mapa, je dvodimenzionalna slikovna datoteka koja se primjenjuje na površinu 3D modela kako bi on dobio jedno od mogućih vizualnih svojstava površine. U ta svojstva površine ubrajaju se boja, hrapavost, prozirnost, sjaj, reflektivnost, reljefnost, i druge informacije. Dakle, više različitih teksturnih mapa jednom 3D modelu će dati informacije o više različitih površinskih svojstava (Jantunen, 2017, str. 3).

Teksturiranje, također poznato kao mapiranje teksture, grubo rečeno je proces dodjeljivanja „boja“ pikselima na 3D modelu, no iz prijašnjih paragrafa moguće je zaključiti da je to je ipak više od dodjeljivanja samo boja (Kumar, 2020, str. 39).

„Teksturiranje je jeftin način generiranja detalja na 3D modelima bez potrebe za povećanjem njegove osnovne geometrije.“ (R. Moreira & S. Lages, 2014, str. 1).



Slika 1: Teksturirani prikaz umivaonika (lijevo), prikaz umivaonika bez tekstura (desno) (autorski rad)

U procesu teksturiranja često se koriste pojmovi *tekstura*, *materijal* i *shader*, i često se ne razumije razlika između tih pojmova.

U stvarnom svijetu, materijal se obično definira kao materija od koje je neka stvar izrađena. U motorima za razvoj igara (eng. *game engines*) i izvanmrežnim rendererima (eng. *offline renderers*), materijal je sredstvo koje se primjenjuje na 3D objekt kako bi se kontrolirala njegova vizualna svojstva (Ayyildiz, 2020, str. 25–26).

Dakle, u 3D softverima materijal je cjelokupan izgled površine 3D modela, s njegovom bojom, hrapavošću, kvrgavošću i ostalim atributima, odnosno to je skup različitih teksturnih mapa (tekstura).

Shader je program (niz instrukcija) u grafičkoj kartici koji definira način na koji se izračunavaju svjetlina i boja piksela 3D površine projicirane na 2D ekran. Većina motora za razvoj igara i izvanmrežnih renderera dopuštaju određeni stupanj podešavanja svojih *shadera* (Souza & Mota, 2021, str. 2).

Pojam materijala se često pogrešno koristi naizmjenično sa *shaderom*. Materijal je konačni rezultat *shadera* nakon što je postavljen, zajedno s njegovim parametrima i dodijeljenim mu teksturama. Dva potpuno različita materijala mogu se napraviti koristeći isti *shader* kad se parametri i/ili teksture *shadera* promijene (Souza & Mota, 2021, str. 2).

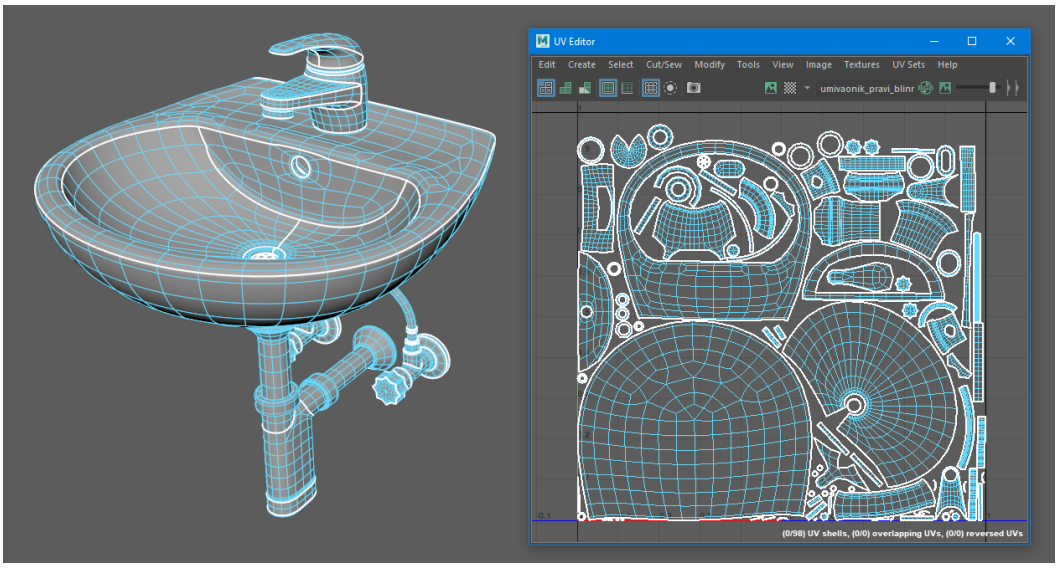
Teksturne mape mogu se izraditi u bilo kojem softveru za obradu slike, no postoje posebni 3D softveri namijenjeni upravo izradi tekstura koji daju mogućnost bojanja teksturnih mapa izravno po 3D modelu, ali i mogućnost automatskog generiranja istih (uz pomoć algoritama) (Jantunen, 2017, str. 3).

Bez obzira na način izrade i korišteni programski softver, teksturne mape se izrađuju na temelju UV mape 3D modela. Sljedeće poglavlje govori o UV mapiranju.

3.1. UV mapiranje

U 3D računalnoj grafici kod teksturiranja se pojavljuje jedan problem – primijeniti teksturu koja je dvodimenzionalna na model koji je trodimenzionalan. Kako bi se 2D tekstura mogla mapirati na površinu 3D modela, prvo 3D model treba „odmotati“ i njegovu odmotanu površinu staviti na 2D ravninu. Taj proces odmotavanja (eng. *unwrapping*) odnosno mapiranja površine 3D modela u 2D prostor se zove UV mapiranje (Chopine, 2011, str. 152).

Rezultat UV mapiranja je UV mapa. UV mapa je površina odnosno poligoni 3D modela mapirani u 2D prostor – UV prostor. To je 2D reprezentacija površine 3D modela (Jantunen, 2017, str. 3).



Slika 2: Prikaz umivaonika (lijevo) i njegova UV mapa (desno) (autorski rad)

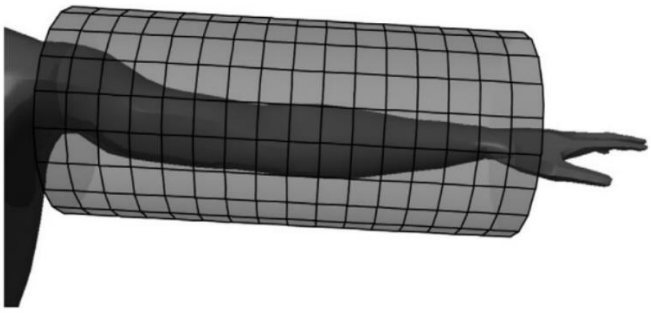
Zove se UV mapa zbog U i V koordinata 2D prostora u kojem se prikazuje. Budući da se za 3D prostor koriste koordinate X, Y i Z, dogovor je bio da se za 2D prostor uzmu koordinate U i V. U predstavlja horizontalnu os, a V vertikalnu os dvodimenzionalnog UV prostora. Svaka UV točka površine 3D modela ima odgovarajuću točku tog modela u 3D prostoru zvanu vrh (eng. *vertex*) (Kumar, 2020, str. 35).

Najjednostavniji način kreiranja UV mape je projekcijom. Osnovne projekcije su planarna, cilindrična i sferična. Planarna projekcija projicira UV mapu na objekt preko ravnine i najprikladnija je za ravne površine. Cilindrična projekcija na objekt projicira UV mapu savijenu u obliku cilindra (valjka). Sferična projekcija na objekt projicira UV mapu savijenu u obliku sfere (kugle) (Chopine, 2011, str. 152–153).

Slika 3: Planarna, cilindrična, i sferična projekcija (Chopine, 2011, str. 152)

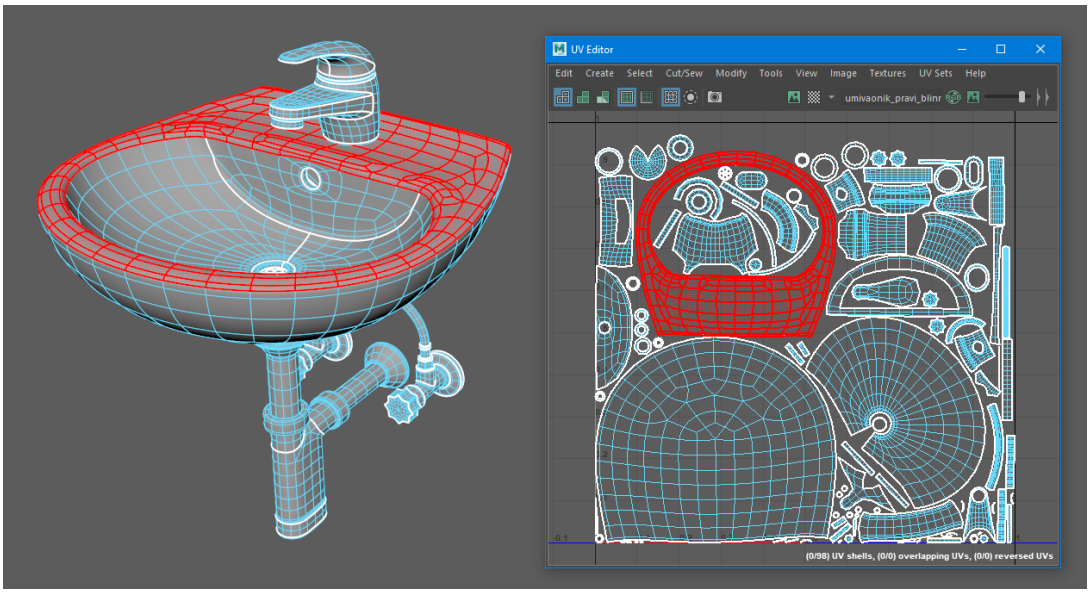
Spomenute projekcije obično nisu dovoljne za kreiranje dobre UV mape svakog objekta, jer nisu svi objekti jednostavna geometrijska tijela. Međutim, svi 3D modeli se mogu

logički podijeliti na jednostavne dijelove pa se različite projekcije mogu koristiti za različite dijelove. Na primjer, cilindrična projekcija je dobra za ruku čovjeka (Chopine, 2011, str. 155).



Slika 4: Cilindrično mapiranje ruke (Chopine, 2011, str. 156)

Svi dijelovi modela za koje su zasebno napravljene različite projekcije su odvojeni u UV mapi i zovu se UV školjke (eng. *UV shells*). Svaka UV školjka se može dodatno odrezati tako da od jedne bude više odvojenih školjaka preko alata za rezanje UV školjaka unutar 3D aplikacije. Svaka UV školjka ima šavove (eng. *seams*). Šav je vanjski rub UV školjke koji je u UV mapi odvojen na dva dijela (jedan dio ruba je u jednoj UV školjci, drugi dio ruba je u drugoj UV školjci), ali u stvarnosti na modelu je spojen i čini jedan rub. Raspored UV školjaka u UV prostoru odnosno u UV mapi se zove UV raspored (Dunlop, 2014, str. 52).



Slika 5: UV školjka (desno) i odgovarajući dio na prikazu umivaonika (lijevo) (autorski rad)

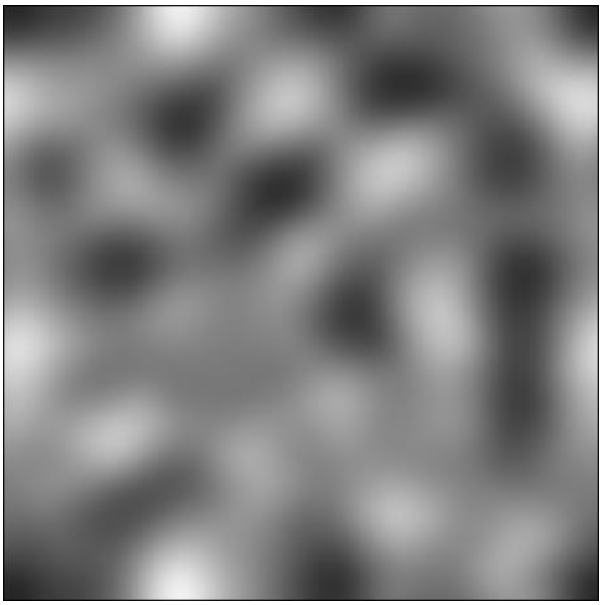
Prva stvar koja se obično uzima u obzir pri UV mapiranju 3D modela je da se najučinkovitije iskoristi UV prostor, tako da tekstura ima što manju količinu izgubljenih piksela odnosno što veću gustoću rezolucije, također poznatu kao gustoća teksela (eng. *texel density*), gustoća piksela (eng. *pixel density*) ili gustoća teksture (eng. *texture density*) (Souza & Mota, 2021, str. 3).

Značenje učinkovitog korištenja UV prostora mijenja se ovisno o 3D modelu i potrebama igre – ponekad je bolje dati prednost važnijim dijelovima modela odnosno dati im veću gustoću teksela u odnosu na ostale dijelove nego imati jednaku gustoću teksela za sve dijelove modela. Na primjer, cipelama lika često nije potrebna tolika rezolucija kao i za njegovo lice, pogotovo ako u 3D animaciji njegove cipele neće nikad biti prikazane izbliza (Souza & Mota, 2021, str. 3).

3.2. Proceduralne teksture

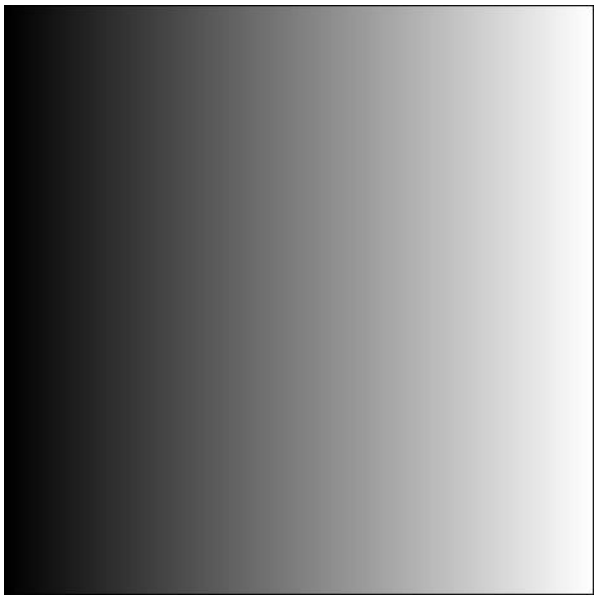
„Proceduralno teksturiranje je proces stvaranja teksture koja je generirana parametarski korištenjem algoritama.“ (Kumar, 2020, str. 43).

„Proceduralni sadržaj je svaki sadržaj koji je generiralo računalo pomoću algoritma.“ (Busquets Duran, 2014, str. 8).



Slika 6: Perlinov šum – primjer jedne proceduralne teksture (autorski rad)

Algoritmi koji generiraju proceduralne teksture mogu se zamisliti kao matematičke funkcije ili formule. Jedan jednostavan primjer proceduralne teksture je horizontalni gradijent u sivim tonovima kojeg je moguće generirati matematičkom funkcijom $f(x,y) = x$, gdje su parovi (x,y) koordinate piksela na teksturnoj mapi. Iz ove funkcije se može zaključiti da kako koordinate piksela rastu po x osi, imat će veću vrijednost. Ako se za manje x vrijednosti uzme da daju tamnije tonove sive odnosno za veće x vrijednosti da daju svjetlije tonove sive, ova jednostavna matematička funkcija primijenjena na svaki piksel teksturne mape dat će tamnije vrijednosti pikselima na lijevoj strani, a svjetlije na desnoj strani teksturne mape (Busquets Duran, 2014, str. 17).



Slika 7: Horizontalni gradijent u sivim tonovima - primjer jedne jednostavne proceduralne teksture (autorski rad)

3.2.1. Prednosti proceduralnih tekstura

Prednosti proceduralne teksture u odnosu na sliku kao teksturu su sljedeće:

- Proceduralna tekstura je izuzetno kompaktna. Veličina proceduralne teksture obično se mjeri u kilobajtima, dok se veličina slike kao teksture obično mjeri u megabajtima – algoritmi su obično tisuće puta manji od slika koje proizvode.
- Proceduralna tekstura nema fiksnu rezoluciju. Pruža beskonačnu razinu detalja bez obzira koliko blizu se gleda. Može se reći da djeluje kao vektorska grafika.
- Proceduralna tekstura ne pokriva fiksnu površinu. Drugim riječima, ima neograničen domet i može pokriti proizvoljno veliku površinu.
- Proceduralna tekstura se može parametrizirati, tako da se mijenjanjem parametara može generirati više varijacija umjesto da bude ograničena na jednu fiksnu teksturu (Ebert, 2003, str. 14; Lasram, 2012, str. 8; Pai, 2019, str. 2–3).

3.2.2. Nedostaci proceduralnih tekstura

Nedostaci proceduralne teksture u odnosu na sliku kao teksturu su sljedeći:

- Proceduralne teksture je teško kontrolirati. Često se ne može dobiti konkretno željeni efekt i na konkretno željenom mjestu, već se uglavnom može dobiti samo otprilike traženi izgled teksture. Često je lakše predvidjeti ishod teksture kod skeniranja ili ručnog bojanja teksture.

- Čekati algoritme da generiraju proceduralnu teksturu može biti duže od čekanja da se učita pohranjena slika kao tekstura. Ovo je klasični kompromis između vremena potrebnog da se generira i prostora potrebnog da se pohrani proceduralna tekstura.
- *Aliasing* je čest problem u proceduralnim teksturama. Postići dobar *antialiasing* može biti vrlo nezgodno. To nije slučaj kod rasteriziranih slika kao tekstura (Ebert, 2003, str. 15; Kumar, 2020, str. 43; *Procedural Texturing*, bez dat.).

4. Fizički zasnovano renderiranje (PBR)

Izgled materijala proizlazi iz složene interakcije svjetlosti, svojstava materijala i geometrijskog oblika objekta. U računalnoj grafici razvijeni su različiti modeli koji opisuju te korelacije. Moderne prakse renderiranja se sve više prilagođavaju filozofiji fizički zasnovanog renderiranja (*eng. Physically based rendering*; PBR) (Bittner, 2021, str. 25).

U mnogim današnjim digitalnim produkcijama fizički zasnovano renderiranje (PBR) igra glavnu ulogu u radu 3D artista. Glavna ideja PBR-a je prikazati virtualni svijet na temelju zakona fizike. Sve više je poželjno ponašanje virtualnog svijeta na temelju zakona fizike, jer ono ga čini realističnijim odnosno bližim stvarnom svijetu. Jedno od područja gdje se koristi su simulacija kretanja tekućine i tkanine, ali posebno renderiranje odnosno sjenčanje materijala (Bittner, 2021, str. 25).

Iako je PBR najpoznatiji po stvaranju fotorealističnih izgleda, mogu se postići i različiti nefotorealistični izgledi (Bittner, 2021, str. 25).

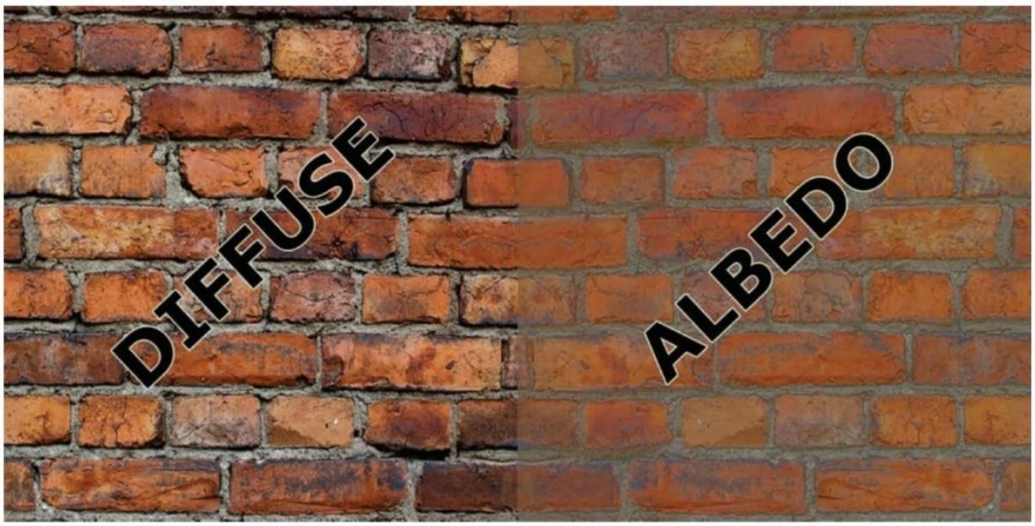
Tijek rada fizički zasnovanog renderiranja igra veliku ulogu u modernoj grafici, posebno u igrama gdje ga je većina današnjih motora za razvoj igara usvojila. S PBR pristupom mogu se stvoriti besprijekorni, točniji i visoko realistični rezultati. (Kumar, 2020, str. 27).

Prije PBR-a, teško je bilo postići rezultate koji bi izgledali donekle realistično. Pogotovo je bilo teško to postići s jednim algoritmom za renderiranje. Zato su postojali različiti algoritmi za renderiranje koji su se primjenjivali na različita okruženja i uvjete osvjetljenja po potrebi, ovisno o tome s kojim algoritmom bi se postigao bolji rezultat u danim uvjetima. Dolaskom PBR-a, složenost algoritama za renderiranje se minimizirala. Bit PBR-a je primjena zakona fizike za izračunavanje svjetlosti i njene interakcije s materijalima, jer se izgled materijala u stvarnom svijetu mijenja s uvjetima svjetlosti i fizike. Dakle, PBR uzima u obzir fizičko ponašanje svjetlosti za renderiranje materijala i na taj način postiže njihov prirodan izgled (Kumar, 2020, str. 27).

Prije PBR-a, iste materijale u različitim scenama i/ili osvjetljenju 3D artisti su morali prilagođavati kako bi materijali dobro izgledali u drugom okruženju. To je bilo tako zato što se generirana slika nije renderirala na temelju zakona fizike, pa su se artisti morali koristiti raznim trikovima i varkama da bi postigli rezultate koji dobro izgledaju. Tako su npr. ranije tehnike renderiranja koristile *diffuse* teksturne mape koje su u sebi osim boje površine sadržavale informacije o sjenama (eng. *shadows*) i svjetlu (eng. *highlights*) (Kelly, 2022, str. 7).

Na slici 8 moguće je vidjeti dvije vrste teksturnih mapa. Lijevo je *diffuse* teksturna mapa koja se koristila u starijim tehnikama renderiranja, sadrži informacije o sjenama i svjetlu.

Zasjenjene dijelove je moguće vidjeti u šupljinama, a od sunca osvijetljene dijelove na ciglama. Takva tekstura otežava promjenu rasvjete, a da pritom materijal ostane dosljedan. Na desnoj strani je *albedo* teksturna mapa koja sadrži samo informacije o boji površine. To daje mogućnost promjene rasvjete unutar 3D softvera, čime se pomoću PBR-a u stvarnom vremenu mijenja smjer zasjenjenih i osvijetljenih dijelova na površini teksture (Kelly, 2022, str. 7, 8).



Slika 8: *Diffuse* mapa ima informacije o sjenama i svjetlu, *albedo* mapa ima samo boju (Kelly, 2022, str. 7)

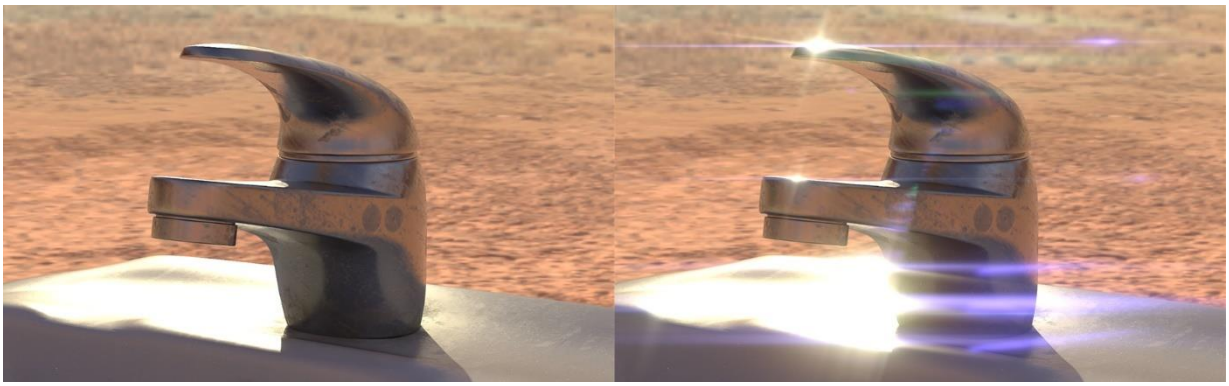
PBR nam omogućuje dosljedan i realističan izgled u svim okruženjima i u različitim uvjetima osvjetljenja koristeći matematičke izračune. Iako će neka okruženja PBR materijalu dati drugačiji izgled, on će uvijek ostati dosljedan zbog primjene zakona fizike. Korištenjem PBR-a eliminirana je potreba za teksturnim mapama koje sadrže informacije o zasjenjenim i/ili osvijetljenim dijelovima. To znači da se može u stvarnom vremenu mijenjati rasvjeta u 3D okruženju i da nema potrebe teksturne mape prilagođavati istoj (Kelly, 2022, str. 15–16).

Na slici 9 moguće je vidjeti kako različito okruženje pridonosi različitoj estetici 3D modela umivaonika iako se nijedna od teksturnih mapa modela nije promijenila. To je zato što se u PBR-u izračunava interakcija svjetlosti i modela kako bi se proizveo prirodniji i realističniji izgled kao što bi to bilo u stvarnom svijetu.

Slika 9: 3D model umivaonika u različitim uvjetima rasvjete korištenjem HDRI (autorski rad)

Okruženje kao svjetlo (eng. *environment light*) popularna je tehnika rasvjete kod fizički zasnovanog renderiranja, poznata kao rasvjeta na temelju slike (eng. *image-based lighting*). Slika visokog dinamičkog raspona (eng. *high dynamic range image*; HDRI) je tekstura obrađena iz jedne ili više spojenih slika stvarnog okruženja. HDRI se projicira na beskonačno udaljenu sferu kako bi simulirala svjetlost iz stvarnog svijeta kojeg predstavlja. Brzo i jednostavno se postavlja, a pruža vrlo detaljno osvjetljenje, sjene, refleksije i refrakcije – sve je to važno pri ocjenjivanju kvalitete materijala. Dobra postava rasvjete pri izradi materijala osigurava da će se oni ponašati dosljedno i ispravno reagirati u drugačijem scenariju rasvjete (Parsons, 2017, str. 13).

Renderiranje u 3D računalnoj grafici obično znači generiranje slike na temelju nekog modela virtualne kamere koja „gleda“ u 3D scenu. U računalnoj grafici obično se izrađuje prilično jednostavan model kamere, budući da većina 3D aplikacija su takve da prikazuju kako nešto *doista* izgleda nego kako izgleda konkretna *fotografija* istoga fotografirana sa stvarnom kamerom. To je idealizirana kamera kakvu bismo željeli imati u stvarnom životu. Međutim, upravo taj nedostatak defekata virtualne kamere može sam učiniti da generirane slike izgledaju nerealno, jer su previše savršene, a ljudi su navikli na efekte nesavršenosti stvarnih kamera/fotoaparata koji se pojavljuju na slici (Shirley i ostali, 2012, str. 2).



Slika 10: Desna renderirana slika slavine umivaonika izgleda realističnije zbog efekta bljeska u objektivu kamere (autorski rad)

Nadalje, istraživanje koje je provela Bittner (2021) je pokazalo da postoji nedosljednost među različitim alatima za renderiranje (rendererima) kad je u pitanju apstraktno bilježenje svojstava materijala preko modela. Svaki renderer ima svoju realizaciju i vjerojatno će ostati izazov osigurati besprijekornu razmjenu između njih. To je posebno problem digitalnim produkcijama koje u različitim odjelima rade u različitim alatima.

Istraživanje je pokazalo dva glavna problema. Jedan je da ne postoji univerzalni model materijala - svaki renderer ima svoje modele materijala s različitim svojstvima. Drugi problem je da različiti rendereri čak i za iste metode renderiranja imaju svoje implementacije istih koje su malo drugačije od drugih renderera. Sve to skupa rezultira na način će isti materijal, pod istom rasvjetom, položajem kamere i ostalim uvjetima, u različitim rendererima izgledati drugačije.

Ipak, iste metode/mehanizmi renderiranja ispitanih renderera daju dosta slične rezultate, za razliku od dviju različitih metoda renderiranja. Zato postoji mogućnost da će u budućnosti postojati jedan standardizirani model materijala sa svim potrebnim svojstvima zbog lakše razmjene između različitih renderera. Naravno, poželjno je da postoje i modeli materijala specifični za dani renderer, koji će iskoristiti pojedine snage tog renderera, no takvi modeli materijala se onda mogu koristiti u produkcijama koje koriste taj jedan i isti renderer u svim odjelima (Bittner, 2021).

4.1. Tehničke osnove procesa fizički zasnovanog renderiranja

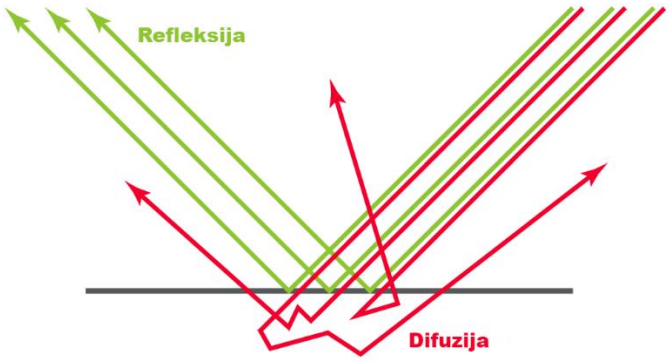
U ovom potpoglavlju je objašnjena tehnička strana PBR procesa koja je važna za bolje razumijevanje različitih vrsta teksturnih mapa o kojima se govori u sljedećem poglavlju.

4.1.1. Difuzija i refleksija

Difuzija i refleksija su dva pojma koji opisuju najosnovnije interakcije između svjetla i površine. Difuzija je još poznata kao difuzno svjetlo (eng. *diffuse light*), a refleksija kao zrcalno svjetlo (eng. *specular light*).

Ovisno o materijalu, zrake svjetlosti će se drugačije ponašati pri interakciji s površinom. Kada svjetlost udari u glatku površinu, svjetlost će se reflektirati (odbiti) i otići u smjeru suprotnom u odnosu na normalu površine. To je slučaj refleksije svjetlosti i rezultirat će izglednom poput zrcala. Riječ *specular*, koja se često koristi za opis ovog efekta, potječe od latinskog za zrcalo.

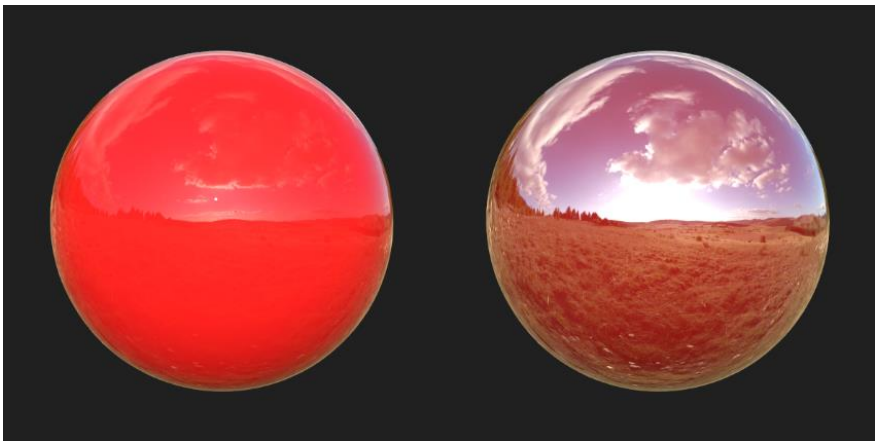
Međutim, na nekim drugim materijalima neće se sva svjetlost reflektirati od površinu. Obično će neke zrake prodrijeti u unutrašnjost osvijetljenog objekta. Tamo će ih materijal ili apsorbirati (pretvoriti u toplinu) ili će se zrake raspršiti iznutra. Dio ove raspršene svjetlosti može naći izlaz iz površine i ponovno postati vidljiv našim očima odnosno kameri. Takva svjetlost je poznata pod nazivima *difuzija*, *difuzno svjetlo*, i *potpovršinsko raspršivanje* (eng. *subsurface scattering*).



Slika 11: Refleksija i difuzija (Kelly, 2022, str. 8)

Apsorpcija i raspršenje difuzne svjetlosti često su prilično različiti za različite valne duljine svjetlosti, a to je ono što objektima daje njihovu boju – npr. ako objekt apsorbira većinu svjetlosti, a raspršuje crvenu, izgledat će crveno.

Difuzna (raspršena) svjetlost je takva da izgleda poprilično jednako bez obzira na kut gledanja, dok reflektirana svjetlost (npr. zrcalo) ovisi o kutu gledanja odnosno položaju kamere. Upravo zbog te karakteristike difuzne svjetlosti, u računalnoj grafici za prikaz difuzne svjetlosti *shaderi* koriste teksturnu mapu koja predstavlja boju površine. Ona definira koje boje će biti difuzna svjetlost ovisno o dijelu površine iz koje se raspršuje. Ta teksturna mapa se uglavnom zove *albedo* ili *diffuse color* (Kelly, 2022, str. 8–9; Ratković, 2017, str. 6; Wolfram, bez dat.).



Slika 12: Lijeva sfera svjetlost više raspršuje, desna sfera ju više reflektira (autorski rad)

4.1.2. Translucencija i prozirnost

U materijalima koji imaju veće udaljenosti raspršivanja, difuzija svjetlosti nije jednostavna. Najbolji primjeri toga su koža i vosak. U tim slučajevima, jednostavna boja obično neće biti dovoljna za ispravan rezultat – sustav sjenčanja mora uzeti u obzir oblik i debljinu predmeta koji je osvjetljen.

Ako su objekti dovoljno tanki, na njima se često može vidjeti raspršivanje svjetlosti sa stražnje strane. To je slučaj translucencije (eng. *translucency*), a takvi objekti se zovu translucentni (eng. *translucent*) objekti. Dobar primjer jednog takvog objekta je list stabla.

Ako je difuzija još slabija, tada se gotovo uopće ne vidi raspršivanje i cijela svjetlost može proći kroz predmet s jedne strane na drugu. To je slučaj prozirnosti (eng. *transparency*), a takvi objekti se zovu prozirni (eng. *transparent*) objekti. Primjer jednog takvog objekta je staklo.

Ova ponašanja su različita od običnog difuznog raspršivanja i zahtijevaju drugačiji opis *shadera* (Adobe, bez dat.-c; Kelly, 2022, str. 9; McDermott, 2018, str. 19–20).

4.1.3. Metali

Metali (električno vodljivi materijali; vodiči) su posebno važni. Osim što su uglavnom više reflektirajući (60 – 90 %) od izolatora/nevodiča (0 – 20 %), uzrokuju bojanje refleksije.

Refleksija na metalima ponekad varira u vidljivom spektru, što znači da njihove refleksije izgledaju obojenima. Neki od primjera gdje je moguće vidjeti obojene refleksije su zlato, bakar, mjed. Izolatori u pravilu ne pokazuju taj efekt i njihove refleksije su neobojene.

Upravo ova podjela materijala na metale i izolatore je dovela do toga da neki sustavi renderiranja usvoje metalnost kao teksturnu mapu. U takvim sustavima artisti određuju stupanj

do kojeg se materijal ponaša kao metal, umjesto podešavanja refleksije i njene boje. Ovaj način rada preferira većina artista, jer pruža jednostavniju izradu materijala, ali nije nužno karakteristika fizički zasnovanog renderiranja (Kelly, 2022, str. 10; McDermott, 2018, str. 33–37).

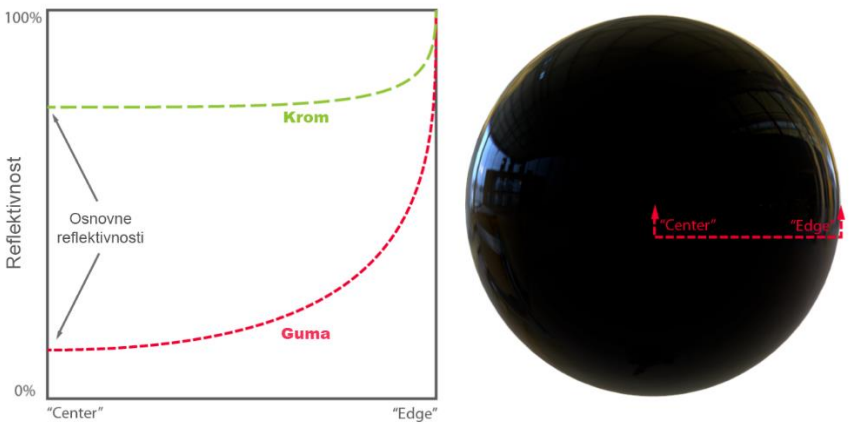
Slika 13: Primjer obojene refleksije metala (autorski rad)

4.1.4. Fresnel

Termin Fresnel dolazi od Augustin-Jean Fresnela, francuskog fizičara koji je prvi precizno opisao istoimeni efekt. Vrlo je popularan termin u PBR tijeku rada kad se govori o refleksiji. U računalnoj grafici termin Fresnel se odnosi na variranje refleksije s obzirom na kut gledanja. Konkretno, veća je vjerojatnost da će se reflektirati svjetlost koja padne na površinu pod tangentnim kutom nego što će ona koja padne okomito na površinu. To znači da će objekti s Fresnelovim efektom imati veću refleksiju blizu rubova.

Svim materijalima reflektivnost postaje potpuna (100 %) pod tangentnim kutovima – rubovi gledani na bilo kojem glatkom objektu djeluju kao savršena (bezbojna) zrcala, bez obzira na materijal. Fresnel efekt brzo postaje manje očit kako površine prestaju biti glatke.

U PBR tijeku rada artist definira osnovnu reflektivnost – minimalnu količinu reflektirane svjetlosti. Fresnel efekt će nadodati reflektivnost povrh vrijednosti koju je definirao artist, dosežući do 100 % pod tangentnim kutovima. Drugim riječima, artist definira bazu, a povrh nje djeluju Fresnelove jednadžbe, čineći površinu više reflektirajućom pod tangentnim kutovima (Adobe, bez dat.-c; Kelly, 2022, str. 10–12).

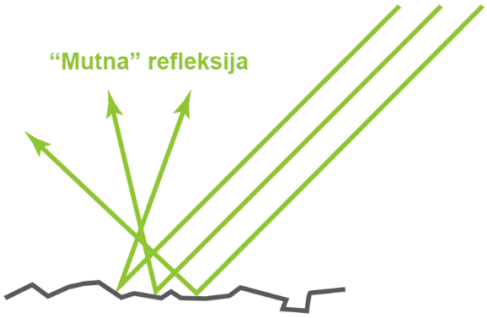


Slika 14: Fresnel efekt (Kelly, 2022, str. 11)

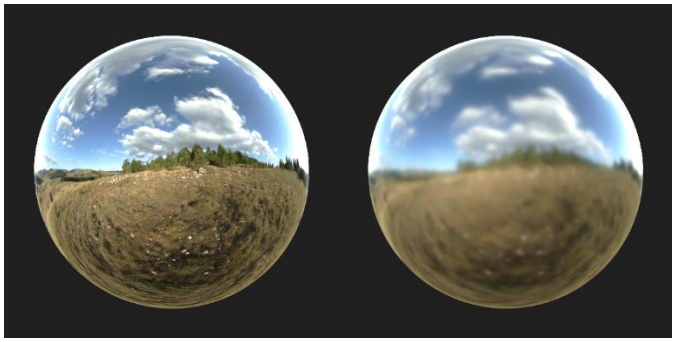
4.1.5. Mikropovršina

Većina površina u stvarnom svijetu ima vrlo male nesavršenosti: sitne žljebove, pukotine, i kvržice, premale da bi ih oko moglo vidjeti. Unatoč tome što su nevidljive golom oku, ove mikroskopske osobine utječu na difuziju i refleksiju svjetlosti. Najuočljiviji utjecaj imaju na refleksiju svjetlosti.

Paralelne zrake svjetla počinju se razilaziti kada se reflektiraju od grublje površine, jer svaka zraka pogađa dio površine s drugačijom orijentacijom. Ukratko, što je površina grublja/hrapavija, to će se reflektirana svjetlost više razilaziti odnosno izgledati mutno.



Slika 15: Utjecaj mikropovršine na refleksiju svjetlosti (Kelly, 2022, str. 12)



Slika 16: Desna sfera ima hrapaviju površinu pa refleksija izgleda mutnije (autorski rad)

Nažalost, evaluacija svake individualne mikropovršinske osobine za potrebe sjenčanja bila bi preskupa u smislu korištenja memorije i računskih operacija. Umjesto toga može se odrediti opća mjera hrapavosti površine i napisati *shaderi* koji daju slične rezultate, i to sve bez dodatnih troškova. Ova mjera se naziva sjaj (eng. *gloss*), glatkost (eng. *smoothness*) ili hrapavost (eng. *roughness*). Može se opisati preko teksturne mape ili odrediti kao konstanta za dani materijal. U slučaju teksturne mape, mjera hrapavosti duž površine varira ovisno o vrijednostima teksturne mape. U slučaju konstante, cijeli materijal duž čitave površine ima jednaku mjeru hrapavosti.

Mikropovršinske osobine imaju i druge suptilne učinke na refleksiju. Na primjer, Fresnel efekt donekle se smanjuje s grubljim površinama, jer se raspršuje i sprječava promatrača da ga jasno vidi.

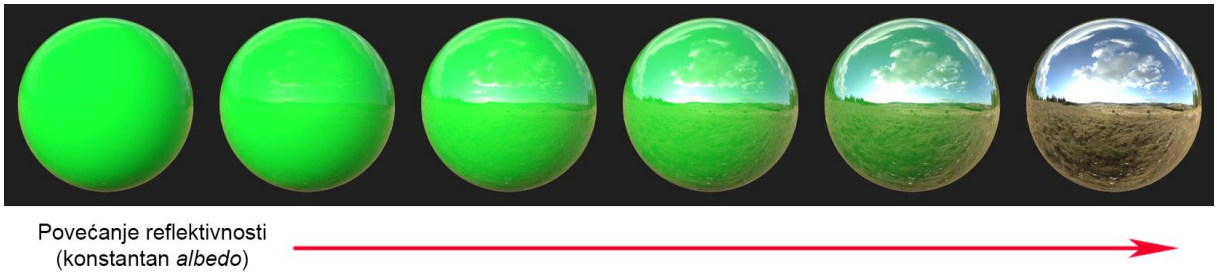
Stvarni svijet je pun širokog spektra mikropovršinskih detalja, te su oni važna karakteristika svakog materijala. Mapiranje hrapavosti/sjaja igra ključnu ulogu u fizički zasnovanom renderiranju budući da mikropovršinski detalji znatno utječu na refleksiju svjetlosti (Kelly, 2022, str. 12–14; Ratković, 2017, str. 5; Wolfram, bez dat.).

4.1.6. Očuvanje energije

Iz prethodnih potpoglavlja može se zaključiti da se refleksija i difuzija međusobno isključuju – da bi se svjetlost reflektirala, ne smije prodrijeti u površinu objekta odnosno raspršiti se, i obrnuto. To je u jeziku sjenčanja poznato kao očuvanje energije (eng. *energy conservation*) i lako se realizira – ako se svjetlost većinom raspršuje, smanji joj se refleksija, i obrnuto.

To znači da će visoko reflektirajući objekti pokazivati malo do nimalo difuzne svjetlosti, jer malo do nimalo svjetlosti prodire u njihovu površinu, dok se gotova sva svjetlost reflektira. Vrijedi i obrnuto – ako predmet ima jaku difuziju, ne može biti posebno reflektirajući.

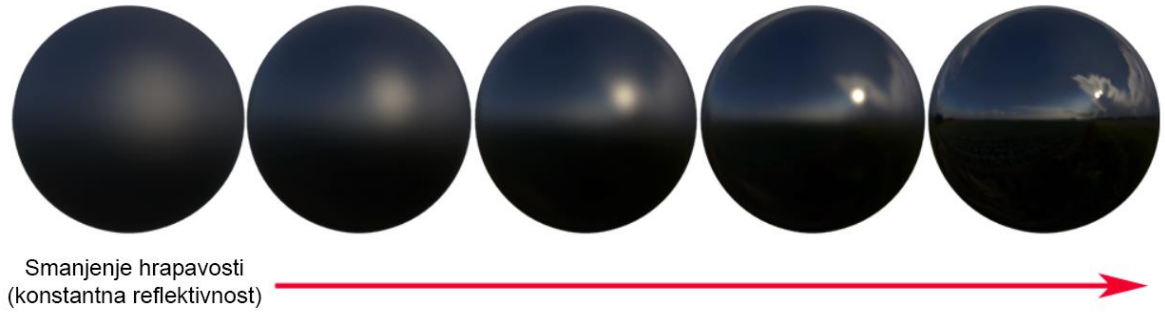
Ovakvo očuvanje energije važan je aspekt fizički zasnovanog sjenčanja/renderiranja, jer sprječava materijale da „krše“ zakone fizike i da postanu nedosljedni pod različitim svjetlosnim uvjetima (Kelly, 2022, str. 9–10; McDermott, 2018, str. 30).



Slika 17: Utjecaj reflektivnosti objekta na refleksiju i difuziju svjetlosti (autorski rad)

Uz opisani balans difuzije i refleksije, potrebno je spomenuti još jedan oblik očuvanja energije koji se pojavljuje, a koji je također važan za fizički zasnovano renderiranje.

U PBR tijeku rada hrapave površine su prikazane s većim točkama refleksije, koje izgledaju prigušenije od onih manjih, oštrijih točaka glatke površine. Obe površine reflektiraju istu količinu svjetlosti, ali grublja (hrapavija) površina je raspršuje u različitim smjerovima, dok glađa površina reflektira koncentriraniji snop. Ovo je još jedan vrlo važan oblik očuvanja energije bez kojeg renderiranje ne bi bilo fizički zasnovano (Kelly, 2022, str. 13–14; LearnOpenGL, bez dat.).



Slika 18: Utjecaj hrapavosti objekta na refleksiju svjetlosti (Kelly, 2022, str. 13)

5. Vrste teksturnih mapa

Postoje mnoge vrste teksturnih mapa ovisno zahtjevima 3D-a. Vrste teksturnih mapa također ovise o korištenom rendereru, motoru za razvoj igara ili tijeku rada u studiju za 3D grafiku (Kumar, 2020, str. 35).

Svaka teksturna mapa je ili RGB mapa ili mapa u sivim tonovima (eng. *grayscale*). RGB mapa znači da je u boji. Mapa u sivim tonovima znači da je to mapa koja ima samo različite nijanse sive boje.

5.1. *Diffuse* mapa

Diffuse je RGB teksturna mapa, starija vrsta mape koja se koristila ranije u 3D računalnoj grafici. Bila je najčešće korištena vrsta teksturne mape, a predstavljala je opću boju 3D modela. Osim što sadrži informacije o boji modela, također sadrži informacije o sjenama i svjetlu, što je i razlog zbog kojeg se danas rijetko koristi – utisnute informacije o sjenama i osvjetljenju otežavaju/onemogućuju promjenu rasvjete u 3D softveru u stvarnom vremenu (Busquets Duran, 2014, str. 11).

Ranije tehnike renderiranja (ponajviše u igrama) nisu davale visoko realistične rezultate (posebno sjene), pa je najbolji način da se donekle postigne realizam bio pohraniti unaprijed dotjerane sjene u teksturu modela.

Diffuse mapa je obična bitmapa, snimljena fotoaparatom, obojena u 2D ili 3D programu, ili skenirana (Kumar, 2020, str. 36).

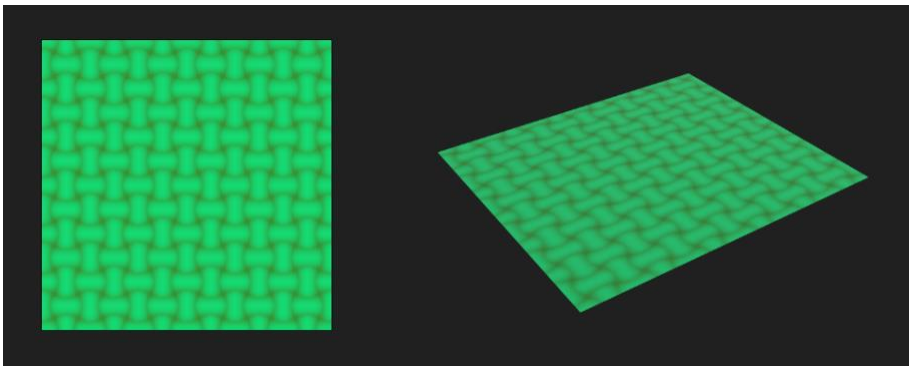
5.2. *Albedo* mapa

Albedo je RGB teksturna mapa, također poznata kao *base color* mapa, slična je *diffuse* mapi. Jedina razlika je što u sebi ne sadrži informacije o sjenama (eng. *shadows*) i svjetlu (eng. *highlights*). Dakle, to je teksturna mapa koja definira boju materijala odnosno površine, ona predstavlja difuznu svjetlost odnosno onu koja je do kamere došla raspršivanjem s površine 3D modela.

Za *albedo* mapu se može reći da je kralježnica materijala, jer s njom materijal izgleda potpun i dovršen.

Iako je ovo teksturna mapa koja, u srži, ne pohranjuje informacije o sjenama i svjetlu, kako bi materijal izgledao dodatno detaljnije većina materijalnih artista dodaje vrlo mali postotak informacija o zasjenjenim (oko 0,5 %) i osvijetljenim (oko 20 %) dijelovima.

Albedo (*Base color*) je često korištena teksturna mapa u PBR tijeku rada. Iz razloga što sadrži samo informacije o boji površine (bez sjena i svjetla), daje mogućnost scenarija dinamičnog osvjetljenja koji dinamički stvara sjene na površini (Kelly, 2022, str. 27; Kumar, 2020, str. 36; Sanchis Albert, 2019, str. 21).



Slika 19: Primjer *albedo* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad)

5.3. *Alpha* mapa

Alpha, poznata još kao *opacity* i *transparency*, teksturna je mapa u svim tonovima (eng. *grayscale*), a predstavlja prozirnost odnosno neprozirnost površine.

Obično bijeli dijelovi mape predstavljaju potpunu neprozirnost, a crni potpunu prozirnost. Svi ostali tonovi sive boje predstavljaju određen stupanj prozirnosti odnosno neprozirnosti (Jantunen, 2017, str. 4; Kumar, 2020, str. 36).

Slika 20: Primjer *alpha* mape (lijevo) i njena manifestacija na površini (desno) – bijeli dijelovi otkrivaju *albedo* mapu iz prethodnog primjera (autorski rad)

Alpha mapa se često koristi u igrama za teksturu lišća na stablima gdje, umjesto modeliranja geometrije lista, korištenjem ove mape je moguće teksturu lista staviti na običan četvrtasti poligon, prikazati samo dio teksture s listom, a ostatak teksture sakriti (Chopine, 2011, str. 24).

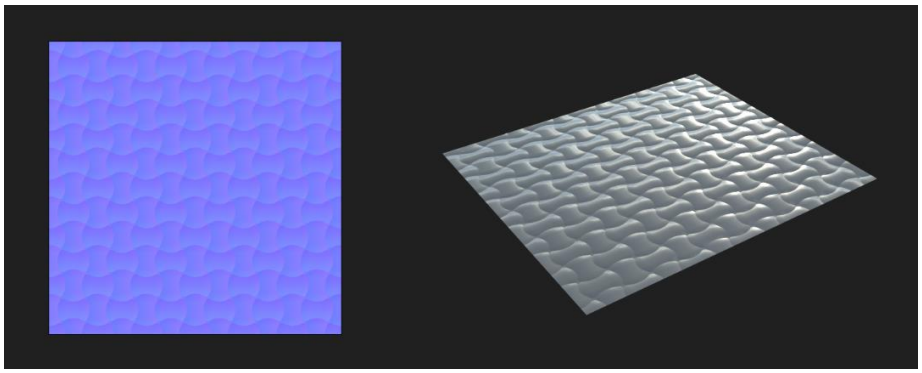
Ova mapa se u teksturiranju često koristi u ulozi maske. Maska, slično gornjim objašnjenjima, ograničava područje djelovanja druge teksturne mape ili materijala. Bijela boja znači da će se teksturna mapa ili materijal vidjeti, a crna da neće. Međutim, u ulozi maske, crni dio ne znači da će objekt na tim mjestima biti proziran, već da će na tim mjestima biti prikazana druga teksturna mapa ili materijal (Epic Games, bez dat.-b).

5.4. *Normal* mapa

Normal je RGB teksturna mapa, jedna je od najvažnijih teksturnih mapa. Koristi se za simulaciju detalja površine. To je RGB mapa u kojoj svaka komponenta (R, G i B) odgovara X, Y i Z koordinatama vektora normale površine. Na taj način ona pohranjuje 3D informacije. (Sanchis Albert, 2019, str. 21).

Govori površini da poprimi složeniji oblik dodajući joj površinske detalje i ukazuje kako bi svjetlost trebala reagirati i reflektirati se od površinu. *Normal* mape simuliraju mala udubljenja i ispupčenja na površini – individualni pikseli uključuju kalkulacije osvjetljenja što rezultira osvijetljenim i zasjenjenim dijelovima koji bi bili prisutni da geometrija ima više topologije (Kelly, 2022, str. 25).

Dakle, za izračunavanje osvjetljenja, umjesto računanja vektora normale po vrhu 3D modela, s *normal* mapama mogu se računati vektori normale po pikselu, i na ovaj način znatno povećati detalji površine (Busquets Duran, 2014, str. 13).



Slika 21: Primjer *normal* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad)

Ova vrsta mape najčešće se koristi za projiciranje detalja modela visoke rezolucije na model niske rezolucije (Kumar, 2020, str. 37).

Ona zapravo i ne mijenja površinu 3D modela, već stvara iluziju dubine (Jantunen, 2017, str. 3).

Normal mapa po svojoj ulozi slična je *height* mapi, ali efikasnija od korištenja stvarne geometrije za prikaz detalja (Pai, 2019, str. 5).

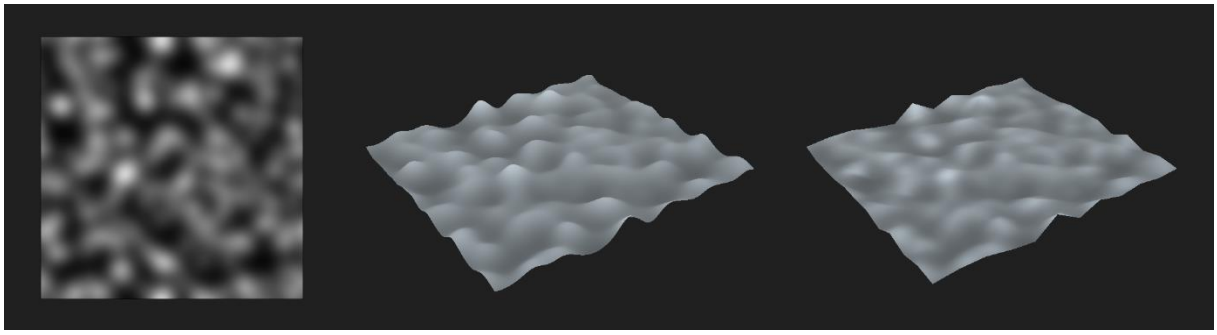
5.5. *Height* mapa

Height teksturna mapa, također poznata kao *displacement* mapa ili *bump* mapa, još je jedna važna teksturna mapa u sivim tonovima (eng. *grayscale*). Realističnija je teksturna mapa od *normal* mape, jer za razliku od *normal* mape može stvarno modificirati površinu 3D modela. Ova mapa postaje sve popularnija, jer je snaga modernih računala posljednjih godina počela znatno rasti (Jantunen, 2017, str. 3).

Height mapa pomiče površinu modela prema vrijednostima sive – crno dolje/unutra (prema unutrašnjosti modela), bijelo gore/van (van od modela). Crna predstavlja minimalnu, a bijela maksimalnu visinu. *Height* mapa mijenja geometriju površine duž vektora normale površine (Pai, 2019, str. 5; Sanchis Albert, 2019, str. 20).

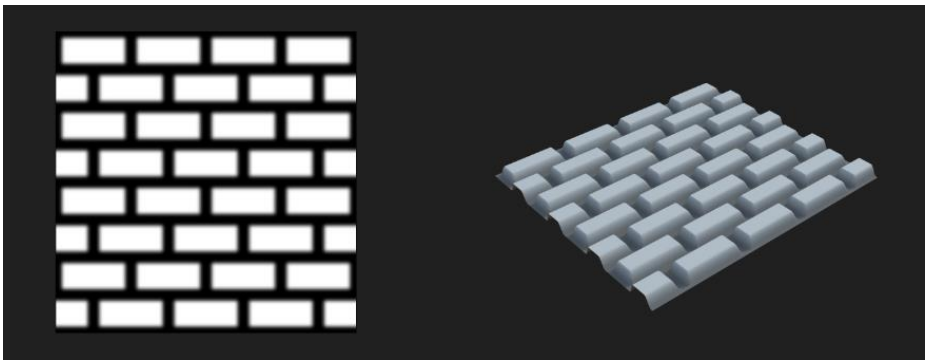
Ova mapa ovisi o vrhovima (eng. *vertex*) 3D modela odnosno o njegovoj rezoluciji (broju vrhova, broju poligona), jer ono što ona modificira jesu vrhovi modela. Što je model veće rezolucije, to će na njemu modifikacija geometrije bolje izgledati (Kumar, 2020, str. 37).

Dakle, za generiranje ispravne siluete potrebni su model visoke rezolucije (subdivizirana površina), ali i visokokvalitetna teksturna mapa (poželjno 64-bitna). Obično se koriste mape niske frekvencije (mape bez sitnih detalja) i radi zamućenje (eng. *blurring*) mape kako bi se izbjegao šum u geometriji odnosno ekstremni položaji nekih vrhova modela (Pai, 2019, str. 5).



Slika 22: Primjer *height* mape (lijevo), njena manifestacija na površini visoke rezolucije (sredina), njena manifestacija na površini niske rezolucije (desno) (autorski rad)

Height mape se obično koriste kad bi stvarno modeliranje detalja neke površine bilo dugotrajno i zamorno. Obično artist ručno modelira osnovni oblik nekog objekta, a kasnije, po potrebi, primjeni na njega *height* mapu za generiranje manjih detalja. Na primjer, može se ručno modelirati zid u obliku običnog kvadra, a zatim primijeniti na njega *height* mapa za generiranje reljefnih varijacija na način da izgleda kao da je izgrađen od cigla (Kelly, 2022, str. 25).



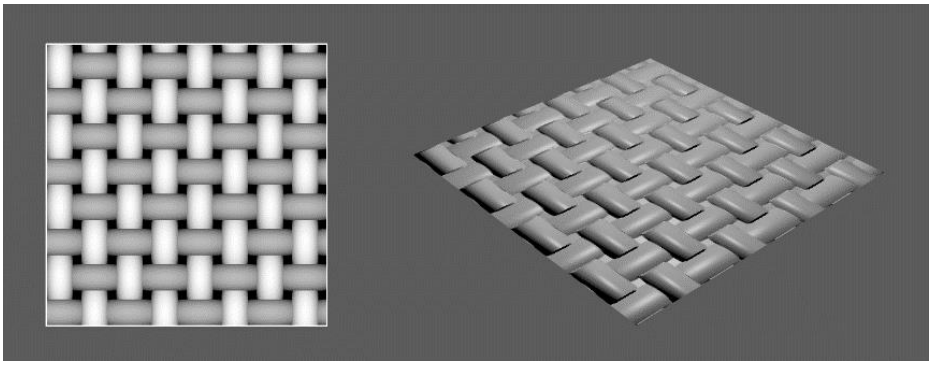
Slika 23: Primjer *height* mape (lijevo) i njena manifestacija na površini (desno) – daje efekt cigla na zidu (autorski rad)

5.6. *Bump* mapa

Već je ranije spomenuto da je *bump* mapa još jedan naziv za *height* mapu. Međutim, na *bump* mapu se ponekad može gledati kao na zasebnu teksturnu mapu, koja ima malo drugačiju ulogu od *height* mape.

Bump mapa je zapravo ista mapa kao i *height* map – mapa u sivim tonovima (eng. *grayscale*) za simuliranje površinskih detalja. Bitna razlika je ta što *bump* mapa ne modificira stvarnu geometriju modela.

Po ovome može se zaključiti da *bump* mapa ima istu ulogu kao *normal* mapa – simulira detalje površine bez da je zapravo mijenja. I to je dijelom točno. *Bump* mape su zapravo preteča *normal* mapama. One su jeftiniji način simuliranja reljefnih detalja, jer daju samo informacije o visini, dok *normal* mape daju informacije i o kutu (važno za ispravno osvjetljenje površine) – *bump* mape se sastoje od samo jedne dimenzije (tonovi sive), dok *normal* mape koriste tri dimenzije (kanale R, G i B), na taj način pružajući im prostora za više informacija od samo visine (Kumar, 2020, str. 36, 37; Mesquita, 2021b).

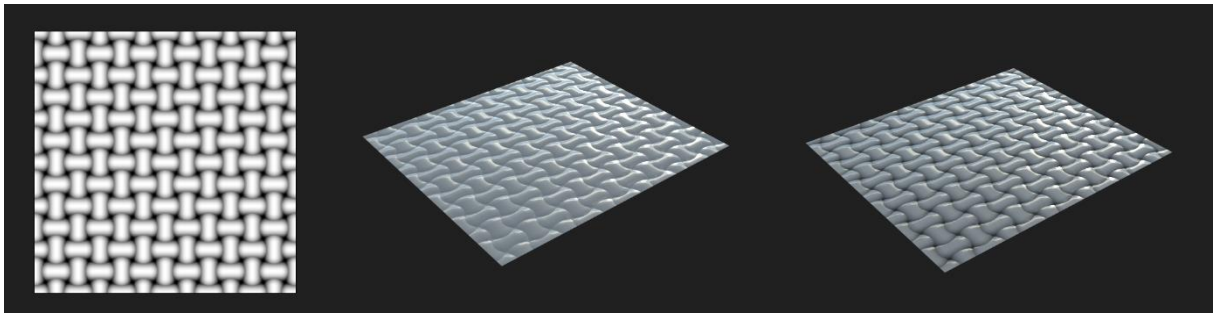


Slika 24: Primjer *bump* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad)

5.7. *Ambient occlusion* mapa

Ambient occlusion (AO) je teksturna mapa u sivim tonovima (eng. *grayscale*) koja opisuje koliko je svaki piksel površine izložen ambijentalnom osvjetljenju (osvjetljenju svog okruženja). Koristi se za generiranje sjena odnosno tame uzrokovane blizinom objekata. Tamnija vrijednost znači da površina prima manje svjetla (Busquets Duran, 2014, str. 15; Kumar, 2020, str. 37).

Ambient occlusion teksturna mapa nastaje tzv. „pečenjem“ (eng. *baking*) informacija iz visokofrekventne *normal* mape. To je prilično jednostavna tehnika i obično daje vrlo dobre vizualne rezultate (Pai, 2019, str. 5).



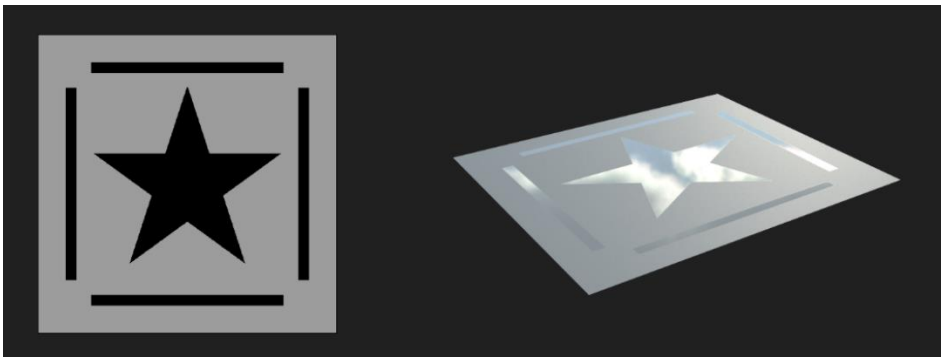
Slika 25: Primjer *AO* mape (lijevo), površina s *normal* mapom bez *AO* mape (sredina), površina s *normal* mapom i *AO* mapom (desno) (autorski rad)

5.8. *Roughness* mapa

Roughness je još jedna važna teksturna mapa u sivim tonovima (eng. *grayscale*). Opisuje mikropovršinske nepravilnosti koje uzrokuju razilaženje reflektirane svjetlosti. Definira koliko je hrapava mikroskopska površina materijala (Kelly, 2022, str. 21–22; Sanchis Albert, 2019, str. 20).

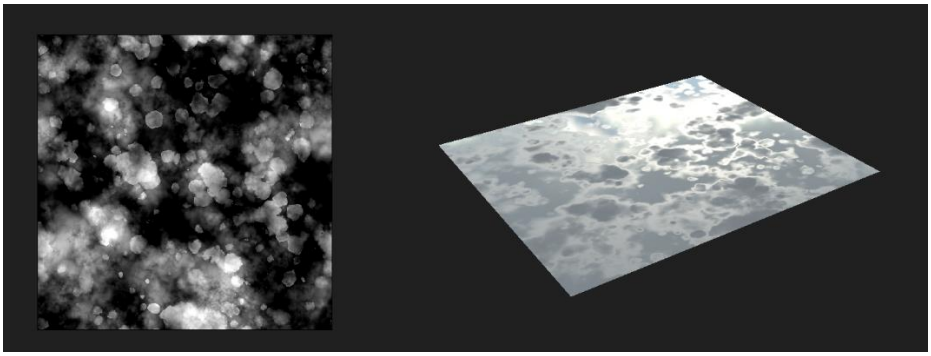
Ako je površina hrapava, reflektirana svjetlost će se razilaziti (mutna refleksija), površina će izgledati matirano. Ako površina nije hrapava odnosno ako je glatka, svjetlost će se reflektirati uniformno (zrcalna refleksija), moći će se vidjeti slika okruženja reflektirana na površini (Busquets Duran, 2014, str. 14).

Bijela boja predstavlja maksimalnu hrapavost, a crna boja predstavlja minimalnu hrapavost. Svi ostali sivi tonovi definiraju različite stupnjeve hrapavosti površine: vrijednosti bliže bijeloj – veća hrapavost, vrijednosti bliže crnoj – manja hrapavost (Kelly, 2022, str. 26).



Slika 26: Primjer *roughness* mape (lijevo) i njena manifestacija na površini (desno) – na crnim dijelovima se vidi refleksija okruženja (oblaci) (autorski rad)

Imajući samo jednu sivu vrijednost za cijelu *roughness* mapu, površina neće izgledati privlačno i zanimljivo, jer nema detalja. Da bi površina izgledala realističnije, moraju postojati male varijacije kako bi izgledala prirodnije. Kombinacijom svjetlijih i tamnijih tonova postići će se bolji rezultati od jedne sive boje (Kelly, 2022, str. 21–22; Mesquita, 2021c).



Slika 27: Primjer *roughness* mape s varijacijama sivih tonova (lijevo) i njena manifestacija na površini (desno) – izgleda realističnije i zanimljivije (autorski rad)

Za *roughness* mapu se može reći da je visokofrekventna *height* mapa s invertiranim vrijednostima sivih tonova, tako da viši i konveksni dijelovi modela budu sjajniji, a niži i konkavni hrapaviji, pod pretpostavkom da se u udubljenjima površina nakuplja prljavština (Pai, 2019, str. 5).

5.9. *Glossiness* mapa

Glossiness je teksturna mapa u sivim tonovima (eng. *grayscale*) koja definira mikropovršinske osobine površine tj. koliko je površina sjajna ili nije sjajna (odnosno matirana je). Obično se koristi umjesto *roughness* mape, i zapravo je invertirana *roughness* mapa.

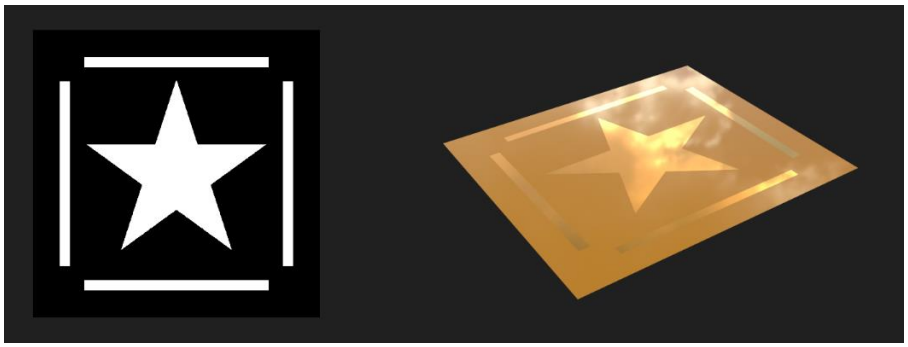
Bijela boja predstavlja maksimalan sjaj (savršeno glatka površina), a crna boja minimalan sjaj (potpuno hrapava površina) (Busquets Duran, 2014, str. 14; Mesquita, 2021c).

5.10. *Metallic* mapa

Metallic je važna teksturna mapa u sivim tonovima (eng. *grayscale*) koja definira je li površina metal ili nemetal. Bez nje, teško bi bilo kreirati metalne i nemetalne površine koje izgledaju realistično. Važno je imati na umu da metali većinom bojaju refleksiju, pa bi bez ove mape 3D artist trebao, osim intenziteta refleksije, ručno podešavati i njenu boju (Kelly, 2022, str. 22–23).

U poglavlju o PBR-u spomenuto je da su metali puno više reflektirajući od nemetala. To znači da će metali svjetlost više reflektirati (refleksija), dok će je nemetali više raspršivati (difuzija). Iz same ove činjenice može se zaključiti da *metallic* mapa ima utjecaj na *albedo* mapu, na način da će se na nemetalima više vidjeti *albedo* mapa (difuzija), a na metalima jako malo (njima će se više vidjeti refleksija okruženja).

Crna boja u mapi označava da u tom dijelu površina *nije* metalna, a bijela boja označava da površina u tom dijelu *je* metalna. Kad na površini u isto vrijeme na mjestima postoji metal i na mjestima nemetal, važno je da prijelaz iz bijele u crnu na teksturnoj mapi bude brz i precizan, bez gradijenta – npr. trenutku kada se boja s lakirane metalne podloge oguli, ispod nje je odmah metal, nema gradijenta između boje i metala (Kelly, 2022, str. 23–24; Pai, 2019, str. 5).



Slika 28: Primjer *metallic* mape (lijevo) i njena manifestacija na površini (desno) (*albedo* mapa je zlatne boje) – na metalnim dijelovima refleksija je jača i obojena (vidljiva je obojana refleksija oblaka), dok je na nemetalnim dijelovima slabija i neobojena (autorski rad)

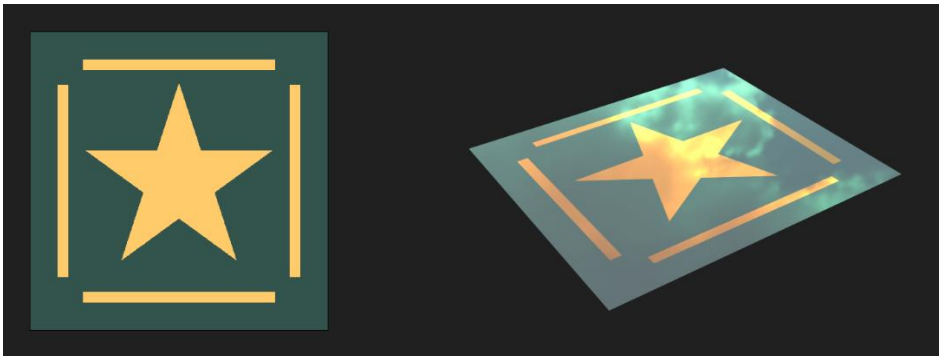
U stvarnom svijetu postoji samo metal (bijela boja u *metallic* mapi) ili nemetal (crna boja u *metallic* mapi), ne postoji između (Pai, 2019, str. 5). To znači da u *metallic* mapi, iako su mogući, sivi tonovi između crne i bijele nemaju smisla. Međutim, u kasnijim poglavljima će biti objašnjen smisao svih drugih nijansi sive osim crne i bijele.

5.11. *Specular* mapa

Specular je RGB teksturna mapa koja definira intenzitet i boju refleksije površine (upravo ono što kod *metallic* mape *shader* radi automatski). Svjetliji tonovi znače veća refleksija, dok tamniji manja refleksija. Obično se koristi umjesto *metallic* mape.

Daje veliku slobodu 3D artistu na način da on sam podesi vrijednosti intenziteta i boje refleksije, no upravo zbog toga je vrlo složena teksturna mapa, jer ponekad može rezultirati nerealističnim izgledom površine ako te vrijednosti nisu precizne za materijal iz stvarnog svijeta koji se pokušava simulirati. Zato većina artista preferira *metallic* mapu, u kojoj artist samo definira je li površina metalna ili nije, a sam *shader* se brine o intenzitetu i boji refleksije (A23D, bez dat.-a; CGHero, 2022; Kumar, 2020, str. 36; Mesquita, 2021c).

U poglavlju o PBR-u naglašeno je da su refleksija i difuzija svjetla međusobno isključive, pa se može zaključiti da *specular* mapa (isto kao *metallic* mapa) ima utjecaj na *albedo* mapu. Što je veća refleksija svjetla, to je manja njegova difuzija (više će se vidjeti refleksija okruženja, a manje *albedo* mapa). Vrijedi i obrnuto – što je manja refleksija svjetla, to je veća njegova difuzija (više će se vidjeti *albedo* mapa, a manje refleksija okruženja). Ovdje isto postoji poveznica s metalima i nemetalima – kad materijal ima iznimno visoku refleksiju, on zapravo vizualno postaje metal (samo metali imaju jako visoku refleksiju tj. vrlo malu difuziju).



Slika 29: Primjer *specular* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad)

5.12. Ostale vrste teksturnih mapa i zaključak

Uz prije spomenute, postoje još mnoge druge vrste teksturnih mapa koje se ponekad koriste u 3D produkciji kao što su *ID* mape, *fuzz* mape, *cavity* mape, *curvature* mape, *translucency* ili *thickness* mape, *emissive* mape itd. (Kumar, 2020, str. 37; Mesquita, 2021d).

Kombiniranjem različitih teksturnih mapa dobiva se konačan izgled materijala, no nije potrebno koristiti sve moguće mape – pojedina mapa se koristi po potrebi ovisno o tome koja osobina materijala se želi postići.

Teksturiranje se također može izvesti samo s *albedo* (*base color*) mapom, ali na taj način teksturirani 3D model općenito ne bi bio vrlo impresivan u rezultatima renderiranja (Jantunen, 2017, str. 4).

Isto tako, bilo bi moguće modelirati sve oblike i detalje 3D modela ručno u programu za 3D modeliranje, ali to obično nije najefikasniji način rada u 3D produkciji i za njihovu izradu bilo bi potrebno previše vremena. S postojećim teksturnim mapama moguće je dobiti rezultate gotovo jednako dobre kvalitete kao i s 3D modelima visoke rezolucije (Jantunen, 2017, str. 4).

6. Uobičajeni pristupi teksturiranju

Teksture se mogu kreirati u bilo kojem softveru koji dopušta rad s grafikom i spremanjem rezultata u nekom od formata slike (*JPEG*, *PNG*, *TIFF* itd.). To mogu biti 2D aplikacije za obradu slike poput popularnog i dobro poznatog Adobe Photoshopa, ili 3D aplikacije specijalizirane za teksturiranje.

Fizički zasnovano renderiranje (PBR) je samo tehnika renderiranja i, kao bilo koja druga tehnika renderiranja, interpretira dane joj teksturne mape na definirani način, neovisno o tome jesu li one napravljene u jednom ili drugom softveru, na ovaj ili onaj način. Dakle, za *shadere* koji čitaju dane im teksturne mape uopće nije bitno na koji način su one napravljene, bitan je samo rezultat – da je to grafička datoteka odnosno jedan od formata slike i da se drži standarda (da je RGB slika ili slika u sivim tonovima, ovisno o vrsti teksturne mape koju predstavlja).

Naravno, u današnje moderno vrijeme popularno je i najadekvatnije korištenje 3D aplikacija specijaliziranih za teksturiranje, jer je njihova primarna svrha upravo izrada tekstura pa je proces teksturiranja u njima intuitivan, brz i lagan. Međutim, i Adobe Photoshop, koji je izvorno bio samo aplikacija za 2D obradu slike, danas je puno više od toga. S raznim nadogradnjama kroz povijest i dodacima (eng. *plug-ins*), danas ima alate koji podržavaju i 3D teksturiranje. Ipak, postoje neke druge aplikacije koji služe samo za običnu 2D obradu slike i, iako nije najbolje, u njima je također moguće raditi teksture („3D Texturing Tools“, 2020; „Which Software Is Best for Texturing 3D Models?“, 2021; Kumar, 2020, str. 44–45).

6.1. Programski alati za teksturiranje

U ovom potpoglavlju je nabrojano osam najpopularnijih programskih alata za teksturiranje u 2022. godini. Valja odmah reći da su svi alati relativno dobri, i razlog zašto teksture na nekom 3D modelu izgledaju loše je često zbog slabih artistovih vještina, a ne zbog odabranog lošeg programskog alata za teksturiranje.

Substance 3D Designer je jedan od najboljih programskih alata za teksturiranje u vlasništvu Adobea. To je softver na bazi čvorova koji nudi proceduralan način izrade tekstura. U programu postoje mnogobrojni filteri, generatori šumova i uzoraka, i mnogi drugi korisni alati. Softver koristi fizički zasnovano renderiranje što pruža visoki realizam tekstura i renderirane 3D scene.

Substance 3D Painter je još jedan Adobeov programski alat za teksturiranje. Za razliku od Substance 3D Designera koji služi za dizajniranje popločavajućih tekstura, ovaj

softver služi za izravno bojanje tekstura po 3D modelu. Najbolji je za izradu personaliziranih tekstura za 3D modele. Također nudi mnogo filtera i generatora, te koristi fizički zasnovano renderiranje. Može se koristiti interaktivno sa Substance 3D Designerom za brzu razmjenu tekstura.

Photoshop je popularan Adobeov softver za 2D obradu slike. Međutim, zadnjih godina postao je puno više od toga. Raznim nadogradnjama kroz povijest i dodacima (eng. *plug-ins*) danas ima alate koji podržavaju rad s 3D grafikom. Moguće je bojati teksturne mape izravno po 3D modelima. Iako nije softver namijenjen upravo teksturiranju i nije najbolja opcija među svim 3D softverima koji postoje, može se koristiti za tu svrhu.

ZBrush softver je industrijski standard za digitalno 3D kiparstvo razvijen od strane Pixologic tima. Artisti ga najviše koriste za 3D modeliranje složenih organskih oblika vrlo visoke rezolucije (3D modeli s milijunima poligona), no softver također nudi alate za bojanje tekstura. Softver koristi PolyPaint tehnologiju u kojoj svaki poligon pohranjuje informaciju o boji. Najbolja je opcija za artiste koji ga već koriste za 3D modeliranje, tako da mogu odmah potom teksturirati u njemu, bez potrebe za korištenjem dodatnog 3D softvera.

Mari je profesionalan softver za teksturiranje razvijen od strane Foundry tima. Vrlo je složen softver, podržava do tisuće teksturnih mapa po jednom 3D modelu. Podržava teksture iznimno visoke rezolucije. Obično ga koriste profesionalni i iskusni artisti. Zbog svojih vrhunskih mogućnosti i visokog stupnja realizma popularan je 3D alat koji se obično koristi za teksturiranje u filmskoj industriji.

3D Coat je softver za digitalno 3D kiparstvo i bojanje razvijen od strane Pilgway tima. Koristi fizički zasnovano renderiranje što omogućuje izradu fotorealističnih tekstura. Profesionalci ga preporučuju kao dobar softver za ručno bojanje tekstura. Jedan je od najboljih i najpoznatijih softvera za retopologiju 3D objekata, što je proces pojednostavljivanja topologije 3D modela. Dobar je i jednostavan alat za korištenje, adekvatan za početnike.

Mudbox je softver primarno namijenjen teksturiranju, no pruža i digitalno 3D kiparstvo, razvijen od strane Autodeska. Zahvaljujući Autodeskovoj reputaciji, također je popularan među artistima. Pruža izravno bojanje tekstura po 3D modelu i ima napredne alate za retopologiju 3D objekata. Korisničko sučelje softvera je intuitivno i jednostavno za korištenje.

ArmorPaint je softver za teksturiranje kojeg je razvio Lubos Lenco. Artistima pruža mogućnost bojanja više teksturnih mapa u isto vrijeme što omogućuje jednostavno stvaranje koherentnih teksturnih mapa. ArmorPaint se u potpunosti izvodi na grafičkoj kartici (GPU) i njenim resursima, što pruža besprijekorno iskustvo izrade tekstura. Povrh svega, postoje i eksperimentalne verzije softvera za iPad i Android, što artistima daje dodatnu slobodu kreativnosti.

Osim spomenutih, postoje još Substance 3D Sampler, Quixel Mixer, Blender, Cinema 4D, SlefCAD, PaintCube, Sculptris, BodyPaint 3D, Granite, i mnogi drugi programski alati za teksturiranje.

Odabrani programski alat za teksturiranje ovisi o potrebama i zahtjevima, o tome što se želi postići i koliko jednostavne odnosno složene teksture se očekuju. Jedni alati su napredniji od drugih. Na primjer, ako zahtjevi teksturiranja nisu ozbiljni, dovoljno je koristiti Adobe Photoshop. Međutim, za ozbiljno i visokokvalitetno teksturiranje treba razmisliti o naprednim 3D aplikacijama namijenjenim upravo teksturiranju, kao što su Substance 3D Designer, Substance 3D Painter, Mari, Mudbox i druge specijalizirane 3D aplikacije („3D Texturing Tools“, 2020; *14 Best 3D Texture Painting Softwares as of 2022*, bez dat.; „Which Software Is Best for Texturing 3D Models?“, 2021; Manghnani, 2022; Mike, 2021; Sharma, 2021).

6.2. Tijek rada teksturiranja

Kao što je bilo moguće vidjeti u ranijem poglavlju, postoji velik broj različitih vrsta teksturnih mapa. Ne koriste se sve teksturne mape u isto vrijeme, a ovisno o tome koje vrste teksturnih mapa se koriste, govorimo o nekom teksturnom tijeku rada. Ono što je u centru ovog diplomskog rada je PBR teksturiranje, a ono podrazumijeva dvije glavne vrste: *metallic-roughness* tijek rada i *specular-glossiness* tijek rada.

U PBR *metallic-roughness* tijeku rada, dvije glavne teksturne mape su *metallic* mapa i *roughness* mapa. Iz samog naziva se da zaključiti da su u PBR *specular-glossiness* tijeku rada dvije glavne mape *specular* mapa i *glossiness* mapa. U jednom i u drugom teksturnom tijeku rada ove mape imaju istu svrhu, samo im je drugačija implementacija.

Oba tijeka rada daju slične rezultate. *Metallic-roughness* pruža jednostavan način postizanja materijala koji izgledaju metalno odnosno nemetalno, pa 3D artisti često preferiraju njega.

Nazivi ovih teksturnih tijekova rada govore koje su njihove glavne teksturne mape koje se koriste isključivo u tom tijeku rada, sve ostale teksturne mape su zajedničke za oba tijeka rada (A23D, bez dat.-b; Kumar, 2020, str. 37–38; McDermott, 2018, str. 47, 64; Parsons, 2017, str. 20).

Korišteni tijek rada ovisi o artistovim preferencijama i/ili odabranom rendereru / motoru za razvoj igara. Na primjer, Unity i Unreal Engine, dva vrlo popularna motora za razvoj igara, podržavaju oba tijeka rada, no Unity je primarno namijenjen *specular-glossiness* tijeku rada,

dok je Unreal Engine primarno namijenjen *metallic-roughness* tijeku rada (Epic Games, bez dat.-c; Unity Technologies, bez dat.-b).

6.3. Teksturiranje u igrama i teksturiranje u filmova

U nastavku se navode neke razlike u procesu teksturiranja za igre i filmove, no valja odmah reći da jedini razlog zbog kojih ti procesi nisu isti su performanse. Igre se izvode u stvarnom vremenu i, s obzirom na hardverska ograničenja računala, jedini način da korisnik postigne besprijekorno iskustvo je da se one optimiziraju. To znači da je u igrama kvaliteta žrtvovana zbog boljih performansi.

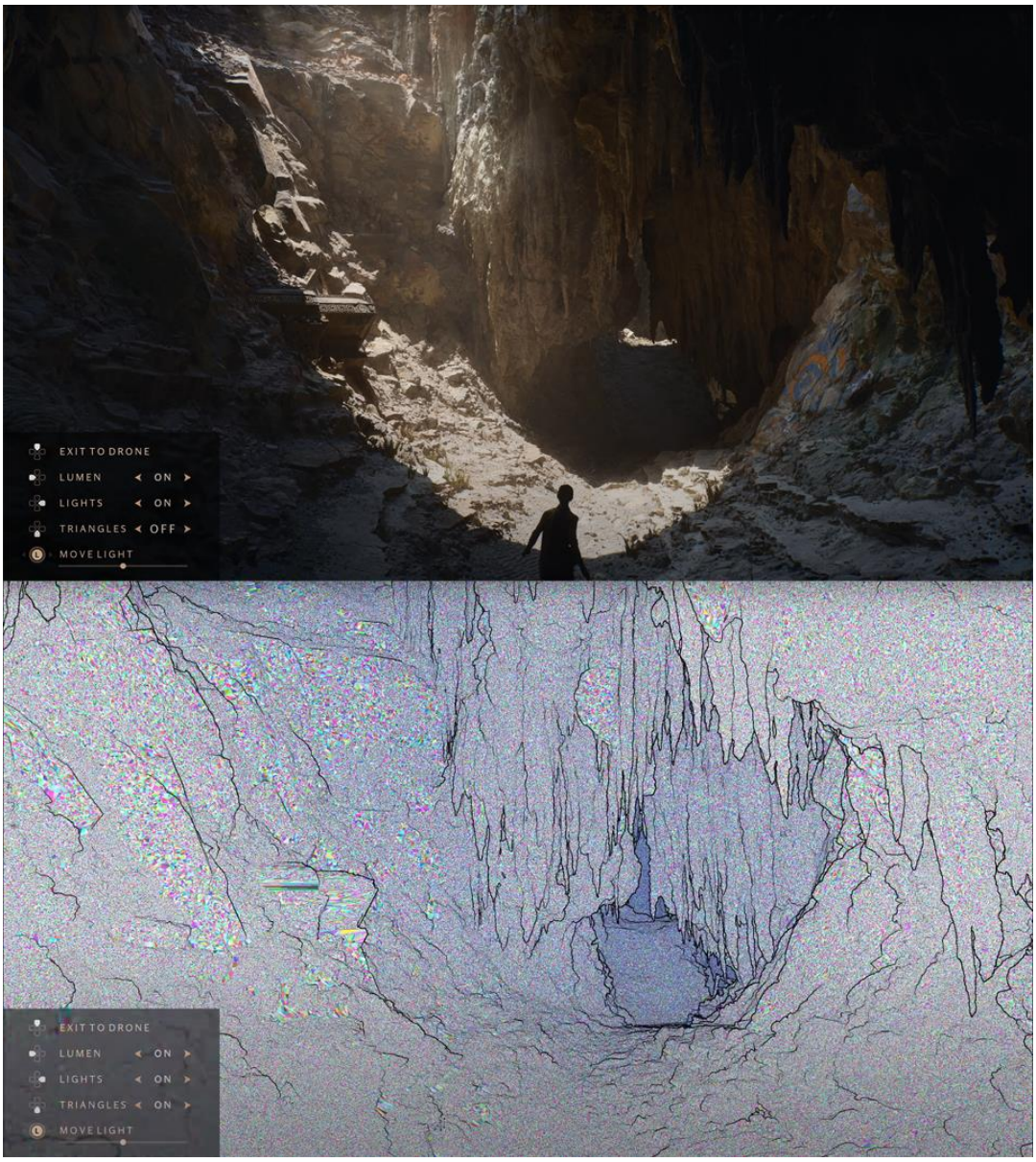
Za besprijekorno iskustvo u igrama broj renderiranih slika u sekundi ne bi smio biti ispod 60. Film je unaprijed renderirani sadržaj, pa za njega nije bitno koliko je vremena bilo potrebno za renderiranje jedne slike. Iako suvremena vrhunska računala imaju kapacitete za renderiranje tekstura iznimno visokih rezolucija, većina korisnika ipak si ne može priuštiti toliki luksuz, pa se igre uglavnom prilagođavaju tržišnoj većini.

Teksture u igrama su u pravilu niže rezolucije od tekstura u filmovima.

Objekti u filmovima znaju imati desetke teksturnih mapa, sve to da bi se postigao što veći realizam. U igrama objekti obično imaju do pet teksturnih mapa.

U filmovima se obično koriste 3D modeli visoke rezolucije (modeli s velikim brojem poligona) s *height* teksturnim mapama za postizanje visoke detaljnosti i realizma, dok se u igrama uglavnom koriste 3D modeli niže rezolucije, pa se u igrama za simulaciju detalja koriste *normal* teksturne mape (Kumar, 2020, str. 39, 40; *Why Can't We Create Video Game Graphics as Good as Cinematics?*, bez dat.).

5. travnja 2022. Epic Games je lansirao svoj najnoviji motor za razvoj igara – Unreal Engine 5. Njegovim dolaskom predstavljene su dvije revolucionarne tehnologije: Lumen i Nanite. Ovim tehnologijama Epic Games je, može se reći, na neki način uklonio jaz između filmova i igara – cilj im je bio postići realizam filmske kvalitete u realnom vremenu. Lumen omogućuje globalno osvjetljenje u realnom vremenu. No, ovdje je u fokusu Nanite. Nanite je tehnologija koja omogućuje beskonačnu razinu detalja 3D modela – 3D model može imati milijune i milijune poligona, zahvaljujući ovoj tehnologiji to neće utjecati na performanse igre. To znači da dolaskom ove tehnologije više nema potrebe raditi *normal* mape modelima (Epic Games, bez dat.-b, bez dat.-a).



Slika 30: U ovom demu od Epic Gamesa napravljenom u Unreal Engineu 5, testiranom u stvarnom vremenu na PlayStationu 5, slika prikazane scene sadrži oko 20 milijuna trokuta (poligona). Donja slika prikazuje kako ti trokuti izgledaju – svaki susjedni trokut je prikazan drugom bojom kako bi se bolje uočio. Većina trokuta je veličine piksela na ekranu, zbog čega slika izgleda kao šum (Prema: Unreal Engine, 2020)

Unreal Engine 5 je nedavno izašao. Za korištenje spomenutih njegovih tehnologija u punom potencijalu, ipak su potrebna jača računala. Iako je zahvaljujući Nanite tehnologiji u igrama stvarno moguće imati modele iznimno visoke rezolucije bez da oni utječu na performanse, ja ipak mislim da se oni još neko vrijeme neće koristi u igrama 100 % takvi, već da će se koristiti dijelom i *normal* mape, jer bi na taj način veličine igara bile u terabajtima. Iako su ove revolucionarne tehnologije stvarno prisutne i mogu se koristiti, imaju svoja tehnička ograničenja i još uvijek se razvijaju.

6.4. Korisni savjeti za artiste teksturiranja

Svaki artist teksturiranja, bez obzira u kojem softveru ili na koji način kreira teksture, za dobre rezultate bi trebao uzeti u obzir dva važna savjeta.

Proučavanje referenci. Artist bi trebao proučiti fizička svojstva materijala iz primjera stvarnog svijeta, od najosnovnijih oblika koji se pojavljuju na površini do složenijih. Također, trebao bi vidjeti kakav kontakt svjetlost ostvaruje s površinom. Ovo posebno vrijedi za materijale iz stvarnog svijeta koje teksturama pokušava simulirati (Sanchis Albert, 2019, str. 20). Međutim, svi ovi materijali ne moraju stvarno biti dostupni artistu, danas na webu je moguće pronaći fotografije gotovo bilo kojeg materijala, što dovodi do sljedećeg savjeta.

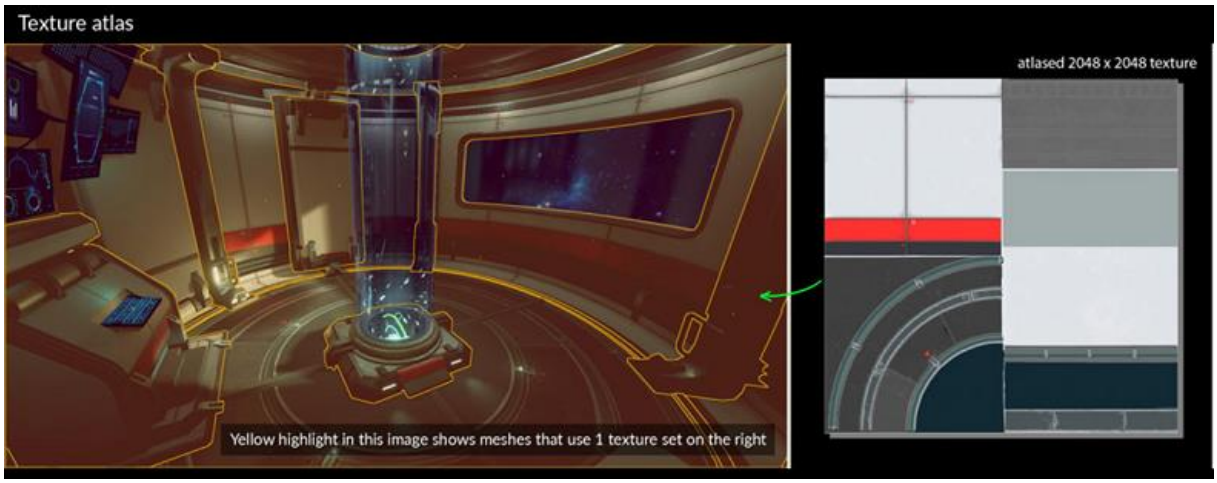
Vlastita zbirka referenci. Kad god putuje svijetom i pronađe nešto zanimljivo, dobra praksa artista je snimiti fotografije i spremiti ih u svoju vlastitu zbirku referenci. To mogu biti fotografije bilo čega, npr. drvo, cigle, pločice, tkanina, oblaci itd. (Kumar, 2020, str. 34).

6.5. Tehnike efikasnog teksturiranja

Izgleda da je trenutni trend da svaka nova generacija igara mora biti grafički detaljnija od prethodne kako bi se zadovoljili sve veći zahtjevi igrača. Postizanje ove razine detalja pomiče granice i hardvera na kojem se igre izvode i ljudi koji ih stvaraju. To je razlog zašto programeri igara moraju razvijati svoje igre imajući na umu efikasnost. To je posebno istinito za AAA igre koje žude hiperrealističnom stilu, ali čak i s jednostavnijim grafičkim stilovima mogu se poduzeti radnje za poboljšanje performansi igara, pogotovo na jeftinijim platformama.

6.5.1. Teksturni atlasi

Teksturni atlasi jedna su od najčešćih tehnika koje artisti koriste za optimizaciju igara. To su teksture koje u sebi sadrže druge, manje teksture – više tekstura u jednoj. Te male teksture mogu biti teksture samo jednog objekta, no često je praksa da se jedan teksturni atlas koristi za više različitih objekata. To ima smisla za objekte koji su na neki način povezani, npr. svi objekti računala (kućište, ekran, tipkovnica itd.). Ako je skup objekata uvijek daleko od kamere, što znači da im nije potrebna tekstura visoke rezolucije, za njih je također dobro koristiti teksturni atlas. Koristiti istu teksturu za više objekata je poželjno, jer će objekti koji dijele teksturu izvršiti samo jedan poziv za iscrtavanje (eng. *draw call*) (Aaltonen, 2022, str. 7, 9; Arm Developer, bez dat.).

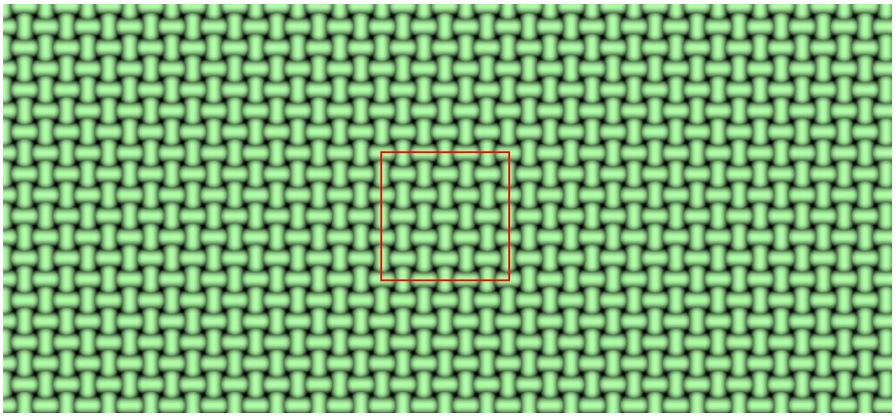


Slika 31: Žuto označeni objekti dijele istu teksturu (Arm Developer, bez dat.)

„Poziv za iscrtavanje (eng. *draw call*) je skup instrukcija iz procesora (eng. *CPU*) koje govore grafičkoj kartici (eng. *GPU*) što treba renderirati na zaslonu.“ (Aaltonen, 2022, str. 7). Više zasebnih tekstura će uzrokovati više poziva za iscrtavanje, što usporava performanse igre.

6.5.2. Popločavanje

Popločavajuće teksture (eng. *tiling textures*) već se dug niz godina koriste u igrama. To su teksture koje nemaju vidljive granice tzv. šavove (eng. *seams*) kada se polože jedna uz drugu u svim smjerovima (Aaltonen, 2022, str. 8).

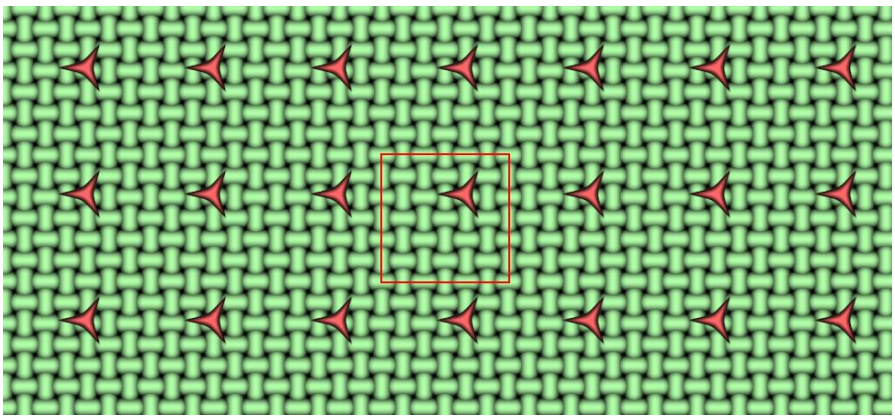


Slika 32: Primjer popločavanja jedne popločavajuće teksture, granice teksture su označene crvenim kvadratom (autorski rad)

Ideja popločavajućih tekstura je da se jedna mala tekstura može koristiti za popločavanje veće površine. To štedi na memoriji, a budući da se one ponavljaju po površini, nemaju fiksnu rezoluciju odnosno mogu dobro izgledati bez obzira koliko blizu se gledaju.

Velike površine, kao što su zidovi, podovi i tereni obično koriste popločavajuće teksture, jer bi jedinstveno teksturiranje istih zahtijevalo masivne veličine tekstura za dobar rezultat.

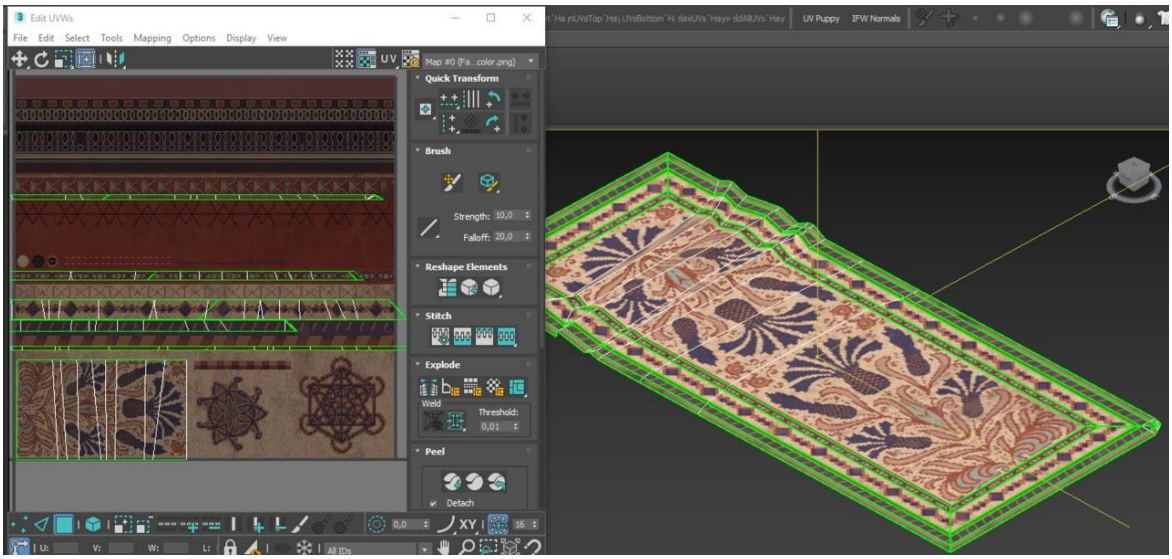
Popločavajuće teksture bi općenito trebale biti prilično ujednačene, niti jedan dio se ne bi trebao isticati. Ako takva tekstura u sebi ima upadljivi uzorak, njeno ponavljanje će biti vrlo očito. Budući da ne bi smjele imati uzorke koji se ističu, s ovim teksturama artisti su ograničeni u smislu dodavanja detalja (Aaltonen, 2022, str. 9–10; Ta'eed, 2022).



Slika 33: Primjer kako upečatljivi uzorci na popločavajućim teksturama još više ističu njihovu karakteristiku popločavanja odnosno ponavljanja (autorski rad)

6.5.3. Trake

Trake (eng. *trim sheets*) su slične popločavajućim teksturama, jedina razlika je što se ponavljaju (popločavaju) po jednoj osi. Često se koriste za teksturiranje dugih dijelova geometrije, npr. za ornamente na rubovima tepiha. Obično se izrađuju u obliku teksturnog atlasa, tako da tvore više tankih traka na jednoj teksturnoj mapi (Aaltonen, 2022, str. 10–11).



Slika 34: Primjer korištenja traka (Kronenberger, 2020)

6.5.4. Tehnika ultimativne trake

Osnova tehnike ultimativne trake (eng. *The Ultimate Trim technique*) je da se sve teksture rade na način da slijede standardizirani raspored. Taj raspored se sastoji od više traka različitih veličina, i svaka veličina ima dvije verzije zbog više varijacija. Različite veličine traka postoje zbog mapiranja na dijelove modela različitih veličina kako bi se zadržala ujednačena gustoća teksture po cijelom modelu. Osim toga, više varijacija traka u jednoj teksturi štedi na pozivima za iscrtavanje. Na dnu svake teksture je slobodan prostor rezerviran za sve posebne detalje koji bi modelima mogli zatrebati, poput gumba, naljepnica, čepova i slično. Naravno, za uspješno korištenje ovog rasporeda, svi modeli se moraju UV odmotati u trake i mapirati na način da slijede taj raspored traka (Aaltonen, 2022, str. 13; Olsen, 2015).

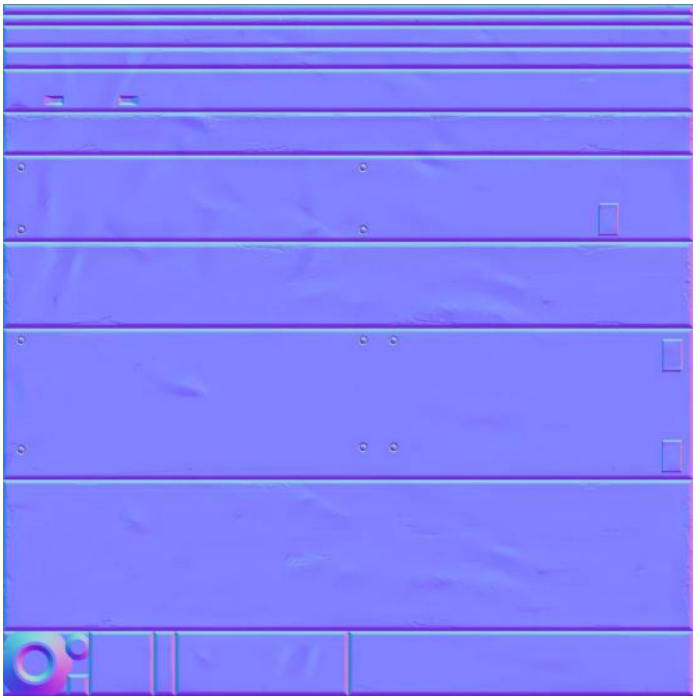
Slika 35: Primjer jedne teksture na kojoj je primijenjena tehnika ultimativne trake – vidljiv je standardizirani raspored traka (Olsen, 2015)

Koristeći ovu tehniku, svi objekti su UV mapirani prema istom rasporedu, te artist može mijenjati teksture objektu bez problema. To znatno štedi na vremenu. Ova tehnika također znatno štedi na memoriji, jer se teksture kreiraju za višekratnu upotrebu (Lazzaro, 2019).

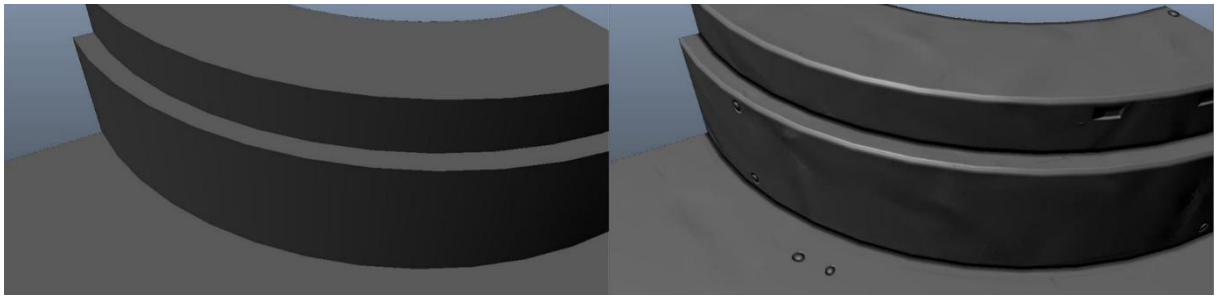
Ovu tehniku je razvio Insomniac Games kad je imao mali tim za razvoj okruženja AAA igre. Morali su dati prednost brzini nad svim ostalim budući da su imali veliko okruženje otvorenog svijeta za teksturiranje, a relativno mali tim za realizaciju (Olsen, 2015).

Osim što standardizira UV raspored, tehnika ultimativne trake također na neki način standardizira *normal* mape. U *normal* mapi, svaka traka duž svojih rubova ima ucrtan kut od

45 stupnjeva. To stvara iluziju ukošenog (tupog) ruba kada se koristi po rubu dvaju spojenih poligona. Ovo je odličan način simuliranja geometrije visoke rezolucije bez dodatnih troškova nad performansama (Aaltonen, 2022, str. 14; Olsen, 2015).



Slika 36: Primjer jedne *normal* mape na kojoj je primijenjena tehnika ultimativne trake – vidljiv je kut od 45 stupnjeva na rubovima svih traka (Olsen, 2015)



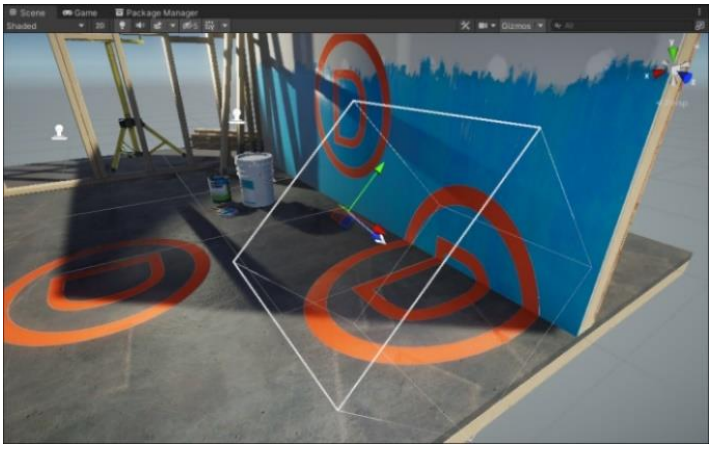
Slika 37: Objekt bez *normal* mape (lijevo), objekt s *normal* mapom (desno). Kut od 45 stupnjeva na *normal* mapi stvara iluziju da objekt ima veću rezoluciju (Prema: Olsen, 2015)

6.5.5. Naljepnice

Naljepnice (eng. *decals*) su poligoni ili mali objekti s teksturom postavljeni povrh većeg objekta. One su dobar način skrivanja repetitivnosti popločavajućih tekstura. Dakle, koriste se za razbijanje monotonije popločavajućih tekstura odnosno za dodavanje raznolikosti. Međutim, koriste se još i za dodavanje odgovarajuće atmosfere scenama. Uobičajeni slučajevi korištenja su simulacija oštećenja, kao što su pukotine ili ogrebotine, prljavština i razne oznake (Aaltonen, 2022, str. 15).

Slika 38: Različite naljepnice na popločavajućoj teksturi (Hurtubise, 2018)

Postoje dvije različite vrste naljepnica koje se koriste u različite svrhe. Prva vrsta su projicirane naljepnice. One nastaju tako da se njihova tekstura planarno projicira na površinu objekta. Obično se koriste samo u igrama za projiciranje po potrebi, a najbolji primjer jednih takvih naljepnica su oštećenja na zidu zbog metaka u igrama pucanja (Unity Technologies, bez dat.-a).



Slika 39: Primjer projiciranja teksture naljepnice na površine objekata unutar Unity motora za razvoj igara (Unity Technologies, bez dat.-a)

Druga metoda izrade naljepnica podrazumijeva korištenje male količine geometrije koja sadrži teksturu naljepnice. To je mali komad dodatne geometrije naslonjen na glavni objekt. Jedan dobar primjer korištenja su naljepnice koje se stavljaju na rubove objekta za simulaciju njihovih oštećenja – obično se izrađuju u obliku više različitih traka (eng. *trims*) u jednom teksturnom atlasu zbog većeg izbora mogućih varijacija oštećenja. To stvara visoke detalje i realizam, dok popločavajuće teksture koje se nalaze ispod ostaju netaknute (Aaltonen, 2022, str. 16–18).

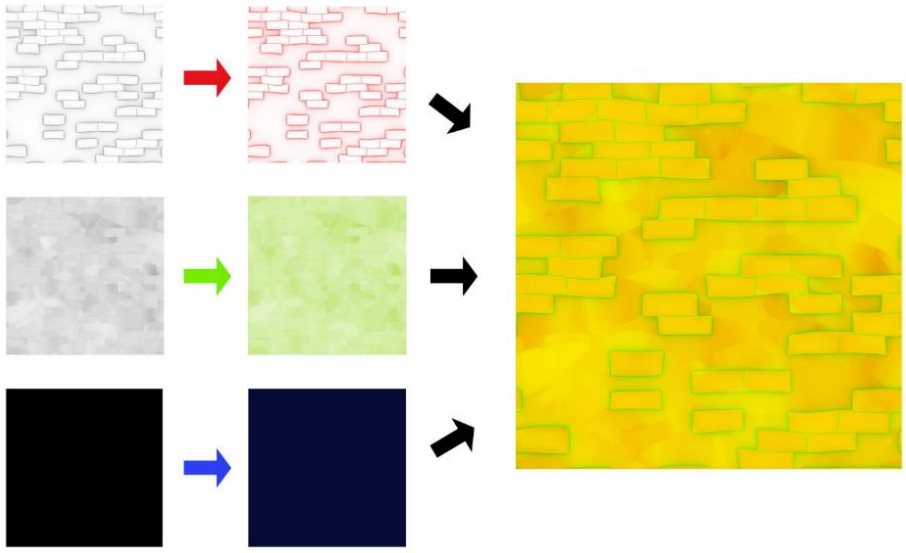
Dok projektor naljepnica uvijek projicira cijelu teksturu koja mu je dana, naljepnice koje su objekt sam za sebe su UV mapirane, što znači da se više vrsta naljepnica može pohraniti na jednu teksturnu mapu i koristiti zasebno prema potrebi. Također, korištenje iste teksturne mape s više varijacija naljepnica pridonosi poboljšanju performansi (Aaltonen, 2022, str. 19; Unity Technologies, bez dat.-a).

6.5.6. Pakiranje kanala

Kao što se moglo vidjeti u poglavlju o teksturnim mapama, mnoge od njih su slike u sivim tonovima (eng. *grayscale*). Kad je slika u boji, onda je to RGB ili RGBA slika. To znači da ima pojedinačne kanale koji pohranjuju crvene, zelene, plave, a ponekad i alfa (prozirnost) vrijednosti slike, u svaki kanal zasebno. Slika u sivim tonovima je slika u kojoj crveni, zeleni, plavi i alfa kanali svi imaju iste vrijednosti. Dakle, teoretski, nisu potrebni svi kanali za držanje tih informacija, jedan kanal može pohraniti sve potrebne informacije.

Pakiranje kanala (eng. *channel packing*) je proces u kojem se više različitih slika u sivim tonovima ubacuje u svaki od RGB, a ponekad i alfa (A) kanal slike, umjesto da postoje četiri zasebne slike u sivim tonovima. Kasnije, kad program treba neku određenu mapu, samo pročita podatke iz odgovarajućeg kanala slike. To štedi na memoriji (jedna slikovna datoteka umjesto četiri) i pozivima za iscrtavanje.

Sve mape u sivim tonovima se mogu pakirati u kanale, npr. *metallic* mape, *roughness* mape, *ambient occlusion* mape, *height* mape, *alpha* mape (pogotovo njihov poseban slučaj – maske) itd. (Aaltonen, 2022, str. 19–20; Busquets Duran, 2014, str. 15–16; Mesquita, 2021a).



Slika 40: Primjer pakiranja *AO*, *roughness* i *metallic* mapa u jednu mapu (Aaltonen, 2022, str. 20)

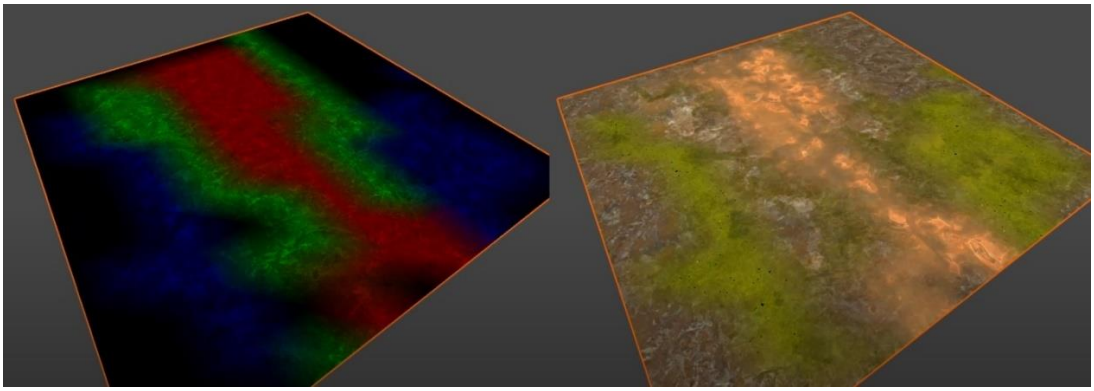
6.5.7. Miješanje tekstura

Miješanje tekstura (eng. *texture blending*) je tehnika kojom se spaja više različitih tekstura ili materijala. To je jedan od najčešćih načina za postizanje veće raznolikosti i više detalja u okruženju. Miješanje tekstura također se može koristiti za skrivanje očitog ponavljanja (popločavanja) kada se koristi u kombinaciji s popločavajućim teksturama. Postoji nekoliko različitih načina postizanja ove tehnike, sa svojim prednostima i nedostacima (Aaltonen, 2022, str. 22; Parsons, 2017, str. 11–12).

Slika 41: Tekstura pločica je pomiješana s teksturom zemlje i s teksturom trave (Prema: Quixel, 2020)

Bojanje vrhova (eng. *vertex painting*) je vrsta miješanja tekstura, gdje svaki vrh modela pohranjuje informacije o R (*red*), G (*green*) i B (*blue*). Svaki vrh ima samo jednu od R, G i B boja. Između dva vrha postoji gradijent između njihovih boja. Ove R, G i B vrijednosti se koriste za miješanje između različitih materijala.

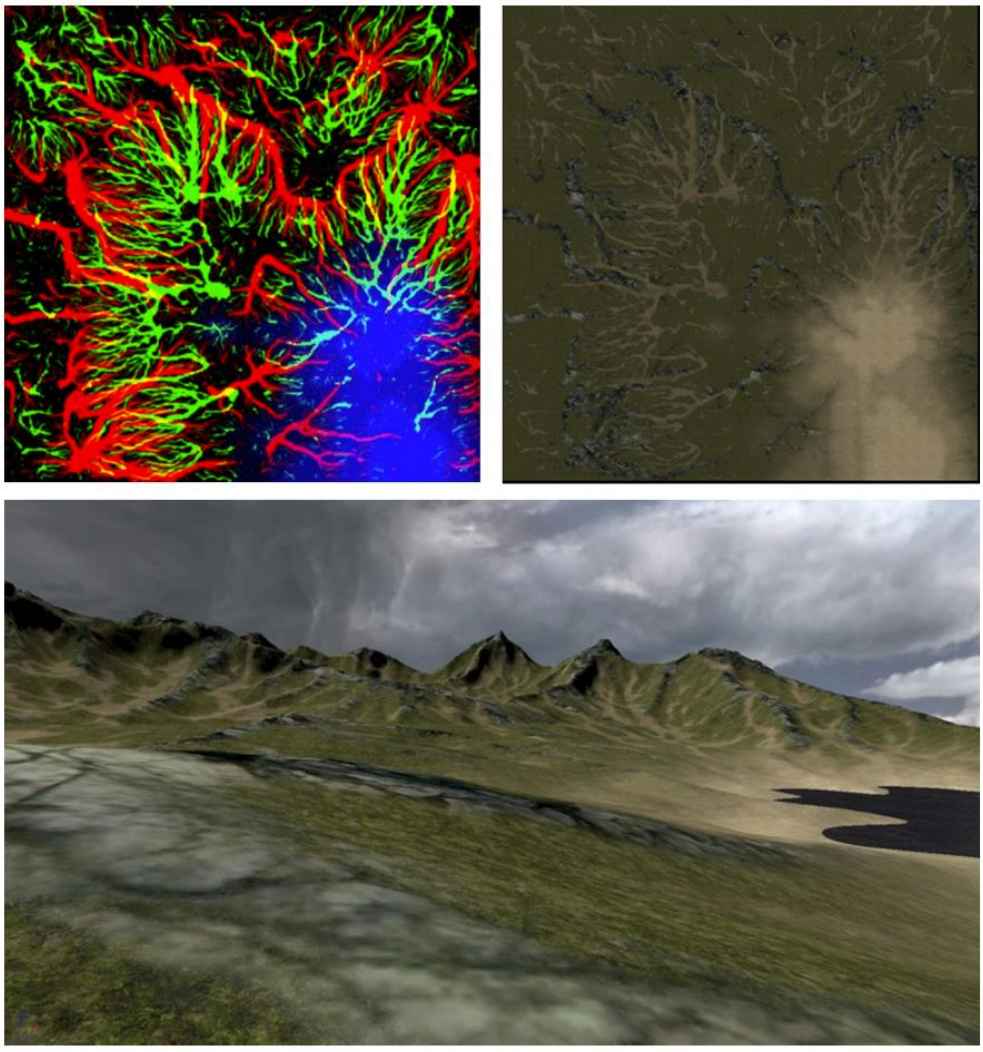
Za potrebe primjera, neka R predstavlja prvu teksturu, G drugu teksturu. Ako je R vrijednost 100 % (G vrijednost 0 %), prikazat će se prva tekstura. Ako je G vrijednost 100 % (R vrijednost 0 %), prikazat će se druga tekstura. Ako je R vrijednost 30 % odnosno G vrijednost 70 %, prva tekstura će biti vidljiva 30 %, a druga 70 % itd. Na isti način se mogu miješati tri teksture, gdje će vidljivost svake teksture oviti o udjelu kanala iz RGB kojeg zastupa.



Slika 42: Miješanje tekstura metodom bojanja vrhova (Prema: OmniShade, 2022)

Bojanje vrhova je vrlo isplativa metoda, jer masovno štedi memoriju – objekti ne trebaju teksturu koja će definirati miješanje, objekti samo trebaju pohraniti jednu RGB vrijednost u svaki od svojih vrhova. Najveći nedostatak ove metode je u samoj njenoj srži – modeli moraju imati dovoljno vrhova odnosno moraju biti veće rezolucije da bi miješanje izgledalo dobro (Aaltonen, 2022, str. 23–25).

Alternativna metoda za miješanje tekstura je korištenje *blend* mape. *Blend* mape su bitmape čiji RGB(A) kanali su pakirani mapama sivih tonova. Ove mape sivih tonova se koriste kao maske za definiranje gdje se određena tekstura treba prikazati na modelu. *Blend* mape se najčešće koriste s popločavajućim teksturama na vrlo velikim površinama, kao što je teren u igrama otvorenog svijeta. To je slučaj u kojem bojanje vrhova ne bi bila najbolja opcija, jer bi teren morao imati nevjerojatno visoku rezoluciju da bi miješanje tekstura izgledalo dobro (Aaltonen, 2022, str. 26; Epic Games, bez dat.-b).



Slika 43: Primjer korištenja *blend* mape za miješanje tekstura na velikoj površini (Aaltonen, 2022, str. 26)

6.6. Koncept trošenja materijala i njegova uloga u vizualnom pripovijedanju

Vrlo malo površina u stvarnom svijetu je bez nesavršenosti, čak i potpuno novi objekti imaju neke površinske nepravilnosti. Kod računalno generiranih (eng. *computer generated; CG*) objekata postoji rizik da ne budu uvjerljivi kada su im površine previše savršene. Kada materijalni artist dobije scenu na kojoj će raditi, njegova je prva odgovornost procijeniti kontekst te scene. Kontekst scene je važan jer informira artista gdje i na koji način primijeniti nesavršenosti. Na primjer, je li se radi o potpuno novoj i modernoj građevini iz 21. stoljeća, ili se radi o staroj kolibi iz 16. stoljeća; je li predmet nov ili se već nekoliko godina koristio i slično (Parsons, 2017, str. 1).

Nesavršenosti poput ogrebotina, hrđe i mahovine nastaju zbog različitih čimbenika. Njihovi formirani uzorci mijenjaju cjelokupni izgled materijala. Takve promjene i nesavršenosti općenito se definiraju kao raspadanje ili trošenje materijala (eng. *weathering*). Trošenje materijala se može podijeliti u tri opće kategorije: biološko, kemijsko, mehaničko (Parsons, 2017, str. 4–5; *Weathering*, bez dat.).

Slika 44: Primjer biološkog trošenja materijala (lijevo), primjer kemijskog trošenja materijala (sredina), primjer mehaničkog trošenja materijala (desno) (Parsons, 2017, str. 4)

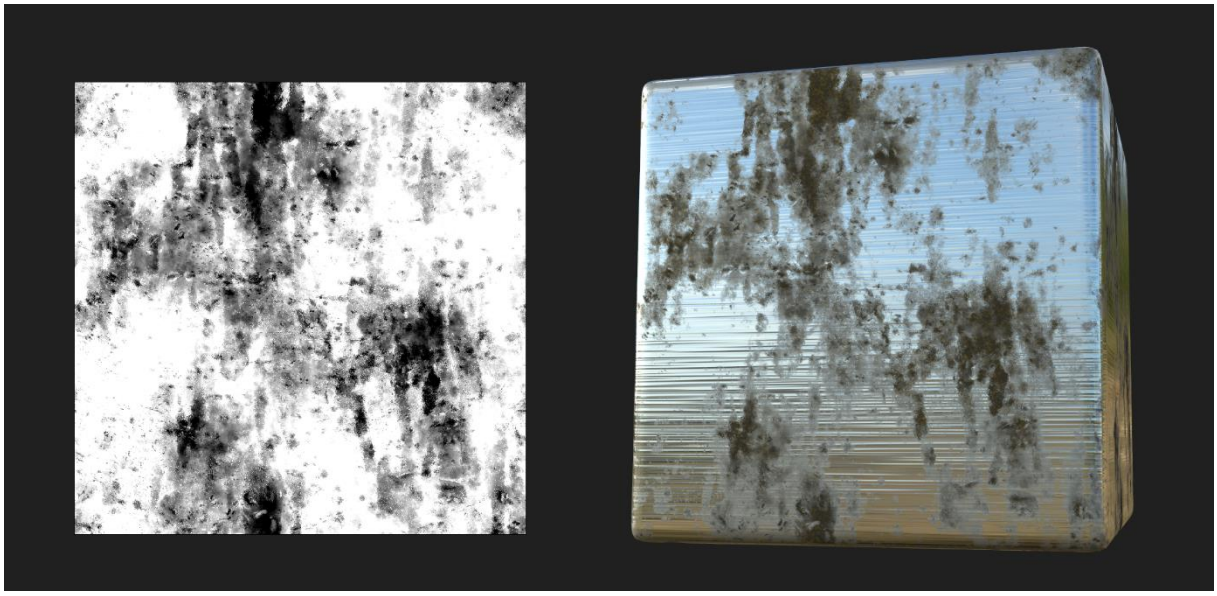
„Biološko trošenje je proces rasta živih organizama i stvaranja njihovih uzoraka na površinama, na primjer plijesni ili mahovine.“ (Parsons, 2017, str. 5).

„Kemijsko trošenje mijenja fizičke karakteristike izvornog materijala, bilo stvaranjem ili uništavanjem minerala.“ (Parsons, 2017, str. 5). Neki primjeri za to su oksidacija, hidroliza i karbonizacija.

Mehaničko ili fizičko trošenje slabe materijale zbog fizičkih sila. U te fizičke sile se ubrajaju temperaturne fluktuacije i abrazija. Temperaturne fluktuacije mogu uzrokovati širenje

ili skupljanje površine stvarajući fizički stres na površini, što dovodi do pukotina ili lomova na površini. Abrazija od stalne uporabe ili samo od izlaganja prirodnim elementima polako razgrađuje površinu. Uobičajeni rezultati su ljuštenje ili pucanje boje, ogrebotine i udubljenja. Neke fizičke sile uzrokuju pomicanje i taloženje drugih materijala, poput prljavštine ili prašine (Parsons, 2017, str. 6; *Weathering*, bez dat.).

Na primjer, u slučaju jednog trošenja materijala, dobro je prisjetiti se *metallic* teksturne mape. To je mapa u sivim tonovima gdje crna označava nemetal, a bijela metal. Taj *metallic* parametar iz *metallic* teksturne mape u stvarnom svijetu postoji samo u crnim (dielektrični materijal) ili bijelim (metalni materijal) vrijednostima koje ljudsko oko može vidjeti, što znači da sivi tonovi između nemaju smisla. Međutim, ti sivi tonovi između bijele i crne se mogu iskoristiti upravo za simulaciju neke vrste trošenja materijala. Na primjer, metalna površina koji na sebi ima malo prašine neće izgledati 100 % metalna – u isto vrijeme se vidi metal, ali i prašina koja ga pokriva. Što je više prašine na metalnoj površini, ona će izgledati manje metalna. Dakle, za male količine prašine na metalu, vrijednost *metallic* parametra se može staviti na nijansu malo tamniju od bijele. Ako metalna površina na sebi ima previše prašine, ta vrijednost se može znatno potamniti (Pai, 2019, str. 5).



Slika 45: Primjer prljavštine na metalu: na mjestima gdje su male količine prljavštine, na *metallic* mapi (lijevo) su nijanse sive; na mjestima gdje je prljavština potpuno pokrila metal, na *metallic* mapi je crno (autorski rad)

Svi elementi koji čine okruženje (osvjetljenje, boje, materijali i oblici), pa tako i koncept trošenja materijala, igraju važnu ulogu u vizualnom pripovijedanju – mogu utjecati na emocije, pružiti informacije o svijetu ili podržati priču koja se predstavlja. Svi ti elementi u kompoziciji rade zajedno kako bi izgradili atmosferu i prenijeli priču (Ayyildiz, 2020, str. 19, 24).

Slika 46: Scena u *Fallout 4* igri prije rata (gore), ista scena nakon rata (dolje) (Prema: deathmule, 2015)

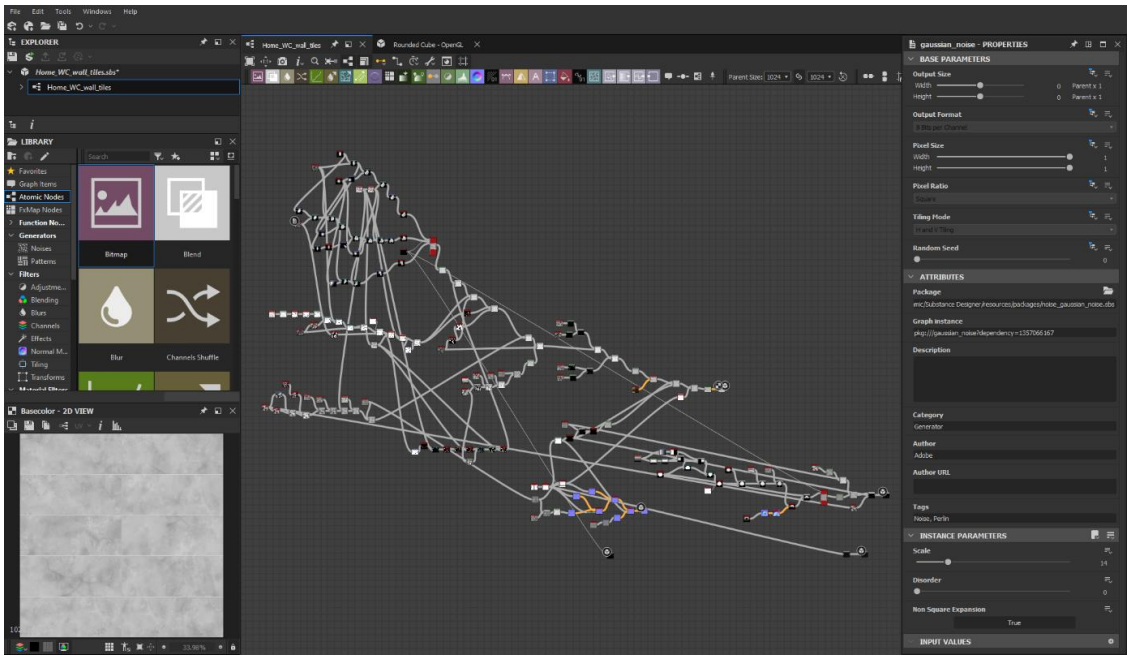
7. Primjer izrade tekstura za 3D modele

U ovom poglavlju detaljno je objašnjen Substance 3D Designer programski alat za teksturiranje nakon čega je u praktičnom primjeru prikazan proces izrade jednog popločavajućeg materijala u istome.

7.1. Substance 3D Designer

Substance 3D Designer je nelinearan (eng. *non-linear*), nedestruktivan (eng. *non-destructive*) softver na bazi čvorova (eng. *node based*), koji omogućuje artistima da generiraju teksture odnosno materijale na proceduralan način. Razvijao ga je od 2010. godine Allegorithmic tim, no u siječnju 2019. Adobe je kupio Allegorithmic, zajedno s cijelom obitelji Substance 3D aplikacija (Adobe, bez dat.-b; Young & Skuratovskiy, 2019).

Ovaj softver koristi fizički zasnovano renderiranje (PBR) kao primarnu metodu sjenčanja, što pruža izradu visoko realističnih materijala. Dva glavna PBR tijeka rada koji se u njemu mogu koristiti za izradu tekstura su *metallic-roughness* i *specular-glossiness* (Adobe, bez dat.-a).



Slika 47: Primjer jednog projekta u Substance 3D Designeru (autorski rad)

7.1.1. Softver na bazi čvorova

Čvor je osnovni element projekta u Substance 3D Designeru, građevni blok koji pridonosi razvoju teksturnih mapa odnosno materijala. On sadrži informacije 2D slike

(teksturne mape). Te informacije mogu biti generirane u tom čvoru ili mu mogu biti proslijeđene iz drugog. Svaki čvor ima svoju funkciju sa svojim vlastitim parametrima koje artist može mijenjati kako bi modificirao rezultat.

Čvorovi se međusobno povezuju zbog razmjene i nadogradnje informacija. Svi povezani čvorovi čine graf.

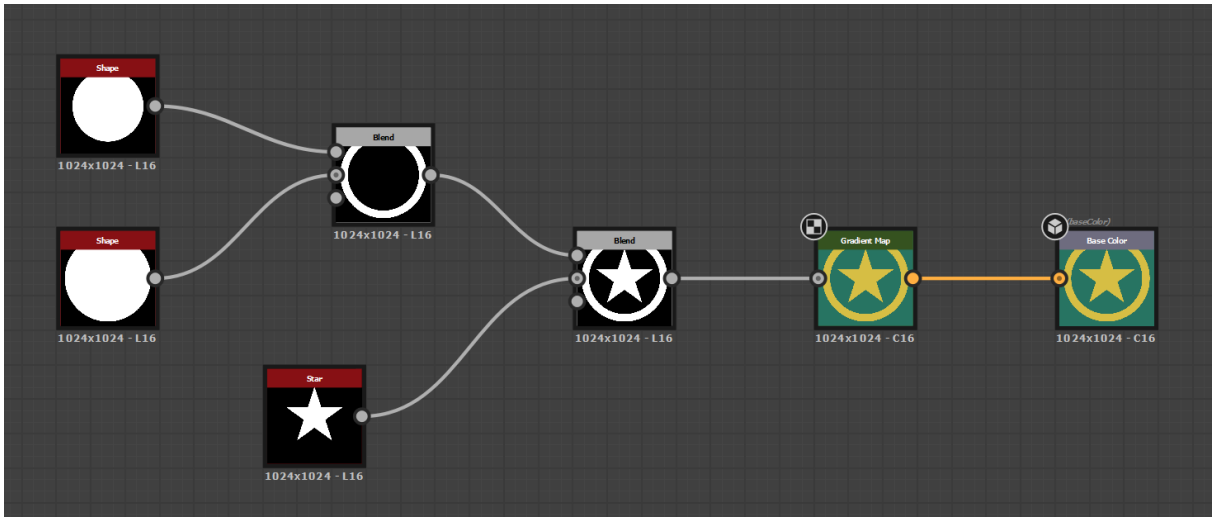
Mjesta na kojima se čvorovi mogu povezati predstavljena su malim kružićima. Ako je kružić s lijeve strane čvora, to znači da čvor može primati informaciju. Ako je kružić s desne strane čvora, to znači da čvor može proslijediti informaciju drugom.

Čvorovi koji imaju samo desne kružiće (veze) nazivaju se generatorima, jer stvaraju informacije (npr. osnovne oblike, šumove) – ne modificiraju informacije, već ih stvaraju.

Čvorovi koji imaju samo lijeve kružiće (veze) su izlazni čvorovi (eng. *outputs*), posebni čvorovi koji završavaju grafove.

Čvorovi koji imaju obje veze nazivaju se filteri, jer modificiraju postojeće informacije. Imaju kružiće lijevo i desno, za primanje i prosljeđivanje informacija, korisnik modificira informacije koristeći specifične parametre čvora i kasnije prosljeđuje izmijenjenu sliku izlaznom čvoru da završi graf ili drugom filteru da nastavi s varijacijama (Adobe, bez dat.-a; Sanchis Albert, 2019, str. 17–18).

Informacije o slici (teksturnoj mapi) koja se razvija putuju s lijeva na desno kroz građevne blokove odnosno čvorove povezane vezama koje određuju put informacija. Svaki čvor, ako je povezan, doprinijet će konačnim rezultatima (Ayyildiz, 2020, str. 56).



Slika 48: Primjer jednostavnog grafa na kojem je moguće vidjeti sve prethodno spomenute vrste čvorova (autorski rad)

7.1.2. Nelinearan softver

Nelinearan softver znači da kad artist izmijeni parametar nekog čvora, svi čvorovi koji su povezani s njim i primaju informacije od tog čvora, mijenjaju se u skladu s njim, sve do izlaznih čvorova. Ovaj domino efekt je fenomen rada s čvorovima – jedna mala promjena u čvoru može uvelike promijeniti konačni rezultat, pa čak i stvoriti materijale potpuno drugačijeg izgleda (Adobe, bez dat.-a; Sanchis Albert, 2019, str. 18–19).

7.1.3. Nedestruktivan softver

Sve učinjene promjene se mogu poništiti bez gubitaka u radu. Iako su svi čvorovi međusobno povezani, niti jedan od njih ne ovisi o drugom da bi funkcionirao, tako da artist može izbrisati ili zamijeniti skup čvorova s drugima – neće doći do rušenja programa (eng. *crash*) ili nekompatibilnosti (Adobe, bez dat.-a; Sanchis Albert, 2019, str. 19).

7.1.4. Proces izrade materijala

Prvi korak u procesu izrade materijala je razmisliti koji materijal iz stvarnog svijeta se želi simulirati, potražiti referentne fotografije ili ga proučiti na primjeru iz stvarnom svijeta. Treba dobro proučiti materijal i podijeliti ga na njegove osnovne oblike.

Sljedeći korak je odabrati željeni teksturni tijek rada i započeti izradu materijala u programu. Stvaranje materijala provodi se nizom čvorova koji se međusobno povezuju sve dok ne dođu do izlaznih čvorova (eng. *output nodes*). Izlazni čvorovi su konačni rezultati teksturnih mapa i oni služe za izvoz (eng. *export*) istih u nekom od formata slike za korištenje u drugim 3D aplikacijama (rendererima ili motorima za razvoj igara) (Sanchis Albert, 2019, str. 20).

7.1.5. Ugrađeni 3D objekti za testiranje materijala

U Substance 3D Designeru postoji više jednostavnih geometrijskih tijela na koje se može primijeniti materijal koji se razvija radi testiranja njegove kvalitete. To su kocka, cilindar, ravnina, sfera, torus, kocka sa zaobljenim rubovima itd. (Adobe, bez dat.-a).

Jedan od ugrađenih 3D objekata za testiranje materijala je sfera modelirana s puno utora, pukotina i rubova. Ova sfera je gotovo suprotna od kocke. Koristi se za procjenu drugog kraja spektra - detaljnija geometrija sa zaobljenim površinama. Često se koristi za razvoj materijala. Njen sferni oblik je idealan, jer artistu daje odličan osjećaj za refleksije. Detalji koje ima mogu se koristiti za razvoj slojevitih materijala (miješanje materijala), time pružajući podršku za testiranje realističnosti takvih materijala i njihovih interakcija (Parsons, 2017, str. 16–17).

Slika 49: Sfera s utorima i oštrim rubovima (autorski rad)

7.1.6. Substance 3D Designer i proceduralni razvoj tekstura

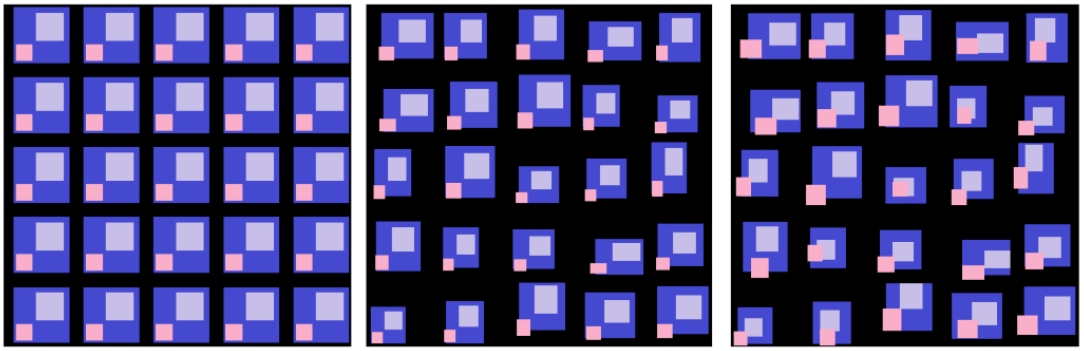
Substance 3D Designer pruža proceduralan način izrade tekstura, no same teksture nisu proceduralne. Iako ih artist dizajnira na proceduralan način, završene teksture odnosno krajnji rezultat su obične bitmape odnosno rasterizirane slike. Već je bilo govora o proceduralnim teksturama u njihovom poglavlju, kao i o njihovim nedostacima, od kojih je najkritičniji vrijeme potrebno da se generiraju.

Ako teksture nisu proceduralne, znači da su fiksne rezolucije odnosno da pokrivaju fiksnu površinu. Zato Substance 3D Designer pruža izradu popločavajućih tekstura (eng. *tiling textures*), no nedostatak toga su uzorci koji se ponavljaju. Što se tekstura više ponavlja po površini, oni će biti očitiji.

Iako je način izrade tekstura proceduralan (algoritmi definiraju izgled teksture), problem koji se pojavljuje u Substance 3D Designeru je taj što u se posljednjem koraku rezultat sprema u sliku, i u tom trenutku se gubi sva kontrola koju je artist imao tijekom dizajniranja teksture.

R. Moreira i S. Lages u svom dokumentu „*Organic Tile Textures using Fragment Shaders*“ predlažu zgodno rješenje kao alternativu popločavajućim teksturama čije ponavljanje uzoraka je neizbježno. Metoda predložena u njihovom dokumentu može zadržati jednu razinu kontrole nad teksturom čak i nakon što je artist gotov s njenom izradom, jer u toj metodi *shader* može interferirati u teksturu.

3D artist treba dizajnirati predložak jedne pločice (eng. *tile*). Taj predložak se sastoji od roditelja kojeg predstavlja cijela pločica, i djece koju predstavljaju elementi te pločice. Zatim artist definira parametre (položaj, veličina, rotacija, prozirnost itd.) nad roditeljem i djecom koje će *shader*, svaki put kad ponovi pločicu, promijeniti i na taj način napraviti varijaciju – svaka nova popločena pločica će biti drugačija od one prije. Ovom metodom *shader* može generirati beskonačne varijacije pločica.



Slika 50: Primjer kako *shader* upravlja varijacijom i roditelj-dijete odnosom. S lijeva na desno: nema varijacije; varijacija roditelja; varijacija roditelja i djece (R. Moreira & S. Lages, 2014, str. 1)

Shader u njihovom radu nije univerzalan, već je specifičan za svaki materijal. Broj osnovnih tekstura (djece) i njihovi parametri ne prosljeđuju se kao ulaz (eng. *input*) *shaderu*, već se upisuju u kod *shadera*. Dakle, za drugi materijal potrebno je napisati novi *shader*, specifičan za taj materijal (R. Moreira & S. Lages, 2014).

Rad autora R. Moreira i S. Lages predstavlja tehniku stvaranja velikih neperiodičnih tekstura. Umjesto definiranja jedne fiksne teksture koja se ponavlja, njihov rad predstavlja varirajuće teksture koje se generiraju u stvarnom vremenu, otvarajući nove mogućnosti za dizajn tekstura. Iako predstavljeni *shader* daje očekivane rezultate, na kraju dokumenta autori navode da je glavna briga budućeg rada poboljšanje performansi.

Moje mišljenje je, iako je ovo vrhunska ideja, vjerojatno još neko vrijeme neće zaživjeti, barem u igrama. Prvo i veliko, zato što generiranje teksture u stvarnom vremenu zapravo i nije u stvarnom vremenu odnosno ono traje i još uvijek treba optimizaciju.

Drugo, zato što je svakom novom materijalu potreban jedinstven *shader*, prilagođen samo tom materijalu, *shader* drugačiji od svih drugih – dakle ne može postojati jedan standardizirani opći *shader* koji bi se primjenjivao za svaki materijal. Netko tko zna dobro programirati i razumije programski kod *shadera* može sam isprogramirati jedan takav *shader*, ali za one koji to ne razumiju nije baš moguće ili lako napraviti softver koji će pomoći kreirati takve *shadere*. Drugim riječima, nije lako isprogramirati program koji će dinamički isprogramirati *shadere* za svaki materijal posebno.

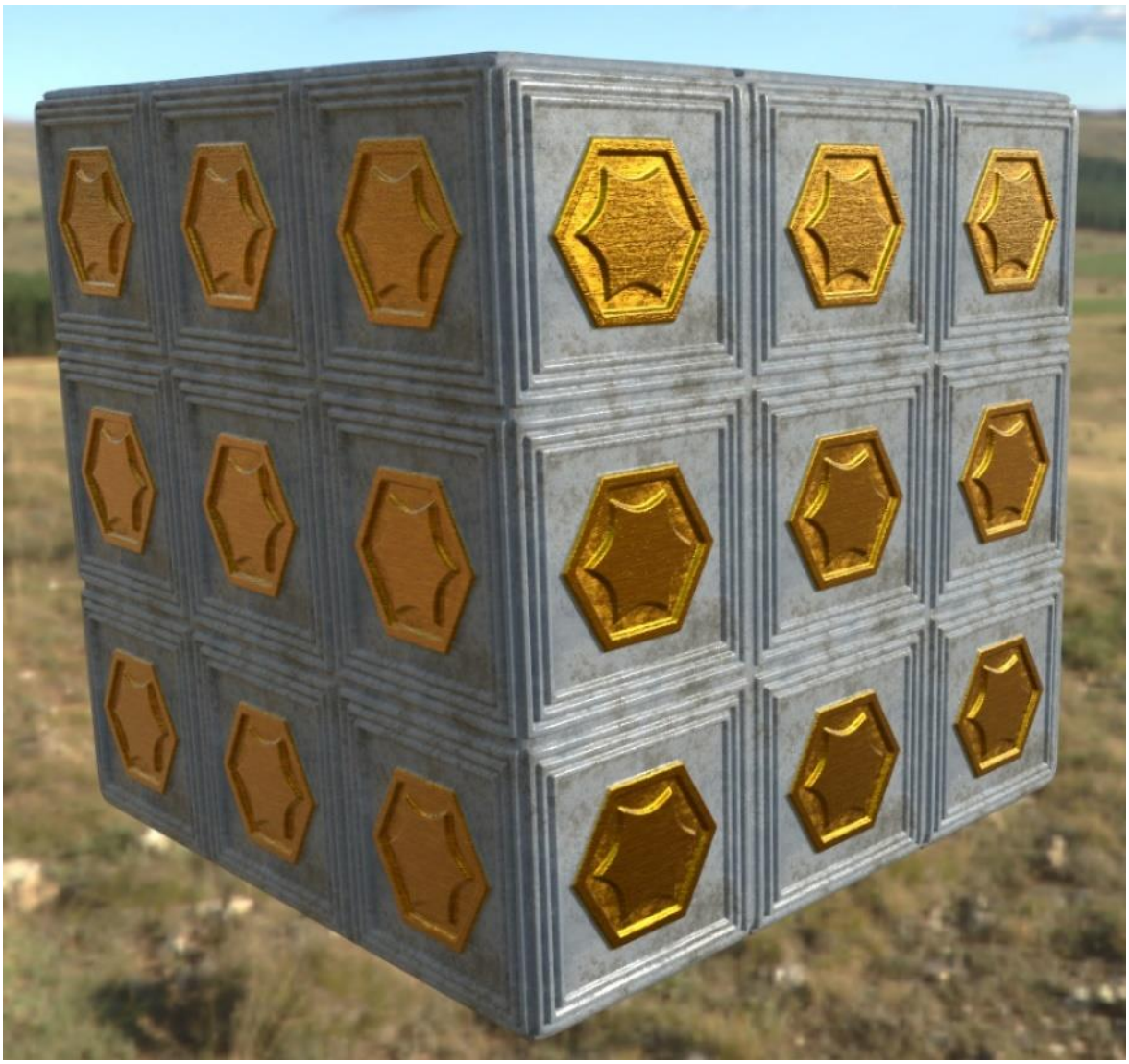
Kao zaključak svega navedenog, s obzirom da Substance 3D Designer zasad kao krajnji rezultat generira obične slike odnosno teksture fiksne rezolucije koje se popločavaju, bitno je dizajnirati teksture na način da budu prilično ujednačene po cijeloj svojoj površini odnosno da se ne napravi upečatljivi uzorak koji će još više istaknuti njihovu karakteristiku popločavanja odnosno ponavljanja.

7.2. Izrada tekstura u Substance 3D Designeru – praktičan primjer

Ne postoji jedan univerzalan vodič kako stvarati materijale u Substance 3D Designeru. Teksturiranje je umjetnost, stoga nije bitan način rada, bitan je rezultat – bitno je da materijal bude vizualno ugodan i privlačan.

Međutim, tijekom rada u Substance 3D Designeru preporučljivo je da tijek rada bude modularan (logičko odvajanje i grupiranje čvorova s obzirom na njihovu svrhu), pa tako čist i logičan. Treba imati na umu da najbolji način rada za jednog artista možda nije najbolji način i za druge (Jantunen, 2017, str. 14).

U ovom poglavlju je prikazan i objašnjen proces izrade popločavajućeg materijala vidljivog na slici 51.



Slika 51: Izgled materijala iz praktičnog primjera (autorski rad)

Ideja izrade ovog materijala je primijeniti najvažnije teksturne mape i uz pomoć primjera bolje razumjeti njihove uloge. Kao što je bilo moguće vidjeti na slici 51, radi se o materijalu koji ima metalnih (zvijezda u šesterokutu) i nemetalnih (sivi kvadrat) dijelova. Samim time, odmah se može zaključiti da je *metallic* mapa jedna od korištenih teksturnih mapa.

Prije početka rada, potrebno je dobro razmisliti kako postići svako pojedino svojstvo materijala odnosno kako napraviti svaku od teksturnih mapa. Za to je važno dobro razumjeti uloge teksturnih mapa iz poglavlja o teksturnim mapama, no i znati koje sve čvorove softver nudi za rad s teksturnim mapama (dokumentacija softvera). Ako jedno od spomenutog nije dovoljno dobro poznato, artist neće moći napraviti materijal dovoljno dobro, ili će biti ograničen svojim znanjem o softveru i izraditi materijal na teži način.

Korišteni teksturni tijek rada je PBR *metallic-roughness*, a izrađeni materijal se sastoji od šest teksturnih mapa: *base color*, *normal*, *metallic*, *roughness*, *height*, *ambient occlusion* (*AO*).

7.2.1. Kratki pregled procesa izrade materijala

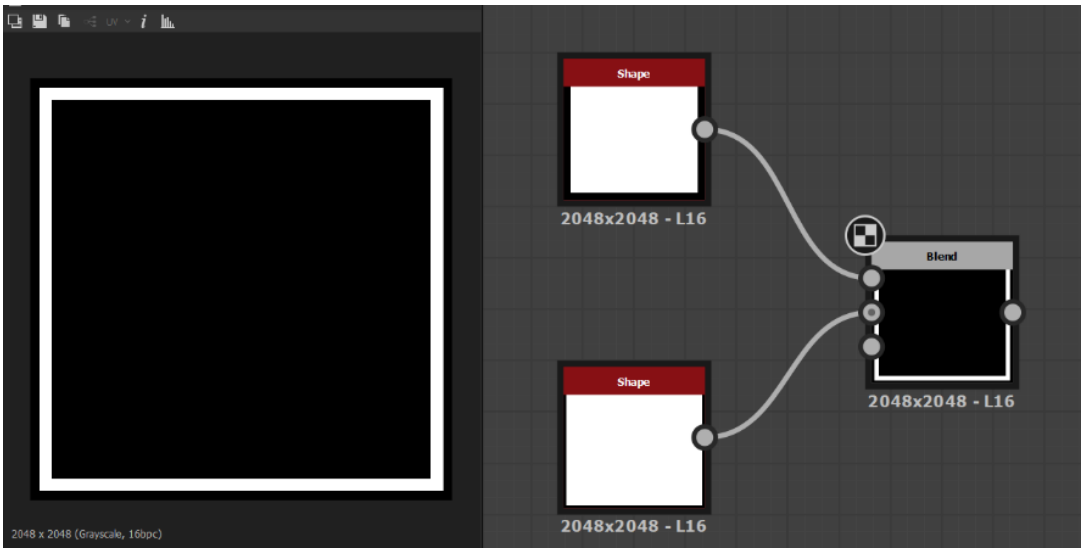
Izrada materijala započinje *height* mapom budući da ona daje osnovni trodimenzionalni oblik materijala po kojem se dalje mogu razvijati ostale mape. Odmah potom je, u nekoliko koraka, napravljena *ambient occlusion* mapa na temelju *height* mape, kojoj su potrebne informacije o konveksnim i konkavnim dijelovima geometrije iz *height* mape i koja koristi te informacije za sjenčanje konkavnih dijelova. Zatim je izrađena *roughness* mapa koja materijalu daje sjaj odnosno hrapavost – *roughness* mapa je napravljena tako da metalni dio bude sjajniji od nemetalnih dijelova. *Metallic* mapa je korištena za definiranje metalnih (zvijezda u šesterokutu) i nemetalnih (sivi kvadrat) dijelova – važno rendereru za ispravno renderiranje metala odnosno nemetala. Zadnje dvije izrađene mape su *base color* mapa i *normal* mapa. *Base color* mapa daje bolju materijalu. Jedna je od zadnjih mapa u procesu izrade materijala, jer bi u suprotnom bilo teško uočiti učinke drugih teksturnih mapa na površini ako ona već ima boju. Zadnja izrađena mapa je *normal* mapa koja materijalu daje sitne površinske detalje poput ogrebotina, minijaturnih neravnina i udubljenja. Zadnja je mapa u procesu izrade materijala, jer ideja je da se materijal razvija od većih i općih oblika do manjih i sitnih detalja.

7.2.2. Izrada *height* mape

Izrada materijala započinje *height* mapom budući da ona daje osnovni trodimenzionalni oblik materijala po kojem se dalje mogu razvijati ostale mape.

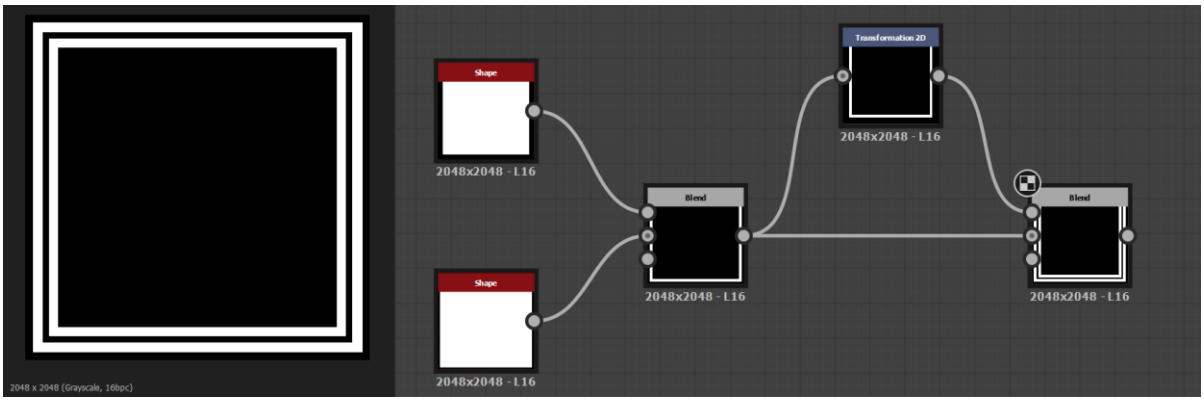
Kao što je bilo moguće vidjeti na slici 51, materijal ima konveksne (ispupčene) dijelove na rubovima i u sredini. Prvo je prikazan proces izrade konveksnih rubova.

U programu se uzme čvor *Shape* i u njemu se odabere bijeli kvadrat. Još jednom se uzme isti čvor i malo mu se smanji veličina. Sad postoje dva bijela kvadrata različitih veličina. Uzme se čvor *Blend* i pomiješaju se dva kvadrata na način da se od većeg oduzme manji (operacija *substract* u *Blend* čvoru). To će rezultirati rupom na većem kvadratu, odnosno vidjet će se samo njegov rub (slika 52).



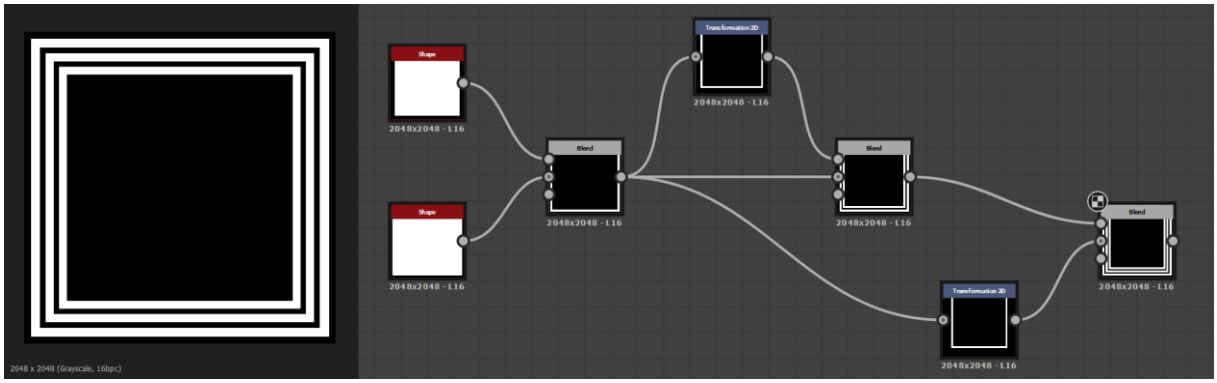
Slika 52: Koraci (desno) i rezultat (lijevo) (autorski rad)

Sada treba napraviti takve rubove prema unutra još nekoliko puta. Međutim, sad kad već postoji jedan takav rub, nije potrebno od početka raditi druge. Može se uzeti čvor *Transformation 2D* koji će kopirati dosad dobiveni rezultat (rub) i onda toj kopiji samo malo smanjiti veličinu da kopirani rub bude unutar većeg ruba. Tim čvorom je dobiven još jedan, manji rub. Sad treba taj rub pridružiti onom većem tako da na jednoj teksturnoj mapi budu dva ruba, jedan unutar drugog. To se radi opet s *Blend* čvorom, ali ovaj put operacijom *add* koja zbraja bijele tonove dvaju čvorova (slika 53).



Slika 53: Koraci (desno) i novi rezultat (lijevo) (autorski rad)

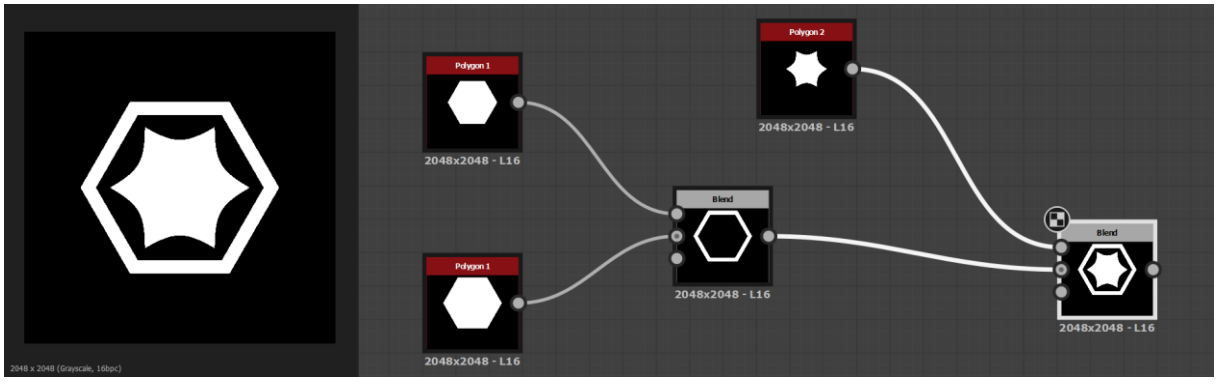
Još jedan (zadnji) put se dodaje novi rub unutar postojećih, na isti način – korištenjem *Transformation 2D* čvora za kopiju i smanjenje postojećeg ruba, i *Blend* čvora za pridruživanje novog ruba već postojećim rubovima (slika 54).



Slika 54: Koraci (desno) i novi rezultat (lijevo) (autorski rad)

Na sličan način napravljena je sredina *height* teksturne mape – rub šesterokuta i šesterokraka zvijezda.

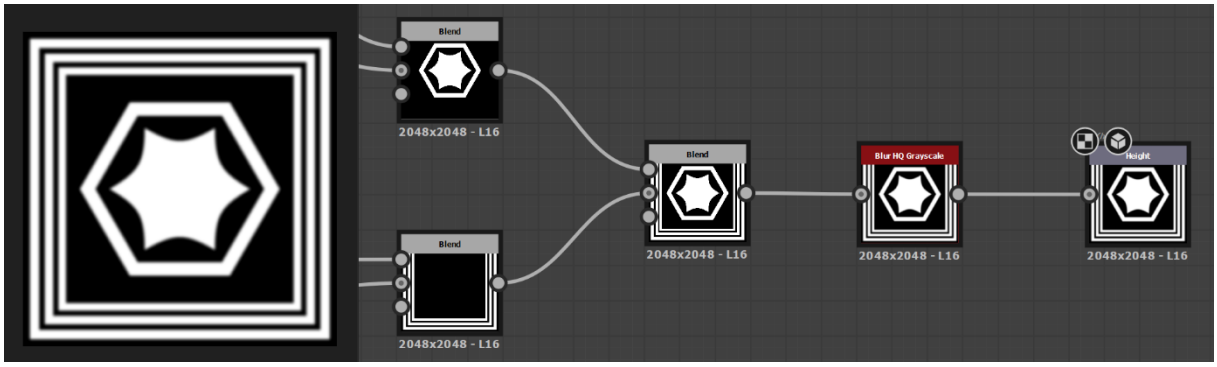
U programu se uzme *Polygon 1* čvor i u njemu se namjesti da poligon ima šest strana (da bude šesterokut). Duplicira (kopira) se čvor i dupliciranom čvoru se malo smanji veličina šesterokuta. Zatim se oba čvora spoje u jedan *Blend* čvor na način da se od većeg šesterokuta oduzme manji (*substract* operacija u *Blend* čvoru) i tako ostane samo rub većeg šesterokuta. Za šesterokraku zvijezdu se može uzeti *Polygon 2* čvor (napredniji čvor od *Polygon 1* čvora) i namjestiti da poligon ima šest strana (šesterokut), ali također namjestiti da su mu rubovi zakrivljeni prema unutra tako da izgleda kao zaobljena šesterokraka zvijezda. Na kraju treba samo spojiti rub šesterokuta i šesterokraku zvijezdu u novom *Blend* čvoru (slika 55).



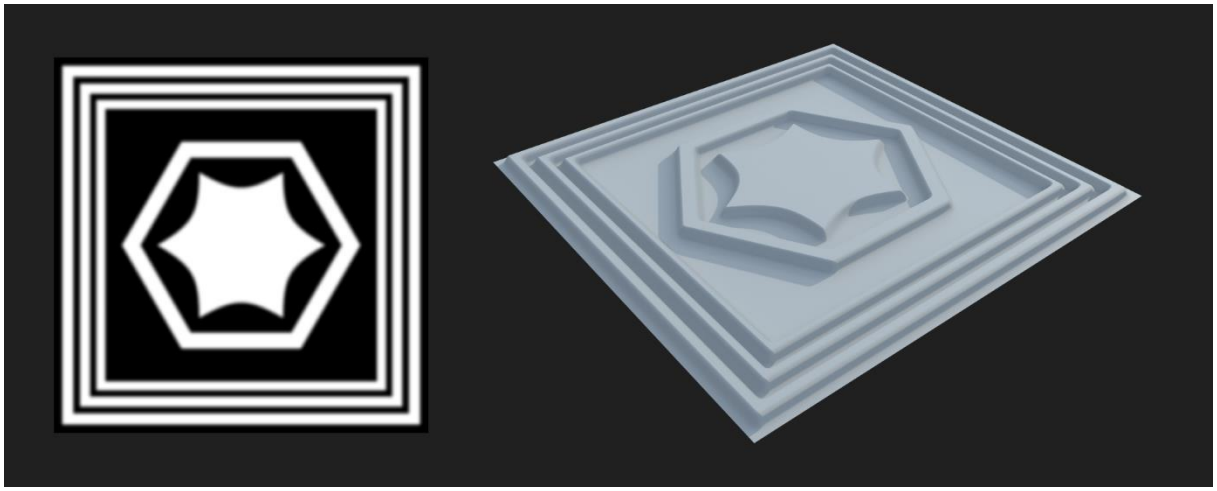
Slika 55: Koraci (desno) i rezultat (lijevo) (autorski rad)

U ovom trenutku su gotovi rubovi i sredina *height* mape, ali su u zasebnim čvorovima. Sad ih treba spojiti u jednu sliku preko *Blend* čvora. Jednom kad su spojeni, može se reći da je *height* mapa završena. Međutim, budući u dobivenoj teksturnoj mapi nema glatkog prijelaza između crne i bijele boje, to bi rezultiralo oštrim rubovima modificirane površine. Zato je dodan

još samo jedan čvor, *Blur HQ Grayscale*, koji malo zamućuje cijelu sliku tako da prijelaz između nižih i viših dijelova geometrije bude blaži. Sad je *height* mapa 100 % završena, i *Blur HQ Grayscale* čvor se može spojiti s izlaznim čvorom za *height* mapu (slika 56).



Slika 56: Kraj razvoja *height* mape; koraci (desno) i konačan rezultat (lijevo) (autorski rad)



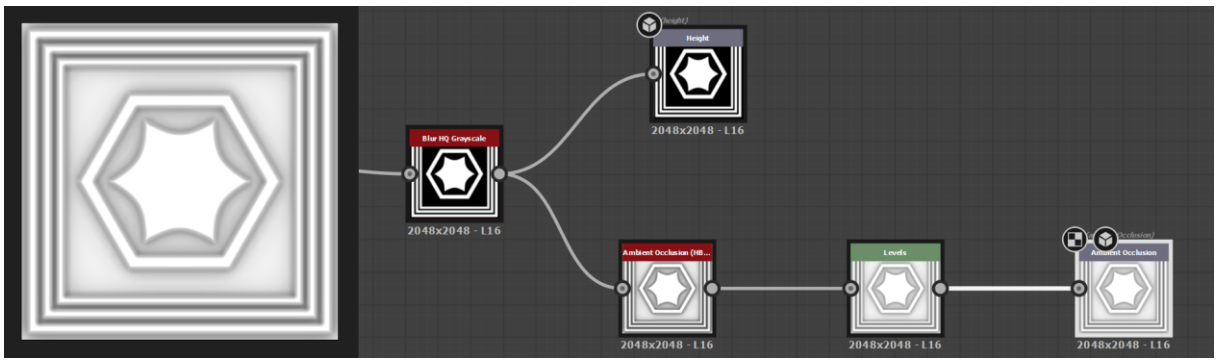
Slika 57: *Height* mapa (lijevo) i njena manifestacija na površini (desno) (autorski rad)

7.2.3. Izrada *ambient occlusion* mape

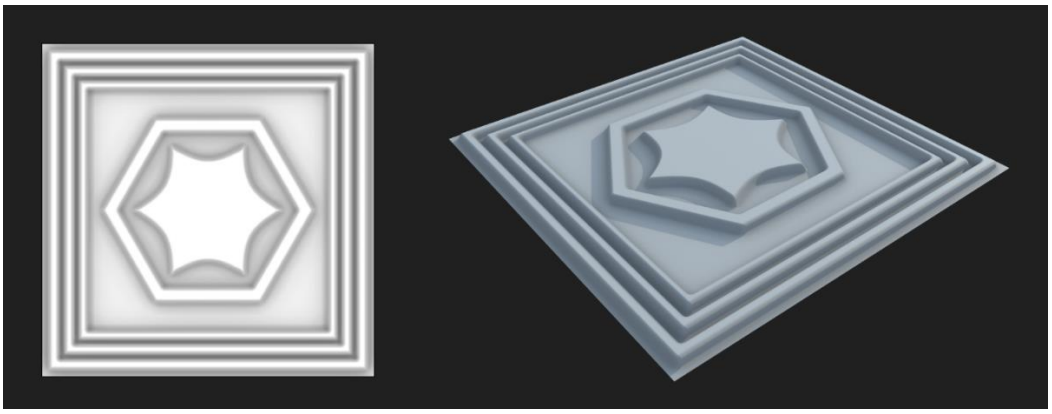
Svrha *ambient occlusion* mape je da malo zasjeni konkavne dijelove geometrije radi veće detaljnosti i realizma. Izrada ove mape je vrlo jednostavna, jer postoji čvor *Ambient Occlusion (HBAO)* koji na temelju *height* mape odmah generira *ambient occlusion* mapu – *height* mapa je potrebna upravo zbog informacija o konveksnim i konkavnim dijelovima geometrije, što pomaže *Ambient Occlusion (HBAO)* čvoru da potamni konkavna područja manje izložena okolini.

U programu se kreira *Ambient Occlusion (HBAO)* čvor i, budući da već postoji gotova *height* mapa, spoji se njen čvor kao ulaz u *Ambient Occlusion (HBAO)* čvor. Ovaj čvor je sasvim dovoljan za *ambient occlusion* mapu, no dodan mu je još samo jedan čvor, *Levels*, kojim su malo podignuti tamni tonovi tako da *ambient occlusion* ne bude previše intenzivan,

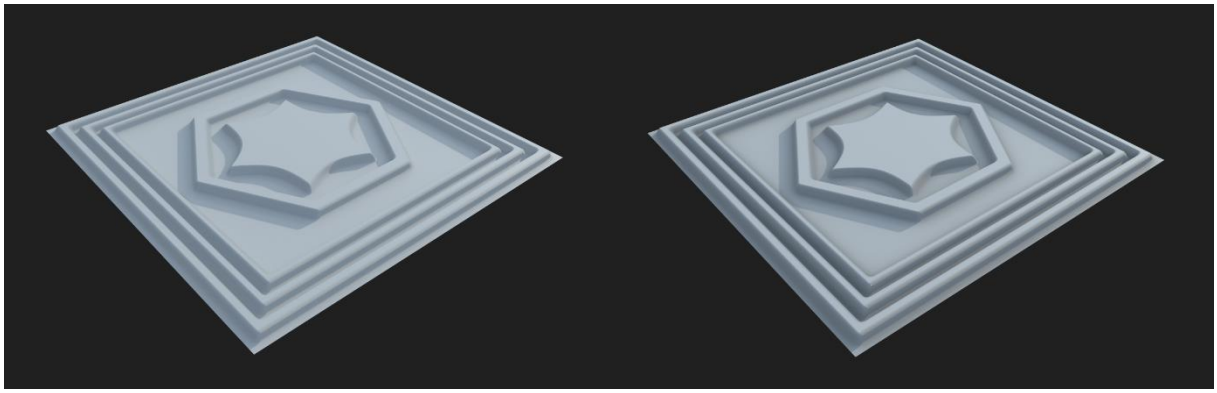
već vrlo suptilan. Time je *ambient occlusion* mapa završena i sad se čvor *Levels* može povezati s izlaznim čvorom za *ambient occlusion* mapu (slika 58).



Slika 58: Koraci (desno) i rezultat (lijevo) (autorski rad)



Slika 59: *Ambient occlusion* mapa (lijevo), izgled površine s *ambient occlusion* i *height* mapama zajedno (desno) (autorski rad)

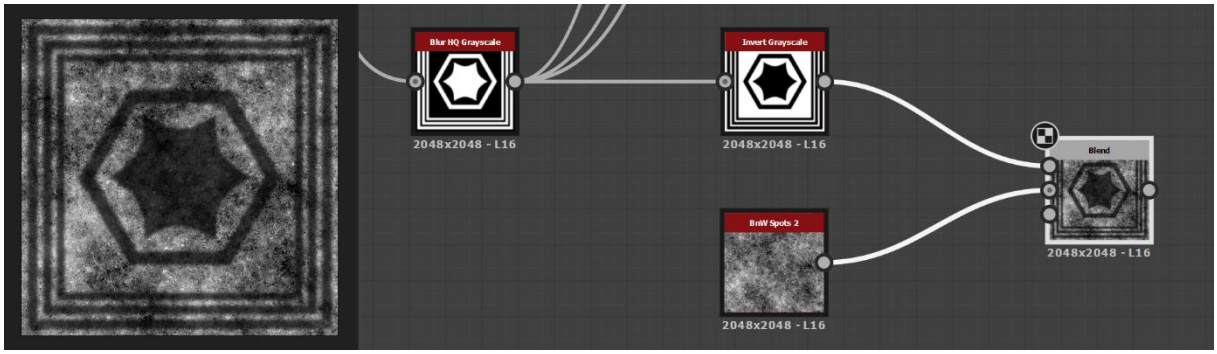


Slika 60: Izgled površine samo s *height* mapom (lijevo), izgled površine s *height* i *ambient occlusion* mapama (desno) (autorski rad)

7.2.4. Izrada *roughness* mape

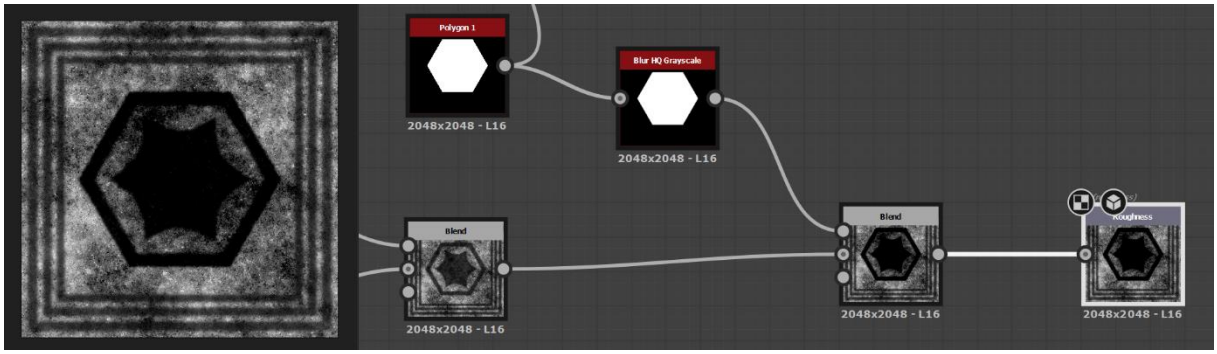
Roughness mapa definira koliko je površina hrapava odnosno glatka. Tamniji tonovi označuju manju hrapavost (veći sjaj), a svjetliji veću hrapavost (manji sjaj).

Kako površina ne bi imala jednu uniformnu vrijednost hrapavosti po cijeloj površini, što izgleda monotono, korišten je *BnW Spots 2* čvor (skraćeno od *black and white spots*) koji daje zanimljivu distribuciju crnih i bijelih mrlja. No, također ne izgleda realistično kad je cijela površina uniformno pokrivena tim mrljama. Zato je napravljeno da konveksni dijelovi površine budu sjajniji od onih konkavnih koji su manje izloženi okolini i koji su podložni nakupljanju prljavštine. Za to je korišten čvor *height* mape od prije. Njemu je dodan čvor *Invert Grayscale* koji invertira *height* mapu – bijeli (ispupčeni) dijelovi postaju crni, a crni (udubljeni) dijelovi postaju bijeli. Sad se *Invert Grayscale* čvor spoji s *BnW Spots 2* čvorom u *Blend* čvoru s operacijom *multiply*. Ta operacija rezultira tako da *Invert Grayscale* čvor potamni sliku od *BnW Spots 2* čvora na mjestima gdje su crne vrijednosti u *Invert Grayscale* čvoru (slika 61).

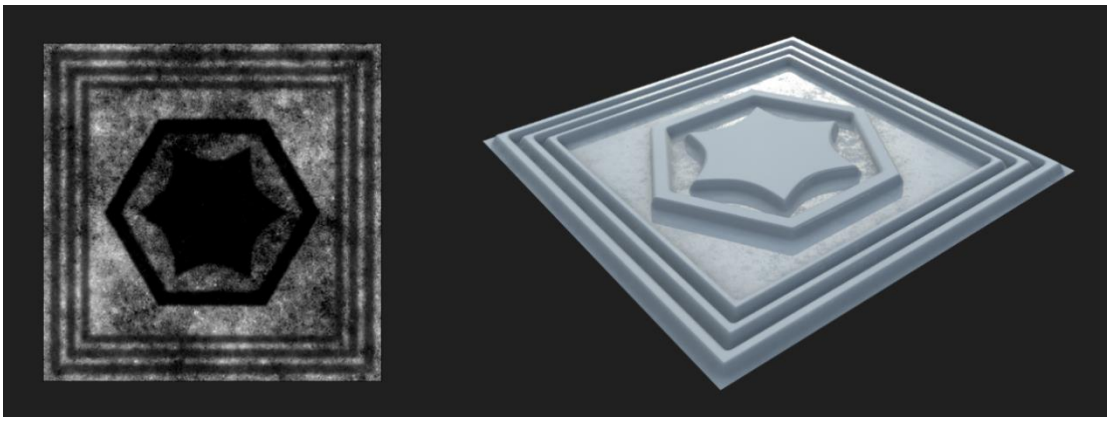


Slika 61: Koraci (desno) i rezultat (lijevo) (autorski rad)

Važno je razumjeti što se ovime postiglo. Potamnjeni dijelovi (konveksni dijelovi geometrije) znači da će biti manje hrapavi odnosno više će se sjajiti. Još za kraj ove mape, napravljeno je da metalni dio, koji je u sredini, bude dodatno sjajniji. Za to se može iskoristiti čvor šesterokuta napravljen od prije. Doda mu se *Blur HQ Grayscale* čvor tako da ga malo zamuti odnosno da ne bude oštar prijelaz između bijele i crne boje. Sada taj čvor se spoji s dosad dobivenim rezultatom u novom *Blend* čvoru s operacijom *substract*. Na taj način bijeli šesterokut potamnjuje rezultat od prije. Sad je *roughness* mapa završena i može se spojiti s izlaznim čvorom za *roughness* mapu (slika 62).



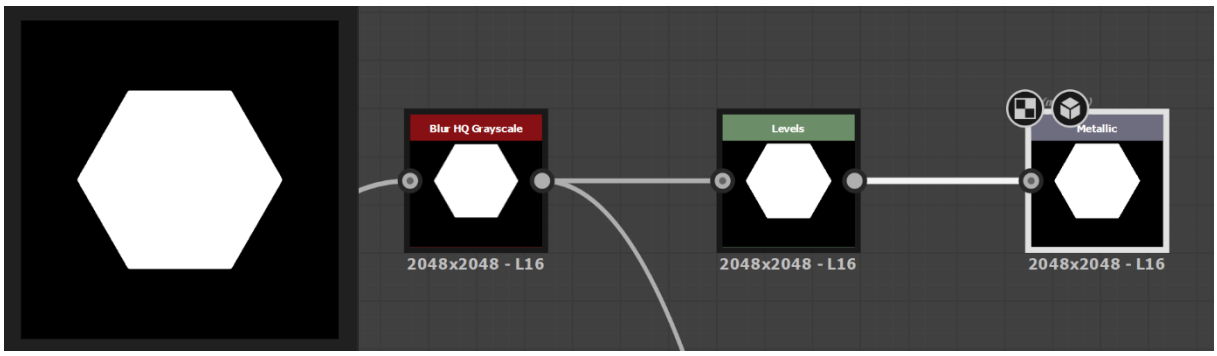
Slika 62: Kraj razvoja *roughness* mape; koraci (desno) i rezultat (lijevo) (autorski rad)



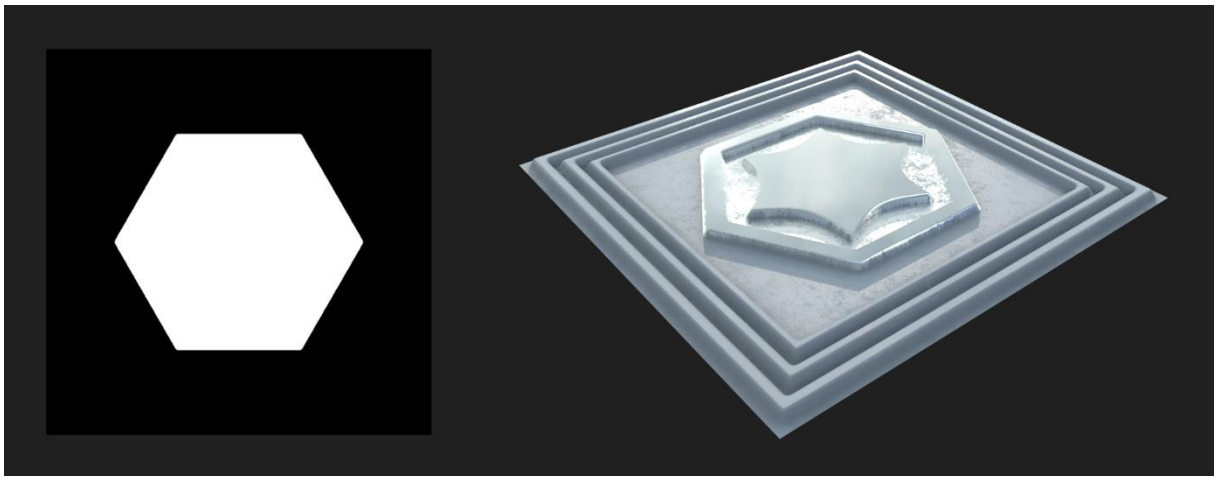
Slika 63: *Roughness* mapa (lijevo); izgled površine s *roughness* mapom zajedno s *height* i *ambient occlusion* mapama (desno) (autorski rad)

7.2.5. Izrada *metallic* mape

Metallic mapa definira metal na bijelim odnosno nemetal na crnim dijelovima mape. Ovu mapu u ovom praktičnom primjeru je vrlo jednostavno napraviti. Metalni dio površine je cijeli šesterokut u sredini, što znači da se može iskoristiti čvor sa šesterokutom od prije, i to njegova zamućena verzija (*Blur HQ Grayscale*). Budući da *Blur HQ Grayscale* čvor od prije je zamutio sliku šesterokuta više nego potrebno za ovu mapu (ipak je poželjno da prijelaz između metala i nemetala bude oštriji) iskorišten je na njemu *Levels* čvor s kojim crni i bijeli tonovi postaju malo suzbijeni tako da prijelaz iz bijele u crnu bude malo oštriji (i dalje ostaje zamućen, ali manje). I to je sve, sada se *Levels* čvor može povezati s izlaznim čvorom za *metallic* mapu (slika 64).



Slika 64: Koraci (desno) i rezultat (lijevo) (autorski rad)

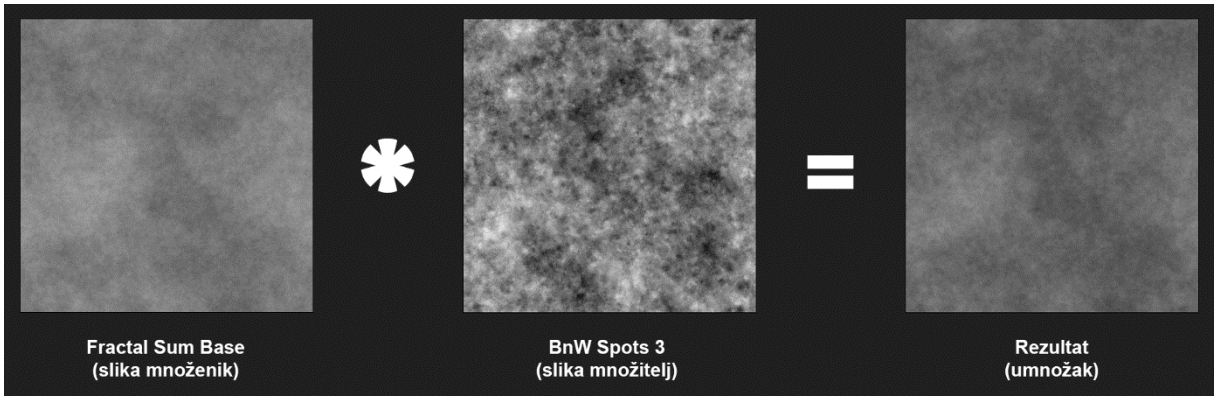


Slika 65: *Metallic* mapa (lijevo); izgled površine s *metallic* mapom zajedno s *height*, *ambient occlusion* i *roughness* mapama (autorski rad)

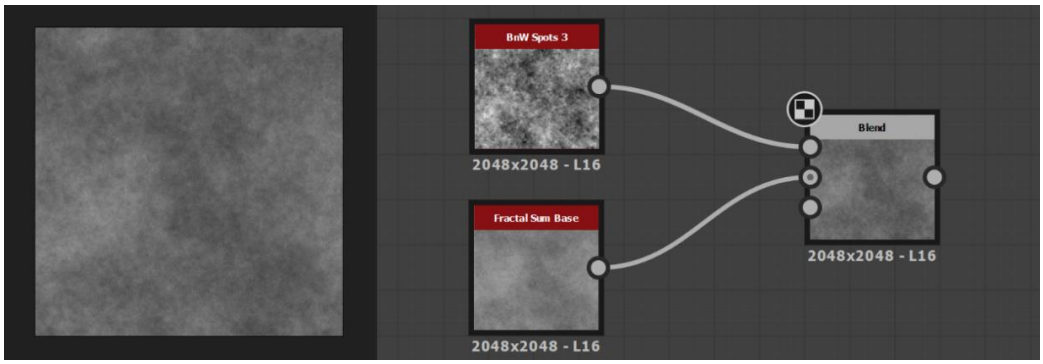
7.2.6. Izrada *base color* mape

Base color mapa je grubo rečeno mapa koja definira boju površine. Izrada ove mape podijeljena je na dva dijela: boja nemetalnog dijela koji je zamišljen kao površina od kamena, i boja metalnog dijela koji je zamišljen kao površina od zlata. Prvo je napravljen nemetalni dio – površina od kamena.

Za izradu zanimljive teksture kamena mogu se iskoristiti čvorovi *Fractal Sum Base* i *BnW Spots 3*. Za kombinaciju tih dviju tekstura, njihovi čvorovi se mogu staviti u *Blend* čvor s malim postotkom operacije *multiply*. *Multiply* je operacija kojom slika množitelj potamnjuje sliku množenik na mjestima gdje slika množitelj ima tamne tonove (slika 66, slika 67).

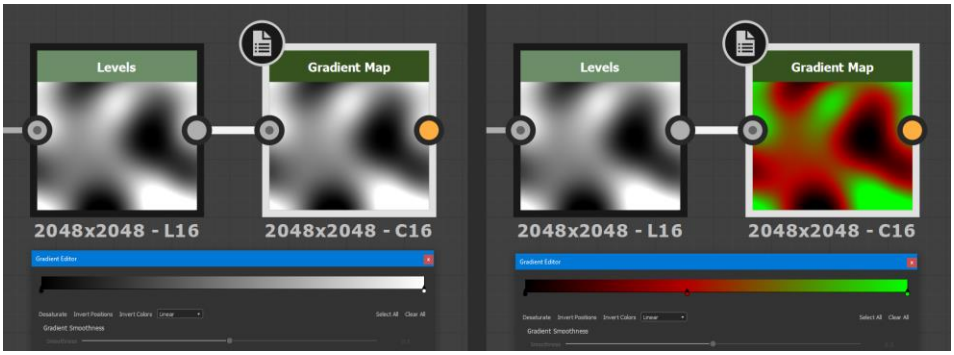


Slika 66: Miješanje dviju slika u *Blend* čvoru preko operacije *multiply* (autorski rad)



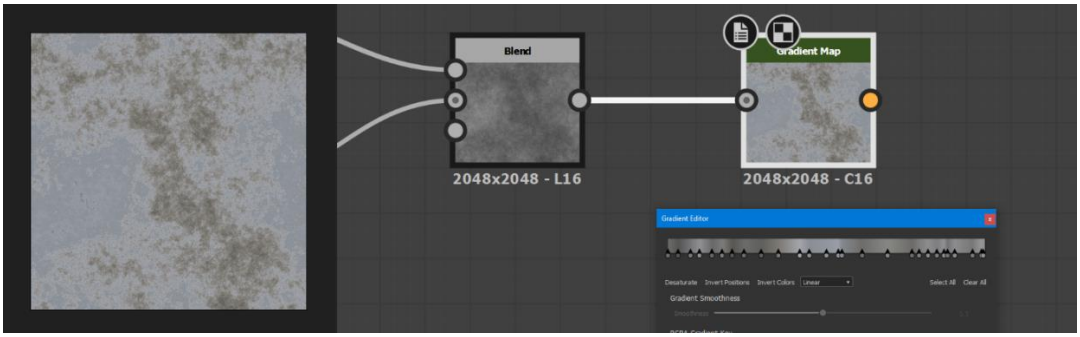
Slika 67: Koraci (desno) i rezultat (lijevo) (autorski rad)

To bi sad mogla biti tekstura kamena, međutim ona nema boju, ona je u sivim tonovima (eng. *grayscale*). Kako bi joj se dodijelila boja može se iskoristiti *Gradient Map* čvor. *Gradient Map* čvor služi za mapiranje slike u sivim tonovima na prilagođenu rampu boja (slika 68).



Slika 68: Primjer korištenja *Gradient Map* čvora za mapiranje slike u sivim tonovima na prilagođenu rampu boja (autorski rad)

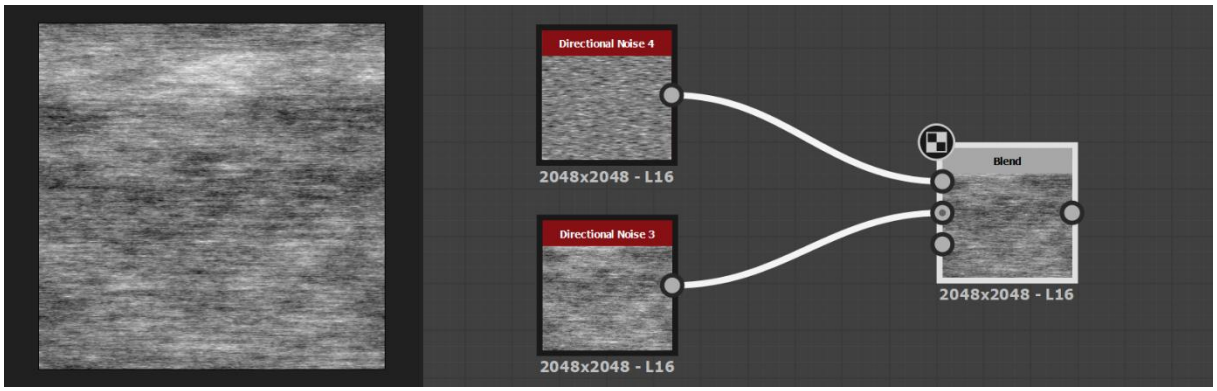
Dakle, spoji se čvor s teksturom kamena na *Gradient Map* čvor i teksturi u sivim tonovima se dodjeli boja. Rampa boja se može definirati ručno, međutim postoji opcija „*Pick Gradient*“ koja povlačenjem linije kursorom preko neke slike/fotografije automatski kreira rampu boja na temelju povučene linije. Za kreiranje rampe boja korištena je upravo ta opcija. Može se iskoristiti bilo koja fotografija s kamenom čija tekstura zadovoljava traženu paletu boja (slika 69).



Slika 69: Koraci (desno) i rezultat (lijevo) (autorski rad)

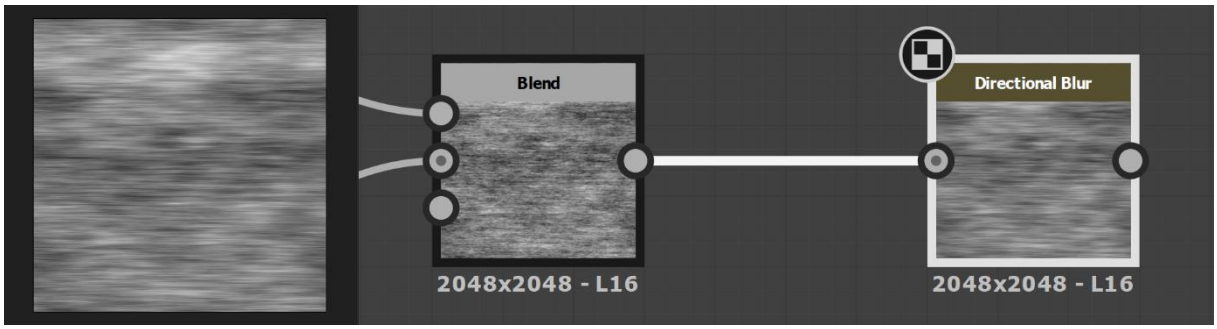
Sad je boja kamene površine završena, no sad još treba napraviti boju za metalni dio. Ona se radi na isti način.

Metalni dio je zamišljen kao uglađeni metal pa se za njega mogu iskoristiti čvorovi *Directional Noise 3 i Directional Noise 4* čije slike izgledaju kao usmjerene linije. Oni se mogu pomiješati u *Blend* čvoru na isti način, preko *multiply* operacije (slika 70).



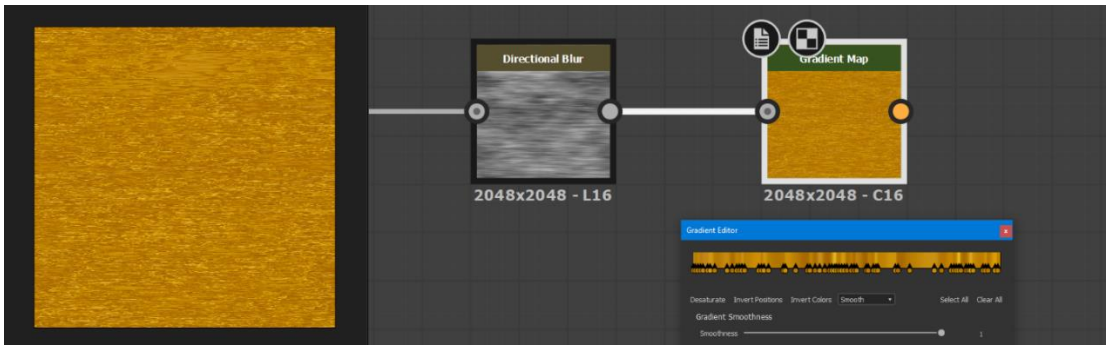
Slika 70: Koraci (desno) i rezultat (lijevo) (autorski rad)

Dobivena slika odnosno usmjerene linije bi se mogle još dodatno rastegnuti na način da se zadrži isti smjer, tako da slika izgleda još više uglađena. To se može napraviti s *Directional Blur* čvorom koji radi zamućenje slike u samo jednom, korisnički definiranom smjeru. Odabrani smjer zamućenja je isti kao i smjer linija (slika 71).



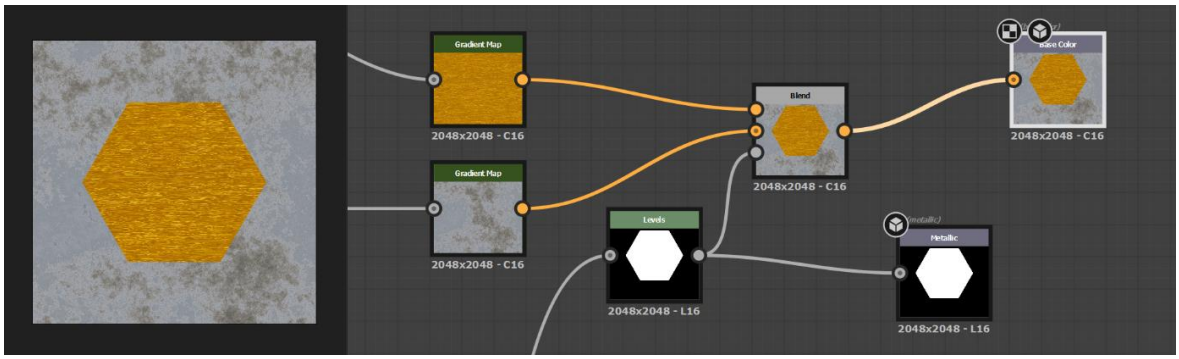
Slika 71: Koraci (desno) i rezultat (lijevo) (autorski rad)

Sad je dobivena tražena tekstura metalne površine, samo još treba na nju mapirati boju. Boja je mapirana na isti način kao što je bila mapirana boja za kamenu površinu – preko *Gradient Map* čvora. Odabere se „*Pick Gradient*“ opcija i kursorom povuće linija preko neke fotografije sa zlatom (slika 72).

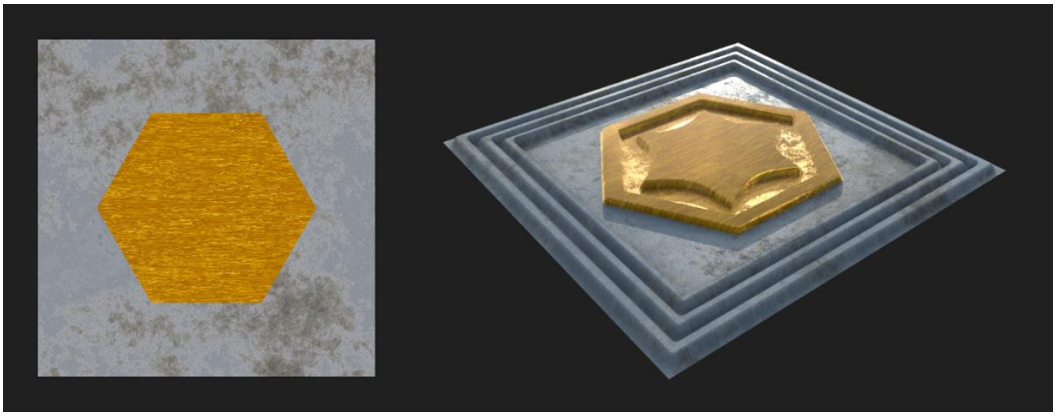


Slika 72: Koraci (desno) i rezultat (lijevo) (autorski rad)

Sad postoje dvije zasebne teksture u boji: jedna za kamenu površinu, druga za metalnu površinu. Treba ih spojiti u jednu teksturu koja će teksturu kamene površine prikazati na nemetalnom dijelu, a teksturu zlata prikazati na metalnom dijelu. Za to se može iskoristiti čvor s *metallic* mapom od prije, koji će ovdje poslužiti kao maska. Dakle, spoje se teksture kamena i zlata s novim *Blend* čvorom, ali ovaj put s *Blend* čvorom se poveže i *metallic* mapa na mjesto maske. To znači da će bijela boja iz *metallic* mape prikazivati jednu teksturu (teksturu zlata), a crna boja drugu teksturu (teksturu kamena). Ovime je *base color* mapa završena i može se povezati s izlaznim čvorom za *base color* mapu (slika 73).



Slika 73: Kraj razvoja *base color* mape; koraci (desno) i rezultat (lijevo) (autorski rad)

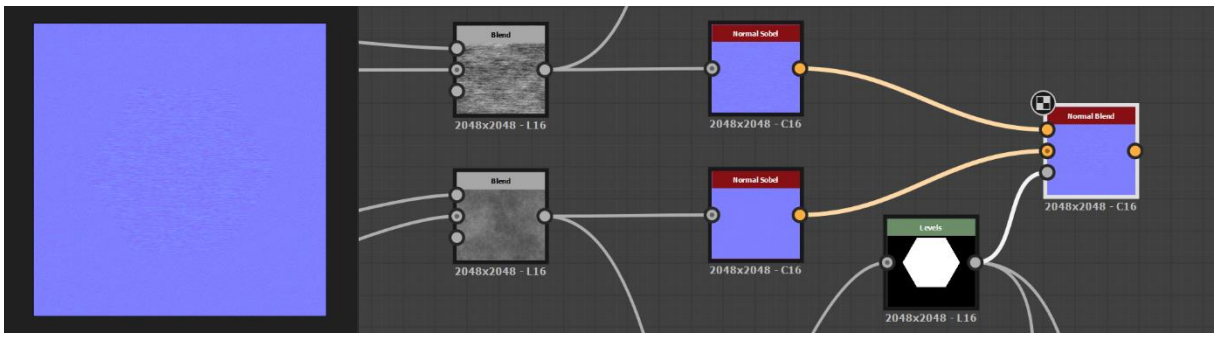


Slika 74: *Base color* mapa (lijevo); izgled površine s *base color* mapom zajedno s *height*, *ambient occlusion*, *roughness* i *metallic* mapama (autorski rad)

7.2.7. Izrada *normal* mape

Normal mapa je korištena za izradu sitnih površinskih detalja. Ova mapa ne modificira stvarnu površinu, već stvara iluziju da površina ima više geometrije. *Normal* mapa, kao i *base color* mapa, napravljena je na dva dijela – jedan dio za kamenu površinu, drugi dio za metalnu površinu, pa se oni kasnije spoje u jednu teksturnu mapu.

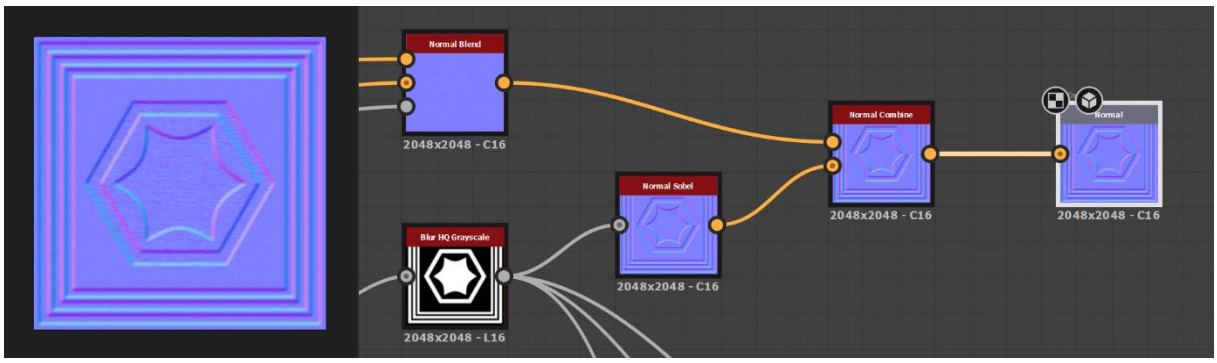
Za *normal* mapu kamene površine može se iskoristiti čvor koji drži teksturu kamene površine od prije, koja je u sivim tonovima. *Normal* mapa u Substance 3D Designeru se kreira na temelju mape u sivim tonovima (eng. *grayscale*) koja djeluje kao *height* mapa – svjetliji tonovi predstavljaju ispupčenja, tamniji udubljenja. Dakle, čvor s teksturom kamene površine u sivim tonovima od prije se poveže s *Normal Sobel* čvorom koji na temelju mape u sivim tonovima generira *normal* mapu. Identičan postupak se napravi za kreiranje *normal* mape za metalnu površinu – čvor s teksturom metalne površine u sivim tonovima od prije se poveže s *Normal Sobel* čvorom. Sad postoje dvije zasebne *normal* mape, jedna za kamenu površinu, druga za metalnu površinu. Sad ih treba spojiti u jedan *Normal Blend* čvor (isto kao *Blend* čvor, ali namijenjen samo *normal* mapama), ali na način da se na metalnom dijelu prikaže *normal* mapa metalne površine, a na nemetalnom dijelu *normal* mapa kamene površine. To se postiže preko maske na isti način kao što je to bio slučaj kod *base color* mape. Dakle, u *Normal Blend* čvor na mjesto maske se stavi *metallic* mapa od prije koja će na bijelim mjestima prikazati prvu *normal* mapu (*normal* mapu metalne površine), a na crnim mjestima prikazati drugu *normal* mapu (*normal* mapu kamene površine) (slika 75).



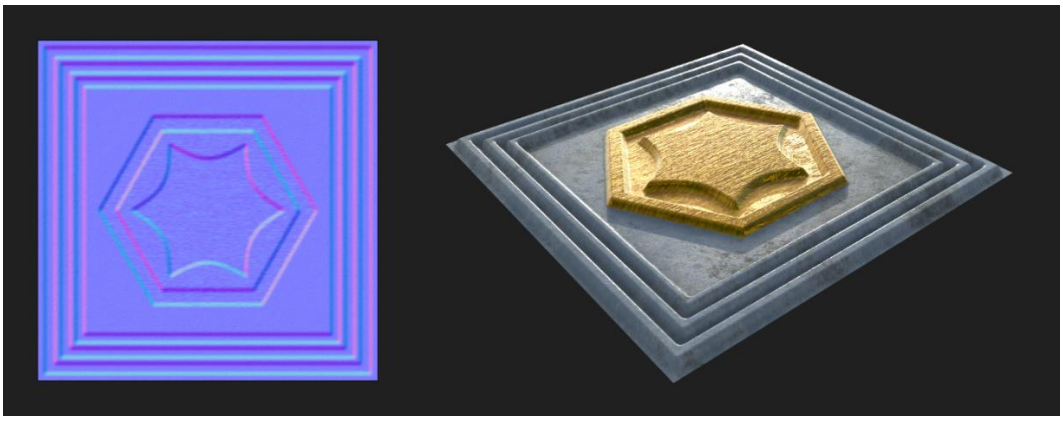
Slika 75: Koraci (desno) i rezultat (lijevo) (autorski rad)

Za kraj, može se još napraviti nova *normal* mapa na temelju *height* mape od prije i spojiti s trenutnom *normal* mapom. Ovim će se dodatno naglasiti trodimenzionalnost površine koju je napravila *height* mapa, ali će i dobro poslužiti u slučaju kad se *height* mapa ne bi koristila – bez *height* mape površina nema stvarnu trodimenzionalnost, pa bi u tom slučaju *normal* mapa simulirala trodimenzionalnost koju bi inače generirala *height* mapa. Nova *normal* mapa se sada kombinira s trenutnom preko *Normal Combine* čvora. Za razliku od *Normal Blend* čvora koji miješa dvije *normal* mape na temelju maske, *Normal Combine* čvor ih kombinira

odnosno združi (nadoda jednu *normal* mapu povrh druge). Sad je *normal* mapa završena i može se povezati s izlaznim čvorom za *normal* mapu (slika 76).



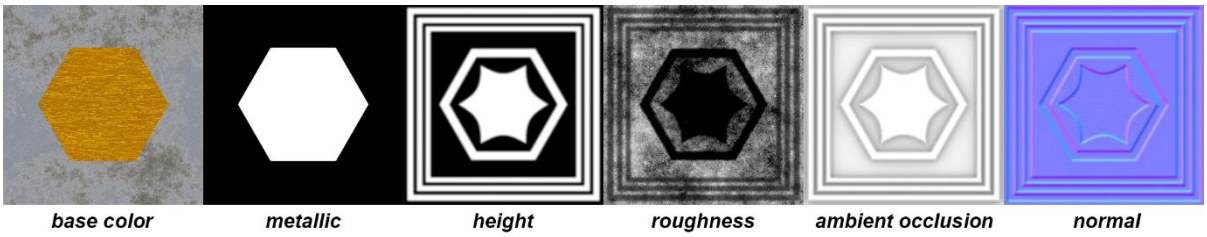
Slika 76: Kraj razvoja *normal* mape; koraci (desno) i rezultat (lijevo) (autorski rad)



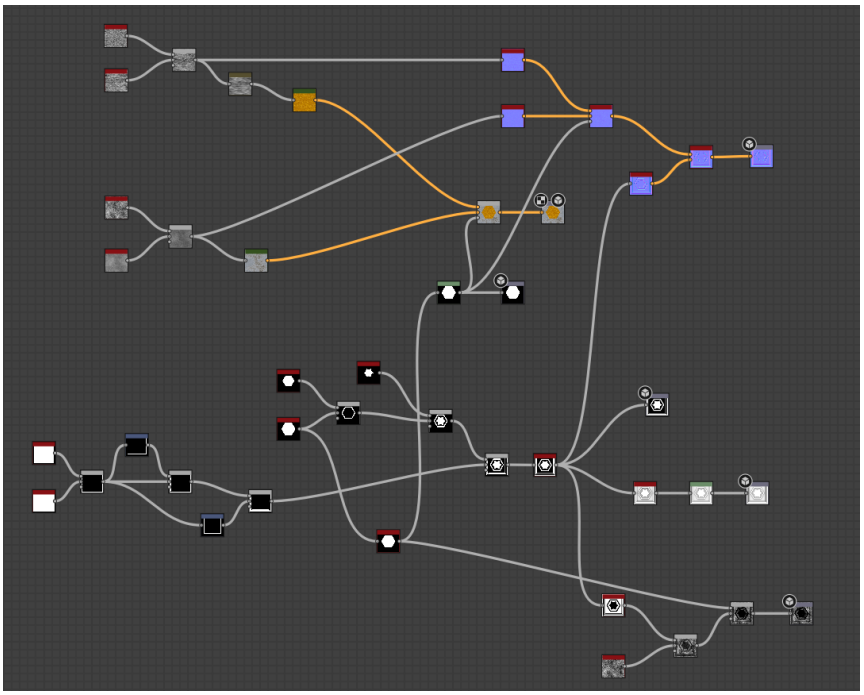
Slika 77: *Normal* mapa (lijevo); izgled površine s *normal* mapom zajedno s *height*, *ambient occlusion*, *roughness*, *metallic* i *base color* mapama (autorski rad)

7.2.8. Kraj izrade materijala

Sad je izrada materijala iz praktičnog primjera završena. Materijal se sastoji od šest teksturnih mapa: *base color*, *metallic*, *height, roughness*, *ambient occlusion* i *normal*. Kombinacijom spomenutih teksturnih mapa materijal izgleda potpun – ima sva najvažnija površinska svojstva materijala iz stvarnog svijeta. Ima boju, dubinu, sjaj, detalje itd. Izrađeni materijal je popločavajući materijal, što znači da se može beskonačno popločavati po površini i nema granice.



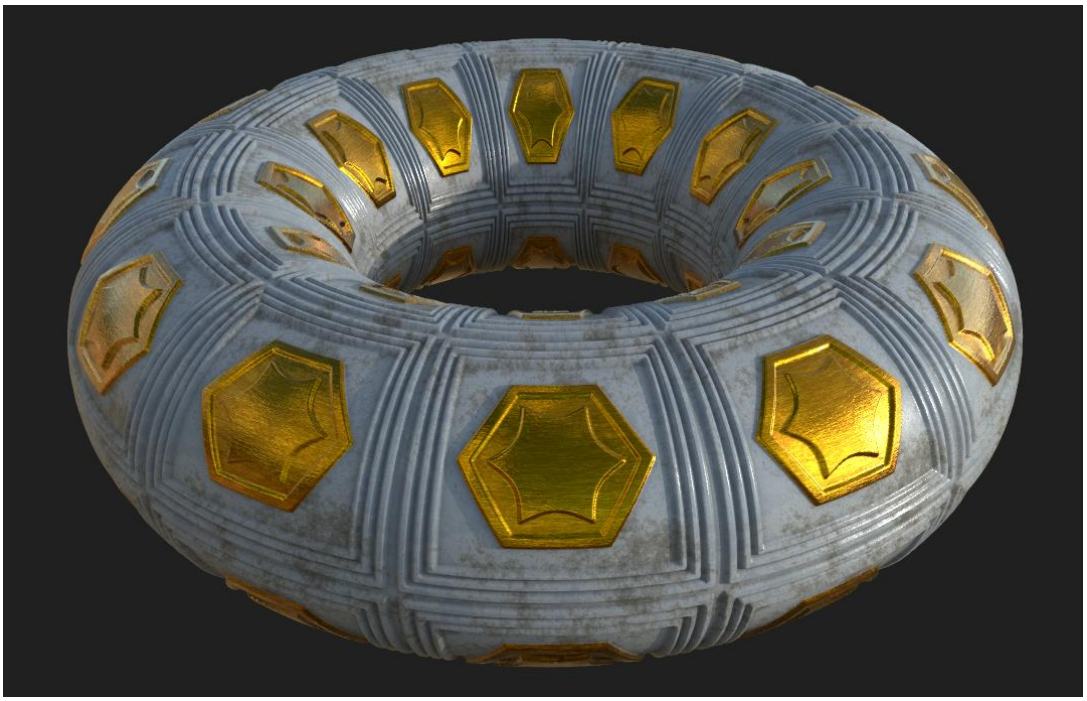
Slika 78: Sve teksturne mape materijala iz praktičnog primjera (autorski rad)



Slika 79: Graf materijala iz praktičnog primjera (autorski rad)

Slika 80: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu zaobljene kocke (autorski rad)

Slika 81: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu zaobljenog cilindra (autorski rad)



Slika 82: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu torusa (autorski rad)

8. Zaključak

Teksture su jedan od najvažnijih elemenata 3D računalne grafike, bez tekstura 3D modeli bi izgledali dosadno, neprivlačno i nerealistično. Danas je vrlo popularan trend fizički zasnovanog renderiranja (PBR) koji koristeći zakone fizike o interakciji svjetlosti s površinama 3D modelima daje visok stupanj realizma.

Postoje mnoge različite vrste teksturnih mapa, od kojih svaka daje površini jedno svojstvo. Kombinacijom više teksturnih mapa, simulirani materijal postaje potpun – dobiva sva najvažnija površinska svojstva materijala iz stvarnog svijeta (boju, dubinu, sjaj, detalje itd.). Teksturiranje se također može izvesti tako da se 3D modelu dodijeli samo boja preko *base color* mape, ali na taj način teksturirani 3D model općenito ne bi bio vrlo impresivan. Isto tako, bilo bi moguće modelirati sve oblike i detalje 3D modela ručno u programu za 3D modeliranje, ali to obično nije najefikasniji način rada u 3D produkciji i za njihovu izradu bilo bi potrebno previše vremena. S postojećim teksturnim mapama moguće je dobiti rezultate gotovo jednako dobre kvalitete kao i s 3D modelima visoke rezolucije.

Iako se teksturiranje može raditi u bilo kojim aplikacijama za obradu slike, u današnje moderno vrijeme popularno je i najadekvatnije korištenje 3D aplikacija specijaliziranih za teksturiranje, jer je njihova primarna svrha upravo izrada tekstura pa je proces teksturiranja u njima intuitivan, brz i lagan. Postoje mnogi programski alati za teksturiranje. Valja odmah reći da su svi alati relativno dobri, i razlog zašto teksture na nekom 3D modelu izgledaju loše je često zbog slabih artistovih vještina, a ne zbog odabranog lošeg programskog alata za teksturiranje. Istina jest da su jedni alati naprednijih od drugih s određenog stajališta, no odabir programskog alata za teksturiranje ovisi o potrebama i zahtjevima, o tome što se želi postići i koliko jednostavne odnosno složene teksture se očekuju.

Substance 3D Designer je već duži niz godina industrijski standard za izradu tekstura. Pruža proceduralan način izrade tekstura, no same teksture nisu proceduralne. Iako ih artist dizajnira na proceduralan način, završene teksture odnosno krajnji rezultat su obične bitmape odnosno rasterizirane slike. Budući da Substance 3D Designer zasad kao krajnji rezultat generira obične slike odnosno teksture fiksne rezolucije koje se popločavaju, bitno je dizajnirati teksture na način da budu prilično ujednačene po cijeloj svojoj površini odnosno da se ne napravi upečatljivi uzorak koji će još više istaknuti njihovu karakteristiku popločavanja odnosno ponavljanja.

Svi programski alati za teksturiranje kao krajnji rezultat kreiraju rasteriziranu sliku. Problem toga su teksture fiksne rezolucije koje pokrivaju ograničeno područje površine. Zato se često izrađuju u obliku popločavajućih tekstura, kao najpopularnija tehnika efikasnog

teksturiranja. Rješenje spomenutog problema su proceduralne teksture, no i one imaju svoje nedostatke od kojih je najkritičniji vrijeme potrebno da se generiraju, zbog čega još nisu zaživjele u igrama. Proceduralne teksture su vrlo vjerojatno budućnost teksturiranja, no još uvijek je područje kojem je potreban razvoj.

Područje 3D računalne grafike se konstantno razvija. Hardver postaje snažniji, otkrivaju se nove metode optimizacije renderiranja, a realizam filmova i igara je sve veći. Također, polako nestaje jaz između filmova i igara. Teži se hiperrealističnim grafikama i realizmu filmske kvalitete u realnom vremenu koji svakim danom postaju naša stvarnost.

Popis literature

3D Texturing Tools: 10 Best Software Options in 2021. (2020, studeni 19). *CGI Furniture*. https://cgifurniture.com/3d-texturing-tools-top-10/

14 Best 3D texture painting softwares as of 2022. (bez dat.). Slant. Preuzeto 12. srpanj 2022., od https://www.slant.co/topics/8643/~3d-texture-painting-softwares

A23D. (bez dat.-a). *Different maps in PBR Textures*. Preuzeto 26. veljača 2022., od https://www.a23d.co/blog/different-maps-in-pbr-textures/

A23D. (bez dat.-b). *PBR Textures Metallic vs Specular Workflow*. Preuzeto 26. veljača 2022., od https://www.a23d.co/blog/pbr-textures-metallic-vs-specular-workflow/

Aaltonen, E. (2022). *Efficient Texturing Techniques for Game Environment Art* [Bachelor's thesis]. Tampere University of Applied Sciences.

Adobe. (bez dat.-a). *Substance 3D Designer Documentation*. Preuzeto 26. svibanj 2022., od https://substance3d.adobe.com/documentation/sddoc/substance-3d-designer-102400008.html

Adobe. (bez dat.-b). *Top 3D design software to create parametric 3D assets, models, materials, and patterns | Adobe Substance 3D Designer*. Preuzeto 26. svibanj 2022., od https://www.adobe.com/products/substance3d-designer.html

Adobe. (bez dat.-c). *What Is PBR (physically based rendering)?* Preuzeto 15. siječanj 2022., od https://www.adobe.com/products/substance3d/discover/pbr.html

Arm Developer. (bez dat.). *Real-time 3D Art Best Practices: Texturing*. Arm Developer. Preuzeto 15. ožujak 2022., od https://developer.arm.com/documentation/102449/0100/Texture-atlasing

Ayyildiz, G. (2020). *Environmental Storytelling Through Procedural Materials in Video Games* [Master's thesis]. Aalto University.

Bittner, F. (2021). *Comparison of material models in modern physically based rendering pipelines*. Gesellschaft für Informatik.

Busquets Duran, R. (2014). *Procedural Textures Generation: Adaptation into a Unity tool* [Bachelor's thesis]. Universitat Politècnica de Catalunya BarcelonaTech.

CGHero. (2022, ožujak 4). *What is PBR Texturing?* CGHero. https://cghero.com/glossary/pbr-texturing

Chopine, A. (2011). *3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation*. Elsevier/Focal Press.

deathmule. (2015, studeni 11). *FALLOUT 4 Before & After the Fallout!* https://www.youtube.com/watch?v=WR1moYnbHKc

Dunlop, R. (Ur.). (2014). *Production Pipeline Fundamentals for Film and Game*. Focal Press,Taylor Francis Group.

Ebert, D. S. (Ur.). (2003). *Texturing & Modeling: A Procedural Approach* (3rd ed). Academic Press.

Epic Games. (bez dat.-a). *Unreal Engine 5*. Preuzeto 26. svibanj 2022., od https://www.unrealengine.com/en-US/unreal-engine-5

Epic Games. (bez dat.-b). *Unreal Engine 5.0 Documentation*. Preuzeto 26. svibanj 2022., od https://docs.unrealengine.com/5.0/en-US/

Epic Games. (bez dat.-c). *Unreal Engine 5.0 Documentation: Physically Based Materials*. Preuzeto 20. svibanj 2022., od https://docs.unrealengine.com/5.0/en-US/physically-based-materials-in-unreal-engine/

Hurtubise, F. (2018, veljača 13). *Using Tileable Textures in Game Environments*. 80 Level. https://80.lv/articles/using-tileable-textures-in-game-environments

Jantunen, J. (2017). *Creating procedural textures for games with Substance Designer* [Bachelor's thesis]. TURKU University of Applied Sciences.

Kelly, H. (2022). *Environment Art in the Game Industry: A Guide to Rich and Realistic Environments* (First Edition). CRC Press.

Kronenberger, L. (2020, rujan 14). *Balancing modularity and uniqueness in Environment Art*. Beyond Extent. https://www.beyondextent.com/articles/balancing-modularity-and-uniqueness-in-environment-art

Kumar, A. (2020). *Beginning PBR texturing: Learn physically based rendering with allegorithmic's substance painter*. Apress. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2480326

Lasram, A. (2012). *Exploring and rendering synthesized textures* [Doctoral thesis]. Université de Lorraine.

Lazzaro, J. (2019, travanj 18). *Ultimate Trim & Animated Forcefield Material Study*. 80 Level. https://80.lv/articles/001agt-ultimate-trim-animated-forcefield-material-study/

LearnOpenGL. (bez dat.). *PBR Theory*. Preuzeto 15. siječanj 2022., od https://learnopengl.com/PBR/Theory

Manghnani, D. (2022, travanj 27). 10 Best 3D Texture Painting Software In 2022 (Updated). *We The Geek*. https://wethegeek.com/best-3d-texture-painting-software/

McDermott, W. (2018). *The PBR Guide: A Handbook for Physically Based Rendering* (3. izd.). Allergorithmic.

Mesquita, L. (2021a, siječanj 4). *Channel packing tip and other types of maps*. ArtStation - Everything About PBR Textures And A Little More - PART 5.

https://www.artstation.com/blogs/luismesquita/7DPb/everything-about-pbr-textures-and-a-little-more-part-5

Mesquita, L. (2021b, siječanj 4). *Important texture maps used in any workflow*. ArtStation - Everything About PBR Textures And A Little More - PART 3. https://www.artstation.com/blogs/luismesquita/RYV0/everything-about-pbr-textures-and-a-little-more-part-3

Mesquita, L. (2021c, siječanj 4). *Main texture maps used in the specular and metallic workflows*. ArtStation - Everything About PBR Textures And A Little More - PART 2. https://www.artstation.com/blogs/luismesquita/jGXd/everything-about-pbr-textures-and-a-little-more-part-2

Mesquita, L. (2021d, siječanj 4). *Other situational texture maps*. ArtStation - Everything About PBR Textures And A Little More - PART 4. https://www.artstation.com/blogs/luismesquita/8pGg/everything-about-pbr-textures-and-a-little-more-part-4

Mike. (2021, rujan 2). 14 Best 3D Texture Painting Software You Can Use. *TechBar*. https://techbar.org/3d-texture-painting-software/

Olsen, M. (2015). *The Ultimate Trim: Texturing Techniques of Sunset Overdrive*. GDC Vault. https://www.gdcvault.com/play/1022324/The-Ultimate-Trim-Texturing-Techniques

OmniShade. (2022, veljača 20). *OmniShade—Vertex color & texture painting*. https://www.youtube.com/watch?v=PgEv3Au6X08

Pai, H.-Y. (2019). Texture designs and workflows for physically based rendering using procedural texture generation. *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 195–198. https://doi.org/10.1109/ECICE47484.2019.8942651

Parsons, M. R. (2017). *Usage based materials by simulating layered imperfections* [Master's thesis]. Texas A&M University.

Procedural Texturing. (bez dat.). Introduction to Shading. Preuzeto 03. ožujak 2022., od https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/procedural-texturing

Quixel. (2020, listopad 13). *Megascans Plugin for Unreal Engine: Vertex Blend Material*. https://www.youtube.com/watch?v=j_3_lYyydgA

R. Moreira, L. G., & S. Lages, W. (2014). *Organic Tile Textures using Fragment Shaders*. Universidade Federal de Minas Gerais, School of Fine Arts, Brazil.

Ratković, J. (2017). *Physically based rendering* [Master's thesis]. University of Zagreb, Faculty of Electrical Engineering and Computing.

Sanchis Albert, A. (2019). *Creation of Procedural Materials in Substance Designer and their Application in Unity* [Bachelor's thesis]. Universitat Jaume I.

Sharma, T. (2021, svibanj 20). *10 Best 3D Texture Painting Software In 2021*. TechPout. https://www.techpout.com/3d-texture-painting-software/

Shirley, P., Morley, R. K., Sloan, P.-P., & Wyman, C. (2012). Basics of physically-based rendering. *SIGGRAPH Asia 2012 Courses on - SA '12*, 1–11. https://doi.org/10.1145/2407783.2407785

Souza, D. A., & Mota, R. R. (2021). Materials for games—An overview on creating materials for games. *Anais Estendidos do XX Simpósio Brasileiro de Games e Entretenimento Digital (SBGames Estendido 2021)*, 136–142. https://doi.org/10.5753/sbgames_estendido.2021.19633

Ta'eed, C. (2022, siječanj 24). *How to Turn a Texture Into a Seamlessly Tiled Background*. Design & Illustration Envato Tuts+. https://design.tutsplus.com/articles/how-to-turn-a-texture-into-a-seamlessly-tiled-background--psd-1565

Unity Technologies. (bez dat.-a). *Decal Renderer Feature | Universal RP | 13.0.0*. Unity - Manual. Preuzeto 17. ožujak 2022., od https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@13.0/manual/renderer-feature-decal.html

Unity Technologies. (bez dat.-b). *Unity 2022.2 Documentation: Metallic vs. Specular workflow*. Preuzeto 20. svibanj 2022., od https://docs.unity3d.com/2022.2/Documentation/Manual/StandardShaderMetallicVsSpecular.html

Unreal Engine. (2020, svibanj 13). *Unreal Engine 5 Revealed! | Next-Gen Real-Time Demo Running on PlayStation 5*. https://www.youtube.com/watch?v=qC5KtatMcUw

Weathering. (bez dat.). The Geological Society. Preuzeto 07. svibanj 2022., od https://www.geolsoc.org.uk/ks3/gsl/education/resources/rockcycle/page3461.html

Which Software is Best for Texturing 3D Models? (2021, prosinac 11). *Asoftclick*. https://asoftclick.com/best-3d-model-texturing-software/

Why can't we create video game graphics as good as cinematics? Shouldn't the texturing be the same as making video games? (bez dat.). Quora. Preuzeto 18. ožujak 2022., od https://www.quora.com/Why-can-t-we-create-video-game-graphics-as-good-as-cinematics-Shouldnt-the-texturing-be-the-same-as-making-video-games

Wolfram. (bez dat.). *Physically Based Rendering*. Preuzeto 15. siječanj 2022., od https://reference.wolfram.com/language/tutorial/PhysicallyBasedRendering.html

Young, P., & Skuratovskiy, R. (2019, siječanj 23). *Adobe Acquires Allegorithmic, the Leader in 3D Editing and Authoring for Gaming and Entertainment*. https://news.adobe.com/news/news-details/2019/Adobe-Acquires-Allegorithmicthe-Leader-in-3D-Editing-and-Authoring-for-Gaming-and-Entertainment/default.aspx

Popis slika

Slika 1: Teksturirani prikaz umivaonika (lijevo), prikaz umivaonika bez tekstura (desno) (autorski rad) 3
Slika 2: Prikaz umivaonika (lijevo) i njegova UV mapa (desno) (autorski rad) 5
Slika 3: Planarna, cilindrična, i sferična projekcija (Chopine, 2011, str. 152) 5
Slika 4: Cilindrično mapiranje ruke (Chopine, 2011, str. 156) 6
Slika 5: UV školjka (desno) i odgovarajući dio na prikazu umivaonika (lijevo) (autorski rad) . 6
Slika 6: Perlinov šum – primjer jedne proceduralne teksture (autorski rad) 7
Slika 7: Horizontalni gradijent u sivim tonovima - primjer jedne jednostavne proceduralne teksture (autorski rad) 8
Slika 8: *Diffuse* mapa ima informacije o sjenama i svjetlu, *albedo* mapa ima samo boju (Kelly, 2022, str. 7) 11
Slika 9: 3D model umivaonika u različitim uvjetima rasvjete korištenjem HDRI (autorski rad) 12
Slika 10: Desna renderirana slika slavine umivaonika izgleda realističnije zbog efekta bljeska u objektivu kamere (autorski rad) 13
Slika 11: Refleksija i difuzija (Kelly, 2022, str. 8) 14
Slika 12: Lijeva sfera svjetlost više raspršuje, desna sfera ju više reflektira (autorski rad) ...15
Slika 13: Primjer obojene refleksije metala (autorski rad) 16
Slika 14: Fresnel efekt (Kelly, 2022, str. 11) 17
Slika 15: Utjecaj mikropovršine na refleksiju svjetlosti (Kelly, 2022, str. 12) 17
Slika 16: Desna sfera ima hrapaviju površinu pa refleksija izgleda mutnije (autorski rad) 17
Slika 17: Utjecaj reflektivnosti objekta na refleksiju i difuziju svjetlosti (autorski rad) 18
Slika 18: Utjecaj hrapavosti objekta na refleksiju svjetlosti (Kelly, 2022, str. 13) 19
Slika 19: Primjer *albedo* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad) 21
Slika 20: Primjer *alpha* mape (lijevo) i njena manifestacija na površini (desno) – bijeli dijelovi otkrivaju *albedo* mapu iz prethodnog primjera (autorski rad) 21
Slika 21: Primjer *normal* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad) 22
Slika 22: Primjer *height* mape (lijevo), njena manifestacija na površini visoke rezolucije (sredina), njena manifestacija na površini niske rezolucije (desno) (autorski rad) 23
Slika 23: Primjer *height* mape (lijevo) i njena manifestacija na površini (desno) – daje efekt cigla na zidu (autorski rad) 24

Slika 24: Primjer *bump* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad) 24
Slika 25: Primjer *AO* mape (lijevo), površina s *normal* mapom bez *AO* mape (sredina), površina s *normal* mapom i *AO* mapom (desno) (autorski rad) 25
Slika 26: Primjer *roughness* mape (lijevo) i njena manifestacija na površini (desno) – na crnim dijelovima se vidi refleksija okruženja (oblaci) (autorski rad) 26
Slika 27: Primjer *roughness* mape s varijacijama sivih tonova (lijevo) i njena manifestacija na površini (desno) – izgleda realističnije i zanimljivije (autorski rad).....26
Slika 28: Primjer *metallic* mape (lijevo) i njena manifestacija na površini (desno) (*albedo* mapa je zlatne boje) – na metalnim dijelovima refleksija je jača i obojena (vidljiva je obojana refleksija oblaka), dok je na nemetalnim dijelovima slabija i neobojena (autorski rad) 27
Slika 29: Primjer *specular* mape (lijevo) i njena manifestacija na površini (desno) (autorski rad).....28
Slika 30: U ovom demu od Epic Gamesa napravljenom u Unreal Engineu 5, testiranom u stvarnom vremenu na PlayStationu 5, slika prikazane scene sadrži oko 20 milijuna trokuta (poligona). Donja slika prikazuje kako ti trokuti izgledaju – svaki susjedni trokut je prikazan drugom bojom kako bi se bolje uočio. Većina trokuta je veličine piksela na ekranu, zbog čega slika izgleda kao šum (Prema: Unreal Engine, 2020) 34
Slika 31: Žuto označeni objekti dijele istu teksturu (Arm Developer, bez dat.).....36
Slika 32: Primjer popločavanja jedne popločavajuće teksture, granice teksture su označene crvenim kvadratom (autorski rad) 36
Slika 33: Primjer kako upečatljivi uzorci na popločavajućim teksturama još više ističu njihovu karakteristiku popločavanja odnosno ponavljanja (autorski rad) 37
Slika 34: Primjer korištenja traka (Kronenberger, 2020).....37
Slika 35: Primjer jedne teksture na kojoj je primijenjena tehnika ultimativne trake – vidljiv je standardizirani raspored traka (Olsen, 2015).....38
Slika 36: Primjer jedne *normal* mape na kojoj je primijenjena tehnika ultimativne trake – vidljiv je kut od 45 stupnjeva na rubovima svih traka (Olsen, 2015) 39
Slika 37: Objekt bez *normal* mape (lijevo), objekt s *normal* mapom (desno). Kut od 45 stupnjeva na *normal* mapi stvara iluziju da objekt ima veću rezoluciju (Prema: Olsen, 2015) 39
Slika 38: Različite naljepnice na popločavajućoj teksturi (Hurtubise, 2018).....40
Slika 39: Primjer projiciranja teksture naljepnice na površine objekata unutar Unity motora za razvoj igara (Unity Technologies, bez dat.-a).....40
Slika 40: Primjer pakiranja *AO*, *roughness* i *metallic* mapa u jednu mapu (Aaltonen, 2022, str. 20).....41

Slika 41: Tekstura pločica je pomiješana s teksturom zemlje i s teksturom trave (Prema: Quixel, 2020) 42
Slika 42: Miješanje tekstura metodom bojanja vrhova (Prema: OmniShade, 2022) 42
Slika 43: Primjer korištenja *blend* mape za miješanje tekstura na velikoj površini (Aaltonen, 2022, str. 26) 43
Slika 44: Primjer biološkog trošenja materijala (lijevo), primjer kemijskog trošenja materijala (sredina), primjer mehaničkog trošenja materijala (desno) (Parsons, 2017, str. 4) 44
Slika 45: Primjer prljavštine na metalu: na mjestima gdje su male količine prljavštine, na *metallic* mapi (lijevo) su nijanse sive; na mjestima gdje je prljavština potpuno pokrila metal, na *metallic* mapi je crno (autorski rad) 45
Slika 46: Scena u *Fallout 4* igri prije rata (gore), ista scena nakon rata (dolje) (Prema: deathmule, 2015) 46
Slika 47: Primjer jednog projekta u Substance 3D Designeru (autorski rad) 47
Slika 48: Primjer jednostavnog grafa na kojem je moguće vidjeti sve prethodno spomenute vrste čvorova (autorski rad) 48
Slika 49: Sfera s utorima i oštrim rubovima (autorski rad) 50
Slika 50: Primjer kako *shader* upravlja varijacijom i roditelj-dijete odnosom. S lijeva na desno: nema varijacije; varijacija roditelja; varijacija roditelja i djece (R. Moreira & S. Lages, 2014, str. 1) 51
Slika 51: Izgled materijala iz praktičnog primjera (autorski rad) 52
Slika 52: Koraci (desno) i rezultat (lijevo) (autorski rad) 54
Slika 53: Koraci (desno) i novi rezultat (lijevo) (autorski rad) 54
Slika 54: Koraci (desno) i novi rezultat (lijevo) (autorski rad) 55
Slika 55: Koraci (desno) i rezultat (lijevo) (autorski rad) 55
Slika 56: Kraj razvoja *height* mape; koraci (desno) i konačan rezultat (lijevo) (autorski rad). 56
Slika 57: *Height* mapa (lijevo) i njena manifestacija na površini (desno) (autorski rad) 56
Slika 58: Koraci (desno) i rezultat (lijevo) (autorski rad) 57
Slika 59: *Ambient occlusion* mapa (lijevo), izgled površine s *ambient occlusion* i *height* mapama zajedno (desno) (autorski rad) 57
Slika 60: Izgled površine samo s *height* mapom (lijevo), izgled površine s *height* i *ambient occlusion* mapama (desno) (autorski rad) 57
Slika 61: Koraci (desno) i rezultat (lijevo) (autorski rad) 58
Slika 62: Kraj razvoja *roughness* mape; koraci (desno) i rezultat (lijevo) (autorski rad) 58
Slika 63: *Roughness* mapa (lijevo); izgled površine s *roughness* mapom zajedno s *height* i *ambient occlusion* mapama (desno) (autorski rad) 59
Slika 64: Koraci (desno) i rezultat (lijevo) (autorski rad) 59

Slika 65: *Metallic* mapa (lijevo); izgled površine s *metallic* mapom zajedno s *height*, *ambient occlusion* i *roughness* mapama (autorski rad) 60
Slika 66: Miješanje dviju slika u *Blend* čvoru preko operacije *multiply* (autorski rad) 60
Slika 67: Koraci (desno) i rezultat (lijevo) (autorski rad) 61
Slika 68: Primjer korištenja *Gradient Map* čvora za mapiranje slike u sivim tonovima na prilagođenu rampu boja (autorski rad) 61
Slika 69: Koraci (desno) i rezultat (lijevo) (autorski rad) 61
Slika 70: Koraci (desno) i rezultat (lijevo) (autorski rad) 62
Slika 71: Koraci (desno) i rezultat (lijevo) (autorski rad) 62
Slika 72: Koraci (desno) i rezultat (lijevo) (autorski rad) 63
Slika 73: Kraj razvoja *base color* mape; koraci (desno) i rezultat (lijevo) (autorski rad) 63
Slika 74: *Base color* mapa (lijevo); izgled površine s *base color* mapom zajedno s *height*, *ambient occlusion*, *roughness* i *metallic* mapama (autorski rad) 63
Slika 75: Koraci (desno) i rezultat (lijevo) (autorski rad) 64
Slika 76: Kraj razvoja *normal* mape; koraci (desno) i rezultat (lijevo) (autorski rad).....65
Slika 77: *Normal* mapa (lijevo); izgled površine s *normal* mapom zajedno s *height*, *ambient occlusion*, *roughness*, *metallic* i *base color* mapama (autorski rad) 65
Slika 78: Sve teksturne mape materijala iz praktičnog primjera (autorski rad).....65
Slika 79: Graf materijala iz praktičnog primjera (autorski rad) 66
Slika 80: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu zaobljene kocke (autorski rad) 66
Slika 81: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu zaobljenog cilindra (autorski rad) 67
Slika 82: Primjer popločavanja materijala iz praktičnog primjera na 3D modelu torusa (autorski rad) 67