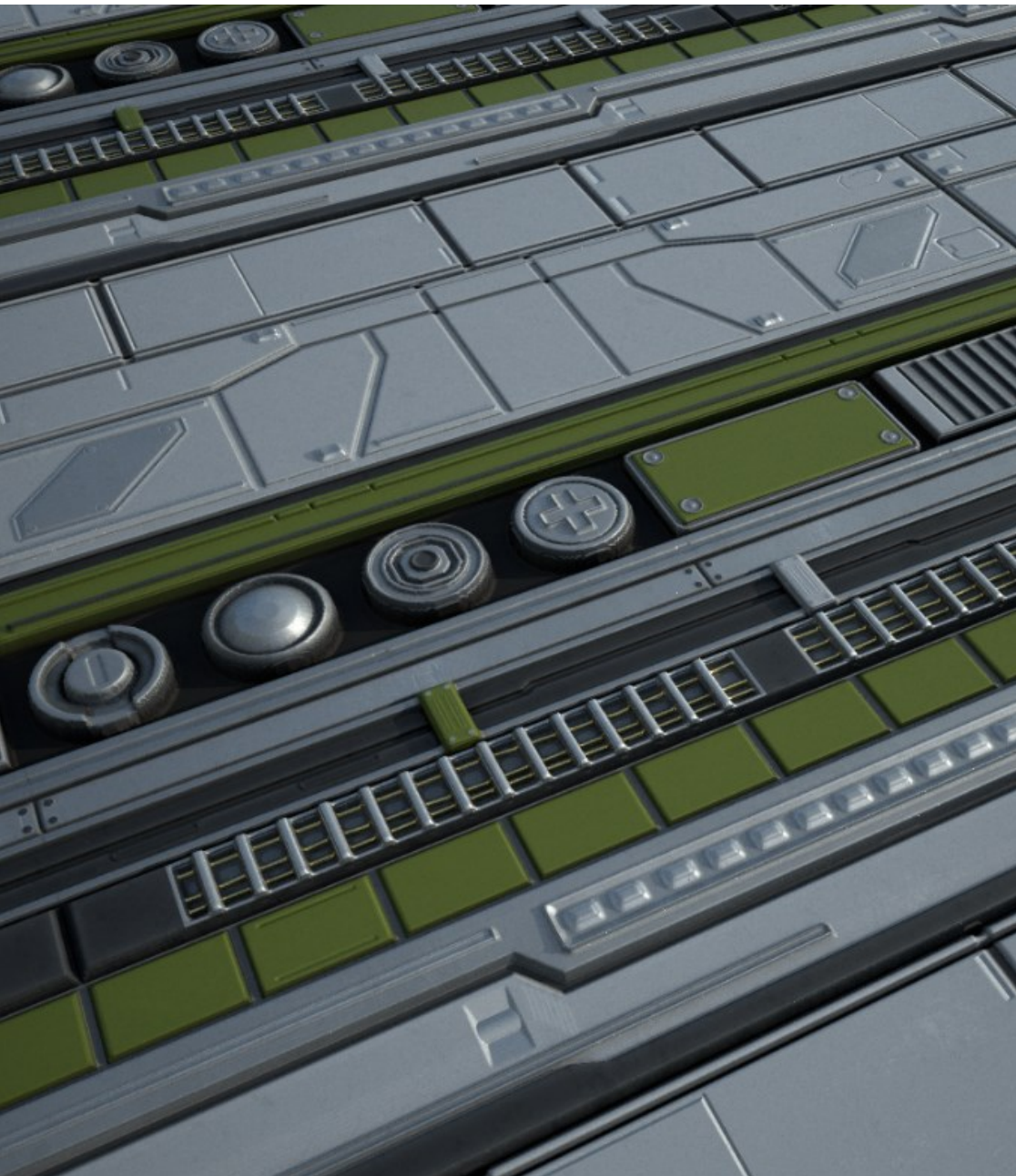


Janne Viitala

Listatekstuurien valmistus ja käyttöönotto



Tradenomi (AMK)

Tietojenkäsittely

Kevät 2021



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä: Viitala Janne

Työn nimi: Listatekstuurien valmistus ja käyttöönotto

Tutkintonimike: Tradenomi (AMK), tietojenkäsittely

Asiasanat: teksturointi, grafiikka, 3D-mallinnus

Tämän opinnäytetyön tavoitteena oli käydä läpi listatekstuurien suunnittelu- valmistus- sekä käyttöönotto vaiheet. Listatekstuurit ovat monikäyttöisiä elementtejä, jotka mahdollistavat useiden peliobjektien tai peliympäristöjen tehokkaan ja vaivattoman teksturoinnin. Työssä kehitettiin työnkulku, jolla tehostettiin tekstuurien valmistusta ja iterointia.

Työn alussa syvennyttiin listatekstuurien ominaisuuksiin ja niiden käyttötapoihin sekä tarkasteltiin esimerkkejä erilaisista tapauksista, joissa listatekstuureja hyödynnetään. Seuraavaksi työssä tutkittiin kovapintaista mallinnusta, tahkonormaalien manipulointia sekä tekstuurien leipomista, jonka jälkeen selvitettiin listatekstuureille ominaisia UV-kartoitus- ja optimointitekniikoita.

Työn käytännön osuudessa suunniteltiin ja mallinnettiin listatekstuurin korkeapolygoninen malli, joka leivottiin tekstuuridatan saamiseksi. Tekstuuridatan avulla luotiin materiaalit, joiden perusteella mallinnettiin sekä UV-kartoitettiin kokoelma peliobjekteja. Lopuksi peliobjekteista rakennettiin peliympäristö pelimootoriin. Käytännön osuudessa keskityttiin teksturointiprosessin tehokkuuden parantamiseen erilaisilla menetelmillä ja lisäosilla.

Listatekstuurien käyttö yhdistettynä erilaisiin työkaluihin voi säästää paljon aikaa verrattuna tavalliseen teksturointiprosessiin. Listatekstuurit mahdollistavat runsaasti eri lähestymistapoja, tarjoten peligraafikoille paljon luovaa vapautta, mutta ilman kunnollista ja perusteellista suunnittelua listatekstuurit eivät tarjoa kaikkea niiden potentiaalia grafiikan valmistuksessa.

Abstract

Author: Viitala Janne

Title of the Publication: Production and Implementation of Trim Sheet Textures

Degree Title: Bachelor of Business Administration

Keywords: texturing, graphics, 3D-modelling

The purpose of this thesis is to explain the design, production and implementation stages of trim sheet textures. Trim sheet textures are multi-purpose elements that allow for efficient and effortless texturing of multiple game assets or environments. During this work, a workflow was developed to streamline the production and iteration of trim sheet textures.

The work began by explaining the essential properties of trim sheet textures and their uses, then examined examples of different cases in which trim sheets are utilized. Next, the thesis focused on high poly modelling, weighted vertex normals and texture baking. After that, the work studied UV mapping and optimization techniques related to trim sheets.

In the practical part of this thesis, a high poly model of a trim sheet was designed and modelled, which was then baked to obtain the essential texture data. The data was then used to help create materials for the trim sheet. Next, a collection of game assets was modelled and UV-unwrapped to suit the trims. The ready assets were then used to form an environment in the game engine. The practical part focuses on improving the texturing pipeline with the help of different tools and add-ons.

The use of trim sheet textures in combination with various tools can save a lot of time compared to a standard texturing process. Trim sheets allow for a wide variety of approaches, offering artists a lot of creative freedom, but without proper planning, trim sheets won't offer all their potential in the production of game graphics.

Sisällys

1	Johdanto	1
2	Listatekstuurit.....	2
2.1	Modulaarisuus.....	2
2.2	Listatekstuurien käyttötapoja	3
2.3	Listatekstuurien suunnittelu	4
2.4	Dekaalit.....	5
3	3D-mallinnus.....	7
3.1	High poly ja low poly	7
3.2	Kovapintainen mallinnus.....	8
3.3	Leipominen.....	9
3.4	UV-kartoitus	9
3.5	UV-setit.....	10
3.6	Optimointi	10
4	Teksturointi.....	12
4.1	Fysikaalisiin ominaisuuksiin perustuva renderöinti ja tekstuurikartat	12
4.2	Kanavien pakkaaminen	13
5	Listatekstuurin valmistus ja peliympäristön rakentaminen	14
5.1	Tekstuurien suunnittelu	14
5.2	High poly -mallin valmistus	14
5.3	Teksturointi	18
5.4	Materiaalien luonti Unityssä ja Blenderissä.....	20
5.5	Mallinnus ja UV-kartoitus.....	22
6	Yhteenveto	29
	Lähteet	30
	Liitteet	

Symboliluettelo

Alfa	Bittikarttakuvan läpinäkyvyysarvo
Bevel	3D-mallin reunan viistoleikkaus
Dekaali	Tekstuurikartan osa, jota voidaan käyttää koristeena
High Poly	Korkeapolygoninen 3D-malli
Instantiointi	Ennalta määritetyn objektin kopiointi dynaamisesti
Low Poly	Matalapolygoninen 3D-malli
Materiaali	Tekstuurikartoista ja varjostimista koottu kokonaisuus
Modifikaattori	Blender-ohjelmiston työkalu
PBR	Fysikaalisiin ominaisuuksiin perustuva renderöintitekniikka
Taso	Neliön muotoinen tahko
Proppi	Peliympäristöissä käytettävä lavaste
Skene	Pelimoottorin tiedosto, johon kootaan pelilogiikka sekä -objektit
Tekselitiheys	3D-mallin pinnalle varattu tekstuurin tarkkuus
UV-tila	Kaksiulotteinen koordinaatisto, johon 3D-mallin tahkojen pinnat määritetään
UV-saareke	3D-mallin tahkoista muodostuva ryhmä UV-tilassa
Varjostin	Koodi, joka määrittelee materiaalien ominaisuuksia ja toimintoja

1 Johdanto

Listatekstuurit ovat 3D-grafiikan tuotannossa monikäyttöisiä ja tehokkaita elementtejä, jotka mahdollistavat useiden eri peliobjektien teksturoinnin vaivattomasti sekä ovat tekniikkana peligraafikoille tärkeä aihe, jos haluaa pysyä ajan tasalla moderneista pelinkehitystyökaluista ja -menetelmistä. Valitsin opinnäytetyöni aiheeksi listatekstuureiden valmistamisen ja käyttöönoton koska se tukee ammattiani sekä koen, että aiheeseen paremmin perehtymällä kehitän osaamistani usealla peligrafiikan tuottamisen osa-alueella, kuten teksturoinnissa, UV-kartoituksessa sekä 3D-mallinnuksessa.

Opinnäytetyö keskittyy paljolti peligrafiikan teknisiin aiheisiin, ja se on tarkoitettu enimmäkseen peligrafiikan perusteet hallitseville lukijoille. Opinnäytetyön tavoite on tarjota peligrafiikasta ja etenkin ympäristögrafiikasta kiinnostuneille lukijoille ohjeistus listatekstuurien valmistusprosessiin.

Opinnäytetyön teoriaosuudessa käydään läpi listatekstuurien valmistamiseen liittyviä aiheita, kuten high poly -mallinnus ja leipominen, tekstuurikarttojen pakkaaminen, dekaalit, UV-kartoitustekniikat sekä objektien optimointi. Toiminnallisen osuuden tarkoituksena on valmistaa projekti-
muotoisesti scifi-aiheinen listatekstuuri, mallintaa kokoelma peliobjekteista, sekä UV-kartoittaa ne listatekstuurille sopivaksi, selittäen valmistusprosessin keskeiset työkalut ja menetelmät järjestelmällisesti. Lopuksi selitetään teksturoitujen mallien käyttöönotto Unity-pelimoottorissa.

2 Listatekstuurit

Listatekstuurin eli trim sheet -tekstuurin pääperiaate on olla kokoelma vaaka- tai pystysuorista liuskoista, jotka toistuvat yhdellä akselilla tekstuuriilassa. Niiden tarkoitus on helpottaa peligraafikkoja teksturoimaan mallien pintoja tehokkaammin, säästää paljon aikaa grafiikan valmistuksessa sekä mahdollistaa optimoidumpia peliympäristöjä. Jotta saumattomien tekstuurien toistuvuus ei kävisi liian ilmiselväksi, on vältettävä suuria, silmiinpistäviä yksityiskohtia tai värin muutoksia. [1.]

Kuvassa 1 on listatekstuuri ja sitä käyttävä scifi-koneisto. Koska kyseisen listatekstuurin liuskat on määritetty tietyn kokoiseksi ja asetettu tiettyyn järjestykseen sekä mallin UV-saarekkeet asetettu tekstuuriin sopivaksi, olisi mahdollista muuttaa mallin ulkonäkö varsin helposti valmistamalla uusi listatekstuuri samoja koko- ja järjestysmäärittäjiä käyttäen.



Kuva 1. Esimerkki listatekstuurista ja sitä käyttävästä mallista.

2.1 Modulaarisuus

Peligrfiikassa hyödyllinen käytäntö on valmistaa peliobjektit uudelleenkäytettävyyttä ajatellen. Eri osien yhdistely ja vaihtelu säästävät paljon aikaa verrattuna jokaisen peliobjektin valmistamiseen erikseen. Monet peliympäristöjen rakennukset on valmistettu modulaarisista palasista, joita

hyödyntämällä voidaan rakentaa useita erilaisia kokonaisuuksia. Varsinkin ympäristögrafiikan modulaaristen palasten valmistuksessa on tärkeää säilyttää niiden yhtenäisyys; objektien mitta-suhteet ja pivot-pisteen sijainti helpottavat palasten asettelua kenttäeditoreissa. Listatekstuurit ovat modulaarisia siinä mielessä, että niitä pystytään hyödyntämään useissa eri tapauksissa. Kuvassa 2 on esitelty modulaarinen kokoelma palasista, jotka on teksturoitu yhtä listatekstuuria ja yhtä kokonaan saumatonta tiilitekstuuria käyttäen. Palasista on sitten koottu rakennuksen seinä. Modulaarisuus ei kuitenkaan keskity pelkästään ympäristögrafiikkaan, vaan sitä hyödyntävät muutkin peliobjektit, kuten hahmot tai ajoneuvot. [2.]

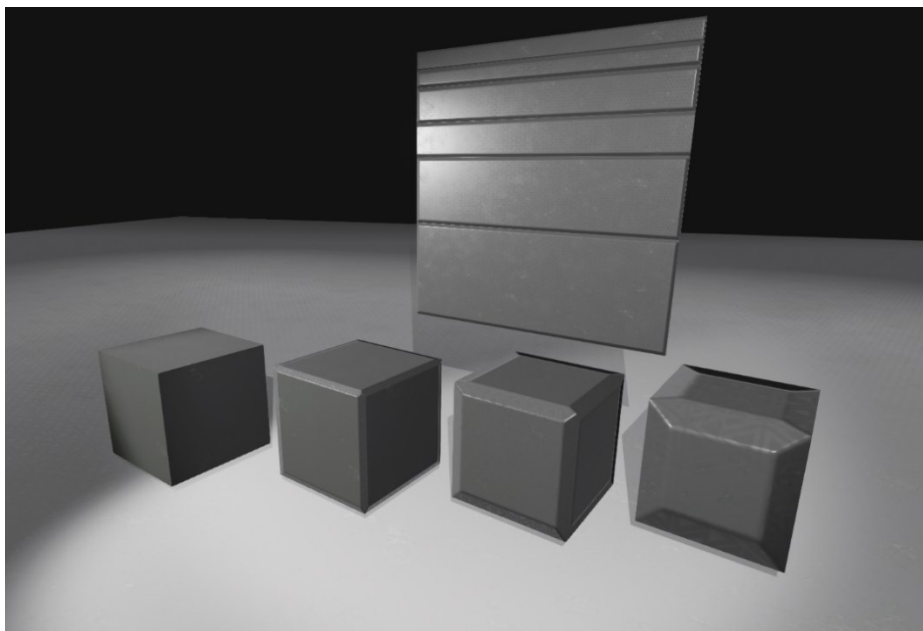


Kuva 2. Modulaarinen listatekstuuria käyttävä kokoelma, josta on luotu tiilirakennuksen seinä.

2.2 Listatekstuurien käyttötapoja

Listatekstuureja käytetään ympäristöjen ja esineiden yksityiskohtien nopeaan ja tehokkaaseen teksturointiin. Liuskoilla voidaan erotella objekteja ja saumattomia materiaaleja toisistaan ja teksturoida ohuita ja pitkiä tahkokaistaleita. Näin vältetään ylimääräisten uniikkien tekstuurien valmistamiselta kaikille objekteille. Yleisiä listatekstuureita hyödyntäviä objekteja ovat esimerkiksi huoneiden katto- ja lattialistat, kaapelit ja köydet. Olemalla luova listatekstuurien kanssa avaa loputtomat mahdollisuudet eri ratkaisuille niiden käytössä. [1.]

Listatekstuurien yksi hyödyistä on normal bevel -tekniikka. Se mahdollistaa illuusion pyöreämistä reunoista säilyttäen mallin alkuperäisen muodon. Illuusio saadaan aikaiseksi lisäämällä listatekstuurin normaalikartan liuskojen reunoihin viistoleikkaus. Leikkauksen kulman kannattaa olla 45 astetta, sillä se sopii kaikkiin 90 asteen kulmien teksturointiin. Jos teksturoitavan mallin kulma ei ole tekstuurin bevelien yhteenlaskettu arvo, voi lopputulos näyttää vääraltä. Yhdessä kaksi oikein kartoitettua tahkoa muodostavat saumattoman bevel-efektin. Kuvassa 3 on esimerkki listatekstuurista, jonka liuskoissa on 45 asteen bevel sekä kuutioita. Vasemmanpuoleisella kuutiolla ei ole normaalikarttaa, muut on kartoitettu eri liuskoille sopiviksi. Äärimmäisistä kuvakulmista katseltuina sekä liian isojen normal bevelien omaavien kuutioiden fyysiset kulmat ovat herkästi havaittavissa. Joskus kannattaa siis suosia geometrian lisäämistä oikean viistoleikkauksen muodossa. Normal bevel -tekniikan toinen hyöty on UV-kartoitusvaiheessa listojen helpompi erottaminen toisistaan. [3.]



Kuva 3. Normal Bevel -tekniikkaa hyödyntävä listatekstuuri ja siihen sopivaksi kartoitettuja kuutioita.

2.3 Listatekstuurien suunnittelu

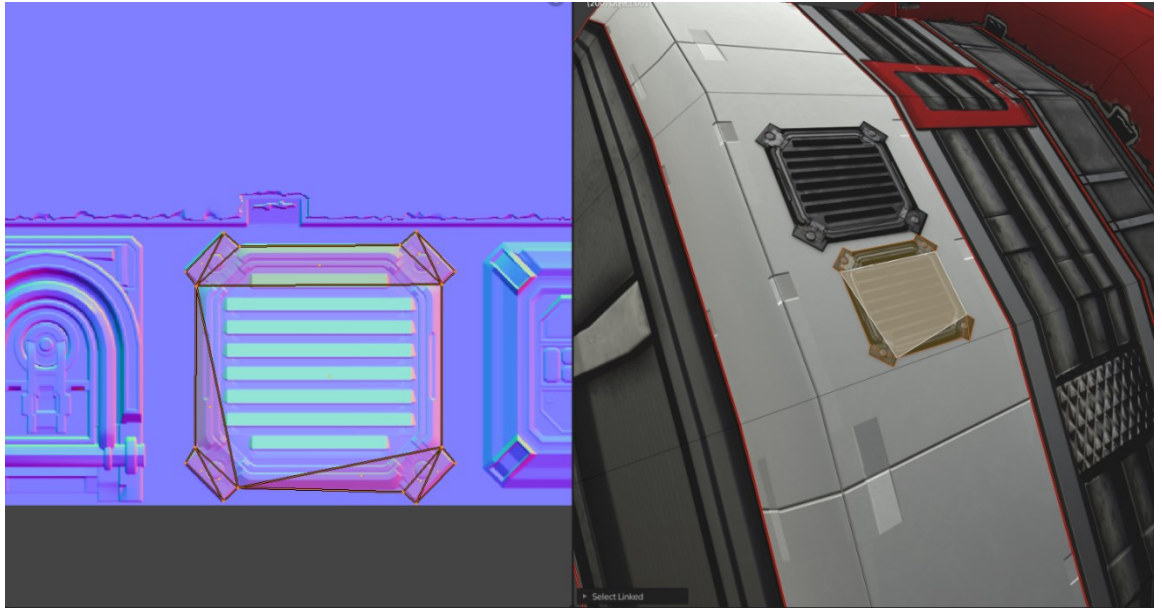
Suunnitteluvaiheessa on hyvä käydä läpi, mihin ja miten listatekstuuria käytetään, mitä materiaaleja se sisältäisi, minkä kokoiset listat tarvitaan, sekä tehdäänkö tekstuurista useita variaatioita. Jos tavoite on selvillä ja saatavilla on referenssikuvia tai konseptikuvia, voidaan paloitella kuvissa

näkyvät elementit eri osiin niiden toistuvuuden, koon sekä materiaalin mukaan. Referensseihin perehtymisen jälkeen on helpompaa valmistaa myös kokoelma modulaarisista 3D-malleista, kuten rakennuksen osista.

Listatekstuurien valmistukseen ei ole ainoastaan yhtä oikeaa tapaa ja työkalua. Valmistuksen eri vaiheissakin on useita vaihtoehtoja, joista voidaan valita sopivin. High poly -mallin valmistuksessa toimivat 3D-mallinnusohjelmat, kuten Blender, Maya tai 3DSMax. Mallinnus voidaan mahdollisesti myös viedä astetta pidemmälle skulptaamalla. 3D-mallin käyttö listatekstuurin luomiseen ei kuitenkaan ole välttämätöntä, vaan tarvittava informaatio voidaan myös luoda esimerkiksi noo-dipohjaisesti Substance Designer -ohjelmassa.

2.4 Dekaalit

Listatekstuureihin voidaan liuskojen lisäksi sisällyttää koristeita, jotka UV-kartoitetaan käyttäen erillisiä polygoneja. Nämä koristeet eli dekaalit voivat olla esimerkiksi ruuveja, kahvoja, ilmastointisäleikköjä, roiskeita tai luodinreikiä. Niitä käytetään ehkäisemään peliobjektien samannäköisyyttä rikkomalla toistuvuutta sekä toimimaan tekstuureina erilaisissa efekteissä. Jotta koristeet saadaan eriteltyä muusta tekstuurista, on käytettävä joko läpinäkyvyysmaskia tai leikattava geometrialla dekaalille ääriviivat. Dekaalien asettelumenetelmät vaihtelevat niiden käyttötarkoitusten mukaan; jotkut staattiset koristeet mallinnetaan suoraan pinnoille mallinnusohjelmistossa, kun taas dynaamiset dekaalit, kuten luodinreiät tai veriroiskeet, instantioidaan pelimoottorissa haluttuihin kohtiin. Dekaalit säästävät mallien valmistuksessa kallisarvoista aikaa sekä vähentävät pelaajien mahdollista kyllästymistä samannäköisiin ja toistuviin objekteihin. Listatekstuurien uudelleenkäytettävyyden ansiosta dekaalit sopivat niihin hyvin, mutta ne voidaan eritellä omalle tekstuurilleen tarpeen vaatiessa. Kuvassa 4 on esimerkki ilmastointisäleikködekaalista leikattuna ja asetettuna mallin pinnalle.

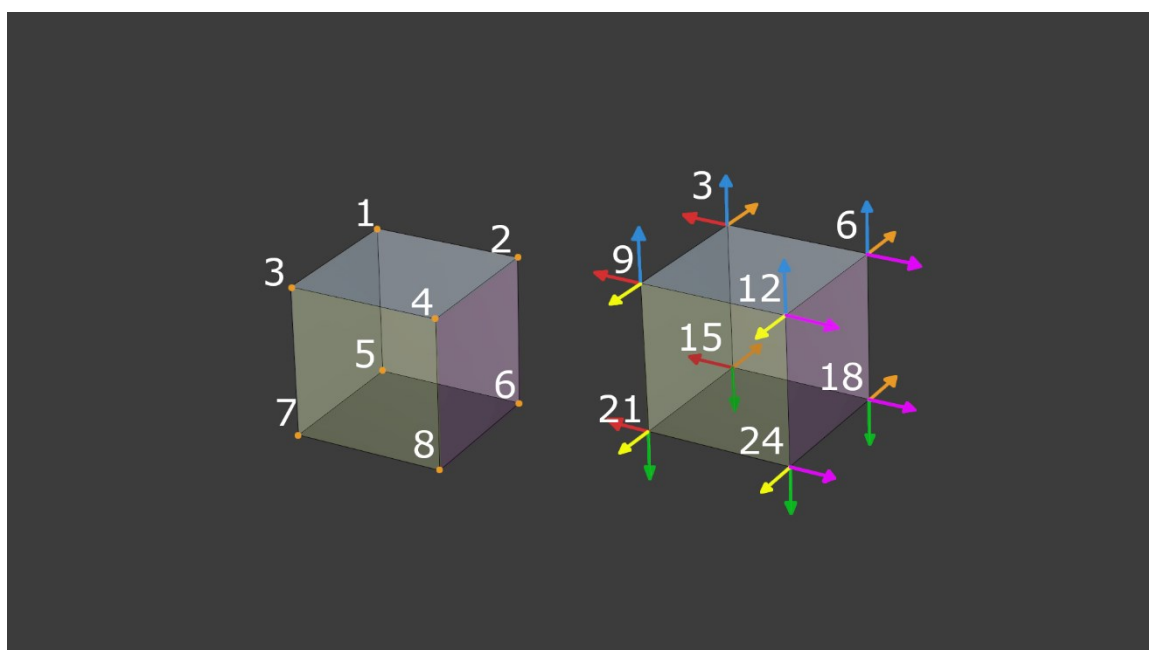


Kuva 4. Dekaalit ovat koristeita, joita voidaan lisätä mallin pinnalle rikkomaan toistuvuutta.

Reunadekaalien yleinen käyttötarkoitus on lisätä eri pintojen reunoille esimerkiksi kulumavaurioita tai hitsaussaumoja. Yksinkertaisuudessaan reunadekaalit valmistetaan skulptaamalla haluttu vaurio tai sauma ja leipomalla high poly -informaatio, kuten normaalikartta, listatekstuurin tapaisesti toistuviksi liuskoiksi. Pohjamallin reunoille mallinnetaan sitten hieman pinnan päällä leijuva malli, joka UV-kartoitetaan sopimaan halutun listatekstuurin kohdalle. [4.]

3 3D-mallinnus

3D-malli muodostuu vertekseistä, pisteistä kolmiulotteisessa avaruudessa. Niitä yhdistämällä muodostuu reunoja ja tahkoja. Jokaisella verteksillä on omat attribuutit, jotka sisältävät niiden sijainnin 3D-avaruudessa ja UV-koordinaatistossa sekä normaalidatan. Jos mallissa on kova reuna tai UV-sauma, joutuvat kyseisen reunan verteksit jakautumaan, jotta tarvittavat attribuutit säilyisivät. Kuvitellaan tavallinen kovareunainen kuutio, se muodostuu kahdeksasta verteksistä, mutta koska jokaisella niistä on kolme eri normaalidataa, joutuvat ne jakautumaan kolmeen osaan, muodostaen kaksikymmentäneljä verteksiä (kuva 5).



Kuva 5. Demonstraatio kuution verteksin jakautumisesta.

Jakautumista on hyvä pitää silmällä pelin optimoinnin kannalta. Tähän voi vaikuttaa pitämällä mallin UV-saarekkeet mahdollisimman yhtenäisinä sekä määrittämällä saumat mieluiten kovien reunojen kohdalle, koska niissä verteksit ovat valmiiksi halkaistuja.

3.1 High poly ja low poly

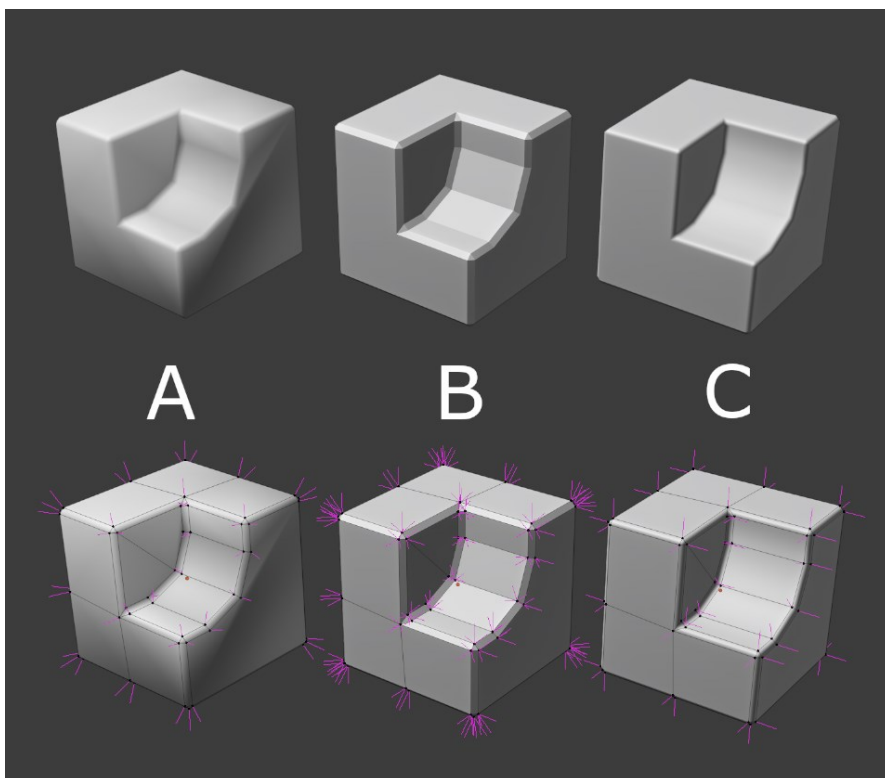
Peligrafiikassa polygonimäärä vaihtelee riippuen esimerkiksi pelialustasta, pelin tyylistä sekä asetin tärkeydestä ja käyttötarkoituksesta. High poly -mallissa voi olla jopa miljoonia polygoneja, joita manipuloidaan skulptaamalla tai muilla 3D-mallinnusohjelman työkaluilla, kuten Blenderin

Subdivision Surface -modifikaattorilla. Nämä korkeapolyiset mallit eivät kuitenkaan siirry itse peliin, sillä ne olisivat prosessointitehoja ajatellen liian raskaita, joten niiden yksityiskohdat leivotaan erillisiin, peliin meneviin vähäpolyisempiin malleihin. Medium poly -termiä käytetään, kun malli ei ole erityisen vähäpolyinen, mutta voitaisiin silti käyttää pelissä, näille malleille yleisiä ominaisuuksia ovat bevelöidyt reunat sekä saumattomat, uudelleenkäytettävät tekstuurit.

3.2 Kovapintainen mallinnus

Yksi 3D-mallintamisen osa-alueista on kovapintaisten, epäorgaanisten objektien mallinnus. Näitä kovapintaisia objekteja ovat esimerkiksi ajoneuvot, aseet sekä koneistot. Kovapintainen mallinnus vaatii erityisosaamista tarkkuudessa ja topologian käsittelyssä. Objektien mallintamisessa hyödynnetään usein high poly -mallinnuksen tekniikoita. Mallinnus keskittyy myös paljon tahkonormaalien manipulointiin eli Face Weighted Normals -tekniikkaan.

Face Weighted Normals -tekniikassa bevelöidyn mallin verteksien normaalit painotetaan osoittamaan laajemman tahkon osoittamaan suuntaan, antaen tahkolle tasaisen pinnan säilyttäen sileät reunat. Tämän avulla verteksien ei tarvitsisi jakautua ja se säästäisi verteksimäärää pelimoottorissa ja antaisi mallille pyöreämmän siluetin. Menetelmän hallitseminen takaa korkealaatuisen lopputuloksen. Kuvassa 6 esitetään tahkonormaalien manipuloinnin periaate. Malli A:n kaikki reunat ovat pehmeät ja sen verteksit ovat yhtenäisiä, mutta varjostus ei näytä hyvältä. Malli B:n reunat ovat kovia, mutta sen verteksit joutuvat jakautumaan säilyttääkseen attribuuttinsa. Malli C käyttää Blenderin Weighted Normals -modifikaattoria, jolla verteksit pysyvät yhtenäisinä ja pinta saa mieluisan varjostuksen.



Kuva 6. Kolme bevelöityä mallia sekä niiden verteksin normaalin osoittama suunta.

3.3 Leipominen

Prosessia, jossa tekstuuritiedostoon tallennetaan 3D-malliin liittyvää dataa, kutsutaan leipomiseksi. Yleinen leipomisen käyttötarkoitus on siirtää high poly -mallin yksityiskohdat yksinkertaiseen, optimoidumpaan malliin erilaisten tekstuurikarttojen muodossa. Tallennettavaan dataan kuuluu muun muassa ympäristön varjostus eli ambient occlusion, normaalit sekä geometrian reunat ja raot eli kurvatuuri. [5.]

3.4 UV-kartoitus

Listatekstuureita käyttävien mallien UV-kartoitusprosessi eroaa tavallisesta UV-kartoituksesta; koska listatekstuurien liuskat ovat ennalta määritettyjä, tulee kartoitettavan mallin UV-saarekkeet asetella teksturiin sopivaksi, usein vaatien niiden muuttamista pitkiksi tahkoliuskoiiksi. Kun taas tavallisessa UV-kartoituksessa mallin saarekkeet pyrittäisiin ensin asettelemaan UV-tilaan

mahdollisimman säästeliäästi, ilman päällekkäisyyksiä, ja teksturoimaan vasta kartoituksen jälkeen.

Kuten tavallisissakin malleissa, myös modulaarisissa malleissa tekselitiheys on tärkeää ymmärtää. Listatekstuureissa tekselitiheyden yhtenäisyys saattaa kärsiä, koska usein UV-kartoittaessa saarekkeet taivutetaan tahkoliuskoina. Yhtenäisyyttä voidaan ylläpitää valmistamalla monipuolinen listatekstuuri, jonka liuskat ovat erikokoisia, ja kartoittamalla saarekkeet niille sopiviin kokoihin.

3.5 UV-setit

Mallilla voi olla useampia eri käyttöihin tarkoitettuja UV-kartta-asetelmia. Niissä mallin UV-saarekkeet voidaan järjestää uudelleen täysin eri järjestykseen. Useampia UV-settejä voidaan käyttää esimerkiksi saumattoman pohjamateriaalin toistuvuuden rikkomiseen lisäämällä erilaisia tekstuureja muokkaamaan sävyjä tai pinnan karheusarvoja. [6.] Listatekstuureita käyttävien mallien ambient occlusion -kartta voitaisiin asettaa toiselle UV-setille, jossa saarekkeet eivät olisi asetettuina päällekkäin. Verteksien määrä kuitenkin kasvaa, jos uusien asetelmien mukana syntyy lisää saumoja.

3.6 Optimointi

Kun objekti piirretään ruudulle, prosessori lähettää näytönohjaimelle piirtokutsun. Piirtokutsut sisältävät näytönohjaimelle tarkoitetut ohjeet ja tiedot siitä, kuinka piirtää objekti. Näytönohjaimen käyttämät ohjeistukset määrittyvät objektin materiaalista ja muista parametreista. Piirrettävän objektin parametrien muuttuessa prosessorin on muodostettava uudet ohjeet ja lähetettävä uusi piirtokutsu. Liian monen parametrin muutoksen tarve ylikuormittaa prosessorin ja muodostaa pullonkaulan grafiikan piirtämisessä. Piirtokutsuja voidaan vähentää erilaisin keinoin, mutta ratkaisut voivat lisätä graafikkojen tai ohjelmoijien työn määrää. [7.]

Piirtokutsujen ryhmittäminen eli batching yhdistää samoja tekstuureja tai materiaaleja käyttäviä objekteja yhdelle piirtokutsulle, vähentäen piirtokutsujen asetusten muutoksia. Unity ryhmittää dynaamiset ja staattiset objektit omille piirtokutsuilleen, kunhan ne jakavat samat tekstuurit tai materiaalit. Staattiset objektit eivät liiku, ja ne yhdistyvät kokonaiseksi malliksi ja piirtyvät nopeammin. Dynaamiseen ryhmittämiseen soveltuvat epästaattiset pienet objektit, joista piirretään

samanlaiset verteksit yhtä aikaa. [8.] Listatekstuurien monikäyttöisyys on hyödyllinen tekijä peliympäristöjen optimoinnin kannalta, koska samoja listatekstuureja voidaan jakaa usealle objektille helpottaen piirtokutsujen ryhmittämistä.

Kun sama pikseli on mukana useammassa piirtokutsuissa ja joudutaan piirtämään useampaan kertaan, tapahtumaa kutsutaan päällekkäispiirroksi. Päällekkäispiirtoa tapahtuu yleisimmin läpinäkyvien objektien, kuten lasien tai savun piirtyessä. [7.] Läpinäkyvyysmaskia käyttävien dekaalien geometrian ääriviivat kannattaa leikellä tiiviisti näkymättömien pikselien vähentämiseksi. Vaikka ääriviivojen leikkelyssä objektien verteksimäärä kasvaakin, on se usein parempi vaihtoehto verrattuna päällekkäispiirtoon.

4 Teksturointi

Teksturoinnilla tarkoitetaan prosessia, jossa kolmiulotteiselle mallille valmistetaan pinta kaksiulotteisista kuvista. Tekstuureiden luomiseen on olemassa paljon erilaisia tekniikoita, kuten fotogrammetria, käsin maalaus joko kuvankäsittelyohjelmistolla tai suoraan 3D-malliin, noodipohjaisella materiaalieditorilla tai lepomalla pinnan ominaisuudet toisesta mallista. [9.]

Värilliset digitaaliset kuvat jakautuvat eri värikanavista: punaisesta, vihreästä, sinisestä sekä mahdollisesti alfa-kanavasta. Harmaasävykuvat tarvitsevat vain yhden kanavan, ja ne toimivat usein maskeina määrittäen erilaisten ominaisuuksien määrää tai voimakkuutta. [9.]

4.1 Fysikaalisiin ominaisuuksiin perustuva renderöinti ja tekstuurikartat

Fysikaalisiin ominaisuuksiin perustuva renderöintitekniikka eli PBR pyrkii renderöimään materiaalit jäljitellen valon vaikutusta erilaisiin pintoihin sekä niiden energiansäästöominaisuuksia oikeassa elämässä [10]. Unity tarjoaa oletuksena Standard Shader -varjostimen, joka käyttää PBR-tekniikan metallisuus-työnkulkua. Varjostimeen voidaan määrittellä materiaalin eri pinnat, kuten väri, läpinäkyvyys, metallisuus ja sileys. [11.]

Materiaali saa pohjavärinsä albedokartasta. Värien valaistus kannattaa säilyttää mahdollisimman tasaisena, koska pinnan varjot sekä korostukset saadaan muiden karttojen avulla. [12.]

Metallisuuskartta on harmaasävyinen, ja sen tarkoitus on toimia maskina ja määrittää, onko jokin materiaalin alue metallia vai ei. Tekstuuri koostuu yleensä vain mustasta tai valkoisesta, mutta joissain tapauksissa, kuten ruostetta teksturoidessa, voidaan käyttää harmaita värejä.

Sileys määrittyy myös harmaasävytekstuurista, ja se määrittelee, kuinka valo kimpoaa pinnasta. Unityn varjostimessa tekstuurin tummat arvot tarkoittavat karheampia pintoja ja vaaleat arvot sileämpiä. [13.]

Normaalikartta vaikuttaa valon heijastumisen suuntiin, ja luo illuusion yksityiskohdista, kuten kohoamista tai painaumista materiaalin pinnalla [14]. Materiaalin läpinäkyvät alueet saadaan läpinäkyvyyskartalla, joka käyttää varjostimessa albedokartan alfa-kanavaa [15].

4.2 Kanavien pakkaaminen

Kanavien pakkaamisella tarkoitetaan toimenpidettä, jossa yhdistetään eri karttojen harmaasävykuvat yhden tekstuurin värikanaville, säästäten tekstuurimuistia. Toisinaan materiaalin varjostin voi käyttää eri kanavien harmaasävytekstuureja myös maskina eri ominaisuuksien, kuten erilaisten äänien tai fysiikkainformaation määrittämiseen mallin pinnalla. Jokaisen kanavan voi myös konfiguroida käyttämään omia UV-settejänsä. Kanavien pakkaaminen voi kuitenkin johtaa varjostimen monimutkaistumiseen ja lisätä piirtokutsuja. [16.]

5 Listatekstuurin valmistus ja peliympäristön rakentaminen

Opinnäytetyön käytännön osuudessa suoritettiin projekti, jossa suunniteltiin ja valmistettiin listatekstuuri sekä muita saumattomia tekstuureja, jonka jälkeen mallinnettiin ja UV-kartoitettiin tekstuureihin sopivat peliobjektit. Lopuksi mallit tuotiin tekstuureineen pelimoottoriin, jossa rakennettiin ympäristö valmiita objekteja käyttäen.

5.1 Tekstuurien suunnittelu

Projektityö alkoi käytettävien materiaalien suunnittelusta. Projektissa luotaisiin scifi-aiheinen ympäristö, johon tulisi yksinkertaista tarinankerrontaa proppien sekä valaistuksen avulla. Materiaalit koostuisivat enimmäkseen puhtaasta valkoisesta maalipinnasta, muutamista eri metallipinnoista sekä toisesta, värillisestä maalipinnasta. Päädyttiin yhteen listatekstuuriin ja yhteen kokonaan saumattomaan tekstuuriin, josta luotaisiin pelimoottorissa kopio, jonka väriä voitaisiin vaihtaa editorissa listatekstuurin värillisiin osiin sopivaksi. Listatekstuuri jaettaisiin yhdeksään osaan: viisi pientä, kaksi keskikokoista sekä yksi isompi liuska, loput tekstuuritilasta varattaisiin dekaaleille.

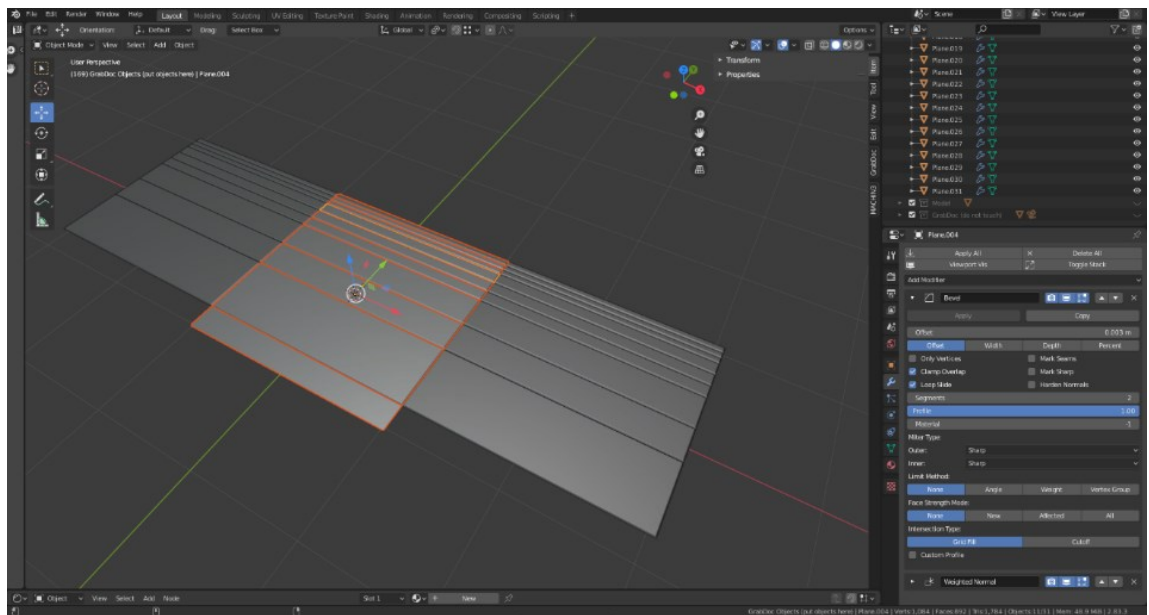
Listat suunniteltiin ilman suurempia tavoitteita niiden käyttötarkoitukselle; annettaisiin listatekstuurin tarjoaman vapauden ohjata objektien valmistumista. Tekstuurien kooksi tulisi 2048 kertaa 2048 pikseliä. Tekselitiheydelle ei asetettu tiettyä arvoa, jota hakea, vaan tärkeintä oli säilyttää tiheys suurin piirtein arvojen 10,24 px/cm ja 5,12 px/cm välillä. Harjoittelulla ja listojen kokojen tarkalla suunnittelulla voitaisiin saavuttaa paremmat tekselitiheydet.

5.2 High poly -mallin valmistus

Listatekstuurin valmistus aloitettiin luomalla high poly -malli listatekstuurista. Linjaston ensimmäinen osa oli Blender. GrabDoc-lisäosan avulla luotiin näppärästi asetelma, jolla pystyttiin esikatselmaan ja leipomaan eri tekstuurikarttoja sekä viemään ne toisiin ohjelmiin. Asetelmassa oli kamera sekä taso, jota vasten high poly -mallit leipoutuisivat. Kameran avulla tekstuurikarttojen esikatselu tapahtui reaaliajassa. Lisäosalla määritettiin tekstuurien resoluutio sekä säädeltiin tekstuurien ominaisuuksia, kuten ambient occlusionin voimakkuutta.

Asetelman lisäksi luotiin uusi kahden neliömetrin taso, johon lisättiin bevel- sekä weighted normals -modifikaattorit. Tason varjostus vaihdettiin myös sileäksi, ja asetettiin auto smooth -toiminto 60 asteeseen. Bevelin sijaan olisi voitu käyttää subdivision surface -modifikaattoria, mutta se olisi vaatinut erilaisen lähestymistavan reunojen asettelussa geometriaan. Bevel riitti tähän tarkoitukseen.

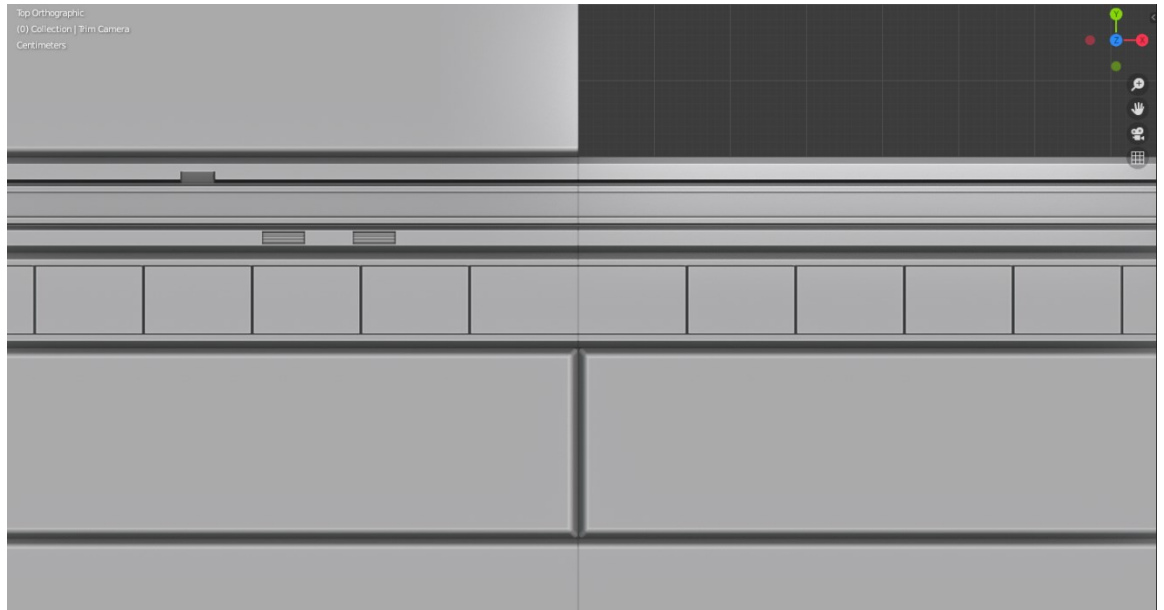
Taso pilkottiin listoiksi, jonka jälkeen ne irrotettiin omiksi objekteiksi. (kuva 7). Modifikaattorit sekä muu objektidata, kuten auto smooth -arvot, kopioituisivat irrotettuihin objekteihin. Sen jälkeen luotiin listoille niiden molemmille sivuille dynaamiset kopiot eli instanssit. Ne eroavat tavallisista kopioista siten, että ne päivittyvät alkuperäisen objektin päivittyessä. Koska instanssit sijaitsevat GrabDocin leivonta-alueen ulkopuolella, ne eivät itse näy leivottavissa kartoissa, mutta niiden tarkoitus on vaikuttaa listojen toistuvuuteen tekstuuriin leivonnan aikana. Esimerkiksi ambient occlusion -kartta hyötyy alueen ulkopuolella vaikuttavista objekteista.



Kuva 7. Taso on pilkottu listoiksi, ja niistä on luotu instanssit objektin molemmille puolille.

Listatekstureille hyödyllinen 45 asteen normal bevel saatiin helposti Blenderin inset-toiminnolla. Toiminnon paksuus- ja syvyysarvoiksi annettiin 0.01. Luvuilla ei ole väliä, kunhan molemmat arvot ovat samat. Sitten valittiin molemmat säätimet samanaikaisesti ja skaalattiin sopivaksi. Inset-toiminnon syvyysarvon muuttaminen negatiiviseksi skaalauksen jälkeen kääntäisi geometrian tarvittaessa sisäänpäin. Kuvassa 8 on esimerkki siitä, jos ei haluttaisi joidenkin listojen päätyihin inset-toiminnon muodostamaa rakoa, vaan toivottaisiin listan toistuvan saumattomasti. Tämä korjattaisiin poistamalla päätyjen viistot tahkot ja paikkaamalla poiston mukana muodostuneet reiät

tasoittamalla niiden päällimmäiset verteksit sopivaksi. Toinen tapa saumattoman toistuvuuden toteutukseen olisi mallintaa haluttu lista tarpeeksi leveäksi, jotta se ylttäisi leivonta-alueen ulkopuolelle molemmin puolin. Koska käytössä oli instanssit, päädyttiin ensimmäiseen ratkaisuun, jotta ei tarvitsisi säätää kyseisen listan instanssien paikkaa mahdollisen overlapiin eli päällekkäisyyden välttämiseksi.

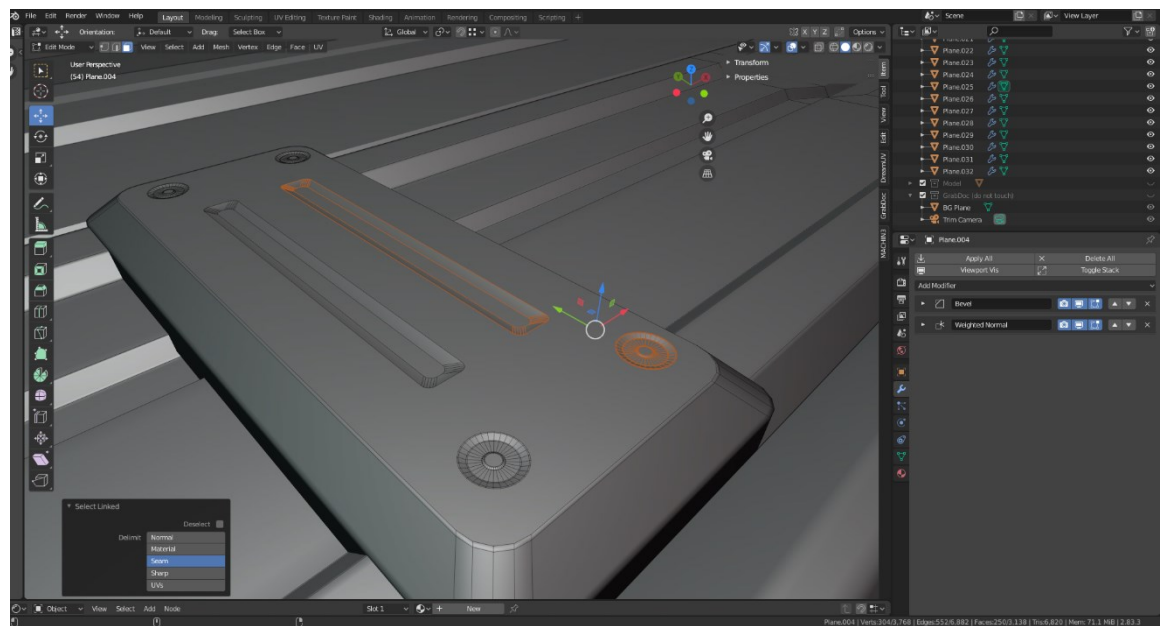


Kuva 8. Inset-toiminnolla luotu 45 asteen bevel käytössä. Ylemmät kaksi pienempää liuskaa ovat ilman pystysuoraa saumaa ja alempi liuska sen kanssa.

Mallintaessa huomattiin, etteivät kameran kanssa täysin yhdensuuntaiset tahkot näkyneet kunnolla normaalikarttaa esikatsellessa. Ongelma ratkaistiin lisäämällä tahkoihin viistoutta. Liuskojen sisällä olevan geometrian tahkojen kulmien ei tarvitse olla juuri 45 astetta, kuten liuskojen ulkoreunoissa normal bevel -efektiä tavoitellessa, vaan riittää, etteivät ne ole täysin yhdensuuntaiset kameran kanssa.

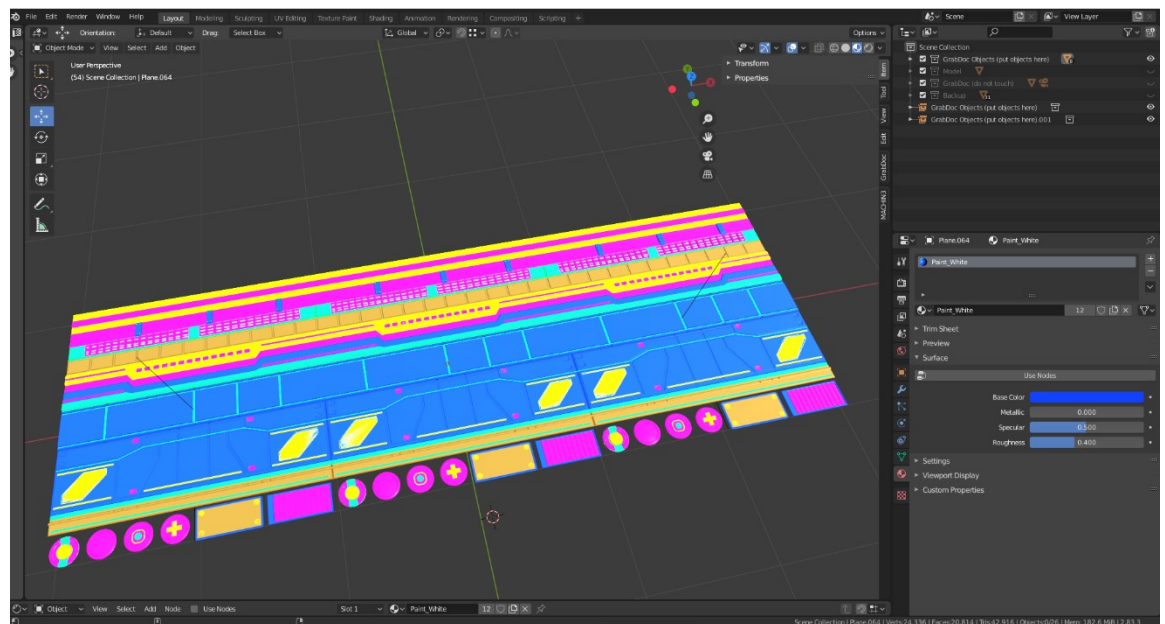
Yksi high poly -mallinnuksen tekniikoista on lisätä yksityiskohtia, kuten painaumia, kohoumia ja ruuvien koloja käyttämällä pintojen päällä leijuvia malleja, floatereita. Ne säästävät paljon aikaa, koska topologiasta ei juurikaan tarvitse huolehtia. Mutta koska floaterit vaikuttavat kuitenkin korkeuskarttaan, painaumien korkeammat reunat voivat muodostaa epätoivotun lopputuloksen, kannattaa siis suosia kohoumia painaumien sijaan tekniikkaa käyttäessä. [17.] Kuvassa 9 lisättiin muutamia painaumia ja kohoumia geometrialla, mutta vastaavien yksityiskohtien asettelu jatkuisi

myöhemmin teksturointivaiheessa. Leijuvalle geometrialle pystyttäisiin myös määrittelemään omat värit, joita hyödynnettäisiin materiaalien erottelussa Substance Painterissa.



Kuva 9. Floatereita pinnan päällä.

Kun listat ja dekaalit oli mallinnettu, jaettiin listojen eri osat materiaalien mukaan. Luotiin eri pinnoille omat materiaalit, joilla oli kirkkaat sekä toisistaan erottuvat värit, ja asetettiin ne listoille (kuva 10). Värien erottuvuus oli tärkeää, koska ne leipoutuisivat väri-ID-kartalle, josta luotaisiin maskit materiaaleille. Jos värit olisivat liian samankaltaisia, maskeihin voisi syntyä epätoivottuja tuloksia eri materiaalien vuotamisen muodossa.



Kuva 10. Listojen osat pilkottu materiaalien mukaan, ja niille annettu sen mukainen väri.

Valittiin GrabDoc-työkalulla vientikansio ja säädeltiin tekstuurikarttojen leipomiset kuntoon. High poly -malli olisi valmis leivottavaksi. Tekstuurien lisäksi valmistettiin kuuden neliömetrin low poly -taso, joka paloiteltiin yhdeksän osan ruudukoksi. Ruudukon tahkot kartoitettiin UV-tilaan siten, että jokainen tahko täyttäisi koko UV-tilan. Kaikki paitsi keskimmäisen tahkon saarekkeet siirrettiin UV-tilan ulkopuolelle tasan yhdellä yksiköllä x-akselilla. Lopuksi ruudukko vietiin Substance Painteriin fbx-muodossa. Taso toimisi pohjamallina, johon Blenderissä leivotut kartat lisättäisiin Substance Painterissa. Ruudukko helpottaisi esikatselua teksturointivaiheessa, sillä listatekstuurin toistuvuus ja mahdolliset saumattomuutta häiritsevät elementit nähtäisiin paremmin.

5.3 Teksturointi

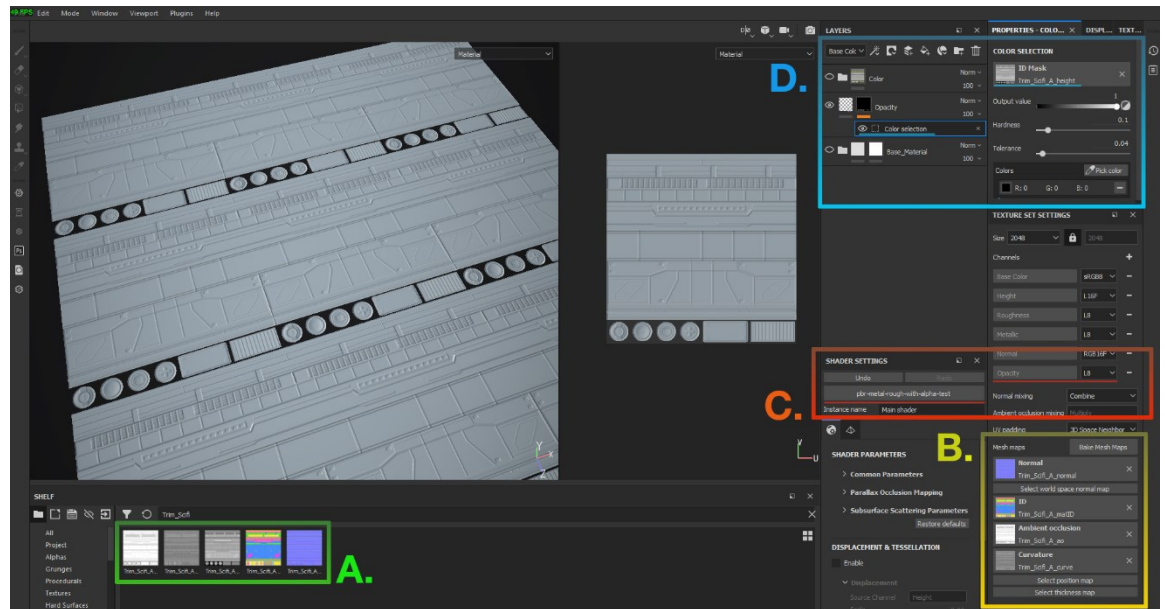
High poly -mallin leipomisen jälkeen oli teksturoinnin vuoro. Leivotut kartat vietäisiin Substance Painteriin, joista muodostuisivat lopulliset tekstuurit. Uudet kartat siirtyisivät teksturoinnin jälkeen takaisin Blenderiin UV-kartoitusta varten, sekä Unityyn materiaalien muodostamiseksi.

Luotiin uusi Substance Painter -projekti Unity 5 -asetuksilla. Asetukset tarjosivat neljä eteenpäin vietävää tekstuurikarttaa, jotka oli pakattu Unityn Standard Shader -varjostimelle sopivaksi: väriläpinäkyvyyskartta, metallisuus-karheuskartta, normaalikartta sekä emissiokartta. Emissiokartan voisi myöhemmin vientivaiheessa ottaa pois päältä, jos sille ei olisi käyttöä. Pohjamalliksi valittiin ruudukoksi jaettu low poly -taso, projektin resoluutioksi 2048, sekä normaalikartan formaatiksi Unityn käyttämän OpenGL:n.

Toinen formaattivaihtoehto on DirectX, siinä normaalikartan vihreä väri on käänteinen. Jos normaalikartan kohoumat näyttävätkin painaumilta tai toisinpäin, kannattaa kokeilla vihreän kanavan päinvastoin kääntämistä. [18.]

Leivotut kartat tuotiin projektiin raahaamalla ne tuontikansiosta Painteriin. Tiedostojen muodoksi valittiin texture. (Kuva 11. A.) Tämän jälkeen tasolle määritettiin normaali, väri-ID-, ambient occlusion- ja curvature-kartat Mesh maps -osiossa. (Kuva 11. B.) Low poly -malliin ilmestyi leivotut yksityiskohdat. Läpinäkyvyys dekaaleita varten saavutettiin lisäämällä tekstuurisettiasetuksissa läpinäkyvyyskanava ja vaihtamalla varjostin pbr-metal-rough-with-alpha-test-tilaan. Se piilottaisi tietyn alfaarvojen alittavat pikselit. (Kuva 11. C.) Seuraavaksi luotiin uusi taso, joka täytettiin täysin läpinäkyväksi ja luotiin sille maski värivalitsimella korkeuskartan mustaa väriä käyttäen.

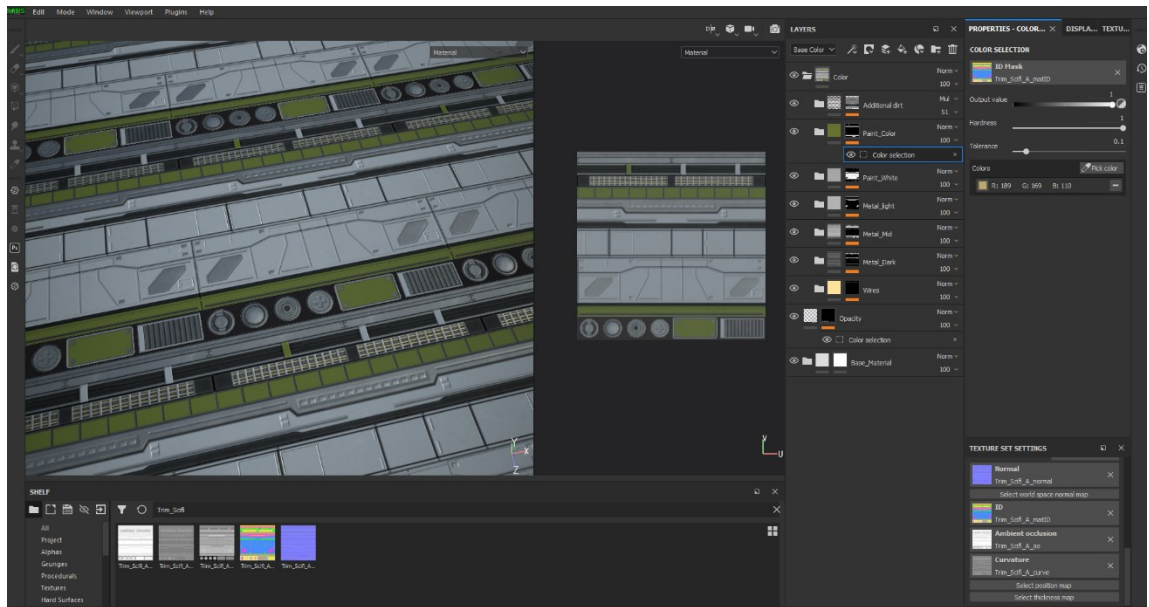
Tämä taso muutti kaikki sen maskiin sisältyvät pikselit läpinäkyviksi. Korkeuskartta sopi valit-
simeen hyvin, koska kaikki listatekstuurin geometria oli hieman GrabDoc:n pohjalevyä korkeam-
malla, ja musta väri erottui selkeästi. (Kuva 11. D.)



Kuva 11. Substance Painter -projektin valmistelu.

Kuvassa 12 luotiin jokaiselle materiaalille oma tasokansio, jolle määritettiin oma maski värivalit-
simella. Väri valittiin väri-ID-karttaa käyttäen. Itse materiaalit muodostuivat Painterin valmiiden
materiaalien muokkaamisella ja yhdistämisellä proseduraalisten maskien sekä sekoitustilojen
avulla. Kun liuskat ja dekaalit oli teksturoitu, oli listatekstuurin ensimmäinen iteraatio valmis vie-
täväksi Unity-projektiin tarkasteltavaksi mahdollisten muutosten varalta.

Saumattoman tekstuurin valmistus tapahtui samalla low poly -tasolla, mutta ilman high polyn yk-
sityiskohtia. Materiaaliksi tehtiin vaalea maalipinta, jonka karheudessa huomaisi toistuvan kuvi-
oinnin.



Kuva 12. Substance Painterissa teksturoitu low poly -malli.

5.4 Materiaalien luonti Unityssä ja Blenderissä

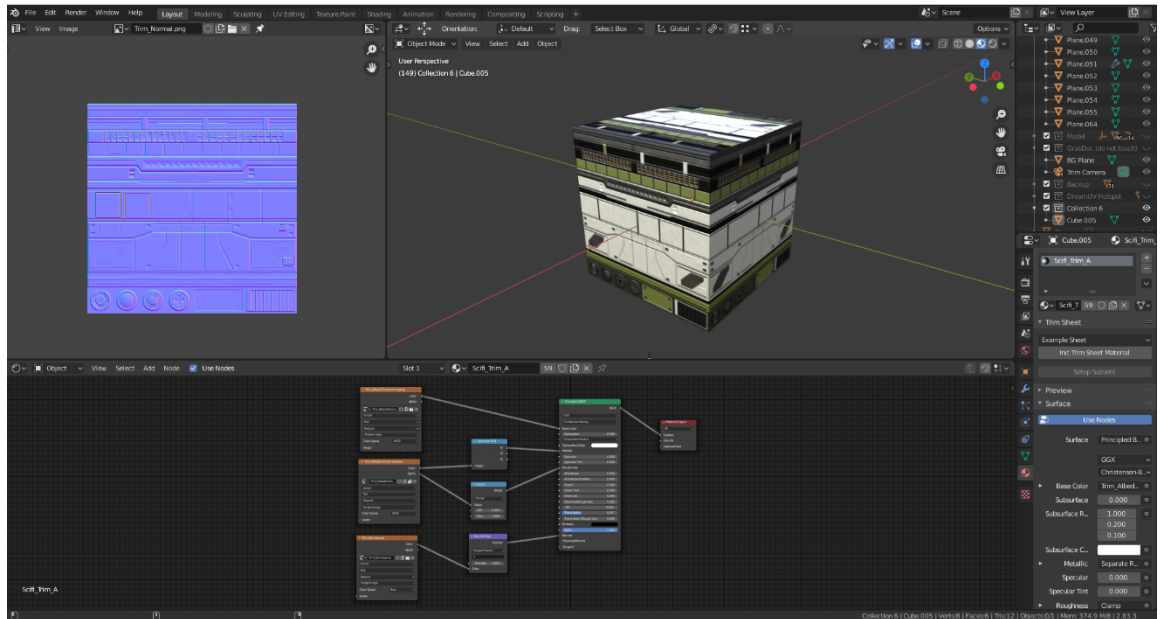
Aloitettiin uusi Unity-projekti, versiona 2019.3, renderöintitekniikkana Built-In Render Pipeline. Luotiin projektin Assets-kansioon alakansiot materiaaleille sekä malleille. Kun kansiot oli tehty, vietiin Painterista tekstuurit suoraan materiaalikansioon. Tämän jälkeen Unityssä luotiin uusi materiaali, varjostimena Standard, renderöintitilana Cutout. Cutout-tila mahdollistaisi läpinäkyvyyden dekaalien ympärille. Alpha Cutoff -arvoksi annettiin 0.5, väri-läpinäkyvyyskartta, metallisuuskarheuskartta sekä normaalikartta asetettiin omille paikoilleen. Listatekstuurin materiaali olisi valmis. Saumaton materiaali valmistui samalla tavalla, sekä siitä luotiin toinen versio eri sävyssä materiaalin värisäädintä käyttämällä.

Kuvassa 13 materiaalien esikatselua varten tehtiin uusi skene, jonne asetettiin kuutio, lattia, muutamia valoja sekä reflection probe eli heijastusluotain. Kuutiolle annettiin listatekstuurmateriaali sekä lattialle saumattoman materiaalin molemmat sävyt. Kameraan lisättiin vielä jälkikäsitteilykomponentti. Huomattiin, että materiaalin karheusarvot olivat liian matalia, ja kuution pinta näytti tarpeettoman kiiltävältä. Siirryttiin takaisin Painteriin ja säädettiin karheusarvoja sopivaksi. Tämän jälkeen tekstuurit vietiin uudestaan Unityyn, ja ne päivittyivät automaattisesti.



Kuva 13. Unity-projektin materiaalien esikatselua varten tehty skene, ja valmis listatekstuurimateriaali.

Blenderissä luotiin myös uudet materiaalit. Varjostimena toimi Principled BSDF, johon yhdistettiin samat, Unity-projektiin tuodut tekstuurikartat. Kuvassa 14 värikartta yhdistettiin suoraan master-noodiin, metallisuus saatiin jakamalla metallisuus-karheuskartan värikanavat ja käyttämällä punaista kanavaa, karheuskartta saatiin viemällä metallisuus-karheuskartan alfakanava Clamp-noodin läpi, normaalikartan piti kulkea erikseen Normal Map -noodin kautta. Noodien arvot eivät tuottaneet täysin samaa lopputulosta kuin Unityssä, mutta sillä ei ollut väliä. Tärkeintä oli, että pystyisi mallinnus- ja UV-kartoitusvaiheissa näkemään listojen rajat ja saamaan käsitys kokonaisuudesta.



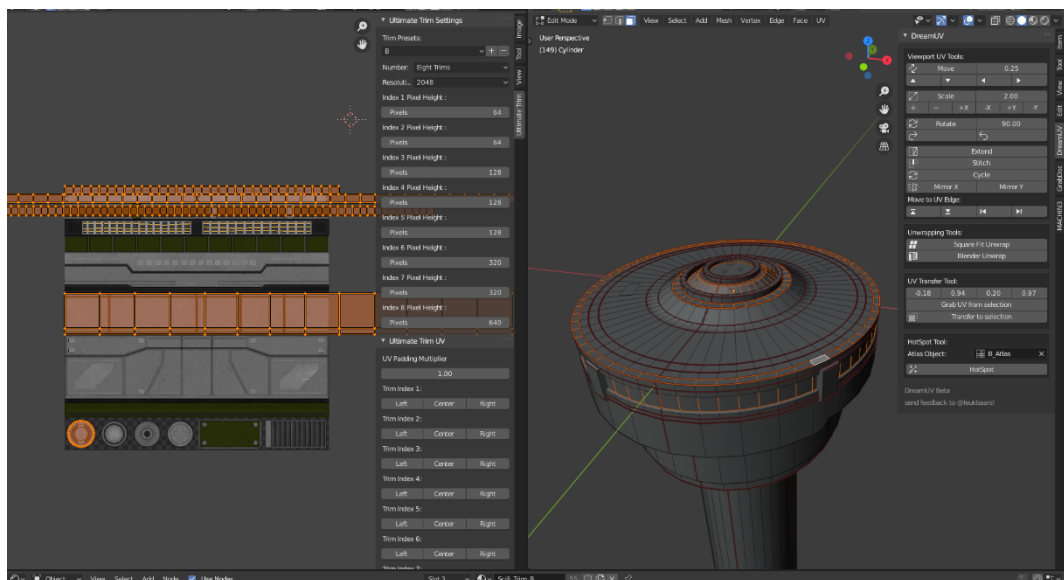
Kuva 14. Tekstuurit tuotuna Blenderiin. Noodeilla valmistettu materiaali kuvasi suurin piirtein lopputulosta.

5.5 Mallinnus ja UV-kartoitus

Valmistusprosessin seuraavassa vaiheessa mallinnettiin kokoelma aseteista, jotka UV-kartoitettiin tekstuureille sopiviksi. Osa malleista olisi uudelleenkäytettäviä, kuten kaiteet, putket, laatikot ja säiliöt. Laajemmat assetit muodostaisivat ympäristölle seinät ja katon. Mallit rakentuivat huomioiden tasaisten tahkoliuskosten säilyvyyden. Tämä helpottaisi kartoitusprosessia sekä ehkäisisi tekstuurien vääristymistä. Samalla pidettiin mielessä myös listatekstuurin eri listojen koot. Saumat määritettäisiin lopussa sopiviin kohtiin.

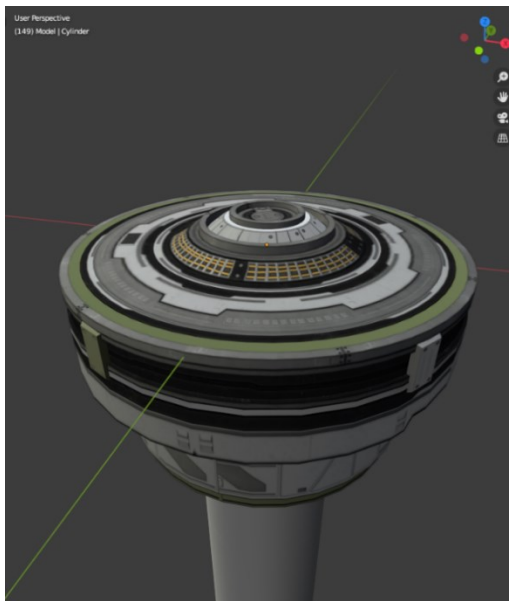
Mallien UV-kartoitusprosessi tapahtui pääsääntöisesti samanlaisella kaavalla, mutta erilaisia lähestymistapoja piti käyttää tarvittaessa. Blenderin oma tavallinen Unwrap-toiminto antoi harvoin halutun tuloksen, varsinkin kun kyseessä oli kaarevat pinnat; kartoitetut UV-saarekkeet eivät olleet suorita linjoja, vaan olivat hieman kaarevia. Tähän ongelmaan ratkaisuksi toimi vaihtoehtoinen kartoitustoiminto, UV Reset ja Follow Active Quads. Toimintojen tarkoitus on levittää valitut tahkot aktiivisen tahkon vierelle UV-tilassa siten, että ne ovat saman kokoisia, muodostaen tasaisen UV-saarekeliuskan. Ilman UV Reset -toimintoa tahkot yleensä vääntyivät väärin. Yksi haittapuoli tekniikassa on kuitenkin ilmennyt; liuskojen tahkot joskus kartoittuvat liian kapeiksi, ja niitä joutui skaalaamaan leveys suunnassa sopivaksi.

Jotta liuskat saataisiin tehokkaasti oikeille kohdilleen UV-tilassa, käytettiin Ultimate Trim UV -lisäosaa. Sen avulla pystyttiin määrittämään listatekstuurin liuskojen määrä ja koko sekä siirtämään saarekkeet halutulle paikalle. Lisäosa myös skaalasi saarekkeet listaan sopivaksi, mutta piti huomioida Follow Active Quads -tekniikan muokkaama saarekkeiden leveys. Ultimate Trim UV antoi mahdollisuuden myös tallentaa asetelmia, eikä asetuksia tarvinnut määrittää joka kerta uudestaan Blender-projektia avatessa. Työssä käytettiin myös apuna DreamUV-lisäosaa. Sen lukuisat ominaisuudet tehostivat kartoitusprosessia huomattavasti. Eniten projektissa käytetyt toiminnot olivat saarekkeiden 90-asteen kääntäminen ja peilaus sekä UV-koordinaattien kopiointi ja liittäminen toisiin saarekkeisiin. Samat toiminnot olivat mahdollisia käyttäen Blenderin oletustoimintojakin, mutta DreamUV säästi pikanäppäimillään paljon aikaa sekä hermoja. (Kuva 15.)

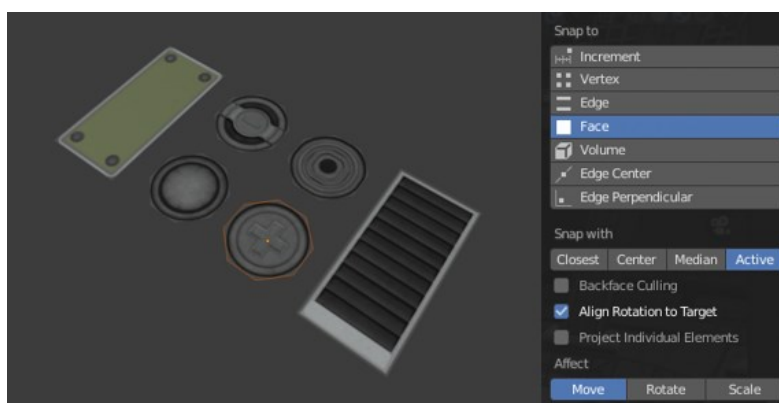


Kuva 15. Mallin UV-kartoitus tapahtui pääsääntöisesti Ultimate Trim UV- sekä DreamUV-lisäosia käyttäen.

Ensimmäinen malli oli tehtynä (kuva 16). Se tulisi olemaan skenen keskipisteenä. Mallille pystyi määrittämään useampia materiaaleja sen eri tahkoille. Käytettyjen materiaalien määrä ja niitä käyttävien tahkojen informaatio siirtyisi Unity-projektiin objekteja vietäessä. Tekstuurissa näkyvä vihreä rengas käyttää toista saumatonta tekstuuria listatekstuuriin sijaan. Mallin pinnalle myös muutamia dekaaleita. Kuvassa 17 leikeltiin dekaaleille sopivat mallit, irrotettiin ne omiksi objekteiksi ja nostettiin niitä hieman ylöspäin niiden pivot-pisteestä. Sitten niistä pystyttiin luomaan instansseja ja kiinnittämään se toisiin pintoihin kiinni tahkojen suuntaisesti hyödyntämällä Blenderin kiinnitysasetuksia. Myöhemmin dekaalit voitaisiin yhdistää pohjaobjektiin.

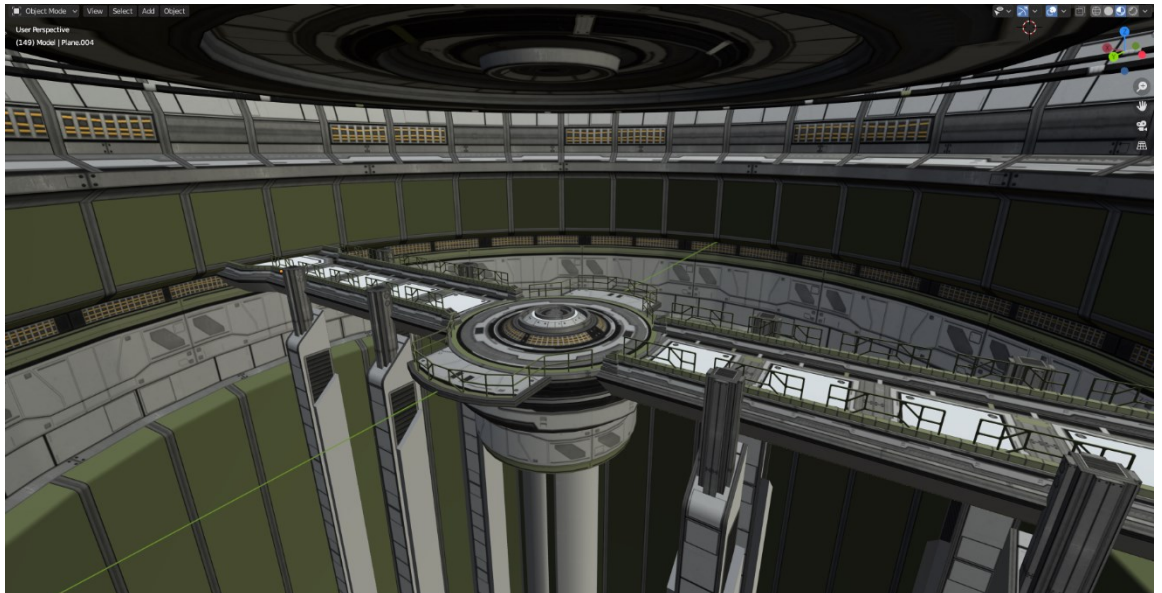


Kuva 16. Ensimmäinen UV-kartoitettu malli sijaitisi ympäristön keskiössä Unity-skenessä.



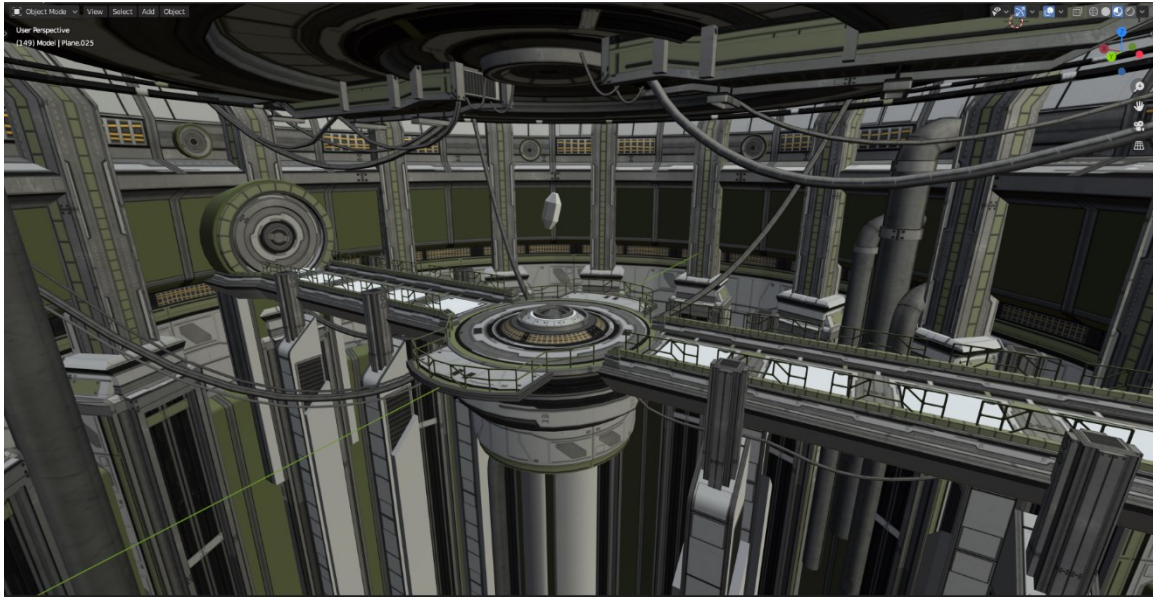
Kuva 17. Dekaaaliobjektit muodostuivat kartoittamalla pieniä tahkoja teksturiin sopiviksi, ja niitä aseteltaisiin mallien pinnoille kiinnitysasetusten avulla.

Seuraavat mallit olivat seinät, katto, keskipisteelle johtavat sillat sekä niiden pylväät ja kaiteet. Kaiteista ja pylväistä mallinnettiin vain yhdet objektit, joita sitten kopioitiin instansseiksi paikoilleen. Sillat ja keskipisteen alustat mallinnettiin peilaus-modifikaattoria käyttäen. Iso osa lattiaista ja seinistä käyttäisi saumattomia materiaaleja tasapainoksi yksityiskohtaiselle listatekstuureille (kuva 18).



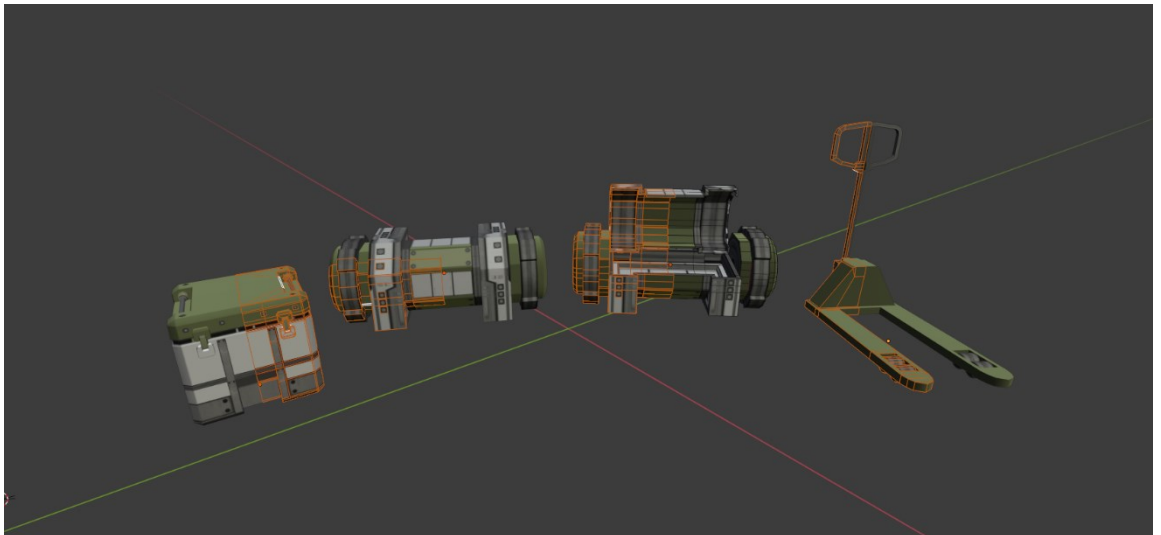
Kuva 18. Ympäristön suuremmat elementit käyttävät saumattomia tekstuureja tasapainottaakseen listatekstuurin yksityiskohtaisuutta.

Seuraavaksi lisättiin kaapeleita, putkia, ovet sekä pylväitä seinille. Roikkuvat kaapelit valmistuivat käyttäen Array- ja Curve-modifikaattoreita, seinien pylväät käyttivät Array-modifikaattorin Object Offset -toimintoa, jolla kohdeobjektia pyörittäen pylväät toistuivat muodostaen kehän. Samalla skenen tarinankerronta alkoi kehittyä; halli olisi energiakristallien latausasema, jossa tyhjät kristallit käytäisiin täyttämässä. Ympäristö vaikutti Blenderissä todella kiireiseltä, mutta kun se vietiisiin Unityyn ja lisättäisiin valaistus ja jälkikäsittely, saataisiin säilytettyä maiseman selkeys (kuva 19). Vielä tarvittaisiin kuljetukseen käytettäviä proppeja. Kristallien latausefekti luotaisiin Unityn puolella partikkeleilla ja valoilla.



Kuva 19. Ympäristö muuttui hektisemmäksi uusien objektien valmistuessa.

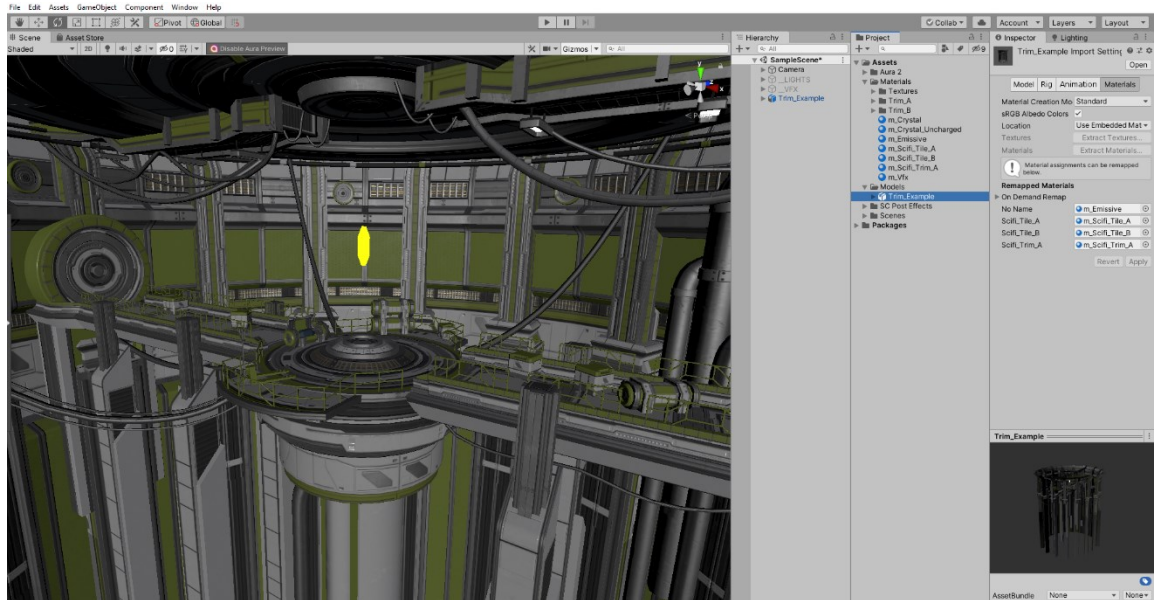
Propit valmistuivat käyttäen hyödyksi peilaus-modifikaattoria. Kovapintainen ulkonäkö ja pyöreät reunat saavutettiin tahkonormaalien manipuloinnilla sekä geometrian lisäämisellä beveliä käyttäen. Mallit käyttivät enimmäkseen saumattomia tekstuureja. Listatekstuureja käytettiin eri pintojen erottelukseen. (Kuva 20.) Proppeja aseteltiin ympäristöön tarinankerronnan mukaisesti.



Kuva 20. Propeilla oli tavallista suurempi polymäärä pyöreiden reunojen takia.

Ympäristö tuotiin Unityn puolelle yhtenä fbx-tiedostona. Objektit olivat silti irrallaan toisistaan ja niitä pystyisi liikuttamaan vapaasti. Fbx-tiedoston asetuksissa määritettiin mallien materiaalit kohdilleen. Fbx-tiedoston siirtyessä skeneen kaikki sen objektit säilyivät samoilla paikoilla kuin

Blenderissä. (Kuva 21.) Värit olivat hieman kylläisemmät kuin Blenderissä, muuten kaikki näytti olevan kunnossa. Saturaatio voitaisiin korjata jälkikäsittelykomponenteilla. Energiakristallille, lampuille ja partikkeleille luotiin omat materiaalit, joissa käytettäisiin emissiivisyyttä. Tämä antaisi niille enemmän hohtoa jälkikäsittelyn bloom-tehosteen renderöityessä.



Kuva 21. Fbx-tiedostolle asetettiin materiaalit ja siirrettiin skeneen.

Kuvassa 22 lopullinen näkymä, johon lisättiin lukuisia jälkikäsittelytehosteita, tärkeimpinä bloom sekä väritasapaino. Skenen tärkeimmät alueet alkoivat erottua paremmin ympäristöstä, kun silloille ja keskipisteelle kohdistettiin kirkkaimmat valot samalla tummentaa ympäristön kokonaisvalaistusta. Kristallille luotiin sähköinen efekti Unityn Shuriken-partikkelisysteemin trail- ja noise-moduuleja käyttäen, hiukkaset tehtiin säätämällä velocity over lifetime- ja noise-moduuleja. Volumetrinen valaistus saavutettiin käyttämällä Aura 2 -lisäosaa; sen avulla lisättiin myös sumua ympäristön pohjalle.



Kuva 22. Lopullinen maisema. Monet jälkikäsittelykomponentit tuovat ympäristöön paljon tunnelmaa.

6 Yhteenveto

Opinnäytetyön tarkoituksena oli pureutua listatekstuurien valmistuksen ja käyttöönoton eri vaiheisiin keskittyen niiden teknisiin menetelmiin ja kehittämään teksturointiprosessia helpottava työkulku. Listatekstuurit olivat minulle ennestään jo tuttuja, mutta opinnäytetyön aikana tekemäni tutkimus sekä käytännön osuus auttoivat minua löytämään parempia ja tehokkaampia ratkaisuja koko valmistusprosessille.

Vaikka kovapintainen mallinnus on minulle mielenkiintoinen aihe, en high poly -mallin valmistukseen käyttänyt liikaa aikaa, koska pyrin keskittymään kokonaisprosessin läpikäymiseen. Listatekstuurin aiheeksi olisi voinut käydä myös jokin muu kuin scifi, mutta silloin valmistusprosessiin olisi mahdollisesti tullut lisäksi skulptausvaihe. Scifiteema tarjosi minulle mahdollisuuden olla luovempi ja vapaampi liuskojen suunnittelussa sekä peliobjektien kartoittamisessa.

Työssä kehitetty työkulku osoittautui tehokkaaksi; yhteys Blenderin, Painterin sekä Unityn välillä oli todella sulava materiaaleja valmistaessa sekä muokatessa. Niin pienet kuin isotkin muutokset high poly -mallissa leipoutuivat nopeasti uudestaan GrabDoc-lisäosan avulla, Painterin proseduraaliset maskit päivittyivät perustuen uusiin high poly -muutoksiin vain muutamalla klikkauksella sekä Unityn materiaalit saivat muutokset nopeasti tekstuurien päivittyessä. Yhdistelemällä Blenderin ja sen lisäosien toimintoja, samalla käsin hienosäätäen, UV-kartoitus sujui tehokkaasti ja helposti. Valmistamani Unity-skene onnistui mielestäni hyvin; ympäristöstä saisi otettua useampia kuvia eri kuvakulmista portfoliooni. Samaa valmistusprosessia käyttämällä pystyisin tulevaisuudessa luomaan tehokkaasti uusia skenejä sekä hyödyntämään säästettyä aikaa materiaalien laadun parantelussa.

Listatekstuurien käyttö on varsin monipuolista, ja omalla luovuudella voi keksiä tekstuureille paljon eri käyttötapoja. Opinnäytetyön aihetta tutkiessa ja käytännön osuutta valmistaessa päädyin usein purkamaan oikean elämän rakennuksia mielessäni ja pohtimaan, millaisista liuskoista eri rakennusten osat koostuivat, ja kuinka valmistaa kyseisen rakennuksen tekstuurit optimaalisesti.

Lähteet

1. Robert Roeder. Arabian Afternoon: Working with Trim Sheets Efficiently. 2020; <https://80.lv/articles/arabian-afternoon-working-with-trim-sheets-efficiently/> Luettu 16.11.2020.
2. Lea Kronenberger. Balancing Modularity And Uniqueness In Environment Art. 2020; <https://www.beyondextent.com/articles/balancing-modularity-and-uniqueness-in-environment-art> Luettu 14.1.2021.
3. Morten Olsen. The Ultimate Trim: Texturing Techniques of Sunset Overdrive. 2015; <https://www.gdcvault.com/play/1022323/The-Ultimate-Trim-Texturing-Techniques> Katsottu 14.1.2021.
4. Leonardo Lezzi. Normal Edge Decal Tutorial – Game Environment. 2018; <https://www.artstation.com/artwork/l3wwa> Luettu 16.11.2020.
5. What is Baking? <https://docs.substance3d.com/bake/what-is-baking-172818449.html> Luettu 28.11.2020.
6. The Dinusty Empire. Basics of Multi UV Sets. 2020; https://www.youtube.com/watch?v=SybpVltvwN0&ab_channel=TheDiNustyEmpire Katsottu 6.12.2020.
7. Keith O’Conor. GPU Performance for Game Artists. 2017; <http://fragmentbuffer.com/gpu-performance-for-game-artists/> Luettu 17.11.2020.
8. Draw call batching. <https://docs.unity3d.com/Manual/DrawCallBatching.html> Luettu 23.7.2020.
9. Texturing. <http://wiki.polycount.com/wiki/Texturing> Luettu 28.11.2020.
10. The PBR Guide – Part 2. <https://academy.substance3d.com/courses/the-pbr-guide-part-2> Luettu 14.1.2021.
11. Metallic vs. specular workflow <https://docs.unity3d.com/Manual/StandardShaderMetallicVsSpecular.html> Luettu 28.11.2020.

12. Thomas Denham. Texture Maps: The Ultimate Guide For 3D Artists. <https://conceptartempire.com/texture-maps/> Luettu 28.11.2020.
13. Smoothness. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterSmoothness.html> Luettu 28.11.2020.
14. Normal map (Bump mapping) <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterNormalMap.html> Luettu 28.11.2020.
15. Transparency map http://wiki.polycount.com/wiki/Transparency_map Luettu 28.11.2020.
16. ChannelPacking <http://wiki.polycount.com/wiki/ChannelPacking> Luettu 17.11.2020.
17. Hai Phan. Essentials, EP 3: Normal Map Floaters. 2015; <https://www.youtube.com/watch?v=ny-M4QRGFN8> Katsottu 14.1.2021.
18. What is the difference between the OpenGL and DirectX normal format? <https://docs.substance3d.com/bake/what-is-the-difference-between-the-opengl-and-directx-normal-format-182256965.html> Luettu 3.12.2020.