



3D-grafiikan tuottaminen lisätyn todellisuuden kokemukseen

Joonas Pauni

OPINNÄYTETYÖ
Joulukuu 2019

Tietojenkäsittely
Pelituotanto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Pelituotanto

PAUNI, JOONAS:
3D-grafiikan tuottaminen lisätyn todellisuuden kokemukseen

Opinnäytetyö 55 sivua, joista liitteitä 2 sivua
Joulukuu 2019

Opinnäytetyön toimeksiantaja oli Eligo.Studio Oy, jonka toiminta keskittyy pääasiassa lisätyn todellisuuden ratkaisuiden tarjoamiseen muun muassa opetus-käyttöön ja markkinointiin niin teollisuuden kuin viihteenkin aloille. Tavoitteena oli selvittää, miten kolmiulotteista grafiikkaa kannattaa tuottaa lisätyn todellisuuden kokemukseen ja mitä käytäntöjä toteutuksessa on syytä käyttää. Tarkoituksena oli toteuttaa grafiikat mobiililaitteilla käytettävän Facebook- tai Instagram-sovel-luksen kameraefektiin.

Kolmiulotteisten mallien tekemiseen käytettiin 3ds Max -ohjelmistoa, josta valmiit mallit siirrettiin Substance Painter -teksturointiohjelmaan pintojen yksityiskohtien lisäämistä varten. Projektiin tarvittiin myös animaatioita, joiden tekemiseen käy-tettiin Mixamo-animaatiokirjastoa ja animaatioiden yhdistämiseen Blender-mal-linnusohjelmaa. Efektin tekemisessä käytettiin Spark AR Studio -ohjelmaa, jolla lopputulos on mahdollista julkaista käytettäväksi Facebook- ja Instagram-sovel-luksille.

Projektin lopputulos oli onnistunut, vaikkakin visuaalisesta laadusta jouduttiin hie-man tinkimään teknisten rajoitusten takia. Alkuperäisenä tarkoituksena oli saada efekti julkaistua Facebook- tai Instagram-alustalle, mutta aikataulun takia tätä prosessia ei ehditty kunnolla tutkia opinnäytetyötä varten. Projektin aikana kui-tenkin löydettiin muutamia käytännöllisiä tapoja grafiikan tuottamiseen kyseisille alustoille, josta on todennäköisesti hyötyä tulevien projektien kannalta.

Työtä tehdessä havaittiin, että toimeksiannon mukaisen efektin toteuttaminen realistisilla grafiikoilla on haastavaa, sillä lopullisen efektin tiedostokoon täytyy olla erittäin pieni, jotta se voidaan julkaista yleiseen käyttöön. Tulevaisuuden pro-jekteja silmällä pitäen voitaisiin todeta, että kehitysalustan rajoitukset kannattaa ottaa huomioon suunnittelun alkuvaiheessa, ettei esimerkiksi tyyliuunnan tai toi-minnallisuuksien suhteen jouduta tekemään isoja muutoksia kesken projektin. Tässä projektissa näin ei kuitenkaan jouduttu tekemään, vaan tiedostokoosta joh-tuvat ongelmat pystyttiin korjaamaan kompressoimalla ja pienentämällä tekstuu-reita.

Asiasanat: lisätty todellisuus, tietokonegrafiikka, 3d-mallinnus, teksturointi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Game Development

PAUNI, JOONAS:
3D Graphics Producing for Augmented Reality Experience

Bachelor's thesis 55 pages, appendices 2 pages
December 2019

The purpose of this thesis was to examine how three-dimensional computer graphics should be produced for an augmented reality experience, and to use these practices to produce graphics for a Facebook or Instagram camera effect. The project was carried out for Eligo.Studio Oy, a software company focusing on virtual and augmented reality solutions for teaching and marketing in both industry and entertainment.

The project utilized a case study approach. Several computer programs were used to produce the graphics for the project, including 3D modeling software and texturing software. The project was considered successful, although some compromises had to be made in terms of the visual quality of the effect. Useful workflow practices for producing graphics were also discovered, which will likely prove to be useful in future projects.

Unfortunately, the effect publishing could not be examined properly in this thesis due to the tight schedule of the work. It was, however, noted that implementing such effect is challenging as the final file size of an effect has to be very small to be approved for publishing. With future projects in mind, the limitations of the development platform should be taken into account in the initial stages of planning to avoid having to make drastic changes to visual style or functionality.

Keywords: augmented reality, computer graphics, 3d modeling, texturing

SISÄLLYS

1	JOHDANTO	6
2	LISÄTTY TODELLISUUS	7
	2.1 Mitä lisätty todellisuus on	7
	2.2 Miten lisätty todellisuus toimii	8
	2.3 Lisätyn todellisuuden käyttötavat	10
3	3D-GRAFIikka	12
	3.1 3D-mallin anatomia	12
	3.2 3D-mallintaminen ja mallinnustavat.....	14
	3.3 Renderöinti ja varjostimet.....	17
	3.3.1 Varjostusmallit	18
	3.3.2 Fysiikkaperusteinen renderöinti	19
	3.4 Tekstuurit ja materiaalit	20
	3.4.1 Tektuurikarttatyytit	21
	3.5 UV-kartoitus	24
4	PROJEKTIN SUUNNITTELU.....	26
	4.1 Kuvaus	26
	4.2 Graafinen tyyli ja rajoitukset	26
	4.3 Työnkulku.....	27
5	PROJEKTIN MATERIAALIN TUOTTAMINEN	28
	5.1 Jalkapallomalli.....	28
	5.2 Jalkapallon tekstuuri.....	33
	5.3 Jalkapallomaalin malli	37
	5.4 Jalkapallomaalin tekstuuri	41
	5.5 Ympäristö.....	42
	5.6 Maalivahti ja animaatiot.....	44
	5.7 Spark AR Studio.....	46
6	POHDINTA	50
	LÄHTEET.....	52
	LIITTEET.....	54
	Liite 1. Projektissa käytetty HDR-kuva	54
	Liite 2. Projektia varten ostettu 3D-hahmo	55

LYHENTEET JA TERMIT

AR	augmented reality eli lisätty todellisuus
assetti	peleissä käytettävä sisältö, kuten 3D-malli, kuvatie-dosto tai ääniefekti (engl. asset)
baking	yksityiskohtien siirtäminen kolmiulotteisesta mallista toi- seen
high poly	korkearesoluutioinen 3D-malli, jota käytetään yksityis- kohtien luomisessa matalaresoluutioiseen malliin
low poly	matalaresoluutioinen 3D-malli, jota käytetään reaaliai- kaisessa renderöinnissä
polygoni	monikulmio (engl. polygon)
renderöinti	prosessi, jossa kolmiulotteisesta skenestä tuotetaan kuva (engl. rendering)
skene	digitaalisesti tuotettu kolmiulotteinen esitys tilasta, joka sisältää valaistusinformaation (engl. scene)
varjostin	tietokoneohjelma, joka kertoo miten kuvapisteen piirre- tään ruudulle (engl. shader)

1 JOHDANTO

Facebook on lähivuosina mahdollistanut mobiililaitteiden käyttämisen lisätyn todellisuuden alustana näiden kameroiden avulla. Parin viime vuoden aikana muidenkin kuin Facebookin kehittäjien on ollut mahdollista tuottaa kameraefektejä, ja vuoden 2019 kesällä Spark AR, entiseltä nimeltään Camera Effects Platform, julkaistiin kaikille avoimeksi työkaluksi. Opinnäytetyön toimeksiantajayritys Eligo.Studio Oy on muiden muassa ollut kiinnostunut kokeilemaan kyseistä teknologiaa ja sen mahdollisuuksia.

Opinnäytetyö käsittelee lisätyn todellisuuden kameraefektin kolmiulotteisten mallien toteuttamista sekä käyttämistä eri ohjelmistoissa. Tarkoituksena on havainnollistaa, miten Facebook- tai Instagram-palveluiden kameraefekteihin on mahdollista tuottaa kolmiulotteista sisältöä sekä mitä niiden tekemisessä on otettava huomioon.

Opinnäytetyön teoriaosuus on jaettu kahteen osaan, joista ensimmäinen käsittelee lisätyn todellisuuden konseptia ja toimintatapaa. Toisessa osassa käsitellään hieman laajemmin 3D-grafiikkaa ja sen tuottamista. Teoriaosuuksessa käydään läpi aiheen kannalta olennaisimmat tiedot, joiden avulla lukijan on mahdollista seurata ja ymmärtää käytännön osuudessa esitellyt työvaiheet ja konseptit. Käytännön osuus koostuu niin ikään kahdesta osasta, joissa ensin tutustutaan projektiin ja sen suunnitteluun ja tämän jälkeen käydään läpi projektin työvaiheet eri ohjelmissa menemättä kuitenkaan liikaa yksityiskohtiin tai ohjelmien toimintaan, jotta prosessin seuraaminen olisi mahdollisimman vaivatonta.

Ymmärrettävyyden vuoksi opinnäytetyössä on pyritty löytämään sekä käyttämään suomenkielisiä ilmauksia alan yleisistä termeistä, vaikkakin suuri osa 3D-grafiikan parissa toimivista käyttäjä, kansallisuudesta huolimatta, pääsääntöisesti englanninkielistä sanastoa tekstissä ja puheessaan. Olennaiset englanninkieliset vastineet ovat ilmoitettu asiayhteydessä, sillä tarvittaessa lisätietoa aiheesta on helpommin saatavilla englanniksi.

2 LISÄTTY TODELLISUUS

2.1 Mitä lisätty todellisuus on

Lisätty todellisuus (augmented reality tai AR) on virtuaalitodellisuuden (virtual reality tai VR) variaatio. Siinä missä virtuaalitodellisuus upottaa käyttäjän virtuaaliseen tilaan, missä tämä ei pysty näkemään oikeaa maailmaa ympärillään, lisätty todellisuus käyttää digitaalista sisältöä, kuten kuvia, videota, ääntä tai tuntoaistiin perustuvia tuntemuksia ja asettaa ne oikeaan ympäristöön. Lisätty todellisuus antaa käyttäjän nähdä oikean maailman, jonka päälle virtuaaliset esineet sitten heijastetaan tai yhdistetään. Näin ollen lisätty todellisuus nimensä mukaisesti lisää tai täydentää todellisuutta sen sijaan, että se korvattaisiin kokonaan. (Kipper & Rampolla 2012.)

Aikaisemman lisätyn todellisuuden perusmääritelmän lisäksi Kipper ja Rampolla (2012) toteavat, että jotta jotakin voitaisiin pitää varsinaisesti lisättyinä todellisuutena, on sen yhdistettävä oikeaa ja virtuaalista tietoa, sen kanssa tapahtuvan vuorovaikutuksen on oltava reaaliaikaista ja näiden lisäksi sen on toimittava ja sitä pitää voida käyttää kolmiulotteisessa ympäristössä. Lisättyä todellisuutta ei voida pitää yhtenä itsenäisenä teknologiana, vaan se on pikemminkin yhdistelmä useampia eri tekniikoita, jotka yhdessä mahdollistavat digitaalisen informaation aistimisen (Kipper & Rampolla 2012).

Craig (2013) esittää, että lisätty todellisuus on ennemminkin viestintäkanava kuin teknologia, joka pelkän lukemisen, katselemisen tai kuuntelemisen sijaan koetaan ja sen kanssa ollaan vuorovaikutuksessa. Lisätyn todellisuuden kokemuksessa käyttäjä toimii samassa fyysisessä maailmassa kuin normaalistikin, mutta lisätty todellisuus lisää digitaalista informaatiota maailmaan, jonka kanssa käyttäjä voi olla vuorovaikutuksessa samalla tavalla kuin fyysisessäkin maailmassa. Käyttäjä on siis vuorovaikutuksessa oikean maailman kanssa, johon on lisätty digitaalista informaatiota, jota normaalisti ei olisi mahdollista nähdä, kuulla ja jopa tuntea tai koskettaa. (Craig 2013.)

Craigin (2013) mukaan lisätyllä todellisuudella on neljä keskeistä piirrettä. Ensinnäkin, lisätyssä todellisuudessa fyysistä maailmaa niin sanotusti laajennetaan heijastamalla digitaalista informaatiota fyysisen maailman näkymän päälle. Toiseksi, informaatio esitetään sovitettuna oikeaan maailmaan. Kolmanneksi, esitetty informaatio on riippuvainen oikean maailman sijainneista ja käyttäjän fyysisestä näköalasta fyysisessä maailmassa. Neljänneksi, lisätyn todellisuuden kokemus on interaktiivinen, eli käyttäjä voi aistia ja tehdä muutoksia tai jopa luoda uutta informaatiota halutessaan. (Craig 2013.)

Lisättyä todellisuutta voidaan käyttää kaikkien viiden aistin lisäämiseen, mutta yleisin nykypäivän käyttötarkoitus rajoittuu lähinnä visuaaliseen aspektiin. Lisättyä todellisuutta voidaan myös käyttää hävittämään sisältöä oikeasta maailmasta pelkän lisäämisen sijaan. (Kipper & Rampolla 2012.)

Becker (2010) tiivistää lisätyn todellisuuden seuraavasti: lisätty todellisuus on teknologia, tutkimusala, näkemys tulevaisuuden tietojenkäsittelystä, kasvava kaupallinen toimiala sekä uusi väline luovuuden ilmaisulle.

2.2 Miten lisätty todellisuus toimii

Lisätty todellisuus tarvitsee toimiakseen useampaa erilaista teknologiaa. Laitteiston osalta kokemusta täytyy ensinnäkin pystyä käyttämään jossakin laitteessa, jolla on laskennallista tehoa, kuten tietokoneella tai matkapuhelimella ja jotta kokemusta olisi mahdollista tarkkailla, tarvitsee laitteessa olla kiinni näyttö. Kuvan tuottamiseen käytetään tietokoneeseen yhteydessä olevaa kameraa. Käyttäjän liikkeiden seuraamiseksi laitteessa täytyy olla seuranta- ja tunnistusjärjestelmä, kuten GPS, kompassi tai kiihtyvyyssanturi tai näiden yhdistelmä. Lisäksi tarvitaan jokin verkkoinfrastrukturi, johon laite voi olla yhteydessä. Jotta tietokone osaa esittää digitaalisen sisällön oikeassa kohdassa, pitää sillä olla markkeri, eli jokin kappale, kuvio tai paikka kiintopisteenä. (Kipper & Rampolla 2012.)

Ohjelmiston puolesta tarvitaan luonnollisestikin jokin lisätyn todellisuuden kokemuksen mahdollistava sovellus, jota käytetään paikallisesti laitteella. Jos sovellus

hakee digitaalisen sisällön internetistä, täytyy laite olla yhdistettynä verkkoon, esimerkiksi langattomalla yhteydellä. Tätä varten on tietenkin oltava olemassa myös jokin sisältöpalvelin, jolta sisältö haetaan laitteelle. Nämä ovat keskeisimmät komponentit, joita tarvitaan sekä liikuteltavissa että paikallaan olevissa AR-laitteissa. (Kipper & Rampolla 2012.)

Todellisuuden lisääminen toimii kutakuinkin samalla periaatteella, tapahtuu se sitten sijainnin tai niin sanotun markkerin perusteella. Markkeriin perustuva lisätty todellisuus upottaa virtuaalisen sisällön fyysiseen esineeseen, kun taas sijaintiin perustuva seuraa samaa kaavaa, mutta markkerin sijaan määrittää sisällön sijainnin koordinaattien perusteella. (Kipper & Rampolla 2012.)

Tunnistus tai tunnistaminen on prosessi, millä laitteisto ja ohjelmisto määrittelevät, mihin ja kuinka todellisuuden lisääminen tapahtuu. Eri laitteet käyttävät eri menetelmiä ja tekniikoita tämän saavuttamiseksi. Yleisesti käytössä olevia tunnistustapoja ovat kuviointiin, ääriiviivaan ja sijaintiin perustuvat menetelmät. Kosketukseen tai kappaleisiin reagoiviin pintoihin, kuten näyttöihin, lattioihin tai seiniin, perustuvaa tekniikkaa on myös käytetty tunnistuksen apuvälineenä. (Kipper & Rampolla 2012.)

Kuviointiin perustuva tekniikka käyttää tunnistukseen jotakin ennalta määritettyä muotoa tai markkeria, ja tunnistaessaan tämän, sen paikalle lisätään haluttua digitaalista sisältöä, esimerkiksi 3D-malli, ääni- tai videoleike tai jotakin muuta informaatiota. (Kipper & Rampolla 2012.)

Ääriivamenetelmässä esimerkiksi käyttäjän keho tai osa kehosta pyritään tunnistamaan konenäöllä ja yhdistämään saumattomasti johonkin digitaaliseen elementtiin. Tällä tavalla käyttäjä voi olla vuorovaikutuksessa 3D-mallin kanssa. Kamera seuraa esimerkiksi käyttäjän käden ääriiviivaa ja asettelee mallin sen mukaisesti. Tämä tapa on verrattavissa kasvojen tunnistamiseen, jossa kasvot määritellään eri kasvonpiirteiden perusteella, kuten silmien, nenän ja suun avulla ja näiden sijaintia käytetään kiintopisteinä digitaalisen kappaleen asettelussa kasvoille. Kun kasvot on tunnistettu, kappale voidaan laittaa mukautumaan liikkeeseen reaaliajassa. (Kipper & Rampolla 2012.)

Sijaintimenetelmä perustuu GPS:n tai kolmiomittauksen avulla saatuun sijaintitietoon. Tämän tiedon ja käyttäjän kameran näkymän perusteella virtuaalis sisältöä voidaan piirtää ennalta määritettyjen kohteiden päälle. Sijaintiin perustuvaa lisättyä todellisuutta käytetään usein mobiililaitteilla, sillä nykyaikaisissa puhelimissa on kaikki tähän tarvittavat komponentit: kamera, näyttö, GPS-paikannus, kiihtyvyysanturi ja digitaalinen kompassi. (Kipper & Rampolla 2012.)

2.3 Lisätyn todellisuuden käyttötavat

Lisätyn todellisuuden laitteet voidaan jaotella liikuteltaviin ja paikallaan oleviin. Liikuteltava laitteisto mahdollistaa käyttäjän liikkumisen vapaasti käyttäessään laitetta, kun taas paikallaan olevaa laitteistoa on mahdollista käyttää vain siellä, mihin se on asetettu paikoilleen. (Kipper & Rampolla 2012.)

Pääsääntöisesti käytössä olevia lisätyn todellisuuden näyttöjä on kolmea eri tyyppiä: mobiililaitteiden näytöt, puettavat näytöt ja spatiaaliset eli avaruudelliset näytöt. Mobiililaitteessa tämän näyttö toimii reaaliaikaisena katseluohjelmana lisätyn todellisuuden sovellukselle, jota laite käyttää. (Kipper & Rampolla 2012.)

Puettavat näytöt, toiselta nimeltään HMD (engl. head-mounted display) ovat näyttöjä, jotka käyttäjä pukee päähänsä samaan tapaan kuin silmälasit. Tyypillisesti puettavat näytöt käyttävät yhtä tai kahta pienikokoista näyttöä, linssettä ja läpi-kuultavia peilejä, jotka ovat upotettu kypärään tai silmälasihin. Näiden avulla käyttäjä pystyy kokemaan lisätyn todellisuuden luonnollisemmin ja isommalla näkökentällä verrattuna perinteiseen näyttöön. (Kipper & Rampolla 2012.)

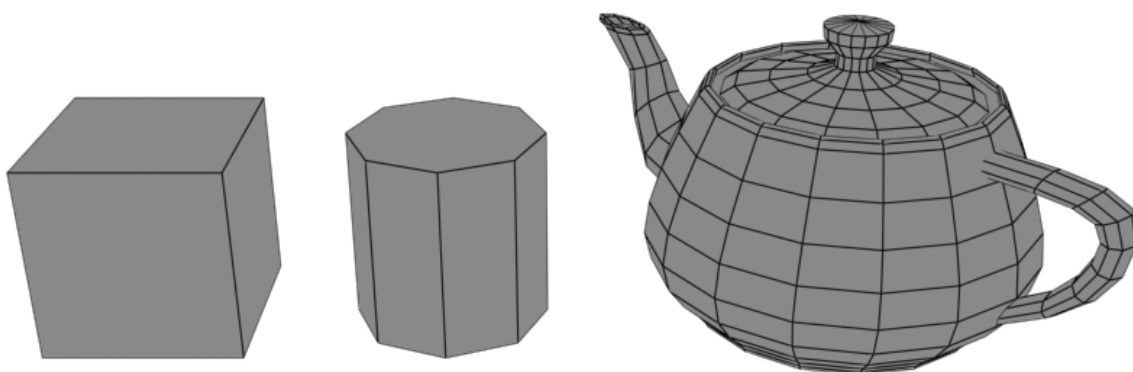
Spatiaalisia näyttöjä on kahta eri tyyppiä. Spatiaaliset videonäytöt toimivat näyttämällä webkameralla AR-markkeria, jonka avulla virtuaalinen sisältö heijastetaan näytölle kameran videosityötteen päälle lisättynä. Spatiaaliset lisätyn todellisuuden näytöt (engl. spatial augmented reality, SAR) taas käyttävät eri teknologioita, kuten projektoreita tai hologrammeja heijastaakseen tietoa fyysisten kappaleiden pinnalle. Näyttö on tyypillisesti osa ympäristöä, toisin kuin muut lisätyn todellisuuden sovellutukset, jotka ovat enemmänkin persoonallisia laitteita. Spa-

tiaalinen näyttö on käytännöllinen, jos sisältöä on tarkoitus näyttää isommille ihmisryhmille ja siksi niitä käytetäänkin yleensä julkisissa tiloissa. (Kipper & Rappolla 2012.)

3 3D-GRAFIikka

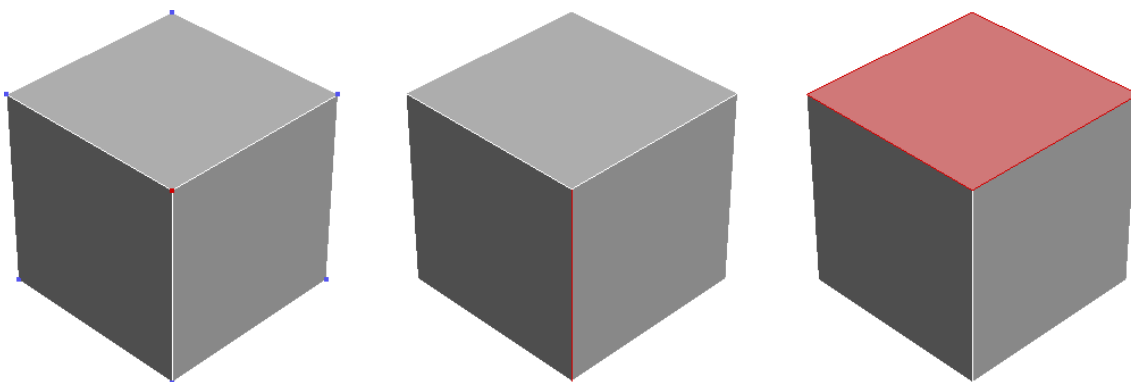
3.1 3D-mallin anatomia

Tietokonegrafiikassa suosituin tapa esittää kolmiulotteinen kappale on polygonimalli (engl. polygon mesh). Tämä tapahtuu esittämällä kappaleen pinta toisiinsa yhdistettyinä polygoneina. Tämän kaltainen esittämistapa on joko eksakti tai suurpiirteinen, kappaleen muodon mukaan. Kuutio voidaan esittää tasan kuuden nelikulmion avulla, kun taas ympyrälieriö on mahdollista esittää polygoneilla vain likiarvoisesti; esimerkiksi kahdeksan suorakaidetta vaipassa ja kahdeksankulmio kummassakin pohjassa, kuten kuvassa 1 on havainnollistettu. Polygonien määrä approksimaatiossa määrittää, kuinka tarkasti kappale voidaan esittää ja tämä vaikuttaa muun muassa mallin laatuun. (Watt 2000, 1.)



KUVA 1. Esimerkki kuution, lieriön sekä teepannun polygonimalleista

Aluetta polygonin sisällä kutsutaan tahkoksi (engl. face). Polygonin kulmissa olevia pisteitä nimitetään kärkipisteiksi (engl. vertex, mon. vertices) ja tahkoa ympäröiviä, kärkipisteet toisiinsa yhdistäviä janoja reunoiksi tai sivuiksi (engl. edge). Polygonit voivat koostua mistä tahansa määrästä kärkipisteitä ja sivuja, kuitenkin vähintään kolmesta, jotka tarvitaan yksinkertaisimman mahdollisen tahkon, kolmion (engl. triangle, lyh. tri), muodostamiseen. Nelisivuista polygonia puolestaan kutsutaan nelikulmioksi (engl. quadrangle, lyh. quad) ja tätä useampisivuiset polygonit ovat n -kulmioita (engl. n -gon) eli polygoneja, joilla on n määrä sivuja. (Vaughan 2011; Chopine 2011; Lehtovirta & Nuutinen 2000, 21.) Kuvassa 2 on kuvattuna polygonimallin eri osat ja kuvassa 3 erityyppiset polygonit.



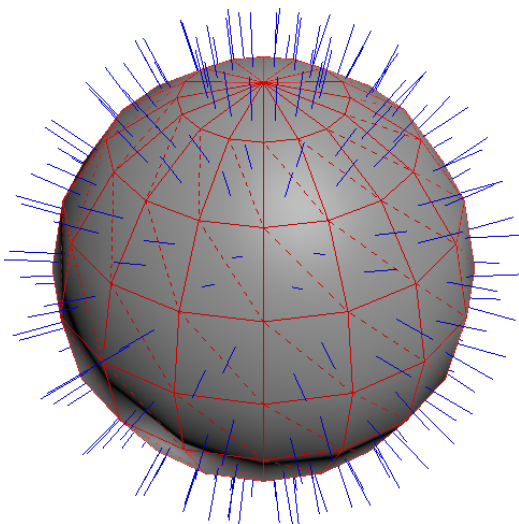
KUVA 2. Polygonin osat visualisoituna kuutiossa, vasemmalla valittuna kärkipiste, keskellä sivu ja oikealla tahko



KUVA 3. Kolmio, nelikulmio ja n-kulmio

Jokaisella kärkipisteellä on oma X-, Y- ja Z-koordinaattinsa. Koska kärkipisteillä ei ole pituutta, leveyttä tai korkeutta, ei niitä näin ollen voida itsenään piirtää ruudulle. (Vaughan 2011.)

Normaali (engl. normal) on tahkoa vastaava kohtisuora eli 90-asteen kulma (Zeman 2014, 25). Se voidaan ajatella suuntana, johon polygoni osoittaa. Joka polygonilla on näin ollen niin sanottu etupuoli (engl. front face) ja takapuoli (engl. back face). (Chopine 2011.) Kappaleessa olevien tahkojen normaalien tulisi aina osoittaa samaan suuntaan, toisin sanoen ulospäin kappaleen pinnasta, ellei tarkoituksena ole saavuttaa esimerkiksi jokin poikkeuksellinen visuaalinen efekti. Joissain tapauksissa mallinnusprosessissa yhden tai useamman tahkon normaali saattaa tahattomasti kääntyä osoittamaan eri suuntaan kuin ympäröivät tahkot, jolloin tämä on syytä korjata kääntämällä normaalin suunta (Chopine 2011).



KUVA 4. Tahkojen normaalit visualisoituna sinisillä viivoilla

3.2 3D-mallintaminen ja mallinnustavat

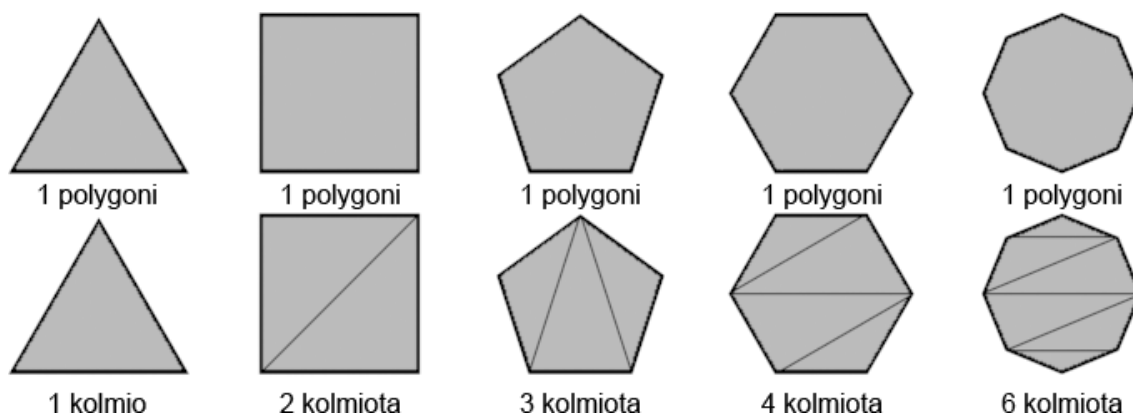
Perinteisesti polygonimalleja muokataan liikuttamalla yhtä tai useampaa kappa-
leen kärkipistettä kerralla 3D-sovelluksessa. Kärkipisteiden asetelua on myös
mahdollista manipuloida erilaisten sovelluksista löytyvien muokkaustoimintojen
avulla, jolloin monimutkaistenkaan muotojen luomiseen ei tarvitse käyttää paljon
aikaa. (Lehtovirta & Nuutinen 2000, 21.)

Mallinnettaessa kolmiulotteisia kappaleita on yleensä suositeltavaa käyttää pel-
kästään nelikulmioita, jotta mallissa pysyy niin sanottu hyvä topologia. Käytän-
nössä tämä tarkoittaa kappaleen rakennetta ja polygonien asetelua, mikä vaikut-
taa kappaleen muotoon. Tarvittaessa myös kolmioita voi käyttää, mutta isommat
kuin neljäisivuiset polygonit on suotavaa jakaa nelikulmioiksi ja kolmioiksi, sillä se
helpottaa mallin muokkaamista. Lisäksi topologia vaikuttaa valaistuksen käyttäy-
tymiseen mallin pinnassa; huono rakenne voi aiheuttaa ei-toivottuja tuloksia mal-
lia tarkasteltaessa. (Zeman 2014, 28–30; Turbosquid n.d.)

Polygon count tai poly count on usein käytetty englanninkielinen nimitys, jolla vii-
tataan polygonien määrään polygonimallissa (Vaughan 2011). Polygonien mää-
rän tarkkailu on olennaista, sillä mitä enemmän polygoneja mallissa on, sitä
enemmän prosessointitehoa sen käsittely ja renderöinti eli kuvanlaskenta vaatii.
Kuvanlaskennassa tämä taas pidentää aikaa, joka tietokoneella kestää tuottaa

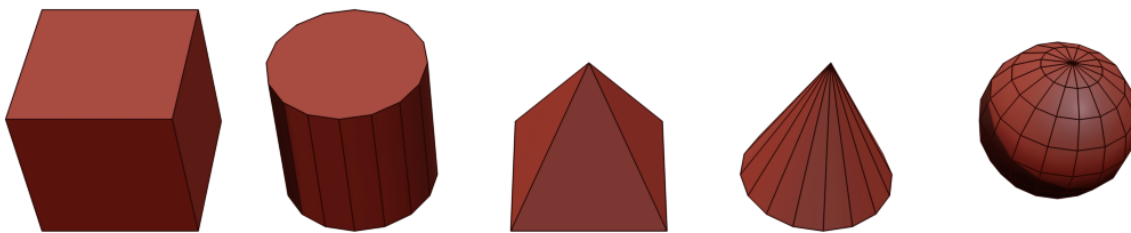
kuva mallista. (Chopine 2011.) Polygoneja laskettaessa polygonilla tarkoitetaan yleensä kolmiota eli kolmikulmaista polygonia, sillä renderöitäessä jokainen polygoni muunnetaan kolmioksi eli trianguloidaan, koska renderöntiohjelmat ja pelimoottorit käyttävät ainoastaan kolmioita grafiikan piirtämiseen näiden yksinkertaisuuden vuoksi. Useimmat renderöntiohjelmat osaavat muuntaa polygonit kolmioiksi renderöitäessä, mutta esimerkiksi pelimoottoreissa käytettävät mallit täytyy trianguloida valmiiksi ennen niiden tuomista moottoriin. (Vaughan 2011.)

Mallinnusohjelmissa on yleensä mahdollisuus jollakin tapaa esittää käyttöliittymässä polygonien yhteenlaskettu lukumäärä (Chopine 2011). On kuitenkin tärkeää varmistaa, että ilmoitettu lukema tarkoittaa nimenomaan kolmioiden lukumäärää, jotta laskemisessa käytettäisiin yhtenäistä laskutapaa (Polycount Wiki 2015). Kolmioiden yhteenlaskettu määrä mallissa on usein suurempi kuin polygonien määrä, kuten kuvasta 5 voidaan todeta.



KUVA 5. Eri monikulmioita ja niiden geometria kolmioiksi rikottuna

Käytännössä jokaisessa nykyaikaisessa mallinnusohjelmassa on oletuksena joukko erilaisia geometrisia kappaleita, niin sanottuja primitiivejä (engl. primitive), joiden ominaisuuksia, kuten resoluutiota, voidaan säätää kappaleen parametrien avulla. Tällaisia kappaleita ovat muun muassa kuutio, lieriö, pallo, kartio, pyramidi ja torus (kuva 6). Vaikka jokainen näistä kappaleista onkin mahdollista toteuttaa perinteisillä mallintamistavoilla, löytyvät ne valmiiksi ohjelmista työskentelyn nopeuttamiseksi. Primitiivejä voi käyttää myös sellaisenaan, mutta ennen kaikkea ne soveltuvat hyvin eri muokkaustyökaluilla editoitaviksi. (Giambruno 2002.)



KUVA 6. Primitiivejä 3ds Max -ohjelmassa, vasemmalta lukien kuutio, lieriö, pyramidi, kartio ja pallo

Splini (engl. spline) on vähintään kahden pisteen määrittämä kolmiulotteisessa tilassa oleva käyrä. Mallinnuksessa yleisimmin käytetyt splinit ovat Bézier-käyriä, joiden avulla voidaan mallintaa pehmeitä kurveja käyttämällä huomattavasti vähemmän pisteitä kuin mitä vastaava polygonimalli tarvitsisi. Käyriä hallitaan niiden kontrollipisteiden avulla ja niillä voidaan vaikuttaa käyrän muotoon vaivattomasti. Splinit eivät ole sidonnaisia resoluutioon toisin kuin polygonimallit, ja niitä voitaisiinkin verrata esimerkiksi Illustrator- tai Photoshop-ohjelmissa käytettävään vektorigrafiikkaan. (Vaughan 2011.)

Subdivision surface on algoritmi, joka luo pehmeän, kaarevan pinnan karkeamman polygoniverkon pohjalta, jota kutsutaan englanniksi nimellä base mesh eli pohjaverkko. Tämän pohjaverkon kärkipisteet toimivat kontrollipisteinä luodulle pinnalle, niin sanottuna häkkinä. Tällä tavoin luodun pinnan sisältämää polygonien määrää voidaan säätää asteittain jakamalla verkkoa hienommaksi, jolloin verkosta tulee tiheämpi. Näin saadaan tehtyä sileitä ja monimutkaisiakin kappaleita nopeasti ja ennalta-arvattavasti pelkän yksinkertaisen pohjaverkon avulla. (Vaughan 2011.)

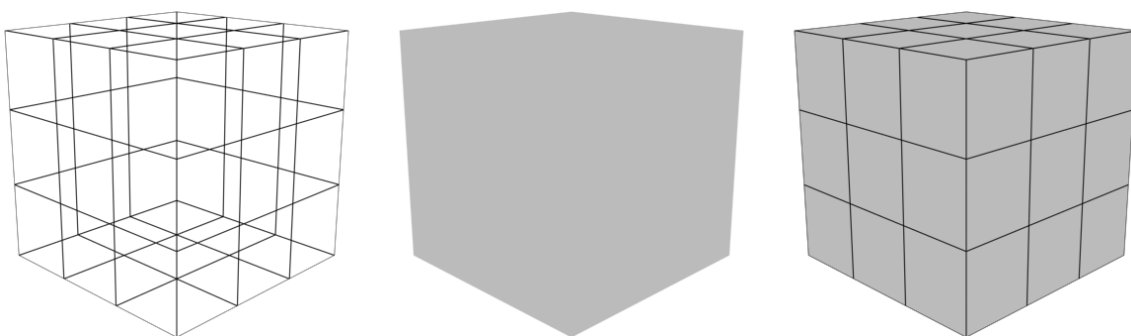
Texture baking (tai pelkästään baking) tarkoittaa yksityiskohtien siirtämistä toisesta mallista toiseen. Prosessissa käytetään kahta sisäkkäin olevaa mallia, joista toinen on korkearesoluutioinen malli ja toinen yksinkertaistettu versio samasta mallista. Työkalu, jolla siirtäminen tehdään, suuntaa säteitä matalaresoluutioisen mallin ulkopinnasta takaisin kohti korkearesoluutioisen mallin pintaa ja kun säteet kohtaavat toisen mallin kanssa, pinnan yksityiskohdat kirjoitetaan muistiin ja tallennetaan tekstuurikarttaan käyttäen ensimmäisen mallin tekstuurikoordinaatteja. Tällä tavoin voidaan tuottaa eri tekstuurityyppejä esimerkiksi peleissä käytettäviä malleja varten, sillä suoritustehon kannalta on tehokkaampaa

käyttää tekstuureita yksityiskohtien esittämiseen kuin varsinaista geometriaa. (Polycount Wiki 2015.)

3.3 Renderöinti ja varjostimet

Renderöinti (engl. rendering) on prosessi, jossa digitaalisesta, kolmiulotteisesti esitetystä tilasta eli skenestä (engl. scene) tuotetaan kuva (Pharr, Humphreys & Jakob 2016). Yksinkertaisimmassa renderöintitavassa, pyyhkäisyviivarenderöinnissä (engl. scanline rendering), jokaisesta näkymän kuvapistestä johdetaan vuorollaan viiva kohti kolmiulotteista tilaa. Jos viiva osuu johonkin pintaan, kyseiselle kuvapisteelle annetaan tämän pinnan väri. Jos viiva ei osu mihinkään, kuvapiste saa taustalle määritetyn, esimerkiksi mustan, värin. Tämä tuottaa käytännössä pelkän siluetin näkymästä. (Chopine 2011.)

Z-puskurointi (engl. Z-buffer tai depth buffer) on tekniikka, jonka avulla lasketaan mitkä pinnat voidaan jättää piirtämättä määrittelemällä onko pinta toisen pinnan takana. Pinnan reuna voidaan piirtää, jos sen edessä ei ole mitään, muutoin se jätetään piirtämättä. (Chopine 2011.) Kuvassa 7 on havainnollistettu kuution mallista eri renderöintitekniikoilla tuotetut kuvat.



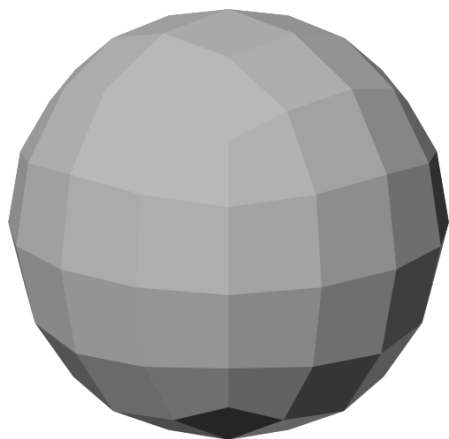
KUVA 7. Kuutio renderöitynä niin sanottuna rautalankamallina, pelkällä pyyhkäisyviivalla sekä pyyhkäisyviivalla yhdistettynä Z-puskuriin

Jotta pyyhkäisyviivalla renderöity kuva saataisiin vaikuttamaan kolmiulotteiselta, siinä näkyvät kappaleet täytyy sävyttää realistisella tavalla. Käytännössä tämä tarkoittaa, että kappaleen sivut, jotka osoittavat pois päin valonlähteestä, ovat sävyiltään tummemman värisiä kuin kappaleen alkuperäinen väri, ja valonlähteen osoittavat sivut puolestaan vaaleamman sävyisiä. Algoritmeja, joilla alun

perin lisättiin pelkästään varjostus kappaleille, kutsutaan varjostimiksi (engl. shader). (Chopine 2011.) Varjostin on käytännössä ohjelma eli pätkä koodia, joka kertoo, mihin kohtaan ja minkä näköisenä kappaleen pinta tulee piirtää ruudulle. Tarkemmin sanottuna varjostimelle voidaan määrittää eri tapoja esittää yksityiskohtaisia pintoja esimerkiksi käyttämällä bittikarttakuvia eli tekstuureita, ja tämän perusteella määritetään, minkä värisenä kukin kappaleen kuvapiste renderöidään. (Halladay 2019.) Nykyään varjostimia siis käytetään varjostusten lisäksi erilaisten graafisten efektien tuottamiseen.

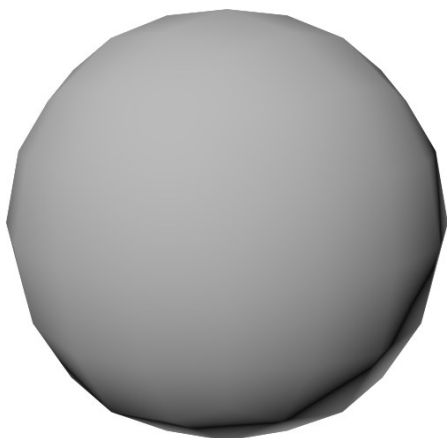
3.3.1 Varjostusmallit

Tasavarjostus (engl. flat shading) on yksinkertaisin menetelmä varjostuksen tuottamiseen. Jokaiselle tahkolle määritetään väri sen mukaan, mihin kulmaan se osoittaa sekä kuinka kaukana valonlähteestä tahko on. Jos valo osoittaa kohti tahkoa, väriä vaalennetaan ja vastaavasti tummennetaan, jos tahko osoittaa pois valosta. (Chopine 2011.) Alla (kuva 8) on esitetty tasavarjostettu kappale.



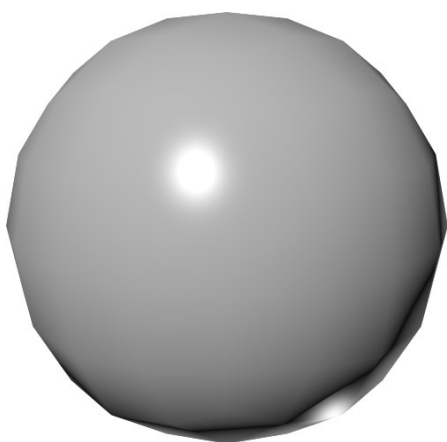
KUVA 8. Tasavarjostus

Gouraud-varjostuksessa (kuva 9) kappaleen jokaiselle kärkipisteelle lasketaan keskiarvoinen normaali niiden tahkojen avulla, joissa kärkipiste on kiinni. Tämän avulla joka kärkipisteelle voidaan antaa väri sen mukaan, missä kulmassa valo osuu kärkipisteeseen. Näitä värejä interpoloimalla saadaan liukuväri kärkipisteiden ja sivujen välille, joka antaa kappaleelle vaikutelman tasaisesta muodosta. Gouraud-varjostus toimii kuitenkin huonosti kiiltävien ja heijastavien pintojen esittämiseen. (Chopine 2011.)



KUVA 9. Gouraud-varjostus

Phong-varjostus (kuva 10) toimii samalla tavalla kuin Gouraud-varjostus, mutta kärkipisteiden värien interpoloimisen sijaan se käyttää interpoloimiseen kärkipisteiden normaaleja. Joka kuvapisteelle lasketaan väri interpoloitujen normaalien avulla, mikä on huomattavasti hitaampaa, sillä joka kuvapiste ottaa huomioon valon tulokulman. (Chopine 2011.)



KUVA 10. Phong-varjostus

3.3.2 Fysiikkaperusteinen renderöinti

Fysiikkaperusteinen renderöinti (engl. physically based rendering, lyhennettynä PBR) tarkoittaa renderöintitekniikoita, jotka yrittävät simuloida todellisuutta käyttämällä fysiikan perusteita mallintaakseen valon ja aineen vuorovaikutusta. Vaikka fysiikkaperusteinen lähestymistapa vaikuttaakin ilmeiseltä tavalta lähestyä renderöintiä on se ollut laajasti käytössä vasta reilun kymmenen vuoden ajan.

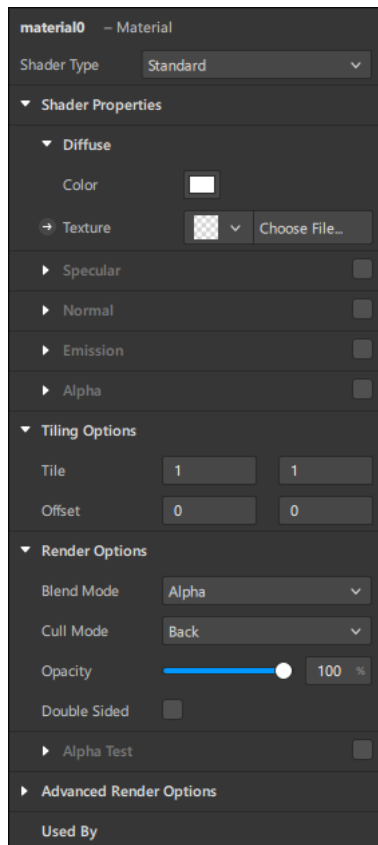
(Pharr, Humphreys & Jakob 2016.) Fysiikkaperusteinen renderöinti on enemmänkin metodiikka kuin standardi, jossa on tiettyjä perusperiaatteita ja ohjenuoria, mutta ei tarkkoja sääntöjä. Toteutustapoja tekniikasta voi siis olla erilaisia ja työnkulku sekä termit voivat vaihdella, mutta perusperiaatteet pysyvät samoina. (McDermott 2018.)

Fysiikkaperusteinen renderöinti ja varjostus mahdollistavat tarkemman hahmonuksen valon vuorovaikutuksesta eri pintojen kanssa. Fysiikkaperusteinen renderöinti helpottaa 3D-artistien työtä monella tapaa, esimerkiksi pintojen ominaisuuksien laatimisessa 3D-malleille, sillä menetelmät ja algoritmit perustuvat fyysisesti tarkkoihin kaavoihin, jolloin realistisen näköiset ja yhdenmukaiset lopputulokset ovat helpommin saavutettavissa, artistista riippumatta. Tämän lisäksi mallit valaistuvat asianmukaisesti eri valaistusolosuhteissa. (McDermott 2018.)

3.4 Tekstuurit ja materiaalit

Tekstuurilla tarkoitetaan kaksiulotteista kuvaa, joka määritetään kappaleelle. Tekstuuri voi olla perinteinen bittikarttakuva, proseduraalinen tekstuuri, joka luodaan matemaattisten algoritmien avulla tai näiden kahden yhdistelmä. Tekstuuri voi lisäksi sisältää tietoa pinnan rakenteesta. Tekstuurien ja varjostimien yhdistelmää kutsutaan yleisesti materiaaleiksi, sillä niillä saadaan aikaan vaikutelma, mistä jokin kappale on tehty. (Chopine 2011.)

Materiaalit määrittävät, miten värit, tekstuurit ja muut materiaalin ominaisuudet valmistellaan renderöintiä varten 3D-ohjelmassa. Jokainen materiaali käyttää yhtä tiettyä varjostinta, mikä vaikuttaa siihen, mitä ominaisuuksia materiaalilla on käytettävissä. Eri ohjelmissa on mahdollista valita, mitä varjostinta materiaali käyttää ja tämä määrittää muokattavissa olevat vaihtoehdot. Ohjelmasta riippuen materiaaleilla voi olla paljonkin muokattavia ominaisuuksia, kuten esimerkiksi offline-renderöintiohjelmissa, kun taas pelimoottoreissa vaihtoehtoja on yleensä rajoitetusti resurssien säästämiseksi renderöitäessä. Materiaalit koostuvat kanavista, jotka renderöidessä yhdistetään ja näistä muodostetaan kuvapisteet lopulliseen kuvaan. (Zeman 2014.) Kuvassa 11 näkyy Spark AR Studion materiaalieditori.



KUVA 11. Standard-varjostinta käyttävä materiaali Spark AR Studio -sovelluksen materiaalieditorissa

3.4.1 Tekstuurikarttatyytit

Tekstuurikarttoja on monta eri tyyppiä ja ne eroavat ominaisuuksiltaan sekä käyttötavoiltaan toisistaan. Fysiikkaperusteinen renderöinti on tuonut perinteisen teksturoinnin käyttämien karttojen rinnalle useamman uuden karttatyytin, jotka mahdollistavat realistisemmän valon käyttäytymismallin verrattuna perinteisiin tekniikoihin. Fysiikkaperusteisessa renderöinnissä on yleisesti käytössä kaksi eri työtapaa, metallinen (engl. metalness tai metallic) ja spekulaari (engl. specular) työtapa, joista kumpikin käyttää osittain samoja peruskarttoja. Toisistaan eroavien karttojen tekotapa on kuitenkin hieman erilainen, vaikkakin ne ominaisuuksiltaan muistuttavat toisiaan. Tässä luvussa esitellään tärkeimmät sekä tämän projektin kannalta oleelliset tekstuurikarttatyytit.

Diffuusikartta

Diffuusikartta (engl. diffuse map) määrittää pinnan värin ja yksityiskohdat. Diffuusikartassa ei tulisi olla lainkaan informaatiota suorasta valaistuksesta, vaan tekstuurin tulisi näyttää joka puolelta tasaisesti valaistulta. Varjoja tai kiiltoja ei myöskään kuulu lisätä diffuusikarttaan, mutta näihin on saatettu perinteisessä teksturoinnissa tehdä poikkeuksia visuaalisen lopputuloksen parantamiseksi. (Gahan 2012.)

Albedo

Albedo on pohjavärikartta (käytetään myös nimityksiä base color map tai color map), joka on perinteisesti tunnettu diffuusikarttana. Se määrittää hajaantuneen valon värin. Albedo eroaa perinteisestä diffuusikartasta siinä suhteessa, että siinä ei esitetä suunnattua valaistusta tai ympäristön okklusiota. Fysiikkaperusteisessa renderöinnissä karttaan lisätty valaistusinformaatio näyttäisi vääraltä tietyissä valaistusolosuhteissa, kun taas ympäristön okklusiokarttaa varten PBR-varjostimissa on sille oma syötteensä. (Wilson 2015.)

Spekulaarikartta

Spekulaarikartta (engl. specular map) voi olla harmaasävy- tai värikuva. Spekulaarikartta määrittää pinnan kiillon kirkkauden ja värin. Spekulaarikartan sävyt kertovat, kuinka kiiltävä pinta mistäkin kohtaa on: vaaleammat sävyt ilmenevät kiiltävimpinä ja vastaavasti tummat sävyt vähemmän kiiltävinä kohtina, mustan ollessa täysin matta pinta. (Gahan 2012.)

Karheus- tai kiiltokartta

Karheus- ja kiiltokartta (engl. roughness/gloss map tai microsurface) määrittää, kuinka karhea tai sileä materiaalin pinta on. Karheampien kappaleiden pinnassa heijastukset näyttäytyvät laajempina ja himmeämpinä, kun taas sileämmillä pinoilla heijastukset ovat kirkkaampia ja terävämpiä. Spekulaarikartan tapaan karheus- ja kiiltokartat ovat harmaasävyisiä: esimerkiksi karheuskartassa musta sävy vastaa sileää pintaa ja valkoinen karheaa. Kartan sävyjen tulkinta voi toimia käänteisesti ohjelmien välillä, jolloin myös käytetty nimi on yleensä eri. (Wilson 2015.)

Metallikartta

Metallikartta (engl. metallic map tai metalness map) määrittelee, mitkä materiaalin osat ovat puhdasta metallia. Metallikartta on mustavalkokuva; valkoinen väri tulkitaan metallina, kun taas musta väri epämetallina. Metallikartta ei ole pelkästään binäärinen, vaan tietyissä tapauksissa myös harmaan sävyjä voidaan käyttää esittämään esimerkiksi likaa metallin pinnassa, mikä laskee metallin heijastuskykyä. (McDermott 2018.)

Ympäristön okklusiokartta

Ympäristön okklusiokartta (engl. ambient occlusion tai AO) määrittää, kuinka paljon ympäröivä valo saavuttaa tietyn pisteen kappaleen pinnassa (McDermott 2018). Perinteisesti okklusiokartta on yhdistetty diffuusikarttaan, mutta fysiikka-perusteisessa renderöinnissä parempi käytäntö on käyttää AO-karttaa erillisenä syötteenä varjostimessa, jotta kyseistä informaatiota pystytään käyttämään älykkäämmin valaistuksessa (Wilson 2015).

Normaalikartta

Normaalikartta (engl. normal map) on RGB-värimallia käyttävä tekstuuri, jolla kappaleelle saadaan vaikutelma siitä, että sen pinnassa olisi enemmän yksityiskohtaa kuin siinä geometriansa puolesta on. Jokainen kolmesta värikanavasta kuvaa pinnan normaalin suuntaa kolmiulotteisessa tilassa käyttäen X-, Y- ja Z-koordinaatteja. Tämän ansiosta normaalikartat toimivat reaaliaikaisen valaistuksen kanssa saaden pinnan valaistumaan kuin siinä olisi geometriaa. Normaalikartta voidaan luoda korkearesoluutioisen kappaleen avulla tai harmaasävykuvan pohjalta. (Gahan 2012.)

Korkeuskartta

Korkeuskartta (engl. height map) vastaa konseptiltaan normaalikarttaa, mutta on suorituskyvyn kannalta raskaampi tekniikka. Niitä käytetään yleensä yhdessä normaalikarttojen kanssa määrittelemään isompia sisään tai ulos työntyneitä alueita kappaleen pinnassa. Siinä missä normaalikartta vaikuttaa pelkkään valaistukseen pinnan tekstuurissa, korkeuskartta varsinaisesti siirtää näkyvän pinnan tekstuuria, mikä saa pinnan näennäisten kohoumien vaikuttamaan siltä, että ne peittäisivät muuta pinnan yksityiskohtaa. Tosiasiassa tämä ei kuitenkaan vaikuta kappaleen siluettiin, eli kappaleen geometria pysyy samana. Korkeuskartta on

harmaasävyinen kuva, jossa tummat sävyt tulkitaan pinnan syvennyksinä ja vaaleat sävyt kohollaan olevina alueina. (Unity Technologies 2019.)

Opasiteetikartta

Opasiteetikartta (engl. alpha map tai opacity map) on joko harmaasävy- tai mustavalkotekstuuri, joka säätelee pinnan läpinäkyvyyttä. Kuvan valkoisten kuvapisteen peittämät alueet näkyvät läpinäkymättöminä, kun taas mustat kuvapisteen tekevät pinnasta läpinäkyvän. (Gahan 2012.)

Ympäristökartta

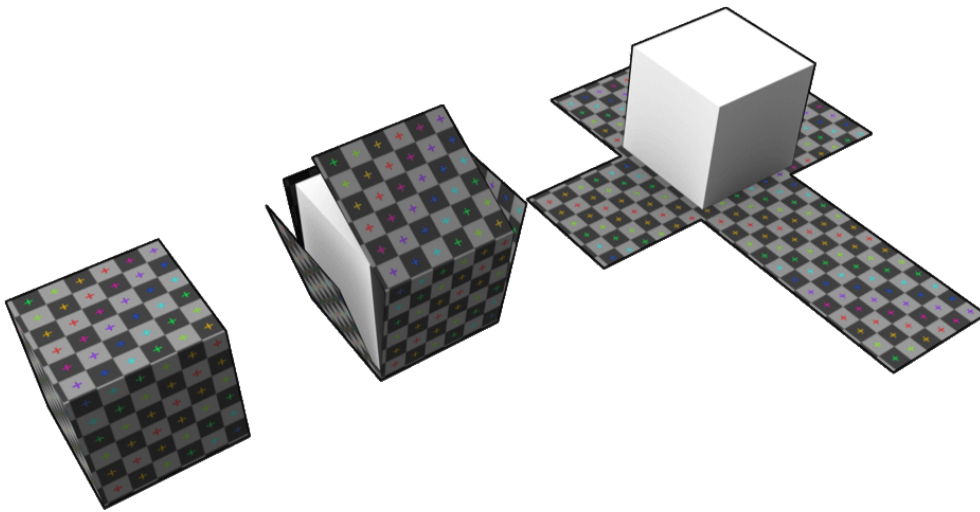
Ympäristökartta (engl. environment map) on kuvatiedosto, jonka avulla voidaan simuloida kuvan mukaisen ympäristön tuottamia heijastuksia kappaleiden pintoihin. Ympäristökarttaa voidaan myös käyttää valaisemaan skeneä sekä jäljittelemään kaukaisempaa näkymää skenen ympärillä, esimerkiksi taivasta. (Polycount Wiki 2015.)

3.5 UV-kartoitus

UV-kartat pitävät sisällään tietoa tekstuurien sijoittelusta mallin pinnassa. UV-koordinaatit ovat kaksiulotteinen esitysmuoto kolmiulotteisesta tilasta. Ne muodostavat yhteyden kaksiulotteisen kuvan ja kolmiulotteisen pinnan välille, jolle kuva levitetään. (Vaughan 2011.) UV-koordinaateista puhuttaessa X-, Y- ja Z-akselien sijasta viitataan U-, V- ja W-koordinaatteihin, joista U-akseli vastaa leveyssuuntaa ja V-akseli pituussuuntaa. W-akseli vastaa Z-akselia ja sitä käytetään lähinnä proseduraalisia karttoja varten, mutta toisinaan myös työskentelyn apuna. (Gahan 2012.) Prosessia, jossa kaksiulotteinen kuva projisoidaan kolmiulotteisen mallin pintaan, kutsutaan UV-kartoitukseksi (engl. UV mapping) (Vectary n.d.).

Jotta kolmiulotteisen mallin UV-koordinaatit saadaan vastaamaan bittikarttaku-
van tekstuurikoordinaatteja, täytyy mallin pinta niin sanotusti levittää tasoon. Tämä pyritään yleensä tekemään pitämällä mahdollisimman moni polygoni kiinni toisissaan, jotta UV-kartta pysyy luettavana. Jokaiselle mallin polygonille löytyy UV-kartasta sitä vastaava polygoni, mikä mahdollistaa tekstuurin heijastamisen

mallin pintaan. Polygonien sivujen avulla määritetään saumat, joiden perusteella polygonit erotetaan tarvittavissa kohdissa toisistaan ja malli levitetään tai avataan tasoon (engl. unwrap). UV-kartan polygonien tulisi vastata mahdollisimman tarkasti niiden vastakappaleita mallissa, jotta tekstuuri mallin pinnassa ei näyttäisi vääristyneeltä, kuten venyneeltä tai puristuneelta. Lisäksi saumoja tulisi olla mahdollisimman vähän ja vain välttämättömissä kohdissa. (Chopine 2011.) Yksinkertaisimpana esimerkkinä tästä on kuutio (kuva 12), joka voidaan kuvitella pahvilaatikkona, joka taitellaan auki kunnes jokainen laatikon sivu on maata vasten. Uloimmaksi jäävät reunat ovat kappaleen saumoja.



KUVA 12. Kuution levittäminen tasoon havainnollistettuna, ruutukuvioitu kappale esittää kuution UV-karttaa (Vectary n.d.)

4 PROJEKTIN SUUNNITTELU

4.1 Kuvaus

Toimeksiantona on toteuttaa lisättyä todellisuutta hyödyntävä jalkapalloteemainen, interaktiivinen kameraefekti. Efektissä käyttäjän kasvot asetetaan jalkapallomaalivahdin vartaloon käyttämällä mobiililaitteen etukameraa. Käyttäjä pystyy kasvonsa eleillä liikuttamaan maalivahtia, jonka tarkoitus on estää mahdollisimman montaa jalkapalloa lentämästä takana olevaan jalkapallomaaliin ennen kuin aika loppuu. Jokaisesta torjutusta pallosta annetaan pisteitä ja ajan loputtua käyttäjä saa näytölle yhteenlasketun pistemäärän sekä vaihtoehdon aloittaa alusta. Efektiä voisi siis luonnehtia sisältönsä puolesta peliksi.

Efekti toteutetaan Windows-käyttöjärjestelmällä käytettävän Spark AR Studion avulla. Toimeksiantajayrityksen aikaisempaa jääkiekkoteemalla toteutettua efektiä käytetään projektin pohjana ajan ja työmäärän säästämiseksi.

4.2 Graafinen tyyli ja rajoitukset

Koska käyttäjän kasvot asetetaan virtuaalisen hahmon vartaloon, täytyy maalivahdin näyttää realistiselta ja uskottavalta, jotta käyttäjän kasvot näyttäisivät kuuluvan hahmolle – tai toisin päin. Fysiikkaperusteinen renderöinti on siis luonteva vaihtoehto tämän saavuttamiseksi, sillä Spark AR Studiossa on tuki PBR-materiaaleille. Tämän vuoksi kaikki projektissa tarvittavien mallien tekstuurit täytyy siis olla toteutettu PBR-tekniikalla, jotta mikään malli ei näyttäisi eroavan visuaalisesti toisistaan ja tyyli olisi yhtenäinen.

Koska efektiä tullaan käyttämään puhelimilla, on sisältöä tehdessä pidettävä huoli, ettei tästä tule liian raskasta mobiililaitteille. Spark AR Studion tekniset suositukset antavat valmiiksi joitain viitteellisiä ohjeita sisällön luomiseen. Esimerkiksi suurin tuettu tekstuurikoko on 1024x1024 kuvapistettä. Lisäksi yhdessä kappaleessa saa olla enintään 20,000 kärkipistettä ja kolmioiden yhteenlaskettu määrä kaikille efektissä kerralla näkyville malleille tulisi olla alle 50,000. Tämän

ei pitäisi olla ongelma, sillä tarkoituksena ei ole näyttää kerralla kuin pelihahmo, muutama jalkapallo sekä maali ja peliympäristö. Tämä on kuitenkin hieman enemmän kuin mitä muut puhelimilla käytettävät AR-efektit yleensä sisältävät, joten sulavan käyttökokemuksen varmistamiseksi kannattaa pelata varman päälle ja yrittää toteuttaa tarvittavat mallit niin yksinkertaisina kuin suinkin mahdollista, kiinnittäen kuitenkin huomiota visuaaliseen uskottavuuteen.

4.3 Työnkulku

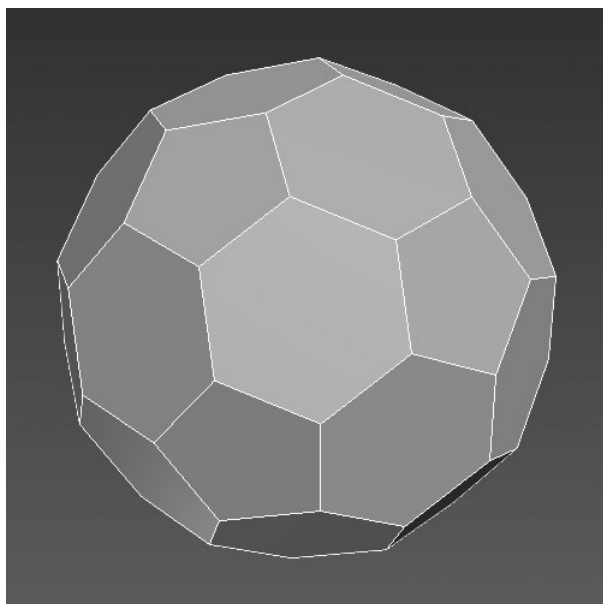
Tässä projektissa 3D-mallien tekemiseen käytetään 3ds Max -mallinnusohjelmaa. Valmiiden 3D-mallien tekstuurit luodaan Substance Painter -teksturointiohjelmassa ja muut tarvittavat grafiikat Photoshop-kuvankäsittelyohjelman avulla. 3D-malleista luodaan ensin korkea- ja matalaresoluutioiset versiot, jotka tallennetaan tiedostoformaattiin, jota voidaan käyttää teksturointiohjelmassa.

Matalaresoluutioinen malli ladataan teksturointiohjelmaan ja korkearesoluutioisen mallin avulla tähän siirretään pinnan yksityiskohdat sekä muu tieto, joilla helpotetaan teksturointiprosessia. Malli teksturoidaan asianmukaisesti, jonka jälkeen tästä saadut tekstuurikartat ja 3D-malli viedään ulos ohjelmasta käytettäväksi Spark AR Studiossa, jossa lopullisen lisätyn todellisuuden efektin toiminnallisuudet toteutetaan. Maalivahdin animointiin käytetään Adoben Mixamo-animaatiopalvelua ja animaatiot yhdistetään yhteen tiedostoon Blender-mallinnusohjelman avulla ennen Spark AR Studioon vientiä.

5 PROJEKTIN MATERIAALIN TUOTTAMINEN

5.1 Jalkapallomalli

Perinteisen jalkapallon kuviointi on samanlainen kuin niin kutsutussa tyypistetyssä ikosaedrissa, joka koostuu 20 säännöllisestä valkoisesta kuusikulmiosta ja 12 mustasta säännöllisestä viisikulmiosta. 3ds Max -ohjelmassa tämä on mahdollista luoda Extended Primitives -valikon Hedra-nimisellä kappaleella. Kappaleen perheeksi valitaan Dodec/Icos, sillä haluttu kappale on ikosaedri. P- ja Q-parametreja muuttamalla on mahdollista säätää tahkojen muotoa; asettamalla P-parametrin arvoksi 0,37 saadaan kuusi- ja viisikulmioista mahdollisimman säännöllisen muotoiset (kuva 13). Muut kappaleen parametrit voidaan jättää oletusarvoihin.

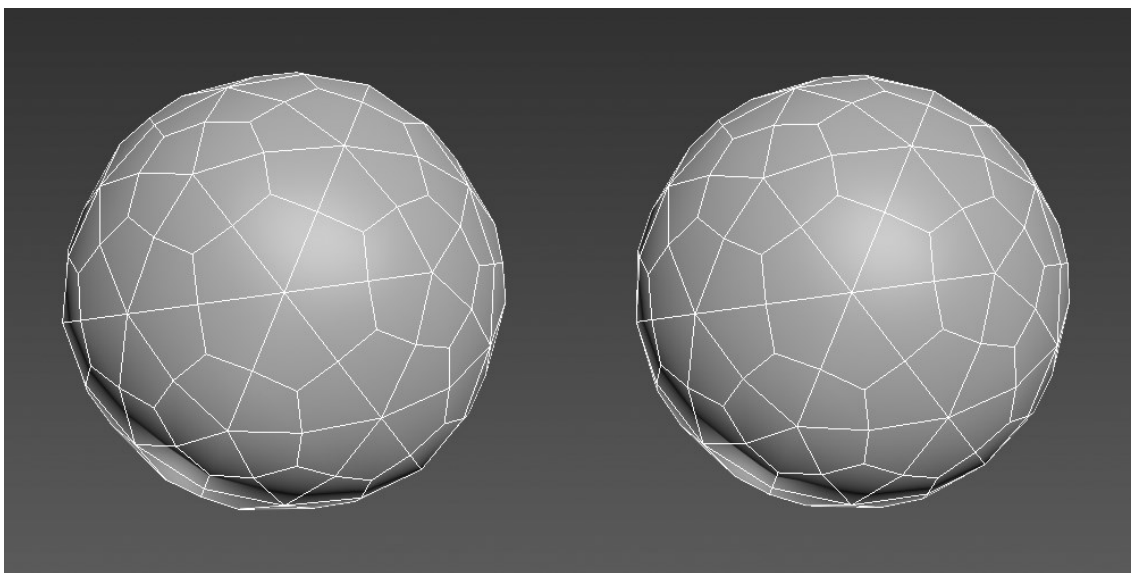


KUVA 13. Mallin geometria lähtötilanteessa

Mallintamisen myöhemmässä työvaiheessa kappaleen geometrian ollessa huomattavasti alkutilannetta monimutkaisempi, jalkapallon viisi- ja kuusikulmaiset paneelit täytyy pystyä valitsemaan nopeasti jollain tapaa, sillä käsin tämä olisi todella hidasta. Tähän voidaan käyttää esimerkiksi materiaalitunnisteita. Lisäämällä Edit Poly -modifikaattori voidaan valintatyökalun avulla valita ensin kaikki

polygonit, joissa on tasan viisi sivua ja antaa Material ID:n arvoksi 1. Tämän jälkeen voidaan tehdä käänteinen valinta ja asettaa näiden polygonien Material ID:n arvoksi 2.

Koska kappale on toistaiseksi kulmikas eikä vielä muistuta jalkapalloa, voidaan siitä tehdä pyöreämpi subdivision surface -algoritmilla. Tämä tapahtuu lisäämällä kappaleelle TurboSmooth-modifikaattori. Modifikaattori ilmestyy kappaleen yläpuolelle niin sanottuun pinon, josta sen parametreja pääsee muokkaamaan missä tahansa mallintamisen vaiheessa. Tällä tavalla muutosten tekeminen on nopeampaa, mikäli kappaleisiin halutaan tehdä muokkauksia myöhemmässä vaiheessa. On kuitenkin syytä noudattaa varovaisuutta pinon pohjimmaisten parametrien asetusten säätämisessä kun pinossa on useampi modifikaattori, sillä jos kärkipisteiden määrä mallissa muuttuu jonkun parametrin muokkauksen seurauksena, kappaleen geometria todennäköisesti vääristyy. TurboSmooth-modifikaattorin lisäämisen jälkeen alkaa pallon kuviointi muistuttaa enemmän haluttua lopputulosta. Kappale ei ole vielä kuitenkaan täysin pyöreä, mutta tämän korjaamiseksi voidaan käyttää Spherify-modifikaattoria, joka pakottaa kappaleen pallon muotoon (kuva 14).

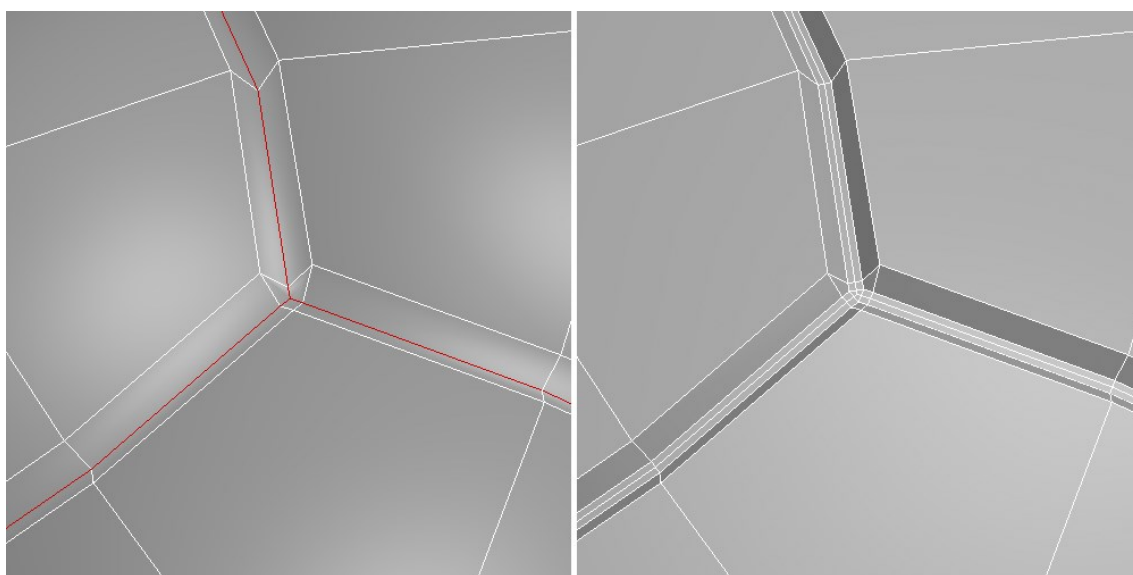


KUVA 14. Malli TurboSmooth- ja Spherify-modifikaattorien lisäämisen jälkeen

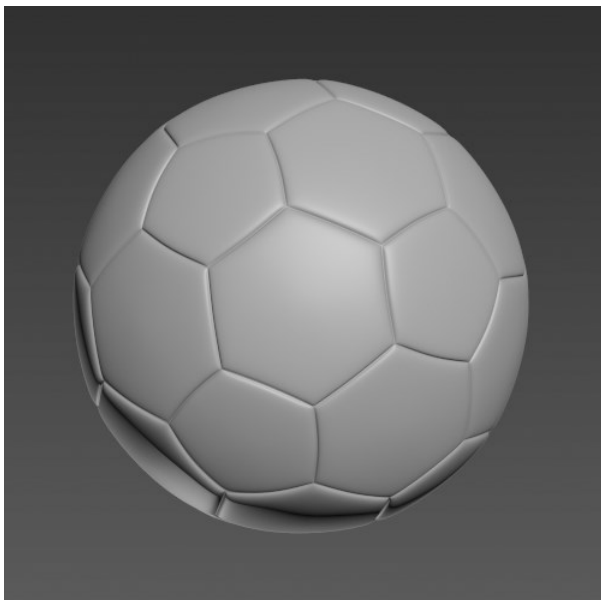
Jotta jalkapallo ei olisi täysin tasainen ja sen paneelit erottuisivat saumakohdista, täytyy sen pintaa muokata. Jotta kappaleen kärkipisteitä, sivuja ja tahkoja pystyt-

täisiin muokkaamaan, lisätään pinoon Edit Poly -modifikaattori. Nyt kun pystymme valitsemaan kappaleen eri komponentteja, voimme määrittää jalkapallon saumakohdat valitsemalla niiden polygonien sivut, joihin haluamme saumojen tulevan. Koska käsin tämä olisi aikaa vievä prosessi, voimme käyttää valintatyökalua, joka valitsee sivut automaattisesti kappaleen pinnan koveruuden perusteella. Kun halutut sivut on valittu, lisätään pinoon Chamfer-modifikaattori, jolla saa lisättyä geometriaa kappaleeseen saumakohtien tekemistä varten. Amount-parametria muuttamalla sauman leveyden saa haluamakseen ja tarvittaessa tätä on mahdollista muokata jälkikäteen.

Koska pallo on vielä tasainen, täytyy paneelien saumakohtiin saada korkeuseroa. Tämä saavutetaan lisäämällä uusi Edit Poly -modifikaattori, jonka avulla voidaan valita sauma-alueiden keskimmäiset sivut. Kun valinta on valmis, lisätään pinoon Push-modifikaattori, jolla työnnetään valittuja komponentteja niiden normaalien suuntaisesti. Koska jalkapallon paneelien halutaan ulkonevan hieman pallon pinnasta, annetaan Push Value -parametriksi jokin negatiivinen arvo, jolloin saumat painautuvat sisäänpäin kohti pallon keskikohtaa. Kun Push-modifikaattori on valittuna pinossa, uudella Chamfer-operaatiolla voidaan tukea saumojen muotoa (kuva 15). Lopuksi lisätään uusi TurboSmooth-modifikaattori, jolla kasvatetaan mallin tarkkuutta, jotta pallon pinta vaikuttaisi mahdollisimman tasaiselta, jonka jälkeen romautetaan modifikaattoripino (kuva 16).

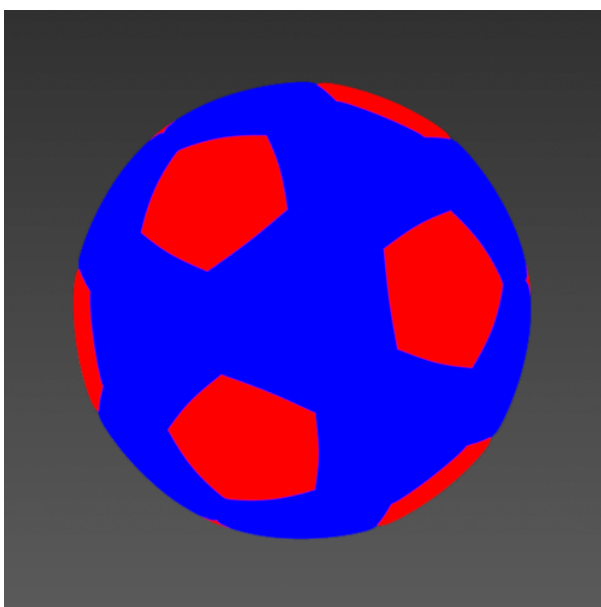


KUVA 15. Kappaleen valitut sivut Chamfer-operaatiota ennen ja sen jälkeen



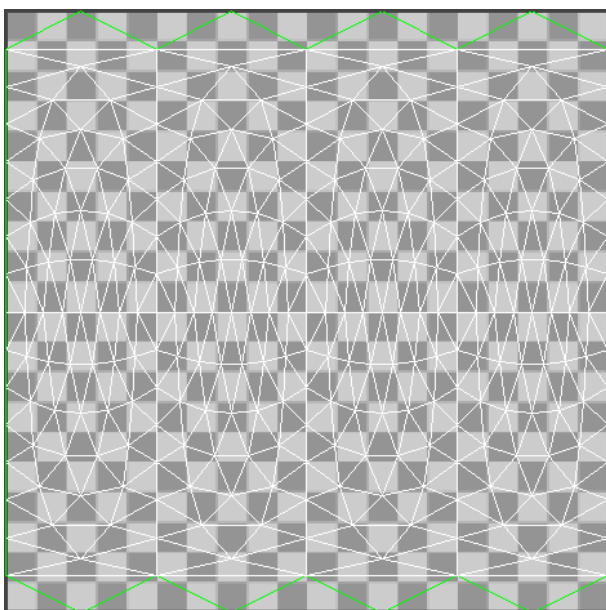
KUVA 16. Valmis korkearesoluutioinen malli jalkapallosta

Teksturoinnin helpottamiseksi kappaleen kärkipisteisiin voidaan tallentaa väri-informaatiota, jota voidaan käyttää yksityiskohtien siirtämisessä low poly -malliin. Kappaleen polygonitasolla valitaan Select ID -painikkeella polygonit, joiden Material ID on 1, jonka jälkeen Vertex Colors -valikosta valitaan jokin erottuva väri, kuten punainen, joka tallentuu valittuihin polygoneihin kuuluville kärkipisteille. Tämän jälkeen tehdään käänteinen valinta ja vaihdetaan kärkipisteiden väriksi esimerkiksi sininen (kuva 19). Malli on nyt käytännössä valmis ja se voidaan piilottaa siksi aikaa, kunnes jalkapallon matalaresoluutioinen versio on mallinnettu.

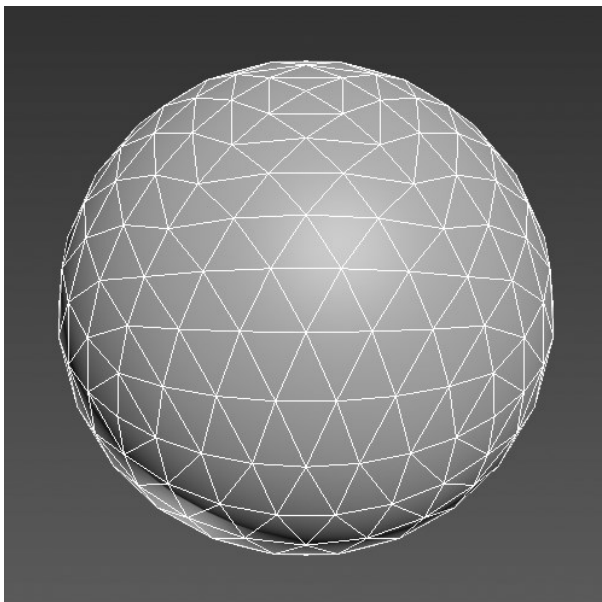


KUVA 19. Mallin kärkipisteisiin tallennetut värit visualisoituna

Pallon low poly -malli voi olla oikeastaan mikä tahansa pallon muotoinen kappale. Tärkeintä on, että kyseisen kappaleen UV-kartoitus saadaan tehtyä mahdollisimman tehokkaasti, jotta tekstuuri voi hyödyntää koko UV-aluetta, sillä tämä vaikuttaa lopullisen tekstuurin tarkkuuteen. Yksi tapa on käyttää GeoSphere-primitiiviä, jonka geodeettiseksi tyyppiä valitaan Octa-vaihtoehto. Säteen arvoksi annetaan 0,11m, jolloin pallostaa tulee realistisen kokoinen eli halkaisijaltaan 22 senttimetriä. Segmenttejä puolestaan tulee olla riittävästi, jotta kappale näyttäisi tarpeeksi pyöreältä. Lisäksi varmistetaan, että Generate Mapping Coordinates -vaihtoehto on valittuna. Tällä varmistetaan, että kappaleelle luodaan automaattisesti oikeanlaiset UV-koordinaatit, joita ei tässä tapauksessa tarvitse muokata itse ollenkaan. Tämän jälkeen kappale voidaan muuttaa Editable Poly -muotoon. Unwrap UVW -modifikaattorilla voidaan vielä tarkistaa, että UV-kartoitus on oikeanlainen (kuva 20). Kuvassa 21 näkyy pallon geometria.



KUVA 20. GeoSphere-primitiiville automaattisesti luotu UV-kartoitus



KUVA 21. Jalkapallon matalaresoluutioinen malli

Lopuksi tuodaan high poly -malli jälleen näkyviin ja verrataan sen kokoa äsken tehdyn mallin kokoon. Jos korkearesoluutioinen malli on kooltaan isompi tai pienempi, muutetaan sen kokoa skaalaamalla kaikkia kappaleen kärkipisteitä alaspäin tai ylöspäin, kunnes koko vastaa mahdollisimman tarkasti toista kappaletta. Mallin ollessa valittuna se tallennetaan FBX-formaatissa nimellä *SoccerBall_high* käyttäen Export Selected -toimintoa. Toinen malli tallennetaan samalla tavalla, mutta nimeksi tulee *SoccerBall_low*.

5.2 Jalkapallon tekstuuri

Teksturointi aloitetaan luomalla uusi projekti Substance Painterissa. Projektissa käytettäväksi tiedostoksi valitaan *SoccerBall_low*. Pudotusvalikoista asetetaan dokumentin resoluutioksi 1024 ja normaalikartan formaatiksi OpenGL. Kun projekti on luotu, avataan ensimmäisenä Baking-ikkuna Texture Set Settings -ikkunasta painamalla Bake Mesh Maps -nappia. *SoccerBall_high*-malli lisätään Common parameters -osiossa näkyvään High Definition Meshes -listaan painamalla listan vieressä olevaa painiketta. Muut ikkunan asetukset voidaan jättää oletusarvoihin; haluttaessa tekstuurien laskostumista voi vähentää laittamalla Antialiasing-asetuksen päälle, mikä vaikuttaa hieman renderöintiin. ID-välilehdeltä on syytä myös tarkistaa, että Color Source -asetus vastaa tapaa, jolla informaatio maskeja varten toteutettiin mallinnusohjelmassa, tässä tapauksessa valitaan siis

Vertex Color, sillä värit tallennettiin kappaleen kärkipisteisiin. Lopuksi välilehtien valintaruuduilla valitaan, mikä informaatio on tarkoitus siirtää teksturoitavaan malliin. Thickness-kartalle ei tässä tilanteessa ole käyttöä, joten se voidaan jättää pois baking-prosessista. Ohjelma aloittaa tekstuurien laskemisen kun Bake Mesh Maps -nappia on painettu.

Kun tekstuurit on laskettu, voidaan mallille alkaa lisätä materiaaleja. Materiaalit toimivat tasoina samaan tapaan kuin esimerkiksi Photoshop-kuvankäsittelyohjelmassa: Layer-ikkunan päällimmäinen materiaali peittää alleen muut tasot mallin pinnassa, ellei tälle ole määritetty maskia. Maskien avulla tasojen sisältöä voidaan pyyhkiä pois tai vastaavasti tuoda näkyviin ilman, että itse sisältöä muokataan: tämä toimii samaan tapaan kuin läpinäkyvyyskartta, eli musta väri piilottaa ja valkoinen tuo materiaalin näkyviin. Materiaalien lisääminen on viisainta aloittaa miettimällä, mikä on kappaleen pintamateriaali, toisin sanoen, mistä materiaalista tai materiaaleista kappale koostuu. Koska kyseessä on jalkapallo, voidaan päätätä, että se on valmistettu esimerkiksi keinonahasta. Substance Painterista löytyy valmiiksi melko kattava valikoima eri materiaaleja, kuten juurikin tarvitsemamme keinonahka. Materiaali raahataan hyllystä (Shelf) Layer-ikkunaan, jolloin koko kappale saa kyseisen materiaalin ominaisuudet. Oletuksena käytetty materiaali on musta, joten väri muutetaan valkoisen sävyiseksi, jotta saadaan perinteiselle jalkapallolle tyypillinen pohjaväri.

Jalkapallon mustia paneeleita varten luodaan uusi täyttötaso (fill layer), joka laitetaan lähes mustan väriseksi. Ei kuitenkaan täysin mustaksi, sillä luonnossa harva pinta on täysin musta tai valkoinen. Koska tämä taso peittää nyt värillään koko mallin, täytyy sille lisätä maski. Add mask with color selection -toiminnolla voidaan käyttää mallinnettaessa värjättyjen kärkipisteiden alueita maskina. Kun tason maski on valittuna, painetaan Pick color -nappia, jolloin mallin pintaan ilmestyy näkyviin ID-kartta, josta valitaan hiirellä haluttu väri maskiksi, siis punainen. Jalkapallon viisikulmaiset paneelit on värjätty nyt mustiksi ja koko kappaleella on realistinen materiaali (kuva 22).



KUVA 22. Jalkapallo, jolle on lisätty pohjamateriaali

Jalkapallo pintaan voidaan myös lisätä käytön jälkiä, jotta se ei olisi niin ilmeettömän näköinen. Pallossa voisi esimerkiksi olla kuraa ja nurmikkoa, jota on tarttunut sen pintaan kun sitä on potkittu. Likaa voidaan lisätä esimerkiksi lisäämällä uusi ruskean värinen täyttötaso, jolle annetaan musta maski. Maskissa käytetään jotakin tyyliin sopivaa, pyyhkäisyjälkiä jäljittelevää harmaasävyistä bittikarttakuvaa, jolla saadaan aikaan kuraa muistuttavia jälkiä. Maskin asetuksia säätämällä saadaan sopiva määrä likaa näkyviin. Lisäksi täyttövärille voidaan lisätä filteri, joka antaa materiaalille rosoisen ulkonäön. Materiaalin ominaisuuksista säädetään lopuksi karheus sopivalle tasolle, jotta valo ei heijastu kirkkaasti kuran pinnasta.

Tämän jälkeen edellinen taso voidaan kopioida ja sen maski tyhjentää tai poistaa. Maskin tilalle raahataan hyllystä valmis niin sanottu älykäs maski (smart mask), mikä lisää roiskeiden näköistä yksityiskohtaa tasolle. Samalla tapaa kuin edellisenkin tason kanssa, maskin asetuksista valitaan sopiva tasapaino lian määrälle. Lopuksi edellinen taso kopioidaan vielä kerran ja toistetaan samat vaiheet kuin edellisellekin tasolle, mutta tällä kertaa maskina käytetään erilaista älykästä maskia, joka lisää likaa kappaleen koverille alueille tämän pinnanmuotojen perusteella, sillä kuraa todennäköisesti kertyisi myös pallon paneelien saumakohtiin.

Viimeisenä yksityiskohtana lisätään irtonaisia ruohonkorsia ympäri pallon pintaa. Tähän käytetään proseduraalista nurmikkomateriaalia, joka lisätään uuden täytötason maskille käyttämällä generator-efektiä. Tällä tavoin ruhosta ei tule peittävää mattoa, vaan yksittäisiä korsia saadaan ripoteltua tasaisesti ympäriinsä. Kun ruohon asetukset on säädetty sopiviksi, lisätään maskiin levels-efekti, jolla ruohomateriaali saadaan erottumaan paremmin pinnasta. Taso varten luodaan kansio, jossa on maski, jonka sisään taso sitten raahataan. Tämän avulla ruohon peittämiä alueita saadaan rajoitettua ja lopputuloksesta tulee satunnaisemman näköinen.

Kun lopputulokseen ollaan tyytyväisiä, tekstuurit voidaan viedä ulos ohjelmasta Export Textures -toiminnolla. Avautuneesta ikkunasta valitaan konfiguraatioksi glTF PBR Metal Roughness -vaihtoehto, jotta tekstuurit toimivat oikein Spark AR Studio PBR-varjostimen kanssa. Muita ikkunan asetuksia ei ole tarpeen muuttaa, kunhan tekstuurien koko ei ylitä 1024x1024 kuvapistettä. On kuitenkin syytä ottaa huomioon kappaleen fyysinen koko ja päättää sen perusteella, kuinka ison tekstuurikoon malli tarvitsee. Pienemmät tai pelaajasta kauempana olevat kappaleet eivät välttämättä tarvitse niin isoja tekstureita, kuin mitä isokokoiset ja läheltä nähtävät mallit, mikä on lisäksi hyvä tapa optimoida tilankäyttöä. Export-nappia painettaessa kartat tallennetaan valittuun tiedostopolkuun. Kuvassa 23 on esitetty valmis teksturoitu jalkapallo.



KUVA 23. Valmis malli renderöitynä Substance Painteristä löytyvällä Iray-renderointiohjelmalla

5.3 Jalkapallomaalin malli

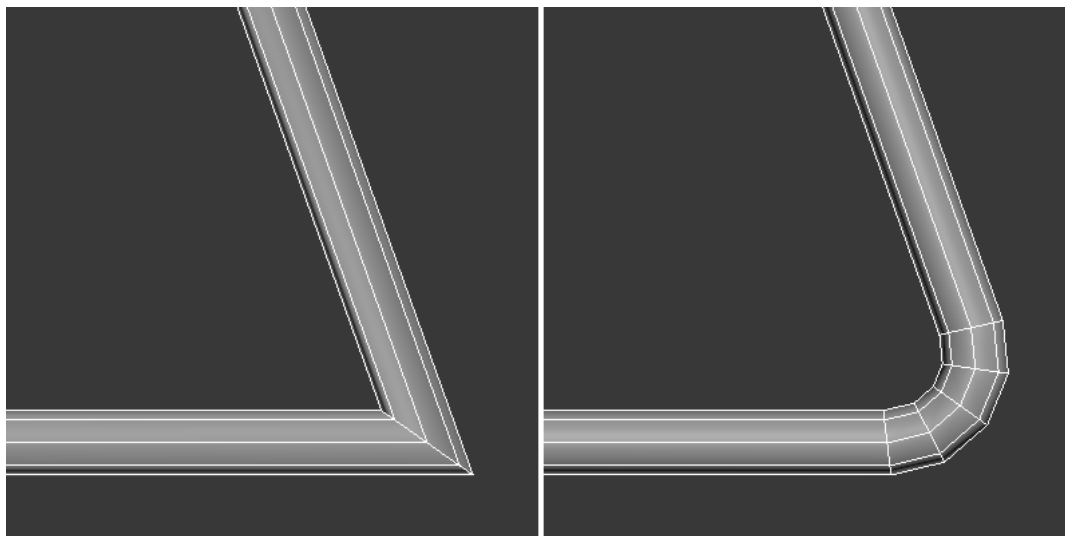
Jalkapallomaalin mallintaminen on puolestaan hieman yksinkertaisempi prosessi. Ennen aloittamista on kuitenkin hyvä tarkistaa, minkä kokoinen oikea jalkapallomaali on, jotta mallista tulee uskottava. Saatujen mittojen perusteella voidaan luoda laatikko, jolle annetaan maalin dimensiot.

Mallintaminen aloitetaan piirtämällä maalin kehikot splineillä, joka tapahtuu Line-työkalulla. Ensin piirretään etutanko käyttämällä apuna laatikkoa, jonka teimme aikaisemmin. Snap-työkalun avulla viivan ensimmäinen piste voidaan asettaa laatikon pitkän sivun jompaankumpaan alanurkkaan, jonka jälkeen luodaan uusi piste saman kyljen yläpäähän, jatketaan pitkää sivua laatikon toiseen reunaan ja tästä suoraan alas aloituspisteen vastakkaisen pään alanurkkaan. Lopuksi päätetään viivan piirtäminen painamalla hiiren oikeaa painiketta.

Muokkaamalla viivan parametreja saadaan tälle luotua renderöitävää geometriaa. Enable In Viewport -asetuksen valitsemalla viiva voidaan piirtää joko suorakaiteen tai lieriön muotoisena työskentelytilassa. Tässä tapauksessa käytetään siis Radial-vaihtoehtoa, jonka paksuudeksi annetaan esimerkiksi 0,1m. Sivuja kappaleella voi olla vaikka 32, jotta kappale vaikuttaa tarpeeksi pyöreältä. Kappaleesta tehdään kopio matalatarkkuuksista mallia varten, jonka sivujen määrä lasketaan ainakin puoleen alkuperäisestä, jonka jälkeen se voidaan piilottaa.

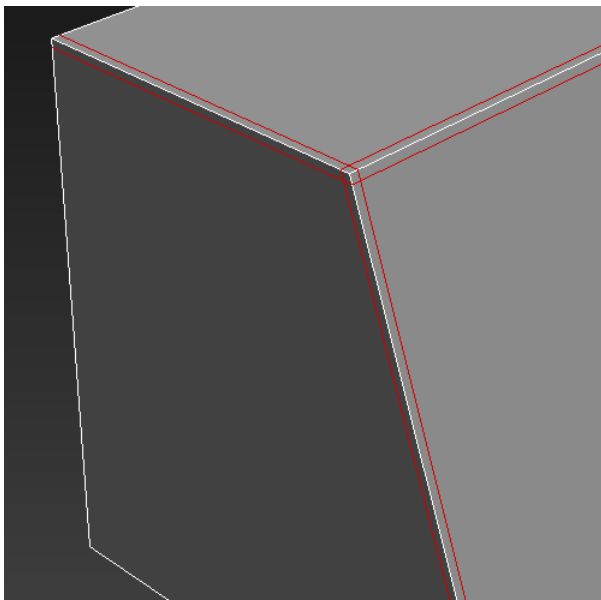
Maalin sivun kehikot mallinnetaan aloittamalla samasta kohtaa kun edellinenkin tanko, mutta tällä kertaa viiva vedetään aloituspisteestä lyhyttä sivua pitkin maalin takaosaan, tästä viistosti ylös yläsivun keskikohtaan ja lopuksi maalin etuosaan etukehikon kulmaan, jonka jälkeen katkaistaan viiva. Kappaleelle voidaan antaa samat parametrit kuin aikaisemminkin, mutta paksuutta voidaan säätää esimerkiksi puoleen alkuperäisestä. Myös sivujen määrää voidaan vähentää, koska tanko on kapeampi. Tämän jälkeen kappale voidaan kopioida ja siirtää maalin toiseen päähän. Kummatkin kopiot yhdistetään lopuksi yhdeksi kappaleeksi Attach-toiminnolla. Tankojen taivutettuihin kulmiin saadaan pyöreyttä valitsemalla nurkkien kärkipisteet ja käyttämällä Fillet-toimintoa (kuva 24). Splinein in-

terpolaation arvoa muuttamalla käyrän kaarevien kohtien tarkkuutta voidaan hieman lisätä. Myös tästä kappaleesta tehdään yksinkertaistetumpi kopio vähentämällä sivujen määrää ja interpolaatiota, joka sitten piilotetaan.



KUVA 24. Splinin kulma ennen Fillet-toimintoa ja sen jälkeen

Maaliverkon tekemiseen voidaan käyttää alussa luotua laatikkoa, josta tehdään ensin kopio. Kopioitu laatikko muotoillaan kehon kulmien mukaisesti, jonka jälkeen kappaleen sisimmäisten sivujen ympärille lisätään geometriaa tukemaan kappaleen muotoa (kuva 25). Kappaleen tahkoja voidaan lisäksi jakaa Connect-työkalun avulla, jotta polygoneista tulisi yhtenäisemmän muotoisia. Tämän jälkeen lisätään TurboSmooth-modifikaattori, jotta kappaleelle saadaan tarpeeksi geometriaa, minkä avulla verkosta voidaan tehdä realistisemmän näköinen. Tähän käytetään Cloth-modifikaattoria, jolla simuloidaan kankaan ominaisuuksia. Modifikaattorin lisäämisen jälkeen maaliverkon alapuolelle luodaan verkon pinta-alan kattava taso esimerkiksi laatikkoprimitiivillä, jotta verkko laskostuu maata vasten. Cloth-modifikaattorin Group-tasolla valitaan verkon kärkipisteet niiltä sivuilta, jotka peittyvät maalin kehojen alle ja näille luodaan uusi ryhmä Make Group -toiminnolla. Ryhmän tyypiksi valitaan Drag, jotta valitut kärkipisteet pysyvät simulaatiossa paikoillaan. Näin saadaan aikaan vaikutelma, että verkko on fyysisesti kiinnitetty metallitankoihin.



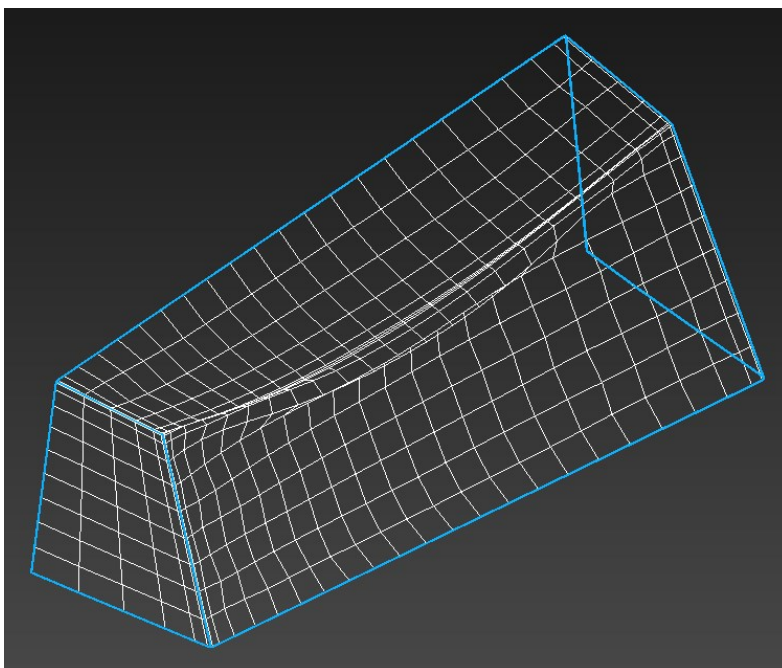
KUVA 25. Lisätyt reunaluopit punaisella valintavärillä korostettuina

Cloth-modifikaattorin Object Properties -ikkunassa määritetään simulaatiossa käytettävät kappaleet. Ohessa olevaan listaan lisätään aikaisemmin luotu taso, jonka tyyppiä valitaan Collision Object eli kappale, mihin simuloitavan esineen on mahdollista törmätä. Itse verkko määritetään kankaaksi valitsemalla Cloth-vaihtoehto, jonka jälkeen on mahdollista muokata kankaan ominaisuuksia eri parametreilla tai valita tiputusvalikosta jokin esiasetettu tekstiilityyppi tai materiaali. Asetukset tallennetaan OK-painikkeella, jonka jälkeen simulaatio voidaan suorittaa painamalla Simulate Local -nappia. Kun kappale on asettunut paikoilleen, voidaan simulaatio pysäyttää painamalla samaa nappia uudestaan. Lopuksi lisätään Edit Poly -modifikaattori ja käännetään kappaleen tahkojen normaalit Flip-toiminnolla, jotta verkko on mahdollista nähdä maalin etupuolelta. Verkon resoluutiota voidaan myös kasvattaa uudella TurboSmooth-modifikaattorilla, jos pinnassa on havaittavissa kulmikkua.

Valmiista verkon mallista tehdään kopio ja pinoon viimeisimpänä lisätty TurboSmooth-modifikaattori poistetaan, jonka jälkeen kyseinen kappale piilotetaan. Jokainen esillä oleva kappale voidaan nyt muuttaa Editable Poly -muotoon ja yhdistää toisiinsa Attach-toiminnolla, jolloin high poly -malli on käytännössä valmis. Kappale voidaan tallentaa tässä vaiheessa FBX-formaatissa nimellä *Soccer-Goal_high* ja piilottaa näkyvistä.

Muut tiedoston kappaleet tuodaan nyt jälleen näkyviin. Verkon mallia voidaan yksinkertaistaa valitsemalla joka toinen kappaleen reunaluoppi, jonka jälkeen ne poistetaan. Tämä voidaan tarvittaessa toistaa, jos mallista halutaan vieläkin yksinkertaisempi. Lisäksi käännetään kaikkien tahkojen normaalit samaan tapaan kun alkuperäisen verkonkin kanssa. Muut kappaleet voidaan tässä vaiheessa yhdistää verkkoon käyttämällä Attach-toimintoa.

Tämän jälkeen kappaleelle lisätään Unwrap UVW -modifikaattori ja avataan UV-editori. Verkolle lisätään saumatyökalulla saumat kuvan 26 mukaisesti, jolloin sen levittäminen tasoon onnistuu nopeasti Quick Peel -työkalulla. Kehikoiden UV:t puolestaan leikataan erilleen niistä kohdista, joissa on tarpeeksi jyrkkä kulma, valitsemalla ensin haluttu osa kappaleesta ja käyttämällä sitten joko Break- tai Detach Edge Verts -toimintoa. Relax-toiminnolla UV:t saadaan vastaamaan paremmin oikeaa muotoaan. Lopuksi avataan Pack-ikkuna, jolla UV:t pakataan optimaalisesti UV-alueelle. Padding-parametrin arvoksi voidaan antaa 0,01, mutta muut ikkunan asetukset voidaan pitää oletuksena. Seuraavaksi painetaan OK, jolloin ohjelma pakkaa UV:t valittujen asetusten mukaisesti. Tarvittaessa järjestykseen voidaan tehdä käsin hienosäätöä, sillä ohjelman suorittama asetteleminen ei aina onnistu täydellisesti. Lopuksi romautetaan kappaleen modifikaattoripino, jotta tehdyt muutokset jäävät muistiin ja tallennetaan malli FBX-formaattiin nimellä *SoccerGoal_high*.

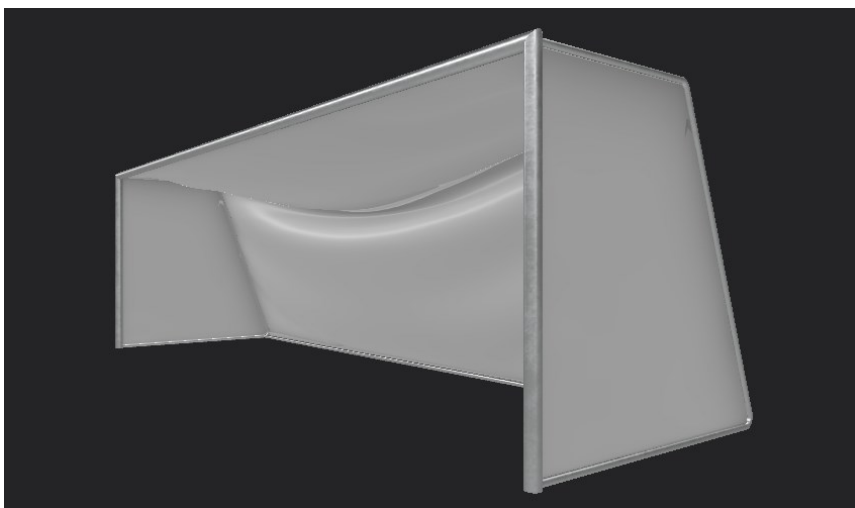


KUVA 26. Kappaleelle lisätyt saumat näkyvät syaanilla värillä

5.4 Jalkapallomaalin tekstuuri

Maalin teksturointi aloitetaan samoin kun jalkapallonkin, eli uuden projektin asetuksista valitaan teksturoitavaksi malliksi *SoccerGoal_low*. Muut asetukset pysyvät samoina kuin aikaisemmassakin projektissa. Ensimmäisenä avataan jälleen Baking-ikkuna ja valitaan yksityiskohtien siirtoa varten *SoccerGoal_high*-malli High Definition Meshes -listaan. Muilta osin asetukset ovat samat kuin aiemminkin, mutta tällä kertaa Thickness-kartan lisäksi ID-kartta voidaan jättää lisäämättä prosessiin. Valittujen karttojen luomisen jälkeen voidaan aloittaa materiaalien valitseminen mallille. Jalkapallomaalille tarvitaan käytännössä vain kaksi eri materiaalia ja koska se näkyy pelissä taka-alalla, ei pienten yksityiskohtien lisääminen tässä tapauksessa ole kovin oleellista.

Ensimmäisenä voidaan lisätä jokin metallinen materiaali maalin kehikkoa varten. Nämä vaikuttaisivat olevan yleensä valmistettu alumiinista, jolle ohjelmasta löytyy muutama erilainen materiaali valmiina. Esimerkiksi kulunut harjattu alumiininen materiaali sopii tähän hyvin, sillä kappaleen ei haluta näyttävän täysin käyttämättömältä. Materiaali raahataan Layer-ikkunaan, jonka jälkeen sille luodaan musta maski. Maskin ollessa valittuna otetaan käyttöön Polygon Fill -työkalu, jonka Mesh fill -toiminnolla valitaan jokainen tankokappale niitä klikkaamalla. Materiaalin ominaisuudet koskettavat nyt siis vain valittuja mallin osia (kuva 27).



KUVA 27. Jalkapallomaali pelkän alumiinimateriaalin kanssa

Seuraavaksi lisätään jalkapallomaalin verkon materiaali. Projektin oletusmateriaaleista ei tässä tapauksessa löydy suoraan materiaalia, jota voisi käyttää verkona. Materiaali olisi mahdollista tehdä myös itse, mutta ajan säästämiseksi voidaan käyttää ohjelmasta löytyvää Substance Source -materiaalikirjastoa, mistä on mahdollista ladata uusia materiaaleja projekteissa käytettäväksi. Materiaalikirjastosta löytyy sopiva synteettisen verkon materiaali, joka voidaan ladata ja siirtää uudeksi tasoksi mallille. Tälle luodaan myös musta maski ja Polygon Fill -työkalulla rajataan materiaalin pinnaksi maalin verkko. Materiaalin ominaisuuksia säätämällä verkon tiheydestä, paksuudesta sekä muista ominaisuuksista saadaan vakuuttavan näköiset vähällä vaivalla.

Teksturointi on tämän jälkeen käytännössä valmis (kuva 28), joten mallin tekstuurit voidaan viedä ulos ohjelmasta käyttämällä samoja asetuksia kuin mitä jalkapallon tekstuurien tallentamisessa käytettiin.



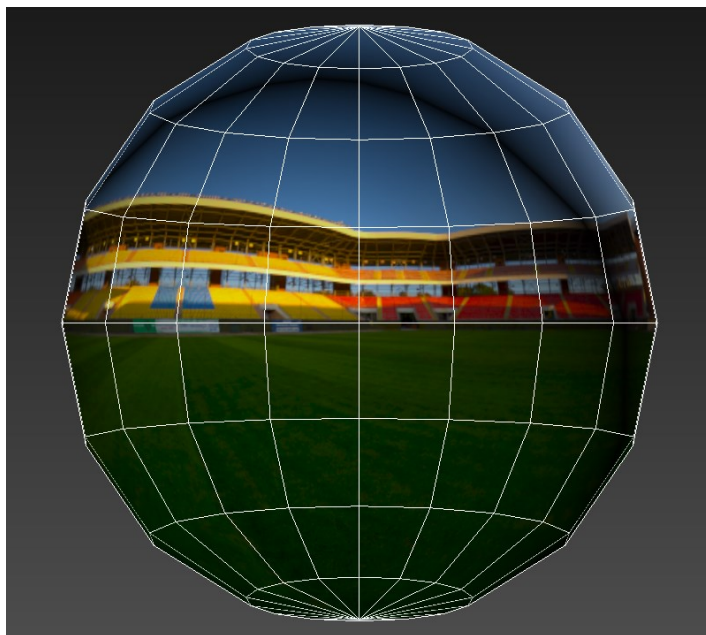
KUVA 28. Valmis jalkapallomaali renderöitynä samalla tavoin kuin jalkapallo

5.5 Ympäristö

Jalkapallokenttää ei tarvitse tai kannatakaan mallintaa itse, vaan geometrian säästämiseksi se voidaan toteuttaa tekstuureilla. Esimerkiksi HDRI Haven -si-

vusto tarjoaa täysin ilmaisia HDR-kuvia. Vaikka HDR-kuvia käytetään perinteisesti tietokonegrafiikalla tuotettujen skenejen valaisemiseen, voidaan tässä tapauksessa käyttää sellaista perinteisenä tekstuurina. Haluttu kuva ladataan sivustolta matalimmassa resoluutiossa, sillä Spark AR Studio ei tue yli 1024x1024 kuvapisteen tekstuureita. Ladattu HDR-kuva (liite 1) avataan kuvankäsittelyohjelmassa, esimerkiksi Photoshopissa, jonka jälkeen se muutetaan 8-bittiseksi. Tämän jälkeen kuva tallennetaan PNG-formaatissa.

Tekstuuria varten mallinnetaan yksinkertainen pallon muotoinen malli, jota käytetään pelialueen ympäröimiseen. Tähän tarkoitukseen sopii hyvin perinteinen Sphere-primitiivi, jonka halkaisijaksi asetetaan esimerkiksi yksi metri. Mittakavalla ei ole juurikaan väliä, sillä kappaletta voidaan skaalata halutun kokoiseksi Spark AR Studiassa. Pallo muutetaan Editable Poly -muotoon, jotta sen normaalit pystytään kääntämään ympäri, sillä malli on tarkoitus nähdä pelkästään sen sisäpuolelta. Kappaleen asetuksista voidaan laittaa Backface Cull -asetus päälle, jotta malli näkyy samoin kuin renderöitäessä eli vain polygonien etupuolet piirretään. Lopuksi lisätään aikaisemmin tehty teksturi mallille, jotta nähdään, että se näkyy oikein kappaleen pinnassa (kuva 29). Valmis malli tallennetaan samalla tavoin kuin edellisetkin mallit.

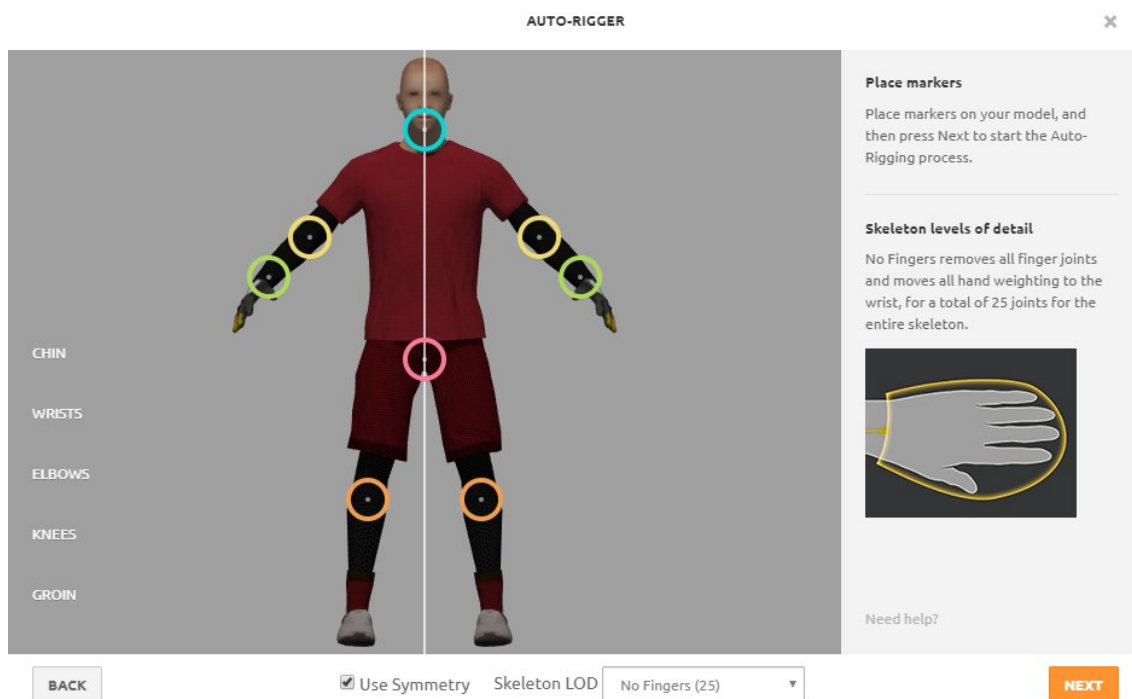


KUVA 29. Ympäristön vaikutelmaa jäljittelevä malli tekstuurin kanssa

5.6 Maalivahti ja animaatiot

Ajan säästämiseksi maalivahtin 3D-malli ja animaatiot päätettiin hankkia ulkopuolisista lähteistä. Projektiin sopiva malli löytyi CGTrader-sivustolta (liite 2) ja animaatioiden tekemiseen pystyttiin käyttämään Mixamo-animaatiokirjastosta löytyviä animaatioita. Mixamo-palvelun avulla hahmolle saatiin myös luotua hahmon animoinnille välttämätön luuranko, jota hahmolta ei valmiiksi löytynyt. Blender-mallinnusohjelmalla eri animaatiot pystyttiin yhdistämään yhdeksi tiedostoksi, mikä olisi esimerkiksi 3ds Max -ohjelmassa ollut hieman hankalampaa, ja Spark AR Studion dokumentaatioissa Blenderin käyttämistä tähän suositeltiin.

Mixamo-palveluun kirjaudutaan Adobe-tunnuksilla ja sen käyttäminen on täysin ilmaista. Palvelun käyttäminen aloitetaan lisäämällä malli animointia varten. Jos mallilla ei ole niin sanottua luurankoa, tämä voidaan luoda palvelun Auto-Rigger-toiminnon avulla (kuva 30). Kun prosessi on valmis, käyttäjälle esitetään esikatselu animoidusta mallista (kuva 31). Jos lopputulos vaikuttaa olevan kunnossa, voidaan siirtyä lisäämään animaatioita hahmolle tai tarvittaessa yrittää parantaa lopputulosta tekemällä prosessi uudestaan.



KUVA 30. Hahmon luuranko luodaan käyttäjän asettamien merkkien perusteella



KUVA 31. Esikatselu valmiista hahmosta

Palvelusta löytyy kattava valikoima erilaisia animaatioita useista eri kategorioista, kuten urheilu, taistelu ja jopa tanssi. Hahmolle valitaan animaatio listasta, jonka jälkeen animaation ominaisuuksia, kuten nopeutta ja pituutta, on mahdollista säätää. Valittu animaatio ladataan painamalla Download-painiketta, jonka jälkeen on mahdollista valita muun muassa animaation kuvataajuus sekä missä tiedostotyyppissä animaatio tallennetaan. Tässä tapauksessa käytetään FBX-formaattia ja kuvataajuutena 30 kuvaa sekunnissa. Hahmolle valitaan vain välttämättömimmät animaatiot, eli paikallaan seisominen, oikealle ja vasemmalle sivulle heittäytyminen sekä päällä puskeminen pallojen torjumista varten.

Ladatut tiedostot tuodaan Blender-mallinnusohjelmaan Import-toiminnolla yksi kerrallaan. Jokainen tuotu tiedosto lisää animaatioidatan lisäksi uuden hahmollin skeneen, joten kaikkien tiedostojen lisäämisen jälkeen ylimääräiset mallit voidaan poistaa, kunnes jäljellä on vain yksi hahmo. Vaikka ylimääräiset animoidut mallit ovatkin poistettu, pystytään niiden animaatioita yhä käyttämään, sillä tiedostoissa olevat hahmot ovat identtisiä toistensa kanssa. Eri animaatioita on mahdollista tarkastella siirtymällä käyttöliittymän Animation-välilehdelle, josta Dope Sheet -tilaksi valitaan Action Editor. Tässä näkymässä voidaan tiputusvalikosta valita toistettava animaatio sekä selvyuden vuoksi nimetä kukin uudelleen,

sillä oletuksena animaatiot ovat alkuperäisten nimiensä perusteella hankala erottaa toisistaan. Kun jokainen animaatio on lisätty ja nimetty asianmukaisesti, voidaan malli tallentaa jälleen FBX-formaatissa Export-toiminnolla.

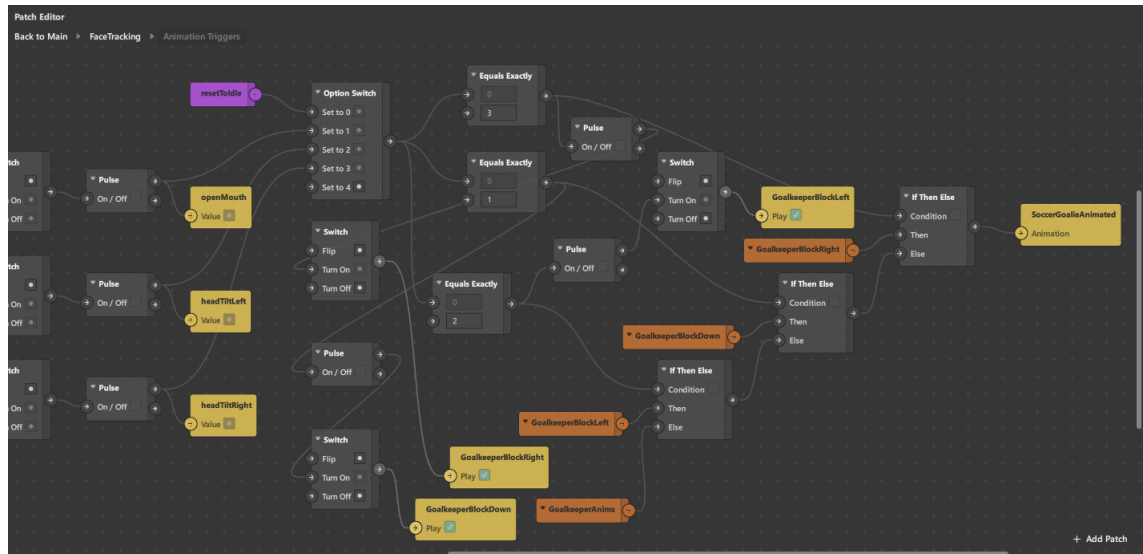
5.7 Spark AR Studio

Viimeisenä työvaiheena lisätään valmiit assetit aikaisemmasta Spark AR Studio -projektista tehtyyn kopioon, jotta valmiit ominaisuudet saadaan suoraan käyttöön. Kopion tekeminen vanhasta projektista on käytännössä helpompaa, kuin kopioida projektin hakemistosta ainoastaan tarvittavat tiedostot ja liittää ne uuteen, tyhjään projektiin. Ainoaksi tehtäväksi jää oikeastaan vanhojen peliobjektien korvaaminen uusilla malleilla ja tekstuureilla sekä pienten hienosäätöjen tekeminen.

Uusien asettien lisääminen tapahtuu Import From Computer -toiminnolla. Substance Painterista ulos vietyjen tekstuurien lisäksi projektiin kannattaa lisätä niiden mukana tallentuneet glTF-formaatin tiedostot, sillä ne pitävät sisällään myös itse 3D-mallin, mikä helpottaa asettien käyttämistä projektissa. Projektiin lisätään myös animoitu maalivahti ja tämän tekstuurit, ympäristöä varten tehty malli ja sen tekstuuri sekä sen tekemiseen käytetty alkuperäinen HDR-kuva, jota voidaan käyttää malleissa ympäristökarttana.

Peli käyttää kasvontunnistusta, jonka avulla käyttäjän kasvot piirretään maalivahdin pään paikalle käyttämällä mobiililaitteen etukameraa. Animoidun hahmon pään kohdalla on sen mukana liikkuva yleismalli kasvoista, joka on luotu ohjelman avulla. Kasvomalli käyttää reaaliajassa animoitua tekstuuria, joka saadaan kameran syötteestä. Näin luodaan vaikutelma, että käyttäjä itse olisi maalivahti. Käyttäjä ohjaa hahmon liikkumista pään ja kasvojensa avulla: maalivahti liikkuu oikealle tai vasemmalle maalin edessä, riippuen kumpaan suuntaan päätä kallistetaan, ja hyppää, jos käyttäjän ilme on hämmästynyt. Muitakin vastaavia toimintoja on mahdollista tehdä, mutta niiden lisääminen on todella hidasta, sillä ohjelma ei sovellu kovin hyvin monen eri animaation toistamiseen kyseisellä logiikalla. Vaikka pelin logiikka on toteutettu JavaScript-tiedostolla, täytyy esimerkiksi

animaatioihin liittyviin toimintoihin käyttää visuaalista Patch Editor -työkalua (kuva 32).



KUVA 32. Hahmon animaatioiden toistamiseen käytetty kuvaaja Patch Editor -ikkunassa

Jokainen projektiin lisätty malli voidaan raahata vuorotellen Assets-ikkunasta skenenäkymään ja asetella suurin piirtein paikoilleen (kuva 33). Kaikki vanhat assetit voidaan poistaa skenestä tai vaihtoehtoisesti piilottaa työskentelyn ajaksi. Tämän jälkeen JavaScript-tiedoston koodista täytyy poistaa kaikki viittaukset vanhojen assettien nimiin ja korvata ne niitä vastaavien uusien assettien nimillä, jotta peli toimii oikein. Jos jokin toiminnallisuuden sisältänyt assetti poistetaan skenestä, täytyy Patch Editorissa käydä linkittämässä uusi assetti tilalle, esimerkiksi pelihahmo, sillä muuten hahmon animaatiot eivät toistuisi ollenkaan pelattaessa.



KUVA 33. Uusien mallien avulla rakennettu valmis skene Spark AR Studiassa

Skeneen voidaan lisätä valonlähteeksi Directional Light -objekti, jolla pystytään jäljittelemään auringon valoa. Valo käännetään osoittamaan suunnilleen samaan suuntaan kuin mihin auringonvalo ympäristön tekstuurissa näyttää lankeavan, kuitenkin niin, että myös pelihahmon kasvot valaistuvat tarpeeksi eikä niihin tule selkeää varjoa. Valon väriksi valitaan vaalea, mutta hieman kellertävä sävy, joka vastaa taustalla näkyvää valaistusta. Valon voimakkuus säädetään esimerkiksi noin 75 prosenttiin.

Lopuksi varmistetaan, että jokaisen mallin materiaalit käyttävät oikeaa varjostinta ja että niiden asetukset ovat kunnossa. Ympäristön materiaalin varjostimen tyyppi on järkevintä olla "Flat", jotta skenessä käytetty valaistus ei näy skeneä ympäröivän pallon pinnassa. Kaikkien muiden mallien varjostimeksi valitaan "Physically-Based". Näille voidaan laittaa ympäristötekstuuriksi projektiin tuotu HDR-kuva, jolla saadaan kappaleiden pintaan ympäristöä jäljittelevät heijastukset. Kaikkien materiaalien asetuksista on lisäksi hyvä varmistaa, että jokainen käyttää värien koodauksessa lineaarista menetelmää sRGB:n sijaan, sillä materiaaleihin ei haluta gammakorjausta, varsinkin kun käytetään PBR-materiaaleja. Kuvassa 34 on esitetty kuvankaappaus pelinäköymästä.



KUVA 34. Valmis peli toiminnassa Spark AR Studion simulaattorissa

Valmiit efektit on mahdollista julkaista Instagram- ja Facebook-sovelluksille käyttämällä Facebookin Spark AR Hub -sivustoa. Tämä onnistuu lataamalla sivustolle efektin vientitiedosto, joka luodaan Spark AR Studion Export-toiminnolla. Jokainen ladattu efekti tarkistetaan ennen julkaisemista, jotta ne noudattavat palvelun käyttöehtoja. Esimerkiksi efektien tiedostokoolle on asetettu seuraavia rajoitteita: Facebookissa optimaalinen efektin koko on 2 megatavua tai vähemmän, kuitenkin maksimissaan 10 megatavua, muussa tapauksessa efektiä ei ole mahdollista julkaista. Instagramissa yläraja on 4 megatavua. Tämän vuoksi asettien luomisprosessissa on oltava säästeliäs, jotta tiedostojen yhteenlaskettu koko ei kasvaisi lopulta liian suureksi. Esimerkiksi tekstuurit vievät yleensä eniten tilaa, minkä takia on suositeltavaa käyttää matalaresoluutioisia tekstuureita, varsinkin jos projekti sisältää useampia eri 3D-malleja. Spark AR Studion avulla on myös mahdollista pakata tekstuureita jos ne vievät liikaa tilaa, mikä tosin laskee niiden tarkkuutta. Pakkaustasosta riippuen tämä voi vaikuttaa merkittävästi kuvanlaatuun. Parempi käytäntö olisikin siis laskea tekstuurin resoluutiota, jos mahdollista.

6 POHDINTA

Projektia tehdessäni opin käyttämään Spark AR Studiota kohtalaisesti ja ymmärän nyt paremmin, miten teksturoidut mallit saadaan helpoiten toimimaan kyseisessä sovelluksessa. Haastavinta projektin toteuttamisessa oli itse pelin rakentaminen, vaikka se veikin vähiten aikaa muihin työtehtäviin verrattuna. Isoimmaksi ongelmaksi tuli mittakaava, jota en ottanut kunnolla huomioon suunnitteluvaiheessa. Koska peli on tarkoitettu käytettävän mobiililaitteilla pystysuunnassa, aiheutti realistisen kokoinen maali päänvaivaa mallien asettelussa pelialueelle. Maali ei olisi voinut olla liian lähellä ruutua, sillä muuten käyttäjä ei olisi pystynyt näkemään maalia kokonaan. Ratkaisuna oli luonnollisesti siirtää maali tarpeeksi kauas, jotta maalin kummatkin tolpat näkyivät ruudulla. Toinen tästä aiheutuva ongelma oli maalivahti, jota pelattavuuden kannalta ei ollut mielekäästä siirtää niin kauas ruudusta, että hahmo olisi realistisen etäisyyden päässä maalista eli aivan sen edessä. Lisäksi käyttäjän kasvoja olisi vaikea erottaa kaukaa varsinkin laitteella, jossa on pieni näyttö. Tästä syystä päädyin kompromissiin asettelun, mittasuhteiden ja realismin suhteen: maalivahti on hieman isompi kuin maali ja jonkin verran kauempana siitä kuin esimerkiksi oikeassa jalkapallopelissä.

Myös mallien skaala projektiin tuotaessa herätti ihmetystä, sillä vaikka jokaisen mallin mitat oli varmistettu ennen FBX-formaattiin tallentamista, saattoivat ne Spark AR Studioon tuotaessa olla aivan eri mittasuhteissa kuin oli tarkoitettu. Merkitystä näytti olevan sillä, käyttikö alkuperäistä 3ds Max -ohjelmalla luotua FBX-mallia vai Substance Painterin luomaa glTF-formaatin mallia, joka osoittautui parhaaksi formaatiksi Spark AR Studion kanssa käytettäväksi. Joka tapauksessa tällä ei juurikaan ollut väliä, sillä malleja on mahdollista skaalata ohjelmassa, mikä on nopeampaa kuin muuttaa alkuperäisen mallin kokoa.

Huomasin mallit jo tehtyäni, että Facebook tarjoaa kehittäjille ilmaisia asetteja ohjelman tukisivulla, joita kehotetaan käyttämään mallina efektien luomisessa. Näiden avulla olisi ehkä ollut mahdollista luoda mallit vastaamaan paremmin ohjelmassa käytettyä mittakaavaa. Vaikeuksia tuotti myös hahmon pään ja kasvojen mallin sovittaminen yhteen. Tämä olisi voitu todennäköisesti myös saada toimimaan paremmin käyttämällä Facebookin tarjoamaa kasvomallia sapluunana, jonka avulla hahmon päähän olisi voitu leikata kasvoille sopiva aukko sen sijaan,

että kasvot pelkästään asetetaan pään eteen, sillä tämä tyypillisesti aiheuttaa ongelmia renderöitäessä. Pienistä virheistä huolimatta olen kuitenkin tyytyväinen lopputulokseen, sillä käyttäjän kasvot sopivat mielestäni visuaalisesti suhteellisen hyvin yhteen muiden grafiikoiden kanssa.

Pelissä käytetty hahmo olisi voitu mallintaa alusta asti itse, sillä käytetty malli ei ollut niin laadukas kuin olisin toivonut. Ominaisuuksiltaan malli kuitenkin sopi yhteen muiden projektin assettien kanssa ja ajoi asiansa, mutta todisti jälleen, että assettien hankkimisessa internetistä on aina riskinsä laadun suhteen. Myös efektin olisin voinut kokonaisuudessaan rakentaa itse Spark AR Studiossa, mutta tämä olisi edellyttänyt jonkin verran ohjelmoinnin opettelua. Aikataulun vuoksi päädyttiin siis käyttämään aikaisempaa projektia apuna. Pääpainona opinnäytetyössä oli kuitenkin grafiikan tuottamisen työnkulku lisätyn todellisuuden sovellukseen, joten ohjelmointi Spark AR Studiossa ei olisi varsinaisesti kuulunut enää käsittelemääni aihealueeseen. Mahdollisuus muokata projektia laajemmin olisi kuitenkin helpottanut sen kanssa työskentelemistä ja ongelmien korjaamista sekä nopeuttanut efektin valmistumista.

Suurin haaste tämän kaltaisen efektin tuottamisessa on kuitenkin efektitiedostojen sallittu koko, mikä rajoittaa todella paljon kolmiulotteisen pelin grafiikoiden toteuttamista, varsinkin jos tarkoitus on pyrkiä realismiin. Efektin suunniteltaessa olisikin hyvä pitää mielessä nämä rajoitukset, sillä ne voivat jonkin verran rajata mahdollisuuksia toteuttaa tietty tyyli. Projektista saamieni kokemusten perusteella voitaisiin todeta, että Facebook ja Instagram eivät ainakaan tällä hetkellä ole parhaimmat mahdolliset alustat lisätyn todellisuuden peleille, etenkin jos tarkoitus on esittää ruudulla enemmän sisältöä kuin mitä kameraefektit perinteisesti tekevät.

Koska efektin ei ehditty saattamaan julkaistavaan tilaan opinnäytetyön tekemisen aikana, voisi mahdollinen jatkotutkimus selvittää, miten efektien julkaiseminen Instagram- ja Facebook-sovelluksille käytännössä tapahtuu, miten julkaisuprosessi etenee sekä mitä muuta tiedostokoon lisäksi tulee ottaa huomioon ennen julkaisua.

LÄHTEET

- Becker, G. 2010. Challenge, Drama & Social Engagement: Designing Mobile Augmented Reality Experiences. Diaesitys. Luettu 23.10.2019. <https://www.sli-deshare.net/ubik/experience-design-for-mobile-augmented-reality>
- Chopine, A. 2011. 3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation. Focal Press.
- Gahan, A. 2012. 3ds Max Modeling for Games: Volume 2. Focal Press.
- Giamb Bruno, M. 2002. 3D Graphics & Animation, 2nd Edition. New Riders.
- Graig, A. 2013. Understanding Augmented Reality. Morgan Kaufmann.
- Halladay, K. 2019. Practical Shader Development: Vertex and Fragment Shaders for Game Developers. Apress.
- Kipper, G. & Rampolla, J. 2012. Augmented Reality: An Emerging Technologies Guide to AR. Rockland: Syngress.
- Lehtovirta, P. & Nuutinen, K. 2000. 3D-sisältötuotannon peruskirja. Jyväskylä: Docendo.
- McDermott, W. The PBR Guide by Allegorithmic. Volume 2: Practical Guidelines for Creating PBR Textures. Verkkokurssi. Luettu 12.9.2019. <https://academy.substance3d.com/courses/the-pbr-guide-part-2>
- Pharr, M., Jakob, W. & Humphreys, G. 2016. Physically Based Rendering, 3rd Edition. Morgan Kaufmann.
- Polycount Wiki. 2015. Environment Map. Wikisivu. Luettu 18.10.2019. http://wiki.polycount.com/wiki/Environment_map
- Polycount Wiki. 2015. Polygon Count. Wikisivu. Luettu 15.9.2019. http://wiki.polycount.com/wiki/Polygon_Count
- Polycount Wiki. 2015. Texture Baking. Wikisivu. Luettu 28.10.2019. http://wiki.polycount.com/wiki/Texture_Baking
- Russell, J. 2015. Basic Theory of Physically-Based Rendering. Blogikirjoitus. Julkaistu 1.11.2015. Luettu 12.9.2019. <https://marmoset.co/posts/basic-theory-of-physically-based-rendering/>
- Turbosquid. n.d. Topology. Verkkosivu. Luettu 24.9.2019. <https://www.turbosquid.com/3d-modeling/checkmate/checkmate-product-presentation/topology/>
- Unity Technologies 2019. Heightmap. Dokumentaatio. Luettu 4.12.2019. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterHeightMap.html>

Vaughan, W. 2011. [digital] Modeling. New Riders.

Vectary. n.d. UV Mapping. Dokumentaatio. Luettu 20.11.2019. <https://www.vectary.com/docs/uv-mapping/>

Watt, A. 2000. 3D computer graphics. Harlow: Addison-Wesley.

Zeman, N. 2014. Essential Skills for 3D Modeling, Rendering, and Animation. CRC Press.

LIITTEET

Liite 1. Projektissa käytetty HDR-kuva

https://hdrihaven.com/hdri/?h=stadium_01

Liite 2. Projektia varten ostettu 3D-hahmo

<https://www.cgtrader.com/3d-models/sports/game/goalkeeper-in-red-clothes>