



Ympäristösuunnittelu mobiiliajopelissä

Petri Helenius

Opinnäytetyö, AMK
Toukokuu 2023
Tieto- ja viestintäteknikka

Helenius, Petri

Ympäristösuunnittelu mobiiliajopelissä

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2023, 40 sivua

Tieto- ja viestintäteknikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Zaibatsu Interactive Oy. Työn tavoitteena oli suunnitella ja toteuttaa uuden kentän ympäristö Boom Karts mobiiliajopeliä varten. Toteutettavan ympäristön teemana oli suurkaupunki syksyllä. Ajettavan radan muoto saatiin valmiiksi toimeksiantajalta, minkä päälle ympäristöä piti lähteä toteuttamaan. Tavoitteena oli myös koostaa syntynyt ympäristö Unity-pelimoottorin sisällä. Työssä tutkittiin ympäristösuunnittelijan roolia pelimaailmassa, erilaisia työkaluja ympäristön toteuttamiseen, mobiilipeleistä yleisellä tasolla ja optimointikeinoja mobiilialustoja varten grafiikan luonnin yhteydessä.

Ympäristön ideoinnissa ja suunnittelussa käytettiin Figma- ja Photoshop-ohjelmistoa. Ympäristön 3D-mallit toteutettiin Blenderillä ja tekstuurit Substance Painter ja Designer-ohjelmistoilla. Mallinnus- ja teksturointi-prosessissa pyrittiin optimoimaan grafiikka mobiilialustoille sopiviksi.

Lopputuloksena syntyi lähes ehjä kokonaisuus Blenderin sisällä. Ympäristöön saatiin tehtyä paljon erilaisia objekteja ja tekstuureita käytettäväksi. Kiireellisen aikataulun vuoksi ympäristöä ei keretty koostaa Unityn sisällä. Toimeksiantajan tehtäväksi jäi kokeilla ja koostaa tehty ympäristö Unityn sisällä. Kaikki projektitiedot luovutettiin toimeksiantajan käyttöön.

Avainsanat (asiasanat)

3D-mallinnus, Blender, Teksturointi, Mobiilipeli

Muut tiedot (salassa pidettävät liitteet)

-

Helenius, Petri

Environment design in mobile racing game

Jyväskylä: JAMK University of Applied Sciences, May 2023, 40 pages

Degree Programme in Information and Communication Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The thesis subject was assigned by Zaibatsu Interactive Inc. The aim was to plan and execute new level's environment in Boom Karts mobile racing game. Theme of the new environment was big city in autumn. The playable track's shape was given by Zaibatsu where environment design took place. The aim was also to compose the new environment in Unity game engine. The thesis researched the role of environment artist in game development, different kind of tools to make graphics, mobile games in general and graphics optimization methods for mobile devices.

In the process of concepting the environment was used Figma and Photoshop software. The environment's 3D-models were made by using Blender and textures with Substance Painter and Designer. 3D-models and textures were made while using suitable optimization methods for mobile devices.

The results were almost complete environment assembled in Blender. The environment got lots of objects and textures to be used. The environment was originally planned to be assembled in Unity, but because of urgent time schedule, it was not possible. Assembling and testing the environment in Unity remained to Zaibatsu. All the project files were handed over to Zaibatsu.

Keywords/tags (subjects)

3D-modeling, Blender, Texturing, Mobile game

Miscellaneous (Confidential information)

-

Sisältö

1	Työn lähtökohdat	6
1.1	Taustaa ja toimeksiantaja	6
1.2	Tehtävä ja tavoitteet	6
2	Ympäristösuunnittelu peleissä	7
2.1	Yleistä pelikehityksestä	7
2.2	Mitä on ympäristösuunnittelu?	7
2.3	Tyylisuunnaukset peleissä.....	7
2.4	Värioppi ja muototeoria peleissä	9
2.5	Ympäristösuunnittelijan rooli.....	10
2.6	Ympäristösuunnittelijan työkalut.....	11
2.6.1	Ideointivaihe	11
2.6.2	Toteutusvaihe	11
2.7	Ympäristösuunnittelijan ja kenttäsuunnittelijan erot.....	13
3	3D-grafiikka	13
3.1	3D-mallintaminen yleisesti.....	13
3.2	3D-malli	13
3.3	UV-kartta	15
3.4	Materiaali, shader ja tekstuuri	17
3.5	Valaistus	19
3.6	3D-Renderöinti	21
4	Pelit mobiilialustoilla	22
4.1	Mobiilipelit yleisesti	22
4.2	Mobiilialustojen tuomat tekniset haasteet grafiikan luonnissa	22
4.3	Optimointikeinoja mobiilipeleissä	23
4.3.1	3D-mallit	23
4.3.2	Tekstuurit.....	24
4.3.3	Materiaalit ja shaderit	25
4.3.4	Valaistus.....	26
4.4	Mobiilipelit tulevaisuudessa	26
5	Case: Kentän ympäristön suunnittelu ja toteutus mobiilijopeliin	27
5.1	Lähtökohdat	27
5.2	Ideointi ja referenssien tutkiminen.....	27
5.3	3D-mallit.....	28

5.4	Tekstuurit	30
5.5	Ympäristön kokoaminen	34
6	Tulokset ja pohdinta	36
	Lähteet	38

Kuviot

Kuvio 1.	3D-mallin rakenne	14
Kuvio 2.	Topologia rakenteet	15
Kuvio 3.	Esimerkki UV-kartoituksesta Blenderin aloitusobjektilla	16
Kuvio 4.	Arkielämän esimerkki UV-kartoituksesta (UV-mapping 2017.).....	17
Kuvio 5.	Esimerkki erilaisista tekstuurityypeistä	19
Kuvio 6.	Valotyyppien havainnollistaminen Blenderissä	21
Kuvio 7.	Referenssien kokoaminen Photoshopissa	28
Kuvio 8.	Mallinnettu suihkulähde	29
Kuvio 9.	Esimerkki UV-kartoituksesta	30
Kuvio 10.	Suihkulähteen teksturointi.....	31
Kuvio 11.	Vertex Colorin asettaminen objektiin	32
Kuvio 12.	ID-maskien avulla teksturoitu objekti	32
Kuvio 13.	Designerilla toteutettu talotekstuuri	33
Kuvio 14.	Atlaksen luominen Photoshopissa	34
Kuvio 15.	Kentän kolme eri aluetta koostettuna	35
Kuvio 16.	Kaikki tehdyt objektit ja tekstuurit koostettuna yhteen.....	36

1 Työn lähtökohdat

1.1 Taustaa ja toimeksiantaja

Työn toimeksiantajana toimi Zaibatsu Interactive Oy. Zaibatsu Interactive on vuonna 2014 perustettu Keski-Suomalainen peli- ja ohjelmistotalo, mikä tarjoaa palveluita erilaisiin asiakasprojekteihin. Zaibatsulla on toimipiste Jyväskylässä ja kaiken kaikkiaan työllistää kirjoitushetkellä 27 henkilöä.

1.2 Tehtävä ja tavoitteet

Opinnäytetyön aiheeksi valikoitui ympäristösuunnittelu mobiiliajopelissä. Zaibatsun yksi merkittävimmistä yhteistyössä kehitetyistä peleistä on vuonna 2021 mobiilialustoille julkaistu Boom Karts ajopeli, mihin opinnäytetyötä lähdettiin tekemään. Boom Karts on globaalisti suosittu mobiili mo-ninpeli, mihin tarvitaan uutta sisältöä jatkuvasti. Aihe rantautui toimeksiantajan tarpeesta suunnitella ja toteuttaa uutta pelattavaa kentän ympäristöä peliä varten.

Tehtävänä oli suunnittelun jälkeen mallintaa ja teksturoida kaikki kenttään kuuluvat objektin eri työkaluja hyödyntäen ja lopuksi koostaa valmis visuaalisesti näyttävä, elävä ja hauska kenttä pelimoottori Unityn sisällä.

Opinnäytetyön tavoitteena oli tarkoitus tutkia ja vahvistaa olemassa olevaa teoriapohjaa mobiilipelien optimoinnista ja mallinnusprosessista ja käyttää näitä oppeja hyödyksi työprosessissa ympäristösuunnittelijan näkökulmasta, sekä kehittää omaa ammatillista osaamista.

2 Ympäristösuunnittelu peleissä

2.1 Yleistä pelikehityksestä

Valmiin pelin tekeminen alusta loppuun on työläs projekti riippumatta alustasta, tyylistä ja pelin koosta. Pelin tekemisessä on paljon eri vaiheita aina konseptoinnista markkinointiin asti. Jokaisella tiimin jäsenellä on oma roolinsa tietyssä kehityksen vaiheessa. Esimerkiksi keskikokoisen tiimin kesken voi olla erikseen ohjelmoijat, graafikot, ja muusikot. Näistä vielä voi joku erikoistua omaan alakategoriaan. Esimerkiksi graafikoissa voi olla 2D- ja 3D-graafikot erikseen. Oli rooli kuitenkin mikä tahansa, se on yhtä lailla tärkeä pelin valmiiksi saamisessa. Ilman ohjelmoijaa mikään ei toimi eikä mitään tapahdu. Ilman graafikoita, pelissä ei ole mitään näytettävää ja niin edelleen. Pelikehityksen vaiheista ja eri rooleista voisi kirjoittaa paljon, mutta tämä työ rajataan nimenomaan ympäristösuunnittelijan näkökulmaan.

2.2 Mitä on ympäristösuunnittelu?

Ympäristösuunnittelu (englanniksi environment design) on tärkeää jokaisessa pelissä riippumatta tyylistä. Pelin ympäristö on se mitä pelaaja näkee suurimman osan ajastaan ja on vaikutuksessa sen kanssa. Ympäristön tarkoituksena on välittää pelaajalle maailman tarinaa eri keinoin ja muodostaa yleiskuva maailman paikasta. Ympäristösuunnittelussa luodaan maailmaa, mikä tuntuisi pelaajalle mahdollisemman elävältä, sekä kaikki mahdolliset näkyvillä olevat asiat luovat jonkun tarkoituksen maailman luomisessa. Lyhyesti ilmaistuna, "miksi ja miten tuo esine on tuohon päätynyt?" (What is an environment artist 2021; Plessis n.d.)

2.3 Tyyliuuntauksiset peleissä

Ennen kuin varsinainen ympäristösuunnittelu peleissä tapahtuu, pitää olla tiedossa sen tyyli-suunta. Tyyliuuntia on monenlaisia ja niitä voidaan kokeilla yhdistellä keskenään tai luoda jotain uutta. Tällä hetkellä 3D-peleissä kuitenkin on selvästi erottuvat tyyliuunnat, mitä käytetään eniten.

Realismi

Realismissa tavoitellaan tuoda kirjaimellisesti tosielämän grafiikkaa digitaaliseen ympäristöön. Eli kaikki hahmot, ympäristöt ja objektit pitäisi näyttää melkein yhtä oikealta kuin tosielämässä. Realistisuuden tarkoitus on tuoda pelaajalla mielikuva siitä, että he itse olisivat siellä maailmassa. Monet simulaatiot ja räiskintäpelit pyrkivät tähän lopputulokseen. (3D Art Styles For Games 2022.)

Low-poly

Low-poly tyyli pyrkii rakentamaan maailmansa yksinkertaisten muotojen, värien ja valaistuksen avulla. Objektit ja hahmot tehdään yksitellen, jotta saavutetaan uniikkisuus. Low-poly suuntausta yhdistellään myös muihin tyyliin saavuttaakseen erilaisia lopputuloksia. (Mt.)

Fantasia-realismi

Fantasia-realismi tuo tosielämän aitouden johonkin asetelmaan, mikä ei olisi mahdollista tosielämässä. Esimerkiksi kaikki alienit ja fantasiaolennot. (Mt.)

Cartoony

Cartoony eli piirrosmainen tyyli on yksi suosituimpia 3D-suuntauksia, koska se mahdollistaa paljon erilaisia vaihtoehtoja luoda uniikkeja peliympäristöjä. Piirrosmaiselle tyyliin ominaispiirteitä on värikkyyys, yksinkertaisuus ja mittasuhteet ei noudata tosielämää (Mt.)

2.4 Värioppi ja muototeoria peleissä

Ympäristösuunnittelussa käytetään hyväksi paljon eri värejä ja muotoja kommunikoidakseen pelaajan kanssa ja mitä pelaaja voi maailmassa tehdä. Jotta värien ja muotojen käyttö onnistuisi hyvin pelimaailmassa, täytyy ympäristösuunnittelijalla olla käsitys väriopista ja muototeorioista

Väriopilla tarkoitetaan erilaisia sääntöjä ja ohjeita, joita käytetään hyödyksi kommunikoidakseen käyttäjän kanssa miellyttävillä värirakenteilla. Parhaiden värien valitsimiseksi käytetään hyödyksi väriympyrää, tietoa ihmispsykologiasta sekä kulttuureista. (Color Theory n.d.)

Väriympyrä jakaantuu kolmeen eri osaan: Päävärit (punainen, sininen keltainen), sekundaarivärit (päävärien sekoituksia) ja tertiäriset värit (pää- ja sekundaarivärien sekoituksia). (Color Theory n.d.). Jos väriympyrän keskellä vedetään viivat, saadaan erotettua lämpimät ja kylmät värit (Decker n.d.). Väriympyrän avulla voidaan luoda erilaisia väriharmonioita ja näitä on esimerkiksi täydennys ja analogiset värisäännöt (Decker n.d.). Täydennysvärit (englanniksi complementary color) ovat väriympyrässä olevia vastavärejä, esimerkiksi vihreä ja punainen. Analogiset (englanniksi analogous colors) värit ovat vierekkäisiä värejä väriympyrässä, esimerkiksi punainen, oranssi ja keltainen.

Väreillä ei pelkästään luoda visuaalisesti näyttävää ja miellyttävää maailmaa, vaan ne toimivat muullakin tavalla. Ihmiset yhdistävät nähdessään tiettyjä värejä erilaisiin tilanteisiin ja mitä heille siitä tulee ensimmäisenä mieleen (Color Theory in Games – An Overview 2018). Esimerkiksi peleissä värejä käyttämällä voidaan pelaajalle vihjata “pahiksen” ja “hyviksen” ryhmät. Sotapeleissä tämä nähdään osoittamalla vastustajaa ja tähtäin muuttuu tällöin punaiseksi. Mutta jos osoitetaan tiimiläistä, tähtäin muuttuu vihreäksi kertoen pelaajalle sen olevan samaa tiimiä. Eli yhdistelemällä eri värejä tähtäimen kanssa voidaan kertoa pelaajalle osoitettavan kohteen olevan vaara tai kaveri. (Mt.)

Väreillä voidaan pelimaailmassa kertoa pelaajille eri esineiden tarkoitusta ja mitä niillä voidaan tehdä. Hyvä esimerkki on kaikki pelit missä on taikuutta. Taikajuomat ovat punaisen, sinisen tai vihreän värisiä, mitkä kertovat pelaajille niiden käyttökohteet. Punaisella taikajuomalla voidaan palauttaa pelaajan “elämää” takaisin, sinisellä palauttaa taikuvaimat ja vihreällä kestävyyttä. (Mt.)

Värien lisäksi muodoilla on iso merkitys, miten pelaaja kokee ympärillä olevan maailman. Tällöin puhutaan muototeoria-tekniikoista ja miten niillä voidaan vaikuttaa pelaajan ajatusmaailmaan (Level Design Library - Shape Theory in Level Design 2019). On kolme perusmuotoa, mitkä toimivat pohjana muototeoriassa: Ympyrä, kolmio ja neliöt tai suorakulmiot. Ympyrä mielletään turvaliseksi ja ei-uhkaavaksi. Kolmio on terävä ja kuvastaa vaarallisuutta tai uhkaa. Neliöllä esitetään jonkin olevan vakaa, hyödyllinen sekä neutraali. Peleissä ympyrä muodoilla pyritään välittämään pelaajalla viesti, että nyt olet "turvahuoneessa" vaaralta. Kolmiomaisilla esteillä pelaaja ymmärtää menevänsä vaaraa kohti. Neliöt ja suorakulmaiset muodot ohjaavat pelaajaa etenemisensuunnan kannalta ja mikä alusta on vakaa edetä. Suorakulmaisuuudella voidaan kertoa pelaajalla myös vertikaalisesta eli pystysuuntaisesta etenemisen mahdollisuudesta. (Mt.)

2.5 Ympäristösuunnittelijan rooli

Pelien maailmojen luomista varten tarvitaan ympäristösuunnittelijoita, jotka ovat koko pelin yleisestä graafisesta ilmeestä vastuussa. Heiltä vaaditaan paljon osaamista sekä grafiikoiden, että tekniseltä puolelta. Ympäristösuunnittelijoilla pitää olla yleisellä tasolla hyvää taiteellista silmää, vahva ymmärrys erilaisista ympäristöistä ja mistä kaikki koostuu. Ympäristön suunnittelussa tarvitaan myös hyvää ymmärrystä väreistä, muodoista, valotuksesta ja miten nämä käyttäytyvät pelimaailmassa sekä pelaajan kanssa. Joskus ympäristön luominen voi viedä paljon aikaa, joten maltti on tärkeää. Ympäristösuunnittelijat tekevät monesti yhteistyötä konseptitaiteilijoiden kanssa, minkä pohjalta lähdetään pelimaailmaa rakentamaan. (What is an environment artist 2021.)

Ympäristösuunnittelijan täytyy myös taiteellisten taitojen lisäksi tietää erilaiset tekniset rajoitukset pelikohtaisesti maailmojen luonnissa. Tämä tarkoittaa, että pelin suorituskyky ei saa kärsiä peliasettien takia. Vaikka maailma olisi kuinka hieno, mutta jos se ei pelaajalla toimi hyvin, kärsii pelikokemus pahasti. Tässä kohtaa ympäristösuunnittelija kohtaa tärkeimpiä haasteita sen roolissa: Miten luoda näyttävää grafiikka ilman, että pelin suorituskyky ei kärsisi. (Mt.)

Ympäristösuunnittelijan on myös monesti hallittava suuri määrä erilaisia työkaluja pelimaailmojen luomiseen. Esimerkiksi erilaiset 3D-mallinnustyökalut, teksturointityökalut sekä lopullista asettelua ja jälkikäsitteilyä jonkin pelimoottorin sisällä. (Mt.) Työkalujen käytössä yhdistyy sekä taiteellinen että tekninen osaaminen.

2.6 Ympäristösuunnittelijan työkalut

Ympäristötaiteilijan pitää monesti hallita monien erilaisten työkalujen käytön. Työkaluja on paljon erilaisia tarjolla nykypäivän markkinoilla moniin eri käyttötarkoituksiin. Kaikkia maailman työkaluja ei voi lähteä esittelemään, joten tämän opinnäytetyön kannalta on valittu keskeisimmät työkalut, mitä työssä tarvitaan ja tullaan käyttämään. Työtavat ja menetelmät ovat aina yksilöllisiä ja yhtä oikeaa tapaa suunnitella tai toteuttaa ei ole olemassa.

2.6.1 Ideointivaihe

Ympäristösuunnittelu alkaa monesti erilaisten referenssien etsimisellä ja kasaamalla näitä johonkin. Referenssien tarkoitus on antaa projektin yleinen ilme, mitä halutaan lähteä jäljittelemään. Tällaisten "moodboardien" tekeminen onnistuu helposti esimerkiksi Figmalla. Figma tunnetaan käyttöliittymien suunnittelutyökaluna, mutta se toimii myös erittäin hyvin moodboardien tekoon. Siinä on helppo siirrellä kuvia, tekstejä ja muotoja, minkä avulla ideointi onnistuu hyvin.

Ideoinnissa melkeinpä tärkeimpiä työkaluja on perinteisesti kynää ja paperia. Oli se sitten fyysisenä taikka digitaalisena. Ympäristön luonnostelu piirtämällä auttaa hahmottamaan kokonaisuutta ja helpottaa myöhemmin toteuttamista.

Ideoinnissa käytetään myös hyödyksi 3D-konseptointia eli pelkillä yksinkertaisilla muodoilla asetellaan ja kokeillaan erilaisia ympäristöskenaarioita jollakin 3D-mallintamissovelluksella. Mitään ei tässä vaiheessa konkreettisesti mallinneta tarkasti, vaan jonkinlainen käsitys saadaan ympäristöstä 3D-versiona.

2.6.2 Toteutusvaihe

Blender

Blender on ilmainen avoimen lähdekoodin kolmiulotteisen mallinnuksen ohjelmisto. Blender tukee koko 3D-grafiikan luomisprosessia. Esimerkiksi mallinnuksen, animaatiot, simulaatiot ja renderöinnin. (Blender about n.d.)

Blender on varsin suosittu ohjelmisto sen maksuttomuuden takia ja myös aktiivisen laajan yhteisön ansiosta. Laajan yhteisön takia Blenderistä löytyy paljon erilaisia opetusvideoita eri tilanteita varten.

Photoshop

Photoshop on Adoben kehittämä kuvankäsittelyohjelma. Photoshopilla voidaan editoida kuvia, suunnitella logoja, maalaila, piirrellä ja periaatteessa tehdä kaikkea taiteeseen liittyvää. Photoshopin laajan piirto ominaisuuksien takia sitä käytetään myös tekstuurien piirtämiseen sekä 3D-mallin UV-karttaan tai erillisenä kuvana, mitä sitten käytetään myöhemmin materiaalissa hyödyksi.

Substance 3D Painter

Substance Painter on nykyään Adoben alla oleva reaaliaikainen 3D-teksturointi ja maalausohjelmisto. Se mahdollistaa käyttäjän teksturoida käsin tai proseduraalisilla keinoilla. (Kumar 2020, luku 6.) Painterissa voidaan maalata suoraan 3D-mallin pinnalle tai sen UV-karttaan. Painteria käytetään melkein pä universaalisti pelimaailmassa asettien teksturointiin (Kumar 2020, luku 6). Se kuuluu Adoben maksullisiin tilauksiin.

Substance 3D Designer

Substance Designer on materiaalien luomistyökalu, missä käyttäjä voi itse luoda nolla-asetelmasta erilaisia materiaaleja hyödyntäen matemaattisia funktioita ja Designerin omia nodeja kuvioitten tekemiseen. (Kumar 2020, luku 6.) Designerilla voidaan luoda todella realistisia materiaaleja taikka tyylliteltyä "käsiniirrettyä". Designer mahdollistaa proseduraalisuudellaan äärettömän määrän eri variaatioita samasta materiaalista. Substance Designer kuuluu myös nykyään Adoben alle ja on maksullinen.

Unity

Unity on monialustainen (englanniksi cross-platform) Unity Technologiesin kehittämä pelimoottori. Pelimoottorilla voidaan luoda 2D- sekä 3D-pelejä. Unity ei nykyään enää rajoitu pelkästään

pelien tekemiseen, vaan sitä hyödynnetään myös erilaisiin visuaalisiin esityksiin kuten arkkitehtuuriin, lyhytelokuviin ja autoteollisuuteen. (Unity (game engine) n.d.) Unity on tiettyyn tulorajaan asti ilmainen käyttää, minkä jälkeen pitää ostaa maksullinen lisenssi. (Unity pricing n.d.)

2.7 Ympäristösuunnittelijan ja kenttäsuunnittelijan erot

Nykypäivänä alkaa ruveta jo sumenemaan raja ympäristösuunnittelijan (englanniksi environment artist) ja kenttäsuunnittelijan (englanniksi level designer) välillä (What is an environment artist 2021). Vedän kuitenkin vielä viivan näiden kahden välille selvyyden vuoksi, sillä nämä käsitteet voivat mennä helposti sekaisin.

Ympäristösuunnittelijan rooli lyhyesti kerrattuna on luoda peliin maailma, mikä on visuaalisesti näyttävä sekä välittää maailman tarinaa pelaajalle. Tässä vaiheessa on siis luotu maailma peliin, mutta siinä ei ole konkreettista pelattavaa eikä mietitty pelin etenemistä pelaajan näkökulmasta lainkaan. Tähän tarvitaan kenttäsuunnittelijaa. Kenttäsuunnittelija luo tehtyyn maailmaan haasteita kohdattavaksi pelaajalla, jotta mielenkiinto säilyisi. Esimerkiksi erilaisilla ansoilla, vihollisen asetteluilla tai satunnaisilla tapahtumilla. Toisin sanoen kenttäsuunnittelijan tarkoitus on luoda pelaajalle tilanteita pelimaailmassa. (Bycer 2018.)

3 3D-grafiikka

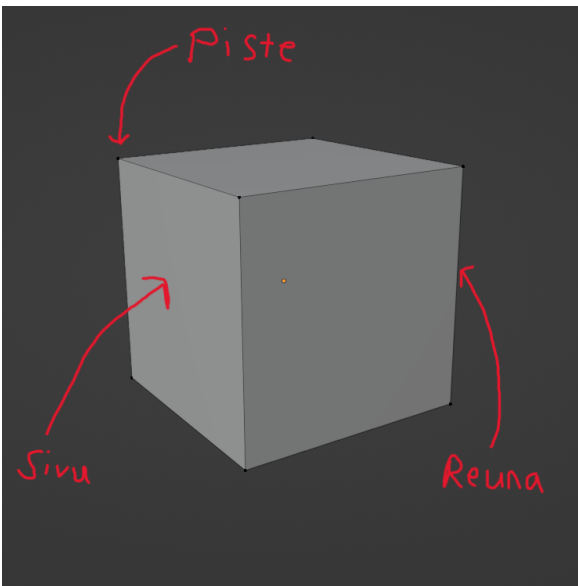
3.1 3D-mallintaminen yleisesti

3D-mallintaminen on prosessi, missä luodaan tietokoneavusteisesti kolmiulotteisia malleja. Mallintamisen aikana 3D-mallille voidaan määritellä koko, muoto ja pintakuvio. (What is 3D modelling and what is it used for 2023.)

3.2 3D-malli

Oli 3D-malli miten kompleksinen muoto tahansa tai yksinkertainen kuutio, koostuu se kolmesta perusasetelmasta: Piste (Vertex), reuna (edge) ja sivu (face). Pisteellä on sijainti 3D-tilassa eli xyz-koor-

dinaatistossa. Yksi piste on itsessään äärettömän pieni, mitä ei voi nähdä lopputuloksessa eli renderöinnissä. Kaksi pistettä yhdistettynä toisiinsa muodostaa reunan. Reuna kahdella pisteellä on myös vielä näkymätön lopputuloksessa, ennen kuin siihen liitetään kolmas piste. Kolme yhdistyvää pistettä ja reunaa muodostavat kaikista yksinkertaisimman pinnan, polygonin, minkä voi jo nähdä lopputuloksessa. Kaikki pinnat, mitkä polygonit muodostavat, kutsutaan myös sivuksi. Yhdistelemällä monia eri polygoneja toisiinsa samoilla pisteillä, muodostuu lopulta 3D-malli. (Connell 2011, luku 1.)

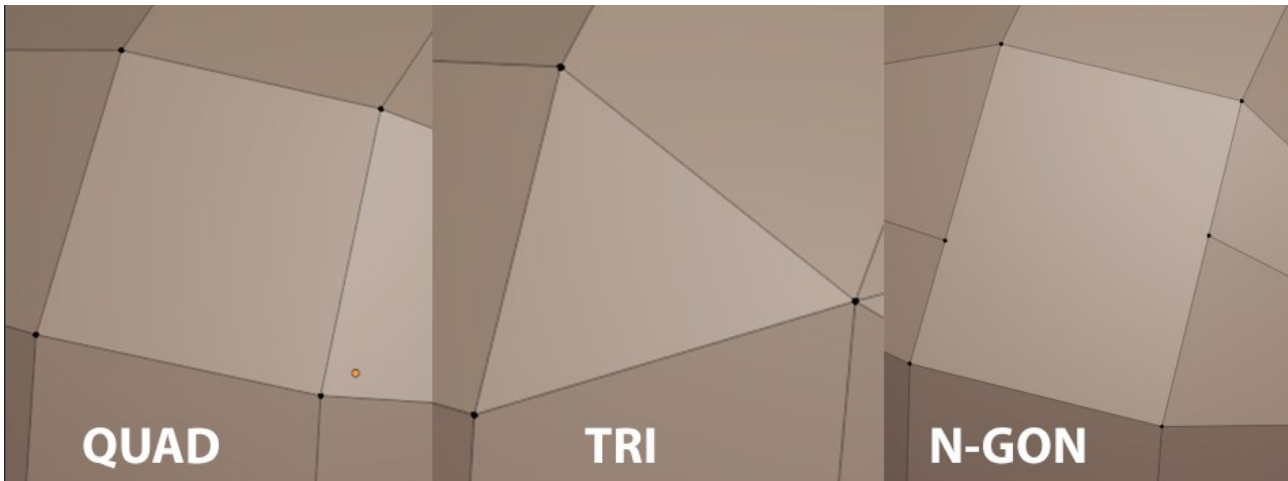


Kuvio 1. 3D-mallin rakenne

3D-mallien yhteydessä puhutaan myös termistä topologia. Tällä viitataan 3D-mallin polygonirakenteeseen. Polygoni voidaan jakaa kolmeen eri osaan: tri, quad ja n-gon. Tri eli kolmio on kolmi-reunuksinen polygoni. Quad eli suorakulmainen polygoni koostuu neljästä reunasta. N-goneja ovat kaikki muut, missä polygonin piste ja reuna määrä ylittää neljä. (Tris, Quads & N-Gons n.d.)

Lähtökohtaisesti mallinnuksessa pyritään pitämään quad-topologia, sillä tämä käyttäytyy parhaiten kaikissa sovelluksissa, mitkä renderöivät 3D-malleja. Quad-topologia on myös edellytys, jos objekti deformatoituu jollakin tavalla. Tästä hyvä esimerkki on vaikka 3D-mallinnettu hahmo, minkä käsi pitää animoida heilumaan. Tri-topologia toimii quad-topologian kanssa suurimmalta osin yhtä hyvin ja joskus 3D-mallin muoto vaatii käyttämään kolmioita. N-goneja vältellään aina kuin on mahdollista. N-gonit deformatoituu huonosti, joten niitä ei pysty animaatioissa käyttämään. N-goneilla

pystyy kuitenkin mallintamaan nopeasti objekteja ja aikaa voidaan säästää. N-gonit toimivat 3D-malleissa, jos tiedetään objektin olevan täysin staattinen ja tasainen pinnaltaan.

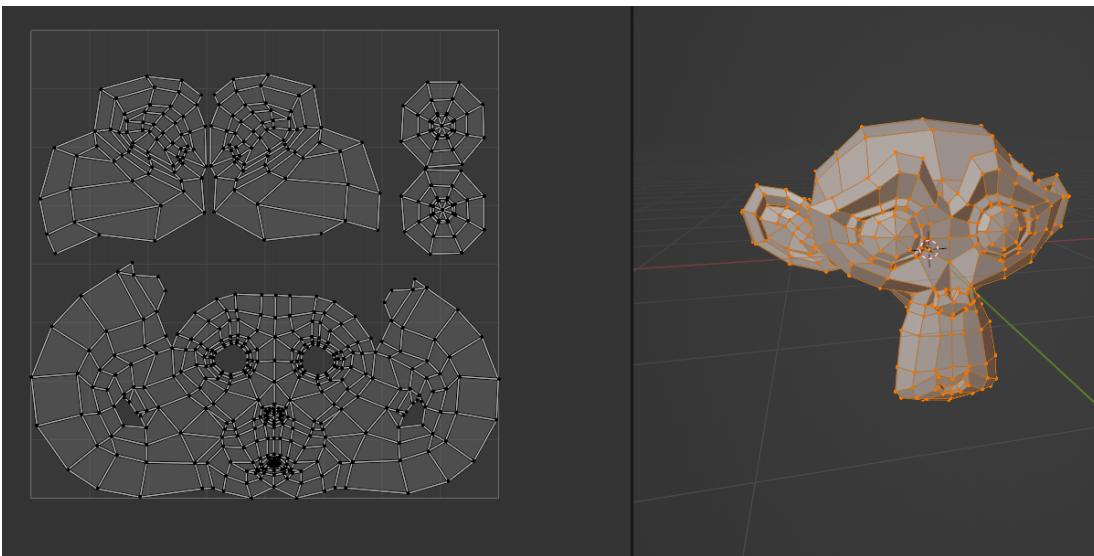


Kuvio 2. Topologia rakenteet

3.3 UV-kartta

Kun 3D-malli on koostunut polygoneista, täytyy se jatkokäyttöä varten UV-kartoittaa. UV-kartta on kaksiulotteinen esitys 3D-mallista. Kirjaimet U ja V tarkoittavat kaksiulotteista koordinaatistoa eli pysty- ja vaaka-akselia. UV-kartat mahdollistavat helpomman ja tarkemman käytön kuvapohjaisille tekstuureille. (Calvello 2022; Connell 2011, luku 3.) Tämä tarkoittaa sitä, että käyttäjällä on täysi kontrolli 3D-mallin pinnan kuvioinnista. UV-kartoitus voidaan tehdä manuaalisesti tai käyttää hyödyksi mallintamisohjelman automaattitoimintoja. Manuaalisessa tavassa käyttäjä itse UV-kartoittaa 3D-mallin polygoni kerrallaan, luoden myös järkeviä saumakohtia tarvittaessa. Automaattinen toiminto yrittää tehdä nämä asiat valmiiksi, mutta tämä toimii parhaiten yksinkertaisissa 3D-malleissa. Kompleksit mallit vaativat yleensä manuaalista tekemistä parhaan lopputuloksen saamiseksi.

UV-kartoitus on 3D-mallintamisprosessissa monesti käyttäjien mielestä ärsyttävin tai hankalin asia. Tämä johtunee siitä, koska hyvä UV-kartoitus on edellytys 3D-mallin tekstuurin oikein näkymiselle. Huonosti tehty UV-kartoitus näkyy 3D-mallin tekstuurin vääristymillä ja venymillä. Näitä vääristymiä vähentääkseen käyttäjä yrittää luoda mahdollisemman järkeviä saumakohtia, jotta UV-kartta levittyisi tasaisesti. UV-kartoituksen hankaluus on myös sidonnainen 3D-mallin pisteiden määrään. Suuri määrä pisteitä kaksiulotteisessa kartoituksessa hankaloittaa todella paljon saumakohtien tekemisestä sekä pitääkseen vääristymät vähäisinä.



Kuvio 3. Esimerkki UV-kartoituksesta Blenderin aloitusobjektilla

UV-kartoituksen ymmärtää paremmin, jos puhutaan arkielämän esineistä. UV-kartan voi ajatella sen olevan litistetty versio jostakin arkielämän esineestä. Ja se miten se litistyy, määräytyy saumakohtien mukaan. Hyvä esimerkki arkielämästä on vaikka karamellipakkaukset, missä karamellin ympärille on kääritty jokin paperi tai folio. Se sopii hyvin karamellin muotoon ja on ns. kolmiulotteinen. Mutta jos sen kääreen poistaa ja levität sen pöydälle tasaiseksi, syntyy kaksiulotteinen pinta eli UV-kartta.



Kuvio 4. Arkielämän esimerkki UV-kartoituksesta (UV-mapping 2017.)

3.4 Materiaali, shader ja tekstuuri

Jotta 3D-mallin pinta ei olisi vain pelkkää harmaata, tarvitaan materiaaleja ja tekstuureita. Materiaali sisältää yksinkertaisuudessaan tiedon valonkäyttäytymisestä sen pinnalla ja mahdollisia referenssejä tekstuureihin. Materiaalin ydinominaisuuksia on esimerkiksi väri, heijastuvuus, läpinäkyvyys ja hohtavuus. Yksinkertainen materiaali käyttää vakioarvoja näyttääkseen perusvärejä ja kiiltävyyttä, mutta jos haluaa paljon yksityiskohtaisempia pintoja, tarvitaan tekstuureita eli bittikarttoja. (Connell 2011, luku 1.)

Shader on koodia, mikä sisältää matemaattisia laskentoja ja algoritmeja laskeakseen jokaisen renderöidyn pikselin värin valaistuksen ja materiaalin arvojen mukaan. Shader renderöi tekstuurit 3D-mallin pinnalle ja laskelmoi pinnan olevan vaikka rosoinen tai kiiltävä. (Materials, Shaders & Textures n.d.)

Tekstuuri voi koostua joko kaksiulotteisista rasterikuvista tai matemaattisista funktioista, mitkä esittävät värejä lähtöarvojen mukaan. Rasterikuvat antavat käyttäjällä laajan muokattavuus mahdollisuuden, sillä käyttäjä voi itse määrittellä kaiken tarvittavan mihin kohtaan tahansa jollain tekstuurin editointityökalulla. Rasterikuvien heikkous on kuvan pikselöityminen, jos resoluutio ei ole tarpeeksi suuri. Matemaattisilla funktioilla toteutettu (tunnetaan myös nimellä proseduraalinen

teksturointi) tekstuuri on resoluutiosta vapaa, mutta muokattavuus mahdollisuudet käyttäjällä pienenevät. Matemaattisilla keinoilla voidaan luoda paljon erilaisia ja todella monimutkaisia kuvioita, mutta niiden muotoutuminen on sattumanvaraista. Tämä tarkoittaa, että käyttäjä ei voi täysin itse hallita kuvion sijaintia. (Connell 2011, luku 1.)

3D-maailmassa käytetään paljon erilaisia tekstuuri tyyppiä näyttääkseen esimerkiksi 3D-mallin pinnan värejä, kiiltävyyttä ja epätasaisuutta. Yleisemmin käytettyjä tekstuurikarttoja ovat esimerkiksi diffuse, transparency, specular, normal ja ambient occlusion.

Diffuse

Diffuse tekstuuri tyyppi on kaikista yleisin ja se esittää 3D-mallin värejä. Se voi olla kuva jostakin tai maalattu käsin. Diffuse pitää sisältää tiedon valoista ja varjoista. (Kumar 2020, luku 3.)

Transparency

Transparency tekstuuri kertoo objektin olevan läpinäkyvä tai läpinäkymätön. Se käyttää harmaasävyjä hyödyksi eli mustat osat tekstuurissa yleensä tarkoittaa 3D-mallin olevan läpinäkymätön ja valkoiset osat läpinäkyviä. (Kumar 2020, luku 3.)

Specular

Specular kertoo 3D-mallin kiiltävyydestä sen pinnalla. Valkoiset kohdat tarkoittavat kiiltävyyttä ja musta kiiltämättömyyttä. Kiiltävyys on riippuvainen valon tulosuunnasta, objektin pinnan kulumasta ja katsojan katselukulmasta. (Specular n.d.)

Normal

Normal tekstuurin avulla voidaan luoda 3D-mallin pinnalle yksityiskohtia kuten naarmuja ja kouhousia lisäämättä polygonien määrää. (Normal map (Bump mapping) n.d.)

Ambient Occlusion

Ambient occlusion tekstuurin avulla voidaan näyttää 3D-mallin alueet, mitkä valaistuvat epäsuorasta valaistuksesta ja mitkä ei. Ambient occlusion on harmaasävy kuva, missä valkoiset alueet kertovat 3D-mallin saavan valaistusta ja mustat alueet ei. (Occlusion Map n.d.)

Tekstuurityyppien havainnollistamiseksi tein oman kristallimateriaalin, mikä koostuu vasemmalta oikealle katsottuna kuvasta (ks. kuvio 5.) Diffuse, Normal ja Ambient Occlusion tekstuureista.



Kuvio 5. Esimerkki erilaisista tekstuurityypeistä

3.5 Valaistus

3D-maailman valaistus on kehittynyt koko ajan hurjaa vauhtia saavuttaakseen enemmän realistisia tuloksia. Kuitenkin 3D-maailman valaistus lähtee liikkeelle perinteisistä valotyypeistä (englanniksi traditional lights) joita on käytetty vuosia 3D-grafiikan maailmassa ja jäljittelevät tosielämän valoja perustuen matemaattisiin laskuihin valon kirkkaudessa, koosta ja suunnasta (Connell 2011, luku 4).

Directional light

Directional light on 3D-maailmojen vakio valotyyppi, mikä jäljittelee aurinkoa. Koska se jäljittelee aurinkoa ei sillä ole tiettyä pistettä 3D-maailmassa mistä se näyttää valoa. Sen sijainti on äärettömän kaukana, joten 3D-maailman valaistuksen kannalta ei ole väliä missä sijainnissa se on. Eli ei ole väliä, miten lähellä tai kaukana objektit ovat valosta, sillä valon kirkkaus ei muutu. Ainut asia millä on väliä, on sen tulokulma. Sen ominaisuuksiin kuuluu myös valon kirkkaus, varjotyyppi ja miten terävänä se näyttää varjot. (Mt.)

Point light

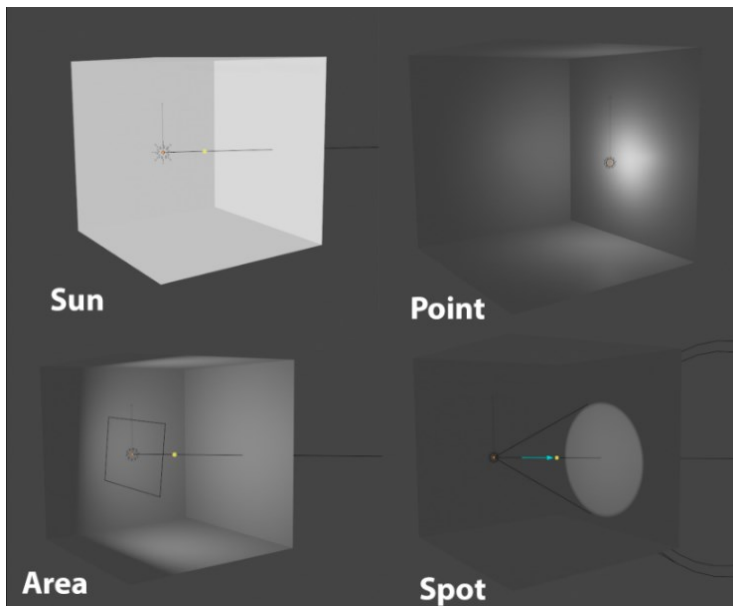
Point light on yksinkertainen valo, millä on jokin yksi sijainti 3D-maailmassa. Koska se näyttää valoa yhdestä sijainnista, sen kulmalla ei ole väliä. Tätä valotyyppiä voi verrata tosielämän hehku-lamppuun. Sen ominaisuuksiin kuuluu valon kirkkaus ja sen säteilyalue (Mt.)

Spot light

Spot light eli kohdevalo toimii kuten tosielämässäkin. Se säteilee valoa yhteen suuntaan tietystä pisteestä ja sen kulma määrittelee valaistun alueen. Valaistun alueen kokoa ja kirkkautta voidaan säädellä. (Mt.)

Area light

Area light nimensä mukaisesti valaisee määritetyltä alueelta eli se ei ole 3D-maailmassa enää vain yhdessä pisteessä. Se on suorituskyvyltään raskaampia valotyyppisiä, mutta sillä saadaan paljon tasaisempi valotus ja pehmeämmät varjot kuin muilla tyypeillä. Valaiseva alue voi olla suorakulmaisen tai ellipsin muotoinen. (Mt.)



Kuvio 6. Valotyyppien havainnollistaminen Blenderissä

3.6 3D-Renderöinti

3D-renderöinnillä tarkoitetaan prosessia, missä luodaan kaksiulotteinen kuva kolmiulotteisesta ympäristöstä. 3D-renderöintiohjelmisto ottaa kaiken datan 3D-malleihin liittyen ja kääntää sen kaksiulotteiseksi. (Guide to 3D rendering n.d.) Datalla tarkoitetaan esimerkiksi 3D-malleihin liittyviä tekstuureita ja valotusta.

3D-renderöinnissä on kahta erilaista renderöintityyppiä, missä nopeus on merkittävänä erona. Ensimmäinen on reaaliaikainen- renderöinti, mitä käytetään varsinkin peleissä, koska kuvia täytyy prosessoida todella nopeasti 3D-datasta. Toinen on offline- renderöinti, missä nopeudella ei ole väliä. Offline- renderöintiä käytetään erilaisissa animaatiofilmeissä ja pyrkiessä fotorealismiin. (Slick 2019.)

4 Pelit mobiilialustoilla

4.1 Mobiilipelit yleisesti

Mobiilipelaaminen on yksi suurimpia ja houkuttelevaisimpia teollisuuksia nykypäivänä ja sen suosio kasvaa koko ajan. Mobiilipelejä pystyy pelaamaan milloin ja missä vain ja on aina kätevästi saatavilla. Mobiilipeleillä saavutetaan myös ihmisiä, jotka eivät perinteiseen tapaan pelaa tietokone- tai konsolipelejä. Mobiilipelien valikoima on kasvanut hurjasti, sillä pelejä on tarjolla jokaisesta genrestä, esimerkiksi pulmapelistä roolipeliin saakka. (The Complete Guide to Mobile Game Development for Entrepreneurs 2022.) Vuoden 2022 tutkimustuloksissa peliteollisuuden tuotoista noin puolet oli mobiilipelaamisen puolelta (Wijam 2022).

4.2 Mobiilialustojen tuomat tekniset haasteet grafiikan luonnissa

Mobiilipelien optimointi pitää samoja keinoja sisällään kuin tietokone- ja konsolipelitkin. Molemmissa yritetään etsiä keinoja säästää resursseissa ja pitämään suorituskyky maksimissaan. Mobiilipeleissä tätä optimointia pitää viedä vaan vielä pitemmälle. Mobiililaitteita on paljon erilaisia markkinoilla ja tehokkuus on näissä erilainen. Siksi kehittäjät päättävät heti alussa kohdelaitteen, eli millä peli toimii varmasti. Esimerkiksi jos lähdetään Android-laitteelle kehittämään peliä, tämän hetkisen tilastojen mukaan suurin osa käyttäjistä käyttää Android 9.0 versioita tai suurempaa (Top Android OS versions 2023). Tästä saadaan jo hyvä käsitys kohdelaitteesta mihin lähteä kehittämään peliä.

Mobiilipelikehityksessä haasteena on luoda hyvin toimiva peli heikommille mobiilialustoille, mutta kuitenkin rajaamatta kokemusta tehokkaimmilla laitteilla. Pelin alkutekijöissä kannattaa suunnitella toimivuus heikomman laitteen mukaan. Pelin grafiikka-asetuksilla saadaan peli toimimaan sekä heikommilla että tehokkailla laitteilla. (Ghamaty 2016.) Heikommat laitteet voivat esimerkiksi laittaa "low"- asetukset päälle, mikä karsii graafista ilmettä, mutta mahdollistaa pelin toimivuuden. Tehokkaimmilla laitteilla taas voidaan käyttää vaikka "high"- asetuksia eli kaikki mahdolliset grafiikat ovat käytössä.

4.3 Optimointikeinoja mobiilipeleissä

4.3.1 3D-mallit

3D-mallien polygoni määrä mobiilipeleissä on arvokasta ja siksi niiden käytössä pitää olla järkevä. Polygoneja kannattaa käyttää niin vähän kuin on mahdollista saavuttaakseen minimin laatutason. Vähempi polygoneja tarkoittaa vähemmän pisteitä grafiikkasuorittimen prosessoinnissa ja tätä kautta suorituskyky on parempi. (Geometry Best Practices for Artists n.d a.)

Polygoneja kannattaa käyttää enemmän objekteissa, mitkä ovat lähempänä kameraa ja vähemmän kamerasta kaukana olevista objekteissa. Puhelimen näytöllä oleva pelialue on muutenkin pieni, joten kaukana oleviin objekteihin ei kannata käyttää resursseja turhaan, koska ei pelaaja niitä huomaa tai erota laadussa ollenkaan. 3D-mallin maksimi polygoni määrää ei voi määrittellä suoraan, koska se riippuu eri tekijöistä. Jos pelissä on esimerkiksi paljon samoja juoksevia hahmoja, täytyy yhden mallin polygonimäärä oltava pieni. Mutta jos pelin kohtauksessa on vain pari tärkeää objektia, voi polygonimäärä olla yllättävän suuri. Tämäkin riippuu mobiilialustasta, koska uudemmat mobiililaitteet pystyvät käsittelemään paljon enemmän polygoneja kuin vanhemmat. (Mt.)

3D-malleissa ei kannata ruveta mallintamaan pieniä yksityiskohtia polygoneilla, sillä ei nämä välttämättä näy ollenkaan pelaajille ja resursseja menee hukkaan. 3D-malleissa kannattaa keskittyä isoihin muotoihin, mitkä oikeasti muodostavat objektin rakenteen. Pienien yksityiskohtien tekemiseksi kannattaa käyttää normaalikarttoja sen sijaan kuin mallintaa. Polygoneja voidaan myös säästää jättämällä tai poistamalla tiettyjä osia objektista, mitkä näkyvät harvoin pelaajalle tai ei koskaan. Esimerkiksi autopeleissä auton pohjaan ei kannata käyttää paljoa polygoneja, sillä se harvoin näkyy pelaajalle. Tai objektit mitkä näkyvät pelaajalla vain yhdestä kuvakulmasta, voidaan poistaa niiden taka- ja alaosa kokonaan. (Mt.) Tästä hyvä esimerkki on staattiset talot pelimaailmassa, mitkä näkyvät etupuolelta pelkästään. Vaikka tällä keinolla voidaan säästää polygoneja, vähentää tämä 3D-mallien uudelleenkäytettävyyttä pelin muissa osioissa (mt).

Yksi keino optimoida pelin sisällä 3D-mallien polygonimäärää on Level of Detailin (LOD) käyttö. LOD on tekniikka vähentää objektien polygonimäärää kameran etäisyyden mukaan (Geometry

Best Practices for Artists n.d b). LODia ei kannatta käyttää, jos 3D-mallien polygonimäärä on lähitiessään pieni tai pelissä, missä kamerakulma on staattinen. Mikään ei liiku tällöin kamerasta pois päin, joten täytyy eri keinoilla vähentää polygoneja. LODia käytetään, koska objektista mikä siirtyy kauemmas kamerasta, halutaan karsia turhia polygoneja pois. Yksityiskohtia ei enää juuri erota kauas, joten kaukana kamerasta olevalla objektilla on turhaan suuri määrä polygoneja. Perussääntönä ajatellaan jokaisen LOD-tason vähentävän 50 prosenttia polygoneja edelliseen verrattuna. LODien määrää yhdelle objektilla ei voida asettaa suoraa, koska tämä riippuu objektin koosta ja sen tärkeydestä pelissä. Liian vähällinen LODin käyttö ei paranna suorituskykyä ja liian suuri polygonien vähennys LOD-tasojen välillä näkyy pelaajalle liian selvänä. Liiallinen LODien käyttö rasittaa suoritinta (CPU), koska sen pitää prosessoida tieto oikean LODin näyttämisestä. LODin käyttö kasvattaa myös pelin tiedostokokoa, koska objektista on monta eri versioita tallennettuna. LOD-tasoja voidaan luoda manuaalisesti jossain mallinnusohjelmassa, missä artisti vähentelee polygoneja jokaista tasoa varten itse. Tai sitten käytetään jotain ohjelmistoa mikä generoi automaattisesti LOD-tasot tai pelimoottorin sisällä mikä tukee tätä ominaisuutta. (Mt.)

4.3.2 Tekstuurit

Tekstuureilla on myös iso merkitys pelin optimoinnissa. Yksi keino optimoida tekstuurien puolella on tehdä "atlaksia". Tekstuuri atlas on kuva, mikä on pakattu useista pienemmistä kuvista. Tämä mahdollistaa sen, että jokaiselle objektille ei tarvitse tehdä omaa tekstuuria, vaan monet objektit voivat käyttää yhtä isompaa samaa tekstuuria. Objektit UV-kartoitetaan atlaksen mukaisesti. Atlaksien tekeminen vaatii aikaista suunnittelua. (Texture Best Practices n.d a.)

Tekstuureilla on myös objektien tavoin LOD. Tekstuureissa se tunnetaan nimellä Mipmap ja se on alkuperäisen tekstuurin kopio pienemmällä resoluutiolla (mt). Kun objekti siirtyy kauemmas kamerasta, sen pienempi resoluutioisempi tekstuuuri otetaan käyttöön ja toisin päin. Mipmappien käyttö parantaa grafiikkasuorittimen suorituskykyä, koska sen ei tarvitse renderöidä kaukana olevien objektien tekstuureita korkea resoluutioisina. (Mt.)

Tekstuurien resoluutio kannattaa pitää niin pienenä kuin on mahdollista yksittäisille objekteille, mitkä ei jaa samaa tekstuuria keskenään. Kaikkien tekstuurien ei tarvitse olla saman resoluutioisia eli vähemmän yksityiskohtia pitävän tekstuurin resoluutiota voidaan pienentää. Esimerkiksi diffuse tekstuuuri voi olla 1024 x 1024 ja specular 512 x 512. Tekstuurien resoluutio kannattaa olla myös

ihan minimissään objekteilla, mitkä näkyvät hyvin vähän pelaajille. Ei ole järkeä tehdä yksityiskoh- taista tekstuuria, jos sitä ei nähdä juuri koskaan. (Texture Best Practices n.d b.)

Hyvä UV-kartoitus mahdollistaa mahdollisimman optimaalisen käytön tekstuureille. Mobiililait- teilla resoluutio tekstuureissa ei ole kovin iso, joten UV-saarekkeet olisi saada mahdollisemman suo- riksi taatakseen parhaimman resoluution tekstuurista. (Texture Best Practices n.d c.)

Tekstuureita voidaan myös pakata keskenään ja tätä kutsutaan nimellä ”texture channel packing”. Tämän avulla voidaan pakata kolme erilaista tekstuuria yhdeksi tekstuuriksi hyödyntämällä RGB- kanavia. (Texture Best Practices n.d c.)

4.3.3 Materiaalit ja shaderit

Unity tarjoaa vakioasetuksissa valmiita shadereita eri käyttökohteita varten ja mobiililaitteille löy- tyy myös oma ”mobile shader”. Mobiililaitteille suunnattuja shadereita kannattaa käyttää aina kun on mahdollista. Ne ovat kevyempiä kuin Unityn ”standard shader”, sillä niistä on karsittu ominai- suuksia pois. Shaderin luonnin yhteydessä voidaan määritellä materiaalin käyttäytyminen valon kanssa. Tällöin puhutaan unlit ja lit shadereista. Unlit shader tarkoittaa, että se ei reagoi valon kanssa mitenkään. Tällöin suorituskyky paranee, koska kaikki varjoihin ja kiiltävyyksiin liittyvät pro- sessoinnit materiaalista jäävät kokonaan pois. Lit shader puolestaan reagoi valon kanssa luoden materiaalin pinnalle varjoja tai kiiltävyyttä. Lit shaderia käytetään useimmissa peleissä, mutta mo- biilipelejä varten kannatta harkita unlit shaderin käyttöä, jos haluaa suorituskyvyn olevan paras mahdollinen. (Material and shader best practices for artists n.d a.)

Materiaaleissa kannattaa minimoida läpinäkyvien (englanniksi transparency) materiaalien käyttöä, sillä ne heikentävät suorituskykyä paljon enemmän kuin läpinäkymättömät (englanniksi opaque) materiaalit. (Material and shader best practices for artists n.d b.)

4.3.4 Valaistus

Valaistuksen yhteydessä voidaan käyttää kolmea erilaista valaistustyyppiä ja nämä ovat baked, mixed ja real-time. Baked eli esilaskettu valaistus on kevyin vaihtoehto näistä suorituskyvylisesti. Baked valaistus on staattinen eli objektien valaistus ja varjot pelin aikana ei voi muuttua. Baked valaistus esilaskelmoi ja tallentaa kaiken valodatan objektien tekstuureihin. (Arm Guide for Unity Developers - Real-Time 3D Art Best Practices – Lighting n.d a.)

Mixed valaistus hyödyntää baked ja real-time tekniikoita. Staattisille objekteilla esilaskelmoidaan valodata tekstuureihin, mutta dynaamisella objekteilla on reaaliaikainen valaistus. Eli valo ja varjot muuttuvat liikkuvissa objekteissa. Valon voimakkuutta voidaan muuttaa pelin aikana. (Mt.)

Reaaliaikaisessa valaistuksessa valaistus ja varjot ovat kaikki dynaamisia ja valaistusta voidaan muokata pelinaikana eikä mitään esilaskelmoida. Mobiililaitteille tähdätessä ensisijaisesti kannattaa käyttää staattista valaistustyyppiä, koska reaaliaikaisissa menetelmissä valaistus laskelmoidaan ja päivitetään joka ruudunpäivityksen aikana, mikä heikentää paljon suorituskykyä mobiililaitteilla. (Mt.)

Valaistuksessa voidaan käyttää myös apuna Unityn Light Probes tekniikkaa. Niiden tarkoituksena on valaista dynaamisia objekteja, käyttämättä raskaampia reaaliaikaisia valaistustyyppijä. Light probet käyttävät hyödyksi esilaskelmoituja valaistuksia, mitkä tallentavat ympäröivän staattisen kohtauksen valaistuksen ja varjot itseensä. Tämän datan avulla dynaamisen objektin liikkeessa light proben ohi, saa se ympäröivän kohtauksen valaistuksen itseensä. (Arm Guide for Unity Developers - Real-Time 3D Art Best Practices – Lighting n.d b.)

4.4 Mobiilipelit tulevaisuudessa

Mobiilipelien kehitys kasvaa koko ajan uusien ja tehokkaampien laitteiden myötä. Uudet lippulivaalaitteet tulevat joskus olemaan keskitasonlaite, mikä taas tarkoittaa minimi kohdelaitteen paranevista. Eli yhä vaativampi peli tulee toimimaan varmemmin sen hetkisillä kohdelaitteilla.

Yksi tulevaisuudessa mobiililaitteiden rajoituksiin vaikuttava tekijä on pilvipelaaminen. Pilvipelaamisella tarkoitetaan pelien pelaamista jossakin etäpalvelimessä (What are the advantages and disadvantages of Cloud gaming 2022). Pilvipelaamisen idea on pelata uusimpia pelejä riippumatta laitteesta edellyttäen, että omistaa hyvät verkkoyhteydet ja tilauksen pilvipelaamista varten. Pilvipelaaminen on ollut markkinoilla jo usean vuoden ajan, mutta se ei ole vetänyt vertoja perinteisille laitteistoriippuvaisille peleille. Kuitenkin kehityksen mennessä eteenpäin ja verkkoyhteyksien nopeudet ovat kasvaneet, alkaa pilvipelaaminen saavuttamaan suosiotaan takaisin. (Mt.)

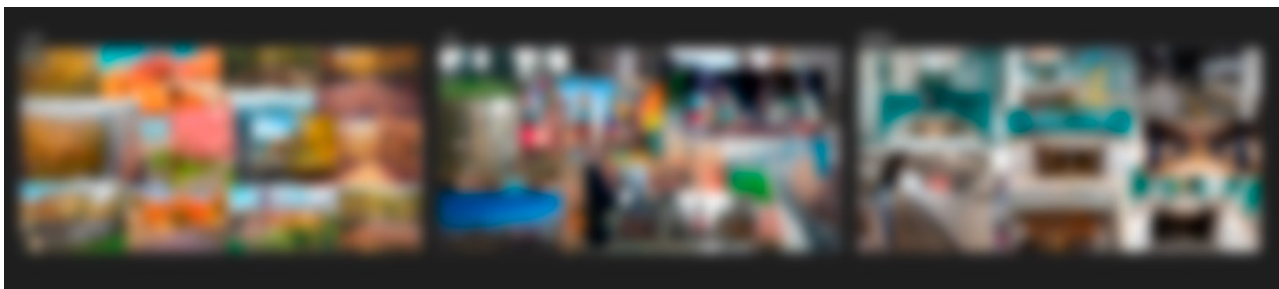
5 Case: Kentän ympäristön suunnittelu ja toteutus mobiiliajopeliin

5.1 Lähtökohdat

Työn lähtökohdana oli suunnitella ja toteuttaa ympäristöä uuteen pelattavaan kenttään. Toimeksiantajan puolelta suunniteltiin pääpiirteittäin valmiiksi radan muoto, minkä ympärille oli tarkoitus kentän ympäristö kasaantua. Kentän haluttiin olevan suurkaupunki teemainen syksyllä, mikä pitäisi sisällään kolme erilaista lohkoa, joiden läpi ajettaisiin. Radan muodon oli tarkoitus antaa suuntaa ajettavasta reitistä, mitä kuitenkin sai muutella tarvittaessa.

5.2 Ideointi ja referenssien tutkiminen

Työn tekeminen lähti käyntiin tutustumalla toimeksiantajan puolelta saatuun radan muotoon. Tämän jälkeen alkoi puhdas referenssien etsiminen tosielämän suurkaupungeista ja sen tunnelmasta. Näistä referenssikuvista ruvettiin kokoamaan ”moodboardia”, minkä tarkoituksena oli olla pohjana kentän yleisilmeelle. Figmaan ruvettiin aluksi kokoamaan kuvia, mutta päädyttiin kuitenkin lopulta kasaamaan kaikki Photoshopin puolelle, koska siinä pystyttiin samalla vapaasti piirtämään ja hahmottelemaan luonnoksia. Photoshopissa tehtiin ”moodboardi” jokaiselle kentän lohkolle, jotta tekeminen tulisi pysymään mahdollisemman selkeänä. Referenssikuvat ovat sumennettu (ks. kuvio 7.), koska niiden alkuperää ei ole selvitetty. Referenssien on tarkoitus olla vain omaa käyttöä varten ideoinnissa ja yleisilmeen hakemisessa.



Kuvio 7. Referenssien kokoaminen Photoshopissa

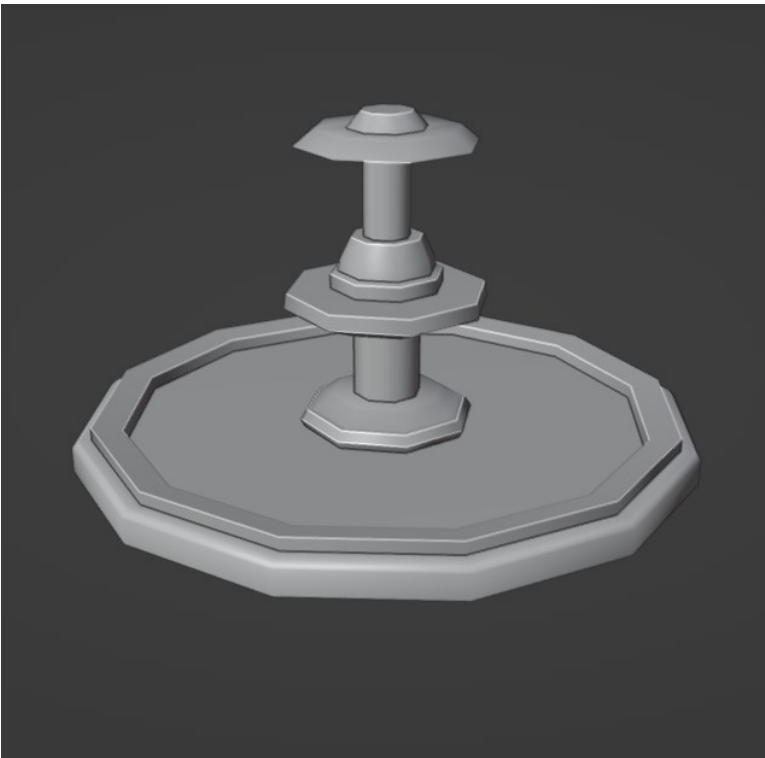
5.3 3D-mallit

Ideoinnin ja "moodboardin" tekemisen jälkeen aloitettiin objektien mallinnus Blenderiä käyttäen. Objekteja lähdettiin mallintamaan yhtä kentän lohkoaluetta kerrallaan. Eli yhden lohkon objektit mallinnettiin kokonaisuudessaan ennen kuin siirryin seuraavaan lohkoon. Pelin tyyllisuuntaus on low-poly, joten oli tärkeää, että objektit pysyivätkin sen näköisinä. Objektien polygoni määrä piti tällöin pitää minimissään ja miettiä mihin objektiin kannattaa käyttää enemmän polygoneja ja mihin ei. Lähtökohtaisesti läheltä nähtäviin ja tärkeisiin objekteihin käytettiin enemmän polygoneja, jotta niistä saisi mahdollisemman hyvälaatuisia. Jotkut objektit myös vaativat hieman enemmän polygoneja sen rakenteen vuoksi. Objektin polygonimäärään vaikutti myös se, kuinka usein sitä näkee pelin aikana. Esimerkiksi yhteen näyttävään objektiin pystyi käyttämään enemmän polygoneja kuin objektiin, mitä piti monistaa useita kappaleita. Puut täytyi mallintaa tällä periaatteella, koska niitä tulee olemaan monta kentässä, joten yhteen ei voinut käyttää paljoa polygoneja.

Objektien polygoni määrää saatiin vähennettyä poistamalla semmoisia alueita objektista pois mitä pelaajaa ei tule todennäköisesti näkemään. Tämä tarkoittaa, että suurimmasta osasta objekteista pystyi pohjassa olevat polygonit poistamaan sekä joidenkin rakennuksien takaosat. Esimerkiksi suihkulähde objektista (ks. kuvio 8.) on poistettu pohja kokonaan, koska asettelun vuoksi pelaaja ei näe sen pohjalle koskaan.

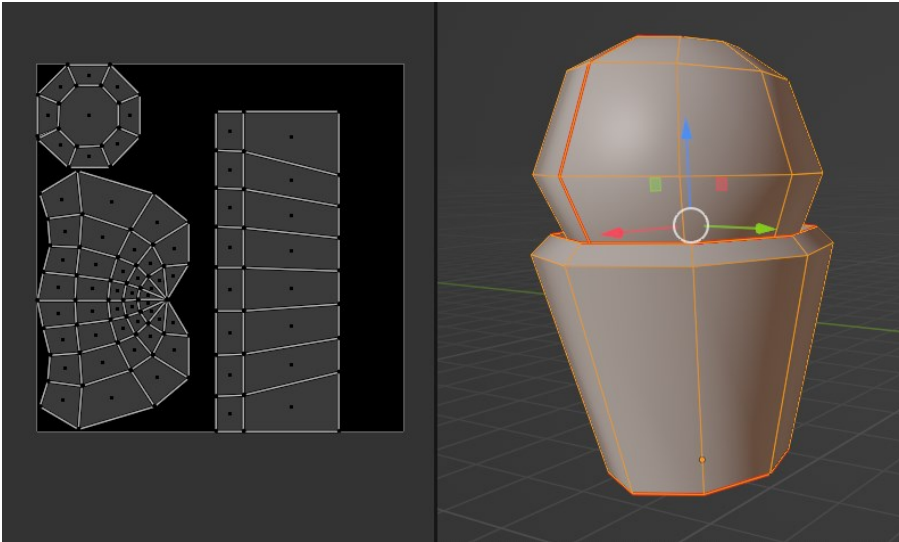
Mallinnusprosessi Blenderin sisällä oli aika suhteellisen yksinkertaista. Objektit mallinnettiin käyttämällä Blenderin vakio työkaluominaisuuksilla. Tämä tarkoittaa, että polygoneja skaalattiin, pyöritettiin ja liikuteltiin toistensa eteen saavuttaakseen halutun muodon objektille. Objektiin saatiin enemmän polygoneja käyttämällä "Loop Cut"-toimintoa, mikä lisää objektin ympärille reunan

(edge) ja yhdistää tarvittaessa risteäviin reunoihin pisteen (vertex). ”Bevel”-toimintoa käytettiin reunojen pyöristystä varten. Tätä toimintoa piti kuitenkin käyttää aika harkitusti, koska reunojen liiallinen pyöristäminen tarkoittaa myös suurta määrää polygoneja. ”Extrude”-toiminolla saatiin halutusta polygonista monistettua sitä ulospäin tai sisäänpäin luoden isoja painaumuksia tai kohoamisia. Mallintamisen aikana yritettiin myös liioitella objektien rakennetta, koska mobiililaitteelta ei näe pieniä painaumuksia ja tosi teräviä reunoja. Eli mobiililaitteille suunnatuissa objekteissa oli tärkeää muotojen korostaminen reilusti.



Kuvio 8. Mallinnettu suihkulähde

Mallintamisen jälkeen objektit piti myös UV-kartoittaa, jotta ne voitaisiin teksturoida myöhemmin. UV-kartoitus onnistui yksinkertaisissa objekteissa nopeasti käyttämällä Blenderin automaattista ”Smart UV project” UV-kartoitustoimintoa. Tämä toimi hyvin esimerkiksi kaikissa kuutiomaisissa objekteissa, missä ei ollut paljon pyöristettyjä reunoja. Objektit missä oli enemmän geometriaa ja monimutkaisempi rakenne, piti yrittää manuaalisesti merkata saumakohtia järkeviin kohtiin. Saumakohtien tarkoitus on taata mahdollisemman tasainen ja venymätön UV-kartta.



Kuvio 9. Esimerkki UV-kartoituksesta

5.4 Tekstuurit

Objektien teksturoinnissa tapahtui niin sanotusti ”taikuus”, koska objektien vähällisen polygonimäärien takia ei voitu luoda kovin tarkkaa ja näyttävää lopputulosta. Teksturoinnilla oli melkein päällimmäinen rooli objektin lopputuloksessa, koska sillä saatiin objekteihin yksityiskohtia lisättyä mitä polygoneilla ei voinut toteuttaa. Teksturoinnissa käytettiin työkaluina Substance Painteria ja Designeria.

Tekstuurien resoluution pidettiin lähtökohtana 1024 x 1024 kokoisena riippuen oliko objekti miten tärkeässä asemassa ympäristössä. Jos objekti ei ollut kovin iso tai tärkeä sen lähtöresoluutio oli 512 x 512. Tekstuurien lähtöresoluutio kannatti pitää mahdollisemman isona sen takia, koska resoluutiota voidaan aina pienentää tarvittaessa helposti, mutta toisinpäin ei ole aina mahdollista. Tämä korostui varsinkin käsin maalatuissa tekstuureissa suoraan objekteihin, mutta proseduraalisilla keinoilla tämä ei ollut ongelma. Tekstuurit minkä tarkoituksena oli olla ajettavan reitin pinnalla, asetettiin niille lähtöresoluutioksi 1024 x 256, koska isompi leveyssuunta ja pienempi pystysuunta helpottivat UV-kartoitusta ajettavan tien pinnalle.

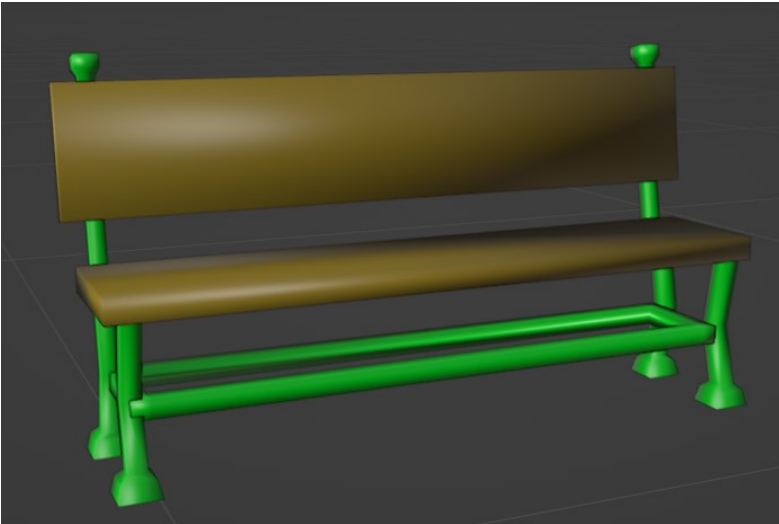
Substance Painteria käytettiin yksittäisten objektien käsin maalaamiseen. Painter sisältää suuren valikoiman erilaisia toimintoja ja siveltimiä. Siinä voi maalata käsin objektin pinnalle tai käyttää hyödyksi myös proseduraalisia ominaisuuksia. Molempia ominaisuuksia käytettiin hyödyksi, jotta

objekteista tulisi mahdollisemman hienoja. Proseduraalisilla keinoilla tehtiin pohjaväri objekteille ja loput yksityiskohdat maalattiin käsin, jotta tyyliuuntaus pysyisi oikeanlaisena.



Kuvio 10. Suihkulähteen teksturointi

Teksturointi vaiheessa saatiin aika paljon optimoitua myöhemmin käytettävien materiaalien määrää hyödyntämällä "ID-maskia". Jotkut objektit koostuivat useista muista objekteista ja näille pitäisi asettaa oma materiaali. Esimerkiksi puistopenkki koostui selkänöjasta ja jaloista. Tällöin voisi ajatella niillä olevan omat materiaalit mihin myöhemmin maalaaminen kohdistuisi. Jos objekti kuitenkin on kokonaisuudessaan yhden materiaalin alla, vaikeutuu maalaaminen siinä. Tämä tarkoittaa, että maalaaminen ei kohdistu eri alueisiin, vaan pitää yrittää yhteen samaan alueeseen maalaata kaikki. Blenderin ja Painterin avulla voidaan kuitenkin tehdä melko pitkälle maalaamista yhden materiaalin alle säästämällä myöhemmin pelimoottorissa käytettävien materiaalien määrää. "ID-maskin" periaate on määritellä objektiin Blenderin sisällä "Vertex Color" objektin eri kohtiin. Eli tässä prosessissa asetettiin pisteille (vertices) värit niihin kohtiin, minkä halutaan olevan erillinen alue maalaamisvaiheessa.



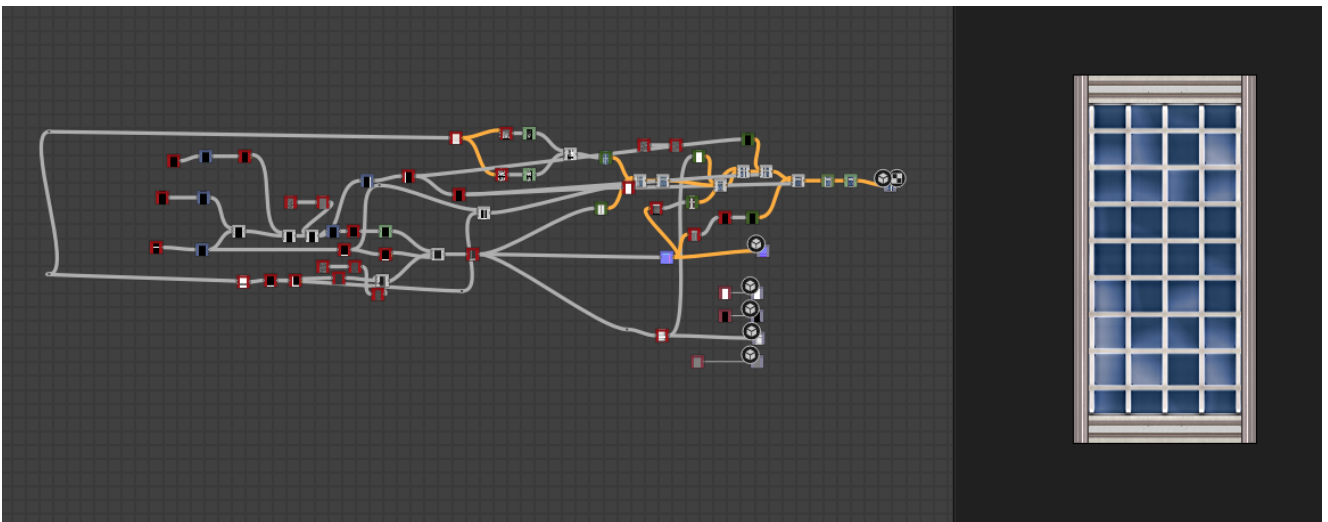
Kuvio 11. Vertex Colorin asettaminen objektiin

Kun objektiin oli määritelty alueet "Vertex Colorin" avulla, pystyi sitä hyödyntämään Painterissa. Painterin sisällä "ID-maskin" sai aika vaivattomasti käyttöön, koska se tunnistaa objektista heti määritetyt alueet, kun on painettu "Bake ID-mask"- nappia. Tämän jälkeen voidaan luoda määritetyille alueille omat kansiot ja kaikki kansion sisällä tehdyt maalaamiset kohdistuvat tähän alueeseen eikä voi niin sanotusti maalata yli. Tämä toiminto ei pelkästään auttanut optimoinnissa vaan työskentelystäkin tuli paljon tehokkaampaan, kun pystyttiin keskittymään yhteen alueeseen kerrollaan.



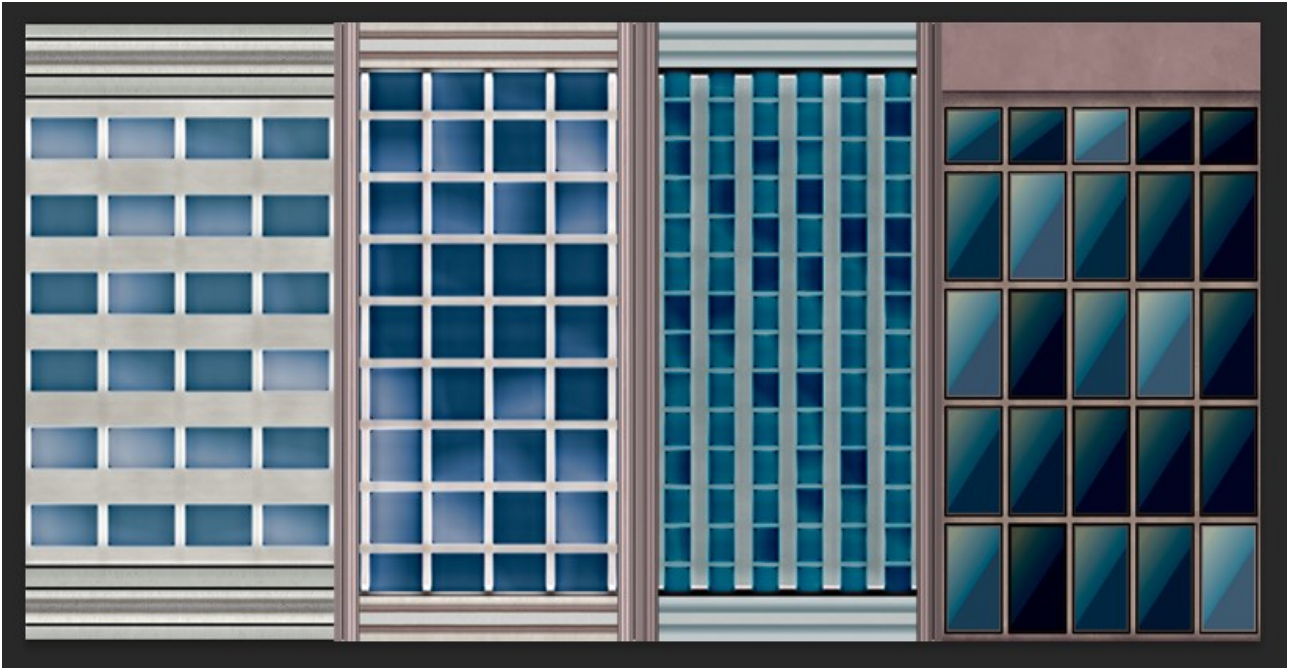
Kuvio 12. ID-maskien avulla teksturoitu objekti

Kaikki tekstuurit mitkä tulivat radan pinnalle, seinille, maahan ja kaikki mitä ei voinut maalata objektiin, tehtiin ne Substance Designerilla. Designerilla pystyi luomaan proseduraalisilla keinoilla saumattomia tekstuureita eli niitä voidaan skaalata tarvittaessa mieluiseseen ilman, että syntyy näkyviä katkoksia tekstuurin skaalautuessa. Designerissa pystyi vaihtamaan työskentelyresoluutiota milloin tahansa eli siinä onnistui nopea katsaus tekstuurin laadusta eri resoluutioilla. Designerin toimintaperiaate perustuu yhdistelemällä suuri määrä eri kuvioita toisiinsa, jotta saadaan haluttu lopputulos. Designeri valittiin tekstuurien tekoon, koska yhdestä valmiista tekstuurista saa tehtyä nopeasti eri variaatioita kuin verrattuna käsin piirtämiseen esimerkiksi Photoshopissa. Jos tekstuurit olisi maalattu käsin ja haluttaisiin vaikka jonkun kuvion muuttuvan paljon sekä samalla sen väriä vaihtaa, pitäisi ruveta kumittamaan tehtyä työtä ja maalata uudestaan. Designerissa ei ollut tätä ongelmaa, vaan kaikki tehdyt muutokset säilyttävät tekstuurin rakenteen ja syntyy uusi lopputulos. Mutta samalla tästä syntyi haasteita, koska tekstuurit piti saada näyttämään ikään kuin ne olisi käsin maalattu.



Kuvio 13. Designerilla toteutettu talotekstuuri

Tekstuureita yritettiin myös optimoida luomalla ”atlasia”. Esimerkiksi rakennuksille oli tarkoitus olla monta erilaista variaatiota ja jokaiselle yksittäiselle rakennukselle oma materiaali olisi hukattua resurssia, joten Designerissa tehtiin neljä erilaista rakennusvaihtoa ja koostettiin ne yhdeksi tekstuuriksi Photoshopin puolella (ks. kuvio 14). UV-kartoituksen avulla voidaan samasta tekstuurista käyttää neljää erilaista rakennusta.



Kuvio 14. Atlaksen luominen Photoshopissa

5.5 Ympäristön kokoaminen

Kun kaikki tarvittavat objektit ja tekstuurit oli tehty mainituilla tavoilla ja tekniikoilla, oli aika ruveta kokomaan ympäristöä yhdeksi kokonaisuudeksi Blenderin sisällä. Objektit aseteltiin monistamalla, skaalailamalla ja pyörittämällä eri suuntaan, jotta saisi variaatioita aikaan ympäristössä. Ajoreitin monistaminen tapahtui käyttämällä ”Array” ja ”Curve”-työkalua, joilla voidaan monistaa objektia käyrän muotoisesti eli se sopi hyvin ajoreitin hahmottamiseen. Objektien materiaalin luominen oli yksinkertaista, koska niihin piti asettaa vain yksi tehty tekstuuri, ”diffuse”. Asetteluuden jälkeen syntyi kolme erilaista aluetta (ks. kuvio 15.) mallinnetuista objekteista ja tekstuureista.



Kuvio 15. Kentän kolme eri aluetta koostettuna

6 Tulokset ja pohdinta

Lopputuloksena syntyi paljon erilaisia objekteja ja tekstuureita aluekohtaisesti. Objekteista syntyi yksi iso kokonaisuus Blenderin sisällä. Syntynyt ympäristö oli oma näkemys siitä, minkälainen kentän yleisilme voisi olla. Lopullinen ympäristön asettelu olisi pitänyt tapahtua Unity pelimoottorin sisällä, mutta erittäin rajallisen aikataulun takia tämä vaihe jäi kokonaan pois. Ympäristö kuitenkin valmistui Blenderin sisällä, jotta toimeksiantajalle tulisi näkemys kentän visuaalisesta ilmeestä.



Kuvio 16. Kaikki tehdyt objektit ja tekstuurit koostettuna yhteen

Toimeksiantajan mielestä lopputuloksena syntyi lähes ehjä kokonaisuus, mitä he voivat käyttää hyödyllisenä väliaskeleena lopullista kenttää varten. Suurin haaste oli kentän jakautuminen kolmeen eri alueeseen, koska yksikin näistä alueista olisi voinut olla oma kokonaisuus opinnäytetyötä varten. Kiireellisen aikataulun vuoksi palautekierrokset tapahtuivat lyhyellä ajalla ja tästä syystä esimerkiksi tekstuurien värimaailmaa olisi voinut tarkentaa ja parannella normaalilla aikataululla. Toimeksiantajalle jäi kokeiltavaksi ja koostettavaksi tehty kokonaisuus Unityn sisällä. Toimeksiantajalle palautettiin kaikki projektiin liittyvät tiedostot, minkä myötä he pystyivät vielä tehdä muutoksia objekteihin ja tekstuureihin.

Opinnäytetyön teoriaosuus valmistui melkein ennen käytännöntyön aloittamista, mikä mahdollisti lopussa täyden keskittymisen siihen, että ympäristö saataisiin valmiiksi. Itse olen lopputulokseen tyytyväinen, vaikka käytännöntyölle ei jäänyt paljoa aikaa hukattavaksi. Sain vahvistettua omaa

tietämystä enemmän ympäristösuunnittelijan roolista teoriassa sekä käytännössä. Käytännöntyön aikana sain paljon kokemusta mallintamisesta ja varsinkin teksturoinnista, koska sillä oli merkittävin rooli objektien näyttävyuden takia. Käytännöntyö tuki teoriaa siitä, mitä kaikkea ympäristösuunnittelijan täytyy osata sekä tekniseltä että taiteelliselta puolelta.

Lähteet

3D Art Styles For Games. 2022. Eri taidesuuntauksia peleissä. Julkaistu 20.12.2022. Viitattu 22.4.2023. <https://moonmana.com/blog/3d-art-styles-for-games/>.

Arm Guide for Unity Developers - Real-Time 3D Art Best Practices – Lighting. N.d a. Ohjeita erilaisien valotyyppeiden käytöstä Unityn sisällä. Viitattu 5.5.2023. <https://developer.arm.com/documentation/102109/0100/Light-mode>.

Arm Guide for Unity Developers - Real-Time 3D Art Best Practices – Lighting. N.d b. Ohjeita light probesien käytöstä Unityn sisällä. Viitattu 5.5.2023. <https://developer.arm.com/documentation/102109/0100/Light-probes>.

Blender about. N.d. Tietoa Blender-ohjelmistosta. Viitattu 6.5.2023. <https://www.blender.org/about/>.

Bycer, J. The Difference Between Environmental and Level Design. 2018. Ympäristö- ja kenttäsuunnittelun erot. Viitattu 1.4.2023. <https://www.gamedeveloper.com/design/the-difference-between-environmental-and-level-design>.

Calvello, M. 2022. What Is UV Mapping? How It Makes 3D Models Come to Life. UV-kartoituksen määritelmä. Viitattu 7.5.2023. <https://www.g2.com/articles/uv-mapping>.

Color Theory in Games – An Overview. 2018. Väriteoria peleissä. Julkaistu 4.6.2018. Viitattu 24.4.2023. <https://dvnc.tech/2018/06/04/color-theory-in-games-an-overview/>.

Color Theory. N.d. Väriteorian perusteita. Viitattu 24.4.2023. <https://www.interaction-design.org/literature/topics/color-theory>.

Connell, E. 2011. 3D for graphic designers. Yhdysvallat: John Wiley & Sons Inc. Viitattu 12.4.2023. <https://janet.finna.fi/>, Skillport.

Decker, J. The fundamentals of understanding color theory. N.d. Väriteorian perusteet. Viitattu 25.4.2023. <https://99designs.com/blog/tips/the-7-step-guide-to-understanding-color-theory/#4>.

Geometry Best Practices for Artists. N.d a. Ohjeita 3D-mallien polygoneihin. Viitattu 2.5.2023. <https://developer.arm.com/documentation/102496/0100/Triangles-and-Polygons-Usage>.

Geometry Best Practices for Artists. N.d b. Ohjeita Level of Detailin käytöstä. Viitattu 2.5.2023. <https://developer.arm.com/documentation/102496/0100/Level-of-Detail---LOD>.

Ghamaty, S. Optimizing Games for Low-End Devices. 2016. Optimointikeinoja mobiililaitteille peleissä. Julkaistu 7.9.2016. Viitattu 1.5.2023. <https://www.gamedeveloper.com/programming/optimizing-games-for-low-end-devices>.

Guide to 3D rendering. N.d. 3D-renderöinnin määritelmä Adoben verkkosivuilla. Viitattu 2.5.2023. <https://www.adobe.com/products/substance3d/discover/3d-rendering.html>.

Kumar, A. 2020. Beginning PBR Texturing: Learn Physically Based Rendering with Allegorithmic's Substance Painter. Yhdysvallat: Apress. Viitattu 23.4.2023. <https://janet.finna.fi/>, Skillport.

Level Design Library - Shape Theory in Level Design. 2019. Muototeoria peleissä. Youtube-video käyttäjältä Level Design Lobby. Julkaistu 1.1.2019. Viitattu 25.4.2023. <https://www.youtube.com/watch?v=wOhuXSZaLH8>.

Material and shader best practices for artists. N.d a. Ohjeita erilaisten materiaalien ja shaderien käytöstä Unityn sisällä. Viitattu 4.5.2023. <https://developer.arm.com/documentation/102576/0100/Shader-and-materials-hints-and-tips>.

Material and shader best practices for artists. N.d b. Ohjeita läpinäkyvien materiaalien käytöstä Unityn sisällä. Viitattu 4.5.2023. <https://developer.arm.com/documentation/102576/0100/Transparency-best-practices>.

Materials, Shaders & Textures. N.d. Tietoa materiaaleista, tekstuureista ja shadereista. Viitattu 9.4.2023. <https://docs.unity3d.com/540/Documentation/Manual/Shaders.html>.

Normal map (Bump mapping). N.d. Ohje normal mapin käytöstä. Viitattu 2.5.2023. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterNormalMap.html>.

Occlusion Map. N.d. Ohje occlusion mapin käytöstä. Viitattu 2.5.2023. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterOcclusionMap.html>.

Plessis, L. What Is Game Environment Design and How to Get Started?. N.d. Tietoa ympäristösuunnittelusta peleissä. Viitattu 2.4.2023. <https://www.domestika.org/en/blog/10804-what-is-game-environment-design-and-how-to-get-started>.

Slick, J. What Is 3D Rendering in the CG Pipeline?. 2019. 3D-renderöinnin määritelmä. Päivitetty 1.11.2019. Viitattu 28.4.2023. <https://www.lifewire.com/what-is-rendering-1954>.

Specular. N.d. Ohje Specular shaderin ja tekstuurin käytöstä. Viitattu 2.5.2023. <https://docs.unity3d.com/Manual/shader-NormalSpecular.html>.

Texture Best Practices. N.d a. Ohjeita tekstuuri atlaksen tekemiseen. Viitattu 3.5.2023. <https://developer.arm.com/documentation/102539/0100/Texture-Atlas--Filtering-and-Mipmap>.

Texture Best Practices. N.d b. Ohjeita tekstuurin resoluution käytöstä. Viitattu 3.5.2023. <https://developer.arm.com/documentation/102539/0100/Texture-size--color-space-and-compression>.

Texture Best Practices. N.d c. Ohjeita UV-kartoituksesta ja tekstuuriin pakkaamisesta. Viitattu 3.5.2023. <https://developer.arm.com/documentation/102539/0100/UV-Unwrap--Visual-and-Channel-Packing>.

Textures. N.d. Ohje tekstuuriin käytöstä. Viitattu 2.5.2023. <https://docs.unity3d.com/Manual/Textures.html>.

The Complete Guide to Mobile Game Development for Entrepreneurs. 2022. Ohjeita mobiilipelikehitykseen. Julkaistu 14.12.2022. Viitattu 30.4.2023. <https://techjournal.org/a-complete-guide-to-mobile-game-development/>.

Top Android OS versions. 2023. Android käyttöjärjestelmäversioiden määrä asennettuna mobiililaitteisiin. Päivitetty 3.5.2023. Viitattu 3.5.2023. <https://www.appbrain.com/stats/top-android-sdk-versions>.

Tris, Quads & N-Gons. N.d. Topologian rakenteet. Viitattu 7.4.2023. <https://resources.turbosquid.com/training/modeling/tris-quads-n-gons/>.

Unity (game engine). N.d. Tietoa Unity pelimoottorista Wikipedian verkkosivuilla. Päivitetty 7.5.2023. Viitattu 7.5.2023. [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)).

Unity pricing. N.d. Tietoa Unityn hinnoista. Viitattu 7.5.2023. <https://unity.com/pricing>.

UV-mapping. 2017. Havainnollistaminen UV-kartoituksesta. Twitter-julkaisu käyttäjältä xxivips. Julkaistu 6.12.2017. Viitattu 10.4.2023. <https://twitter.com/xxivips/status/938366852039888896>.

What are the advantages and disadvantages of Cloud gaming?. 2022. Pilvipelaamisen hyödyt ja haitat. Julkaistu toukokuussa 2022. Viitattu 2.5.2023. <https://www.kingston.com/en/blog/gaming/cloud-gaming-advantages-disadvantages>.

What is 3D modelling and what is it used for?. 2023. 3D-mallinnuksen määritelmä. Julkaistu 18.3.2023. Viitattu 5.4.2023. <https://www.futurelearn.com/info/blog/general/what-is-3d-modelling>.

What is an environment artist?. 2021. Ympäristösuunnittelijan määritelmä ja sen rooli pelikehityksessä. Julkaistu 27.10.2022. Viitattu 31.3.2023. <https://www.unrealengine.com/en-US/blog/what-is-an-environment-artist>.

Wijam, T. The Games Market in 2022: The Year in Numbers. 2022. Videopelien markkinaosuus vuonna 2022. Julkaistu 21.12.2022. Viitattu 1.5.2023. <https://newzoo.com/resources/blog/the-games-market-in-2022-the-year-in-numbers?>.