

PELIPROTOTYYPIN VISUAALINEN KEHITYS VR-LAITTEELLE

LAB-ammattikorkeakoulu
Muotoilu- ja taideinstituutti
Viestinnän koulutusohjelma
Mediasisällön suunnittelu
Medianomi AMK-tutkinto
Opinnäytetyö AMK
Kevät 2020
Jussi Ijäs

Lahden ammattikorkeakoulu
Muotoilu- ja taideinstituutti
Viestinnän koulutusohjelma
Mediasisällön suunnittelu
Medianomi ammattikorkeakoulututkinto

IJÄS JUSSI: Peliprototyypin visuaalinen kehitys VR-laitteelle

Opinnäytetyö
AMK
38 sivua
Kevät 2020

TIIVISTELMÄ

Opinnäytetyö seuraa prototyypin visuaalista rakennusprosessia VR-laitteelle tarkoitetussa pelissä. Työ oli Virtual Dawn -yritykselle tilattu toimeksianto, jossa roolini oli pääasiassa tuottaa 3D-malleja, 2D-grafiikkaa, pelisuunnittelua sekä pelinrakentamista Unity-pelinkehitysohjelmassa.

Opinnäytetyön tavoite oli tuottaa esittelykelpoinen minimituote, joka voidaan esitellä kiinnostuneille julkaisijoille jatkokehitystä varten. Pelin viimeistelyä jatketaan kohti valmista tuotetta opinnäytetyön jälkeenkin. Opinnäytetyö on rajattu vain pelin ensimmäiseen esittelydemoon.

Prototyypin taso oli riittävä ammatillisessa ympäristössä. Julkaisija kiinnostui ja pyysi yritystä jatkamaan tuotteen kehittämistä pidemmälle, jotta jatkosopimusta voitaisiin harkita.

Prototyyppi oli kokeiluversion valmistettavasta pelistä ja luonteensa vuoksi muutti muotoaan jatkuvasti. Mitään mitä opinnäytetyön aikana rakennettiin, ei todennäköisesti koskaan selviä sellaisenaan lopulliseen tuotteeseen. Osa ei selvinnyt edes ensimmäiseen demoan. Tärkeintä on, että se työ mitä tein on olemassa muodossa tai toisessa. Mikään palanen käyttöarvostaan huolimatta ei ollut turha, koska resursseja ei uhrattu liikaa niiden tekemiseen. Elementit rakennettiin kokeilemista varten ja olivat korvaamattomia ponnahduskiviä kehityspäätöksissä.

Avainsanat: peli, peliprototyyppi, pelisuunnittelu, peligrafiikka, VR, virtuaalitodellisuus

Lahti University of Applied Sciences
Institute of Design and Fine Arts
Degree Programme in Visual Communication
Media Content Design
Bachelor of Media (Bachelor of Culture and Arts)

IJÄS JUSSI: Game prototype's visual development in VR-headset

Bachelor's Thesis
38 pages
Spring 2020

ABSTRACT

Thesis tracks the process of building visuals for a VR-headset game prototype. The work was commissioned for the VR-indie company Virtual Dawn, where my main role was to create 3D-models, 2D-graphics, game design and a working Unity game engine scenario.

My thesis's purpose is to produce a minimum viable product for a selling showcase which can be shown to interested investors and publishers for follow-up development. The game's development will be continued after this thesis. My thesis is limited to the first showcase demo.

Quality of the prototype was sufficient in the professional environment. The publisher was interested in the game and has asked the company to provide further development and samples so that any contracts could be made.

Prototype was a test version of the game and by its nature was very susceptible to continuous change. Nothing created during the thesis period won't probably ever survive as it is into the final product. Some did not even get in the first demo. The most important thing is that the work exists in some form or another. No asset despite its level of use was useless because the amount of resources used were low. Assets were made to be tested and were invaluable stepping stones in the development decisions.

Keywords: game, game prototype, game design, game graphics, VR, virtual reality

SANASTO

Asset	Mikä tahansa komponentti, malli, prosessi tai arvokehys, jota voidaan hyödyntää tai käyttää uudelleen.
Blender	3D-ohjelma.
Buff	Yleisesti videopeleissä käytetty termi parannukselle tai voimakkuuden kohenemiselle.
Chunk	Tämän projektin sisällä käytettävä pelikentän osa, pala tai alue.
Deform	Mallin muodon muunteleminen animoinnilla tai koodilla.
Pikseli	Pienin mahdollinen graafinen elementti ruudulla, joista kaikki kuvat näytöllä koostuvat.
Pinta	Särmien väliin syntyvä alue.
Piste	Yksi piste 3D koordinaatistossa.
Polygoni	Toinen nimi pinnalle. Kolmen tai useamman pisteen alue.
Prefab	Unityn sisäinen tapa tallentaa ja monistaa objekti viemättä ylimääräisiä laskentatehoja tietokoneelta.
Rig	3D-mallin topologiaa liikuttava luusto.
Referenssi	Esimerkki tai malli.
Subdivide	3D-mallinnuksessa käytettävä matemaattinen tekniikka jakaa pinnat neljällä.
Synthwave	Elektronisen musiikin tyyli.
Särmä	Kahden pisteen välinen viiva.
Topologia	3D-pistejoukkojen geometrinen alue, joka kestää deformaation.
Unity	Pelin rakentamiseen käytetty pelimoottori.
UV	Tekstuureissa käytettävä koordinaatiston muuttujien yksikkö, koska X ja Y on varattu.
VR	Virtuaalitodellisuuslaite.

SISÄLLYS

1 Johdanto	7
1.1 Aihe.	7
1.2 Lähtökohdat	7
1.3 Tavoitteet	7
2 Prototyypin aloittaminen	8
2.1 Mikä on peliprototyyppi?	8
2.2 Mikä on VR-peli?	8
2.3 Konsepti	10
2.4 Projektiarvio	12
2.5 Graafinen tyyliuunta.	14
2.6 Prototyypin alkio	15
3 Blender-mallinnusohjelma	17
3.1 Päähahmon mallinnus	17
3.1.1 Mallin luuranko.	21
3.1.2 Teksturointi	22
3.2 Peliobjektit.	24
4 Unity-pelimoottori	26
4.1 Jälkikäsitteily.	26
4.2 Ympäristöt	28
4.3 Efektit	30
4.4 Käyttöliittymä.	34
4.5 Pelivalikko	36
5 Lopuksi	37
Lähteet.	38
Elektroniset lähteet	38
Kuvalliset lähteet	39

1 Johdanto

1.1 Aihe

Opinnäytetyöni on Virtual Reality -laseilla pelattava peli. Se on ottanut vaikutteita 80-luvun synthwave-musiikin graafisesta ulkomuodosta. Rakensin tätä projektia toimeksiantona tamperelaisen indie-firman Virtual Dawnin kanssa. Kirjallisessa osiossa käyn läpi fyysisen työn vaiheita ja puran vastaan tulleita ongelmia ja oivalluksia. Käyn läpi pelimaailman ja pelinkehitysterminologiaa, sekä alan käytännön tapoja ja käytänteitä.

Valitsin aiheekseni pelin, koska pelimaailma on minulle henkilökohtaisesti omien harrasteiden puolesta kaikkein tutuin ja motivoivin mediasisällön osa-alue. Olen aina halunnut lapsuudesta asti keksiä omia leikkejä ja pelejä, sekä asettaa ja muokata niiden sääntöjä. Haluan tuottaa ja jakaa viihteellisiä kokemuksia muille mielikuvituksellani. Pelit ovat mielestäni täydellinen viihteen välitysmuoto kaltaiselleni interaktiiviselle taiteilijalle, joka haluaa vaikuttaa omilla päätöksillään lopputulokseen.

Tärkein valintakriteerini työni aiheelle oli yhtenäisen kokonaisuuden rakentaminen. Yksittäisen palasen tai osan tekeminen tuntui irralliselta, ja siten vaikealta keksiä tai toteuttaa ilman kunnon kontekstia. Kokonaisuus tarvitaan myös idean välittämiseksi, joka merkitsee minulle työssäni eniten. Samaan aikaan työn piti olla sopivan pieni, jotta aikatauluissa pysyttäisiin ja opinnäytetyön vaatimuksiin päästäisiin. Kokonaista peliä en voi rakentaa leikkaamatta siitä elementtejä, joista haluamani pelikokemus rakentuisi. Siksi päätin tehdä prototyypin, jonka henkeä ja ulkomuotoa illustroin graafisesti.

1.2 Lähtökohdat

Aloitin pelien suunnittelua ja rakentamista vuonna 2014 ostettuani Wacom Intuous Pro 5 -piirtopöydän, jolla opettelin digimaalaamista. Törmäsin internetissä muihin pelien rakentamisesta kiinnostuneisiin harrastelijoihin. Asetin tulevaisuuden tavoitteeksi opetella rakentamaan pelejä. Toinen tavoitteeni oli opetella luomaan 2D- ja 3D-grafiikkaa, mitä voitaisiin käyttää ammattitason tuotannossa. Keräsin ympärilläni useita harrastelijoita ja rakensimme vapaa-ajalla peliprototyyppejä, joista nykyinen kokemukseni pelialaan perustuu.

Pääsin opiskelemaan Lahden ammattikorkeakouluun mediasisällönsuunnittelun linjalle. Käytin kaiken aikani ja energiani opiskelemalla koulun ulkopuolellakin Unitya, Blenderiä ja muita graafisia ohjelmia. Olen työskennellyt ammattilaisten kanssa muutaman kerran. Opinnäytetyöni on ensimmäinen kerta, jossa omaa alkuperäistä visiotani jalostetaan ammattitason studiossa kaupalliseen käyttöön. Harrastukseksi pelaan pelejä melkein päivittäin. Osallistun joskus peliyhteisöjen suunnittelukilpailuihin.

1.3 Tavoitteet

Opinnäytetyöni tavoite on vastata tutkimuskysymykseen, millainen prototyyppi riittää kaupalliseksi esittelytuotteeksi? Kuinka paljon työtä tarvitaan ja miten tavoitteeseen päästään vähäisilläkin resursseilla. Virtual Dawn -yrityksen kanssa yhteistyössä toteutettava peliprototyyppi on tarkoitettu kaupalliseksi tuotteeksi. Luonnollisesti tämä tarkoittaa sitä, että tavoite ei ollut rakentaa pientä osaa, vaan rajata opinnäytetyö isommasta kokonaisuudesta. Opinnäytetyöni päättymisen jälkeen tulevien kuukausien aikana peli jatkaisi kehitystään julkaistavaksi VR-peliksi.

Henkilökohtaisesti tavoitteeni oli tehdä työ, joka avaisi minulle käsitystäalani työelämästä, työllisyydestä ja alan vaatimuksista Suomessa. Halusin tehdä portfoliooni uuden lippulaivan, joka näyttäisi tulevan työelämäni suunnan.

2 Prototyypin aloittaminen

2.1 Mikä on peliprototyyppi?

Määrittelen prototyypin lainaamalla tieteen sanakirjasta seuraavasti.

Prototyyppi on varhainen malli rakennetusta objektista, joka on tarkoitettu testaamaan suunnittelua tai suunnitelmaa. Sana tulee kreikan sanasta prototypon, jolla tarkoitetaan alkeellista mallia.

Prototyyppien ei ole tarkoitus olla täydellisiä versioita suunnitellusta tuotteesta. Prototyyppissä on usein mahdotonta käyttää samoja materiaaleja, joita oikeassa tuotteessa käytetään.

*Prototyyppi voi replikoida vain yhtäkin osaa mallista. Prototyyppi voi olla objektin ulkomuoto ilman sen toiminnallista osaa tai toimiva osa ilman lopullista ulkomuotoa.
(Hackney Blackwell, Manar 2015.)*

Vaikka osaan C#-koodikielen perusteet ja Unity-moottorin sisällä hieman edistyneempääkin koodaamista, rajasin opinnäytetyöni prototyypin visuaalisuuteen aika- ja roolirajoitteiden vuoksi. Ammattialoilla peliala mukaan lukien on tehokkuuden vuoksi jaettu roolit, joihin palkataan alan erikoistujia tiettyyn tarkoitettuun tehtävään. Projektissamme roolini oli pääosin 3D-mallintaja, joten opinnäytetyöni rajaantui vain prototyypin visuaaliseen puoleen.

Opinnäytetyötäni ei saa sekoittaa visuaaliseen prototyyppiin. Työni on toimivan prototyypin visuaalinen osa. Visuaalinen prototyyppi on piirros tai mallinnus tuotteesta, joka on tarkoitettu testattavaksi tai visuaaliseksi ohjeeksi rakentamista varten (3D Prototype designer, 2015). Se ei siis ole toimiva objekti, vaan kuva tai kuvitelma. Toimivassa prototyyppissä konseptia voidaan jo testata, arvioida tai esitellä ostajille (The product design expert, 2016). Pelinkehittäjä Pietro Polsinelli kirjoittaa blogissansa peliprototyyppien jakaantuvan kahteen suureen kategoriaan:

*Prototyyppi pelikehittäjille itselleen pelimekaniikkojen testaamiseksi ja prototyyppi myyntipuheeksi, jolla suostutellaan muita julkaisemaan peli.
(Polsinelli 2018).*

Peliprojektini luokitellaan Polsinellin jälkimmäiseen kategoriaan eli peliprototyyppiin. Tarkoitus on miellyttää julkaisijaa julkaisemaan peli. Keskityin opinnäytetyössäni peliprototyypin ulkoasuun, jotta julkaisija saisi mahdollisimman kokonaisvaltaisen kuvan lopullisesta tuotteesta.

2.2 Mikä on VR-peli?

Peli itsessään on laaja käsite. Omasta mielestäni ja nettisanakirjan määritelmän mukaan peli on fyysinen tai henkinen leikki tai haaste. Siinä on säännöt ja yksi tai useampi pelaaja toisiaan vastaan. Peli on tarkoitettu viihdyttäväksi tai haastavaksi ja käytetään joskus opetustarkoituksissa.
(Merriam-Webster, 2020.)

Videopeli on peli, jota pelataan elektronisella laitteella. Nämä elektroniset laitteet, toiselta nimeltään pelialustat ovat eri peliyritysten tuottamia näyttölaitteita. Kaikilla alustoilla on oma käyttöliittymänsä. Esimerkiksi Microsoft Corporation Xbox One -konsoli on pelialusta, joka käyttää Microsoftin Windows 10 -käyttöliittymää. Videopelit ovat usein sidonnaisia yrityksen omiin pelialustoihin ja käyttöliittymiin.

Esimerkiksi vanhat Sony Entertainment PlayStation 2 -pelit, eivät ole yhteensopivia uudemman pelialustan PlayStation 4:n kanssa. Poikkeuksia on aina olemassa, kun yritykset tekevät alustariippumattomia pelejä, kuten Epic Games -peilyhtiön omistama jättihitti Fortnite. Se on pelattavissa tietokoneella, Sonyn ja Microsoftin konsoleilla (Gilbert 2018).

VR-peli on VR-laitteella pelattava opetus- tai viihdekäyttöön tarkoitettu peli tai simulaatio. VR on lyhenne sanoista Virtual Reality eli virtuaalitodellisuus. VR:n tarkoitus on simuloida tai luoda pelaajalle immersio ympärillä olevasta tilasta sulkemalla käyttäjän näkökentästä kaiken muun pois. Kahdella näytöllä saadaan katsojalle aikaan syvyysnäköefekti. VR-laite on pelialusta, mutta se ei ole konsoli. VR-laite on harvoin itsestään toimiva laite. Se on melkein aina tietokoneeseen tai pelikonsoliin kytkettävä lisälaitte samoin kuin hiiri, näppäimistö, peliohjain tai printteri. Virtuaalilasit asetetaan päähän kuulokkeiden tavoin. Laitteessa on molemmille silmille tarkoitettut pienet ruudut valoa sulkevassa kammiossa. Suuressa osaa laitteista on mukana myös kuulokkeet. Lasien kanssa voi tarpeen mukaan olla liikeohjaimet molemmille käsille, jotka seuraavat pelaajan käden ja sormen liikkeitä simulaatiossa. (Kuchera 2016).



Kuva 1. Oculus Rift CV1 Headset. (Evan Amos 2017)



Kuva 2. Oculus Touch Controllers for the Oculus Rift CV1. (Samwalton9 2017)

2.3 Konsepti

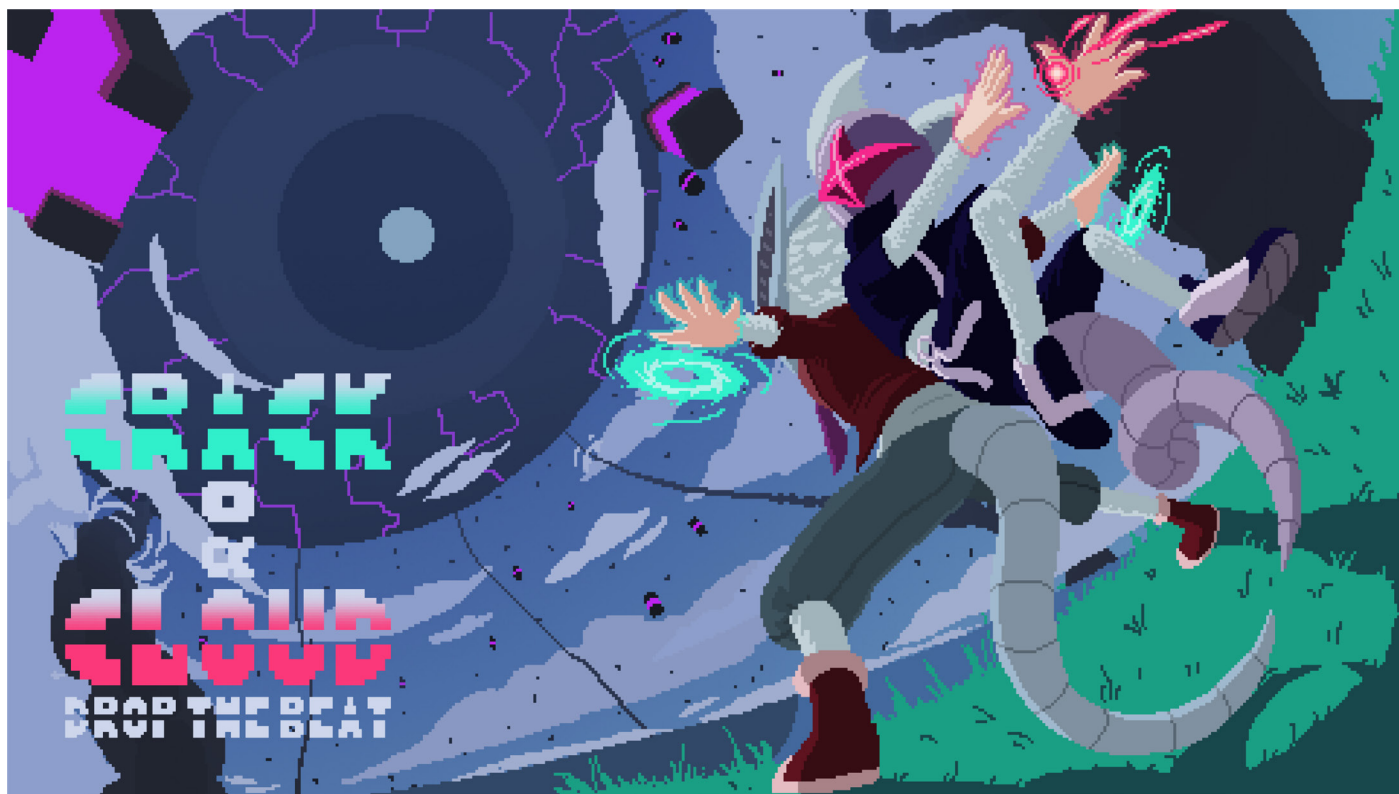
Vuonna 2016 olin Suomen Maanpuolustuskorkeakoulussa suorittamassa yhdeksänkuukautista asepalvelustani erikoisjoukoissa varusmiesgraafikkona. Töiden jälkeen inspiroiduin Daft Punk -yhtyeen soittajien kypäristä ja Get Lucky -kappaleesta. Luonnostelin kaksi rottamaista pelihahmoa samantyyllisillä kypäriillä (kuva 3). Myöhemmin viimeistelin kokonaisen mainoskuvan fiktiivisestä pelistäni (kuva 4). Siitä pari vuotta myöhemmin idea oli vieläkin tuoreena mielessäni. Halusin käyttää hahmojani ja pelikonseptiani annetuissa koulutehtävissä. Tein parhaimman versioni hahmoista toisena opiskeluvuotenani (kuva 5).

Vuonna 2020 indie-pelifirma Virtual Dawn, joka erikoistuu VR-pelien tekemiseen, palkkasi minut 3D-mallintajaksi heidän projekteihinsa. Ulkoisten tilausten loppuessa firmalle aukesi aikaa suunnitella omia projekteja, joista yksi oli VR-peli kolmannesta perspektiivistä. Kolmannessa perspektiivissä kamera on hahmon niskan takana. Molemmat olivat harvinaisia ideoita, ehkäpä ensimmäisiä alallaan. VR-pelit ja simulaatiot ovat melkein aina ensimmäisestä persoonasta. Ensimmäisessä persoonassa kamera on katsojan silmissä, koska VR-laitteiden alkuperäinen kehitystarkoitus on immersion luominen. Esimieheni halusi haastaa VR-lasien rajoja luomalla epätavallisen pelikonseptin. Ehdotin pelin ulkoasuksi vanhoja hahmojani ja työnantajani innostui teemasta.

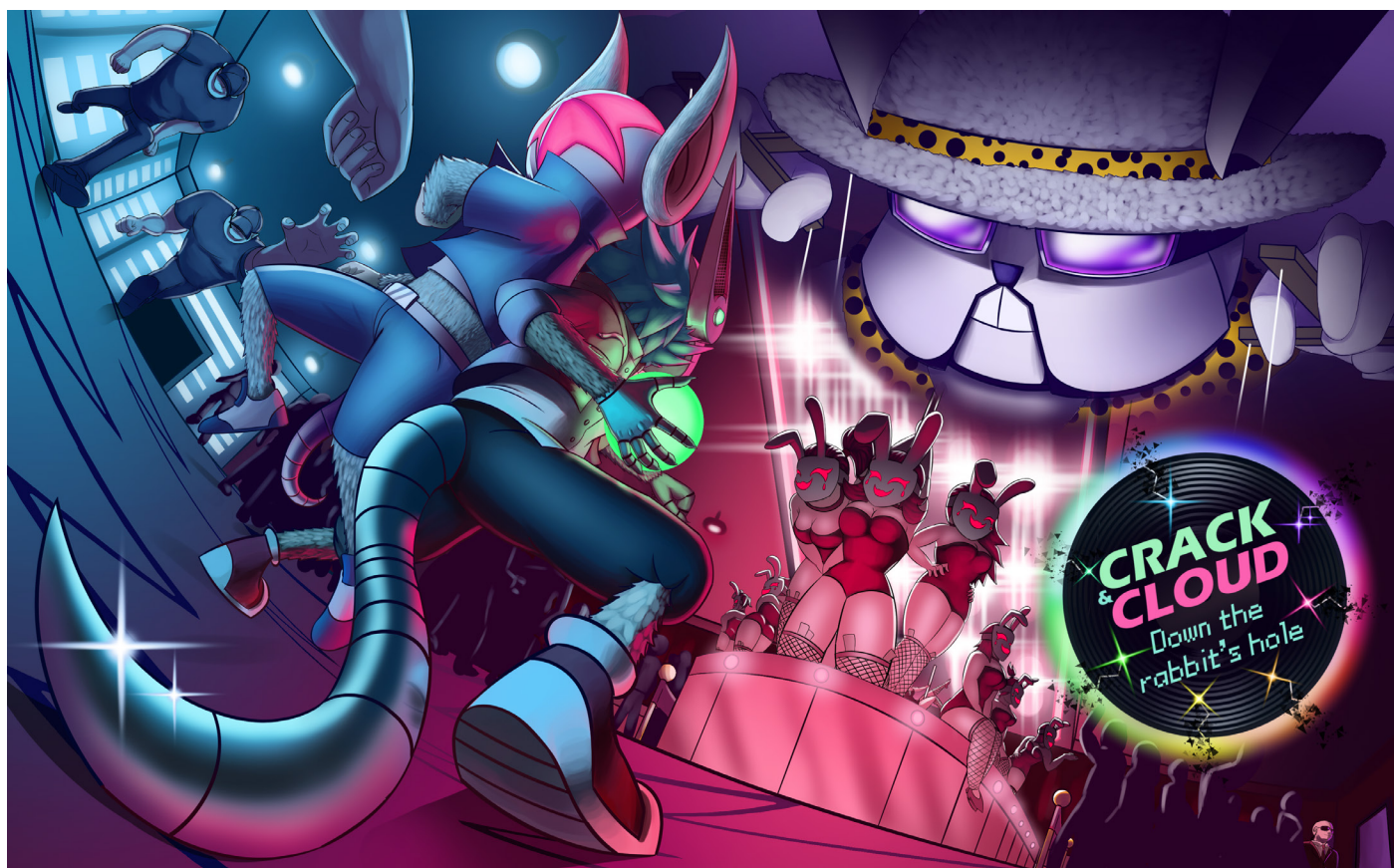
Emme aluksi tienneet projektista mitään muuta kuin hahmot, kameran kulman, teeman, ja että hahmo juoksisi ikuisesti eteenpäin kentässä. Halusimme pelin tuntuvalta nopealta ja vauhdikkaalta. Viholliset tulisivat joko kohti, ohi tai jäisivät pelaajan eteen. Ensimmäisenä päivänä lisäsimme pelimekaniikallistalle hahmon käsien liikuttamisen ohjaimilla.



Kuva 3. Crack & Cloud -konsepti. (2016)



Kuva 4. Crack & Cloud Drop the Beat. (2016)



Kuva 5. Crack & Cloud Down the rabbit's hole. (2017)

2.4 Projektiarvio

Suuremmissa yrityksissä ryhmänjohtaja tekee tilannearvion tulevan projektin koosta, budjetista, aikatauluista, tarvittavista rooleista ja muista vastaavista vaatimuksista. Esimieheni antoi projektille pari kuukautta aikaa, kunnes toisin päätettäisiin. Pienemmissä firmoissa, joissa jokainen osaa hieman kaikkea, on tehtävä henkilökohtaisesti projektiarvio omasta vastuualueestaan. Tämä on oma mielipiteeni omien kokemusteni kautta. Tapa ei ole sen kummallisempi kuin maalarin henkilökohtainen arvio siitä, minkä kokoisen paperin hän tarvitsee. Entä siveltimet ja maalipurkkien koko? Arvioon liittyy aina hahmotus omista rajoista, kyvyistä, varasuunnitelmista sekä työkaluista ongelmiseen.

Pelialalla on yleinen tapa aloittaa prototyyppien rakentaminen primitiiveistä. Primitiivi on yksinkertainen 3D-malli kuten kuutio, pallo, pyramidi tai muu yksinkertainen muoto. (Autodesk 2017). Hyvä esimerkki oikeaoppisesta peliprototyyppistä on Spider-Man 4 -pelistä otettu ruutukaappaus (kuva 6). Kaupungin rakennukset ovat pelkkiä valkoisia laatikoita ja vain hahmoon on panostettu astetta enemmän. Käytäntö on olemassa turhan työn ja hävikin välttämiseksi. Valmiit rakennukset laatikoiden sijaan eivät tuo myöskään lisäarvoa tai nopeuta perusidean testaamista. Kuvitellaan, että tuotannon alussa mallinnettaisiin, teksturoitaisiin ja animoitaisiin ympäristöksi metsä. Kuukausien päästä kenttä halutaankin jättää pois kokonaan mekaanisten tai taiteellisten päätösten takia. Tuhannet työtunnit menisivät hukkaan. Nykypäivän pelimoottoreissa on helppo vaihtaa laatikon tilalle valmis malli, joten yksinkertaisten muotojen käyttäminen on sama asia kuin maalauksen lyijykynähahmotelma.

Tee prototyyppi "roskalla", tyypillisesti laatikoilla ja kapseleilla. Ulkoasu on otettava huomioon vain, jos voit varmistaa onko prototyypin pelattavuus hauskaa vai ei. Jos se ei ole hauskaa ennen ulkoasun tekemiseen sitoutumista, pitäisi prototyyppi hylätä ja tehdä jotain muuta.
(LaneFox 2011.)

Edeltävä lainaus on Unity-pelimoottorin foorumeilta, jonne alan ammattilaiset ja harrastajat jättävät kysymyksiä tai esityksiä. Otetaan vertailuksi toinen alan kehittäjän lausunto. Danny Forest kirjoittaa nettijulkaisussaan seuraavasti:

Prototyypin rakentaminen on kaiken kaikkiaan mielessä olevan konseptin hauskuuden varmentamista.

Enemmän on vähemmän ja vähemmän on enemmän.

Tähtäisin itse aina täyden peliluupin tekemiseen.

Näytä vilaus tulevaisuudesta.
(Forest 2017.)

Jos verrataan näiden kahden kehittäjän tiivistelmää, huomataan kaksi painotettua yhtäläisyyttä. Prototyypin on pelimekaanisesti oltava hauska ja se pitää rakentaa nopeasti mahdollisimman vähällä vaivalla.

Opinnäytetyöprojektissani on otettava huomioon, että annetun kahden kuukauden aikana pitäisi tuottaa sellainen prototyyppi, jonka voi esitellä työpaikan ulkopuolisille. Pelimekaniikan hauskuus ei ole suoranaisesti vastuualuettani, koska sen onnistuminen on tiukasti sidoksissa koodaajiin. Grafiikalla voi kuitenkin vaikuttaa viihdyttävyyteen erityisesti efekteillä ja käyttöliittymällä.

Tässä vaiheessa henkilökohtainen arviointikykyne tulee tarpeeseen. Ehdinkö tehdä efektejä tai UI:ta ja jos ehdin, miten paljon? Millainen on hyvä välimuoto yksinkertaiselle, mutta myyväälle peligrafiikalle? On pakko ainakin osittain käyttää primitiivejä ja tehdä niin sanottua palikkaa. Kuitenkin pitäisi jo rakentaa esittelyvalmista tuotetta minimaalisella miehityksellä. Vastaus näihin ongelmiin on peligraafinen tyyllittely.

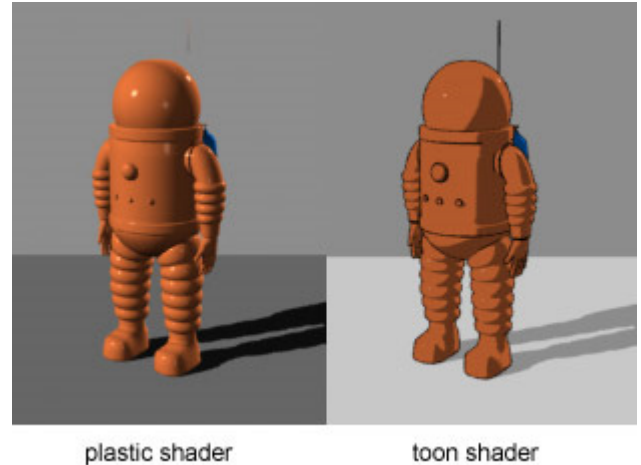


Kuva 6. Spider-Man 4, video game prototype. (togemet2 2019)

2.5 Graafinen tyyliuunta

Nykyään Microsoftin omistama Minecraft-videopeli on pelaajien tuottamista modifikaatioista huolimatta pelkkää pikselipalikkaa (kuva 7). Pikseligrafikka on 2D-maailman yksinkertaisin grafiikan tyyli. Vastaavasti kuutiot ovat 3D-maailman yksinkertaisin muoto. Minecraftin luoja Markus Persson peliä tehdessään oli alun perin ainoa kehittäjä ja teki viisaan ratkaisun käyttää pikselitekstruointia kuutiomalleissa. Hän supisti pelinsä graafisen ulkoasun yksinkertaisimpiin muotoihin ja pystyi keskittymään pelattavuuteen. Pelin grafiikan alhaisesta tasosta huolimatta, sen yhteneväisyys teki siitä omaperäisen tyylin.

Opinnäytetyössäni en käyttänyt pikselöityä kuutiotyyliä. Kyseinen tyyli ei ole pelin pelattavuuden kannalta käytännöllinen tai luo haluttua tunnelmaa. Yrityksen koon vuoksi kehitysryhmästämmme puuttui konseptitaiteilija. Ainoa vihje pelin tyyliuuntaan oli viimeisin kuvani (kuva 5). Tyylini on hyvin sarjakuvamainen. Ääriviivat ovat tummia värejä ja sisälle jäävät alueet on väritetty kirjavasti. 3D-malleille on olemassa Cel Shade -tyyliuunta, joka on luotu jäljittelemään yksinkertaista väri- ja varjostustekniikkaa (Luque, Raul 2012, 4), (kuva 8). En halunnut käyttää Cel shade-tyyliä, koska se on usein käytetty tyyli. Se ei tuo mielestäni projektiin alkuperäisyyttä tai mitään muuta kuin turhaa työtä. Pahimmillaan se leikkaisi ulkoasusta pois liukuvia sävyjä.



Kuva 8. Cel Shader.
Toiselta nimeltään toon shader. (T-tus 2005)

Mallinnuksen nopeuttamiseksi tyydyin luomaan kompromissin teksturoinnin ja mallien topologian välille. Jätin tekstuurit melkein kokonaan pois ja korvasin ne materiaaleilla. Tärkeimmille objekteille tein yksityiskohtaisemmat 3D-mallit. Ympäristöille ja nopeasti ohi meneville paloille tehtiin yksinkertaisemmat 3D-mallit. Tällainen rajaaminen ei kuitenkaan vielä riitä. Vaikka mallien tasoa alennettaisiin niin paljon kuin voidaan, tarvittavia malleja on aivan liikaa. Otimme käyttöön ilmaisia asetteja Unityn omasta verkkokaupasta. Unity Storeen voi kuka vain laittaa jakoon tai myyntiin mitä tahansa asetteja. Tähän kuuluvat lisäosat, koodit, mallit, tekstuurit ja jopa valmiit pelit.



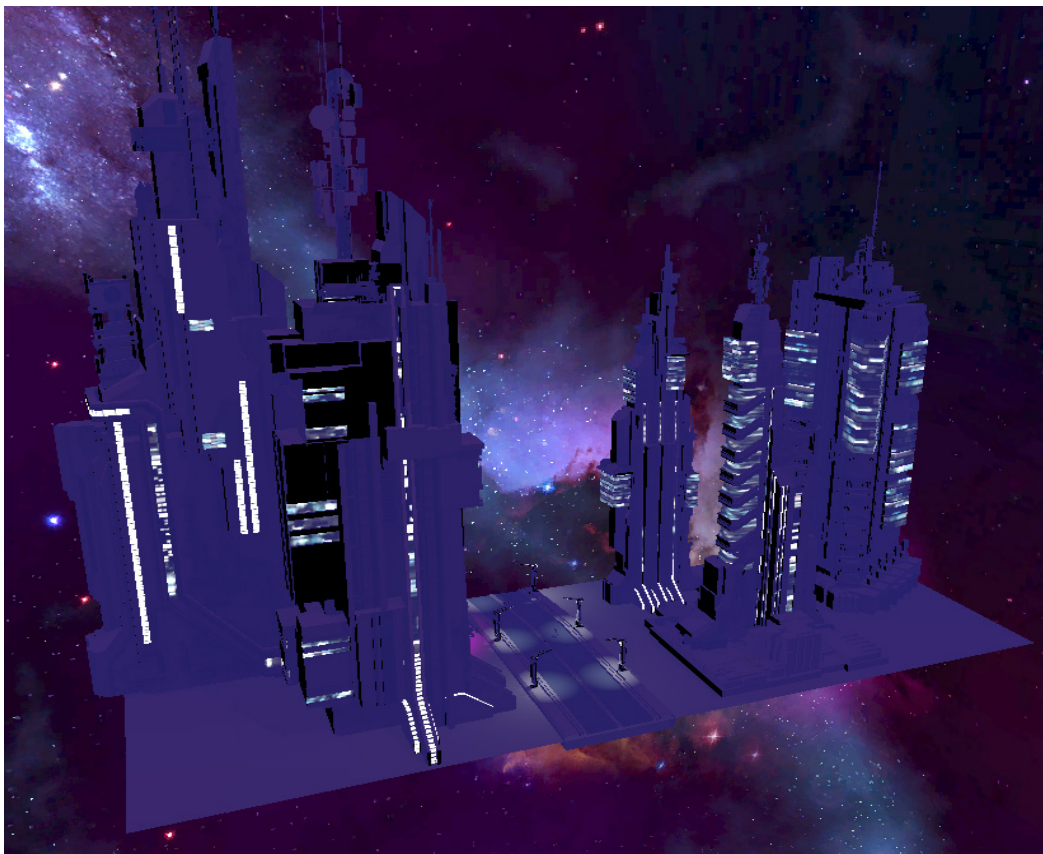
Kuva 7. Minecraft pikseligrafikkaa 3D-primitiivinelioilla. (2020)

2.6 Prototyypin alkio

Laiskat mutta toimivat ratkaisut ovat prototyypin rakentamisessa avainasemassa. Todennäköisesti kaikki palat ja osat tullaan tekemään kokonaan uudestaan tai viimeistellään uudelle tasolle. Ei ole montaa elementtiä, joiden pysyvyyttä voitaisiin taata. Eikä ole kyse siitä, etteikö prototyypissä saisi hienosäätää mitään. Olennaista on hahmottaa, missä ja milloin on syytä nähdä vaivaa enemmän lopputuloksen kannalta. Prototyypin tehtävä on luoda raamit oikealle tuotteelle ja vain testaamisen kannalta pakolliset osa-alueet viedään loppuun asti.

Projektiarvioinnin jälkeen Unity-pelimoottorissa seuraava vaihe on kerätä ja asentaa kaikki tarpeelliset assetit pelimoottoriin. Unity on kuin polkupyörä pelkällä rungolla. Kehittäjä joutuu lataamaan Unity Storesta tarvitsemansa lisäosat, jotta mikään toimisi halutulla tavalla. Vaihtoehtoisesti kehittäjä joutuu luomaan tarvitsemansa osat itse. Aloitin projektin asettien valitsemisella ja yhteensovittamisella. Kaikki lisäosat eivät sovi yhteen. Projektin alussa on tärkeää kerätä halutut osat ja karsia niistä pois osat, jotka aiheuttavat suurimpia ongelmia tai pahimmillaan rikkovat koko projektin olemassaolollaan. Projektissani törmäsin heti pitkään listaan virheilmoituksia. Ratkoin Unity-konsolin antamia virheraportteja valomoottorin toimintavirheestä muutaman tunnin. Ongelmatiedostojen korjaamisen jälkeen yhdistelin ensimmäiset Unity Storesta lataamani assetit. Ladatut väliaikaiset assetit korvataan omilla malleilla myöhemmin.

Tässä vaiheessa pelin perusideasta oli jo keskusteltu ja sovittu ryhmän kesken. Pelaajan hahmo tulee juoksemaan fyysisesti paikallaan, mutta ympäristö liikkuu hahmoa kohti luoden illuusion eteenpäin juoksemisesta. Kenttä koostuu paloista, joita kutsuimme lainasanoittain ”chunkkeiksi” (kuva 9). Peli luo tyhjästä chunkkeja, jotka olen ennalta luonut ja tallentanut pelin sisäiseen arkistoon. Koodin avulla chunkit liikkuvat samaa tahtia, pysyvät aina linjassa ja kiinni toisissaan luoden illuusion ikuisesti eteenpäin kulkevasta kadusta (kuva 10). Chunkkeihin voidaan jälkikäteen koska tahansa laittaa vihollisia ja esteitä pelaajaan haastamiseksi halutulla tavalla.



Kuva 9. Ennalta luotu katu-chunkki.



Kuva 10. Chunk-luuppi animoidulla nukella.

Idea ikuisesti kiertävästä kentästä kuuluu peligenreen nimeltä Infinite runner. Genren käyttäminen oli oma ideani, mutta Curved World -työkalun käyttäminen oli esimieheni idea. Curved World on kolmannen osapuolen tuottama ja myymä työkalu, jonka firma on ostanut käytettäväksi. Työkalu mahdollistaa suorien osien kaartamisen illuusion. Oikeasti tie tulee suoraan kohti, mutta peli näyttää kuin tie kääntyisi (kuva 10). Samaan työkaluun kuului koodi, joka mahdollisti chunkkien ilmestymisen ja lomittain kulkemisen kohti pelaajaa.

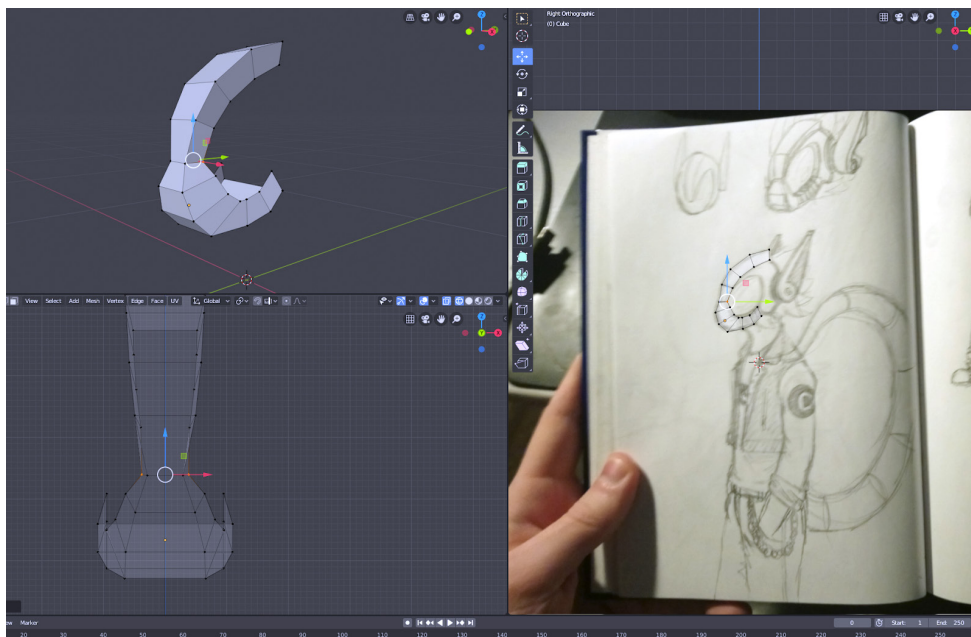
Esimieheni opetti chunkkien luomisen ja käyttämisen Curved World -koodissa. Chunkkien koko voi vaihdella, muttei kannata selkeyden vuoksi. Tässä tapauksessa 200 kertaa 60 Unity-yksikön kokoiselle pinta-alalle asetetaan halutut objektit eli katulamput, tiet ja rakennukset. Prosessi sinänsä on todella helppo, mutta vaatii tarkkuutta, jotta palojen etu- ja takaosat ovat toisiinsa nähden identtisesti kohdillaan saumattomuuden vuoksi. Tätä työtä helpotti Unityn Vertex-Snap-työkalu, joka yhdistää objektien halutut pisteet täsmällisesti.

Kokonaisuus ilmoitetaan Unitylle sen omaksi "prefab"-objektiksi, joka vedetään koodin antamaan tyhjiin sarakkeeseen. Chunkkeja voi olla useita erilaisia ja koodi arpoo chunkkien tulojärjestyksen. Rakennukset ja katujen mallit ovat myös väliaikaisesti lainattuja osia. Lainaosia käyttämällä voidaan hyvin lyhyellä ajalla kokeilla ja testata konsepteja, kuten juuri teimme. Kun perusidea näytti mukavalta, oli aika alkaa korvaamaan lainattuja osia omillamme. Ensimmäisenä jonossa oli päähahmo.

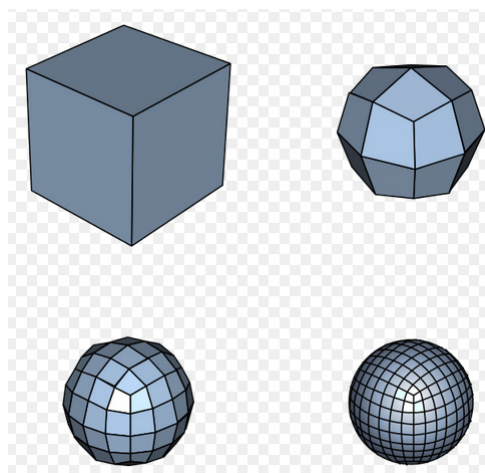
3 Blender-mallinnusohjelma

3.1 Päähahmon mallinnus

3D-mallintaminen on helpompaa referenssikuvien kanssa. Ensimmäiseksi hahmotelin päähahmon uudelleen ja aloitin mallinnuksen kypärästä (kuva 11). Pelin päähahmosa saa olla eniten yksityiskohtia, koska se on pelin näkyvillä olevin objekti ja aina huomion keskiössä. Tärkeätkin objektit täytyy mallintaa mahdollisimman yksinkertaisesti. Tarpeettomat yksityiskohdat syövät tietokoneelta tehoa turhaan. Aina ruudulla näkyvissä oleva hahmo on pysyvä taakka laskentatehoille. Päähahmon mallinnus on tasapainottelua estetiikan ja tehokkuuden välillä. Hyvä kultainen keskitie on aloittaa hahmon mallinnus yksinkertaisilla muodoilla ja antaa mallinnusohjelman laskea lisää pisteitä automaattisesti pehmentäen mallia siistimmäksi. Pehmennys tapahtuu jakamalla 3D-mallin neliö neljällä. (kuva 12). Kun käytetään muunninta pehmentämään mallia, alkuperäinen rujompi versio säilyy tallessa, jotta muutoksia voidaan tehdä helposti jälkikäteenkin.



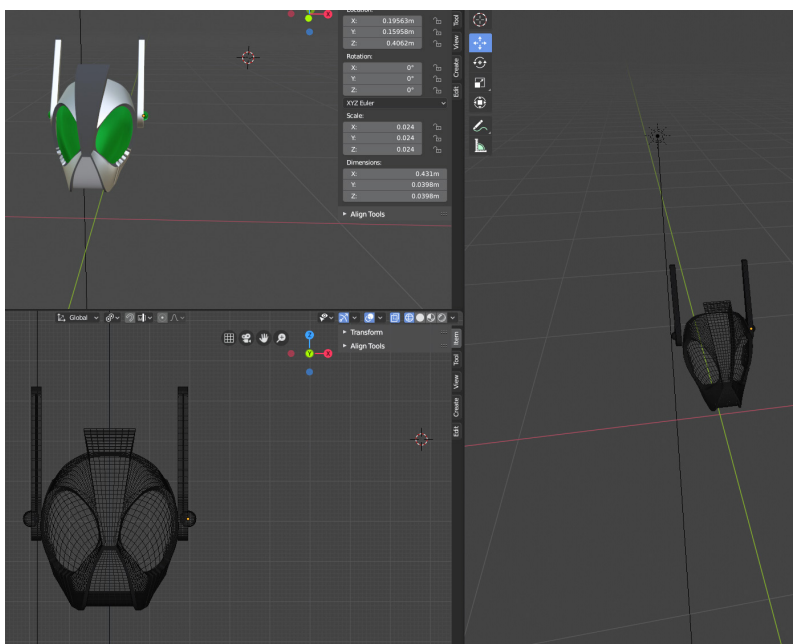
Kuva 11. Päähahmon hahmotelma.



Kuva 12. 3D-mallin pehmennys jakamalla neliö neljään "Subdivide-muunnin". (ico83 2006)

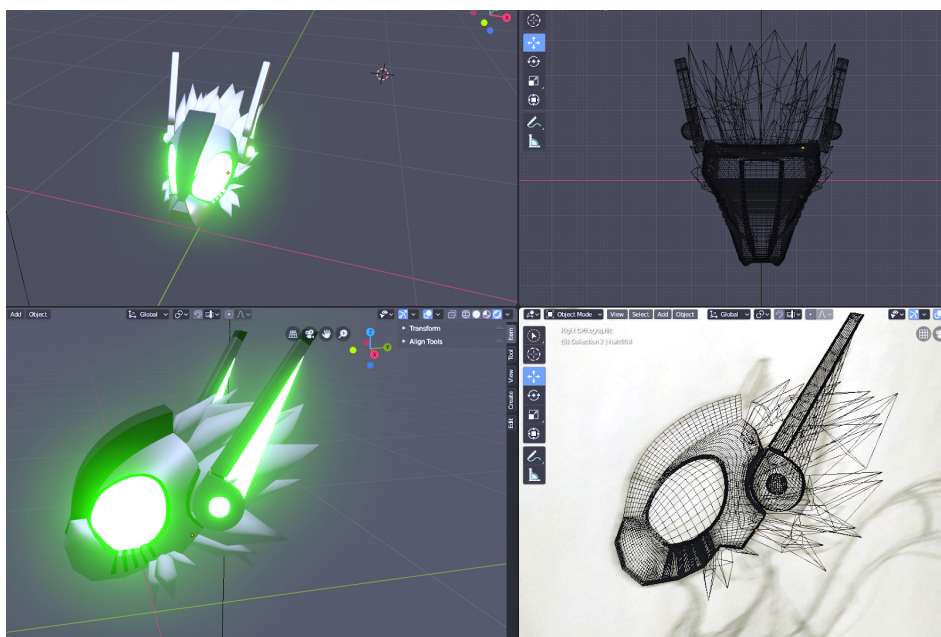
Mallintamisen nopeuttamiseksi käytin peilimuunninta. Peilimuunnin peilaa valitun koordinaatiston akselin mukaisesti 3D-mallin pisteet. Plus yhdelle asetettu piste X-akselilla ilmestyy myös miinus yhdelle, luoden täydellisen symmetrian. Valkeille paksuille hiuksille tein yksinkertaisen muodon, jonka kopioin useasti ja siirsin ne täyttämään kypärän takaosaa (kuva 14). Vaihtelevuus tuo orgaanisuutta. Muokkasin jokaisen hiuspalan kulmaa, asentoa ja jopa kärkipisteen sijaintia hieman satunnaiseen suuntaan. Blenderissä on olemassa hiusgenerointi, joka pystyy tuottamaan suhteellisen realistisen näköisiä hiuksia, mutta en halunnut käyttää sitä tyyli- ja tehokkuussyistä. Palikkamaiset hiukset sopivat sarjakuvamaiseen tyyliin paremmin ja ovat mallintamisen kannalta käytännölliset.

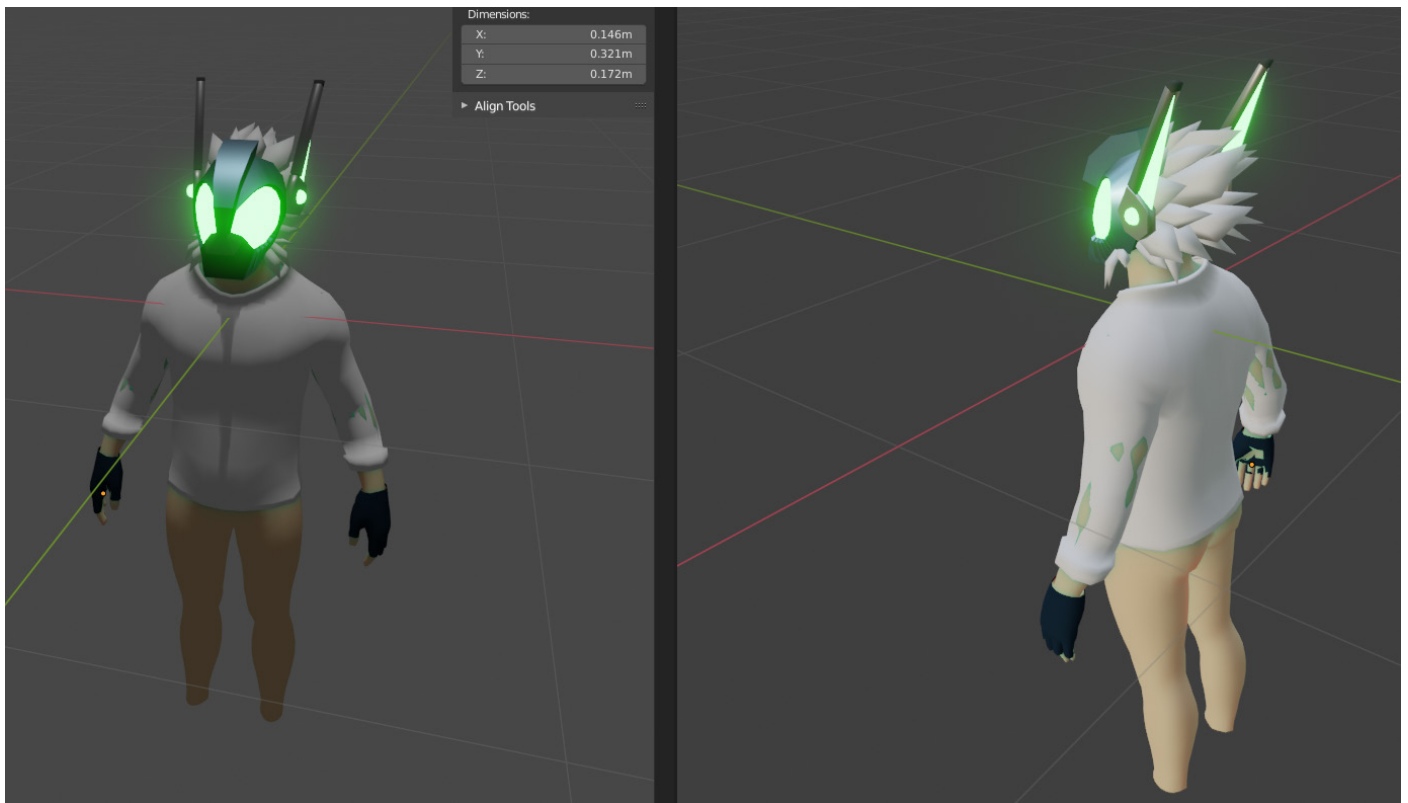
T-paidan ja hanskojen topologia on kopioitu suoraan keskikehosta. Kasvatin kopioitujen osien kokoa juuri ja juuri peittämään kehon mallia. Korjailin t-paidan leveyttä reuna- ja taivealueilta, jotta viimeisimmätkin läpipaistavat kehon osat peittyisivät luonnollisen näköisesti. Lopuksi lisäsin mallien topologiaan kaulukset, hihat ja materiaalit. (Kuva 15.) Housut ja liivit on tehty samoin kuin t-paita ja hanskat (kuva 16).



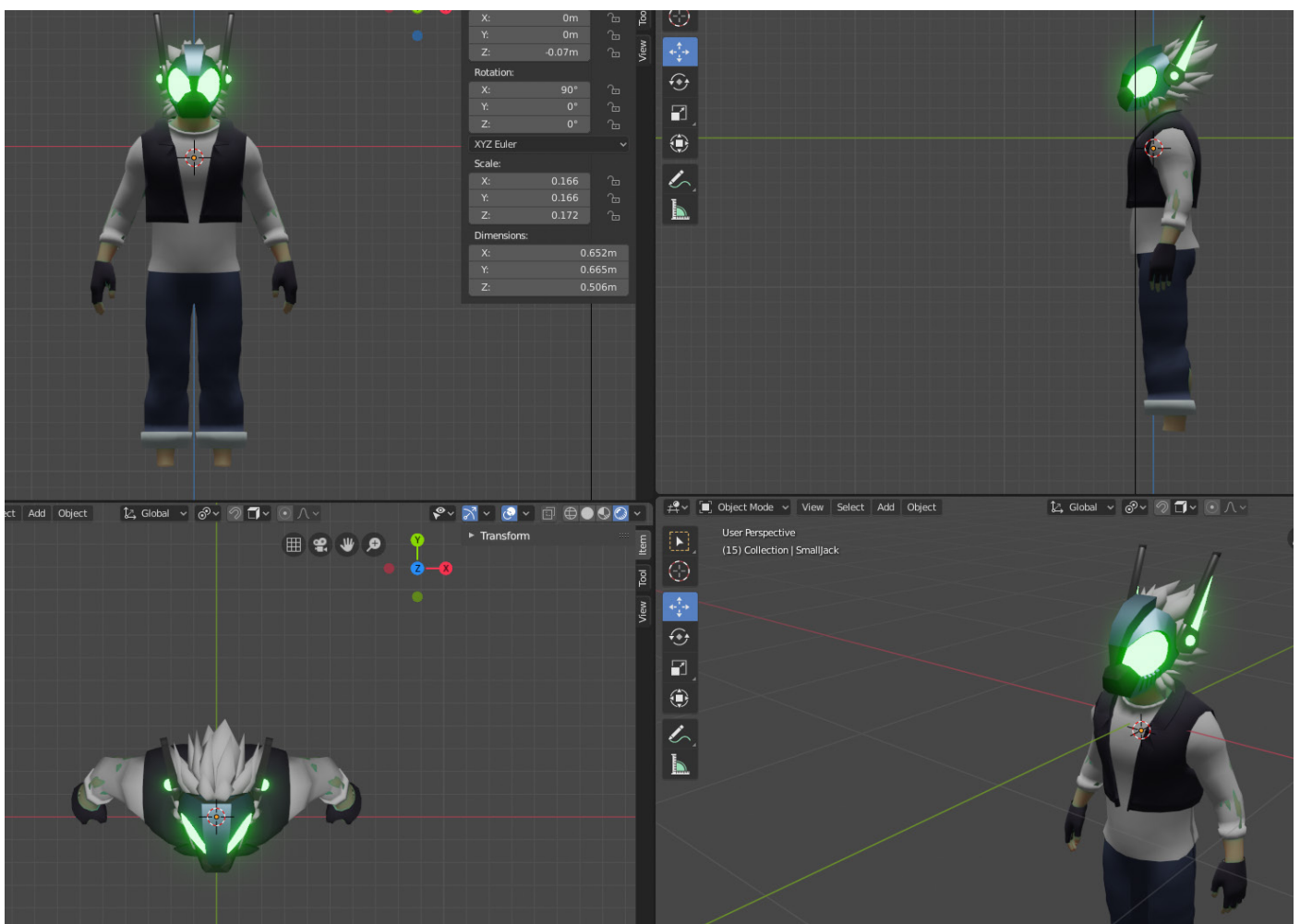
Kuva 13. Päähahmon kypärän alku. Mallissa on "Subdivide"-muunnin.

Kuva 14. Päähahmon kypärä valmiina.





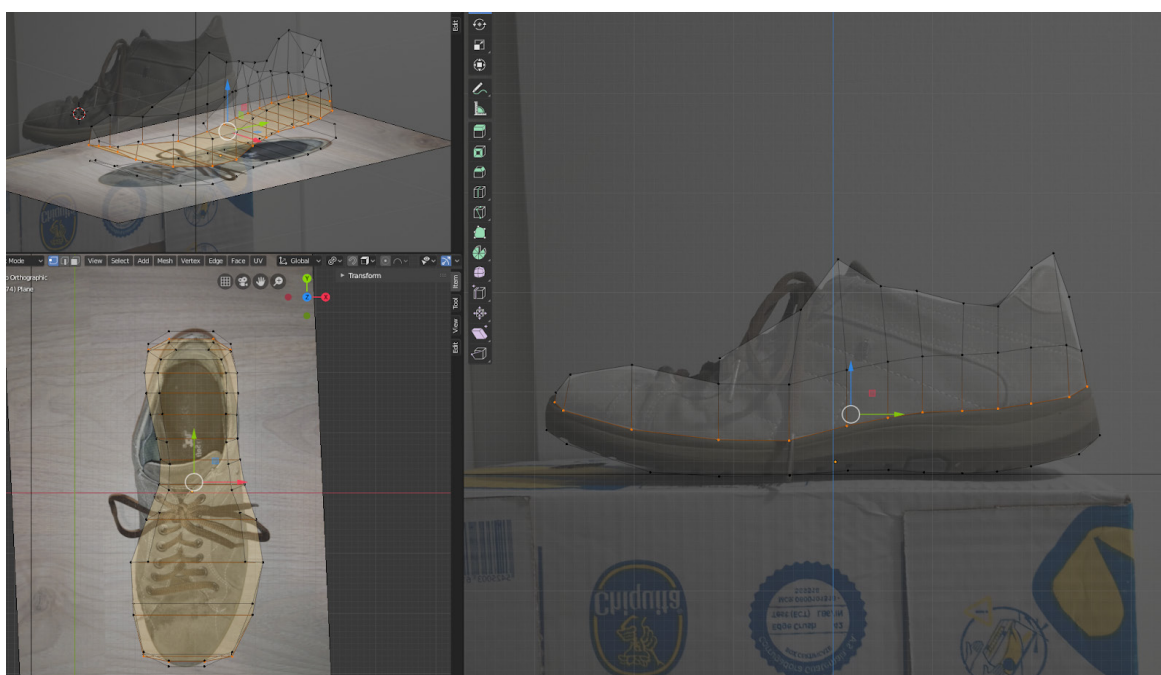
Kuva 15. Päähahmon paita ja hanskat.



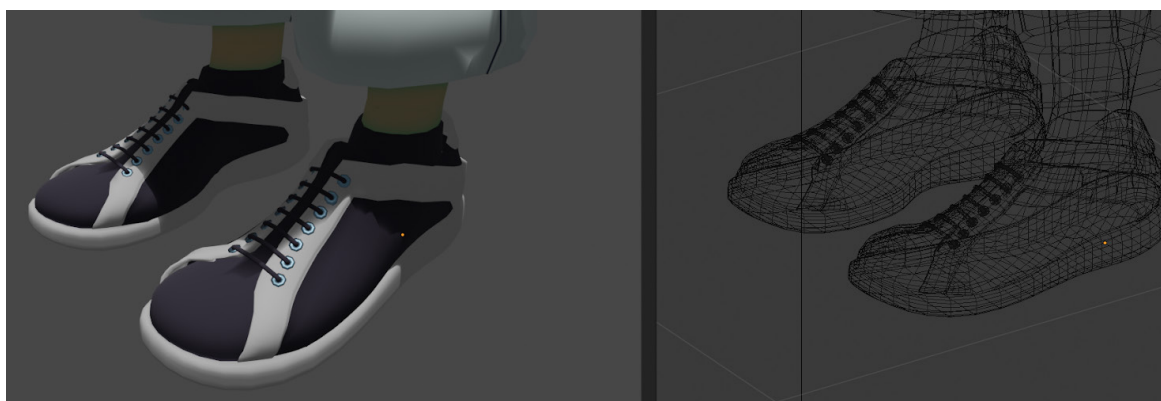
Kuva 16. Päähahmon liivi, housut ja loput materiaaleista.

Kenkien mallintamiseen kului eniten aikaa. Realististen kenkien mallintaminen hyvällä topologialla tuotti hankaluuksia, koska kengät eivät ole symmetrisiä objekteja. Ideaalissa topologiassa jokainen polygoni olisi nelikulmio mallin animaatiossa tapahtuvan deformaation vuoksi. Juoksuanimaatiossa jalkaterän luut tahtuvat ja sitä myöten niitä ympäröivä kengän topologia tahtuu luun mukaisesti. Esimerkiksi neliö tahtuu siististi kulmasta kulmaan kuin keskeltä tahtettu paperi, mutta mistä viisikulmainen paperi tahtuu tasaisesti? Sitä ei tiedä mikään muu kuin tietokone, joka useimmissa tapauksissa tekee taitoksen satunnaisesti ja rumasti.

Täydellinen neliötopologia ei ole aina mahdollista. Tähtiä eli pisteitä, joihin yhdistyy enemmän tai vähemmän kuin neljä muuta pistettä syntyy usein. Tähtiä ja monikulmioita pyritään hyvällä topologialla siirtämään alueisiin, jotka animoinnin aikana liikkuvat tai vääntyvät mahdollisimman vähän. Hyvän topologian edesauttamiseksi otin valokuvan omista kengistäni ja vein sen Blenderin taustalle referenssikuvaksi (kuva 17). Aloitan mallintamisen aina karkeasta muodosta vähäisillä polygoneilla. Lisään pisteitä neliöiden väliin, kunnes pystyn muovaaman mallin näyttämään halutulta muodolta (kuva 18).



Kuva 17. Oikeasta kengästä mallin ottaminen.



Kuva 18. Päähahmon kengät valmiina.

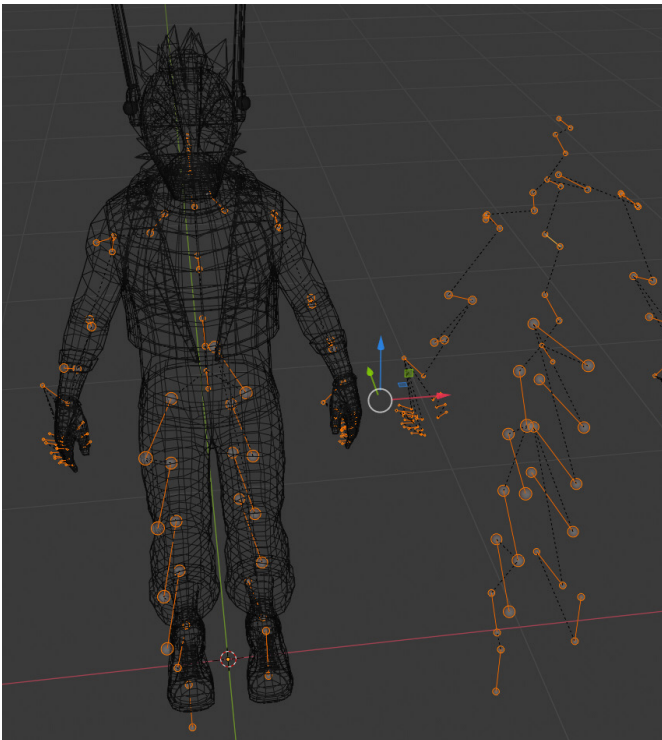
3.1.1 Mallin luuranko

3D-ohjelmassa täytyy asettaa ja määritellä mallin pisteille luut, jotta 3D-mallia voidaan animoida tai sen pisteitä edes liikuttaa. 3D-mallin luurangon englannin kielinen nimi on rig ja siitä seuraavaa prosessia kutsutaan rigaamiseksi. Normaalisti luut asetellaan manuaalisesti objektiin, jonka jälkeen luuranko voidaan animoida ja tuoda ohjelmaan valmiina. Prosessini ei ollut niin yksinkertainen. Työryhmässämme ei ole erillistä animoijaa, eikä resursseja kiinnittää yhtä työntekijää viikoiksi animoimaan itsetehtyä luurankoa tarvittaviin animaatioihin. Tehokkuuden vuoksi käytimme kolmannen osapuolen animaatioasetta, jossa oli kymmeniä laadukkaita animaatioita valmiina. Harmiksemme ne eivät suoraan sellaisenaan toimi hahmomme 3D-mallin kanssa. Luuranko täytyi irrottaa valmiiksi animoidusta hahmosta ja siirtää se osaksi omaa malliamme automaattisella prosessilla Armature Deform ja With Automatic Weights. (kuva 19). Valmiin rigin luiden asetelma ei tietenkään täsmännyt mallini anatomiaan ja ongelmia tuli heti. Hahmo kyllä juoksi kuten animoitu, mutta kädet, jalat ja yksittäiset pisteet vääntyivät sijoiltaan (kuva 20).

Oikeaoppisessa 3D-mallissa kädet ovat irti kehosta ja luut aseteltu anatomisesti oikein. Silloin automaattinen rigaaminen toimii yleensä melkein virheettömästi. Minun täytyi manuaalisesti määritellä jokaisen luun painoarvo jokaiselle mallin pisteelle erikseen, koska rigi ei ollut omani. Tätä prosessia kutsutaan nimellä weight painting. Suora käänös on painon maalaaminen. Se on keino kertoa tietokoneelle, kuinka voimakkaasti mikäkin rigin luu saa vaikuttaa mihinkin pisteeseen. Tumman sininen väri tarkoittaa, ettei luu vaikuta lainkaan. Vastakkaisesti punainen tarkoittaa, että kyseiset pisteet liikkuvat sataprosenttisesti luun mukana (kuva 21).



Kuva 20. Luurangon ja mallin välinen painoarvovirhe näkyvänä.



Kuva 19. Luuranko eli rigi.

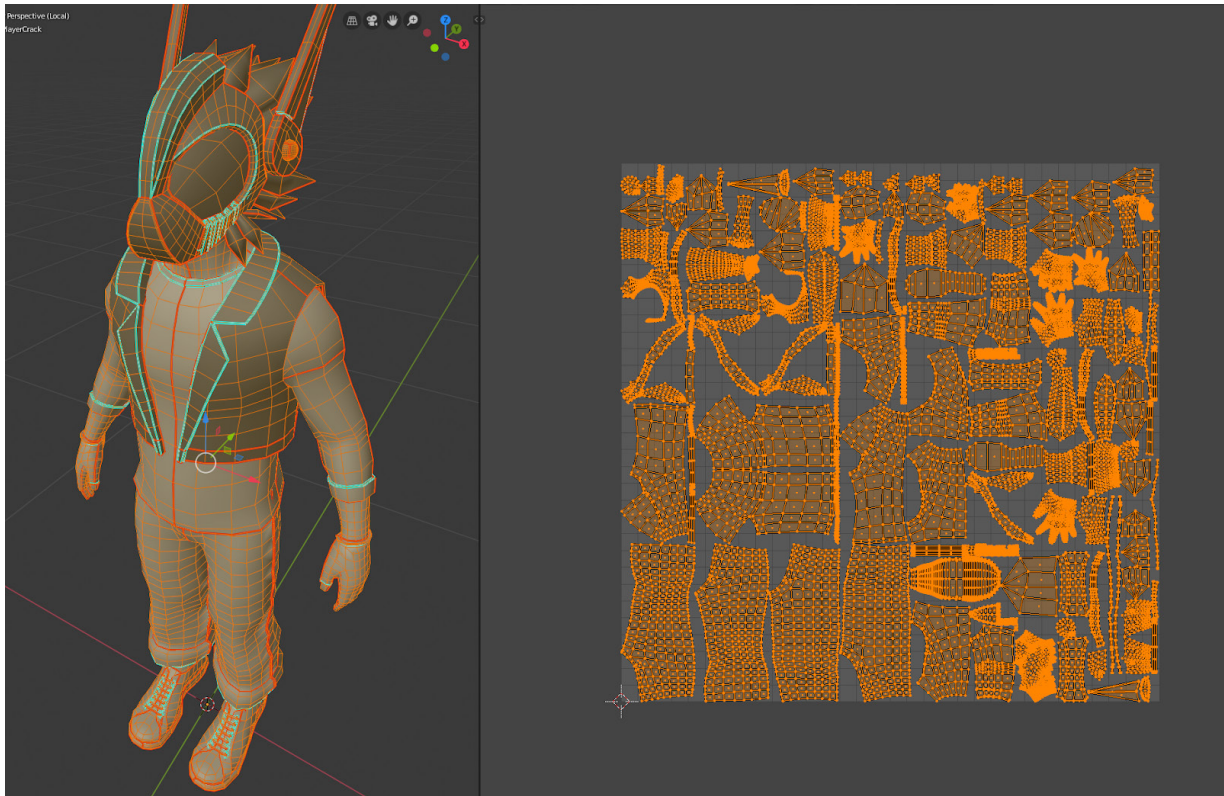


Kuva 21. Rigin painoarvon maalaaminen mallin pisteisiin.

3.1.2 Teksturointi

Rakensin hahmon 3D-mallin niin, että jokainen vaatekappale oli erillinen objekti. Erillisiä objekteja on aina helpompi käsitellä niin painoarvottamisessa kuin teksturoinnissa. Alkuperäisessä mallissani oli kymmenen eri materiaalia. Teksturointi on hidas prosessi ja siksi halusin säästää aikaa käyttämällä useaa materiaalia. Esimieheni yhdisti kaikki objektit yhdeksi ja usean eri materiaalin sijaan hahmolle jäi enää vain kaksi materiaalia. Syy tälle on Unityn tehovaatimusten helpottaminen VR-ympäristössä.

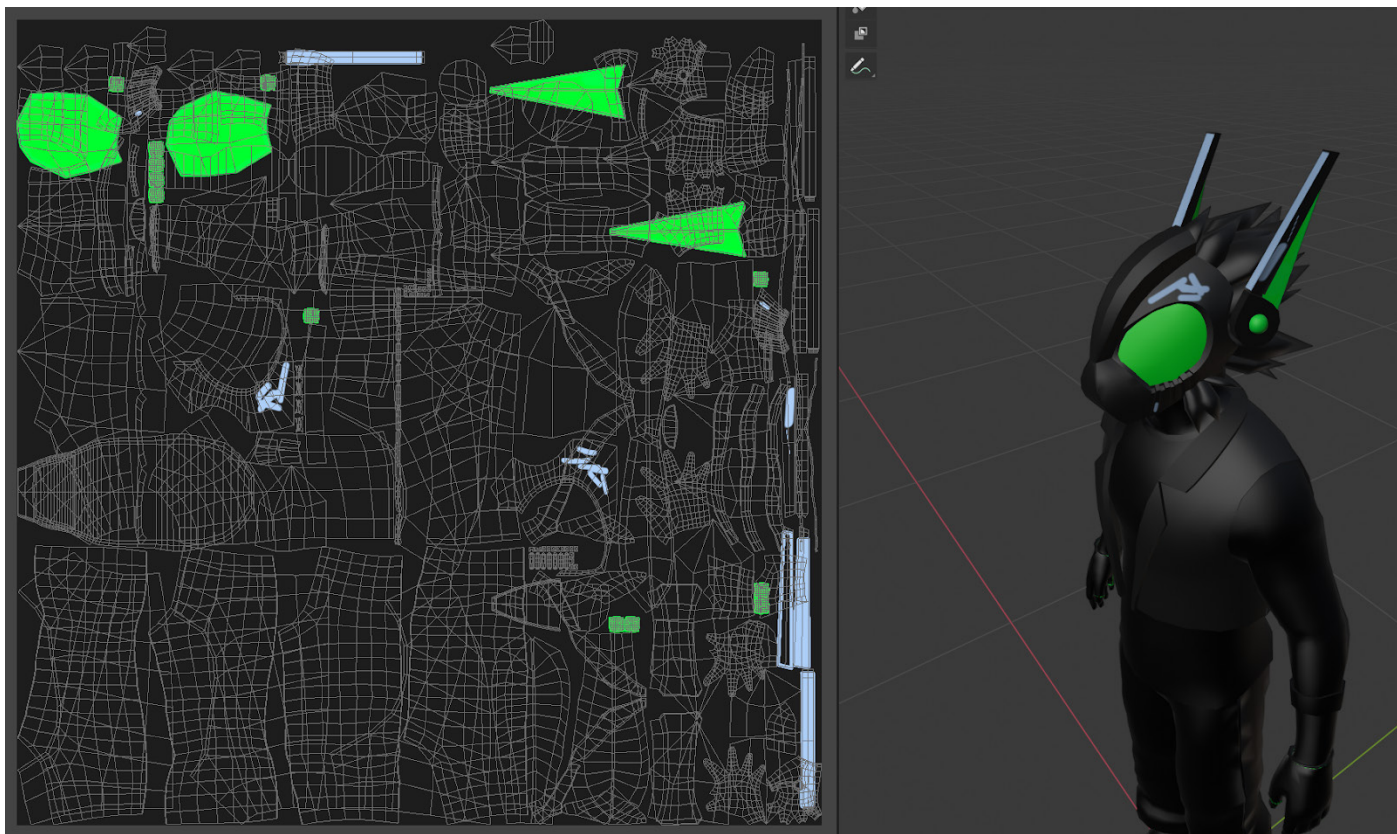
VR-lasien täytyy laskea pelin tehovaatimukset kahdelle eri ruudulle yhden sijaan. Unity lähettää VR-laseille jokaisesta materiaalista piirtovaatimuksen 60 kertaa sekunnissa eli yhteensä 1200 kertaa. Materiaalisupistuksen jälkeen kymmenestä kahteen, päähahmon tehovaatimus laski suhteellisesti 80 prosenttia.



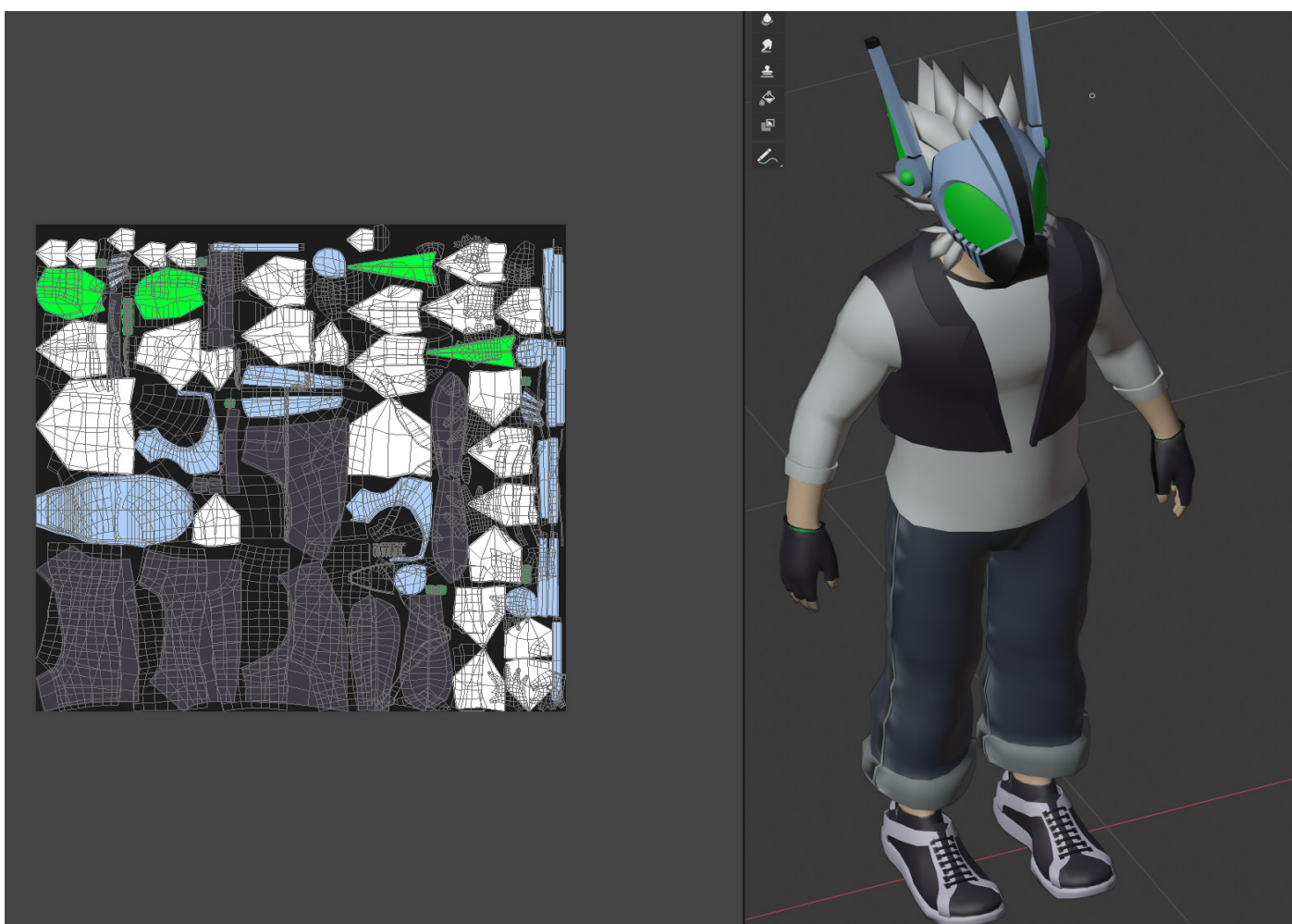
Kuva 22. UV-kartoitus.

Hahmon mallin teksturointi alkaa UV-kartoituksesta. Mallin särmät merkitään komennolla "mark seam", joka näyttää särmät blenderissä turkooseina. UV-kartoituksen tarkoitus on avata malli 3D-muodosta 2D-muotoon. Särmän merkitseminen on tapa kertoa ohjelmalle, mistä kohtaa topologian saa irrottaa, jotta 3D-malli avautuu litteäksi halutulla tavalla. Merkitsemisen jälkeen objekti avataan komennolla "UV unwrap" ja saadaan näkyviin 3D-malli 2D-muodossa. (kuva 22)

Pystyn piirtämään suoraan mallin päälle tai avattuun 2D-tekstuuriin. (kuva 23). Tekstuurin värityksen jälkeen se tallennetaan omaksi tiedostoksi ja viedään Unityyn, jossa se liitetään osaksi hahmon materiaalia. Unity osaa näyttää tekstuurin mallissa automaattisesti oikein, kunhan mallin tiedostossa on täsmävä UV-kartoitus. UV-kartoituksen jälkeen malli pitää tuoda uudelleen Unityyn. Pelimoottori tunnistaa määritetyt pisteet ja sitä kautta väritetyn 2D-kuvan oikean sijainnin suhteessa malliin. Teksturointia ei kannata prototyypeissä tehdä, ellei se ole myynnin tai pelimekaniikan vuoksi tärkeää. Päähahmon kaltainen keskeinen peliobjekti on aina pelaajan huomiopisteenä. On järkevää panostaa sen ulkonäköön enemmän kuin muihin prototyyppin osiin. Päähahmon mallissakaan ei kannata viedä teksturointia tarpeettoman pitkälle, vaan jättää se yksinkertaiseksi mahdollisten muutosten vuoksi.



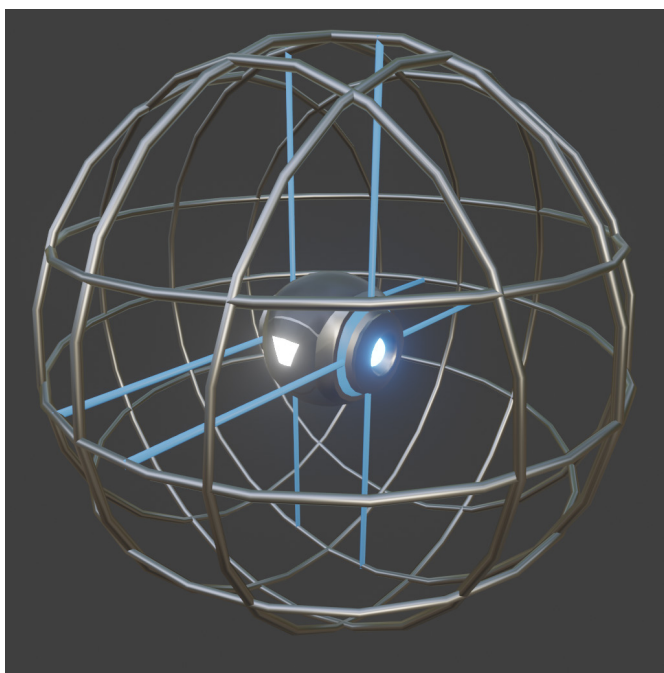
Kuva 23. Tekstuurien maalaaminen malliin.



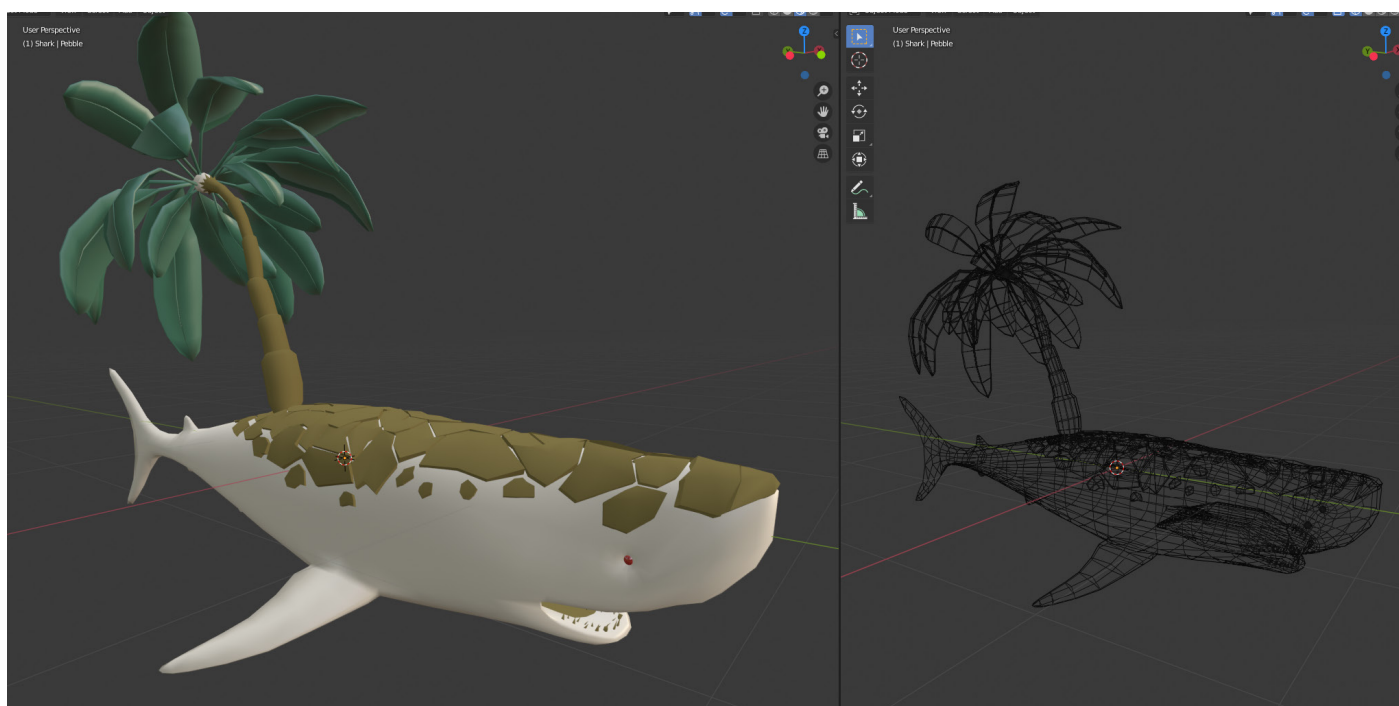
Kuva 24. Teksturointi valmiina.



kuva 26. Kaupungissa pelaajan ohi ajava rekka, joka tiputtaa pommeja.



Kuva 27. Mekaaninen arokierijä.



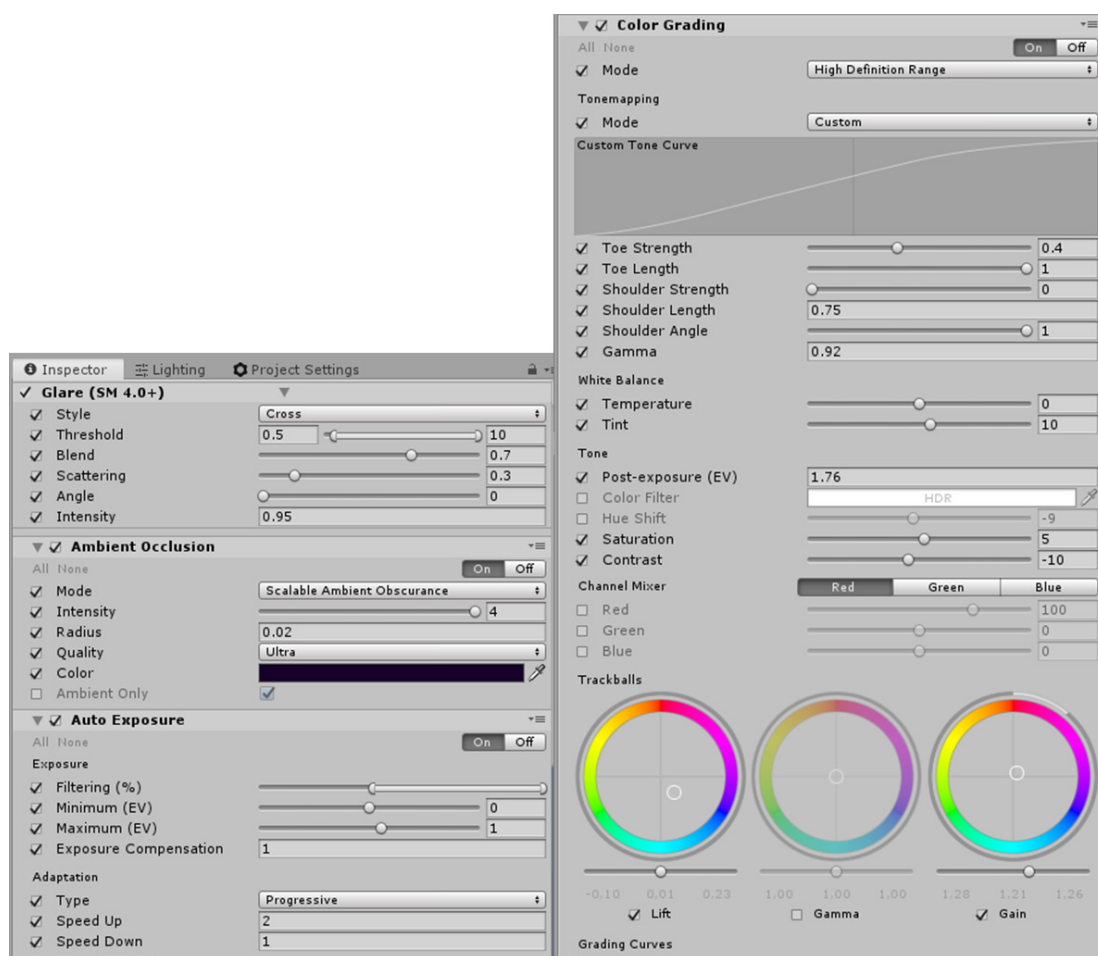
Kuva 28. Dyyneillä elävä hiekkahai.

4 Unity-pelimoottori

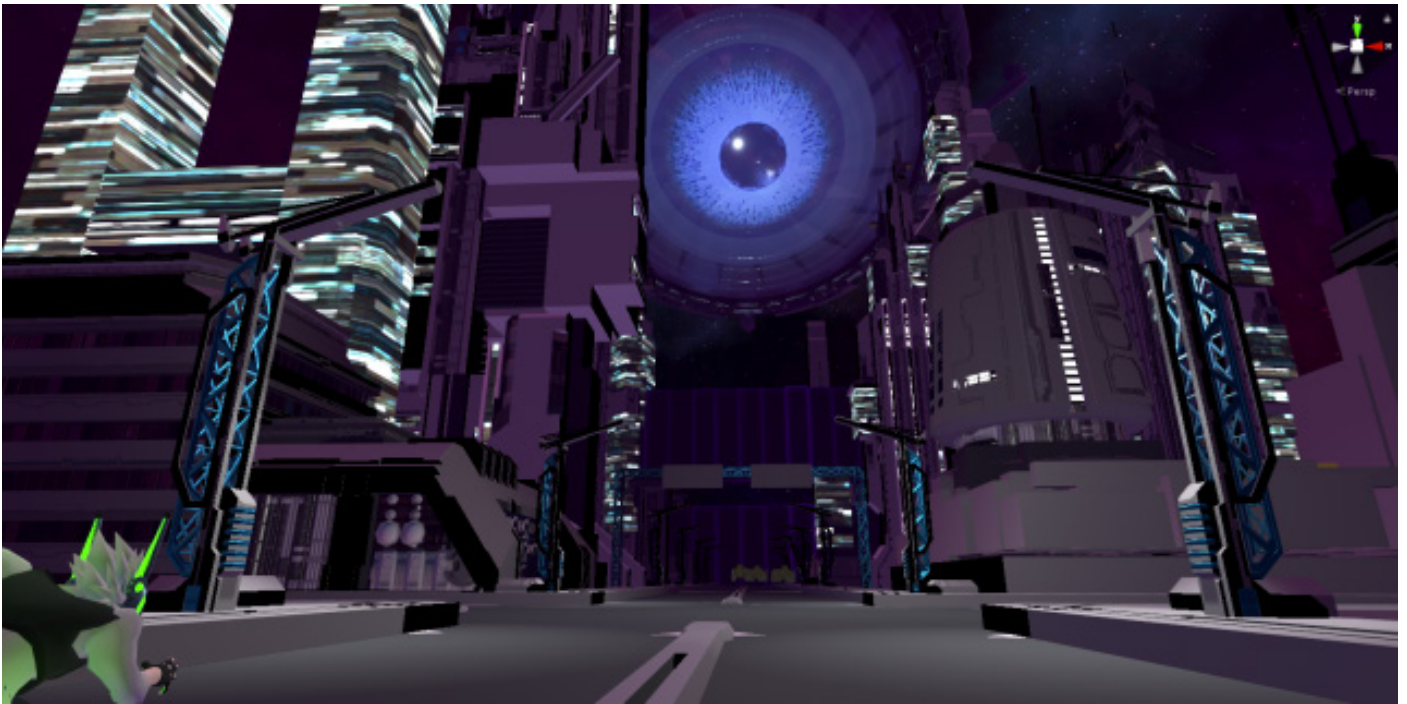
4.1 Jälkikäsittely

Normaalisti audiovisuaalisessa mediatuotannossa kuvien ja videoiden jälkikäsittely tapahtuu vasta aivan viimeisenä. Pelialalla jälkikäsittelyn määrittelemisen ensin ei ole ongelma, koska se on nopea tehdä ja helppo muokata. Prototyypin tapauksessa se auttaa hahmottamaan tunnelmaa paremmin. Prototyyppiä luodessa kaikki lisäarvoa tuottava työ, joka on helppo tehdä ja muokata on kannattavaa. Jälkikäsittely tapahtuu Unityn omalla "postprocess"-koodilla, jossa voi määritellä väriarvoja, hehkuefektiä, valoarvoja ja pienempiä efektejä, kuten rakeisuutta. Jälkikäsittely ei vaikuta itse pelin pelattavuuteen millään tavalla. Se on tietokoneella laskettu väriarvojen muutos.

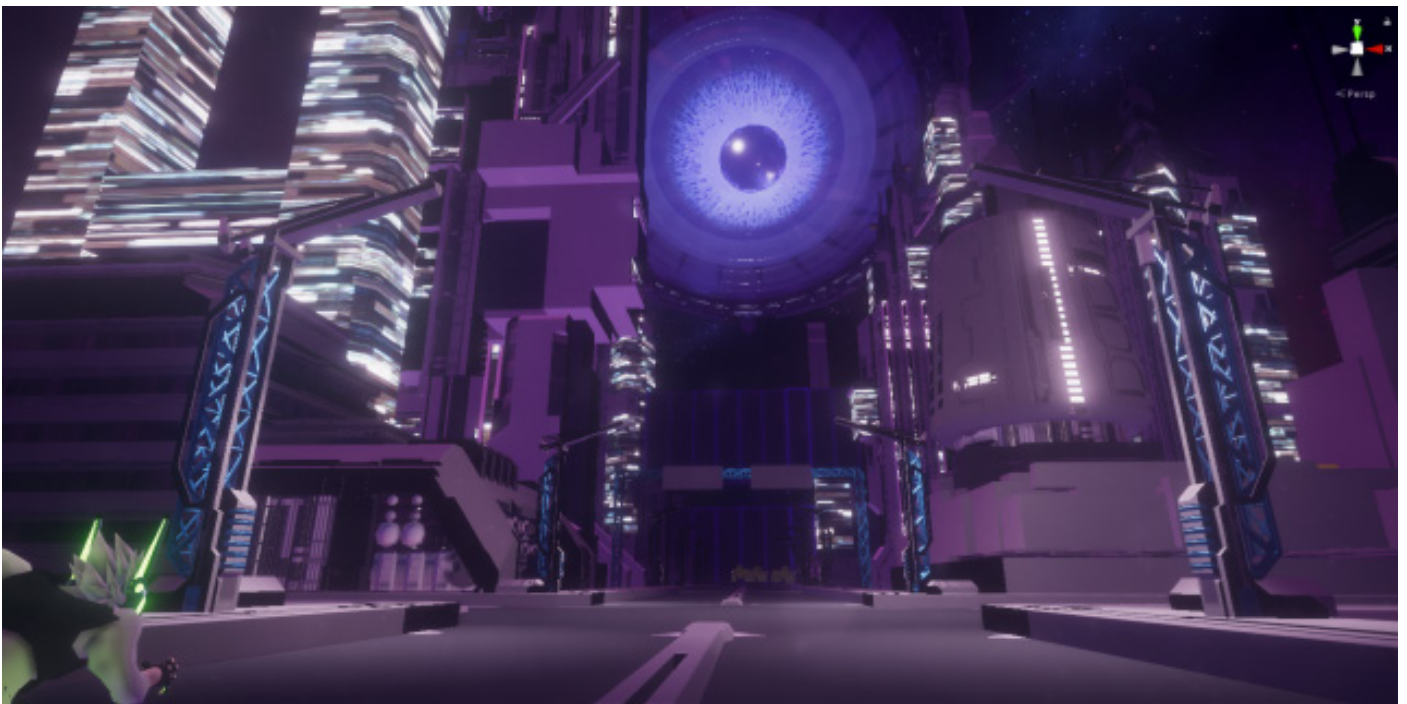
Projektini tärkeimmät jälkikäsittelyasetukset ovat Color Grading ja Glare. Color Grading muuttaa pelin väriarvoja niin, että varjot ovat tummempia ja vaaleat kohdat vaaleampia. Glare lisää vaaleimpiin pikseleihin ympärille hohtoeffektin, joka oikeassakin elämässä tulee valolähteiden ympärille. Asetukset tallennetaan Unityyn profiliksi, jota voidaan käyttää tai muokata koodilla tarvittaessa kesken pelin. (kuva 29). Jälkikäsittely tehdään peleissä siksi, että on tietokoneen laskentatehoille halvempaa väärentää efekti kuin oikeasti tuottaa se. Jos tietokoneen pitäisi laskea valojen hohdot 3D-tilassa, eikä jälkikäteen 2D-tilassa, se veisi liikaa tehoja eikä peli toimisi sujuvasti. Tein jälkikäsittelyprofiilin hyvin varhain prototyyppiin. Se on auttanut hahmottamaan ongelmia erityisesti liian kirkkaiden valojen tai materiaaliongelmiin kanssa.



Kuva 29. Unity PostProcess-profiili.



Kuva 30. Katukuva ilman jälkikäsittelyä.



Kuva 31. Katukuva jälkikäsittelyn jälkeen.

4.2 Ympäristöt

Peliympäristöt ovat laajoja alueita ja määrittävät olemassaolollaan pelimaailman tunnelman. Pelitestaamisen kannalta tilat voitaisiin hyvin jättää valkoisiksi kuutioiksi. Myynnin takia ympäristöihin piti panostaa ja niiden vaihtelevuutta esitellä yhtenä pelin myyntivalteista. Yhdenkin ympäristön valmiiksi rakentaminen on valtava urakka. Objekteja täytyy luoda satoja. Ne täytyy mallintaa, UV-kartoittaa, teksturoida ja animoida. Objektit pitää määritellä Unityssa käyttämään oikeita materiaaleja, tekstuureja, animaatioita ja koodeja yksi kerrallaan. Lopuksi ympäristö pitää rakentaa asettelemalla palat paikoilleen, sekä säätämällä valaistukset, heijastukset ja varjoasetukset. Prosessi pitäisi tehdä jokaiselle eri ympäristölle. Normaalisti objekteja voisi kierrättää. Yhtä tuolia voidaan käyttää useassa rakennuksessa, mutta mitä jos pelissä on vain yksi rakennus? Yleensä objekteja kierrätetään ovelasti. Tuolista voi tehdä ränsistyneen version metsän mättäälle, mutta sekin vaatisi lisää teksturointia. Työmäärä ei siis vähene eikä kierrättäminen tuoliesimerkissä veisi pois mallintamisen tarvetta muilta ympäristön objekteilta. Oma tapani on tiivistetysti: "Karsi, tyyllitele, lainaa, kierrätä ja piilota."

Samasta syystä jätin malleista pois yksityiskohtia. Pelaajat tuskin huomaavat ohi juostessaan, että ovista puuttuu lukko. Keskityin olennaisiin elementteihin. Esimerkiksi yökerhon ikonisimmat elementit ovat mielestäni diskopallo, valot ja tanssilattia. Sen sijaan, että lähtisin rakentamaan arkkitehtuurisesti yksityiskohtaista rakennusta, käytin Unityn sisäistä työkalua Pro-BUILDERia. Työkalulla voi rajatusti mallintaa Unityn sisällä 3D-malleja suoraan objekteiksi. Tekemällä yksinkertaisia muotoja ja toistamalla niitä saavutin uskottavan yökerhomaisen neonvalotunnelman suhteellisen nopeasti (kuva 32).



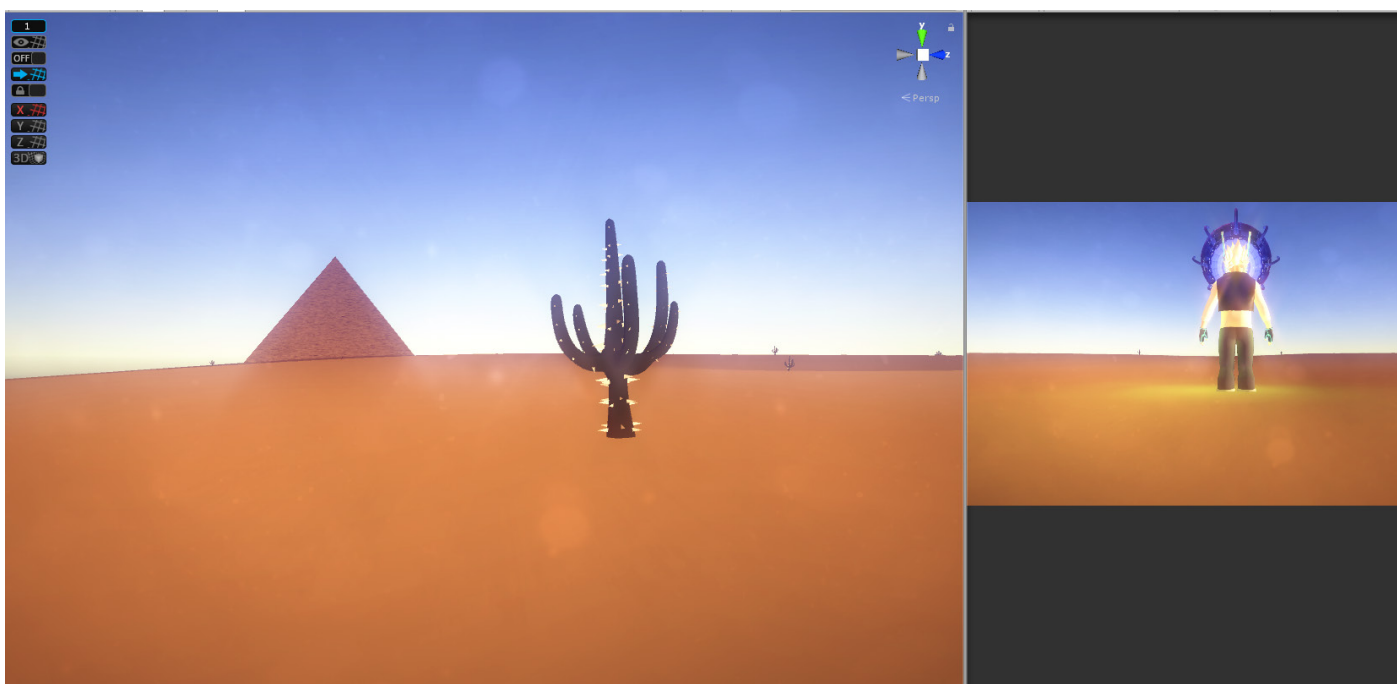
Kuva 32. Yökerho.

Yksinkertaiset muodot toimivat riittävästi sisätiloissa. Pro builder ei ole toimiva tapa tehdä orgaanisesti haastavia muotoja kuten puita. Tässä kohtaa oli viisasta käyttää taas Unity Storesta valmiita luonto-asetteja. Kuitenkaan hyviä yksittäisiä puu- tai pensas-asetteja ei ollut tarpeeksi. Kasvillisuutta pitäisi tehdä itse samalla tyyliä lisää. Se on vaativampaa kuin Unity:sta isomman paketin ostaminen. Molemmat vaihtoehdot vievät aikaa tai rahaa. Fiksumpi ratkaisu on luoda illuusio vaihtelevuudesta. Kierrätin vähäisiä objektejani vaihtelemalla puiden kokoa ja kiertokulmaa asetellen niitä sikin sokin satunnaisesti. (Kuva 33.)



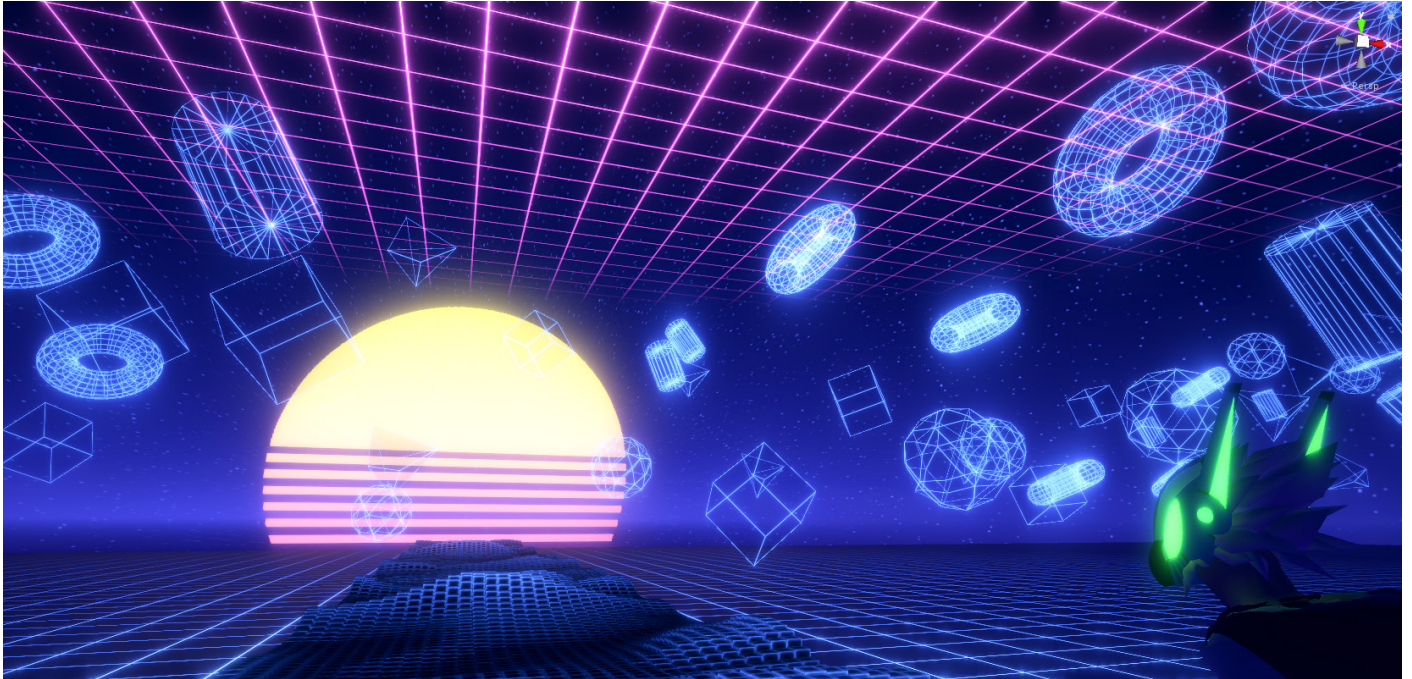
Kuva 33. Metsän raja.

Yksi tapa luoda lisää ympäristöjä pienellä vaivalla on valita ympäristön teemaksi sellaisia alueita, jotka luonnostaankin ovat hyvin autioita tai yksinkertaisia. Esimerkiksi aavikko on hyvin aukea ja karu ympäristö. Mallinsin Blenderissä valtavan hiekkadyynin, jonne tein kaveriksi kaktuksen ja pyramidin rikkomaan tyhjää horisonttia. Pyramidi oli yksinkertainen mallintaa muotonsa vuoksi, mutta kaktuksen orgaaninen topologia oli hieman hankalampaa. (Kuva 34.)



Kuva 34. Aavikkodyynit.

Viimeiseksi halusin luoda ympäristön, joka on uskollinen Synthwave-genrelle. Tein virtuaalisen kyberavaruustyylin suoraan 80-luvun elektroniikkamusiikin estetiikasta. Tyyliin kuuluu valtava auringonlasku, laserruudukko ja digitaalisia muotoja. Kierrätin samaa ruudukkomateriaalia ja tein Photoshop-ohjelmassa yksinkertaisen liukuväriympyrän auringonlaskuksi. Tyylittelin aurinkoa kumittamalla muutaman viivan pois sen alaosasta. Hahmon juoksema tie on Unity Storesta ostettu lisäosa, joka tuottaa musiikkiin reagoivan lattian. Tehosyistä kyseinen lisäosa todennäköisesti ei selviä lopulliseen tuotteeseen, mutta tuo toistaiseksi tunnelmaa. (kuva 35.)



Kuva 35. Synthwave-musiikkigenrestä vaikutteita ottanut virtuaaliympäristö.

4.3 Efektit

Tarpeellisuuden puolesta efektit ovat yhtä mitättömät ja tilanneriippuvaiset kuin tekstuurit. Efektit ovat oma taiteenlajinsa, jotka vaativat mielikuvitusta ja aikaa. Videopeleissä efektit voidaan mielestäni jaotella kahtia koriste-efekteihin ja informaatioefekteihin. Vaikka kaikki visuaalinen materiaali on informaatiota, koriste-efektit ovat nimensä mukaan vain tarpeettomia koristeita. Niiden tarkoitus on luoda tunnelmaa ja vaikuttaa pelaajan tunteisiin. Kauniit efektit voivat olla pelin myyntivaltti. Pelottavat efektit lisäävät kauhun tunteita. Peittävät efektit oikaisevat työmäärää peittämällä kentästä alueita, joihin muuten pitäisi mallintaa lisää objekteja.

Informaatioefektit ovat verrattavissa pelimekaanisiin osiin kuten käyttöliittymään. Näiden efektien on tarkoitus kertoa pelaajalle informaatiota pelistä. Riippuen efektin tärkeydestä, pelin testaaminen ilman niitä voi olla hankalaa tai mahdotonta. Kuvitellaan esimerkki virtuaalisesta lautapelistä. Pelaajilta suljetaan näköyhteys peliin, pimentämällä ruutu vuoron vaihtuessa. Ruudun pimeneminen on informaatiota rajoittava efekti, eikä tarvitse edes mitään grafiikkaa. Ruudun pimentyminen voidaan koodata pelkillä numeromuutoksilla. Se on silti efekti ja oleellinen mekaniikka pelin pelaamiselle.



*Kuva 36.
Päähahmon asean lataamisefekti.*

Projektiin piti tehdä jonkin verran koriste-efektejä tunnelman parantamiseksi. Lisäksi myös osittain informatiivisia efektejä. Päähahmon kranaatin lataaminen on yksi esimerkki (kuva 36). Pelaamisen kannalta on tärkeää tietää, kun kädessä on kranaatti. Testaamiseen riittäisi laajeneva ympyrä tai neliö. Vihreän sähköefektin luominen on koristeellinen efekti, eikä tuo lisäarvoa testattavuuteen. Se on mukana parantamassa pelin ulkonäköä myyntiesittelyihin. 3D-malliin tarvitaan mallinnusohjelma ja mallit toimivat pelimoottoreissa aina samoin. Efektit voidaan toteuttaa niin monella eri tavalla kuin mielikuvitus sallii. Ne voivat olla

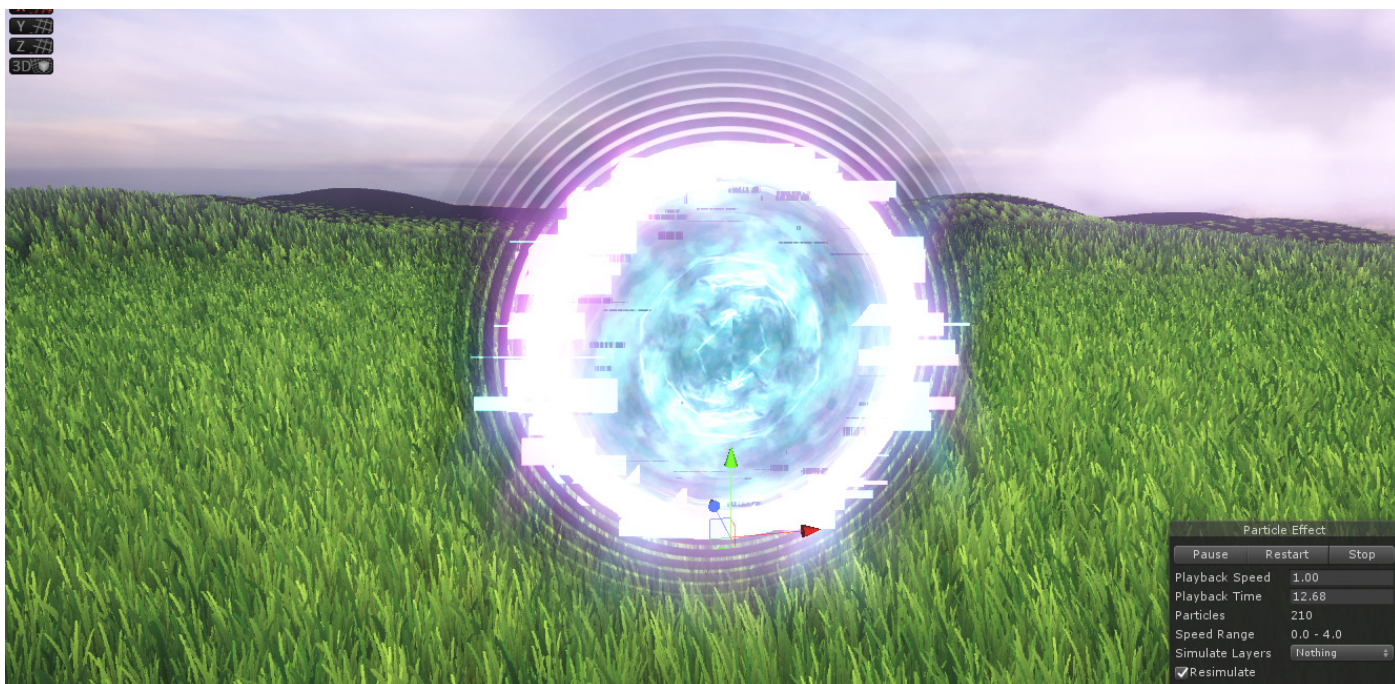
- fyysisiä objekteja
- 2D-kuvia 3D-ympäristössä
- koodin tuottamia väriefektejä
- jälkikäsittelyn tuottamia efektejä
- partikkelisysteemin tuottamia erikoisobjekteja
- ääntä
- videoita
- ohjaimen manipuloimista tärinänä.

Kranaatin latausefekti on uudelleen kierrätetty 2D-sähkötekstuuri lisättynä Unityn partikkelisysteemille. Partikkelisysteemin etu on moninkertainen objektimäärä vähäisellä laskentateholla. Eniten aikaa vie piirtää 2D-tekstuuri. Partikkelisysteemi itse on nopea tapa tehdä efektejä.

Rakensin koriste-efektiksi portaalin chunkkien siirtymisen välille, koska se on yhtä keskeinen elementti pelissä kuin päähahmo. Löysin internetistä hienon referenssikuvan digitaalisesti hajoavasta portaalista. Ajattelin sen sopivan pelin kyberavaruusmaiseen estetiikkaan (kuva 37). Käytin Unityn partikkelisysteemiä luomaan vastaavan efektin pelissä. En tarvinnut efektiin muita 2D-grafikoita kuin yksinkertaisen renkaan ja suorakulmion (kuva 38). Portaalin kaltainen efektikokonaisuus koostuu monesta useasta pienemmästä efektistä (kuva 39). Efektit voidaan laittaa omiksi objekteikseen, jotka jaetaan yhden pääobjektin alle. Vaihtamalla väriä, tapaa ja partikkeleiden elinikää, lopputulos näyttää paljon monipuolisemmalta kuin uskoisi saavutettavan yhdellä renkaalla ja suorakulmiolla.

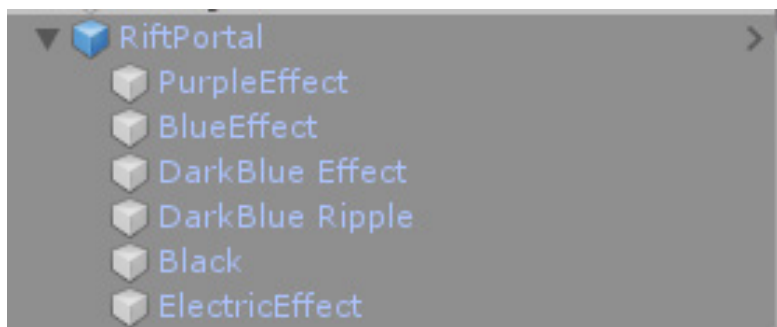


Kuva 37. Digitaalinen Glitch-rinkula vektorigrafikalla. (Zoa-Arts 2020)



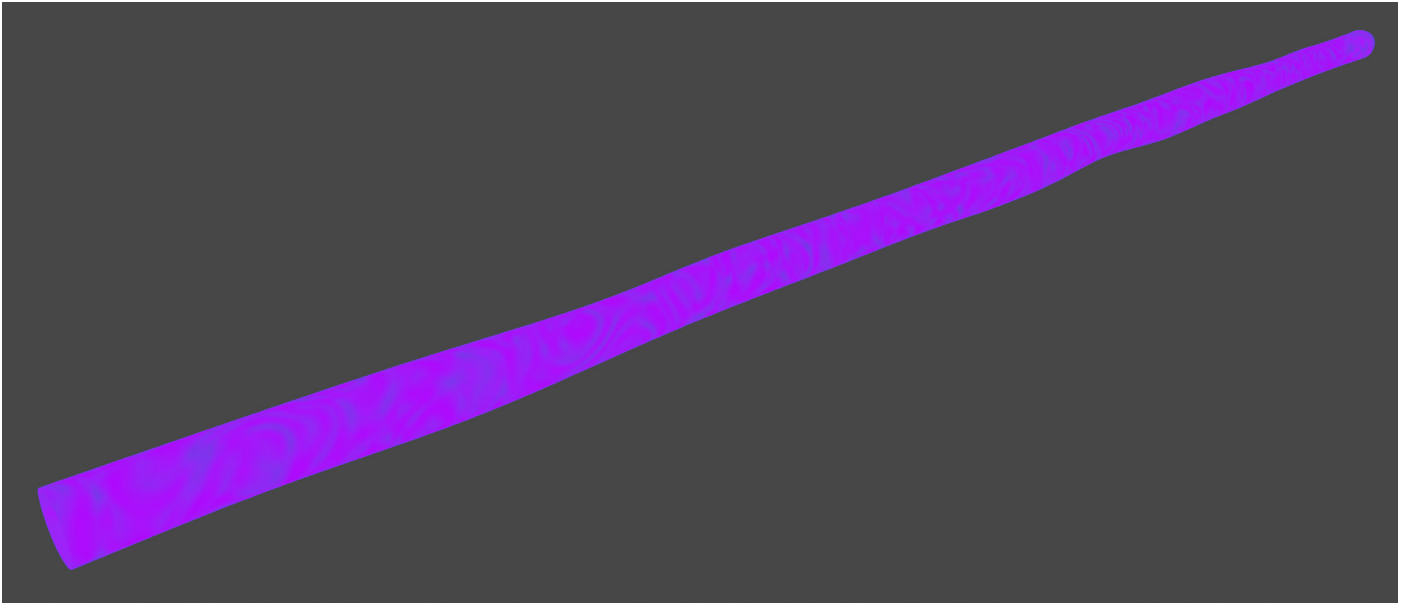
Kuva 38. Kenttien siirtymiseen käytettävä portaaliefekti.

Portaalin rakentaminen Unityn omilla työkaluilla antaa siistin, hienon ja tehooptimaalisen efektin, joka ei vie liikaa aikaa tehdä. Horisonttiin asti yltävät efektit, vievät liikaa tehoa jopa partikkelisysteemiltä. Partikkelisysteemi on täydellinen, kun objekteja on käytössä muutamasta kymmenestä tuhanteen. Suurempia efektejä kuten madonreikiä, jotka näkyvät loputtomasti horisonttiin asti, täytyy tehdä jotenkin toisin. Teksturoitu 3D-putki on tehokas tapa tehdä madonreikäefekti. Tekstuurin tekeminen ei ole kuitenkaan niin helppoa kuin yksi suorakulmio ja rinkula.

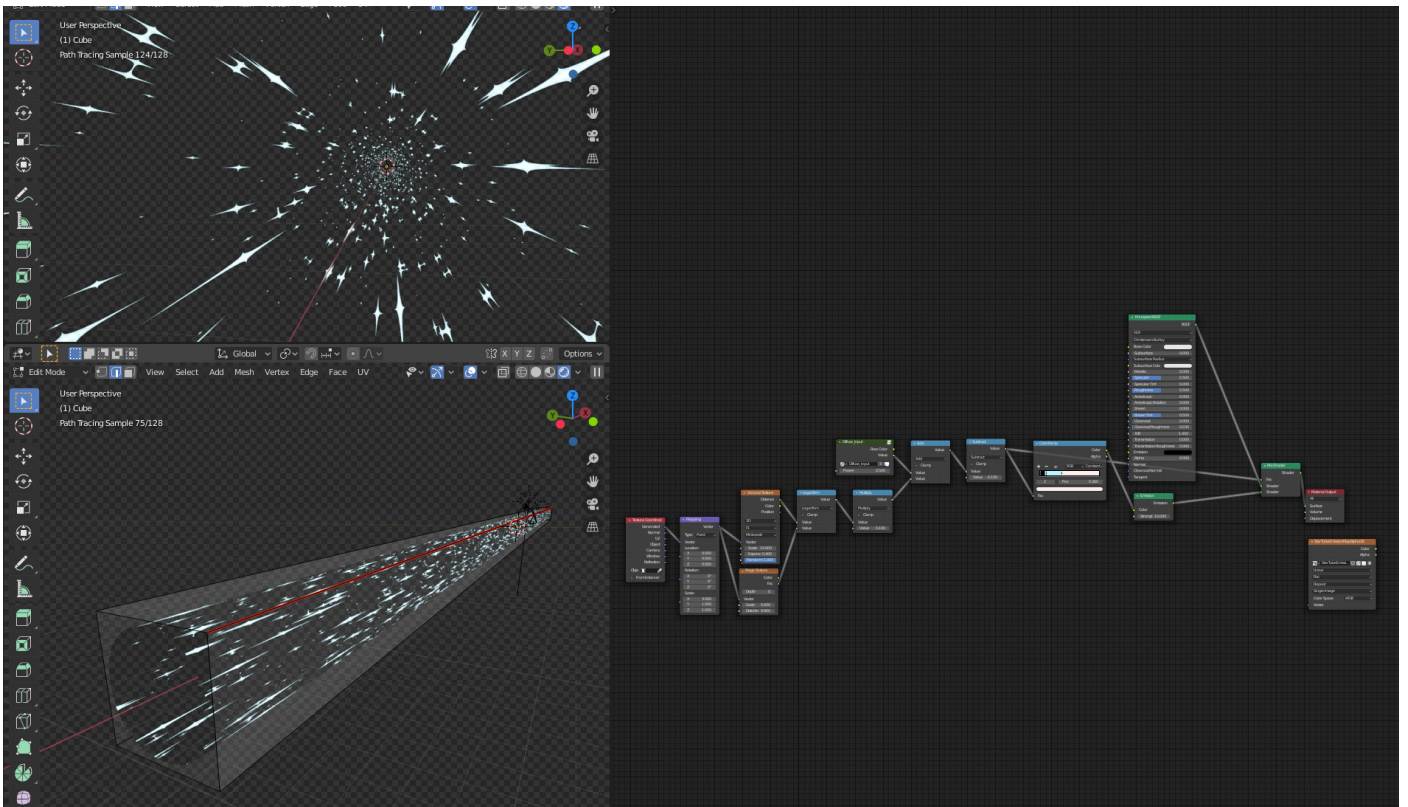


Kuva 39. Portaalin rakenne ja hierarkia Unityssä.

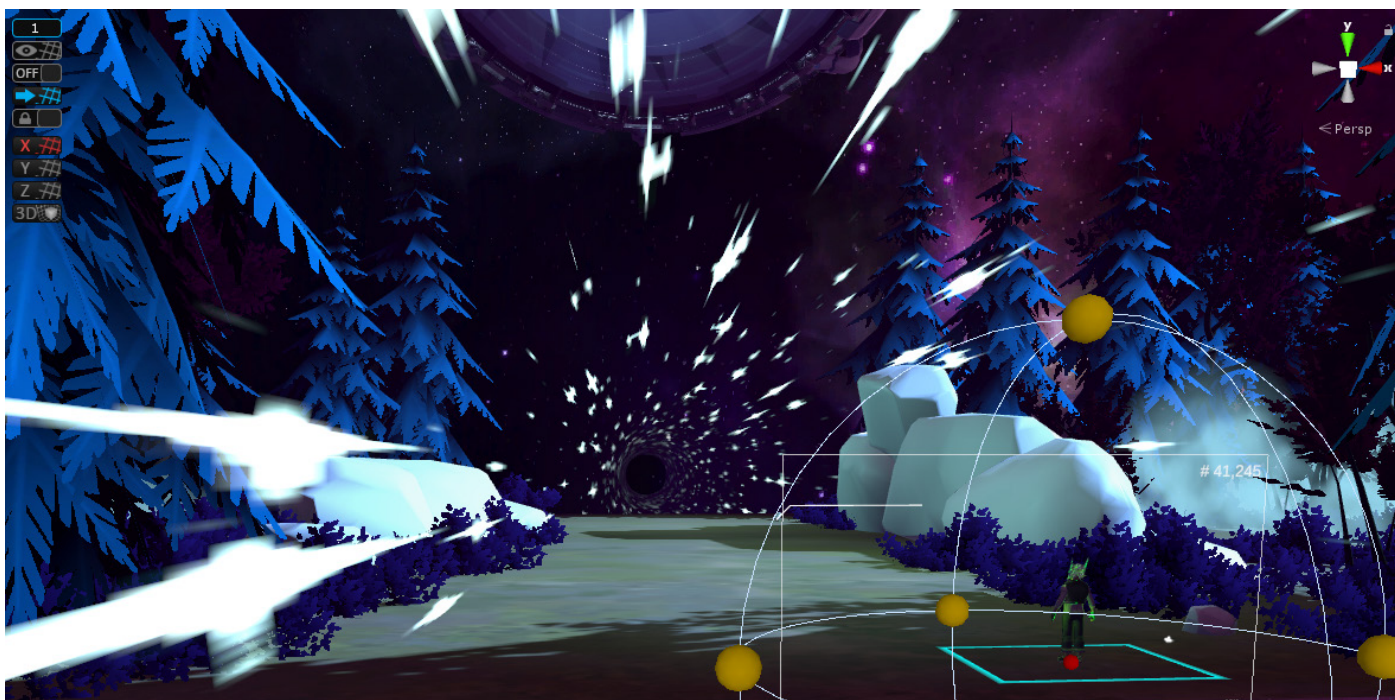
Putken mallintaminen on helppoa. Putki on primitiivinen sylinteri ja se löytyy jopa Unityn Pro builderin valikosta sellaisenaan. Tekstuuriksi voidaan generoida Photoshopissa tai Blenderissä tietokoneen laskema pilvitekstuuri (kuva 40). Pilvitekstuuri riittäisi yksinkertaiseksi testiksi. Jos halutaan parempia tai vaikeampia tekstuureja, ne täytyy joko itse piirtää tai generoida Blenderin Node-systeemillä (kuva 41). Node on käyttäjille tehty visuaalinen käyttöliittymä matemaattisista väriarvoista ja niiden yhdistämisestä. Sama asia voitaisiin koodata, mutta Nodessa se on käyttäjäystävällisempi tapa luoda tekstuureja. Node-generoiminen on näppärää, mutta vaatii taitoa, mielikuvitusta, kokemusta ja virheiden kautta kokeilemista. Luomisprosessi voi kestää matemaattisesta taidoista riippuen minuuteista muutamaa työpäivään. En tekisi prototyyppiin tämän kokoisia efektejä, ellei niitä oikeasti tarvittaisi.



Kuva 40. Tietokoneen generoima pilvitekstuuri putkessa.



*Kuva 41. Yksi osa portaalin siirtoefektistä Blenderin sisällä.
Oikealla puolella Blenderin Node-resepti efektin luomiseksi.*



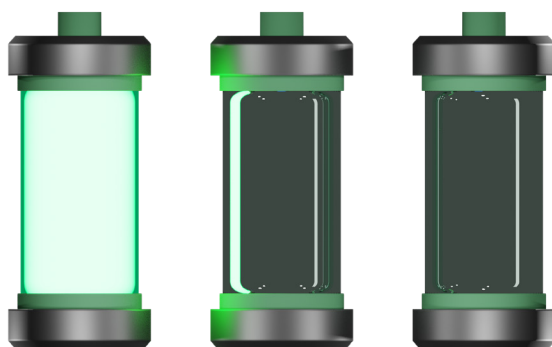
Kuva 42. Yksi portaalin efekteistä Unityn sisällä metsän keskellä.

4.4 Käyttöliittymä

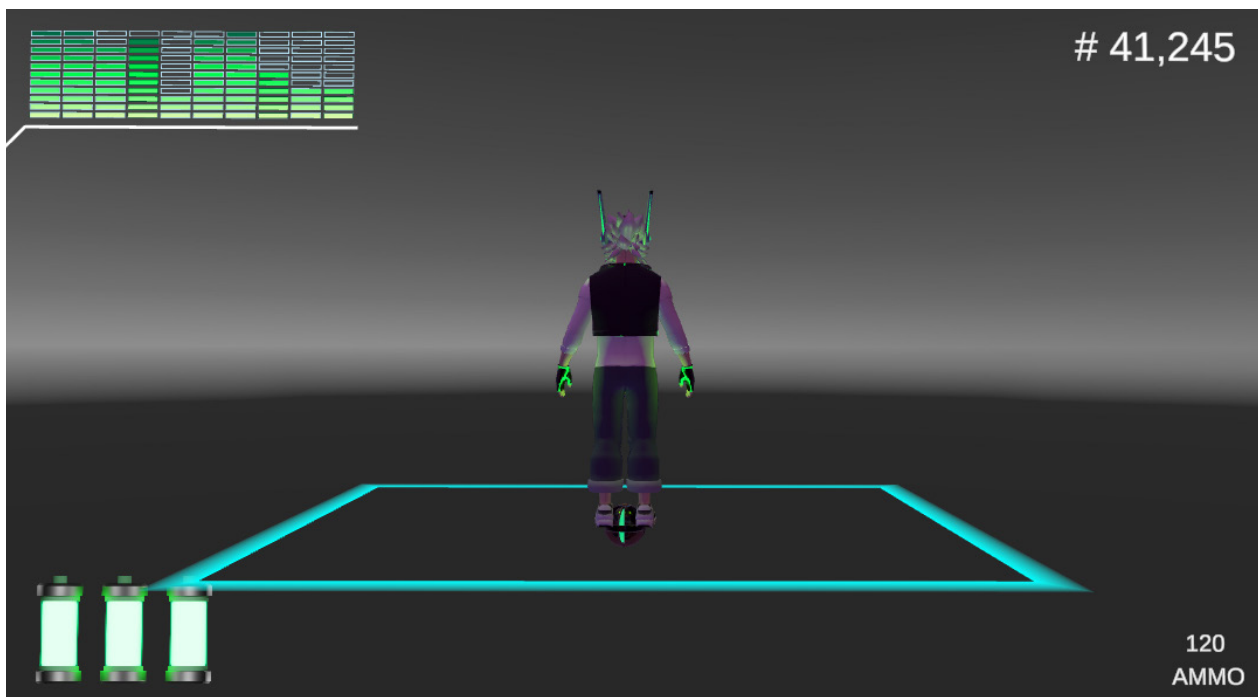
Globaalilta nimeltään UI eli User Interface on peleissä pelaajaa itseään varten. Siihen kuuluvat kaikki napit, numerot, tiedot ja grafiikat, joilla viestitetään pelaajalle haluttua informaatiota pelistä. Esimerkiksi elämäpisteet, ammusten määrä tai jopa vihollisten sijainti ja lukumäärä. UI jaetaan neljään kategoriaan. Spatial, meta, diegetic ja non-diegetic. (Bowers 2019).

Non-diegetic elementit eivät sijoitu pelin sisälle, eivätkä ole osa tarinaa. Esimerkiksi elämäpistemittari. Diegetic vastakohtaisesti on osa tarinaa ja osa pelikenttää. Esimerkiksi 3D-taskukello. Meta UI on osa tarinaa muttei pelikenttää. Esimerkiksi ruudulle tulevat 2D-efektit kuten veri tai lasisäröt. Spatial UI on vastakohta metaalle. Esimerkiksi nuolihologrammi maantiellä osoittamassa suuntaa. (Bowers 2019).

Yritin ensin sijoittaa UI:n VR-lasien kulmiin. (kuva 44). VR-laitteessa se on valitettavasti huono ratkaisu. Pään liikuttelun mukaan UI katoaa osittain tai liikkuu paikoiltaan jatkuvasti. Siirsimme UI:n interaktiiviseksi osaksi peliä eli osittain diegeettiseksi ja spatiaaliseksi käyttöliittymäksi. Koodia ei ole vielä tehty, mutta elämäpisteet tulevat sijoittumaan hahmon jalkojen alle vihreäksi kehäksi. Kranaatit jotka alun perin loin 2D-kuvaksi tulevat takaisin 3D-muotoon. Kranaatit ilmestyvät leijaillemaan hahmon viereen pelaajan komennosta (kuva 43). Pisteet tulevat ilmestymään hetkellisesti hahmon viereen numeroina ja katoavat pois tieltä nopeasti.



Kuva 43. Kranaattipatruunat 2D-mallina.



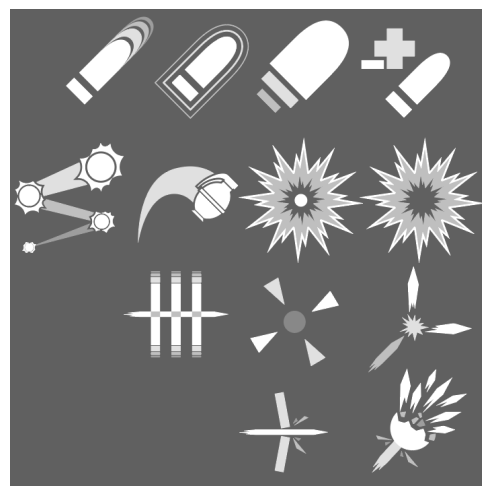
Kuva 44. Käyttöliittymän prototyyppi.

Ikonit ovat tapa jakaa nopeasti informaatiota pelaajalle pelistä (Gamedesigning 2019). Ne ovat merkkejä, jotka merkitsevät asian, paikan ja informaation. Ikonit tehdään erottumaan pelin muusta grafikasta. Esittelydemoon haluttiin saada mukaan hahmon aseeseen kerättävät lisäosat eli buffit. Buffi on pelaajan asetta muunteleva tai voimistava efekti (kuva 45). Näille buffeille piti saada ikoni merkitsemään pelaajalle pelin sisällä, mikä buffi on kyseessä ja mitä se tekee aseelle. Tehtävänantoni oli tehdä isoja yksinkertaisia toisistaan erottuvia muotoja. Muodot on tehty vektorigrafiikalla Adobe Illustrator -ohjelmassa yksinkertaisuuden ylläpitämiseksi.

Unity on siitä kätevä pelimoottori, että se voi vaihdella syötetyn kuvamateriaalin väriarvoja. Tekemäni ikonit ovat valkoisen ja harmaan sävyjä, jotta värejä voi Unityssa vaihdella vapaasti (kuva 46). Ne ovat valkoisuutensa takia kirkkaita ja pelikentässä hyvin erottuvia jopa väritettyinä. Ideaalisesti ikonit olisivat niin universaalisesti selkeitä, ettei niitä tarvitsisi selittää lainkaan, vaan pelaaja tietäisi heti mistä on kyse. Kaikkea voi tulkita ja varsinkin fiktiivisessä pelimaailmassa ikonien vastike voi olla todella monimutkainen. Tärkeämpää on eri efektien keskinäinen erottuminen ja samaa sarjaa olevien buffien tunnistaminen. Pelaajat kyllä ehdollistuvat nopeasti ja oppivat pelatessaan yhdistämään efektit ikoneihinsa.



Kuva 45. Unityn sisällä oleva keltaiseksi värimuunneltu räjähdde-buffi.



Kuva 46. Vektorigrafiikkaa hahmon aseeseen päivitystä varten.

4.5 Pelivalikko

Pelimme suoraviivaisuudesta huolimatta halusimme lisätä jonkinlaista taustatarinaa pelille. Päävalikko on yksi sellainen elementti, joka normaalisti jätetään vähälle huomiolle prototyypeissä. Päävalikko on pelimekaniikan testaamiseen tarpeeton ja palvelee tarkoitustaan vain yleisön käytössä. Päätimme kuitenkin tehdä jonkinlaisen päävalikon, koska prototyyppiä tulisi esittämään toimiston ulkopuolella hyvin varhain.

Tässä vaiheessa pelin nimi otettiin keskusteluun ja toistaiseksi kutsuimme sitä nimellä Mindframe. Nimi on viittaus pelin tarinaan, jossa päähenkilön kypärä vaikuttaa todellisuudentajuun. Tein alkuvalikon ympäristöksi ruohoniityn, koska halusin viestiä pelaajalle rauhallista tunnelmaa (kuva 46). Pelaaja pystyy valikossa pelaamaan päähahmolla samoin kuin itse pelissä. Tarkoitus on antaa pelaajien harjoitella ja tottua ohjaimiin kaikessa rauhassa. Kun pelaaja haluaa aloittaa pelin, hän ampuu valikon Start-painikkeeseen ja peli alkaa. Heti aloituksen jälkeen ympäristö muuttuu punertavaksi pätsiksi, ja meteoriitit alkavat tippua taivaalta räjähdellen maahan (kuva 47). Kierrätin tuliefektit ja kivet olemassa olevista Unity Storen elementeistä ja työpari auttoi koodin kanssa. Pelaaja juoksee melkein heti portaaliin ja pois vaarasta. Tarkoitus on sytyttää pelaajalle tunne nopeatempoisesta pelistä, jossa pysähtyminen ei ole vaihtoehto.

Ympäristön muutosefekti on yhdistelmä ruohomateriaalin värin-, taivaankannen- ja valaistuksen vaihtoa, sekä jälkikäsittelyprofiilin muokkaamista. Vaihtuminen tapahtuu koodissa työparilleni ohjeistamallani tavalla. Kerroin esimerkiksi, että haluan taivaan valaistusarvon muuttuvan 0.5 yksikköä 10 sekunnin aikana.



Kuva 47. Päävalikko ruohokentässä.



Kuva 48. Päävalikkokentän pätsiversio.

5 Lopuksi

Opinnäytteeni on oudoin koskaan kokemani projekti. Se on yksi onnistuneimpia ja parhaimpia projektejani vieläpä omasta ideastani. Samaan aikaan se on herättänyt epäilystä kyvyistäni, tavoitteistani ja motivaatiostani alalle. En osaa sanoa onko osasyy elämäntilanne, työn etätyöluonne vai 2020-vuoden koronapandemian karanteeni.

Lopputulos on, että toivoin itseltäni enemmän ja ehkä liikaakin. Haaveilin saavani enemmän, isompaa ja parempaa aikaan. Mutta toisaalta se mitä tein, on jo nyt enemmän kuin koskaan ennen. On vaikea hyväksyä tekemänsä työ, kun uskottelee itselleen, ettei antanut parastaan. Ja uskoo niin, siksi koska peli ei näytä sadan miehen studion tuotokselta viiden vuoden jälkeen. Yksi prototyypin rakentamisen ongelmista on selvän suunnan puute. Minulla ei ollut selkeää kuvaa kaikista yksityiskohdista, vaikka teema oli omani tai edes siitä millaiseksi projekti muokkaantuisi alkuperäisestä. Suunnitelmat muuttuivat, elämäntilanteet vaikuttivat ja paljon näkymätöntä työtä tuli tehtyä.

Peli on toimiva prototyyppi, se näytettiin sijoittajille ja saimme heiltä kiinnostusta jatkaa pidemmälle idean puolesta. Ne tavoitteet, jotka kuukausia sitten vedin lonkalta arvioiden muutamassa minuutissa, tulivatkin hyvin todellisiksi ja realistisiksi rajoiksi. Pelikehityksessä ja prototyypin rakentamisessa tehdään paljon elementtejä, joita ei koskaan tulla käyttämään eivätkä ehdi demoon sisään. Niitä ovat muun muassa hiekkahai, jotkin yökerhon chunkit ja portaalin efektit. Olisiko lopputuloksessa ollut parantamisen varaa? Ehdottomasti, mutta ehkä tärkeimmät kaikista pelikehittäjän kyvyistä ovat vaatimattomuus ja nöyryys (Koen 2019). Näillä sanoilla päätän opinnäytetyöni kirjallisen osion:

”Tällaista tällä kertaa.”

Lähteet

Elektroniset lähteet

A. Hackney Blackwell, E. Manar. 2015. UXL Encyclopedia of Science (3rd ed.).

Prototype. [Viitattu: 15.4.2020].

Saatavissa: http://ic.galegroup.com/ic/scic/ReferenceDetailsPage/ReferenceDetailsWindow?failOverType=&query=&prodId=SCIC&>windowstate=normal&contentModules=&display-query=&mode=view&displayGroupName=Reference&limiter=&currPage=&disableHighlighting=false&displayGroups=&sortBy=&search_within_results=&p=SCIC&action=e&catId=&activityType=&scanId=&documentId=GALE%7CENKDZQ347975681&source=Bookmark&u=dclib_main&jsid=63ed7b4b8ba090d7346c2d926005d307

The product design expert. 2016.

What Is A Working Prototype. [Viitattu: 17.4.2020].

Saatavissa: <https://product-design-prototype-experts.com/what-is-a-working-prototype/>

3D printing expert. 2015.

What Is A Visual Prototype. [Viitattu: 17.4.2020]

Saatavissa: <https://3d-printing-expert.com/what-is-a-visual-prototype/>

P. Polsinelli. 2018.

What is a game Prototype? Blogi. [Viitattu: 17.4.2020]

Saatavissa: <https://medium.com/@ppolsinelli/what-is-a-game-prototype-506472ec67ad>

Merriam-Webster. 2020.

Definition of game, Nettisanakirja. [Viitattu: 17.4.2020]

Saatavissa: <https://www.merriam-webster.com/dictionary/game>

D. Tach. 2018.

The creators of 'Fortnite' just unveiled an ambitious plan to make PlayStation and Xbox work together, whether they like or not. [Viitattu: 17.4.2020]

Saatavissa: <https://www.businessinsider.com/epic-games-cross-play-xbox-playstation-nintendo-pc-2018-12?r=US&IR=T>

B. Kuchera. 2016.

The complete guide to virtual reality in 2016 (so far). Nettilehti. [Viitattu: 17.4.2020]

Saatavissa: <https://www.polygon.com/2016/1/15/10772026/virtual-reality-guide-oculus-google-cardboard-gear-vr>

Luque, Raul. 2012.

The Cel Shading Technique. PDF [Viitattu: 24.5.2020]

Saatavissa: https://raulreyesfinalproject.files.wordpress.com/2012/12/dissertation_cell-shading-raul_reyes_luque.pdf

S. Koen. 2019

Why humility is the most important trait of a great developer. [Viitattu: 21.4.2020]

Saatavissa: <https://www.freecodecamp.org/news/a-first-class-temperament-the-emergence-of-a-star-developer-9321eeb57005/>

Autodesk. 2017

Standard Primitives. [Viitattu: 21.4.2020]

Saatavissa: <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-66152BDE-BA64-423F-8472-C1F0EB409E16-htm.html>

LaneFox. 2011.

Should you build your game when you do not have the artwork? Unity-forumi. [Viitattu: 21.4.2020]

Saatavissa: <https://forum.unity.com/threads/should-you-build-your-game-when-you-dont-have-the-artwork.338797/>

GameDesigning. 2019.

Beginners Guide to Video Game Icons. [Viitattu: 25.4.2020]

Saatavissa: <https://www.gamedesigning.org/learn/icons/>

M. Bowers. 2019.

Level Up - A Guide to Game UI (with Infographics). [Viitattu: 29.5.2020]

Saatavissa: <https://www.toptal.com/designers/gui/game-ui>

Kuvalähteet

E. Amos. 2017

Oculus Rift CV1 Headset. #1 Kuva. [Viitattu: 17.4.2020]

Saatavissa: <https://commons.wikimedia.org/wiki/File:Oculus-Rift-CV1-Headset-Back.jpg>

Samwalton9. 2017

Oculus Touch Controllers for the Oculus Rift CV1. #2 Kuva. [Viitattu: 17.4.2020]

Saatavissa: https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Oculus_Touch_Controllers.jpg

togemet2. 2019

Spider-Man 4 video game prototype. #6 Kuvakaappaus. [Viitattu: 17.4.2020]

Saatavissa: <https://www.gamingalexandria.com/wp/2019/12/01/spider-man-4-wii-unreleased/>

T-tus. 2005

Toon-shader. #8 Kuva. [Viitattu: 25.4.2020]

Saatavissa: <https://commons.wikimedia.org/wiki/File:Toon-shader.jpg>

ico83. 2006.

Catmull-Clark subdivision of a cube. #12 Kuva. [Viitattu: 24.4.2020]

Saatavissa: https://commons.wikimedia.org/wiki/File:Catmull-Clark_subdivision_of_a_cube.svg

Zoa-Arts. 2020.

Abstract glitch ring with distortion light effect vector image. #37 Kuva. [Viitattu: 23.4.2020]

Saatavissa: <https://www.vectorstock.com/royalty-free-vector/abstract-glitch-ring-with-distortion-light-effect-vector-29383886>