



TAMPEREEN  
AMMATTIKORKEAKOULU

# 3D-SISÄLLÖN TUOTTAMINEN VR-PELIIN

Arttu Laurila

Opinnäytetyö  
Toukokuu 2017  
Tietojenkäsittelyn koulutusohjelma  
Pelituotanto



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Pelituotanto

LAURILA, ARTTU  
3D-sisällön tuottaminen VR-peliin

Opinnäytetyö 61 sivua, joista liitteitä 4 sivua  
Toukokuu 2017

---

Tämän opinnäytetyön aiheena on 3D-grafiikan tuotantoprosessi VR-peliä varten. Prosessin kuvaus käsittää 3D-grafiikan tuottamisen konseptitasolta aina valmiiksi kokonaisuudeksi asti. Opinnäytetyön toimeksiantajana oli peli- ja ohjelmistoalan yritys nimeltään Virtual Dawn. 3D-grafiikka tuotettiin toimeksiantajan kehittämään VR-peliin nimeltään Outrageous Grounds: The Maze. Opinnäytetyön tarkoituksena oli dokumentoida 3D-grafiikan tekoprosessi ja tutkia miten prosessi eroaa perinteisestä pelinkehityksestä.

Työn alussa käsitellään VR:n konsepti, historia ja alustan vahvuudet sekä heikkoudet. Sen jälkeen 3D-grafiikan suunnitteluprosessi selvitetään osa kerrallaan alkaen konseptivaiheesta, asiakkaan tarpeiden määrittelystä ja käytettävästä teknologiasta. Lopuksi opinnäytetyö esittelee itse 3D-mallinnusprosessin, tekstuurikarttojen tuottamisen ja kaiken yhdistämisen valmiiksi lopputuotteeksi.

Opinnäytetyöprojektin aikana luotiin pelikenttä, joka koostuu seitsemästäkymmenestä projektia varten luodusta 3D-mallista. Pelikenttä on rakenteiltaan valmis kokonaisuus, johon toimeksiantajan on helppo lisätä projektin päätyttyä pelilliset ominaisuudet ja julkaista osana peliä.

## **ABSTRACT**

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

LAURILA, ARTTU  
Production of 3D content for a VR game

Bachelor's thesis 61 pages, appendices 4 pages  
May 2017

---

The subject of this thesis is the production of 3D graphics for a VR game from conceptual level to a complete, textured whole. Production of graphics was commissioned by a company called Virtual Dawn that is developing a VR game called Outrageous Grounds: The Maze. The aim of the thesis was to document the production process of 3D graphics for the client and research how the process differs from conventional game development.

The beginning of the thesis explains the concept and history of virtual reality and the strengths and weaknesses inherent to the platform. After that the work process will be explained bit by bit starting from the conceptual stage, the needs and wants of the client and the technology used. Finally, a discussion is provided on the 3D modeling process, the production of texture maps and the combining of everything into an end product in Unity game engine.

During the course of the thesis project a game level consisting of seventy models made for the project was created. The game level is a structurally ready whole, into which the client is able to easily add gameplay functions and release as a part of the game after the project has finished.

---

Key words: 3D, modeling, VR

## SISÄLLYS

1	JOHDANTO.....	6
2	VIRTUAALINEN TODELLISUUS SELITETTYNÄ .....	7
3	MITEN VR-PELI EROAA TAVALLISESTA PELISTÄ? .....	11
4	MIKSI VALITA VR ALUSTAKSI?.....	12
5	MITEN SUUNNITELLA TASO VR-PELIIN .....	13
5.1	Asiakkaan lähtökohta.....	13
5.2	Käytettävä teknologia .....	17
5.2.1	Unity.....	17
5.2.2	Maya.....	17
5.2.3	xNormal.....	18
5.2.4	Quixel Suite 2.....	18
5.3	Rajoitteet .....	18
5.4	Konsepti .....	19
6	MITEN LUODA VR-PELIN TASO VISUAALISESTI.....	22
6.1	Kuinka luoda tarvittavat 3D-mallit .....	22
6.2	3D-mallin konseptointi .....	23
6.3	3D-mallin geometrian luonti .....	24
6.4	Huomioitavat asiat 3D-mallin geometrian tuotannossa .....	27
6.5	Mallien UV-projisointi.....	32
6.6	Mallien valmistaminen teksturointia varten.....	35
6.7	Teksturointi .....	40
7	MITEN KOKOONPANNAA TASO UNITY-PELIMOOTTORISSA.....	44
7.1	Sijoittelu ja hierarkia.....	45
7.2	Mallin ja tekstuurien tuominen Unityyn .....	47
7.3	Prefabien luonti.....	50
8	POHDINTA .....	51
9	LÄHTEET.....	53
10	LIITTEET .....	55

**SANASTO**

VR	virtuaalinen todellisuus
Mesh	3D-mallin ulkokuori, verkko
Verteksi	3D-mallin kulmapiste tai kärki
Face	3D-mallin taho
Edge	3D-mallin sivu
UV-mappaus	3D-mallin tahojen projisointi kaksiulotteiselle pinnalle
Tekstuuri	Pintamateriaali tai väritys 2D- & 3D-malleille.
Asset	Yleisnimitys tuotetulle 3D-objektille, tekstuurille äänelle tai koodinpätkälle.
Scene	Unity-pelimoottorin termi pelikentälle

## 1 JOHDANTO

Tämän opinnäytetyön aiheena on 3D-tuotanto VR-peliä varten. Projektin ja opinnäytetyön toimeksiantajana on Virtual Dawn -niminen yritys. Opinnäytetyö sai alkunsa toimeksiantajan ehdotuksesta yhdistää yritykseen tehty työharjoittelu ja opinnäytetyö yhdeksi kokonaisuudeksi. Koska varsinaisen työharjoittelun tarkoituksena oli tuottaa toimeksiantajan kehittämää peliä varten taso, keskittyy opinnäytetyön aihe kentän rakentamiseen vaadittavien 3D-mallien tuotantoon ja niiden kokoonpanoon Unity-pelimoottorissa.

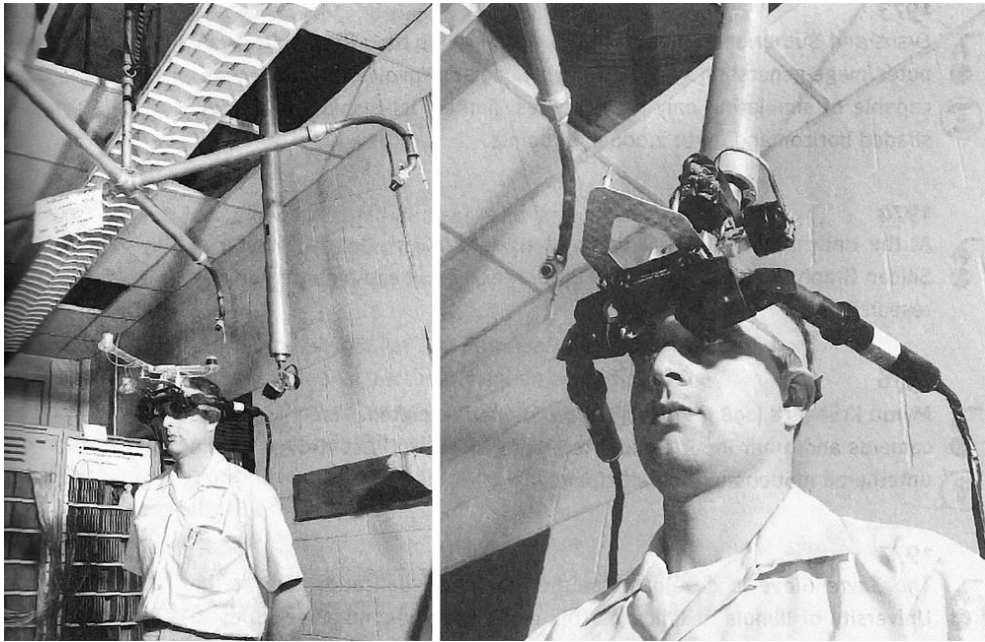
Opinnäytetyössä selvitetään mitä VR eli virtuaalinen todellisuus on, minkälainen historia sillä on ja miksi VR on kiinnostava valinta alustaksi. Tämän jälkeen käydään konseptoinnin ja työkalujen esittelyn kautta läpi koko mallintamisprosessi, tekstuurien tuottaminen ja mallien tuominen Unity-pelimoottorin sisälle. Opinnäytetyön tarkoituksena oli luoda dokumentaatio ja malli 3D-tuotannosta asiakasyrityksen käytäntöjen mukaisesti sekä tutkia, miten VR-pelin sisällön suunnittelu ja 3D-objektien tuotanto eroavat perinteisestä pelinkehityksestä. Opinnäytetyöprojektin tavoitteena oli kehittää toimeksiantajalle taloudellista hyötyä tuottavia ratkaisuja ja luoda pohjaa kirjoittajan ammattitaidolle tulevana pelisuunnittelun ja 3D-grafiikan ammattilaisena.

## 2 VIRTUAALINEN TODELLISUUS SELITETTYNÄ

Viime vuosien aikana virtuaalinen todellisuus on ilmestynyt osaksi arkielämäämme. Lukuisat uutiset ja artikkelit ovat käsitelleet aihetta, pitkälti innostuneeseen sävyyn. Markkinoille on saapunut päätelaitteita, joiden avulla tavallisen kuluttajan on mahdollista päästä sisään tähän uuteen ja kiinnostavaan virtuaaliseen maailmaan.

Kuitenkaan pelkkä termin “virtuaalinen todellisuus” toisto ei varsinaisesti kerro mistä on kyse. Termi on syytä selittää ja määrittää. Koska kyseessä ei ole ontologinen teos, määritys pyritään pitämään tiiviinä ja olennaiseen keskittyvänä menemättä syvemmälle todellisuuden luonteeseen. Karkeasti sanottuna todellisuus on kaikki se mitä on olemassa (Kielitoimiston sanakirja 2017). Tämä on kuitenkin turhan laaja määritelmä. Lauri Rauhala kirjoittaa aiheesta näin: “Todellisuudesta voimme tietää vain tajunnallisen kokemuksen kautta” (Rauhala 2005). Tajunnalla viitataan tässä kokemuksen kokonaisuuteen, johon kuuluu havaintoja, mielikuvia, tunne-elämyksiä, ajatuksia ja päätelmiä (Rainio n.d.). Tiivistäen voidaan siis sanoa, että todellisuus on kaikki se, mitä pystymme kokemaan fyysisellä tasolla.

“Virtuaalinen” on onneksi helpompi määriteltävä, sillä sanana se tarkoittaa kuvitteellista, näennäistä tai keinotekoisia (Kielitoimiston sanakirja 2017). Virtuaalinen todellisuus voidaan siis määritellä aistien kautta koettavaksi keinotekoiseksi todellisuudeksi. Itsessään ajatus tällaisesta keinotekoisesta todellisuudesta ei ole uusi, mutta nopea kehitys 1900-luvun aikana on muuttanut tapaa, jolla käsitämme ilmiön. Nykyään virtuaalinen todellisuus tarkoittaa pääasiallisesti teknologisin keinoin luotua todellisuutta. Teknologiasta kumpuavaa virtuaalista todellisuutta on filosofian ulkopuolella käsitelty muun muassa Matrix-elokuvissa ja vaikkapa cyberpunk-genressä, jossa hakkerit voivat fyysisesti liikkua tietoverkkojen sisällä “kyberavaruudessa”.



Kuva 1. Damokleen miekka (Cargocollective.com)

Faktuaalisessa maailmassa eli reaalityodellisuudessa ei kuitenkaan ole mahdollista jättää kehoaan taakse eikä elää koko elämänsä virtuaalityodellisuudessa. Tällä hetkellä virtuaalityodellisuuden kokeminen perustuu päätelaitteisiin, jotka vaikuttavat vain tiettyihin aisteihin, joista tärkeimpinä näkö-, kuulo- ja tuntoaistit.

Ensimmäinen varsinainen päätelaitteen kehittivät Ivan Sutherland ja Bob Sproull vuonna 1968 (History of Virtual Reality). Kyseessä oli nykyajan näkökulmasta varsin primitiivinen ja kömpelö laite, joka oli niin painava, että se täytyi kiinnittää kattoon. Laite, nimeltään Damokleen miekka, pystyi esittämään niin sanottuja rautalankamalleja huoneista. 80- ja 90-luvuilla teknologia oli kehittynyt jo siihen pisteeseen, että kaupalliset sovellutukset olivat mahdollisia. Kuitenkaan läpimurtoa, joka olisi tuonut teknologian kaikkien saataville, ei pystytty tekemään, vaan laitteet jäivät tuolloin kuriositeetin asemaan.

2000-luvun alussa internet ja tietokoneiden kehityskulku johtivat kuitenkin fokuksen muuttumiseen. Virtuaalityodelliset maailmat siirtyivät internetiin, eikä niiden käyttämiseen vaadittu muita laitteita kuin PC tai internet-yhteydellä varustettu konsoli. Nyt kehityksen painopiste oli selkeästi immersiiivisten virtuaalityodellisten maailmojen luomisessa eikä niinkään tavassa kokea näitä maailmoja. Tänä aikana niin sanotut MMO:t, eli massiiviset, tuhansia pelaajia samanaikaisesti sisältävät nettipelit räjähtivät



kansan tietoisuuteen. Tärkeimmällä näistä, Blizzard Entertainmentin (nykyään Activision Blizzard) vuonna 2004 julkaistulla World of Warcraftilla (wow), oli aikoinaan jopa 12 miljoonaa samanaikaista kuukausittaista tilaajaa. Wow:n pelimaailma perustui muiden pelien kautta aiemmin luotuun fiktiiviseen Azerothin maailmaan, jossa pelaajahahmot toimivat suurina sankareina temaattisesti mytologiaa ja epiikkaa yhdistelevässä fantasiatarinassa. Osaltaan wow oli niin suuri menestys juuri siksi, että se pystyi käyttämään pohjanaan valmista kontekstia ja valmista brändiä esittäen sen yleisölle uudella tavalla.

Kuitenkin vuotta ennen World of Warcraftin ilmestymistä toinen yritys lähti myös luomaan omaa virtuaalimaailmaansa täysin eri lähtökohdista. Linden Labsin Second Life julkaistiin kesäkuussa 2003 ja sen perusajatuksena oli poistaa kokonaan konteksti ja antaa pelaajien luoda omansa. Second Life onkin enemmän virtuaaliympäristö kuin perinteinen peli. Siinä käyttäjille annettiin helppokäyttöiset työkalut oman sisältönsä luomiseen, oli sisältö sitten 3D-malli tai koodinpätkä. Second Lifessä ei ole yhtä yhtenäistä maailmaa, jossa kaikki sen käyttäjät toimisivat samanaikaisesti, vaan virtuaalimaailma on jaettu tontteihin, jossa jokaisella on omansa. Omalle tontilleen voi rakentaa pelin työkaluilla mitä haluaa tai ostaa pelin sisäisellä valuutalla muiden tekemiä objekteja ja koodiskriptejä. Sekä Wow:n, että Second Lifen serverit ovat vielä pystyssä ja vaikka molemmat ovat käyneet läpi lukuisia iteraatioita sekä päivityksiä, on niiden molempien kultakausi selvästi ohitse käyttäjämäärissä mitattuna.



Kuva 2. Oculus Rift (roadtovr.com)

Jos 2000-luvun alku oli virtuaalimaailmojen kompleksisuuden kehittämisen kultakautta, niin 2010-luvulla palattiin taas tutkimaan tapoja kokea näitä virtuaalimaailmoja. Syksyllä 2012 Palmer Luckey käynnisti yhteisörahoituspalvelu Kickstarterin kautta

kampanjan rahoittaakseen uuden sukupolven virtuaalitodellisuuspäätelaitteen nimeltään Oculus Rift. Kampanja oli menestys ja se johti kehitystyön kautta loppuvuonna 2016 julkaistuun kaupalliseen laitteeseen. Oculus Rift ilmestyi kuitenkin kilpailuntäyteisille markkinoille. Oculusin alkuperäisen Kickstarter-kampanjan ja kaupallisen julkaisun välillä kaksi yritystä, HTC ja Valve, yhdistivät voimansa ja loivat kilpailevan virtuaalitodellisuuspäätelaitteen nimeltään HTC Vive. Vive julkaistiin kaupallisena versiona huhtikuussa 2016.

Vaikuttaa siltä, että aikaisempi julkaisupäivämäärä on siivittänyt HTC Viven merkittävään johtoon, kun tutkitaan myytyjä kappalemääriä. Unreal-pelimootoria kehittävän Epic Gamesin perustajan Tim Sweeneyn mukaan HTC Vive on myynyt kaksinkertaisesti niin paljon yksiköjä kuin Oculus Rift (Hayden 2017). Toinen merkittävä muutos markkinoiden rakenteessa on että PC ei ole enää ainoa alusta, jolle on VR-päätelaitteita. Sony julkaisi vuonna 2016 oman PSVR-laitteensa käytettäväksi Playstation 4 -konsolin kanssa. Myöskään Microsoft ei ole jäänyt lepäämään laakereillaan, vaan yritys on julkaisemassa lähitulevaisuudessa omaa päätelaitettaan, Windows 10 VR:ää. Tämän lisäksi lukuisat aasialaiset laitevalmistajat ovat tuomassa tai tuoneet markkinoille omia päätelaitteitaan. Vaikka tulevaisuutta on vaikeaa ennustaa, niin tämänhetkiset trendit näyttävät viittaavan siihen suuntaan, että elämme aikakautta, jossa VR viimein on lyönyt itsensä siinä määrin läpi, että teknologian jatkokehitys tulevaisuudessa on varmallalla pohjalla.



Kuva 3. HTC Vive (arstechnica.com)

### 3 MITEN VR-PELI EROAA TAVALLISESTA PELISTÄ?

Ennen VR:ää kuluttajilla on ollut käytännössä kolme erilaista tapaa pelata videopelejä. Kotitietokoneet, eli PC, konsolit ja mobiili ovat kaikki alustoja, joilla voi pelata. Kaikissa kolmessa on näyttöpäätte ja ohjauslaite, joka on joko erillinen tai integroitu osaksi laitetta. Vaikka tietokoneissa pääasiallinen ohjauslaite on aina ollut näppäimistö ja hiiri, on ollut jo pitkään mahdollista käyttää konsolin ohjaimia PC -pelaamisessa. PC:lle on ollut jo vuosikymmenien ajan myös muita ohjauslaitteita, kuten joystickejä, ajopelejä varten tehtyjä ratteja ja vielä esoteerisempia laitteita, kuten TrackIR (Kauppinen, J.O., 2015.) joka seuraa pään liikkeitä ja siirtää sen mukaan pelin sisäistä näkymää. Hyötyä TrackIR:stä on silloin, jos omistaa enemmän kuin yhden näytön, sillä katseen kääntäminen vaatii myös katselualaa. Myös konsoleille on olemassa näppäimistöjä ja hiiriä, joten selkeät erot laitealustojen välillä ovat nykyään vähemmän selkeitä. Mobiilialustat ovat yleisesti ottaen niitä, joissa ohjainlaite on rakennettu laitteeseen sisälle. Suurinta osaa tavallisista mobiilipeleistä ohjataan laitteen ruudulle piirrettyjen ohjausmetodien, kuten nuolien ja virtuaalisten nappien avulla.

VR -pääte-laite ei kuitenkaan ole uusi alusta siinä mielessä, että se korvaisi konsolin tai kotitietokoneen ja olisi täten näistä erillinen, yksikseen toimiva laite. Tietystä näkökulmasta katsottuna VR onkin vain lisälaite olemassa oleville laitealustoille, joka mahdollistaa pelien pelaamisen uudella tavalla. Tarkemmin sanottuna, suurin osa perinteisistä ohjauslisälaiteista vaikuttavat vain siihen kuinka peliä ohjataan standardilla näyttöpäätteellä. Joystick luo immersiiivisen kokemuksen lentokoneen tai avaruushävittäjän ohjaamisesta, mutta ei vaikuta mitenkään siihen mitä ruudulla visuaalisesti tapahtuu, jos sitä verrataan vaikka hiireen. TrackIR on siten lähempänä VR:n toimintaperiaatteita: Omilla liikkeillä on mahdollista muuttaa pelin sisäistä näkymää, mutta toisin kuin TrackIR, VR toimii myös ruutuna, eikä vain tavallisen näyttöpäätteen tehosteena.

Voidaan sanoa, että VR on tässä mielessä hybridilisälaite. Se muuttaa tavan nähdä ja kokea näyttöpäätteen antaman informaation ja tämän lisäksi antaa molemmille käsille

ohjainlisälaitteen, joiden toimintaperiaate eroaa tavallisista ohjauslisälaitteista. Lähin mahdollinen vertailupinta kahden suosituimman VR-päätelaitteen, Riftin ja Viven, ohjainlaitteiden ja perinteisten ohjainten välillä löytyy Nintendo Wii -konsolista, jonka Wiimote -ohjain on toimintaperiaatteeltaan hyvin lähellä VR-ohjainta. Molemmat tunnistavat ohjaimen osoitus suunnan ja pystyvät jäljittämään tämän fyysistä sijaintia reaali maailmassa. Tähän väliin on hyvä huomauttaa, että mainituista samankaltaisuuksista huolimatta VR -ohjainlaitteille ei ole olemassa standardia eri valmistajien välillä, vaan esimerkiksi Oculus Riftin ja HTC Viven:n ohjaimet eroavat toisistaan nappien ja padien määrien ja laadun suhteen. Myöskään kaikki valmistajat eivät myy tällaisia ”bundleja”, joissa saa VR-lasit ja -ohjaimet samassa paketissa. Kiinalaiset valmistajat myyvät lähinnä pelkkiä VR-laseja, jolloin ohjainlaitteiden hankinta täytyy suorittaa erikseen. Lisäksi esimerkiksi mobiili-VR on toiminut pitkälti niin, että mobiililaitte asetetaan ”aitoja” VR-laseja muistuttavaan koteloon, joka asetetaan silmien eteen. Tällöin kotelossa sijaitsevat ainoastaan linssit ja niiden säätömekanismi. Tästä esimerkkinä toimii kirjaimellinen pahvilaatikkomalli, Google Cardboard ja Samsungin oma GearVR. Näissä vaihtoehdoissa ainoa ohjausmetodi on pään liike, jota mitataan koteloon laitettujen älypuhelimien gyroskoopeilla. Google Daydream View sisältää sekin tällaisen kotelon, mutta sen mukana tulee myös kolminappinen ohjain, joka muistuttaa ulkonäöltään pientä kauko-ohjainta. Kuitenkaan mistään varsinaisesta peliohjaimesta ei voida tässä yhteydessä puhua, vaan Daydreamin ohjaimen sovellutukset ovat pitkälti applikaatioiden ohjaamisessa.

#### **4 MIKSI VALITA VR ALUSTAKSI?**

Ennusteiden perusteella vuoden 2021 aikana tullaan rikkomaan 100 miljoonan myydyn VR-, sekä AR-päätelaitteen (Augmented Reality) määrä, mikäli myyntiä seurataan vuositasolla (Statista. 2017.). Kyseessä on siis kasvava markkina, jolla on vähemmän tungosta, kuin perinteisillä videopelimarkkinoilla. Esimerkiksi johtavassa pelien verkkomyyntipalvelussa Steamissa on kirjoitushetkellä yli 19,000 (yhdeksätoistatuhatta) peliä myynnissä (Steamdb. 2017.) joten uuden, tuntemattoman, vaikka vähän pienemmän yrityksen peli saattaa yksinkertaisesti jäädä huomaamatta. Toisaalta, saman palvelun VR-portaalissa on huomattavasti vähemmän pelejä, joten huomatuksi tuleminen siellä on paljon helpompaa. On myös totta, ettei VR-alustalle ole

vielä julkaistu ns. “killer appia”, eli peliä tai ohjelmistoa, jonka pelkkä olemassaolo myisi laitteita (Lang, B., 2017.) vaikkakin kohtuullisen hyvin menestyneitä pelejä onkin jo julkaistu, joista esimerkkinä miljoonamyynnteihin yltänyt Job Simulator (Donnelly, J., 2017.).

Voidaankin sanoa, että VR on vielä uutta ja tutkimatonta aluetta, joten yleinen innostus tarttua alustaan kehittäjän näkökulmasta on verrattavissa kultaryntäykseen. Vaikka tällä hetkellä ei enää puhutakaan “ensimmäisestä aallosta”, eli peleistä jotka julkaistiin samaan aikaan VR-päätelaitteiden kanssa, on teknologian parissa mahdollista innovoida suuresti. Innovaation lähteenä toimii se etteivät VR-pelit vielä täysin pysty käyttämään alustan luonnollisia etuja hyväkseen. Suuri osa peleistä on näet ns. “tavallisia” pelejä, joihin on lisätty tuki VR-laseille, tai pelikonsepteja, jotka toistavat vanhoja malleja ottamatta huomioon ympäristön ja ohjaustavan erilaisuutta. Täten pelkkä mahdollisuus luoda itse jotain uutta tai määrittää tulevia konventioita teknologian hyödyntämisen, kuten esimerkiksi VR:n sisällä liikkumisen suhteen, on monelle kehittäjälle herkullinen ajatus. Voidaan sanoa, että VR on varsinkin indie-kehittäjälle herkullinen aluevaltaus, sillä itse rahoitetuilla, pienen budjetin kokeellisilla projekteilla on potentiaalia luoda uusia innovaatioita. Tällöin ei myöskään ole tarvetta tyydyttää vaikkapa perinteisessä julkaisuprosessissa läsnä olevaa rahoittajaa. (Johnson, J., 2017.).

## 5 MITEN SUUNNITELLA TASO VR-PELIIN

### 5.1 Asiakkaan lähtökohta



Kuva 4. Outrageous Grounds: The Maze logo (Virtual Dawn 2017)

Toimeksiantaja Virtual Dawn / Midnight Forge on työstänyt Outrageous Grounds: The Maze -peliä jo noin vuoden ajan kirjoittajan aloittaessa opinnäytetyöprojektinsa vuoden 2017 tammikuussa. Outrageous Grounds on ensimmäisen persoonan räiskintä- / seikkailupeli-hybridi, joka on kehitetty yksinomaan VR-laitteille. Kohdealustoina toimivat kaksi suosituinta PC-pohjaista VR-laitetta: HTC Vive, sekä Oculus Rift. Peliä myydään tällä hetkellä verkkopohjaisessa Steam-pelikaupassa Early Access -nimikkeellä. Tämä siis tarkoittaa, että peliä ei julkaista valmiina, vaan asiakkaille annetaan mahdollisuus seurata ja testata peliä sen kehityksen aikana. Tällöin asiakkaat pystyvät esimerkiksi tarjoamaan palautetta missä pelinkehittäjät ovat onnistuneet tai epäonnistuneet jo varhaisessa vaiheessa kehitysprosessia.

Outrageous Grounds: The Maze on juonivetoinen peli. Pelin tarina kertoo pahan ylikansallisen korporaaation kehittämästä teknologiasta, jonka avulla on mahdollista tuoda virtuaalitodellisuudesta teknologiaa ja esineitä reaali maailmaan. Tämä teknologinen etu on antanut korporaatille suuren määrän valtaa ja lähes rajattomat resurssit. Pelihahmon on käytävä itse lukuisissa erilaisissa pelimaailmoissa, jotta hän voi materianmuuntoteknologiaa hyväksi käyttäen tuoda reaali maailmaan erinäisiä esineitä, jotka auttavat päihittämään edellä mainitun korporaaation. Käytännössä Outrageous Groundsin pelaajahahmo menee siis pelin sisällä vielä muiden pelien sisälle saavuttaakseen tavoitteensa.



Kuva 5. Pelikuvaa Outrageous Grounds: The Mazesta (Virtual Dawn, 2016).

Pelimekaanisesti Outrageous Grounds sisältää kolme tärkeää elementtiä. Ensimmäinen on vihollisten päihittäminen kentistä löytyvällä aseistuksella. Aseita on useita erilaisia ja kaikki muut paitsi pistooli tarvitsevat kentistä löydettäviä ammuksia, eli rajallista resurssia. Pelaaja voi halutessaan käyttää aseita molemmissa käsissä, eli hän voi esimerkiksi käyttää samaan aikaan kahta pistoolia tai vaikkapa haulikkoa ja pistoolia. Aseiden lisäksi on mahdollista löytää ja käyttää myös kahta erilaista kranaattityyppiä. Toinen tärkeä pelillinen elementti on tavaratilan rajallisuus. Pelaaja voi kantaa mukanaan molemmissa käsissään tavaroita. Käsien lisäksi hänellä on kuusi ”taskua”, joihin tavaroita voi asettaa talteen. Tämä rajallisuus luo dilemman, jossa pelaajan on päätettävä onko hänelle tärkeämpää pitää mukanaan kattavaa asearsenaalia, kranaattikimppua tai mahdollisesti useampia avainkortteja, eli esineitä jotka mahdollistavat pelaajan etenemisen pelin tasoissa.

Kolmas iso elementti pelissä on vapaa liikkuvuus. Pelaaja kulkee pelin tasoissa eteenpäin käyttäen peliä varta vasten suunniteltua liikkumisjärjestelmää, joka on kehitetty pahoinvoinnin välttämistä silmällä pitäen. Pelaaja ei kuitenkaan liiku ainoastaan horisontaalisesti, sillä pelaaja tulee pelin aikana kohtaamaan useita kohtia pelin tasoissa, joissa hän ei pysty jatkamaan etenemistä tavallisen liikkumisjärjestelmän puitteissa. Tällöin pelaajan on kiivettävä näiden esteiden yli ensimmäisessä persoonassa tapahtuvan kiipeilymekaniikan avulla.



Kuva 6. Pelikuvaa Outrageous Grounds: The Mazesta (Virtual Dawn, 2016).

Outrageous Grounds - The Maze on lineaarinen peli siinä mielessä, että jokaisella kentällä on alkupiste ja selkeä tavoite, joka pitää saavuttaa, jotta pelissä voi edetä. Täten kentät eivät ole niinkään leveitä, kuin pitkiä kokonaisuuksia. Kenttien itsensä sisällä on kuitenkin useita salaisuuksia ja bonuksia, joilla palkitaan niitä pelaajia, jotka kääntävät jokaisen kannon ja kolon.

Asiakkaan lähtökohta opinnäytetyön pohjalla toimivalle työurakalle oli seuraavanlainen: Tarkoituksena oli suunnitella ja toteuttaa Outrageous Grounds: The Maze -peliin sisältöä kokonaisen pelitason verran. Tarkemmin eriteltynä, tarkoituksena oli työstää asiakasyrityksen tarjoama raakakonsepti valmiiksi kokonaisuudeksi, jossa tekoprosessiin kuuluu tasosuunnittelua, 3D-mallintamista pelituotannolle perinteisellä tekotavalla, eli ns. "3D production pipeline" sovellettuna asiakasyrityksen sisäisten käytäntöjen, prosessien ja kapasiteetin mukaan. Tämän lisäksi luodut assetit täytyi kokoonpanna Unity-pelimoottorissa ja asetella heidän standardihierarkiaansa, jotta asiakasyrityksen muut työntekijät pystyvät prosessin päätyttyä jatkamaan tason kehittämistä julkaisuvalmiiksi kokonaisuudeksi mm. graafisten efektien, valaistuksen viimeistelyn ja pelitason sisäisten tavoitteiden hienosäädön kautta.



## **5.2 Käytettävä teknologia**

### **5.2.1 Unity**

Outrageous Grounds: The Mazea kehitetään Unity-pelimoottorilla. Unity on monialustainen pelimoottori, jota Unity Technologies on kehittänyt vuodesta 2005 asti. Sen kohdealustoja ovat PC, konsolit, mobiililaitteet ja nettisivut. Kehitysversion rakentaminen eri alustoille on tehty vaivattomaksi. Unity on muutenkin suosittu vaihtoehto alhaisen oppimiskynnyksensä ja ilmaisuutensa takia. Koodin puolesta Unity käyttää skripteissään C#-kieltä ja javascriptiä. Tämän lisäksi pelimoottoriin on integroitu kaappasovellus, Unity Store, josta voi rahalla hankkia 3D-malleja, tekstuureja ja lisäosia käytettäväksi moottorin sisällä.

Asiakasyrityksen peli käyttääkin tällaisia lisäosia, sillä ne nopeuttavat pelinkehitystä tarjoamalla valmiita ratkaisuja sen sijaan, että kaikki tehtäisiin alusta alkaen itse. Esimerkiksi SECTR on lisäosa, joka mahdollistaa pelimoottorin sisällä olevan pelitason rikkomisen osiin eli sektoreihin, jotka voidaan aktivoida ja deaktivoida tarpeen mukaisesti. Käytännössä tällä tavalla voidaan ennakoidusti, pelaajan etenemisen mukaan järkevöittää taustalla tapahtuvien alueiden, objektien, skriptien ja vihollisten latausta muistiin, jotta pelin performanssi ei kärsi. Lisäosilla ei ole suurelta osin merkitystä opinnäytetyön kannalta, paitsi kokoonpanossa Unityssa.

### **5.2.2 Maya**

Autodesk Maya on alunperin Alias Systems Corporationin ja myöhemmin Autodeskin kehittämä ja julkaisema 3D-mallinnus -ohjelma, joka julkaistiin alunperin helmikuussa 1998. Mayaa pidetään pitkälti standardina, ohjelmana johon muita verrataan. Tämä johtuu paitsi ohjelman pitkäikäisyydestä, niin myös sen kattavasta työkaluvalikoimasta. Vaikkakin Mayaa käytetään pääasiassa elokuvien ja muun, perinteisen, visuaalisen median erikoistehosteiden luomiseen, käytetään sitä myös pelialalla. Mayan parhaita puolia on sen skriptikieli MEL, jolla on mahdollista luoda omia työkaluja (Pluralsight, 2015.).

### 5.2.3 xNormal

XNormal on ilmainen ohjelmisto, jota käytetään apuna 3D-mallien tekstuurien tuottamisessa. Ohjelmalla on mahdollista luoda automaattisesti tai asetuksia hienosäätämällä 3D-malleja varten normal-, ambient occlusion-, että displacement -karttoja, joita käytetään yksityiskohtien esiin tuomiseen. Ohjelmalla on myös mahdollista verrata keskenään saman 3D-mallin korkearesoluutioista versiota matalaresoluutioiseen malliin ja luoda näiden erojen peruusteella tarvitut tekstuurikartat.

### 5.2.4 Quixel Suite 2

Quixel Suite on lisäosa Adobe Photoshopiin, jolla teksturoidaan muutoin valmiita 3D-malleja. Lisäosa käynnistyy samaan aikaan Photoshopin kanssa ja lisää ohjelman päälle omia käyttöliittymävalikoitaan. Teksturointiin vaaditaan 3D-malli, xNormalilla luodut tekstuurikartat ja ns. "Material ID" -kartta, joka määrittää ohjelmalle teksturoitavat alueet 3D-mallissa. Lähtökohtaisesti ohjelmalla on mahdollista tehdä vähäiselläkin kokemuksella näyttävän näköisiä tekstuureja. Kokeneemmat käyttäjät pystyvät myös luomaan ohjelman sisällä kokonaan omia tekstuurimateriaalejaan valmiiden käyttämisen sijaan. Nämä ominaisuudet tekevät Quixel Suite:sta monipuolisen ja tehokkaan työkalun 3D-mallien tuotantolinjaan.

## 5.3 Rajoitteet

Suurin rajoittava tekijä pelin kehityksessä ja opinnäytetyöprojektissa on henkilöstöresurssien vähyys. Koska peliä aktiivisesti kehittävä tiimi on nelihenkinen kirjoittaja pois lukien, on selvää, että jokaisen tiimin jäsenen on pystyttävä työskentelemään oman mukavuus- ja osaamisalueensa ulkopuolella ja omaksuttava rooleja kehitysprosessissa, joista ei välttämättä ole kokemusta. Asiakasyrityksellä ei esimerkiksi ollut vakituista konseptitaiteilijaa, jonka visuaalisten tuotosten rakentaman konseptuaalisen peruskiven avulla tason olisi voinut rakentaa, vaan tämäkin osaprosessi täytyi tuottaa alusta asti itse.

Toinen rajoittava tekijä on aika. Koska kirjoittajan harjoittelu asiakasyrityksessä kesti vain kolme kuukautta, oli tehtävä työ pidettävä suhteellisen kompaktina kokonaisuutena tai riskeerata sen jääminen puolivalmiiksi, epäkoherentiksi, rasisiteeksi, josta ei olisi asiakkaalle taloudellista hyötyä.

Kolmas rajoittava tekijä on asiakasyrityksen ohjelmistolisenssien laatu. Ei voida lähtökohtaisesti olettaa, että kehitystyötä päästään tekemään aina itselle tutuilla ohjelmistoilla tai tekotavoilla, vaan on pärjättävä jälleen kerran niillä resursseilla, joita on mahdollista käyttää. Esimerkiksi kirjoittajalla on entuudestaan kokemusta 3D-mallien teksturoinnista käyttäen Allegorithmicin Substance Painteria, kun taas asiakasyrityksen 3D -tuotantoputkeen käytetään Quixel Suitea.

Neljänneksi, koska VR vaatii ruudunpäivitysnopeuden 90 toimiakseen sulavasti ja pysyäkseen immersiivisenä, on toteutettavan työkokonaisuuden oltava optimoitu ja rakennettu niin ettei esimerkiksi kentän sisäisten alueiden välillä liikkuminen aiheuta hyytymisiä pelin pyörimisnopeuteen. Pidimme tavoitteena, että kenttä sisältäisi kokonaisuudessaan alle 200,000 polygonia, mutta ettei se kuitenkaan olisi mikään tavoiteluku.

## **5.4           Konsepti**

Asiakasyrityksen alkuperäinen, idean tasolla oleva, konsepti luotavasta pelitasosta oli yksikerroksinen kauppa, joka toteutettaisiin temaattisesti ja visuaalisesti muistuttamaan keskiajasta vaikutteita lainaavaa fantasiaympäristöä. Tämä kauppa olisi täynnä kaikenlaisia pienesineitä ja pelaajan ratkaistavia pulmia. Taso toimisi vastapainona muille pelin tasoille, jotka ovat toimintapainotteisempia, antaen pelaajalle mahdollisuuden vetää henkeä ja tutkia oma-aloitteisesti ympäristöä vailla kiirettä.



Kuva 7. Raakakonsepti pelikentälle

Asiakasyrityksen edustaja antoi tarkoituksella mahdollisimman kompaktin ja suppean ideatason konseptin tason toteutukseen ja kirjoittajalle annettiin huomattavan paljon vapautta tason lopullisen muodon ja tarkan tyylin toteutuksen suhteen.

Kuten kuvasta huomaa, sijaitsee alkuperäisen konseptin kauppa puunrunkon sisällä. Siellä on tiski, hyllyjä, soittimia ja portaat ylös. Näissä ylemmissä kerroksissa tuli alun perin olla kauppiaan asuintilat. Visuaalinen ilme, jota alkuperäisessä konseptissa tavoiteltiin asiakasyrityksen tiimoilta, oli fantasiatyylinen ja kaupan sisätiloja varten haettiin innoitusta Tolkienin hobintinkoloista. Kuitenkin kirjoittajalle sanottiin, että konseptia voi vapaasti muuttaa, mikäli saa ideoita.

Näistä lähtökohdista kirjoittaja sitten lähti työstämään kenttää. Hyvin nopeasti tuli huomattua, että käytettävissä olevat resurssit mahdollistavat laajemman kokonaisuuden työstämisen. Alkuperäinen konsepti puun sisällä olevasta kaupasta kasvoi neljäkerroksiseksi kokonaisuudeksi, joka sijaitsee massiivisen onton puun sisällä. Kaupan keskellä on näyttämö, jonka edessä on tilaa katsojille. Onton puurungon yläosassa on massiivinen reikä, josta loistaa valoa näyttämölle, toimien “luonnollisena” näyttämövalona. Näyttämön päällä sijaitsee soittimia, joita pelaaja voi soittaa. Näyttämön ympäri kiertää puoliympyrän muotoisia porraskokonaisuuksia, jotka johdattavat pelaajaa yhä ylemmäs kaupan sisällä. Jokaisessa kerroksessa on lukuisia

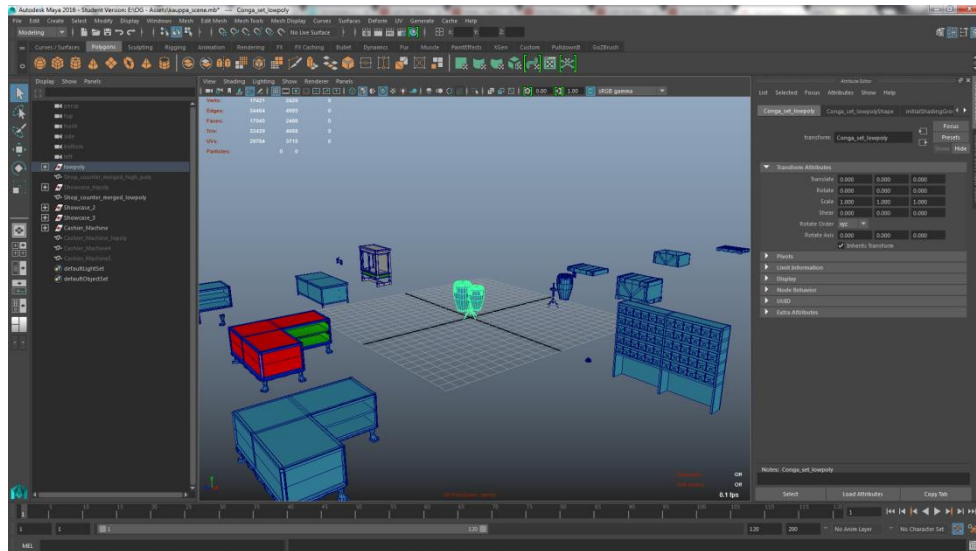
hyllyjä, pedestaaleja ja vitriinejä, jotka ovat kuitenkin pääosin tyhjiä. Kaupassa ei näet ole käynyt aikoihin asiakkaita, joten sen hyllyillä on enää muutamia esineitä jäljellä. Kiitos keskuksena toimivan näyttämön pelaajalla on kaiken aikaa kiintopiste, johon hän pystyy vertaamaan sijaintiaan. Lisäksi kauppa on tilallisesti hyvin avoin, joten sijainnin määrittämisen ei pitäisi tuottaa hankaluuksia. Tämän lisäksi kaupan ensimmäisellä tasolla on luukku, joka johtaa kellarikerrokseen. Kellari sijaitsee puun alla kiviseinäisessä luolassa ja sen seinämiä vasten heijastuu vain neonväristen sienien tuottama valo. Edetäkseen pelaajan on käytävä luolastossa ja palattava salaista reittiä pitkin takaisin kauppaan. Päästessään kaupan myöhemmin kaupan korkeimpaan kerrokseen, huomaa pelaaja, että eräs puunrungon oksista on ontto. Seurattuaan hetken aikaa oksaa, kohtaa pelaaja siinä ulos johtavan reiän.

Ulkona odottaa keskiaikaistyylinen fantastinen linnamaisema, jossa on kymmeniä metrejä korkeita muureja ja massiivista arkkitehtuuria. Käy ilmi, että puun ulkokuori on kasvanut läpi useista muureista ja sen lonkeromaiset juuret ovat kietoneet itsensä kivisen arkkitehtuurin ympärille. Pelaajan täytyy tässä maisemassa löytää tiensä areenalle, jossa hän taistelee pomovihollista vastaan. Tämän jälkeen hänen tulee navigoida kalliojyrkänteiden ja sen alla odottavan meren ylle rakennettua, silloista ja torneista koostuvaa kokonaisuutta. Kiivettyään hyvän aikaa ylempänä sijaitsevaa puun oksaa kohti, pelaaja voi tiputtautua takaisin kauppaan, josta hän pääsee loppuinteraktioiden jälkeen lopettamaan tason.

Kuten selityksestä huomaa, lopullinen konsepti ja tuote ovat siis useita kymmeniä kertoja suurempi kokonaisuus, kuin alkuperäinen konsepti. Se sijoittuu kahteen tyyllillisesti erilaiseen maisemaan, jotka kuitenkin kietoutuvat yhteen massiivisen kiintopisteen, eli jättiläispuun kautta. Ihmisten tekemät rakenteet ovat jännitettä tuova vastakohta puun orgaanisille muodoille (Bradley, S., 2012.). Ajatus on luoda tätä kautta tunne vanhasta, mystisestä paikasta, jonka alkuperäiset asukkaat ovat aikoja sitten kadonneet jonnekin ja jäljellä on vain kaupungin hidas rapistuminen, kunnes mitään ei ole enää jäljellä. Tason lopullista konseptikokonaisuutta ei ole kuvattu missään, sillä se rakennettiin Maya-ohjelman sisällä primitiivisistä muodoista, joita tarkennettiin prosessin aikana. Liitteessä 1 on esitelty kuvia, joiden pitäisi tukea visuaalisesti äsken selitettyä konseptia.

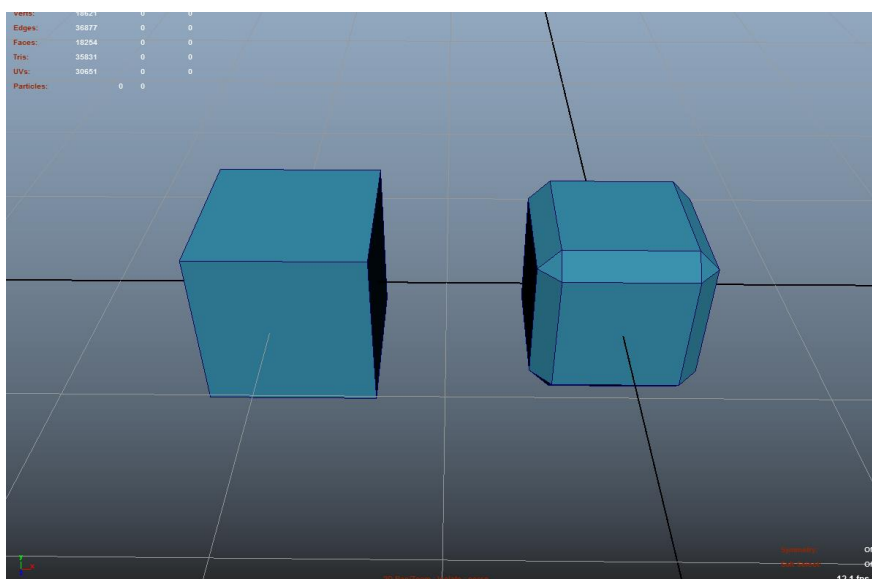
## 6 MITEN LUODA VR-PELIN TASO VISUAALISESTI

### 6.1 Kuinka luoda tarvittavat 3D-mallit



Kuva 8: Autodesk Mayan käyttöliittymä

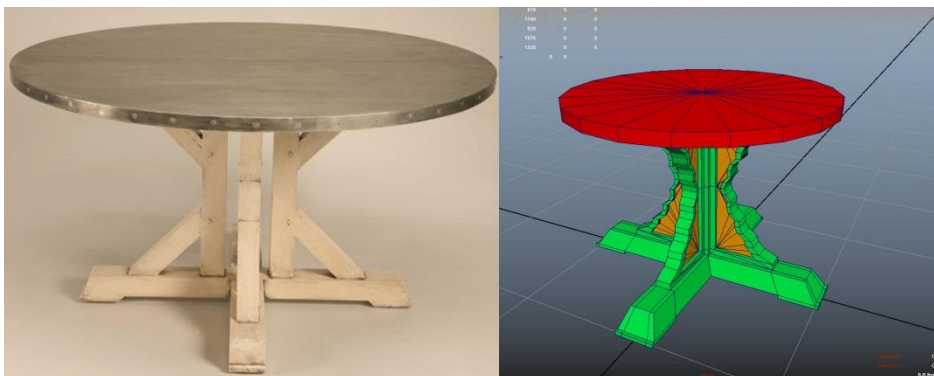
Tässä luvussa käydään lävitse 3D-mallien tuottaminen konseptivaiheesta valmiiksi asseteiksi. Kun tekstissä mainitaan jonkin tietyn työkalun käyttö työvaiheessa, sitä seuraa sulkumerkkien sisällä sen sijainti ohjelman sisällä. Esimerkiksi jos haluttaisiin käyttää Bevel-toimintoa, merkittäisiin sen sijainti seuraavasti: (Edit Mesh -> Bevel). Tätä käytäntöä käytetään ohjelmasta riippumatta. Maya-ohjelmassa voi myös joidenkin toimintojen kohdalla olla pieni laatikon muotoinen ikoni. Tämä tarkoittaa, että valitsemalla laatikon pääsee muokkaamaan työkaluun liittyviä lisäasetuksia. On kuitenkin huomautettava, ettei tämä opinnäytetyö pyri olemaan opas näiden ohjelmien käyttöön. Kaikkia ominaisuuksia ei täten käsitellä, vaan pelkästään tehdyille työprosessille olennaiset.



Kuva 9: Oikeanpuolimmaiseen kuutioon on käytetty Bevel-toimintoa

## 6.2 3D-mallin konseptointi

Jokaisen 3D-mallin meshin luonti lähti samasta pisteestä. Koska asiakasyrityksellä ei ollut käytettävissään konseptiartistia, tuli 3D-mallintajan luoda konseptikin itse. Käytetyimmät työkalut olivat tässä tapauksessa googlen kuvahaku ja tavallinen lyijykynä ja paperi. Googlen kuvahakua käytettiin eniten huonekalujen mallintamisessa, sillä temaattisesti sopivia huonekaluja oli helppo löytää valituilla hakusanoilla. Kuitenkin tässäkin piilee riski, sillä laillisten syiden takia kuvissa olleita kohteita, huonekaluja ja pienesineitä ei voinut mallintaa yksi yhteen kuvan kanssa. Täten Googlen kuvahaun tuloksista valittiin osuvimmat tulokset joiden perusmuodot tai siluetti näyttivät hyvältä ja niitä jatkokehitettiin tavoilla, jotka saivat luomuksen näyttämään omalta. Googlen kuvahaun lisäksi myös Pinterest-palvelusta on mahdollista löytää aihepiireittäin kuvia konseptoinnin avuksi. Artstation-palvelu taas tarjoaa mahdollisuuden tutkia muiden tekemää taidetta ja malleja, joista on myös mahdollista hakea innoitusta. Käytännössä ei kuitenkaan ollut mitään ohjenuoraa, lukuun ottamatta asiakasyrityksen raakaa ideaa tason teeman suhteen, että mikä olisi sopiva lähtökohta mallille, vaan mallintajan täytyi luottaa omaan silmäänsä ja makuunsa. Lyijykynään ja paperiin turvauduttiin tapauksissa, joissa pelkkä kuva ei riittänyt alkumalliksi ja jotain täytyi selventää itselleen ennen mallintamisen aloittamista.



Kuva 10: Vasemmalla konseptikuva (decoadvisor.net), oikealla siitä johdettu malli.

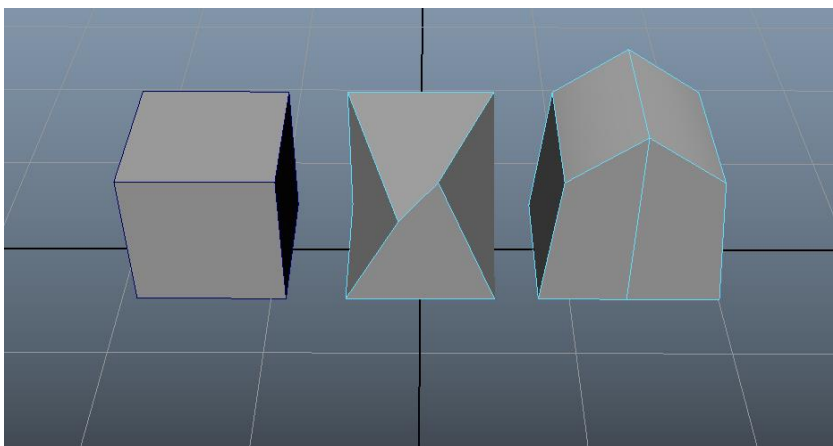
Luotu taso rakennettiin lähes kokonaisuudessaan Mayan sisällä, josta se sitten myöhemmin tuotiin ulos ja siirrettiin tekstuurien luomisen jälkeen Unity-pelimoottoriin. Vaikkakin työvaiheen olisi voinut myös hoitaa niin, että tehty palanen tuodaan Unityn sisälle jolloin raaka-asettelu ja tason luominen tapahtuisi Unityssä, niin kirjoittaja koki valitun lähestymistavan paremmaksi. Syitä tähän on lukuisia. Mayan sisällä on helppoa luoda geometriaa ja esineitä niin, etteivät niiden väliset mittasuhteet rikkoudu, vaan kaikki pysyy suhteellisesti halutun kokoisena. Toisekseen, lähestymistapa on kirjoittajalle itselleen nopeampi ja helppokäyttöisempi, sillä uuden 3D-geometrian luonti ja muokkaus tapahtuu Mayan sisällä ilman tarvetta tuoda 3D-objektia ulos ohjelmasta ja sisään toiseen ohjelmaan ainoastaan muutamaa tarkistusta varten.

### 6.3 3D-mallin geometrian luonti

Kaikki staattinen kenttägeometria, huonekalut ja pienemmät liikuteltavat objektit luotiin käyttäen Autodesk Mayaa. 3D-mallin meshin luomisen jälkeen käytettiin muita ohjelmia tekstuurien luomista varten. Nämä vaiheet käydään lävitse myöhemmin, fokuksen ollessa nyt itse mallintamisessa. Metodi, jolla malleja luotiin oli suurimmassa osassa tapauksia ns. “Subdivision modelling”. Helpoin tapa selittää prosessi on ajatella mielessään kuutio. Kuutiossa on kuusi sivua ja kahdeksan kulmaa. 3D-mallinnuksen termeissä sivut ovat “faceja” ja kulmat ovat “verteksejä”. Verteksit luovat muodon ja facet muodostavat pinnan. Verteksien välillä kulkevat linjat, jotka yhdistävät ne keskenään ovat nimeltään “edgejä”. Alla olevassa kuvassa on esiteltynä miten nämä



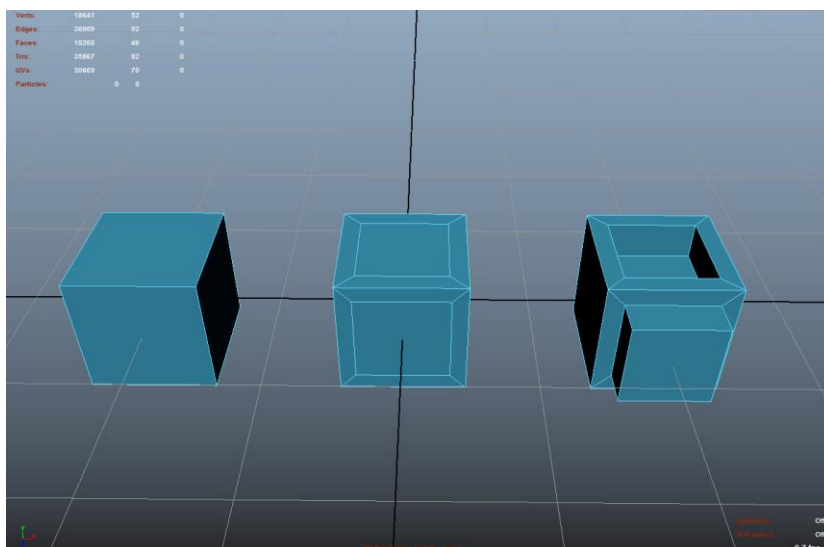
toimivat keskenään. Vasemmalla puolella on tavallinen kuutio, jota ei ole muokattu mitenkään. Keskimmäinen objekti oli alun perin kuutio, mutta kun sen verteksejä siirreltiin, sen muoto muuttui. Oikeanpuolimmainen objekti oli myös alunperin kuutio, mutta sen läpi kulkee ylimääräinen edge, l. “edge loop” (Mesh Tools -> Insert Edge Loop) ja tämän johdosta se sisältää neljä verteksiä enemmän, kuin muut kuutiot.



Kuva 11: Verteksin ja edgejen esittelyä

Teoriassa se miten subdivision modelling toimii on, että yksinkertaisista primitiiviobjekteista luodaan monimutkaisia 3D-malleja lisäämällä ja siirtelemällä verteksejä, jotta saavutetaan haluttu muoto. Esimerkiksi oikeanpuoleisin kuutio voisi jatkokehittelynä olla vaikka talo. Siinä on nähtävissä harjakattoa muistuttava muoto, josta olisi helppoa jatkaa verteksejä lisäämällä valmiiksi taloksi. Käytännössä mallintaminen on kuitenkin paljon monimutkaisempi prosessi, kuin edellämainittu antaa olettaa. Jokaisen mallin, eli 3D-objektin luomisen vaikeus riippuu vaaditusta detaljitasosta, jossa yksityiskohtaisempi on vaikeampi.

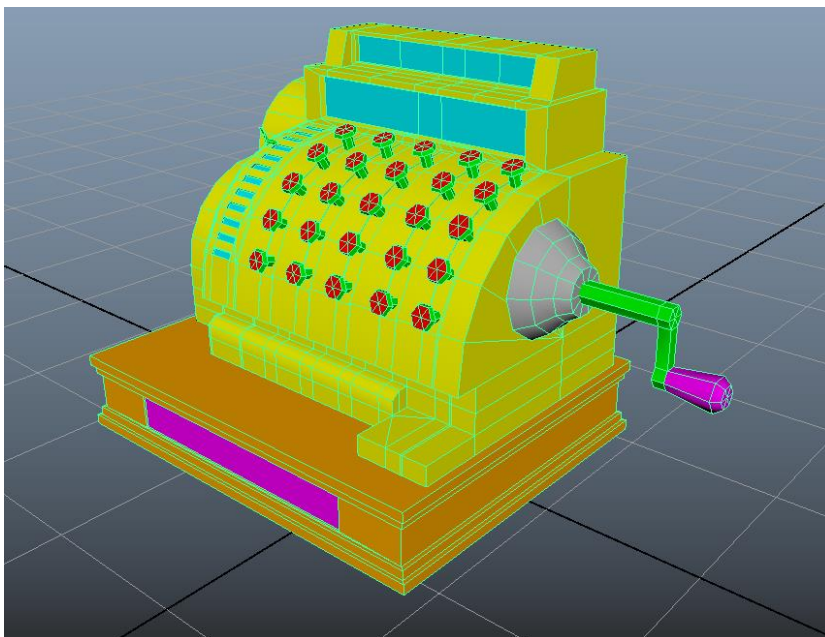
Kuvassa 13 on kassakone, joka tuotettiin yhtenä Outrageous Grounds: The Maze:en tehdyistä malleista. Kuten kuvasta huomaa, on malli paljon kompleksisempi, kuin aiemmassa kuvassa nähdyt neliöt. Kuitenkin se noudattaa täysin samoja periaatteita mallinnuksen suhteen, kuin mitkä esiteltiin aiemmin. Lisäämällä edge loopeja ja käyttämällä “extrude” -toimintoa, l. sisennys- tai ulonnus- komentoa (Edit Mesh -> Extrude) , on mahdollista luoda monimutkaisia muotoja. Verteksin lisäksi on myös mahdollista valita kokonaisia edge loopeja ja faceja, joita voi skaalata tai kääntää Mayan manipulaatiotyökaluilla (pikanäppäimet Q,W,E,R).



Kuva 12. Extrude-toiminnon esittelyä. Oikeanpuolimmainen muoto vaatii kaksi extrudea.

Eräs merkillepantava asia on, että kassakonetta ei ole mallinnettu suoraan yhdestä primitiivistä, vaan useista osista. Mikäli malli siirrettäisiin tässä vaiheessa esimerkiksi Unityyn, pelimoottorin sisälle, olisi jokainen osan oma erillinen objektinsa Unityn käyttämässä hierarkiassa. Tämä ei ole toivottua, joten tämän vuoksi, ennen tätä vaihetta, täytyy käyttää Mayan Combine -toimintoa. Tämä toiminto yhdistää jokaisen erillisen osan yhdeksi kokonaisuudeksi, jolloin sitä on helpompi käsitellä pelimoottorin sisällä. On kuitenkin joitain tapauksia, joissa koko objektin kaikkia osia ei ole soveliasta yhdistää yhdeksi kokonaisuudeksi, kuten jos objekti halutaan vaikka animoida. Esimerkiksi kassakoneen oikealla puolella oleva kampi voisi hyvin olla erillisenä objektina ja se voitaisiin animoida pyörimään.

Toinen merkillepantava asia on kassakoneen värit. Se palvelee kahta tarkoitusta. Ensiksikin se tuo paremmin esille visuaalisessa mielessä mallin monimutkaisuuden kuvassa, sillä mikäli se olisi harmaa ja yksivärinen, kuten aiemmin nähdyt kuutiot, olisi sen muodoista vaikea saada selkoa. Toinen, pääasiallinen tarkoitus, liittyy mallin teksturointiin, josta enemmän asiaankuuluvassa osiossa.

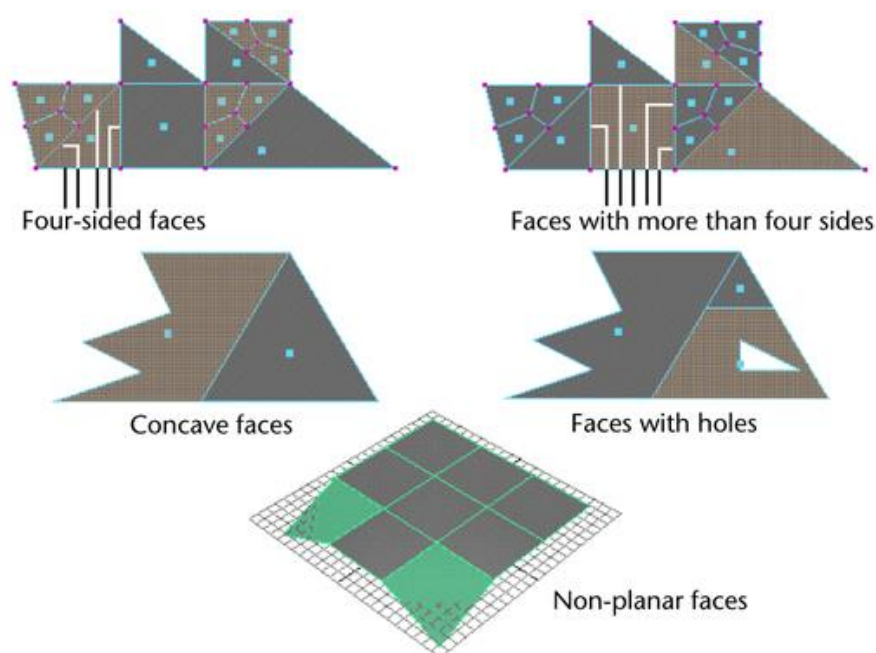


Kuva 13. Matalaresoluutioinen kassakone

#### 6.4 Huomioitavat asiat 3D-mallin geometrian tuotannossa

Koska 3D-mallit voivat olla todella yksityiskohtaisia ja monimutkaisia, saatetaan luomisvaiheessa tehdä virheitä. Nämä virheet johtavat pahimmillaan siihen, että mallia ei pysty tuomaan 3D -mallinnusohjelmasta muihin ohjelmiin seuraavia työvaiheita varten, tai siinä voi esiintyä visuaalisia virheitä. Yksityiskohtaisuutta voi esimerkiksi vertaamalla keskenään aiemmin mainittuja kuutioita ja kassakoneetta. Kuutio sisältää kahdeksan verteksiä, kun taas kassakoneessa niitä on 1356 kappaletta. Kassakoneessa on siis paljon enemmän ”liikkuvia osia”, mikä itsessään nostaa mahdollisuutta tehdä virheitä.

Eräs tapa tarkistaa mallin virheitä on käyttää ”smooth preview” -ominaisuutta (3). Tämä ominaisuus luo esikatselun mallista korkearesoluutioisena, jolloin virheet mallin verkossa on helppo paikantaa. Muutoin virheiden korjaamiseen Mayassa on olemassa ”Cleanup” -ominaisuus (Mesh Display -> Cleanup), jossa ohjelma joko korostaa facet ja verteksit, joissa on ongelmia, tai suoraan korjaa ne. Yleisimmät kohdattavat ongelmat, jotka cleanup pystyy korjaamaan on esitelty graafisesti alla, mutta selvennän jokaisen esimerkin tapauskohtaisesti.



Kuva 14. Tesselaatio-ongelmia

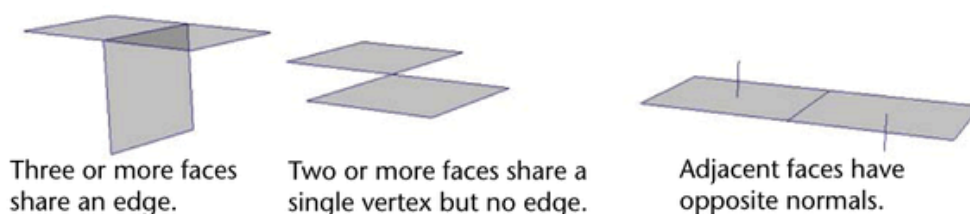
Alkaen ylhäältä vasemmalta, cleanup-toiminnolla on mahdollista korjata neljästä kulmaverteksistä muodostuvat facet. Vaikkakin tämänkaltaisen face on täysin hyväksyttävä ja toimiva, on olemassa joitain tapauksia joissa on haluttavaa muuttaa nämä “triangleiksi”, eli kolmesta verteksistä muodostuvaksi faceksi. Esimerkiksi jos malli halutaan siirtää Mayasta johonkin toiseen ohjelmaan, joka ei välttämättä pysty käsittelemään muita faceja, kuin triangleja. Kyseinen cleanup-ominaisuus on kuitenkin selkeästi vähiten käytetty. Seuraava kuvan osio, eli facet joissa on enemmän, kuin neljä verteksiä on taas hyvinkin käytetty. Ongelma vaikkapa viisiverteksisissä faceissa on, että mikäli verteksit eivät pysy yhdeltä akselilta katsottuna samassa linjassa, facen geometria näyttää virheelliseltä ja rumalta.

Kuva ei havainnollista kovin hyvin, mistä tarkalleen Concave Faces -kohdassa on kysymys. Joissain tapauksissa voi käydä niin, että geometrian eri kohtien välissä oleva face ei enää muistuta muodoltaan neliötä, vaan vaikkapa jonkinlaista sahanterää. Tällöin face tulisi jakaa pienempiin osiin, jotta yksi face ei joudu olemaan monimutkaisen muotoinen. Toki concave face kuvassa rikkoo myös sääntöä yli neljän verteksin facesta, mutta concave face voi tapahtua myös täysin normaalin facen kanssa. Seuraava kohta, eli “faces with holes” tarkoittaa juurikin mitä kuvassa näytetään. Facessa ei saisi olla reikiä, vaan mahdolliset avoimet kohdat täytyy tehdä jakamalla face pienempiin osiin,

jolloin reikä ei ole varsinaisesti enää reikä facen keskellä, vaan joukko faceja, joiden keskellä on aukko.

Viimeinen kuvan esimerkki, eli non-planar faces tarkoittaa seuraavaa: Mikäli face muodostuu vaikkapa neljästä verteksistä ja kolme niistä on samalla tasolla jollain koordinaatti-akselilla, mutta neljäs on vaikkapa alempana, tapahtuu graafinen virhe. Mikäli asiaa haluaa havainnollistaa itselleen, voi ottaa neliön muotoisen paperinpalan ja suorittaa saman kokeen, jossa yksi kulma on muita ylempänä, tai alempana. Tällöin itse face ei pysty käsittelemään kompleksia muotoa, vaan tuottaa graafisen virheen. Helpoin tapa ratkaista tämä on jakaa face edge loopilla pienemmäksi kokonaisuudeksi, jolloin kulmana toimiva edge luo halutun muodon ilman virheellisyyttä.

Juuri esiteltyt virheet faceissa ovat ns. Tesselaatio-ongelmia. Kuitenkin cleanup -toimintoa tarvitaan myös toiseen merkittävään ongelmatyyppiin. 3D-mallin tulisi olla geometrialtaan useista elementeistä koostuvaa geometriaa, eli “two-manifold geometry” periaatteella toteutettua. Käytännössä tämä tarkoittaa, että mikäli malli jaettaisiin osiksi ja projisoitaisiin kaksiulotteiselle pinnalle, mitkään sen osat eivät olisi päällekkäin. Mikäli mallissa on tässä suhteessa ongelmia, on se elementitöntä geometriaa, eli “non-manifold geometry”. Useat Mayan sisäiset työkalut vaativat toimiakseen, että 3D-mallin geometria on useista elementeistä koostuvaa. Alla olevassa kuvassa esitellään esimerkkejä miten malli voi olla elementitöntä geometriaa. Ensiksi, mikäli kolme tai enemmän faceja jakavat yhteisen edgen, tulee ongelmia. Mallia ei tulisi rakentaa alunperinkään niin, että tähän ongelmaan törmää, sillä 3D-mallin meshin tulisi olla käytännössä vain ulkokuori.



Kuva 15. Elementitön geometria

Mikäli vasenta esimerkkiä tutkittaisiin suoraan yläpuolelta, ei vertikaalinen face muutenkaan olisi nähtävillä, joten sen olemassaolo on turhaa. Mitään mallin sisäisiä faceja ei täten tulisi olla, koska niillä ei ole funktiota. Toinen esimerkki voi tapahtua

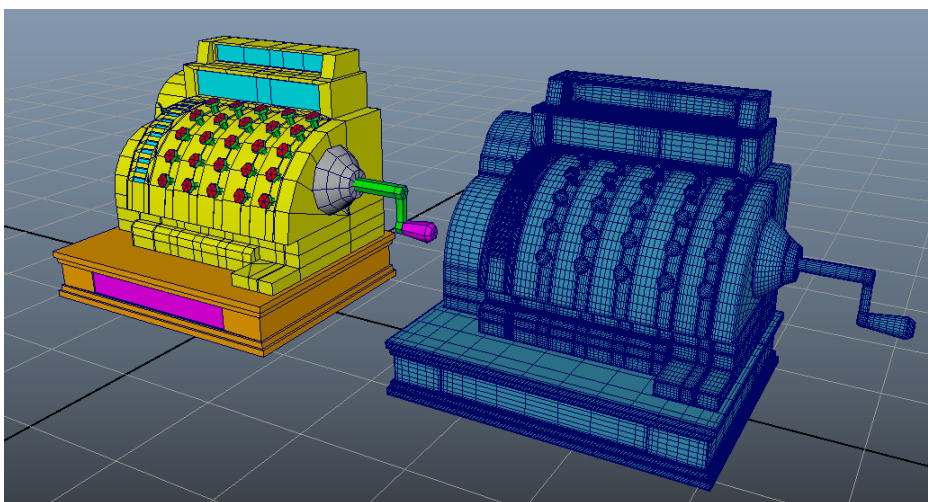
esimerkiksi tilanteessa, jossa mallista poistetaan useita faceja. Tällöin esimerkin oikealla ja vasemmalla olevat tyhjat kohdat voisi täyttää trianglella ja ongelma häviäisi. Esimerkkikuva ei havainnollista kovinkaan hyvin kolmatta ongelmaa, joten selvennän sitä. Aiemmin on todettu, että 3D-mallin tulisi olla käytännössä objektin ulkokuori. Ulkoapäin katsottuna tämän objektin facet ovat läpinäkymättömiä, mutta jos objektia tarkastelisi sen sisäpuolelta, huomaisi että kaikki facet ovat läpinäkyviä. Tämä johtuu siitä, että mallit renderöidään niin, että faceja pystyy tarkastelemaan vain yhdeltä puolelta. Jossain vaiheessa voi kuitenkin käydä niin, että virheen vuoksi jokin faceista on väärin päin, l. sen "normaali" on väärin päin. Vaikkakin virheen pystyy korjaamaan myös cleanup-toiminnolla, pystyy faceja kääntelemään myös Reverse Normal -toiminnolla (Mesh Display -> Reverse). Vaikka cleanup-toiminnossa on myös muita toiminnallisuuksia, ovat nämä paljon harvinaisempia ongelmia ja esimerkiksi koko asiakasyritykselle tehdyn projektin aikana niitä ei tarvittu lainkaan.

Edellä mainitut ongelmat ovat hyvin yleisiä ja jotakuinkin jokaisessa Outrageous Grounds: The Mazea varten tehdyissä malleissa oli virheitä, jotka täytyi korjata. Tämä ei tarkoita, että mallintaja olisi huono työssään, vaan että ongelmien havainnointi ja korjaaminen ovat tavallisia osia mallinnusprosessia. Sillä vaikka ongelmia yrittäisi välttää parhaansa mukaan, on täysin virheetön mallinnusprosessi käytännössä mahdotonta toteuttaa mikäli malli on monimutkaisempi, kuin neliskulmainen laatikko. Suositeltavaa onkin tallentaa työnsä eri tiedostoihin siirryttäessä työvaiheista toisiin, jotta vahingon sattuessa voi palata aiempaan versioon sen sijaan, että joutuisi tekemään kaiken alusta. Toinen suositeltava asia on laittaa Mayan asetuksista päälle autosave (Windows -> Settings/Preferences -> Preferences -> Files/Projects), joka tallentaa työn automaattisesti tietyin väliajoin. Tämä on suositeltavaa siksi, että joskus ohjelma voi kaatua ohjelmavirheen takia ja tällöin kaikki tallentamatta jäänyt on peruuttamattomasti hävinnyt. On myös hyvä mainita, että tämän kaltainen suhtautuminen tallennukseen ja versionhallintaan on oleellista omaksua kaikissa pelinkehitykseen liittyvissä prosesseissa olivat ne sitten 3D-mallinnusta, pelimoottorin kanssa toimimista, tai koodaamista.

Yhteenvetona voidaankin sanoa, että tärkeintä on huomata ja korjata virheet ennen seuraaviin työvaiheisiin siirtymistä, tallentaa usein ja pitää järjestelmällisesti yllä kunnollista versiointia. Muutoin voi vahingossa heittää hukkaan pahimmillaan useita

työtunteja. Mikäli kyseessä olisi iso korkearesoluutioinen malli, voi menettää kymmeniä työtunteja.

Kuten alla olevasta kuvasta voi nähdä, olemme lisänneet aiemmin nähdyn kassakoneen viereen toisen. Lisäksi Mayan sisällä on havainnollistamisen helpottamiseksi pistetty päälle visuaalinen moodi, jossa molempien mallien meshin osat pystyy erottamaan selkeämmin. Kuten vasemmanpuolisesta kassakoneesta voi huomata, se on varsin kulmikas verrattuna oikeanpuoleiseen. Tuon vasemman kassakoneen 3D-malli on siis matalaresoluutioinen. Lähes kaikki peleissä nähtävät mallit ovat tavalla tai toisella matalaresoluutioisia. Koska pelimoottorit joutuvat renderöimään reaaliajassa samanaikaisesti tuhansia objekteja, on niiden oltava graafiselta tasoltaan riisuttuja tai pelimoottori ei pysty pyörittämään niitä sulavalla ruudunpäivitysnopeudella.



Kuva 16. Matalaresoluutioisen objektin vertailu korkearesoluutioiseen.

On kuitenkin olemassa keinoja saada matalaresoluutioinen malli näyttämään paljon sulavammalta, kuin se oikeasti on. Näitä keinoja käytetään sekä peleissä, elokuvissa, että muussa visuaalisessa mediassa. Korkeatarkkuuksisilla tekstuurikartoilla pystytään luomaan illuusio esimerkiksi siitä, että jokin kulma on pyöreämpi, kuin se oikeasti on. Eli jos katsotaan taas kuvaa kahdesta kassakoneesta, tekstuurikarttojen kanssa on mahdollista saada vasemmalla oleva kassakone näyttämään oikeanpuoleiselta kassakoneelta ilman tarvetta tehdä mallin verkosta monimutkaisempi.

Palataksemme kuvaan: vasemmanpuoleinen kassakone sisältää 1356 verteksiä, kuten aiemmin mainittiin. Oikeanpuoleisessa, korkearesoluutioisessa mallissa niitä on 73822



kappaletta, eli sen yksityiskohtien määrä on huomattavasti suurempi. Sen muodot ovat myös pyöreämpiä ja se näyttääkin aidommalta, kuin vasemmanpuoleinen. Korkearesoluutioisen mallin tehtävä on teksturointivaiheessa toimia pohjana matalaresoluutioisen mallin päälle projisoitaville normaalikartoille, i. tietyntyyppisille tekstuurikartoille. Tämä tarkoittaa, että normaalikartta luo illuusion siitä, että matalaresoluutioinen malli olisi yksityiskohtaisempi, kuin se oikeasti on. Normaalikartta esimerkiksi saa kulmikkaat kohdat näyttämään pyöreämmiltä. Vaikkakin normaalikartoilla voidaan tehdä myös muita asioita, kuin kulmien pyöristämistä, käsitellään aihetta enemmän teksturointia koskevassa osiossa.

Korkearesoluutioinen malli tuotetaan lisäämällä edge loopeja malliin, tarkistaen aina välillä smooth preview:n (pikanäppäin 3) avulla miltä malli näyttää. Tässä vaiheessa korkearesoluutioiseen malliin ei saa lisätä uutta geometriaa, sillä muuten se ei täsmää matalaresoluutioisen mallin kanssa. Kun korkearesoluutioinen malli on halutun tarkkuuksinen, käytetään Smooth -toimintoa (Mesh -> Smooth), jolloin se muuttuu smooth preview:n mukaiseksi todellisuudessa.



Kuva 17: Kuvassa sama kynttilänjalka. Oikeanpuolimmaisessa on päällä smooth preview.

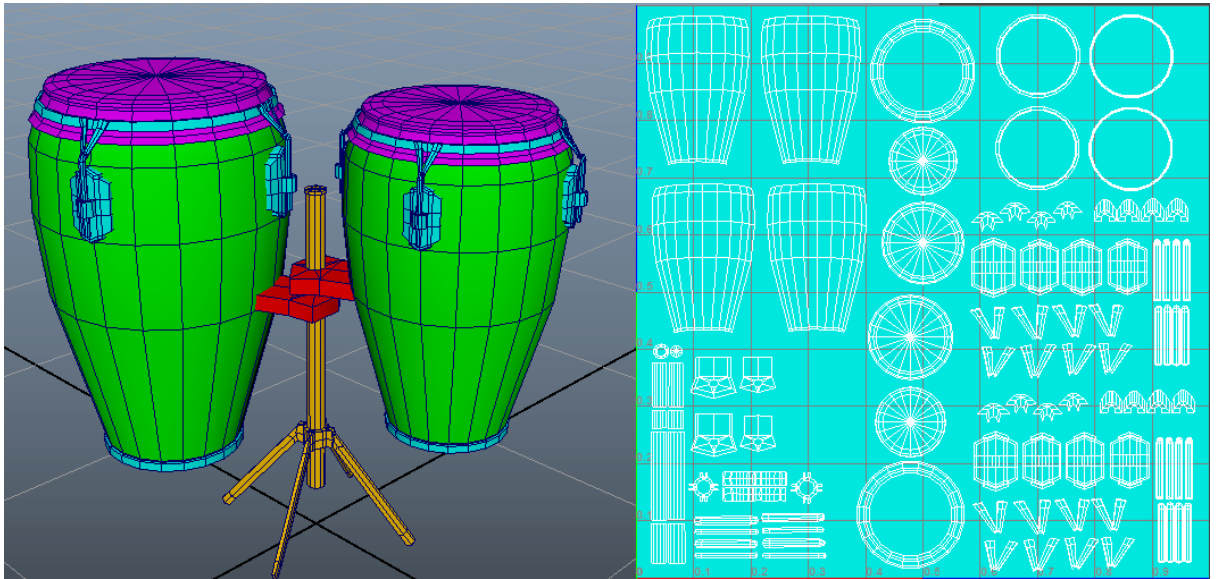
## 6.5 Mallien UV-projisointi



Kun mallin mesh on valmis ja virheetön, on aika siirtyä seuraavaan työvaiheeseen, eli UV-mappaukseen (UV -> UV Editor). Tämä toimenpide tarkoittaa 3D-mallin meshin facejen projisointia automaattisesti tai manuaalisesti kaksiulotteiselle pinnalle. Toimenpiteen tarkoitus on muuntaa facet sellaiseen muotoon, jotta malli voidaan teksturoida. Mikäli tätä toimenpidettä ei tehtäisi ja siirryttäisiin vain teksturointiin, olisi lopputuloksena graafisesti virheellinen malli. On huomautettava, että UV-kartta tehdään vain mallin matalaresoluutioiselle versiolle. Korkearesoluutioisessa mallissa riittää vain virheetön verkko.

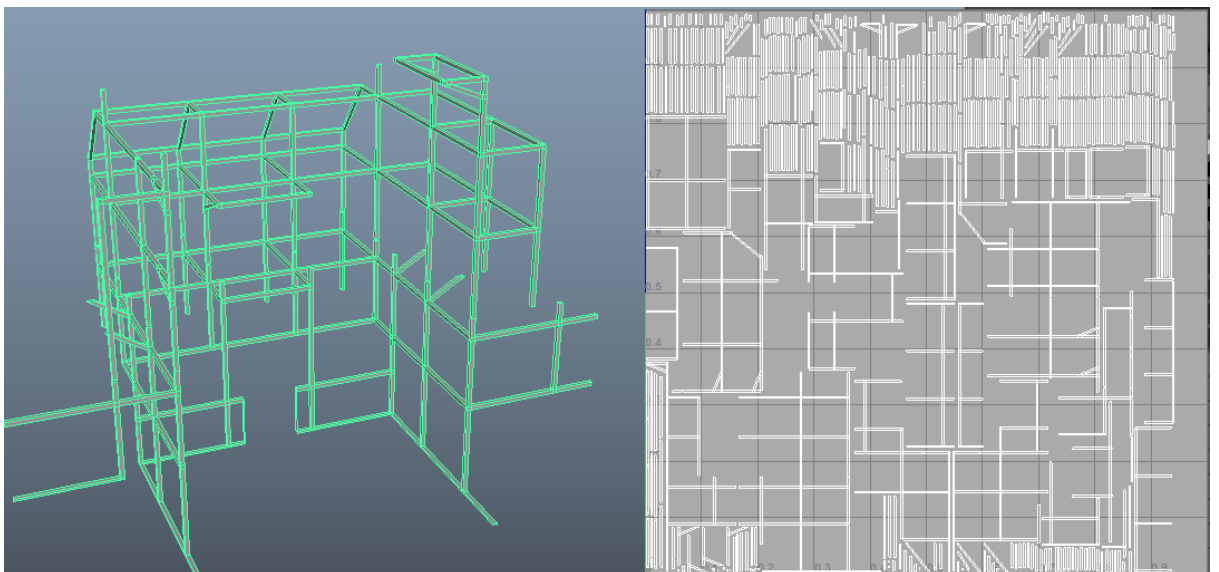
Alla olevassa kuvassa on esimerkkimalli, congarummut, jotka tehtiin osana opinnäytetyöprojektia. Kuten aiemminkin, rumpujen väritys kuvassa ei vastaa lopullista tuotetta, vaan värit ovat havainnollistamista ja teksturointia varten. Rumpujen oikealla puolella on valmis UV-kartta, jossa rumpujen facet on aseteltu paikalleen teksturointia varten. Kuvaa tutkimalla ja vertailemalla kuvan osasia toisiinsa, voi alkaa huomaamaan muotoja ja 3D-geometriaa, jotka toistuvat molemmissa osissa kuvaa. Vierekkäin, vailla välejä olevat palaset käsitetään yhdeksi pinnaksi, jossa tekstuuri jatkuu saumatta. Usein 3D-objekti on niin monimutkainen, että sitä on mahdotonta projisoida UV-karttaan yhtenä palana. Tällöin ainoa ratkaisu on jakaa UV:t osiin. Tällä jakamisella ei ole mitään vaikutusta itse malliin, mutta se tulee vaikuttamaan teksturointiin. Esimerkiksi jos katsotaan UV-kartan vasenta yläkulmaa, johon on projisoitu rumpujen kehikon (kuvassa vihreä) facet, niin huomataan, että koko kehikko on jaettu kahtia. Tämä on siksi, että esimerkiksi muodoltaan lieriötä muistuttavaa objektia ei ole mahdollista projisoida kaksiulotteiselle alustalle nopeasti, vaan kyseessä on aikaa vievä prosessi, josta saatavat hyödyt eivät välttämättä ole käytetyn ajan arvoisia.

Kuitenkin, tämä kehikkojen jako osiin tulee aiheuttamaan teksturoitaessa huomattavan sauman siinä kohdassa jossa tekstuurit kohtaavat. Tällöin tekstuurin eri osat eivät ole välttämättä tasossa keskenään. Tämänkaltaiset UV-saumamat ovat kuitenkin varsin normaaleja 3D-mallien tuotannossa, eikä niistä pääse koskaan kokonaan eroon. Mikäli haluaa, niin on toki tapoja piilottaa tämä sauma, mutta se vaatisi esimerkiksi verkon luomisvaiheessa sitä, että tulevan sauma-alueen päällä on jotain muuta geometriaa, joka peittää tämän. Yksi mahdollinen keino häivyttää UV-saumoja on manuaalisesti sumentaa rajoja, vaikkapa Photoshopin blur-työkalun kanssa.



Kuva 18. Congarummut ja niiden UV-kartta

UV-projisoinnin vaikeus riippuu, kuten aiemmin mainittu, mallin yksityiskohtaisuudesta ja sen geometrian monimutkaisuudesta. Alla olevassa kuvassa on jälleen eräs malli, joka tuotettiin opinnäytetyöprojektia varten ja kuvan oikealla puolella sen UV-kartta. Kyseessä oleva malli sisältää erään pelikentän sisäisen alueen rakenteelliset tuet. Tämän mallin UV-projektio tehtiin automaattityökalulla (UV -> Automatic), sillä se koostuu käytännössä primitiiveistä, eikä siinä esiinny pyöreitä muotoja. Automaattityökalu ei kuitenkaan osaa itse asetella UV-projisoinnin osasia johdonmukaisesti, vaan sen algoritmi pyrkii täyttämään UV-kartan optimaalisesti käytetyn tilan perusteella, jolloin mahdollisimman suuri osa UV-karttaa on käytössä.



Kuva 19. Puurakenne ja sen UV-kartta

Mikäli prosessin haluaa tehdä manuaalisesti, tulee UV Editor avata ja mennä facen valintatilaan (Oikea hiiren painike pohjaan -> liikuta osoitin "Face" -kohdan päälle -> päästä irti). Tämän jälkeen tulee valita ne kaikki facet, jotka haluaa projisoida. Kun facet on valittu, tulee hiiren osoitin viedä UV Editor -ikkunan päälle ja käyttää "planar map" -työkalua (Shift + Hiiren oikea painike -> vie hiiri "planar map" kohdan päälle). Tämä projisoi halutut facet UV-karttaan. Kuitenkin tälläkin tavalla voi UV-karttaan jäädä ongelmakohtia, kuten UV-deformaatioita tai facejen päällekkäisyyksiä. Vaikkakaan facejen päällekkäisyydet eivät ole aina toivottavia, on tekniikkaa mahdollista käyttää toistuvien rakenteiden tekstuurien yhdenmukaistamiseen. Eli esimerkiksi vaikkapa portaikkoa tehdessä olisi mahdollista asettaa jokaisen portaan UV samaan kohtaan. Saatava etu on käytännössä se, että mallin UV-projektio saadaan paremmin mahtumaan sille varattuun tekstuurikarttatalaan, jolloin jokainen UV-projektion osanen voi olla isompikokoinen ja täten tarkempi tekstuuriltaan.

Aiemmin mainittuja ongelmia on mahdollista korjata joko face-valintatilan avulla, tai siirtyä UV-valintatilaan, joka valitsee UV-kartan verteksit (Oikea hiirenpainike pohjaan -> UV -> UV -> vapauta hiiren painike). Tämän manuaalisen sijoittelun päämääränä on pitää UV-kartan osaset saman kokoisina ja niiden muodot johdonmukaisina, sillä mikäli esimerkiksi jos jokin face on väärän kokoinen suhteessa muihin, on myös siihen myöhemmin tehtävä tekstuuri väärän kokoinen ja tulee aiheuttamaan graafisen virheen. UV Editor -ikkunassa on pieni painike, joka muistuttaa shakkilautaa. Laittamalla tämän päälle, muuttuu 3D-mallin tekstuuri shakkilautaa muistuttavaksi. Tämän visuaalisen työkalun tarkoituksena on helpottaa UV-kartan eri osasten kokojen yhdenmukaisuuden. Mikäli työprosessi on mennyt suunnitelmien mukaan, tulisi shakkiruudukon olla saman kokoinen koko mallin alueella.

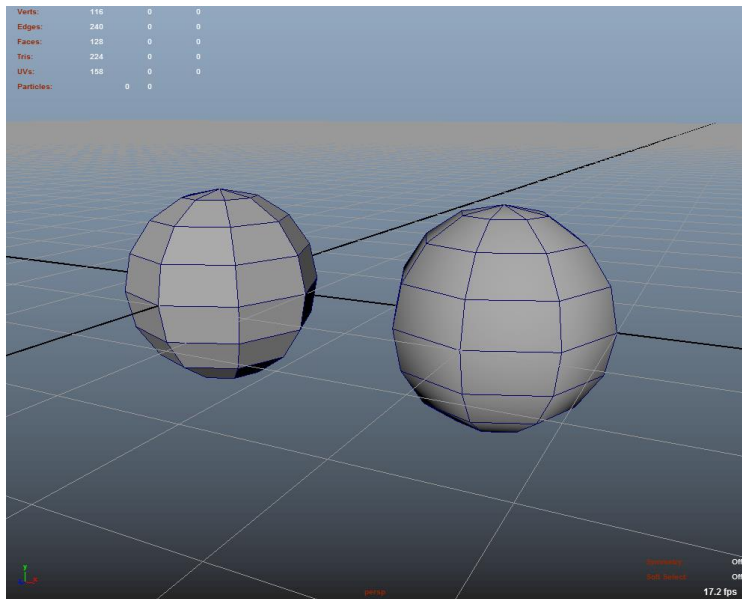
## 6.6 Mallien valmistaminen teksturointia varten.

Tässä vaiheessa prosessia tulee tehdä muutama nopea työvaihe. Tuotetun matalaresoluutioisen ja korkearesoluutioisen mallin tulee sijaita samoissa koordinaateissa Mayan editorinäkyvässä, muutoin seuraavissa työvaiheissa tulee

virheitä. Optimaalisessa tilanteessa mallit sijaitsevat valmiiksi samassa tilassa, mutta mikäli eivät, täytyy ne keskittää. Ensiksi täytyy valita malli. Tämän jälkeen valitaan ”Center Pivot” -työkalu (Modify -> Center Pivot). Tällöin kolmiakselinen graafinen elementti, joka on näkyvissä manipulaatiotyökalujen ollessa valittuna, keskittyy mallin keskipisteeseen. Tämän jälkeen se tulee siirtää alaspäin, niin että se on samassa tasossa mallin alimman tason kanssa (D pohjassa + V pohjassa -> siirrä hiirellä -> päästä irti). Kun pivot on keskitetty, voidaan sen jälkeen itse malli keskittää Mayan koordinaattiruudun keskikohtaan, jossa kaksi paksua mustaa linjaa kohtaa. Keskittäminen onnistuu valitsemalla malli, painamalla X pohjaan ja siirtämällä mallia. X:n avulla tapahtuva siirtäminen siirtää mallia yhden koordinaattiruudun verran suuntaansa kerrallaan.

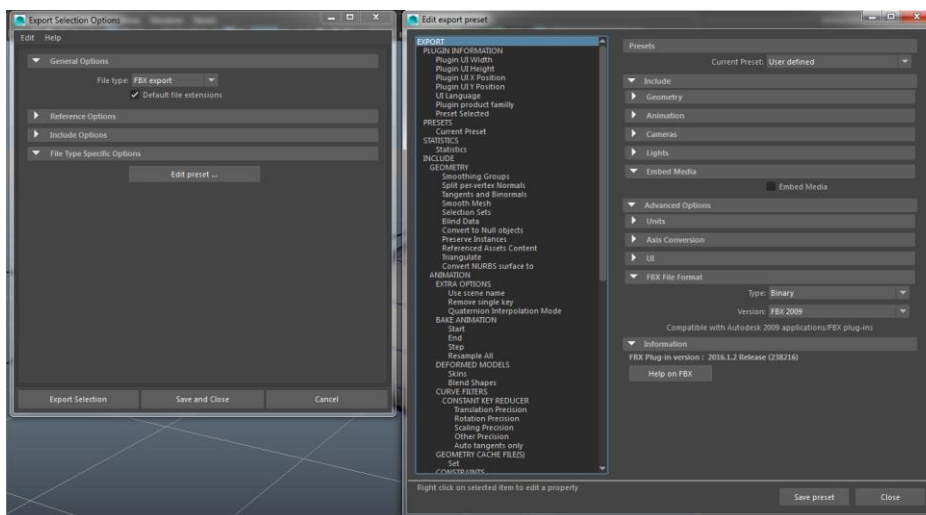
Kun molemmat mallit on keskitetty samaan kohtaan tulee niiden historia poistaa ja ”transformaatio” jäädyttää. Ensiksi mainittu tarkoittaa, että mallin tekoprosessissa tiedostoon voi jäädä aavedataa, joka pitää sisällään mallin muotoutumisen alkuprimitiivistä lähtien. Historian poistaminen tapahtuu valitsemalla (Edit -> Delete By Type -> History). Transformaation jäädyttäminen tarkoittaa sitä, että mallin koordinaatit ja kääntöasteet nollataan, jolloin mallin silloinen sijainti ja kääntöaste vastaavat vaikkapa pelimoottorissa sen käyttämien koordinaattien nollakohtaa. Mikäli tätä ei tehtäisi, vaan mallin X-koordinaatti olisi vaikka 540, niin pelimoottoriin tuotaessa se ei olisi koordinaattien nollakohdassa, vaan koordinaateissa 540 yksikköä suuntaan X. Transformaation jäädyttäminen tapahtuu (Modify -> Freeze Transformations) -työkalun kautta.

Lopuksi on mahdollista käyttää Set Normal Angle -työkalua (Mesh Display -> Set Normal Angle), jos mallin kulmista halutaan pehmeämpiä tai kovempia. Esimerkiksi pyöreissä muodoissa on hyvä käyttää arvoja, jotka tuovat pehmeyttä, sillä muutoin malli näyttää kulmikkaalta. Tämä optio ei vaikuta mallin geometriaan mitenkään, vaan ainoastaan siihen, miten se renderöidään, esimerkiksi pelimoottorin sisällä.



Kuva 20. Kaksi ympyräprimitiiviä, joista oikeanpuolimmaisien Normal Angle on muutettu arvoon 55.

Kun mallin verkko on valmis ja virheetön, siitä on tehty korkearesoluutioinen versio ja matalaresoluutioinen malli on UV-kartoitettu, tulee ne tuoda ulos Mayasta. Maya tallentaa kaikki omat mallitiedostonsa muotoon “.mb”, joka tulee sanoista Maya Binary. Tätä tiedostoa ei ole kuitenkaan mahdollista käyttää teksturoinnissa, tai tuoda pelimoottoriin. Tällöin meidän täytyy käyttää Mayan “Export” -toimintoa (File -> Export Selection). Tämä valmistaa valitun objektin ulostuontia varten. Export -toiminnossa voi valita tiedostotyyppin, jolle valittu objekti käännetään. Opinnäytetyöprojektin tapauksessa export-toiminnolla tuotettiin kolme mallia kahdella eri tiedostotyyppillä. Ensiksikin matalaresoluutioinen, sekä korkearesoluutioinen malli käännetään “.fbx” -muotoon. Tämän jälkeen matalaresoluutioinen malli exportataan uusiksi, “.obj” -muodossa.

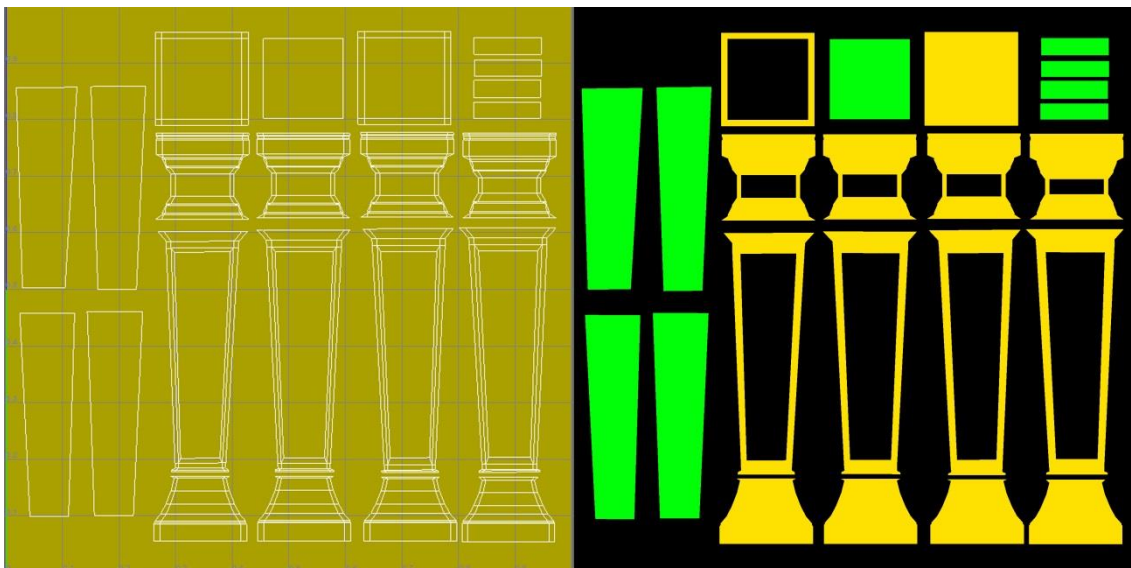


Kuva 21. Vasemmalla "Export Selection" -dialogi. Painamalla "Edit Preset" oikeanpuolimmainen valikko avautuu, joka mahdollistaa tarkemmat säädöt.

Unity, sekä xNormal tukevat FBX-tiedostomuotoa, joten siihen muotoon käännettyjä malleja käytetään sekä normaalikarttojen luomiseen, että pelimoottoriin sisällyttämiseen. FBX-exportia tehtäessä voi olla haluttavaa käydä läpi tiedostomuotokohtaisia lisäasetuksia (File -> Export Selection -> pieni neliö Export Selectionin oikealla puolella). Tämä avaa Export Selection Options -valikon, josta tulee painaa "Edit Preset ..." -painiketta. Tämä avaa jälleen uuden valikon, josta tulee navigoida esiin Advanced Options -> FBX File Format -> Version. Tästä on mahdollista käyttää exportissa uudempia tai vanhempia FBX-versioita. Esimerkiksi opinnäytetyöprojektin kanssa oli tarpeellista käyttää FBX 2009:ää, sillä kohtasimme uudempien versioiden kanssa ongelmia tuodessa ne pelimoottoriin. Obj-muotoon käännetty matalaresoluutioinen malli on teksturointitarkoituksia varten, sillä Quixel Suite ei tue FBX-tiedostomuotoa. Muutoin olisi johdonmukaista käyttää vain yhtä export-tiedostomuotoa.

Viimeinen työvaihe ennen teksturointia on Color ID -karttojen luominen matalaresoluutioisen mallien avulla Mayan sisällä. Myös tämä työvaihe on peräisin Quixel Suiten tarpeista. Ensiksi mallin kaikkiin tahoihin tulee laittaa uusi lambert-materiaali. Kun halutut facet ovat valittuna pidetään oikea hiirenpainike pohjassa ja valitaan ilmestyvästä valikosta "Assign New Material". Tämä avaa dialogin, jossa olevasta listasta tulee valita "Lambert". Kun valinta tehdään, luo Maya valituille faceille uuden materiaalin ja oikealle puolelle Mayan editorinäkymää ilmestyy valikko, josta voi säätää materiaalin nimeä ja ominaisuuksia. Nyt tärkein muutettava asia on "Color", eli

materiaalin väri. Johtuen teksturointiohjelman, eli Quixel Suiten vaatimuksista, on näiden värien parasta olla mahdollisimman kirkkaita ja värikkäitä. Tämän vaiheen tarkoituksena on “merkitä” teksturointiohjelmalle, että missä kohdissa mallia käytetään mitään materiaalia. Tällöin mallintajan tulee merkitä saman värisellä materiaalilla ne kohdat mallista, joissa olisi muutoinkin samaa “materiaalia”. Eli esimerkiksi ne kohdat mallista, joihin halutaan puinen tekstuuri merkitään samalla lambert-materiaalilla, ne kohdat joihin halutaan metallia merkitään samalla lambert-materiaalilla ja niin edespäin. Lambert-materiaalin nimen voi jättää halutessaan muuttamatta, sillä nimellä ei ole vaikutusta teksturointia ajatellen, mutta mikäli haluaa olla johdonmukainen voi tämänkin tehdä.



Kuva 22. Vasemmalla puupedestaalin UV-kartta, oikealla Material ID-kartta.

Kun mallin kaikki facet on merkitty tarvittavilla lambert-materiaaleilla, tulee Color ID -kartta nyt tuoda ulos Mayasta. Vasemmassa yläkulmassa, File -valikon alla on avautuva valikko, jossa lukee “Modeling”. Tätä täytyy klikata ja vaihtaa tila “Rendering” päälle. Valikko itsessään pitää sisällään valmiita asetuksia ja työkalurivejä erilaisia Mayan sisällä tapahtuvia työvaiheita varten, mutta kuten mainittu, tarvitsemme nyt Rendering -tilan. Tämän jälkeen tulee valita “Lighting/Shading” valikosta “Transfer Maps”. Tämä valinta avaa uuden valikon, josta pääsee tekemään tarvittavat asetukset. Ennen seuraavaa vaihetta valitse malli ja paina Ctrl+D. Tämä pikakomento kopioi valitun mallin kokonaisuudessaan ja sen pitäisi näkyä Mayan hierarkiassa.

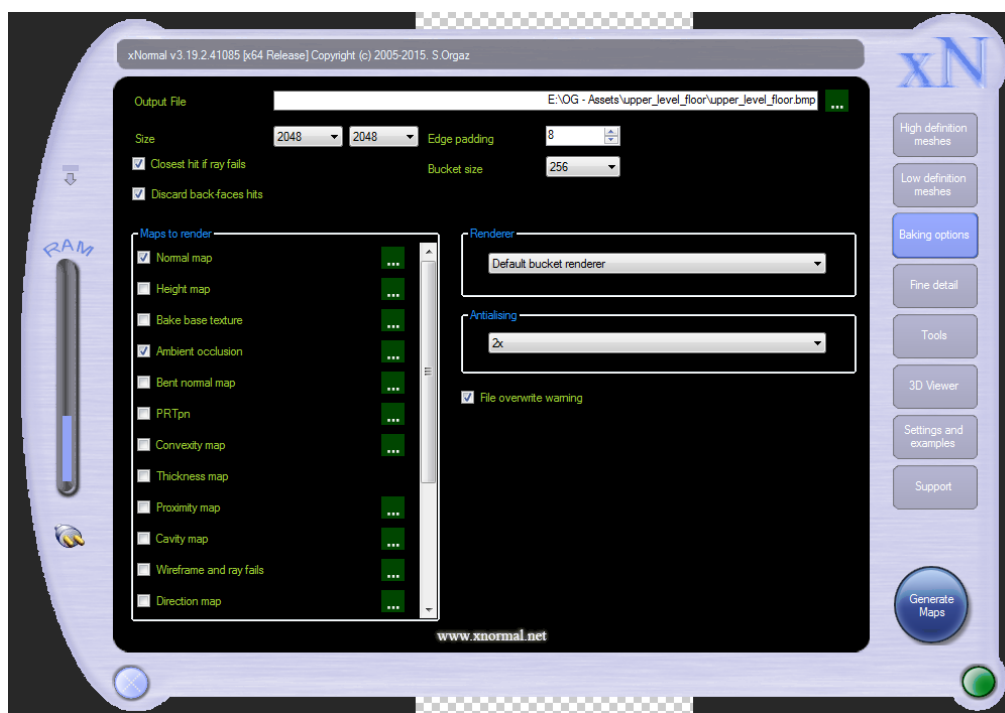
Mallin kopioiminen tehtiin, koska Transfer Maps -valikossa pitää valita sekä Source Mesh, että Target Mesh. Valinta tapahtuu niin, että kun malli on valittu, painetaan “Add Selected”. Laita malli target meshiksi ja sen kopio source meshiksi. Tämä oudolta tuntuva työvaihe johtuu siitä, että käytännössä Transfer Maps -toimintoa käytetään Mayan sisällä, mallien kesken. Mutta vaikka tarkoituksenamme on vain saada color ID -kartta ulos Mayasta, tulee molemmat meshit lisätä valikon asetuksiin. Mesh myös kopioitiin sen vuoksi, ettei valikko anna käyttää samaa meshiä sekä Target Meshinä, että Source Meshinä. Lisäksi kannattaa laittaa “Search Envelope (%)” -asetus nolnaan. Tämä tuottaa lopullisesta Color ID -kartasta mahdollisimman terävän.

Transfer Maps -valikon keskiosassa on kohta “Output Maps”. Tästä tulee valita vain “Diffuse Color Map”, jättäen muut mahdolliset kartat valitsematta. Tiedostomuoto jota opinnäytetyöprojektissa käytettiin kartan formaattina oli .tiff. Kun loput säädöt on tehty, tulee painaa “Bake and Close” ja odottaa, että Maya valmistaa kartan. Tässä voi kestää useita minuutteja. Kun Color ID -kartta on viimein ulkona Mayasta, voidaan siirtyä seuraavaan työvaiheeseen.

## 6.7 Teksturointi

Seuraavaksi tulee tuottaa mallin normaali-, sekä ambient occlusion-kartat. Tämä tapahtuu aiemmin mainitulla ohjelmalla nimeltä xNormal. Ohjelman käynnistyttyä tulee ohjelmaan syöttää tuotetut FBX-muodossa olevat 3D-mallit. Tämä tapahtuu ohjelman oikealla puolella olevasta valikosta. “High definition meshes” on tarkoitettu korkearesoluutioiselle mallille ja “Low definition meshes” matalaresoluutioiselle mallille. Valikon “Baking options” -kohdasta voi säätää tuotettujen karttojen laatua. Baking options -valikon “Output file” -kohdasta pystyy nimeämään tuotettavat kartat. Nimeämiskäytäntö toimii niin, että lopulliset tiedostot nimetään [käyttäjän laittama nimi] [kartan tyyppi] -tavalla. Sen alla sijaitseva “size” -kohta määrittää tuotettavan tekstuurikartan koon. Outrageous Grounds: The Mazea varten tekstuurikartan kooksi oli asiakkaan toimesta määritetty 2048x2048. Vaikka korkeampi resoluutio olisi ollut mahdollinen nähtiin sen aiheuttavan optimointiongelmia. Alhaisempi resoluutio ei taas olisi tarjonnut halutun tasoista visuaalista lopputulosta.



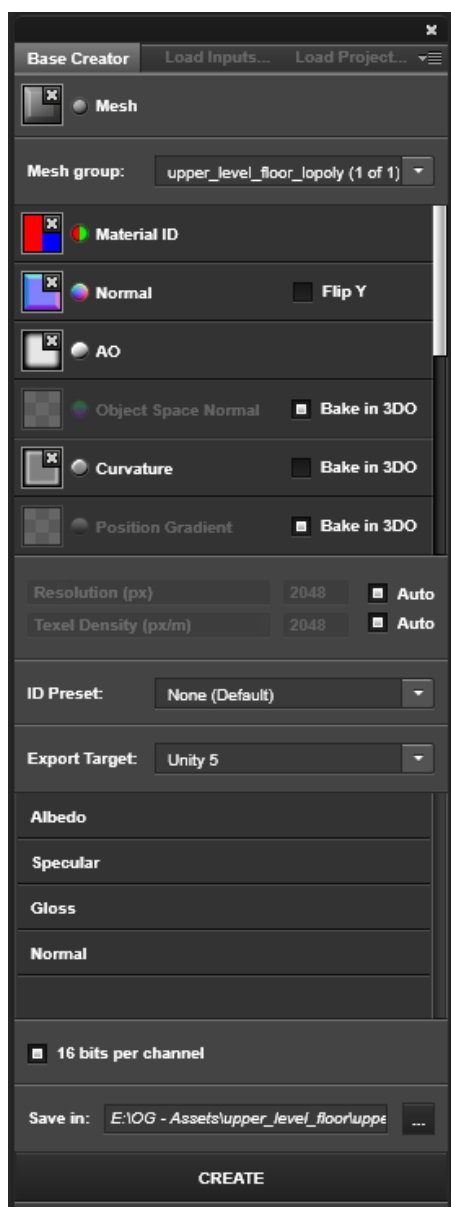


Kuva 23. Xnormal:n asetusikkuna

Baking options -valikosta voi lisäksi valita vasemmasta alalaidasta, kohdasta “Maps to render” tuotettavat tekstuurikartat. Koska tarvitsemme vain kaksi karttaa, valitsemme listasta vain “Normal map”- ja “Ambient Occlusion”-valinnat. “Size” -valinnan oikealla puolella on kohta “Edge padding” säätelee kuinka suuren reunan ohjelma luo 3D-mallin UV-kartan pohjalta tehtävän tekstuurikartan osien ympärille. Liian suuri arvo voi aiheuttaa graafisia virheitä karttaan, sillä projisoitavan kartan osat voivat mennä päällekkäin. Outrageous Grounds: The Mazessa käytettiin arvoa 8.

Kun molemmat 3D-mallit on syötetty ohjelmaan ja tarvittavat asetukset määritetty, painetaan oikeasta alakulmasta “Generate Maps” -nappia. Tämä aloittaa prosessin ja tuo näkyviin graafisen esityksen kartan tekoprosessista. Ohjelma varoittaa virhedialogilla ennen prosessin alkamista mikäli malleissa on virheitä. Lisäksi, mikäli UV-kartoituksessa on virheitä, pystyy ne havaitsemaan graafisen esityksen aikana. Tässä prosessissa voi kestää useita minuutteja, riippuen kartan graafisesta tarkkuudesta. Kun kartat on tuotettu, ilmoittaa ohjelma siitä dialogilla. Tämän työvaiheen jälkeen korkearesoluutioista mallia ei enää tarvita. Se kannattaa kuitenkin säilyttää kaiken varalta.

Sen jälkeen, kun xNormalilla on luotu normaalikartta ja ambient occlusion -kartta siirrytään Quixel Suiteen tekemään loput teksturoinnista. Kuten on aiemmin mainittu, Quixel Suite on lisäosa Photoshop-ohjelmaan. Käynnistettäessä se siis avaa Photoshopin ja lisää sen päälle omat valikkonsa. Teksturointi alkaa valikosta, jossa on stylisoidut kirjaimet N, D, M ja luku 3. Painamalla D:tä avautuu uusi valikko. Tämän valikon avulla malli valmistetaan teksturointia varten Quixel Suitessa. Teksturoinnin pohjana käytetään aiemmin Mayasta ulos tuotua .obj -muotoista matalaresoluutioista 3D-mallin verkkoa. Tämä tehdään valitsemalla Quixel Suiten valikosta "Mesh" ja valitsemalla oikea tiedosto. Valinnan alapuolella on "Material ID" -kohta, johon valitaan Mayasta ulos tuotu Color ID -kartta. Seuraavat kaksi valintaa, eli "Normal" ja "AO" ovat xNormalin kautta tuotettuja tekstuurikarttoja varten. Kun kaikki kartat on valittu, tulee lisäksi valita "Curvaturen" oikealta puolelta valinta "Bake in 3DO". Seuraavaksi alemmaa valikosta tulee valita "Export Target" -kohtaan "Unity 5". Tietenkin mikäli tekstuureja tehtäisiin muita peli- tai renderöintimoottoreita varten, valittaisiin asiankuuluva valinta. Unity 5 on kuitenkin pelimoottori, jolla Outrageous Grounds: The Maze kokoonpannaan, joten esimerkissämme käytämme sitä. Alin valikon osa "Save in" valitsee tallennuspolun ja nimen myöhemmin valmiille tekstuurikartoille. Kun asetukset on valittu, tulee valikon alalaidasta painaa "CREATE".

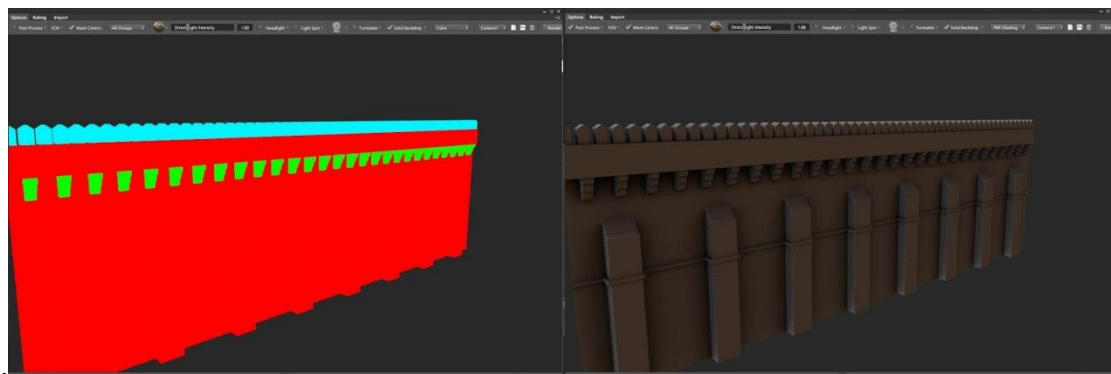


Kuva 24. Quixel Suiten aloitusvalikko

Nyt Quixel Suite valmistelee annettujen asetusten mukaisesti mallin teksturointia varten. Tarvittavien komponenttien lataus kestää hetken aikaa ja Quixel ilmoittaa dialogi-ikkunalla, kun se on valmis. Seuraavaksi tulee valita ensiksi mainitusta Quixelin ylävalikosta kohta “3”. Tämä avaa 3D-näkymän mallista, jossa sitä voi käännellä, teksturoida ja valaista erilaisilla asetuksilla. Tämä näkymä on hyvin havainnollistava ja oleellinen osa teksturointiprosessia Quixel Suitessa

Alla olevassa kuvassa esitellään Quixel Suiten 3D-näkymää. Oikealla puolella on perusnäkymä, jossa näkyy ohjelmaan tuotu 3D-malli sellaisena, kuin se on teksturoinnin

alkaessa. Kyseessä oleva malli on muurinpätkä, joka tuotettiin Outrageous Grounds: The Mazea varten. Kuvan vasemmalla puolella on sama malli, jonka Material ID:t on korostettu pikanäppäimellä “C”. Kuten kuvasta huomaa, Material ID -kartta erittelee muurin pienempiin osiin. Jokaisella erivärisellä osalla voi olla oma tekstuurinsa ja jokaisen värikokonaisuuden voi teksturoida nopeasti muutamalla napinpainalluksella.



Kuva 25. Quixel Suiten työnäkymä. Vasemmalla puolella objektin Material ID korostettu.

Prosessi toimii niin, että kun Material ID on korostettuna (C pohjassa), painetaan “Shift”. Näppäinten ollessa pohjassa painetaan hiirellä jotain mallin osaa. Esimerkiksi kuvassamme voitaisiin klikata jotain vihreäksi korostetuista mallin osista. Tämä avaa materiaalivalikon, jossa on valmiita materiaaleja käytettäväksi teksturoinnissa. Quixel Suite sisältää valmiiksi suuren materiaalikirjaston, jossa on kaikenlaista aina luusta nahkaan ja kankaaseen. Tämän lisäksi on mahdollista luoda omia materiaaleja, mikäli kirjaston tarjoamat vaihtoehdot eivät riitä. Tässä opinnäytetyössä ominaisuutta ei kuitenkaan käsitellä. Materiaalivalikkoa esitellään alla olevassa kuvassa, jossa näkyy myös vaihtoehtojen moninaisuus. Kuvassa on kaksi erilaista kivimateriaalia, joista toisesta löytyy myös sammaleiset versiot.

## 7 MITEN KOKOONPANNA TASO UNITY-PELIMOOTTORISSA

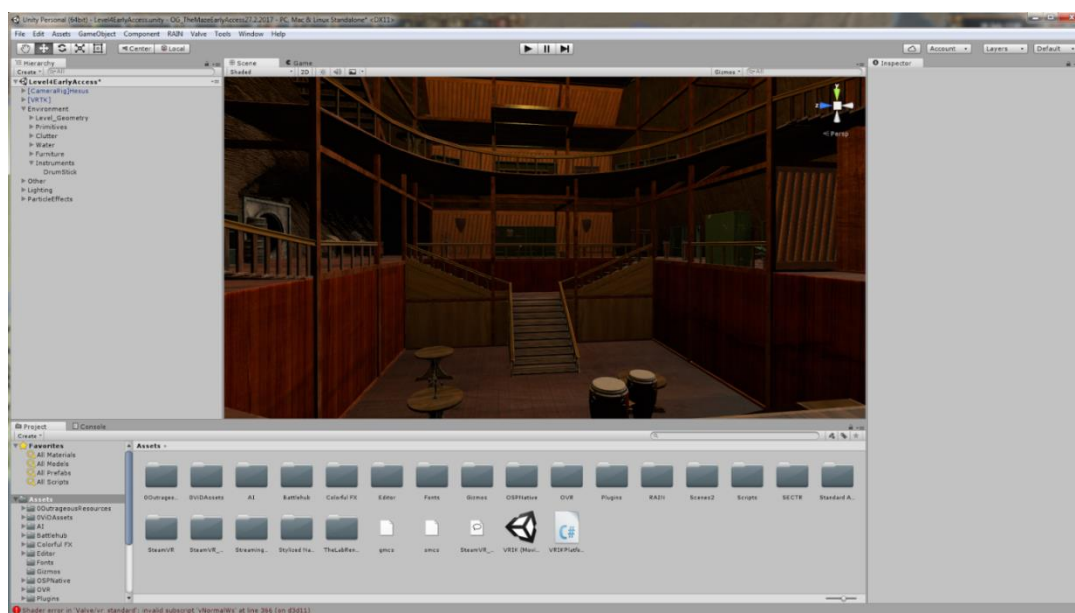
Tämä osio olettaa, että lukija osaa käyttää Unityn editori-ikkunan manipulaatiotyökaluja, joilla ruudulla olevien objektien sijaintia voi muuttaa. Kun teksturointivaihe on päättynyt ja käsillä ovat aiemmin tehdyt mallit FBX-muodossa, sekä Quixelin kautta tuodut tekstuurikartat, voidaan kaikki luodut assetit tuoda Unityyn.

Nimitystä Unity-projekti käytetään tietystä instanssista, jossa on tiettyä tarkoitusta varten tehdyt assetit ja koodipohja. Esimerkiksi Outrageous Grounds: The Maze on Unity-projekti kehitysvaiheessa, ennen julkaisua. Tällä on se merkitys, että normaalisti käyttäjän täytyy luoda uusi projekti, mikäli mitään halutaan tuoda Unityyn. Vaikkakin opinnäytetyöprojektissa oli käytettävissä valmis Unity-projekti, käydään koko prosessi selvyyden vuoksi läpi.

Kun Unity avataan, se tuo ensiksi käyttäjän silmien eteen aloitusvalikon. Valikosta valitaan joko valmis projekti, tai painetaan painiketta “New” Dialogin yläosassa. Avautuvassa valikossa on mahdollista nimetä projekti, määrittää sille tiedostopolku, sekä valita oletuksena käytettävät asset-kirjastot. Tämän jälkeen painetaan “Create project” -nappia, jolloin Unityn editorinäkömä avautuu.

## **7.1 Sijoittelu ja hierarkia**

Ennen mallien tuomista ohjelmaan, on hyvä mainita sananen hierarkiasta Unityn sisällä. Tällä tarkoitetaan eri 3D-objektien, äänien, koodin ja muun pelin osasten sijoittelua kansiorakenteeseen projektin sisällä. Koska projekteissa usein työskentelee useita henkilöitä saman asian parissa, on kaiken oltava järjestyksessä. Muutoin esimerkiksi jotain kriittistä 3D-mallia ei pysty löytämään tarpeen tullen, tai jotkin tekstuurikartat sijaitsevat lukuisissa erilaisissa alikansioissa. Täten kansiorakenteiden tulisi olla ennalta sovittuja ja selkeitä, jotta ongelmilta vältyttäisiin.



Kuva 26. Unityn editorinäkömä ja hierarkiaruudut.

Yllä olevassa kuvassa on esitelty kaksi hierarkiaa. Vasemmalla puolella kuvaa on lista, joka on kuvan keskellä nähtävän scenen hierarkia. Se siis sisältää rakenteen, johon kaikki kentästä löytyvät asiat on järjestetty loogisesti, jotta tarvittavia osia pääsee käden käänteessä muokkaamaan. Kuvan alaosassa taas on koko projektin hierarkia. Tämä sisältää koko projektin kaikki tiedostot, myös äsken mainitun scenen.

Seuraavaksi esitetään ne hierarkiat joita Outrageous Grounds: The Mazessa käytettiin. Koska projektissa käytetään lisäosaa nimeltä SECTR, joka mahdollistaa kentän jakamisen osiin ja osasten lataamisen erikseen, tulee hierarkiaan lisätasoja. Tämä lisätaso löytyy jokaisesta kategoriasta ja toimii niin, että kun yksi sektori ladataan, niin se aktivoi paitsi kenttägeometrian, niin myös valaistuksen, viholliset ja äänet. Samalla aiempi sektori deaktivoituu, jolloin se ei enää syö prosessointitehoa.

Ylimmällä tasolla jaetaan kentän objektit ensiksi kolmeen alaluokkaan, Environment, Other ja [VRTK]. Environment sisältää ympäristön, eli kenttägeometrian ja kaikki staattiset objektit, kuten mahdolliset kasvit. Jos haluttaisiin vaikka löytää tietty kasvi kentästä, vaikkapa saniainen, olisi sen hierarkia seuraavanlainen: Environment -> Level1\_Sectors -> MazeA\_ASector -> ASector\_Items -> Vegetation -> MazePlants -> Ferns -> plant\_Fern. Kuten huomaa, mikäli ei työskentele SECTR:n kanssa, olisi polku paljon lyhyempi: Environment -> Vegetation -> Mazeplants -> Ferns -> plant\_Fern. Ajatus on kuitenkin selvä, eli jokainen pieni osanen kuuluu aina isompaan kategoriaan,

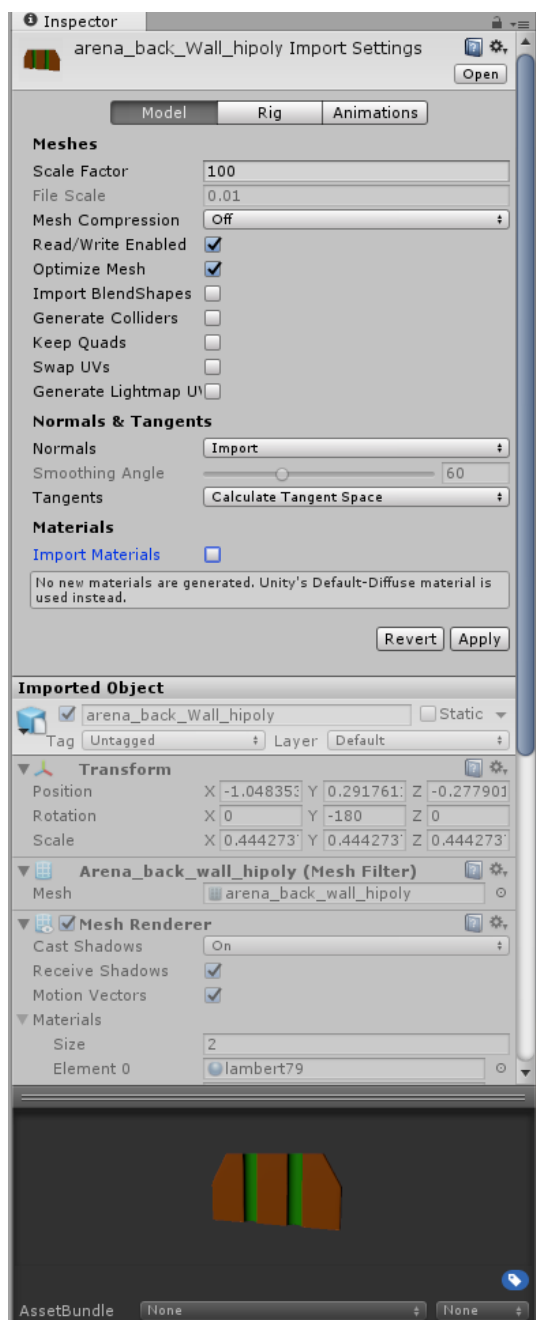
joka itsessään kuuluu johonkin toiseen yläkategoriaan ja niin edelleen. Esimerkiksi aiemman esimerkin saniaisessa hierarkian tason “Ferns” alla voi olla niin monta plant\_Ferniä, kuin halutaan.

Muista kategorioista “Other” sisältää kaiken teknisen, eli käytettävät esineet, äänet, viholliset ja erilaiset triggerit, sekä valaistuksen. Kolmas kategoria sisältää VRTK-lisäosan komponentit, joka säätelee miten VR-ohjaimet toimivat pelin sisällä. Tässä ei esitellä muiden scenekategorioiden hierarkiaa, sillä ne eivät varsinaisesti eroa aiemmin esitellystä.

Projektihierarkiassa ylimmällä tasolla ovat lisäosakansiot, resurssikansiot ja scenekansio. Lisäosakansiot sisältävät jokainen erilaisen lisäosan ja kaikki sen komponentit. Yksi resurssikansioista sisältää kaiken yrityksen itse tekemän, eli itse tuotetut koodit, mallit, äänet ja muut assetit. Muut resurssikansiot sisältävät Unity marketplacesta ostettuja asetteja. Scenekansio sisältää jokaisen pelin kentän Scene-tiedoston. Yrityksen itse tekemien asettien kansio jakautuu erikseen animaatioon, ääniin, materiaaleihin, malleihin, prefabeihin ja tekstuureihin. Jokaisella tällaisella kategorialla on vielä useita alakategorioita. Esimerkiksi 3D-mallit jakaantuvat tyypin mukaan vaikkapa hahmoihin, sälään, maastoon ja rakenteisiin.

## **7.2 Mallin ja tekstuurien tuominen Unityyn**

Kun olet hierarkiassa oikealla tasolla, klikkaa hiiren oikealla painikkeella projektihierarkian oikeanpuolimmaista ikkunaa ja valitse “Import New Asset”. Tämä avaa valikon, jossa voit valita haluamasi assetin ja klikata “Import”. Esimerkin varjolla importaamme nyt 3D-objektin. Malli on matalaresoluutioinen FBX-muodossa oleva 3D-objekti, jolle on tuotettu tekstuurit. Ensiksi valitaan siis malli dialogin kautta. Unity lataa seuraavaksi pienen hetken, jonka jälkeen Unityn editori-ikkunan oikealla laidalla olevaan Inspector-ikkunaan ilmestyy valintoja. Lyhyesti sanottuna Inspector -ikkuna mahdollistaa kentässä olevien asettien ominaisuuksien muokkaamisen. Tässä tapauksessa se kontrolloi millä asetuksilla asset tuodaan ohjelmaan. Ikkuna esitellään tarkemmin alla olevassa kuvassa.



Kuva 27. Unityn inspector-valikko

3D-mallin tapauksessa Inspector jakautuu kolmeen välilehteen. Ensimmäisen käsittelee mallin itsensä tuontia Unityyn, toinen sen animointia varten luodun luurangon, eli rigin tuontia ja kolmas käsittelee mallin animaatioita.



Ensimmäinen välilehti on ainoa oleellinen mikäli tuodaan staattisia objekteja Unityyn. Scale Factor säätelee mallin skaalaa. Opinnäytetyöprojektissa käytettiin asiakkaan puolelta standardina skaalaa 100, koska se on kymmenellä jaollinen ja täten helposti muunneltavissa. Alempaa “Import Blendshapes” pistetään pois päältä, mikäli malleissa ei ole niitä. Blendshape on Mayalle ominainen tekniikka, jossa 3D-mallin verkko voi muokkautua useaan erilaiseen muotoon. Staattisissa objekteissa niitä tuskin on. Lisäksi “Import Materials” laitetaan pois päältä, jotta Unityyn ei tuoda turhaa dataa. Materiaalit tässä tapauksessa tarkoittavat Mayan sisällä asetettuja materiaaleja, joita aiemmin käytettiin Material ID -karttojen pohjana. Tämän jälkeen painetaan nappia “Apply”, jonka jälkeen malli on valmis käytettäväksi. Unityyn tuotu malli ilmestyy projektihierarkiaan, josta sen voi hiirellä raahaamalla tuoda auki olevaan sceneen. Kuitenkin tässä vaiheessa se on vielä harmaa, sillä tekstuureita ei ole vielä tuotu ja materiaaleja asetettu.

Tekstuurien tuominen ohjelmaan toimii samalla periaatteella, kuin mallin tuominen, eli Import New Asset -valinnan kautta. Outrageous Grounds: The Mazen kanssa käytäntö oli, että mallit, tekstuurit ja materiaalit ovat eri kansioissa. Yhdessä tekstuurikansiossa sijaitsevat saman mallin albedo-, normal- ja specular-kartat. Tuodessa normal-karttaa Unityyn on tarpeellista vaihtaa sen tyyppi Inspector-valikon kautta, sillä muuten Unity käsittelee sitä tavallisena tekstuurikarttana. Kohdasta “Texture Type” tulee tyypiksi vaihtaa “Texturen” sijaan “Normal map” ja painaa Apply.

Pelkkä tekstuurien lisääminen ei kuitenkaan riitä, vaan tekstuurit tulee sitoa “materiaaliin”. Tämä tarkoittaa, että kaikki tekstuurikartat yhdistetään yhdeksi instanssiksi. Hierarkiassa tulee seuraavaksi mennä materiaalikansioon, painaa oikeaa hiiren painiketta ja valita Create -> Material. Tämä tuo Inspector-valikkoon valintaruudun, jossa materiaalia pääsee muokkaamaan. Koska opinnäytetyöprojekti oli VR-pelille, tulee ylimmäisenä oleva “Shader” vaihtaa toiseen oletusarvosta. Outrageous Grounds: The Maze käyttää Valve VR -shadereita, joten valikosta tulee valita Valve -> vr\_standard. Nämä shaderit eivät oletusarvoisesti tule Unityn mukana, joten jos esimerkiksi tuotettu malli on tehty “tavallista” peliä varten, ei arvoa välttämättä tarvitse muuttaa. Esimerkissä kuitenkin käytämme Valve VR -shadereita. Specular Mode -valinta alempana määrittää onko materiaali metallista esinettä, vai muita esineitä varten. Mikäli materiaali ei ole metallista esinettä varten, valitaan valikosta “BlinnPhong”.

Specular Moden valinta määrittää myös mitä tekstuurikarttoja materiaaliin pystyy syöttämään. Alla on valintakohdat Albedo, Normal ja Specular, jotka vastaavat tuottamiemme tekstuurikarttoja. Kun kartat on syötetty, on materiaali melkein valmis käytettäväksi.

Materiaalissa on kuitenkin lisäasetuksia, joiden muokkaaminen muuttaa materiaalin ulkonäköä. “Cube Map Scalar” -asetus määrittää miten materiaali heijastaa ympäristöä, joten mikäli valmis materiaalilla varustettu malli näyttää erilaiselta, kuin esimerkiksi teksturointiohjelmassa, voi arvoa olla hyvä säätää. Tämän lisäksi on hyvä kiinnittää huomio alempana oleviin kohtiin “Tiling” ja “Offset”. Tiling määrittää kuinka monta kertaa tekstuuri toistuu. Esimerkiksi jos ajatellaan suorakulmion muotoista litteää tasoa, niin sillä olevan tekstuurin voi pistää toistumaan useita kertoja, mikäli esimerkiksi huomataan, että se venyy liikaa ja graafinen ulkonäkö kärsii siitä. Offset taas kertoo kuinka monella pikselillä tekstuuria siirretään sen sijainnista koordinaateissa X ja Y mallin päällä. Vielä alempana oleva “Secondary Maps” sisältää samat vaihtoehdot ja paikat “Detail Albedo”:lle ja “Detail Normal”:lle. Nämä ovat lisätekstuurikarttoja, joita voi lisätä alkuperäisten tekstuurikarttojen päälle. Niiden käyttötarkoitus on pienien, useita kertaa toistuvien detaljien lisääminen varsinaisten tekstuurien päälle (Unity. 2017.) Kun kaikki tarvittavat säädöt on tehty, voi materiaalin laittaa mallin päälle vetämällä sen projektihierarkiasta editoriruudussa sijaitsevan mallin päälle. Mikäli malli koostuu useasta osasta, on materiaali tarpeellista vetää ja tiputtaa myös vielä teksturoimattomien osien päälle. Nyt malli on pelimootorin sisällä käyttövalmis ja prosessi voidaan toistaa muiden mallien kanssa.

### **7.3 Prefabien luonti**

Mikäli mallia käytetään useita kertoja saman projektin sisällä, vaikkapa eri kentissä, on siitä hyvä tehdä prefab. Mikäli prefabia ei tehtäisi, jouduttaisiin mallia käytettäessä aina aloittamaan teksturoimattomalla mallilla, johon materiaali tulisi aina vetää uusiksi. Kuitenkin, mikäli scenehierarkiassa on tällainen malli, johon on laitettu materiaali, voi sen vetää hiirellä projektihierarkiaan, jolloin Unity luo siitä automaattisesti prefabin. Tällöin malli ja sen materiaali ovat yhdessä paikassa, josta sen voi vaivattomasti vetää takaisin sceneen tarvittaessa. Prefabit kannattaa myös sijoittaa omiin kansioihinsa ja

nimetä selkeästi, etteivät ne sekoitu varsinaisiin malleihin. Mikäli on tarpeellista tarkistaa onko scenessä oleva objekti prefab vai ei, tulee sitä klikata hiirellä, jotta sen Inspector-ikkuna aukeaa. Tällöin voi tarkistaa lukeeko sen yläosassa “Model” vai “Prefab”.

## 8 POHDINTA

Opinnäytetyöprojekti oli kirjoittajalle mielekäs kokemus. Koska projekti toteutettiin toimeksiantajalle tehdyn harjoittelujakson aikana, muistutti se mahdollisimman paljon oikeita päivätöitä alan yrityksessä. Alun perin tarkoitukseni oli toteuttaa opinnäytetyöprojekti sooloprojektina hieman samankaltaisesta aiheesta, mutta koen toimeksiantajalle tehdyn projektin olleen sisällöllisempi.

Opinnäytetyöprojektin aikana kokemukseni 3D-mallintamisesta ylitti kahden vuoden rajapyykin. Kuitenkin olin jo etukäteen tietoinen, että 3D-tuotanto kokonaisuutena, eli mallin verkon valmistaminen, teksturointi, animointi on suunnattoman laaja ala oppia. Täten projektia oli pakko lähteä työstämään avoimin mielin, valmiina oppimaan uutta. Toimeksiantaja esittikin nopeasti uusia työmetodeja ja tapoja tehostaa omia työtottumuksia. Esimerkiksi näppäinoikoteiden hallinta ja omien työkalurivien muokkaaminen tekevät 3D-mallin verkon tekoprosessista jouhevaa ja luontevaa. Kenties tärkeimpänä, itselleni selvisi se taso mitä vaaditaan mallilta graafisesti, jotta se on hyväksyttävä. Tämä pätee paitsi siihen, että missä vaiheessa 3D-malli täyttää minimivaatimukset, niin myös milloin mallin kanssa työskenteleminen kannattaa lopettaa, sillä tehty lisätyö ei tuo siihen arvoa. Aikaisemmin ongelmanani on ollut usein, etten viitsi päästää mallista irti ja todeta sen olevan valmis. Koenkin, että harjoittelujakso auttoi paljon tämän ongelman suhteen.

Toimeksiantajan tarjoama työympäristö mahdollisti ammattimaisen työnteon ja toimivan palauteketjun. Molempien merkitykset korostuivat projektin aikana, sillä kehitystiimi on pienikokoinen. Täten jokaiselta ryhmän jäseneltä vaaditaan kykyä toimia osa-alueilla, jotka ovat heidän oman mukavuusalueensa ulkopuolella. Tämä on siis täysin erilainen lähestymistapa, kuin isoissa alan yrityksissä, jossa jokaista työtehtävää hoitaa alan asiantuntija. Eli esimerkiksi käsikirjoitusta työstää

täysipäiväinen ammattikirjoittaja. Kuitenkin, opinnäytetyöprojektin kannalta toimeksiantajalla oli tarjota asiantuntemusta yhden, myöhemmin kahden, ammattitaidoltaan kirjoittajaa korkeamman 3D-mallintajan muodossa. Koska toinen näistä henkilöistä oli myös esimieheni, pystyi hän antamaan tarkkaa palautetta paitsi tuotettujen 3D-mallien tasosta, niin myös työtavoista ja miten korkeamman tason konseptit sopivat peliin temaattisesti. Voidaankin sanoa, että työelämään perehdyttävänä jaksena Virtual Dawnilla vietetty harjoittelujakso toimi jokseenkin niin hyvin kuin se ideaalitason ulkopuolella vain voi.

VR alustana oli kiintoisa yksityiskohta harjoittelussa. Oman pelitason kokeminen VR-laseilla antoi mielenkiintoisia perspektiivejä paitsi kentän rakentamisen suhteen, myös mallien skaalojen suhteen. Teknisellä tasolla VR ei vaadi kovinkaan paljoa mallintajalta, sillä kaikki mitä opinnäytetyöprojektissa vaadittiin oli shaderin vaihto toiseen Unityn sisällä ja sen komponenttien säätäminen. Tämä tapahtuu siis yleensä vasta ns. loppukokoonpanossa, jolloin se ei tuo ylimääräisiä työvaiheita vaikkapa mallin verkon luomiseen tai teksturointiin. Tämä mahdollisti nopean iteraatioprosessin, mikäli esimerkiksi huomattiin, ettei jokin kentän osanen toiminut, vaan se piti tehdä uusiksi.

Kuitenkin, tehty työ jäi väistämättä kesken. Tekemäni pelitaso ei ole vielä julkaisukelpoinen, joten se täytyy työstää valmiiksi kokonaisuudeksi. Kuten on mainittu jo aiemmin opinnäytetyössä, niin kentästä puuttuvat kaikki pelilliset elementit, kuten viholliset, ei-pelaajahahmot ja yleinen progressio. Nämä kaikki ovat olemassa konseptitasolla, mutta niitä ei ole implementoitu kenttään. Tämän lisäksi kaikki graafiset efektit, kuten valaistus ja partikkeliefektit puuttuvat tai ne ovat tällä hetkellä paikanpidikkeitä. Kyseisten ominaisuuksien tuottaminen ei ollut osa opinnäytetyöprojektia, mutta toimeksiantaja on ilmaissut jo kiinnostuksensa jatkokehitystä varten, joten mitä luultavimmin ne tullaan toteuttamaan kirjoittajan toimesta myöhempanä ajankohtana. Ajatus on mieluinen, sillä valaistus on suuri osa paitsi mallintamista, myös kenttäsuunnittelua. Valolla ja sen puutteella saa luotua eri alueiden välillä dynamiikan vaihteluita, jotka tekevät kentästä kiinnostavamman kokonaisuuden. Koen myös henkilökohtaisesti, että valaistuksen luominen ja suunnittelu ovat sellaisia osa-alueita, joissa kaipaavat vielä kipeästi harjoitusta. Täten kentän viimeistely valaistuksen osalta toimisi luontevana jatkona opinnäytetyöprojektin aikana toteutetulle kokonaisuudelle ja harjoittaisi kirjoittajan ammattitaitoa entisestään.

- [1] MOT. Kielitoimiston sanakirja. 2017a.  
<http://www.kielitoimistonsanakirja.fi/netmot.exe?ListWord=todellisuus&SearchWord=todellisuus&dic=1&page=results&UI=fi80&Opt=1>
- [2] Rauhala, L., 2005. Tajunnan Itsepuolustus. 3. painos. Helsinki: Gaudeamus, 15.
- [3] Rainio, K., n.d. Tajunta ja todellisuus. Luettu 29.3.2017.  
<http://personal.inet.fi/koti/kullervorainio/kotisi42.html>
- [5] MOT. Kielitoimiston sanakirja. 2017b.  
<http://www.kielitoimistonsanakirja.fi/netmot.exe?ListWord=virtuaalinen&SearchWord=virtuaalinen&dic=1&page=results&UI=fi80&Opt=1>
- [6] History of Virtual Reality. Virtual Reality Society. Luettu 10.4.2017.  
<https://www.vrs.org.uk/virtual-reality/history.html>
- [7] Hayden, S., 2017. Epic games CEO: “HTC Vive is outselling Oculus Rift 2-to-1 worldwide”. Road to VR. Julkaistu 9.1.2017. Luettu 8.4.2017.  
<http://www.roadtovr.com/htc-vive-sales-units-oculus-rift-comparison-compared-tim-sweeney/>
- [8] Kauppinen, J.O., 2015. TrackIR 5 Professional lisää pelissä olemisen tuntua. Dome.fi. Julkaistu 18.02.2015. Luettu 7.4.2017  
<https://muropaketti.com/dome/pelit/artikkelit/laitteet/trackir-5-professional-lisaa-pelissa-olemisen-tuntua>
- [9] Statista. 2017. Forecast unit shipments of augmented (AR) and virtual reality (VR) headsets from 2016 to 2021 (in millions). Statista. Luettu 8.4.2017.  
<https://www.statista.com/statistics/426429/hmd-virtual-reality-unit-sales-worldwide/>
- [10] Steamdb. 2017. Luettu 7.4.2017. <https://steamdb.info/genres/>
- [11] Lang, B., 2017. On the Hunt for VR’s Killer App with Oculus’ Head of Content, Jason Rubin. Road to VR. Julkaistu 22.2.2017. Luettu 8.4.2017.  
<http://www.roadtovr.com/hunt-vrs-killer-app-oculus-head-content-jason-rubin/>
- [12] Donnelly, J., 2017. VR-exclusive Job Simulator has made \$3 million in sales. Road to VR. Julkaistu 9.1.2017. Luettu 9.4.2017. <http://www.pcgamer.com/vr-exclusive-job-simulator-has-made-3-million-in-sales/>
- [13] Johnson, J., 2017. Will Indie developers dominate VR? Intel IQ. Julkaistu 3.1.2017. Luettu 6.4.2017. <https://iq.intel.com.au/will-indie-developers-dominate-vr/>
- [14] Pluralsight, 2015. 3ds Max vs. Maya: Is One Better than the Other? Pluralsight. Julkaistu 29.1.2015. Luettu 4.4.2017. <https://www.pluralsight.com/blog/film-games/3ds-max-vs-maya-is-one-better-than-the-other>

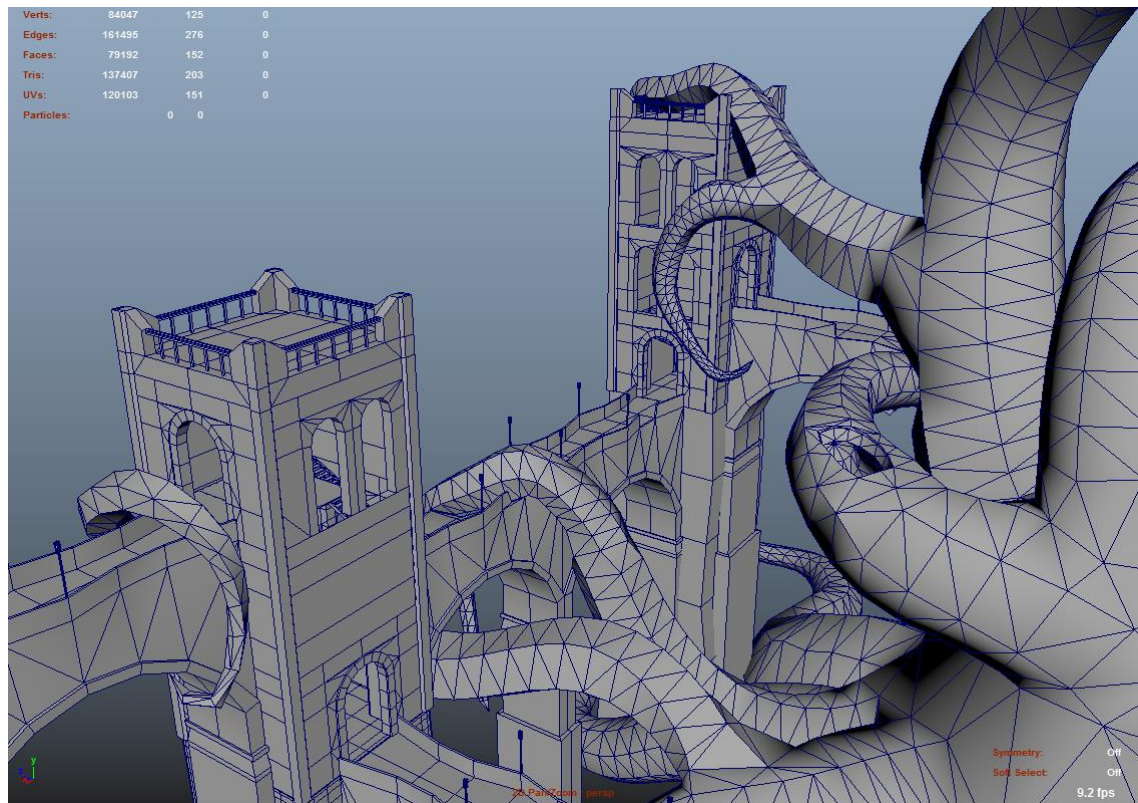
- [15] Bradley, S., 2012. How to create Visual Tension in Your Designs. Vanseo Design. Julkaistu 10.1.2012. Luettu 4.4.2017. <http://vanseodesign.com/web-design/visual-tension/>
- [16] Unity. 2017. Unity User Manual (5.6). Secondary Maps (Detail Maps) & Detail Mask. Julkaistu 21.4.2017. Luettu 26.4.2017. <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterDetail.html>

## 10 LIITTEET

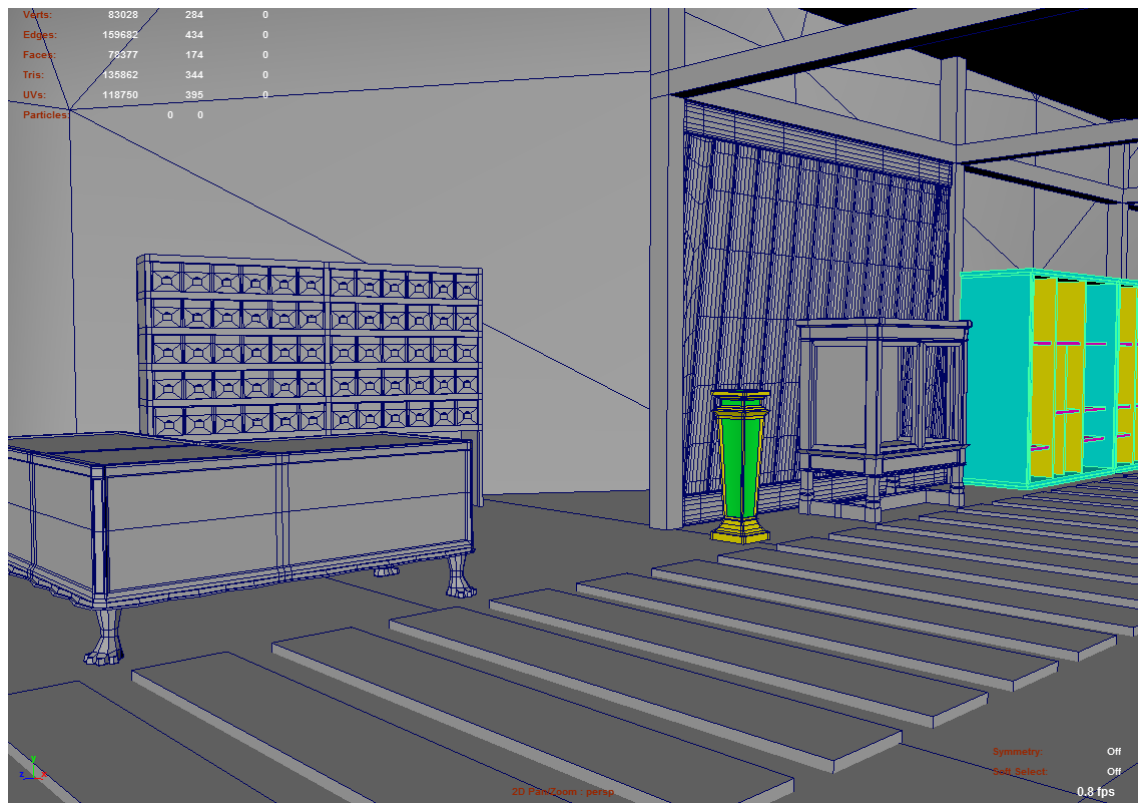
### Liite 1. Kuvia opinnäytetyöprojektista

#### Maya-näkymä juurien repimistä torneista ja silloista

1 (6)



Maya-näkymä kaupan etuosasta. Osa malleista korostettu teksturointia varten. 1 (6)



Unity-näkymä kaupan sisältä

3 (6)





Unity-näkymä juurien repimistä torneista ja silloista

4 (6)



Unity-näkymä taustageometriasta.

5 (6)



Unity-näkymä kaupan sisäänkäynniltä

6 (6)

