



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Joni Hänninen, Janne Harju

VR-pelinkehitys ja Panoraamatekniikan

Hyödyntäminen

Case West Coast Startup

Liiketalous
2016

TIIVISTELMÄ

Tekijä	Joni Hänninen, Janne Harju
Opinnäytetyön nimi	VR-pelinkehitys ja Panoraamatekniikan Hyödyntäminen
Vuosi	2016
Kieli	suomi
Sivumäärä	99
Ohjaaja	Raija Tuomaala, Klaus Salonen

Virtuaalitodellisuus (VR, Virtual Reality) ja erilaiset lisätyn todellisuuden (AR, Augmented Reality) pelit ja ohjelmistot ovat alkaneet nostamaan päätään viimeisten vuosien aikana, ja ne tulevat yleistymään lähitulevaisuudessa merkittävästi VR-laitteiston hintojen laskun myötä, saatavuuden kasvaessa ja laitteistolle kehitettävien ohjelmistojen ja pelien määrän lisääntyessä. Myös panoraamakuvauksen mahdollisuudet lisääntyvät laajalti, sillä kuluttajaystävällisiä sekä -hintoisia laitteita saadaan markkinoille lisääntyvässä määrin. Näitä kahta tekniikkaa yhdistämällä voidaan mahdollisesti luoda aivan uudenlaisia viihdetyylejä sekä ohjelmistoja, joiden potentiaali on jo noteerattu joillain aloilla. Myös Yle on alkanut hyödyntää uutta panoraamakuvauslaitteistoaan sekä tuottamaan mediaa pallopanoraamamuodossa.

Opinnäytetyön tavoitteena oli dokumentoida pelinkehitysprojektin eri työvaiheet, laitteiston hankinnasta panoraamakuvaukseen ja pelin luomiseen. Työssä myös esitellään käytettävä laitteisto sekä ohjelmistot. Tarkoituksena oli myös selvittää panoraamakuvauksen tuoma hyöty sekä mahdollinen käyttöpotentiaali VR-pelien kehittämisessä tulevaisuuden peliteollisuudessa. On toki mahdollista, että tätä tekniikkaa ei voida hyödyntää kyseisellä alalla juuri lainkaan. Koska aihe on melko uusi, jo valmiin tutkimusmateriaalin etsiminen on hankalaa. Teemme kuitenkin yhteistyötä projektin edetessä erinäisten tahojen kanssa, mukaan lukien: MUOVA, Yle ja West Coast Startup. Projektin toimeksiantajana toimii West Coast Startupin keulahahmo, Jussi Loukiainen.

Projektin lopputuloksena saimme aikaan hyvän ja eheän kokonaisuuden ja tavoitteisiin päästiin niin tekijöiden kuin toimeksiantajankin kohdalla.

ABSTRACT

Author	Joni Hänninen, Janne Harju
Title	VR-game Development and Panorama Technology Usage
Year	2016
Language	Finnish
Pages	99
Name of Supervisor	Raija Tuomaala, Klaus Salonen

Virtual reality- and different augmented reality games and programs have started to become more common in the past few years, and they will become significantly more common in near future, because of the price drops of the VR-hardware, growing availability as well as the further development of the games and programs for the hardware. The possibilities of panoramic photography will also grow because of the user-friendly and low priced hardware being shipped to the markets. Combining these two technologies it might be possible to create a completely new style of entertainment and programs, of which some fields of expertise have already taken advantage of. The Finnish broadcasting company Yle has also started to utilize their new panoramic photography hardware and started to produce media in 360 panoramic format.

The aim of this thesis was to document different points of this development project, from hardware acquisition to panoramic photography and creation of the game. There is also an introduction of the hardware and software used in this thesis. The aim was to examine the benefits and potential of panoramic photography in the development of VR-games in the future of the game industry. However, it is also possible that the benefits of using this technique might be little to none in the field of expertise under study. Because the topic is fairly new, the search for already written research material was difficult. However, we co-operated with a variety of companies, for example MUOVA, Yle and West Coast Startup to gain information on the studied area. The employer for this project was Mr. Jussi Loukiainen.

As a final result of this project we made a good and whole product that also reached the goals made by both our employer and ourselves.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KÄSITTEET

1	JOHDANTO.....	12
2	LAITTEISTO JA KÄYTETYT SOVELLUKSET	14
	2.1 HTC Vive.....	14
	2.2 PC-laitteisto.....	15
	2.3 Panoraamalaitteisto	16
	2.4 Unity-pelimoottori	17
	2.4.1 SteamVR liitännäinen Unityssä	17
	2.4.2 C# peliohjelmoinnissa.....	18
	2.4.3 Visual Studio Unityn tukena	18
	2.5 3ds Max.....	18
	2.6 Substance Painter	19
	2.7 Adobe Photoshop & Adobe Premiere Pro	20
	2.8 PTGui Pro	20
	2.9 Audacity.....	21
3	PELIEN SUUNNITTELUPROSESSI	22
	3.1 Mistä aloittaa.....	22
	3.1.1 Aikataulutus ja omat taidot	22
	3.1.2 Tavoitteet ja työsuunnitelma	23
	3.2 Pelisuunnittelu vesiputousmallin mukaisesti	24
	3.2.1 Konsepti; idean kehittäminen, käsikirjoitus, toiminta-ajatus.....	24
	3.2.2 Pelinkonseptin rakentaminen	24
	3.2.3 Kevyt prototyyppi ja testaaminen	24
	3.2.4 Varsinaisen prototyypin rakentaminen	25
	3.3 Pelituotannon osaamisalueet.....	26
	3.3.1 Suunnittelija	26
	3.3.2 Taiteilija	26
	3.3.3 Ohjelmoija.....	26
	3.3.4 Tuottaja	27

4	KÄYTÄNTÖ	28
4.1	Projektin alku ja tehtävien jako.....	28
4.2	Laitteiston hankinta ja kokoonpano	28
4.3	Työtilat, laitteiston käyttöönotto ja työtapa	32
4.4	Peli-idean kehittäminen, käsikirjoitus, perusmekaniikka	34
4.5	Alustan käyttö ja alkuvaiheen prototyypin luominen	35
4.5.1	Uuden projektin luominen.....	35
4.5.2	Tavallisten peliobjektien luonti.....	40
4.5.3	Resurssien tuominen Unityn resurssikaupasta	42
4.5.4	Prefabien luominen muokatusta peliobjektista	44
4.5.5	Skriptojen luominen ja niiden käyttäminen	46
4.5.6	Panoraamakuvien käyttäminen skybox-materiaalina.....	48
4.5.7	Äänien ja videotekstuurien käyttäminen Unityssä.....	50
4.5.8	Particle Systemin käyttäminen Unityssä.....	51
4.6	Audiovisuaalisen materiaalin luominen.....	53
4.7	Panoraamakuvauksen vaiheet	54
4.7.1	Panoraamakuvauksen laitteisto ja kuvien ottaminen	54
4.7.2	Kuvien yhdistäminen ja muuntaminen panoraamaksi	57
4.8	3D-mallien luominen	62
4.8.1	Prototyypit.....	62
4.8.2	Valmiit mallit	66
4.9	Substance Painter	72
4.9.1	Mallien teksturointi Substance Painter 2:ssa.....	72
4.9.2	Materiaalien exporttaus.....	76
4.10	Adobe Photoshop	81
4.11	Adobe Premiere Pro	83
4.12	Audacity	85
4.13	Lopullisen pelin luominen ja digitaalisen materiaalin yhdistäminen	88
4.14	Beetatestaus.....	89
4.15	Muita mahdollisia käyttötarkoituksia panoraamatekniikalle	91
5	YHTEENVETO	94
5.1	Joni Hänninen; päätelmät ja lopputulosten analysointi	94

5.2 Janne Harju; päätelmät ja lopputulosten analysointi	95
5.3 Loppusanat	96
LÄHTEET	98

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. Vesiputousmalli (Roll D6 Games, 2016)	24
Kuvio 2. Komponentit paketeissa	30
Kuvio 3. Normaalista poikkeava kiintolevyjen kiinnitys	31
Kuvio 4. Ongelma suojatarran liiman kanssa	31
Kuvio 5. Kone käynnissä	32
Kuvio 6. HTC Vive lasit ja ohjaimet	33
Kuvio 7. Unityn kirjautumisikkuna	35
Kuvio 8. Projektin aloittaminen	36
Kuvio 9. Uusi projekti	36
Kuvio 10. Unityn perusnäky	37
Kuvio 11. Unityn projekti-ikkuna	39
Kuvio 12. Rakennusasetukset	40
Kuvio 13. Peliobjektien luonti	41
Kuvio 14. Kuutio ja sen komponentit	42
Kuvio 15. Unity asset store	43
Kuvio 16. SteamVR resurssit	44
Kuvio 17. Prefabin luonti	45
Kuvio 18. Prefab-asetusten muuttaminen	45
Kuvio 19. Esimerkkiskriptan pelialueen takkavaloon	46
Kuvio 20. Peliobjektien luominen skriptan avulla	47
Kuvio 21. Panoraamakuvatekstuurin asetukset	49
Kuvio 22. Partikkelisysteemi pelin näyttämöllä	52
Kuvio 23. Partikkelisysteemin muokkausikkuna	52
Kuvio 24. Muokattu partikkelisysteemi	53
Kuvio 25. Neljän ja kuuden kuvan kuvaustekniikka	55
Kuvio 26. Kameran kiinnitys jalustaan	56
Kuvio 27. Kuva laitteiston takaa sekä kameran asetuksista	56
Kuvio 28. Panoraaman ensimmäiset kuvaa ennen suurempia muokkauksia	57
Kuvio 29. Perusnäky PTGui ohjelmassa kuvien latauksen jälkeen	58
Kuvio 30. Näky ilman kiintopisteitä	58
Kuvio 31. Näky kiintopisteiden lisäämisen jälkeen	59

Kuvio 32. Kuvista koostettu panoraama ennen muokkausta	59
Kuvio 33. Valmis panoraama reunojen poiston jälkeen	60
Kuvio 34. Kaksi täysin muokkaamatonta kuvaa mustine pisteineen	61
Kuvio 35. Pano2VR - Trial vesileimat pallopanoraamassa	62
Kuvio 36. Projektin ensimmäinen 3D-malli	63
Kuvio 37. Mökin prototyyppi	64
Kuvio 38. Hirviön ensimmäinen versio luineen osittain läpinäkyvänä	64
Kuvio 39. Embed media toiminto exportauksessa	66
Kuvio 40. Lattiat päällekkäin 3ds Maxissa	67
Kuvio 41. Lattia, seinät ja pönttöuuni	68
Kuvio 42. Saranoiden hahmottelua ja kiinnitysniitit	69
Kuvio 43. Valmis mökki 3ds Maxissa	69
Kuvio 44. Kattoparrun kiinnikkeiden suunnittelua	70
Kuvio 45. Salaman kolme vaihetta: neutraali, välähdys ja lataus	71
Kuvio 46. Ontto karmi seinän rei'ittämisen jälkeen	72
Kuvio 47. Lattia ennen ja jälkeen materiaalin muokkauksen	73
Kuvio 48. Oikealla alareunassa materiaalikartoista valittuna vain korkeus	74
Kuvio 49. Näkymä teksturoidun mökin sisältä	75
Kuvio 50. Saranat teksturoituna	75
Kuvio 51. Lukkopesästä näkee mökin sisälle	76
Kuvio 52. Terästekstuurit takassa ja kattoparrujen kiinnikkeissä	76
Kuvio 53. Baking-ikkuna Substance Painter 2:ssa	77
Kuvio 54. Export-ikkuna Substance Painter 2:ssa	78
Kuvio 55. Malli ilman UVW mappia	79
Kuvio 56. Varjot ”valuneet” pitkin seiiniä	79
Kuvio 57. UVW mapin raja oranssilla viivalla	80
Kuvio 58. Valmis mökki teksturoituna	81
Kuvio 59. Alla logo jota käytimme lopulta pelissä	82
Kuvio 60. Premieren perusnäkyvä menuvideon luonnissa	84
Kuvio 61. Vaihtoehdot eri formaateista exportin yhteydessä	85
Kuvio 62. Aloitusäänien muokkausta	86
Kuvio 63. Microtrack II Audiotallennin	87

Kuvio 64. Audiotallentimen liitännät	87
Kuvio 65. Lopullinen pelialue vanhoilla materiaaleilla	88
Kuvio 66. Lopullinen pelinäyttämö uusilla resursseilla	89
Kuvio 67. Yksinkertaisen pelialueen rakenne	93

OPINNÄYTETYÖSSÄ ESIINTYVÄT KÄSITTEET

Ambient	Tunnelmallinen ”tilan ääni”.
Audiotallennin	Laite jolla tallennetaan ääntä ulkoisista lähteistä.
Bake	Tekstuurin ”leipominen” 3D-malliin.
Bitmap	Yksi pakkaamattoman kuvan tiedostomuodoista (.bmp).
Build	Pelin rakennettu versio joka toimii ilman Unity-ympäristöä.
Collider	Törmäysten laskemiseen tarkoitettu komponentti.
Formaatti	Tiedostomuoto, esimerkiksi: jpg, png, mp4, avi.
Full HD	1920x1080 resoluutio.
Fysiikkamoottori	Ohjelmisto jolla simuloidaan kappaleiden vuorovaikutusta.
Immersio	Immersio, eli ”siellä olemisen” -tunne.
Import & Export	Jonkin tiedoston vienti ja tuonti eri ohjelmien sisällä.
Iso-arvo	Määre joka ilmaisee kennon herkkyyttä valolle.
Kameran aukko	Aukko, josta valo pääsee kameran kennolle.
Laskentateho	Tietokoneen kyky suorittaa laskutoimituksia aikayksikössä.
Layer	Taso, joka tässä yhteydessä sisältää esim. kuvadataa.
Mod/modi	Laitteistojen ja ohjelmien muokkaaminen.
Navigaatiosauva	HTC Viven ohjainkapula.
Overlay	Taso, joka tulee toisen tason päälle peittäen sen osittain.
Pallopanoraama	Panoraamakuva joka on pyöristetty pallon muotoiseksi.

Panoraamakuva	Laajan yleisnäkymän sisältävä kuva.
Pelikonsepti	Ajatus/luonnos pelin perusideasta ja sisällöstä.
Pelimoottori	Videopelin ohjelmistokehys, jonka avulla peli rakennetaan.
Polygoni	Monikulmio.
Prefab	Valmiin luodun peliobjektin pohjapiirros.
Prototyyppi	Jonkin asian ensimmäinen testiversio.
Renderöinti	Kuvan luominen mallista/datasta.
Segmentti	Osa/jako.
Skenaario	Toimintasuunnitelma/-hahmotelma.
Skripti	Ohjelmakoodi, joka ohjaa tietyn asian käyttäytymistä.
Spam/späm	Jonkin asian tiheään tahtiin tekeminen.
Steam	Valve Corporationin digitaalinen PC-pelien alusta.
Suljinaika	Aika, jolloin valo pääsee kameran kennolle.
Texture map	Tekstuurikartta, joka voi sisältää tietoa esimerkiksi väreistä.
Tutoriaali	Internetissä jaettu opas aiheesta x.
Unity	Nykypäivän yleisin ja suosituin pelinkehitysalusta.
User Interface	Käyttöliittymä, eli UI.
UVW map	”Kartta” joka määrittää 3D-mallin tekstuurin koordinaatit.
Virkistystaajuus	Määritelmä siitä, kuinka usein kuva päivittyy sekunnissa.
VR	Virtual Reality, eli virtuaalitodellisuus.

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on dokumentoida ideoimamme virtuaalitodellisuuspelein kehitysprosessi alusta loppuun, eli idean kehittämisestä laitteiston hankintaan, materiaalin luomiseen, demopelin rakentamiseen sekä lopulta varsinaisen pelin rungon kasaamiseen viimeisteltynä materiaaleja käyttäen. Dokumentoinnin ohessa pyrimme selittämään Unityn toimintaa ja miten olemme sitä käyttäneet. Projektin edetessä tarkastellaan myös 360° -panoraamakuvien hyödyntämismahdollisuuksia pelien kehityksessä, sekä kyseistä tekniikkaa käyttämällä luodun visuaalisen materiaalin käyttämistä resurssina itse pelissä.

Opinnäytetyön aiheen valintaan vaikuttivat monet tekijät. Päällimmäisenä niistä olivat laitteiston ja aiheen tuoreus sekä mielenkiinto yleisesti pelialaa ja pelinkehitystä kohtaan. Myös ajankohta oli erinomainen tälle aiheelle, sillä Vaasaan ollaan juuri perustamassa VR-hautomoa yhteistyössä FIVR:n (Finnish Virtual Reality Association) ja IGDA:n (International Game Developers Association) kanssa. Myös Yle on aloittamassa panoraamakuvauksen hyödyntämissä omassa mediasisällössään. Se loi meille erinomaiset edellytykset projektin tekemiseen kehittyvässä ja kasvavassa ympäristössä. Kyseisten ryhmien kanssa yhteistyötä tehdessä avun saaminen tarvittaessa ja kehittyminen puolin ja toisin on odotettavaa.

Projektin ohjelmointi tapahtuu hyödyntäen Unity-pelinkehitysalustaa ja Unitylle luotua SteamVR-liitännäistä sekä -resurssipakettia. Virtuaalitodellisuuslaitteistona toimivat HTC Vive VR -lasit sekä sen mukana tulevat ohjainsauvat. Panoraamakuvien ottamiseen käytämme VAMK:n panoraamakuvakseen tarkoitettua laitteistoa. Ääntenkäsittelyyn käytämme ensisijaisesti Audacity-nimistä ohjelmistoa. 3D-mallien luomiseen sekä osittaiseen teksturointiin käytämme 3DS Maxia. Edistyneempään teksturointiin käytetään Substance Painter 2:a. Pelin ohjelmointikielenä toimii C#.

Projektimme aukeaa parhaiten jo hieman aiheeseen vihkiytyneelle henkilölle. Pyrimme kuitenkin rakentamaan opinnäytetyön siten, että aiheesta kiinnostuneetkin saisivat vastauksen keskeisimpiin kysymyksiin, koskivat ne sitten pelinkehityksen

prosessia, panoraamakuvauksen hyödyntämistä peleissä, VR-pelien suunnittelu-
prosessia tai laitteistoon liittyviä asioita. Toivomme tämän opinnäytetyön myös in-
nostavan tulevia opiskelijoita aiheeseen liittyviin jatkotutkimuksiin tai vastaaviin
projekteihin.

West Coast Startupin tavoitteena on auttaa koulujen henkilökuntaa ja oppilaita lii-
ketoiminnan käynnistämässä. Yrittäjyysshautomon keulahahmona toimii Jussi
Loukiainen. Sen tarjoamat palvelut ovat maksuttomia Vaasan yliopiston ja Vaasan
ammattikorkeakoulun henkilökunnalle. West Coast Startup ja Jussi Loukiainen toi-
mivat loistavasti projektimme toimeksiantajina. (Muova 2015)

Työnjaon projektin osalta sovimme siten, että Janne Harju vastaa projektin audio-
visuaalisesta puolesta ja Joni Hänninen ohjelmoinnista sekä Unityn käytöstä. Lait-
teiston osalta Janne vastaa PC-tekniikasta ja Joni VR-laitteistosta. Pelisuunnitte-
lusta sekä käsikirjoituksesta vastaamme molemmat.

2 LAITTEISTO JA KÄYTETYT SOVELLUKSET

Tässä luvussa käsittelemme käytettyjä laitteita ja sovelluksia. Käymme myöskin läpi niissä hyödynnettyjä keskeisimpiä tekniikoita.

2.1 HTC Vive

HTC Vive on HTC:n ja Valve Corporationin yhteistyössä tekemä ja valmistama virtuaalitodellisuusjärjestelmä, johon kuuluvat kaksi sauvaa joilla seurataan käsien liikettä, virtuaalilasit sekä kaksi laserpohjaista sensorimajakkaa, joilla seurataan sauvojen ja lasien liikettä ja paikkaa pelialueella. Koko pakettia kutsutaan myös SteamVR:ksi. Järjestelmä on suunniteltu luomaan virtuaalinen pelialue majakoiden avulla, jossa käyttäjä voi olla vuorovaikutuksessa laseihin piirretyn maailman kanssa sauvojen avulla. Järjestelmä mahdollistaa myös liikkumisen noin neljän ja puolen metrin alueella. Laitteistossa on myöskin yli 70 erilaista sensoria, mukaan lukien gyroskooppi ja laserpaikannussensori. Näiden laitteiden ja sensorien avulla saadaan luotua immersiivinen ja tarkka peliympäristö. (D’Orazio ym. 2015; Kelion 2015; Orland 2016)

Virtuaalilaseissa on kaksi pientä näyttöä, yksi kullekin silmälle. Näyttöjen virkistystaajuus on 90 hertsiä ja niiden yhteisresoluutio on 2160x1200. Molemmat näytöt kykenevät Full HD -kuvanlaatuun. Korkean virkistystaajuuden vuoksi vaaditaan noin kolminkertainen tehokkuus verrattuna pelaamiseen Full HD -kuvanlaadulla ja 60 FPS -ruudunpäivityksellä. (Kypreos 2016)

Viveä käytettäessä emme tunteneet lainkaan ”liikepahoinvointia”, sillä laitteiston toiminta perustuu pääosin luonnollisiin liikkeisiin, jolloin tavaroita esimerkiksi poimitaan fyysisesti virtuaalisessa ympäristössä, toisin kuin kilpailijansa Oculus ja Playstation VR. Toisaalta taas Viven käyttömukavuus kärsi, sillä päähän asetettava laitteisto oli huomattavasti epämukavampi ja painavampi kuin kilpailijoiden vastaavat. Lisäksi kuulokkeet oli yhdistettävä erikseen, mikä sekin uhrasi hieman käyttömukavuutta verrattuna esimerkiksi Oculus Riftin integroituihin kuulokkeisiin. (Kypreos 2016)

2.2 PC-laitteisto

Laitteiston kokoonpano suunnattiin ensisijaisesti VR-pelien kehittämiseen sekä pelaamiseen. Sen suunnittelun sekä keskusyksiköiden kokoonpanon hoiti Janne. VR-pelien pelaamiseen soveltuvaa tietokonetta kootessa tulee ottaa huomioon jo aiemmassa kappaleessa mainittu VR-laitteiston vaatima teho. Normaalit PC-pelit jo periaatteessa vakiintuneella 1080p (Full HD) tarkkuudella vaativat huomattavasti vähemmän suorituskykyä kuin VR-pelit.

Otetaan esimerkiksi normaali PC-peli nykypäivän vakiintuneilla standardeilla eli 1080p:n tarkkuudella sekä 60Hz:n virkistystaajuudella ja verrataan sitä HTC Viveen. Jos puhutaan puhtaasti siitä, kuinka paljon tehoa vaaditaan, jotta voidaan renderöidä eli ”piirtää” liikkuvaa kuvaa normaaleissa PC-peleissä 1920x1080 resoluutiolla ja 60Hz:n virkistystaajuudella, tulee näytönohjaimen renderöidä 124 miljoonaa varjostettua pikseliä sekunnissa. Jos resoluutiona käytettäisiin normaalissa PC-pelissä yleisesti VR-peleissä käytettyä 2160x1200 resoluutiota sekä 90Hz virkistystaajuutta ja kuva jaettaisiin kahdelle näytölle kuten VR-laseissa, tulisi näytönohjaimen kyetä renderöimään 233 miljoonaa varjostettua pikseliä sekunnissa. Itse VR-peleissä tehoa vaaditaan vieläkin enemmän ja silloin näytönohjaimen tulee kyetä renderöimään 400 miljoonaa varjostettua pikseliä sekunnissa. VR-pelit vaativat siis kolminkertaisen suorituskyvyn näytönohjaimelta verrattuna normaalien PC-pelien vakiintuneisiin 1080p/60Hz spekseihin. (Oculus 2015)

Tietokoneet siis koottiin juuri VR-käyttöä varten. Tärkeimmät kriteerit olivat tarvittava teho, nykyaikainen tekniikka, päivitettävyyys, luotettavuus, käyttömukavuus sekä tietysti järkevä hinta. Kokoonpano koostui pääsääntöisesti Jimm’s:n verkko-kaupasta tilatuista osista, pois lukien kotelot sekä prosessorit, jotka tilattiin paikallisesta Multitronic ATK-liikkeestä. Alla lista osista sekä niiden hinnoista syksyllä 2016:

- ❖ emolevy: Asus Z170-A (159,00 €)
- ❖ näytönohjain: Asus Strix GTX 970 4GB (249,00 €)
- ❖ virtalähde: Seasonic 620W M12II-620 Evolution (89,90 €)
- ❖ kiintolevy 1: Western Digital 1TB Caviar Black HDD (89,90 €)

- ❖ kiintolevy 2: Samsung 250GB 750 Evo SSD (79,00 €)
- ❖ keskusmuisti: Kingston 16GB HyperX Fury (89,00 €)
- ❖ prosessori: Intel Skylake Core i5-6600K (259,00 €)
- ❖ prosessorin jäähdytin: Thermalright True Spirit 120 (34,90 €)
- ❖ kotelo: Fractal Design Define R5 Black (109,90 €)
- ❖ reititin: Asus 4G-AC55U-N (199,00 €)
- ❖ näyttö: Asus VS247HR 23.6" (149,00 €)
- ❖ näppäimistö ja hiiri: Logitech Wireless Combo MK520 (44,90 €)
- ❖ vr-laitteisto 1: HTC Vive (979,90 €)
- ❖ vr laitteisto 2: Oculus Rift (749,90 €)
- ❖ käyttöjärjestelmä: Windows 10 Enterprise 64bit

Kahden tietokoneen hinta oheislaitteineen oli yhteensä: 4834,80 €

VR-pelien kehittämiseen vaadittavan laitteiston kokoamiseen tulee siis olla kohtuullisen suuri budjetti. Toki tässä olisi voitu säästää noin 20% keskusyksiköiden hinnasta tilaamalla valtaosa komponenteista saksalaisesta verkkokaupasta, esim. mindfactory.de:stä. Siellä osien hinnat ovat siis käytännössä poikkeuksetta edullisempia. Aikataulun takia päädyimme kuitenkin suomalaiseen verkkokauppaan.

Koneiden sekä VR-laitteiston hankinnat teki toimeksiantajamme Jussi Loukiainen, jonka kautta saimme myös Win 10 -käyttöjärjestelmän, Office 365 -ohjelmiston ja tarvittavat oheislaitteet.

2.3 Panoraamalaitteisto

Panoraamalaitteistoon sisältyi:

- ❖ Canon EOS 50D -järjestelmäkamera
- ❖ Sigma EX 8mm kalansilmäobjektiivi
- ❖ Nodal Ninja 3 MK II panoraamaapää
- ❖ Manfrotto Neotec kolmijalka ja
- ❖ Canon RC-6 kaukolaukaisin

Kyseisellä laitteistolla otimme kuvat, joista koostimme taustan peliimme yhdistämällä ne yhtenäiseksi pallopanoraamaksi käyttämällä PTGui- sekä Unity-ohjelmia. Kuvat otimme .jpg ja .png -formaateissa.

2.4 Unity-pelimoottori

Unity on Unity Technologiesin kehittämä pelimoottori, jolla voidaan helposti luoda kaksi ja kolmiulotteisia pelejä usealle alustalle, mukaan lukien PC, konsolit sekä älylaitteet ja selaimet. Tässä opinnäytteessä käytämme Unityn ilmaista versiota, mutta tarjolla ovat myös maksullinen pro-versio sekä erillinen yrityksille suunnattu Unity-lisenssi. Unityn fysiikkamoottorina toimii Nvidia PhysX. (Unity Technologies 2016 a)

Unityn vahvuuksiin kuuluu sen helppokäyttöisyys, ilmaisuus sekä hyvin kattava Unity Asset Store, josta käyttäjät voivat ladata ilmaisia ja maksullisia resursseja, kuten 3D-malleja, tekstuureita sekä materiaaleja. Myös tähän opinnäytetyöhön tarvittavat SteamVR -resurssit löytyivät ilmaiseksi Unityn resurssikaupasta. Unityn hyvä dokumentaatio ja kattavat tutorialvideot lisäävät helppokäyttöisyyttä ja ohjelmiston lähestyttävyyttä. Nämä seikat vaikuttivat myöskin meidän valintaamme käyttää kyseistä pelimoottoria tässä opinnäytetyössä. (Unity Technologies 2016 b)

Unityssä voidaan käyttää useita eri ohjelmointikieliä, jotka ovat JavaScript (UnityScript), Boo sekä C#. Valitsimme C#:n ohjelmointikieleksemme tässä projektissa aikaisemman kokemuksemme ja koulutuksen vuoksi. C# oli meille tutuin käytetty kieli ja koulussa olimme opiskelleet tätä kieltä useilla kursseilla. (Unity Technologies 2016 c)

Käytössämme tässä projektissa on Unity 5.x, joka on Unityn uusin versio. Tähän versioon on tuotu mm. uudistettu PhysX-fysiikkamoottori sekä sisäinen äänimikseri. (Unity Technologies 2016 d)

2.4.1 SteamVR liitännäinen Unityssä

SteamVR -liitännäinen sisältää kaiken tarvittavan, jotta Unityllä voitaisiin rakentaa toimiva HTC Vivellä toimiva peli. Yksi tärkeimmistä resursseista paketissa on niin

sanottu CameraRig, joka lisätään aina pelialueelle. CameraRig sisältää pelialueen, molemmat ohjaimet, tarvittavat 3D-mallit sekä ohjelmakoodit sekä ohjaimille että majakoille. Tämän liitännäisen vuoksi pelien kehittäminen HTC Vivelle helpottuu huomattavasti. Se vaatii toimiakseen Unity version 4.7.1 tai uudemman sekä ajon aikaisen ympäristön, SteamVR Runtimen, joka asennetaan suoraan Steam verkkopalvelun sovelluksen avulla. Steam VR tukee myöskin DirectX 11 Windows 7 ylöspäin. (Unity Technologies 2016 e)

2.4.2 C# peliohjelmoinnissa

C# muistuttaa Javaa ja muita C-kieliä, ja on käytetyin kieli Unityllä ohjelmoitaessa. C#:lla on myöskin parempi suorituskyky kuin muilla Unityssä käytettävillä kielillä, ja se pärjää stressitesteissä paremmin kuin kilpailijansa. Aiempi kokemus ja nämä seikat vaikuttivat tämän kielen valintaan. (Russell, 2016)

2.4.3 Visual Studio Unityn tukena

Visual Studio on Microsoftin kehittämä kehitysympäristö, jossa voidaan käyttää useita kieliä, mukaan lukien käyttämämme C#-ohjelmointikieli. Visual Studio tarjoaa tehokkaan kääntäjän ja debuggerin, jota on helppo käyttää Unityn ohjelmakoodien muokkaamiseen ja tarkastamiseen. Visual Studio osoittaa käyttäjälle virheet ohjelmakoodissa, kertoo rivin ja ehdottaa ratkaisua. Myös ikkunoiden asettelu on hyvin intuitiivista, ja tuntuu hyvin luonnolliselta. Kehitysympäristö toimii erittäin hyvin yhteen Unityn kanssa. Myös Visual Studion käyttö perustuu aiempiin kokemuksiimme. (Microsoft, 2016)

2.5 3ds Max

3ds Max on Autodesk-nimisen yrityksen valmistama 3D-ohjelma, joka on tarkoitettu 3D-mallien luomiseen sekä käsittelyyn. Päädyimme 3ds Maxiin aiemman kokemuksen perusteella, sekä siksi, että se on yhteensopiva Unityn kanssa. Tämä helpottaa tutoriaalien löytämistä tarpeen vaatiessa.

3D-ohjelmat ovat pääsääntöisesti myös kohtuullisen kalliita ja 3ds Maxista löytyy educational-versio, joka on opetustarkoitukseen tehty ilmainen lisenssi. Mietimme

projektia aloittaessamme vaihtoehtoisesti Blenderin käyttämistä, ja aiheeseen tarkempi tutustuminen paljasti myöhemmin, että Autodeskin valmistamasta Maya nimisestä 3D-ohjelmastakin löytyy ilmainen versio. Molemmat ko. ohjelmista ovat myös Unityn kanssa yhteensopivia ja tutoriaalien löytäminen niille on kohtuullisen helppoa. (Autodesk, 2017)

3ds Max tarjoaa hyvät työkalut 3D-mallien luomisen sekä muokkaamisen lisäksi myös niiden teksturointiin. Tekstuureina voidaan käyttää 3ds Max:n kohtuullisen kattavaa omaa materiaalikirjastoa, tai esimerkiksi kuvia. Valmiin mallin voi viedä (export) 3ds Maxista, jonka jälkeen se voidaan tuoda (import) toiseen ohjelmaan. Käytimme projektissa .fbx-formaattia, sillä se oli yhteensopiva sekä Unityn että Substance Painter 2:n kanssa.

2.6 Substance Painter

3D-mallien hieman edistyneempään teksturointiin käytettiin Allegorithmic nimisen yrityksen kehittämää Substance Painter 2-ohjelmistoa. Ohjelma on maksullinen ja ostohetkellä se oli 33 %:n alennuksessa, jolloin hinnaksi jäi 93,79 €. Tässä tapauksessa ohjelmiston lataus sekä käyttö tapahtuvat Steamien kautta digitaalisessa muodossa. Substance Painter 2 on tarkoitettu 3D-mallien teksturointiin sekä tekstuurien muokkaamiseen. Täysin uusia tekstuureja ei kyseisellä ohjelmalla ole mahdollista luoda, mutta siihen tarkoitukseen löytyy samasta tuoteperheestä Substance Designer-niminen ohjelma. Lopulta päädyttiin Substance Painterin ostamiseen, sillä sen koettiin olevan hyödyllinen sekä nyt että tulevaisuudessa. (Valve Corporation, 2017)

Erikoisuutena Substance Painter 2:ssa on mahdollisuus maalata käsin kaikille ”tekstuurikanaville”, jolloin 3D-malliin voidaan siis pelkän värin sijasta samalla laittaa tekstuuri, jossa esiintyy esimerkiksi kulumaa, karheutta, naarmuja jne. Ohjelma mahdollistaa myös tekstuurien ”leipomisen” (bake) malliin, jolloin on mahdollista saada ohjelmasta ulos .bmp tai .png -formaatin tekstuurikarttoja (texture map), joissa kaikki materiaalien sisältö on eriteltynä kuviin. Tekstuurimapid voidaan sitten helposti asettaa paikalleen esimerkiksi Unityn sisällä. Unity tukee sekä .bmp että .png -formaatteja, ja tässä projektissa päädyimme käyttämään 8-bittistä bitmappia

materiaalitiedostoina. Kyseisen formaatin käyttöä suositeltiin useissa tutoriaaleissa, joista tärkeimpänä mainittakoon valmistajan Youtubeen lataama video aiheesta. Molemmat ko. formaateista ovat häviöttömiä, jolloin kuvan laatu ei siis kärsi tallennusten ohessa. Suurin ero näiden kahden formaatin välillä onkin se, että .bmp ei ole pakattu, kun taas .png on. Tämä näkyy tiedoston koossa, jolloin .bmp-formaatin kuvatiedosto on kooltaan hieman suurempi. (Allegorithmic, 2017. Allegorithmic, 2016a. Guiding Tech, 2014)

Substance Painter 2:sta käytetään myös lukuisissa ammattitason projekteissa ja peleissä. Tunnetuin niistä taitaa olla Uncharted 4: A Thief's End, joka julkaistiin Playstation 4 konsolille toukokuussa 2016. (Allegorithmic, 2016b)

2.7 Adobe Photoshop & Adobe Premiere Pro

Photoshop ja Premiere Pro ovat Adoben valmistamia ohjelmia. Ohjelmien uusimmat versiot ovat Photoshop CC 2017 ja Premiere Pro CC 2017. Sekä Photoshopin, että Premiere Pro:n käyttöoikeudet voi ostaa Adoben sivuilta opiskelijalisenssillä, johon Adobe vaatii luonnollisesti oppilaitosyhteyden. Sopimus vuoden mittainen ja kuukausittainen veloitus on 20,15€. Ohjelmien lataus sekä asennus tapahtuvat Adobe Creative Cloud-ohjelmiston kautta ja edellä mainittu sopimus sisältää oikeudet myös muihin Adoben ohjelmiin, joita on yli kaksikymmentä. Toinen vaihtoehtoisuus hankkimiselle on kuukausisopimus. Tällöin hinta on korkeampi, mutta vuosittaista sopimusta ei tarvitse tehdä. Esimerkiksi Photoshopin hinta kuukausisopimukselle yksityishenkilölle on 37,19€/kk. Vaasan ammattikorkeakoululta löytyy myös mahdollisuus molempien ohjelmien käyttämiseen luokassa: WB0028. (Adobe Systems, 2017)

2.8 PTGui Pro

PTGui Pro on New House Internet Services BV -nimisen yrityksen kehittämä sovellus panoraamakuvien muokkaamiseen sekä luomiseen. Se antaa hyvät työkalut esimerkiksi useasta kuvasta saumattoman panoraaman yhdistämiseen ja sen muuntamiseen pallopanoraamaksi. Ohjelman lisenssin voi yksityishenkilö ladata valmis-

tajan sivuilta hintaan 79 €. Lisenssillä ei ole aikarajaa, sitä voi käyttää myös kaupalliseen toimintaan ilman rajoituksia ja se sisältää myös yhden vuoden ilmaiset päivitykset. Ohjelmasta on tarjolla myös Pro-versio, joka sisältää perusversiota enemmän ominaisuuksia. Lisenssi ohjelmaan löytyi VAMK:n luokasta WB0028, eli samasta luokasta, jossa oli mahdollista käyttää myös aiemmin mainittuja Adoben ohjelmia. (PTGui, 2017)

2.9 Audacity

Äänenkäsittelyyn projektissa käytettiin The Audacity Teamin kehittämää Audacity-nimistä ohjelmistoa. Audacity on ilmainen sekä kevyt ohjelma, jonka käytöstä oli myös aiempaa kokemusta. Osa efekteistä haettiin freesound.org-nimiseltä sivustolta siten, että sen muokkaaminen ja käyttäminen ”ei kaupallisissa” -projekteissa olisi tekijän valitseman Creative Commons -lisenssin puolesta sallittua. Tiettyjä sääntöjä liittyy toki kaiken ulkopuolisen materiaalin käyttämiseen, josta yleisin ilmaisen materiaalin kanssa on tekijän nimeäminen ja/tai ilmoittaminen materiaalin käytöstä sen alkuperäiselle tekijälle. Musiikit peliin etsittiin freemusicarchive.org-nimiseltä sivustolta. Samat lisensointiin liittyvät säännöt pätevät toki myös musiikkiin. (Creative Commons, 2017)

Toinen puoli efekteistä luotiin itse käyttämällä M-Audio Microtrack II-nimistä audiotallenninta. Microtrack II on pienehkö ja helppokäyttöinen audiotallennin, jota käytimme myös edellisessä peliprojektissamme syksyllä 2014. Pakettiin kuuluu tallennin, irrotettava stereomikrofoni, muistikortti ja latausjohto. Audiotallennin osoittautuikin jo edellisessä projektissa erittäin toimivaksi ratkaisuksi efektien sekä ambientin, eli ”ympäröivän taustaäänien” tallentamisessa. Microtrack II mahdollistaa myös materiaalin kuuntelun paikan päällä kuulokkeiden kautta, jolla pystyy helposti varmistamaan materiaalin tasalaatuisuuden. Tähän projektiin luotiin mm. osa pelin ääniefekteistä ja ambientista kyseisen tallentimen avulla.

3 PELIEN SUUNNITTELUPROSESSI

Tässä luvussa käsittelemme pelien suunnitteluprosessia aina idean kehittämisestä lopulliseen tuotteeseen. Luvussa käydään kattavasti läpi yleispätevät periaatteet, mitä kannattaa noudattaa pelin ideaa kehiteltäessä ja sitä jalostettaessa, jotta prosessi sujuisi mutkattomasti ja säästyttäisiin ylimääräisiltä viivästyksiltä. Luvussa käsitellään myös peliprojektin keskeisiä rooleja ja mitä roolien tehtäviin kuuluu.

3.1 Mistä aloittaa

Ennen kuin peliprojektia aletaan työstämään, on tärkeää luoda pelille selkeät rajat, jonka sisällä pysytään projektin aikana. Myös omat rajat on tärkeä hahmottaa ennen projektin alkua. Useimmat harrastushankkeet ja jotkin kaupallisetkin hankkeet kaatuvat siihen, että yritetään jotain, mihin omat kyvyt tai laitteisto eivät pysty. Joskus ei edes tiedetä, mihin päämäärään pyritään, minkä vuoksi projektit paisuvat liian suuriksi. ”Tehdään peli” ei ole päätös jolla projekti aloitetaan, vaan sitä edeltää useita vaiheita. (Vuorela 2007, 36)

3.1.1 Aikataulutus ja omat taidot

Pelintekijän täytyy tuntea omat sekä muiden tiimin jäsenten kyvyt ja niiden rajat. On suositeltavaa tehdä jokaisesta tiimin jäsenestä pieni lista, joka auttaa taitojen kartoituksessa. (Vuorela 2007, 37-39)

Myös aikataulu tulee pitää kunnossa. Mikäli kyseessä on harrastelijaprojekti, tavoitetulos voi ajanpuutteen vuoksi venyä. Harrasteluprojekteissa myös työ voi olla katkonaista, jolloin ei enää muisteta mitä viimeksi tehtiin. Harrastusprojektille aikataulujen tekeminen ei tunnu välttämättä lainkaan mielekkäältä, mutta aika on kallisarvoinen resurssi siinä missä rahakin. Niiden puute voi johtaa projektin kaatumiseen. Nyrkkisääntönä voidaan pitää, että harrastushankkeille aikaa riittää päivittäin keskimäärin 30 minuuttia, mutta tekeminen ei ole tasaista, vaan joinakin päivinä voi tehdä ja opetella pari tuntia ja toisinaan ei ehdi ollenkaan. Suhteessa yksi viikko ammattilaisen työtä voi vastata esimerkiksi kymmentä viikkoa harrastelijatyötä.

Tämä kannattaa ottaa huomioon projektin aikataulua suunnitellessa. (Vuorela 2007, 39-40)

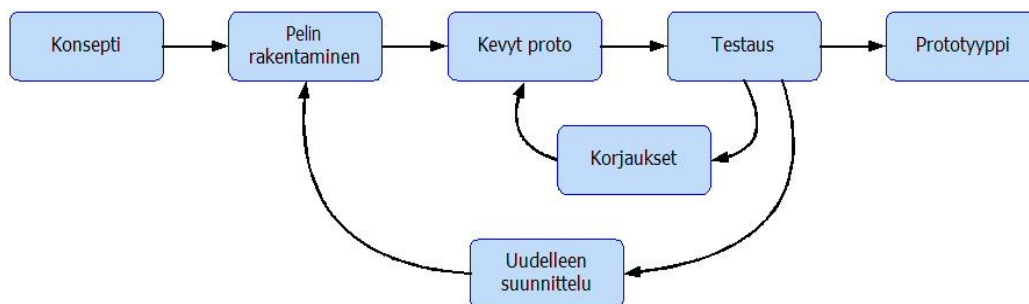
3.1.2 Tavoitteet ja työsuunnitelma

Kun omat taidot ja aikataulu ovat selvillä, tulisi päättää projektin tavoitteet. Mille ja minkä kokoiselle kohderyhmälle peli halutaan julkaista? Tähän kysymykseen kuuluvat myös sosiaaliset, taloudelliset ja kulttuurilliset tekijät. Myös kilpailuasetelma on syytä saada selville; kehitänpö pelin esimerkiksi Mac- vai PC-laitteille, joissa kilpailijoita on erisuuruinen määrä. (Vuorela 2007, 39-40)

Kilpailu on tärkeää. Mikäli sitä ei ole, voidaan tehdä päätelmä että suunniteltua kohdeyleisöä ei ole olemassa, tai sille ei kannata kehittää pelejä. Myös suuret kohdemarkkinat ovat hyvä valinta, sillä ne liikkuvat tukeakseen useampaa tuotetta. Myös muut esimerkiksi samaan genreen liittyvät pelit voivat tukea toisiaan nostamalla oman genrensä myyntiä pelistä riippumatta pitkällä aikavälillä. Yksi superhitti voi nostaa muiden vastaavan genren pelien myyntiä. (Vuorela 2007, 40-41)

Kun tavoitteet ovat selvillä, on aika rakentaa työsuunnitelma. Se kuulostaa tylsältä, mutta projektia työstettäessä säästetään paljon aikaa ja muita resursseja, mikäli liikutaan työsuunnitelman mukaisesti. Pelistä riippuen työsuunnitelmat saattavat poiketa toisistaan, mutta on myös olemassa paljon yleispäteviä työsuunnitelmia, joita voidaan hyödyntää, kun rakennetaan omaa suunnitelmaa. Yleensä suunnitelmaan kuuluu kuitenkin kolme pääasiallista palasta, esituotanto, tuotanto ja jälkituotanto. Esituotantoon kuuluu konseptointi, resursointi, tuotespeksi ja tehtävänjako. Varsinaiseen tuotantoon kuuluu rungon ja sisällön kehittäminen sekä testaus. Jälkituotantoon kuuluvat markkinointi ja levitystavan kehitys. (Vuorela 2007, 40-41)

3.2 Pelisuunnittelu vesiputousmallin mukaisesti



Kuvio 1. Vesiputousmalli (Roll D6 Games 2016)

Vesiputousmalli on ohjelmistojen tuotantoon tarkoitettu prosessimalli, jossa vaiheesta toiseen siirrytään alaspäin vesiputouksen tavoin. Tarvittaessa palataan aina takaisin edelliseen vaiheeseen. Pelin laajuudesta riippuen prosessissa voi mennä viikoista useisiin vuosiin. (Roll D6 Games 2016)

3.2.1 Konsepti; idean kehittäminen, käsikirjoitus, toiminta-ajatus

Konseptissa on kyse pelin alkuvaiheessa tapahtuvasta suunnittelusta. Pelillä saattaa olla hyvä pohjaidea, esimerkiksi nimi, jota lähdetään kehittämään eteenpäin. Myös käsikirjoitus on tehtävä projektin tässä vaiheessa. Se helpottaa idean jalostamista seuraaviin vaiheisiin. Tässä vaiheessa projektia määritellään myös, mikä on pelin idea ja mistä pelissä on kyse, esimerkiksi se, onko kyseessä räiskintä- vai strategia-peli. (Roll D6 Games 2016)

3.2.2 Pelinkonseptin rakentaminen

Tämä ei tarkoita fyysisen pelin koodaamista eikä edes prototyypin rakentamista. Tässä vaiheessa otetaan konseptissa rakennetut ideat ja hiotaan niitä. Oleellisinta olisi saada peliin luotua perusmekaniikat, mistä saadaan pisteitä, mistä pelaajaa rangaistaan, miten pelin voittaa ja miten häviää. (Roll D6 Games 2016)

3.2.3 Kevyt prototyyppi ja testaaminen

Kevyt prototyyppi tarkoittaa tässä tilanteessa pelin olennaisimpien mekaniikkojen testaamista. Pyritään kasaamaan mahdollisimman kevytrakenteinen peli käyttämällä esimerkiksi Unitystä löytyviä ”primitive object” objekteja, joilla testataan

konseptin toimivuutta. Testaus voidaan suorittaa myös eri pelin osille erikseen. Tärkeintä on saada tässä vaiheessa pelimekaanisesti tasapainoinen ja selkeä paketti aikaiseksi. Mikäli tavoitteita ei saavuteta, hiotaan mekaniikkaidea paremmaksi ja rakennetaan uusi kevyt prototyyppi. (Roll D6 Games 2016)

Projektin tässä vaiheessa voidaan siis luoda erilaisia prototyyppejä tai demoja tiettyyn tarkoitukseen. tärkeimpiä näistä ovat esimerkiksi pelikenttien korvikkeet (mock-ups) ja pelattavat demot. Usein ei kuitenkaan ole mahdollisuutta rakentaa edes demon mittaista peliä suunnittelun apuvälineeksi, ja silloin tulisi käyttää esimerkiksi edellä mainittuja edullisia prototyyppejä (low-tech prototypes) ja idea- tai konseptidemoja (concept demos). (Manninen 2007, 174)

Edulliset prototyypit voivat olla millaisia tahansa edullisella budjetilla tuotettuja prototyyppejä. Prototyypin rakentamiseen ei saa mennä liian kauan aikaa. Toisaalta prototyypin rakentamisessa täytyy ottaa huomioon projektin ja idean laajuus ja monimutkaisuus, sillä monipuolisemmalla prototyypillä voidaan säästää tuotantovaiheen kustannuksissa. (Manninen 2007, 175)

Ideademo on hieman jalostetumpi prototyypin versio, jolla voitaisiin viitata seuraavaksi esiteltävään varsinaiseen prototyyppiin. Se on osittain toteutettu peli, jota käytetään peli-idean esittelyssä, kypsyyden arvioinnissa ja etenemisen kuvaamisessa. Ideademo voi olla myös hyvin paljon erilainen kuin lopullinen julkaisu. Sen tarkoitus on välittää konseptin ydinajatuksia vuorovaikutteisella tavalla. Ideademoilla pelinkehittäjät voivat myös hankkia rahoitusta projektilleen. Jalostetumpi prototyyppi pelistä luo uskottavuutta pelisuunnittelijan ja koko tuotantotiimin kyvyistä. (Manninen 2007 176)

3.2.4 Varsinaisen prototyypin rakentaminen

Prototyyppi edustaa sitä ideaa, mikä on testaamisen ja uudelleensuunnittelun sekä konsepti-ideoiden kautta rakentunut pelin uudeksi konseptiksi. Varsinaisessa prototyypissä pyritään rakentamaan mahdollisimman valmis paketti, jota on helppo pa-

rannella omilla digitaalisilla resursseilla. Luodaan ikään kuin pelin runko tai selkäranka, jonka päälle lisätään omia 3D-malleja, ääniä, kuvia sekä tekstuureita. (Roll D6 Games 2016)

3.3 Pelituotannon osaamisalueet

Nykypäivän pelit tehdään tiimityönä, ja on tärkeää, että tiimistä löytyy eri osaamisalueita hallitsevia jäseniä. Harva yksittäinen henkilö taitaa kaiken pelintekemiseen liittyvän mutta tiimin jäsenet voivat täydentää toisiaan tehokkaasti.

3.3.1 Suunnittelija

Suunnittelijan (designer) rooliin sisältyy paljon eri asioita. Suunnittelija voi tarkastella tuotetta esimerkiksi esteettisestä näkökulmasta, tutkia käytettävyyttä ja toimintaa sekä ilmettä. Suunnittelija voi myös selvittää ja määritellä toteutettavan tuotannon toiminnallisuutta, ulkonäköä ja muita tarvittavia yksityiskohtia. Pelisuunnittelija voi suunnitella esimerkiksi pelin olemuksen ja pelimekaniikan ytimen, sekä vision, jonka hän välittää tiimin muille jäsenille, jotka pyrkivät toteuttamaan pelin ja sen sisällön vision ja toimintatapojen mukaisesti. Suunnittelija voi myös hioa ideaa parempaan suuntaan, mikäli testauksen yhteydessä ilmenee esimerkiksi tasapaino-ongelmia. Pääasiassa pelisuunnittelijan rooli on kuitenkin vahvasti viestinnällinen ja kommunikoiava. (Manninen 2007, 46-47)

3.3.2 Taiteilija

Taiteilija (artist) on tehtävä, joka sisältää useita hyvinkin erilaisia osaamisalueita. Taiteilija toteuttaa projektin aikana erilaisia resursseja, joita voidaan käyttää valmiin pelin luomiseen. Näihin kuuluvat esimerkiksi erilaiset tekstuurit, 3D-mallit, äänitehosteet, musiikki ja animaatiot. Myös pelin käsikirjoitus voi olla taiteilijan toiminta-alueella. Taiteilijalla tulee olla luova ote työhön. (Manninen 2007, 49)

3.3.3 Ohjelmoija

Ohjelmoija (programmer) on henkilö, joka toteuttaa järjestelmän, jolla taiteilijan ja suunnittelijan ideoiden ilmaisu onnistuu. Ohjelmoija luo pelin taustalle vision ja

ideoiden mukaisen toiminnan ja mekaniikan, jonka päälle liimataan taiteilijan luomat audiovisuaaliset tuotokset. Ohjelmoija ymmärtää välineensä ja kykyjensä rajat, joten kritiikin ja epäilyiden jakaminen tiimin muille jäsenille lankeaa ohjelmoijalle. Ohjelmointiin liittyy yleensä vahva hyödyn tavoittelu, ja päämäärään tulee päästä keinolla millä hyvänsä. Esteettisyys ei siis ole tässä tapauksessa niin tärkeää kuin muissa tiimin rooleissa, sillä usein ohjelmakoodi jää taustalle näkymättömiin, kunhan pelin toiminta on sillä tasolla kuin sitä vaaditaan. (Manninen 2007, 48)

3.3.4 Tuottaja

Tuottaja (producer) huolehtii usein peliprojektissa tiimin hallinnoinnista. Tuottajan tehtäviin kuuluu mm. rahoituksen hankkiminen, henkilöstörekrytointi, tiimin resursointi, projektin aikatauluttaminen, seuranta ja raportointi sekä tiimin ja toiminnan hallinta. (Manninen 2007, 50)

4 KÄYTÄNTÖ

Tässä luvussa dokumentoimme projektin vaiheet. Kerromme myös seikkaperäisesti vaiheiden sisällön, käytetyt toimintatavat ja mallit.

4.1 Projektin alku ja tehtävien jako

Vaasan ammattikorkeakoulun hyväksytyä aiheemme kävimme projektin toimeksiantajan, Jussi Loukiaisen kanssa käynnistyspalaverin, jolloin sovimme hänen kanssaan projektin käytännön asioista, rooleista sekä kehittämästämme ideasta. Jussi Loukiainen oli toimeksiantajan roolissa projektin tuottaja, Joni Hänninen vastasi pelin kokonaisvaltaisesta ohjelmoinnista ja Unity-alustan opettelemisesta sekä käyttämisestä ja Janne Harju digitaalisen sisällön tuottamisesta peliin, joka sisälsi esimerkiksi 3D-mallien tuottamisen, teksturoinnin ja materiaalien luomisen, äänimaailman sekä panoraamakuvien muokkaamisen. Pelin suunnittelusta vastasimme molemmat, sillä projekti sekä tiimi olivat verrattain pieniä. Suunnitteluun saataisiin sitä myöten sekä teknistä että taiteellista näkökulmaa, sillä sekä ohjelmointiosaaminen ja materiaalin tuottaminen yhdistyisivät siinä. Roolitus oli helppo päättää molempien tiimin jäsenten kohdalla eri suuntautumisten sekä osaamisalueiden vuoksi.

Rajallisesta aikataulusta sekä laitteiston hankkimisen vuoksi päädyimme yhdessä ratkaisuun, että pelistä tulisi eräänlainen tekniikka- ja laitteistodemo, jolla voitaisiin esitellä uutta VR-laitteistoa, Unity-alustaa sekä markkinoida uutta Vaasaan perustettavaa VR-hautomoa. Tarkoitus oli myös ensimmäisenä olla testaamassa uutta laitteistoa ja esittää kehitysideoita tulevaa toimintaa varten. Tekniikkademon luominen myös vähensi tarvittavaa käsikirjoituksen ja synopsisen käsittelyä, sekä digitaalisen materiaalin luomiseen käytettävää aikaa, sillä varsinaista ääninäyttelyä ei tarvittu.

4.2 Laitteiston hankinta ja kokoonpano

Kuten jo toisessa kappaleessa mainittiin, tietokoneet kasattiin ensisijaisesti VR-pelien kehitykseen sekä pelaamiseen. Mitään täysin yksiselitteistä mittaria on vaikea

VR-kokoonpanolle määrittää, mutta nykyään jo useat valmistajat tarjoavat komponentteja ”VR-ready” -merkinnällä, joka tarkoittaa sitä, että kyseisten osien suorituskyky on riittävä VR-käyttöön. Myös Steamin kautta on mahdollista ladata SteamVR Performance Test, jonka suorittamalla ohjelma kertoo testin päätyttyä suuntaa antavan arvion siitä, onko tietokoneen suorituskyky riittävä VR-käyttöön vai ei. (Tom’s Hardware, 2016)

Ennen komponenttien valintaa oli siis hieman työtä selvittää sekä varmistella, että kokoonpano olisi varmasti yhteensopiva VR-tekniikan kanssa. Tässä luotettiin aika pitkälti omaan osaamiseen, sillä koneiden kokoaminen sekä päivitys ovat olleet osa harrastusta jo yli kymmenen vuotta. Toki internet on nykyään tutoriaaleja pullollaan ja monet sivustot tarjoavat myös kasauspalvelua, mikäli itse ei halua tietokonetta kasata. Liikkeelle lähdettiin tiedustelemalla Jussilta koneiden lukumäärää, myöhempiä käyttötarkoitusta sekä niille varattua budjettia. Sen perusteella alettiin määrittelemään sopivaa kokoonpanoa. Lähdimme aluksi liikkeelle siitä, että yhden keskusyksikön hinnaksi tulisi ~800€, ja keskusyksiköitä tulisi kaksi kappaletta. Huomasimme kuitenkin pian, että kovin pitkäikäistä ratkaisua ei tuohon hintaan olisi saanut Suomen verkkokauppojen kautta rakennettua, jolloin päätimme Jussin kanssa, että budjetti tulisi nostaa noin tuhanteen euroon per keskusyksikkö.

Osat tilattiin Jimm’s:ltä sekä Multitronicista, ja ne tulivatkin perille parissa arkipäivässä. Tässä vaiheessa Oculus Riftin olisi ollut jo aika saapua, ja pitkäksi venähtäneen odottelun jälkeen saimmekin tiedon, että myös HTC Vive oli tulossa. Silloin ajatuksemme pelissämme käytettävästä laitteistosta alkoi elää, sillä emme halunneet viivästyttää aloitusta enempää. Vive osoittautuikin tekniikaltaan hieman Oculusista kehittyneemmäksi laitteistoksi, joka toisaalta taas asetti toki pieniä rajoitteita tilan koon puolesta. Tarkoitukseen sopiva tila kuitenkin löytyi, jossa pääsimme projektin aloittamaan.

Varsinainen tietokoneiden kokoaminen vei aikaa hieman alle kaksi tuntia konetta kohden, ja kokoonpanon suunnitteluun meni muutamia tunteja. Tietokoneet ovat

käytettävissämme opinnäytetyön ajan, jonka jälkeen ne olisi tarkoitus ottaa käyttöön West Coast Startupin uudella VR-pajalla Ylen tiloissa. Valitettavasti emme vielä opinnäytetyön aikana päässeet sinne siirtymään.

Suunniteltu käyttöikä tietokoneille on alustavasti noin kolme vuotta. Tämän jälkeen niitä joudutaan todennäköisesti ylikellottamaan tai komponentteja päivittämään uudempiin, jotta koneista saadaan enemmän tehoa irti. Toki tämä seikka on otettu huomioon kokoonpanoa suunniteltaessa ja yhdeksi vaihtoehdoksi jää laittaa molempien tietokoneiden näyttöohjaimet rinnakkain SLI-liitännällä yhteen koneeseen, jolloin saadaan tarvittavaa lisätehoa ja joudutaan ostamaan uusi näyttöohjain vain toiseen keskusyksikköön.

Ainoat ongelmat kasauksen yhteydessä johtuivat prosessorijäähyn suojatarran liimasta, joka sotki yhden pinnan, ja keskusmuisteista, jotka menivät aluksi vahingossa väärin väyliin. Ongelma hoitui varsin nopeasti jäähyn putsauksella ja muistien paikkaa vaihtamalla, jolloin punainen merkkivalo (**kuvio 5.**) sammui ja kone toimi moitteettomasti. Kuvioissa 2-5 kuvia toisen keskusyksikön kokoamisesta.



Kuvio 2. Komponentit paketeissa



Kuvio 3. Normaalista poikkeava kiintolevyjen kiinnitys



Kuvio 4. Ongelma suojatarran liiman kanssa



Kuvio 5. Kone käynnissä

4.3 Työtilat, laitteiston käyttöönotto ja työtapa

Alkuperäinen suunnitelma oli tehdä peli Oculus Rift -VR-laitteistolle, ja sen oli tarkoitus toimia joko näppäimistöllä tai Xbox 360:n ohjaimella. Laitteiston toimitusongelmista johtuen päädyimme kuitenkin tekemään pelin käyttäen HTC Vive VR-laitteistoa, ja vaihdoimme ohjainsuunnitelman Viven omiin navigaatiosauvoihin. Jo tässä vaiheessa pelimekaniikkaa pystyttiin kehittämään pidemmälle, sillä sauvat mahdollistivat pelimekaniikat, joita emme aikaisemmin olisi tavallisilla ohjaimilla kuten näppäimistöllä ja hiirellä, pystyneet toteuttamaan.



Kuvio 6. HTC Vive lasit ja ohjaimet

Laitteiston paikaksi päätettiin alustavasti Muovan rakennuksessa sijaitseva puuverstas, josta kävimme siirtämässä ylimääräiset tuolit, pöydät ja laitteet aulatilaan. Tietokoneet sijoitettiin erillisille pöydille, joista toinen toimi alustana HTC Vivelle ja toinen Oculus Riftille. Verstaan keskelle jätettiin mahdollisimman paljon tilaa, jotta molempia laitteita voitaisiin hyödyntää tehokkaasti sekä pelaamiseen että ohjelmoitavien pelien testaamiseen. Oculusta on tekniikan puolesta mahdollista käyttää istuessa tai seistessä paikallaan, mutta Vive vaatii huoneasetuksilla pelatessa toimiakseen kaksi majakkaa sekä noin 2,0m*1,5m suuren pelialueen. Viven majakat asennettiin alustavasti verstaalla sijaitseviin kattorimoihin, ja myöhemmin majakoille hankittiin omat kolmijalat sekä käännettävät alustat liikuteltavuuden helpottamiseksi.

Molemmille koneille asennettiin Office 365-ohjelmisto, Unity3D-ohjelmisto, Visual Studio Community 2015-kehitysalusta sekä Steam-sovellus, jolla voidaan asentaa ja pelata pelejä. Steam-sovelluksen avulla voitiin asentaa myös SteamVR-sovellus, jotta HTC Viveä voidaan hyödyntää peleissä sekä Unityssä.

Päätimme myös heti alussa lähteä kehittämään peliä vesiputousmallin mukaisesti, sillä se on toiminut hyvin aiempien peliprojektien parissa tiimissämme. Se toimii

myös erittäin hyvin pienemmissä projekteissa, sillä työmäärä ei merkittävästi lisääntynyt, vaikka palataan vaiheissa taaksepäin kehittämään ja hiomaan aiempaa ideaa. Myös alustan tuntemattomuus ja mahdollisuuksien etsiminen tukivat tällaista projektimallia. Suuremmissa ammattimaisissa projekteissa tämä ei kuitenkaan mielestämme olisi kovin hyvä malli toimia, sillä takaisinpalaaminen mallin asteissa voi tuottaa merkittävän määrän lisätyötä ja sitä myötä lisätä kustannuksia.

4.4 Peli-idean kehittäminen, käsikirjoitus, perusmekaniikka

Alkuperäinen idea on lähtöisin molempien kiinnostuksesta kauhuelokuviin ja peleihin, ja aiemmasta kokemuksesta kauhupelin kehityksestä. Tämä myös siirtää pelin painotusta enemmän audiovisuaalisuuteen ja etenkin äänimaailmaan, joka tuki pelinkehitystä, koska ohjelmointitekniikkaan, ohjelmakoodiin ja muuhun tekniseen puoleen ei voitaisi panostaa niin paljon alustan ja laitteiston uutuuden sekä haasteellisuuden vuoksi. Myös toimitusongelmat lyhensivät jo valmiiksi rajallista aikatauluamme merkittävästi.

Alkujaan pelimme idea oli hyvin juonellinen, ja painotus oli hyvinkin pitkälle ääninäyttelyssä sekä tarinankerronnassa. Olosuhteiden vuoksi kuitenkin päädyimme karsimaan juonellisuudesta, ja tekemään kauhupelin hyvinkin rajallisella juonella, mutta samalla säilyttäen kauhuelementin peliä kehitettäessä. Myös pelaajan omalle mielikuvitukselle pyrittiä antamaan tilaa.

Hetken ideaa palloiteltuamme päädyimme luomaan peliin skenaarion, jossa pelaaja on lähtenyt mökkeilemään syrjäiseen kalastajantorppaan. Yön saapuessa mökissä kuitenkin alkaa tapahtua kummia, ja outoja hahmoja alkaa ilmestyä pelaajan mökin ympärille. Pelaajan on tavalla tai toisella selviydyttävä tiettyyn pisteeseen saakka, jolloin peli päättyy.

Perusmekaniikkaan syntyi muutoksia laitteiston vaihduttua HTC Vive -laitteistoon, johon kuuluu kaksi navigaationsauvaa. Näitä hyödyntääksemme päädyimme luomaan peliin mekaniikan, jossa pelaajan toinen sauva on salamavallo, jolla ympäröivät otukset on mahdollista tuhota. Salamavalossa on myös akku, joka latautuu jo-

kaisen käyttökerran jälkeen. Näin peliin luodaan pienmuotoinen strateginen elementti, jolla saadaan toiminnollisuuteen hieman lisää syvyyttä. Tämä tavallaan pakottaa pelaajan käyttämään omaa harkintaansa, sillä salaman käyttöikkuna pienee. Kyseisellä mekaniikalla saadaan toki luotua myös pelaajalle pientä painetta, joka taas korostaa pelin kauhuaspektia.

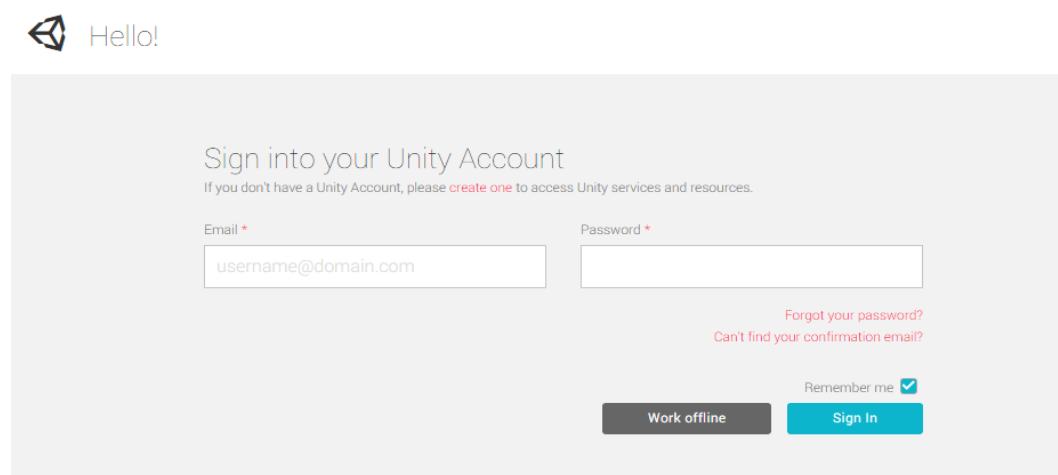
Päätimme tietoisesti jättää pelistä pois ns. HUD-näytön (heads-up display, heijastusnäyttö) joka kertoo pelaajalle tarkalleen, kauanko salaman latautuminen kestää. Näin peli saadaan näyttämään luonnollisemmalta, ja kokemus on immersiivisempi.

4.5 Alustan käyttö ja alkuvaiheen prototyypin luominen

Tässä luvussa käsittelemme Unityn eri ominaisuuksia ja tavallisia toimintoja, jonka ohella sivutaan oman prototyypin rakennusta.

4.5.1 Uuden projektin luominen

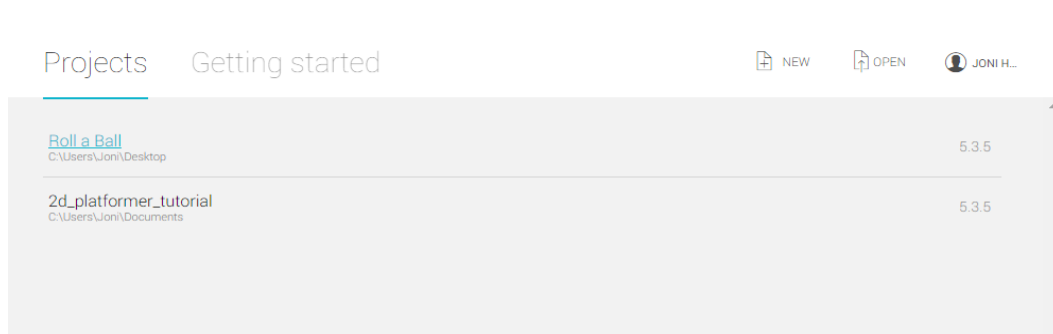
Uusi projekti aloitetaan, kun Unity käynnistetään. Tässä vaiheessa kirjaututaan sisään omalla Unity-käyttäjätunnuksella, tai luodaan uusi, mikäli sellaista ei ole.



Kuvio 7. Unityn kirjautumisikkuna

”Create one” -linkistä voidaan luoda uusi Unity käyttäjätunnus, jossa kysytään esimerkiksi käyttötarkoitusta. Oma käyttäjätunnuksemme on yksityiskäyttöön, mikä sopii käyttötarkoitukseemme, sillä se on luotu vain henkilökohtaisia projekteja varten eikä sen luonne ole voittoa tavoitteleva.

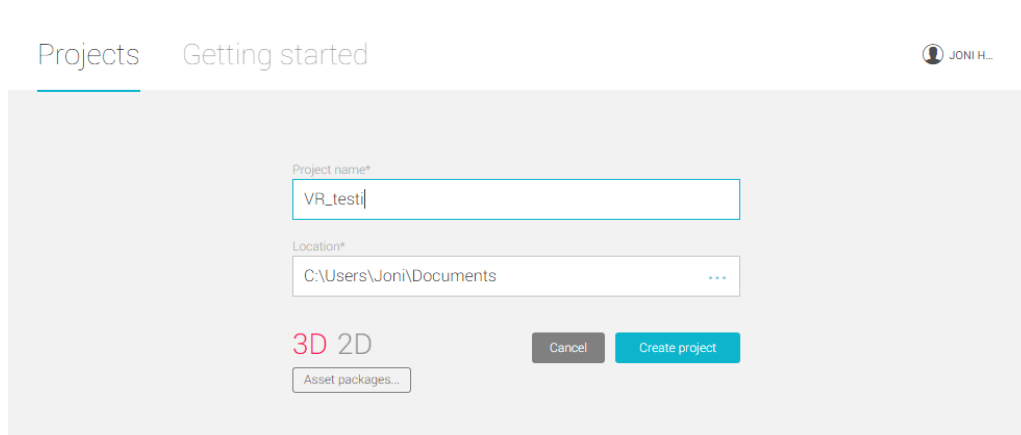
Tämän jälkeen tunnuksille kirjaudutaan, jonka jälkeen avautuu projekti-ikkuna, josta voidaan valita jokin aiempi projekti tai luoda kokonaan uusi.



Kuvio 8. Projektin aloittaminen

Uusi projekti voidaan aloittaa oikean yläkulman ”NEW” -painikkeella. Vanhat projektit löytyvät valmiina listasta, kuten myös tiedostopolut vanhoihin projekteihin. Myös Unity-version numero näkyy samalla sarakkeella.

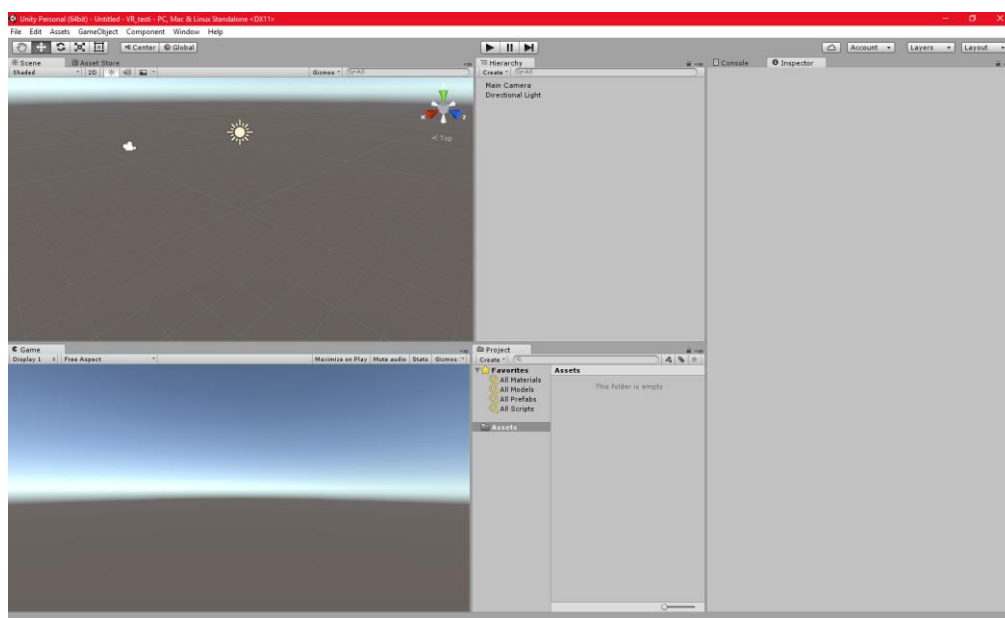
Uutta projektia aloitettaessa tulee ottaa selville ainakin se, tuleeko peli olemaan kaksi vai kolmiulotteinen. Projektia aloitettaessa annetaan projektille nimi ja tiedostopolku, jonne projektiin liitettävät resurssit ja projektitiedostot tallennetaan.



Kuvio 9. Uusi projekti

Projektiin voidaan myös liittää erilaisia resurssipaketteja projekteja luotaessa ”asset packages” -painikkeella, esimerkiksi hahmoja, kameroita tai efektejä. Tämä voidaan tehdä myös jo luotuun projektiin. Omaan projektiimme siirsimme resurssit vasta projektin luotuaamme. Valitsimme kolmiulotteisen projektin, jotta pelistä voitaisiin hyödyntää VR-tekniikkaa täysin, ja jotta voisimme käyttää ottamiamme pa-

noraamakuvia hyödyksi mahdollisimman tehokkaasti. Projektin luonti viimeistellään ”Create project” -painikkeella, jonka jälkeen Unity rakentaa käyttäjälle uuden projektin, luo perusnäkyvän ja tyhjän pelialueen. Unity myöskin luo valmiiksi muutaman peruspeliobjektin (GameObject), joita peli tarvitsee toimiakseen. Nämä voi korvata myöhemmin projektin edetessä. Nämä perusobjektit ovat Directional Light, joka valaisee pelialueen, sekä Main Camera, joka on näkymä, jonka pelaaja näkee pelatessaan peliä. Tässä vaiheessa Unity myöskin tarkastaa, mikäli uusia päivityksiä alustalle on saatavilla, ja pyytää käyttäjää päivittämään alustan, mikäli näin on.



Kuvio 10. Unityn perusnäkyvä

Unityn perusnäkyvä on täysin muokattavissa, tarpeettomia ikkunoita voidaan piilottaa, siirrellä, venyttää ja kutistaa käyttäjän mielen mukaan. Tärkeimpiä ikkunoita ovat kuvassa 10 näkyvissä olevat Scene (näyttämö), Game (peli), Hierarchy (hierarkia), Project (projekti), sekä Inspector (esikatselu) ja Console (konsoli) -ikkuna.

Scene-ikkunassa voidaan tarkastella pelialuetta, sen objekteja ja muokata niitä. Scene-ikkunassa valitut objektit korostetaan Hierarchy-ikkunassa, ja objektin ominaisuudet ilmestyvät näkyviin Inspector-ikkunassa. Sama koskee myös Hierarchy-ikkunassa tehtyjä valintoja; ne korostuvat Scene-ikkunassa ja tiedot tulevat näkyviin Inspector-ikkunaan. Project-ikkunassa voidaan tarkastella projektissa olevia

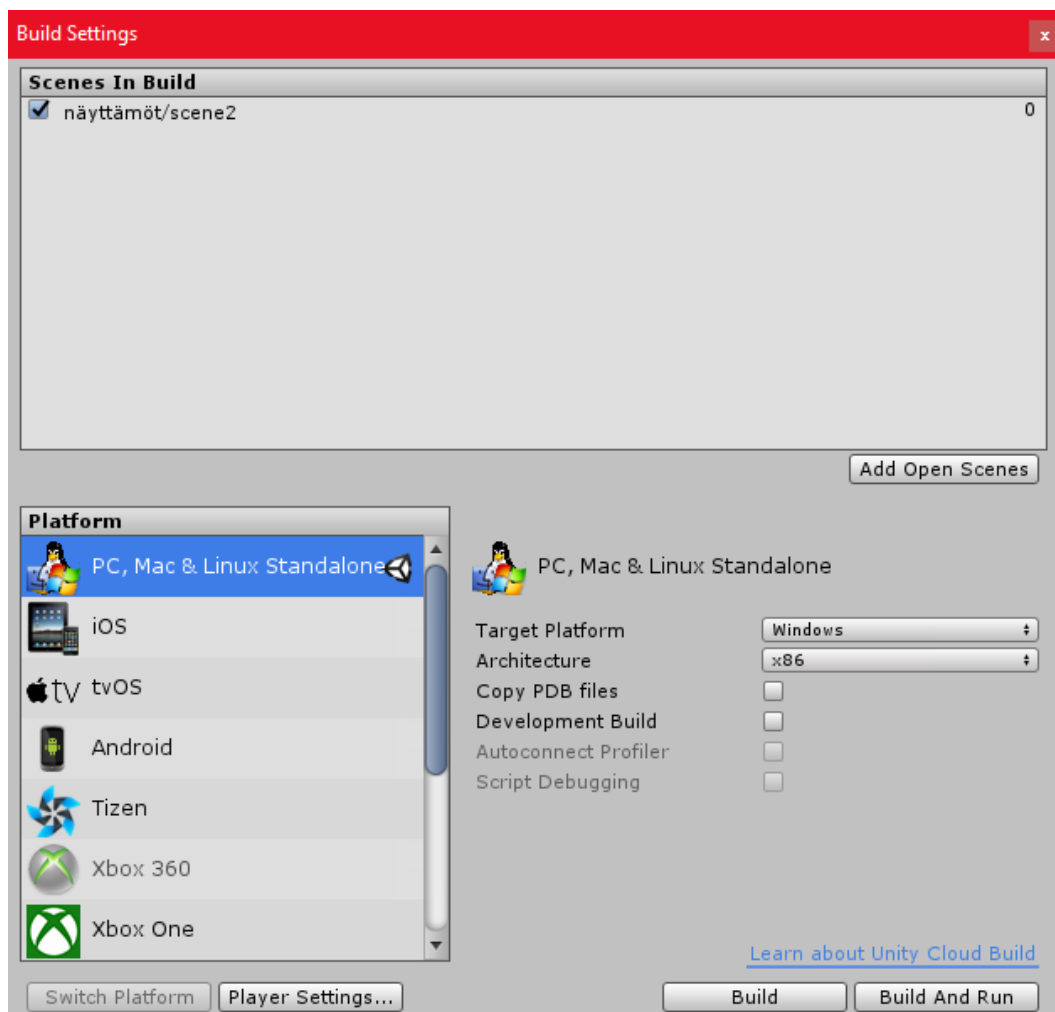
resursseja, vaikka niitä ei olisi lisätty peliin. Inspector-ikkuna ottaa esiin myös projekti-ikkunassa näkyvien objektien ja resurssien tiedot. Game-ikkuna toimii testausvaiheessa pelinäyttönä. Game-ikkunan perspektiivi riippuu kameran sijainnista ja siihen liitetystä skriptoista (ohjelmakoodista), jotka ohjaavat peliobjektin toimintaa. Console-ikkuna antaa tietoa debugausvaiheessa esimerkiksi virheistä ja varoituksista. Myös skriptoja voidaan pyytää antamaan tietoa Console-ikkunaan kirjoittamalla koodiin `Debug.Log("haluamasi teksti");`. Näin Console-ikkuna toimii myös hyvänä välineenä testata skriptojen toimivuutta. Käytimme konsoliviestejä testatessamme skriptojen toimivuutta.

Tallennettaessa uutta projektia tallennetaan myöskin uusi "scene" eli näyttämö. Näyttämöä voi ajatella esimerkiksi pelin yhtenä tasona. Se on pohja, jolle peliobjektit sijoitellaan. Näitä näyttämöitä voi olla pelin sisällä useita, ja niiden välillä voidaan navigoida esimerkiksi valikon avulla, tai yhden tason läpäistyään voidaan siirtää pelaaja toiseen näyttämöön, riippuen siitä, millainen peli on kyseessä. Pelin lopullista versiota eli "buildia" rakennettaessa täytyy halutut näyttämöt lisätä build settings -kohdassa rakennettavaan versioon. Unity määrittää näyttämöiden sijainniksi automaattisesti projektin Assets-kansion, jonne tuodaan myöskin kaikki toisaalla luodut resurssit. Hyvän ja selkeän projektin säilyttämiseksi suosittelemme hyvän kansiorakenteen luomista projektin Assets-kansioon, jolloin objektit löytyvät helpommin. Kansiorakennetta voi muokata suoraan Unityn projekti-ikkunassa. Unity luo uusiin näyttämöihin automaattisesti samat peliobjektit kuin uudessa projektissakin.



Kuvio 11. Unityn projekti-ikkuna

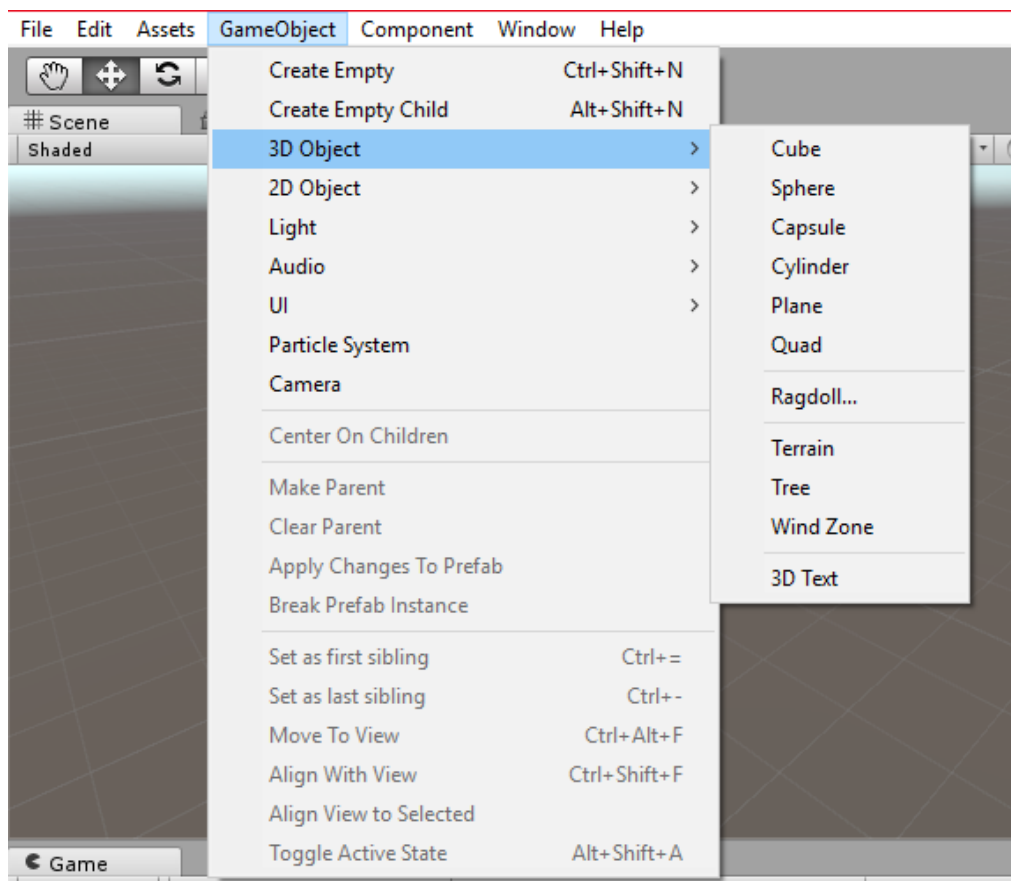
Omassa projektissamme käytimme näyttämöitä testatessamme erikseen panoraamakuvien toimivuutta, pelimekaniikkoja, valikkoa sekä törmäyksiä. Teimme myös lopulliselle pelille oman näyttämön, johon sijoitimme viimeistellyt resurssit ja omasta mielestämme toimivat mekaniikat. Oman näyttämönsä saivat myös menu ja välivideot. Näyttämöt ovat myös hyvä tapa järjestellä tietoa projektin sisällä, sillä kaikkia projektissa olevia näyttämöitä ei tarvitse sijoittaa lopulliseen peliin, sillä valittaessa build settings -asetuksia voidaan valita, mitkä näyttämöt asetetaan mihinkin buildiin.



Kuvio 12. Rakennusasetukset

4.5.2 Tavallisten peliobjektien luonti

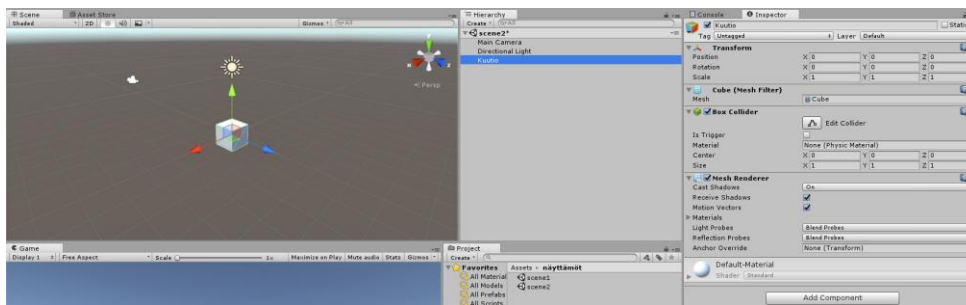
Prototyypin luomisessa koimme hyväksi testata omia pelimekaniikkojamme Unityn sisältä löytyvillä peliobjekteilla. Peruspeliobjekteja luodaan valitsemalla ylhäällä näkyvästä valikosta `GameObject >` valitsemasi objekti. Unityllä voidaan myöskin luoda esimerkiksi tyhjiä peliobjekteja, tehosteita, User Interface -objekteja ja materiaaleja.



Kuvio 13. Peliobjektien luonti

Käytimme perusprimitiivejä testatessamme oman pelimme toimivuutta, mekaniikkoja, törmäyksiä ja skriptoja. Vasta prototyypissä olevien mekaniikkojen testaamisen jälkeen aloimme luoda omia resurssejamme. Kun prototyypin testauksen perusteella tiesimme mikä toimii, palaset oli helppo yhdistää lopulliseen tuotteeseen säilyttäen toiminnollisuuden.

Peliobjekti sisältää myös erilaisia komponentteja. Komponentteja on useita erilaisia. Komponenteilla määritellään esimerkiksi objektissa käytettävä materiaali, 3D-mallin verkkopinta ja törmäyspisteet. Mahdollisia komponentteja ja niiden yhdistelmiä on todella paljon, ja ne toimivat hyvin yhteen. Komponentit yhdessä vaikuttavat peliobjektin käyttäytymiseen pelitilassa. Komponentteja voidaan myös muokata skriptojen avulla. Myös niinkin yksinkertaiset asiat kuin esimerkiksi sijainti, kulma ja koko määritellään peliobjektille Transform-komponentin avulla. Kuutio voidaan saada esimerkiksi liikkumaan pelialueen laidasta laitaan muokkaamalla objektin Transform-komponenttia skriptan avulla.



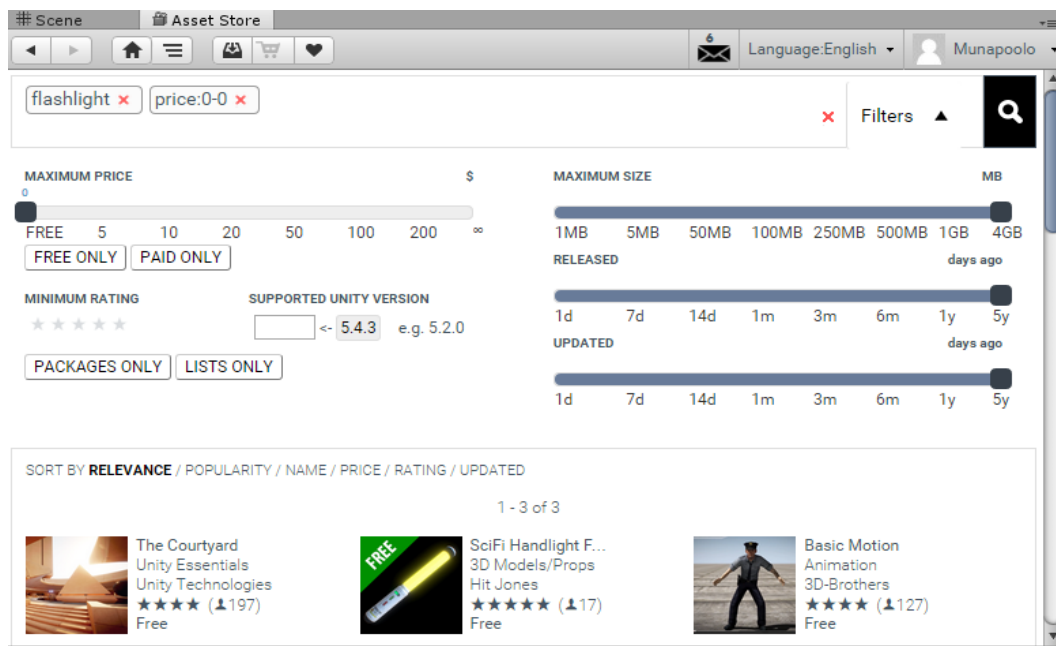
Kuvio 14. Kuutio ja sen komponentit

Aktivoidessa peliohjelmaa hierarkia-ikkunasta tai näyttämö-ikkunasta esikatseluikkunaan ilmestyvät kyseiseen peliohjelmaan liittyvät komponentit. Komponentteja ei voida missään projektissa unohtaa, sillä kaikki pelimaailman muutokset tapahtuvat komponenttien avulla. Myös skripti on yksi peliohjelmaan liittyvistä komponenteista. Komponentteja voi oman tarpeen mukaan lisätä tai poistaa esikatseluikkunasta hyödyntämällä.

Omassa projektissamme mallinsimme vihollisen prototyypissä kuutiona, johon lisäsimme tarpeen mukaan komponentteja. Loimme myös prototyyppiimme alkeellisen huoneen käyttäen plane-peliohjelmaa lattiana sekä cube-objekteja seininä ja katonna. Lisäsimme seinille, katolle ja lattialle omat materiaalikomponentit ja loimme Viven navigaatioalustalle oman taskulampunmuotoisen mallin. Lisäsimme taskulamppuun myös valokeilan ja collider-komponentin, joka sijoitettiin valokeilan sisälle. Collider-komponenttien avulla voidaan tarkastella törmäyksiä objektien välillä. Lisäsimme myös kuutiolle ja taskulampulle oman tag-arvon, jolla pystytään tunnistamaan Collider-komponenttien törmätessä toisen peliohjelmaan liittyvän tag-arvo. Näiden avulla pystyimme luomaan tilanteen, jossa vihollisen osuessa valokeilaan vihollisohjelma katoaa.

4.5.3 Resurssien tuominen Unityn resurssikaupasta

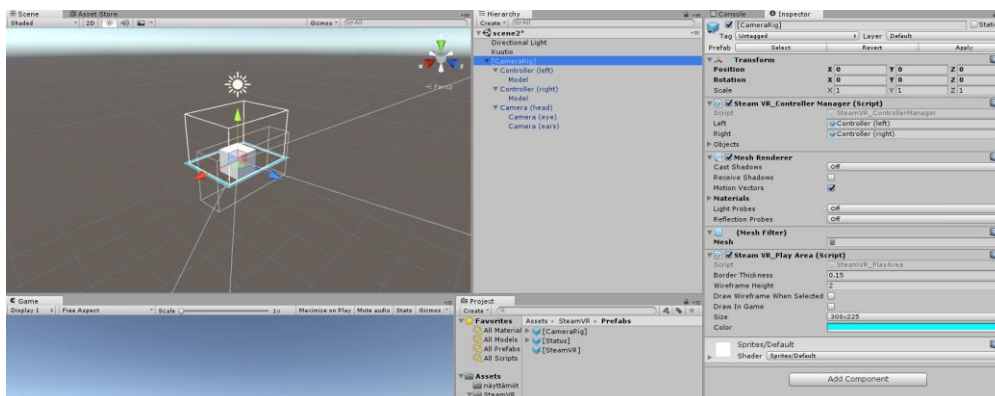
Kaikkia pelin resursseja ei tarvitse itse mallintaa tai kirjoittaa. Unityllä on oma Asset Store -kauppa, josta on mahdollista ostaa tai ladata ilmaiseksi tietynlaisia resursseja. Resursseja on mahdollista ladata laidasta laitaan, esimerkiksi 3D-malleja ja valmiiksi rakennettuja materiaaleja on runsas määrä. Useimmat niistä ovat kuitenkin maksullisia.



Kuvio 15. Unity asset store

Resurssikauppaan pääsee suoraan Unityn sisäisestä käyttöliittymästä. Etsiä voi esimerkiksi tiedostokoon mukaan, hinnan tai pelkkien hakusanojen perusteella. Resurssikauppaan täytyy kirjautua, ja kirjautumisen jälkeen maksaa ostokset tai ladata suoraan. Resurssit ladataan automaattisesti ja tuodaan suoraan avoimna olevaan projektiin, resurssikansioon omaan kansioonsa.

Käytimme resurssikauppaa esimerkiksi 3D-mallien testaamisessa. Resurssikaupasta saimme myöskin ilmaiseksi ladattua SteamVR plugin -resurssipaketin suoraan Unityyn, joka on välttämätön, mikäli aiotaan kehittää peli, joka hyödyntää HTC Vive-laitteistoa.

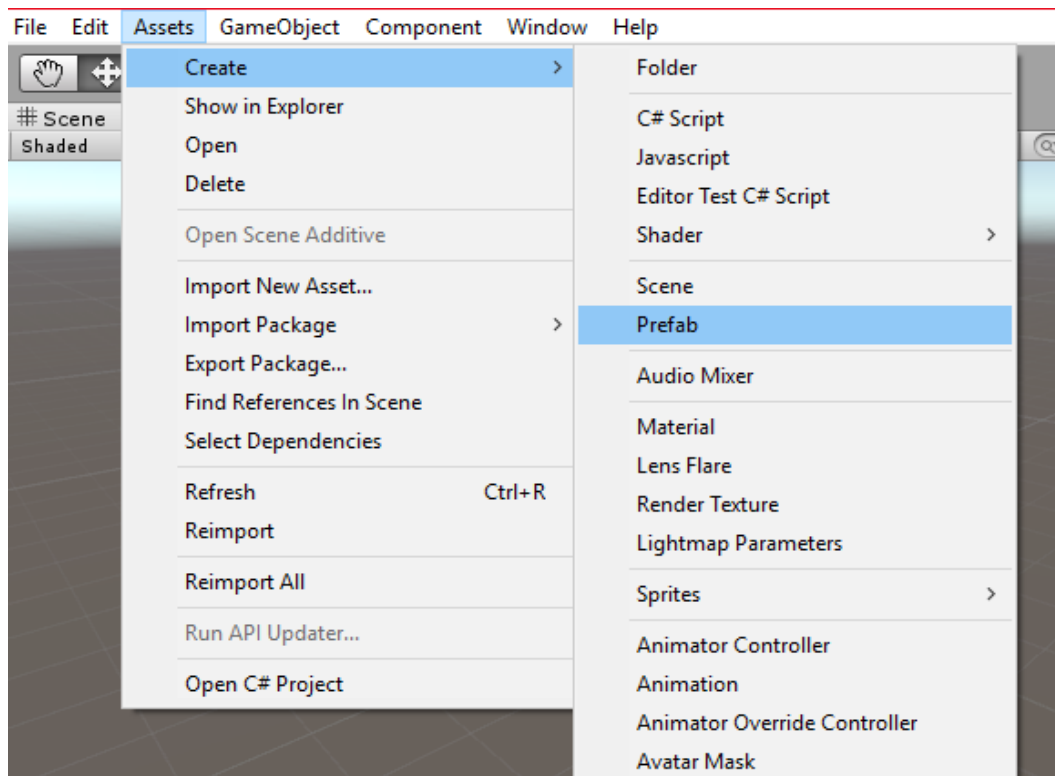


Kuvio 16. SteamVR resurssit

Steam VR plugin tuo kaikki tarvittavat skriptat ja valmiit objektit, jotta kaikki Vive-laitteiston osat toimivat. Tämän paketin ansiosta voidaan helposti sijoittaa näyttämölle valmis peliobjekti CameraRig, jonka jälkeen voidaan hyödyntää sekä navigaation osia että päähän asetettavaa pelinäyttöä. Paketissa valmiina olevat skriptat ohjaavat peliä siten, että peli voi hyödyntää Steamia kautta asennettavaa SteamVR-sovellusta, joka taas mahdollistaa sekä navigaation osien ja pelinäytön seuraamista aktiivisesti. Näin esimerkiksi ohjaimiin sijoitettavat 3D-mallit seuraavat ohjainten sijaintia ja asentoa reaaliaikaisesti pelimaailmassa. Omassa projektissamme sijoitimme taskulampun 3D-mallin toiseen ohjaimen, jonka avulla se liikkuu ohjaimen mukana pelimaailmassa. Näin pystytään luomaan pelaajalle tunne, että tämän kädessä on oikeasti taskulamppu eikä peliohjain.

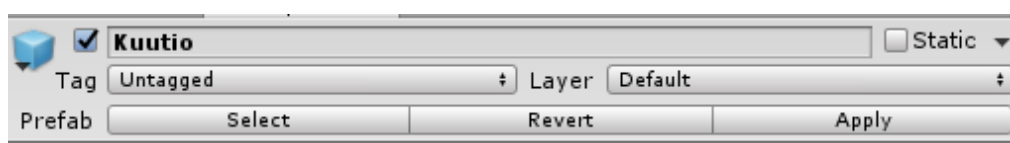
4.5.4 Prefabien luominen muokatusta peliobjektista

Prefab on eräänlainen pohjapiirustus, joka voidaan luoda esimerkiksi itsemuokatusta peliobjektista. Peliobjektiin voidaan liittää komponentteja ja sitä voidaan muokata, jonka jälkeen siitä voidaan tehdä niin sanottu Prefab-objekti. Prefabeja käytetään esimerkiksi vihollisten luomiseen, jotta kaikkia peliobjektin komponentteja ei tarvitse lisätä jokaiselle yksilölle erikseen.



Kuvio 17. Prefabin luonti

Prefab luodaan oikean yläkulman valikosta assets > create > prefab. Tämä luo tyhjän prefab pohjan, johon muokattu peliobjekti raahataan. Tämän jälkeen prefab on valmis. Myös tulevat muutokset prefabiin voidaan tehdä peliobjektin kautta, sillä muutokset voidaan lisätä peliobjektin esikatselu-ikkunasta suoraan klikkaamalla prefab-kohdan apply-painiketta.



Kuvio 18. Prefab-asetusten muuttaminen

Omassa pelissämme hyödynsimme prefabeja vihollisten luomisessa, sekä steamVR pluginin CameraRig-objektin muokkaamisessa, jotta 3D-mallimme ja skriptamme pysyisivät niille kuuluvilla paikoillaan. Käytimme prefabeja myös vihollisten uudelleensyntymispisteiden luomiseen. Näin oli mahdollista saada identtiset uudelleensyntymispisteet vihollishirviöille ja myöskin vihollishirviöt olivat identtisiä, eikä jokaista tarvinnut erikseen muokata skriptojen kautta.

4.5.5 Skriptojen luominen ja niiden käyttäminen

Skriptat ovat ohjelmakoodin palasia, jotka liitetään peliobjektien komponenteiksi. Skriptoja käytetään pääasiallisesti pelinkehityksessä pelin mekaniikkojen ja toiminnallisuuden luontiin. Niillä voidaan viitata peliobjekteihin, ja niiden komponentteihin, joilla voidaan esimerkiksi liikuttaa pelaajan hahmoa pelialueella sekä luoda ja tuhota peliobjekteja. Skriptoja käytetään siten, että niillä muokataan esimerkiksi tietyn peliobjektin komponenttia, joka saa aikaan peliobjektissa muutoksia. Skriptat luodaan lisäämällä peliobjektiin add component -painikkeella uusi komponentti. Listasta valitaan new script, jonka jälkeen skripta nimetään. Skriptoissa on valmiina Start- ja Update-funktiot. Start funktio lähtee pyörimään välittömästi pelin käynnistytksen yhteydessä, mikäli peliobjekti on olemassa. Update-funktio käydään läpi jokaisella pelin framella. Käytimme myös esimerkiksi Unityn sisäänrakennettua Awake-funktiota, joka käynnistetään silloin automaattisesti, kun uusi peliobjekti jossa kyseinen skripta on komponenttina, luodaan pelialueelle.

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class TakkaKaytos : MonoBehaviour {
5      public Light takka;
6      // Use this for initialization
7      void Start () {
8
9      }
10
11     // Update is called once per frame
12     void Update () {
13         takka.intensity = Random.Range(1f, 1.4f);
14     }
15 }
16
17

```

Kuvio 19. Esimerkkiskripta pelialueen takkavaloon

Yllä näkyvä esimerkkiskripta on pelialueemme takan käytöstä ohjaava komento. Sen tarkoitus on luoda takan valon käytöksestä elävä ja realistisen avotulen kaltainen. Skriptan idea on laskea valon intensiteetti jokaisella pelin ruudunpäivitysvälillä uudestaan, jotta näyttäisi, että takasta tuleva valo lepattaisi. Tähän tarkoitukseen loimme julkisen tietotyypin valo, ja nimesimme sen takaksi. Koska tietotyyppi

on julkinen, se ilmestyy skriptaan valmiiksi lokeroksi Unityn omassa editor-ikkunassa. Tähän lokeroon vedetään hierarchy-ikkunasta kyseinen takkavalo, jonka jälkeen pelimoottori tietää, mihin objektiin viitataan. Kaikilla valoilla, joita pelimaailmaan luodaan, on valmiina oma komponenttinsa intensiteetille. Skripta ymmärtää viitata tähän komponenttiin komennolla ”takka.intensity”, sillä tietotyyppi on valo. Tämän jälkeen määritetään satunnainen luku, joka määrää valon intensiteetin. Tämä tehdään komennolla ”Random.Range(1f, 1.4f);”, jonka avulla voidaan valita satunnainen float-tietotyypin arvo väliltä 1 ja 1,4.

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class spawnScript : MonoBehaviour {
5
6      public GameObject monsteri;
7      bool isSpawning = false;
8      public float minTime = 5.0f;
9      public float maxTime = 15.0f;
10     public GameObject[] enemies; // Array of enemy prefabs.
11
12     IEnumerator SpawnObject(int index, float seconds)
13     {
14         Debug.Log("Waiting for " + seconds + " seconds");
15
16         yield return new WaitForSeconds(seconds);
17         Instantiate(monsteri, transform.position, transform.rotation);
18
19         //We've spawned, so now we could start another spawn
20         isSpawning = false;
21     }
22
23     void Update()
24     {
25         //We only want to spawn one at a time, so make sure we're not already making that call
26         if (!isSpawning)
27         {
28             isSpawning = true; //Yep, we're going to spawn
29             int enemyIndex = Random.Range(0, enemies.Length);
30             StartCoroutine(SpawnObject(enemyIndex, Random.Range(minTime, maxTime)));
31         }
32     }
33 }

```

Kuvio 20. Peliobjektien luominen skriptan avulla

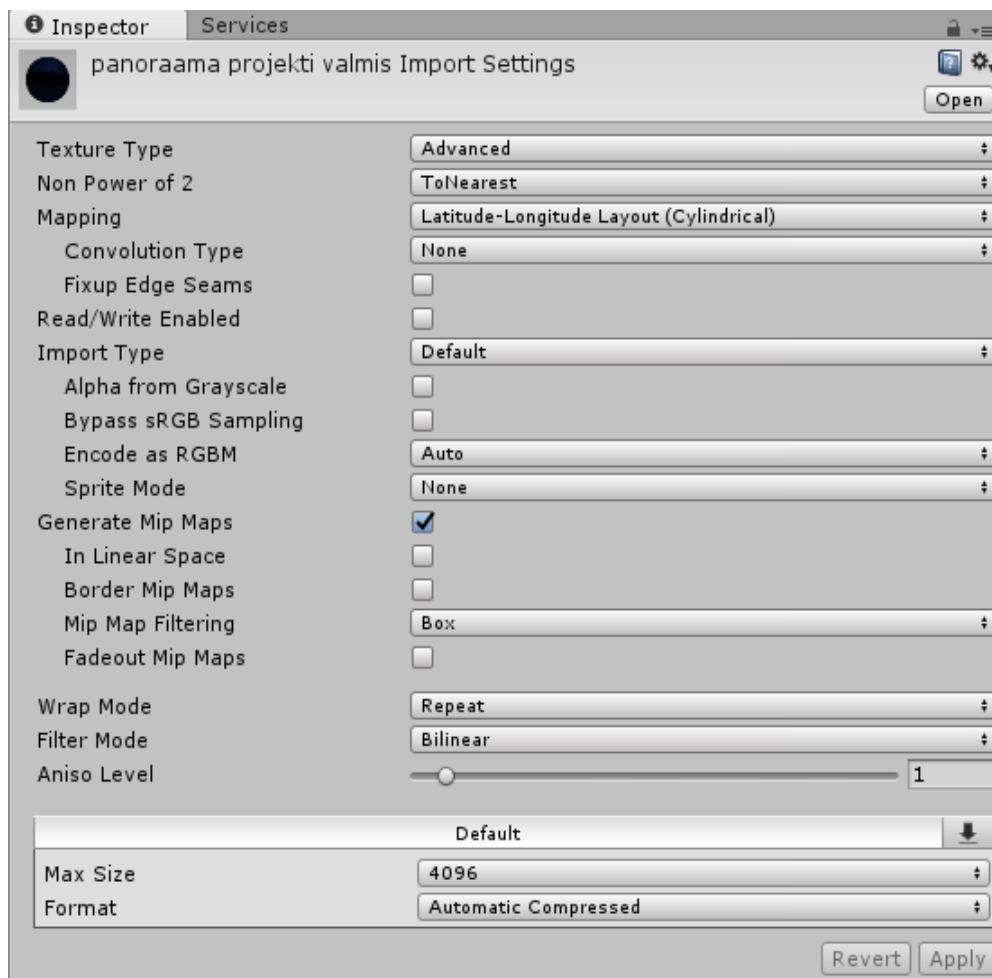
Ylläolevassa esimerkissä on hieman monimutkaisempi skripta, jolla luodaan peliobjekteja prefabien perusteella. Loimme pelialueelle tyhjiä peliobjekteja, joihin liitimme vain kyseisen skriptan. Näistä tyhjistä peliobjekteista tuli skriptan avulla uudelleensyntymispisteitä pelin vihollisille, jotka perivät tyhjän peliobjektin transform-komponentin arvot. Näin vihollisobjektit syntyvät tyhjiä peliobjektien kanssa samaan pisteeseen koordinaatistossa. Vihollisten luomiseen käytimme IEnumerator-metodia, jonka avulla voidaan mahdollistaa erilaiset yield-käskyt. Yield-käskyillä voidaan esimerkiksi laittaa funktio odottamaan tietyn aikaa ennen funktion jatkamista. Update-funktio tarkastaa jokaisella ruudunpäivityskerralla, ollaanko sillä hetkellä luomassa uutta vihollista. Mikäli ei, se lähettää pyynnön

SpawnObject-funktiolle, joka luo satunnaisen vihollisen arrayn avulla, 5-15 sekunnin aikavälillä. Pelissämme käytetään vain yhtä vihollistyyppiä, joten jätimme array-taulukon käyttämättä.

Käytimme skriptoja lähes kaikkiin pelimaailman objekteihin pelissämme, ja ohjailimme niiden avulla objektien käytöstä. Lähestulkoon kaikki pelin objektit tarvitsevat jonkinlaisen skriptan, jotta ne tuntuisivat realistisilta.

4.5.6 Panoraamakuvien käyttäminen skybox-materiaalina

Panoraamakuvia käytimme pelissämme peliympäristön horisontin luomiseen. Tätä horisonttia kutsutaan Unityssä skyboxiksi. Se on pelin ympäristö, joka näyttää siltä, että se on aina äärettömän kaukana pelaajan näkökulmasta, siitä huolimatta liikkuuko pelaaja vai ei. Panoraamakuvia voi myös käyttää erinomaisesti hyödyksi myös esimerkiksi pallon muotoisissa peliobjekteissa. Kokeilimme esimerkiksi tekniikkaa, jossa käytämme panoraamakuvaa tekstuurina pallossa, jonka sisälle pelaajan kamera sijoitetaan. Panoraamakuvatekstuuriin saa pallon sisälle esimerkiksi 3D-mallinnusohjelmien flip normals -komennolla, joka kääntää tekstuuripinnan pallon sisälle.



Kuvio 21. Panoraamakuvatekstuurin asetukset

Kun panoraamatekstuureita käytetään skybox-elementtinä, täytyy tekstuurin asetuksia hieman muokata. Texture typeksi valitaan advanced, mapping asetus täytyy asettaa latitude-longitude layout (cylindrical) kohtaan. Maksimikoko on hyvä asettaa 4096 resoluutioon.

Tämän jälkeen luodaan Unityn valikosta create-kohdasta uusi tyhjä materiaali. Shaderiksi valitaan Skybox/cubemap, jonka jälkeen muokattu panoraamatekstuuri voidaan raahata materiaalin tekstuurilaatikkoon. Tämän jälkeen valitaan asetukset kohdasta window>lighting, ja siirretään uusi skybox-materiaali skybox-material -laatikkoon. Tämän jälkeen pelialueen ympäristö muuttuu panoraamakuvan kaltaiseksi.

Pelkän skyboxin käyttäminen ympäristönä ei ole suotavaa, sillä ympäristö näyttää aina olevan äärettömän matkan päässä joka suuntaan pelaajasta. Tämä rikkoo im-

mersion, sillä varsinkin VR-tekniikkaa hyödynnettäessä pelaaja näyttää olevan äärimmäisen pieni ympäristöön nähden. Päädyimmekin omassa pelissämme prototyypitestauksen jälkeen luomaan pelaajalle erillisen peliympäristön, joka on mökki, jonka ikkunoista näkyy skybox-ympäristö. Tämä tekniikka luo immersivemmän kokemuksen, sillä mökin rajat rikkovat panoraamakuvaa sen verran, että pelaaja vaikuttaa oikean kokoiselta ympäristöönsä nähden ja ikkunoista näkyvä maisema tuntuu realistisemmalta.

4.5.7 Äänien ja videotekstuurien käyttäminen Unityssä

Videotekstuurit eivät suoraan toimi Unityn päällä. Tätä varten koneelle on asennettava Apple QuickTime -sovellus, jota Unity käyttää hyödykseen kääntäessään videoita itselleen sopivaksi. Tämän lisäksi on suositeltavaa käynnistää Unity järjestelmänvalvojana, jotta kääntäminen onnistuu. Unity käyttää .ogg-muotoisia videoita.

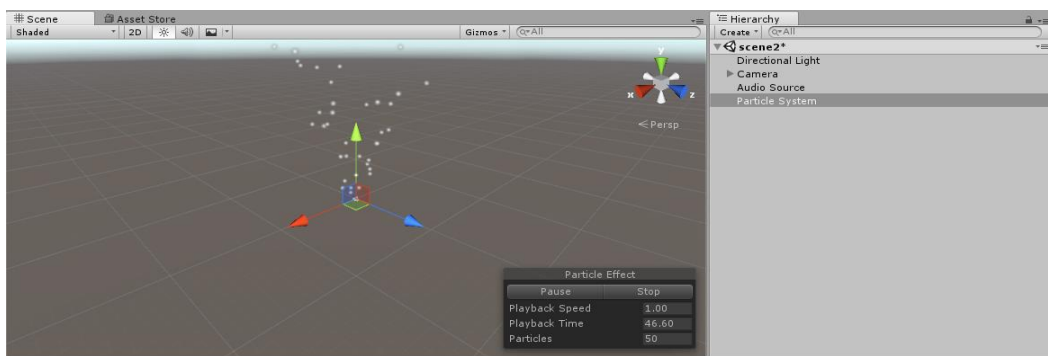
Videotekstuureissa tulee myös ottaa huomioon, että sekä ääni että kuva täytyy käynnistää erikseen skriptojen avulla. Tämä johtuu siitä, että Unity ei ymmärrä ääntä tekstuureihin liittyväksi osaksi.

Videotekstuureihin voidaan viitata skriptoissa tietotyypillä `MovieTexture`, ja niitä käytetään samaan tapaan kuin muitakin tekstuureita; ne raahataan uuden materiaalin tekstuurilaatikkoon. Videoiden äänet voi toistaa raahaamalla pelkän videon ääniraidan erikseen luotuun `sound source` -oliobjektin `clip`-laatikkoon. Sekä video että tekstuuri tulee käynnistää erikseen esimerkiksi komennolla `movTexture.Play()`, `soundSource.Play()`. Ääniä ei voi toistaa, mikäli ne eivät ole erikseen pelimaailmassa tai liitettynä johonkin äänilähteeseen. Erikseen pelimaailmaan liitettävät äänet lähtevät pyörimään automaattisesti, äänilähteeseen liitetyt äänet taas käynnistyvät silloin, kun äänilähde on lisätty pelimaailmaan ja objekti on ”`awake`”-tilassa. Tätä hallitaan ”`play on awake`”-valintalaatikolla äänilähteen komponenteissa. Mikäli valintalaatikko jätetään raksittamatta, täytyy äänilähde käynnistää manuaalisesti esimerkiksi skriptalla. Erikseen liitettyjä ääniä on hyvä käyttää esimerkiksi taustamelun luomiseen tai valikkomusiikkiin. Videotekstuurit toimivat myös skybox-materiaaleina.

Videotekstuureita kannattaa käyttää harkiten, sillä videot haukkaavat suuren osan laitteen tehoista ja voivat hidastaa ruudunpäivitystä sekä peliä. Etenkin VR-ympäristössä tämä on huomioitava, sillä jo laitteen käyttäminen vaatii suurin piirtein kolminkertaisen tehon laitteelta normaaleihin peleihin verrattuna. Video myöskin renderöi taustalla jatkuvasti, vaikka se ei missään näkyisikään. Video kannattaa myös pysäyttää manuaalisesti skriptan avulla, sillä videon renderöinti jatkuu joka tapauksessa muuten, mikäli näyttämö ei vaihdu tai peli sulkeudu. Omassa pelissä käytimme videoita vain pelin välivideoissa, joissa sijoitimme videotekstuurit skybox-materiaalin seinämälle. Ääniä käytimme niin taustamelun luomiseen, valikkomusiikkiin kuin salamavalon käyttämiseenkin. Myös hirviön käyttöön liittyvissä tilanteissa käytimme ääniä hyödyksi, jotta pelaajan on helpompi löytää viholliset. Hyödynsimme edellä mainittua ”play on awake” -valintaruutua vihollisten uudelleensyntymisessä, jolla mahdollistimme hirviön ääntelyn jokaisella uudelleensyntymiskerralla.

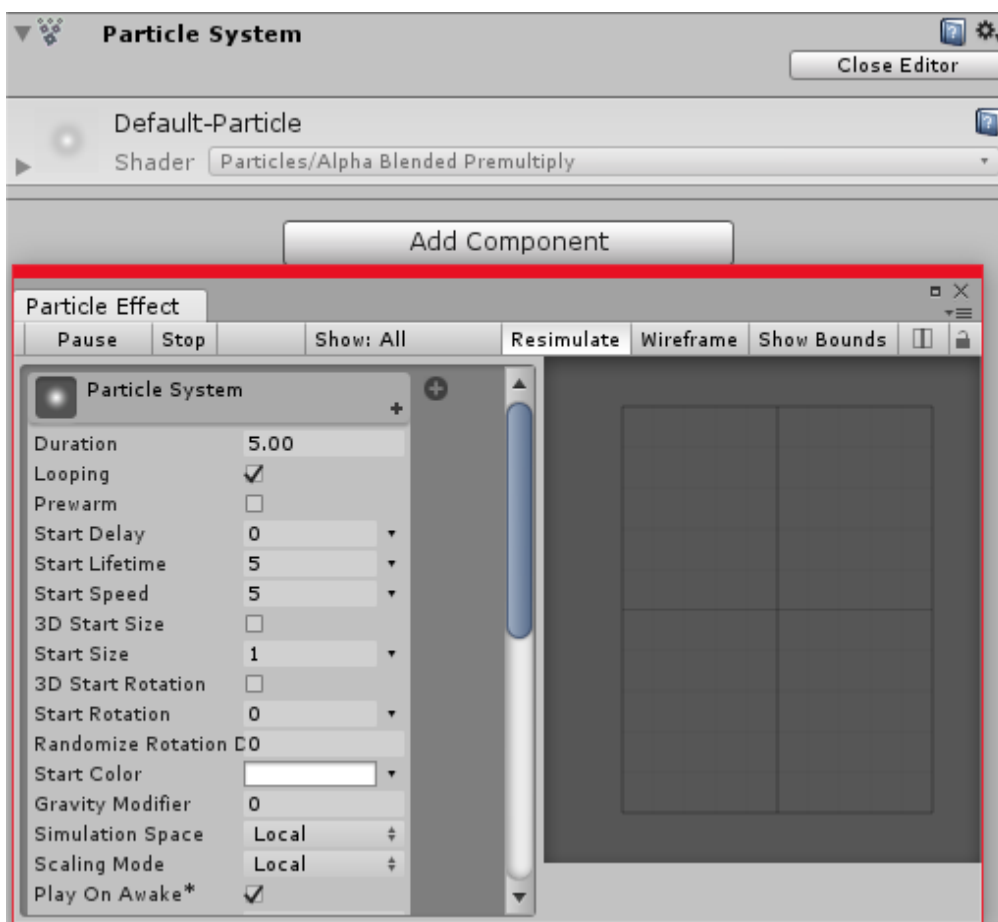
4.5.8 Particle Systemin käyttäminen Unityssä

”Particle system” eli partikkelisysteemi on Unityn peliobjektin tyyppi, joka mahdollistaa useiden pienten tekstuurien tai objektien luomisen ja liikuttamisen peliobjektin transform-komponentin kohdasta. Partikkelisysteemillä on siis mahdollista tehdä peliympäristöön esimerkiksi vesi tai lumisadetta. Partikkelisysteemin muokkaamismahdollisuudet ovat todella monipuoliset, mikä tekee siitä hyvän työkalun. Jokaiselle partikkelisysteemin luomalle partikkelille on mahdollista luoda yksilöllinen collider-komponentti, mikä mahdollistaa esimerkiksi partikkeleiden reagoimisen ympäristöönsä ja törmäilyn pelialueen muihin objekteihin.

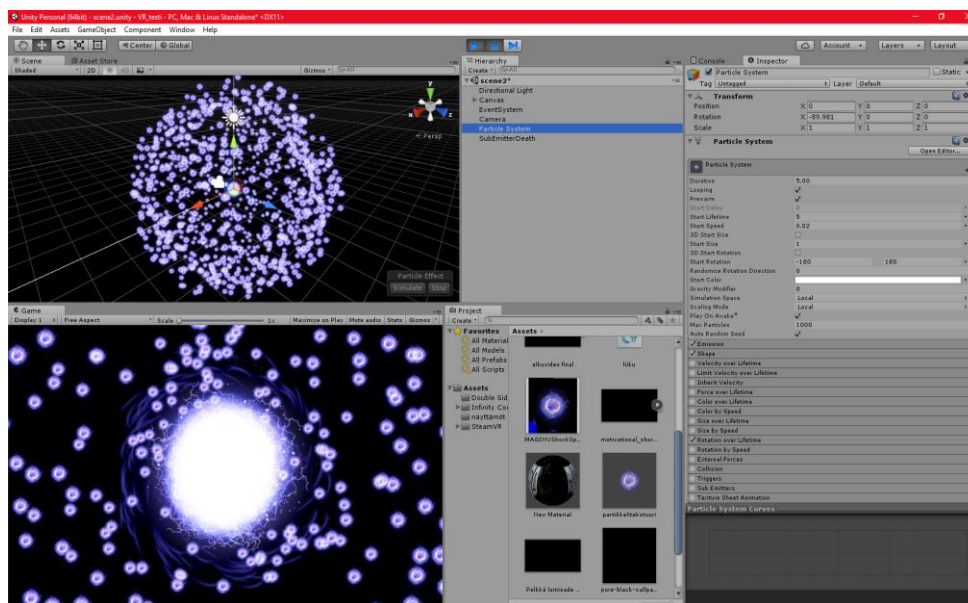


Kuvio 22. Partikkelisysteemi pelin näyttämöllä

Partikkelisysteemi voidaan luoda klikkaamalla oikealla hiiren painikkeella hierarkiaikkunasta ja valitsemalla ilmestyvästä valikosta ”particle system”. Tämä komento luo kuvassa näkyvän partikkelisysteemin, joka syöksee objektin olinpaikasta valkoisia partikkeleita ylöspäin. Tämän lisäksi partikkelisysteemi voidaan esimerkiksi lisätä erikseen tietyn peliohjelman komponentiksi add component -painikkeella.



Kuvio 23. Partikkelisysteemin muokkausikkuna



Kuvio 24. Muokattu partikkelisysteemi

Kuvio 24 on partikkelisysteemin muokkausikkuna, joka mahdollistaa komponentin monipuolisen muokkaamisen. Samaan systeemiin voidaan myös lisätä useita erilaisia partikkeleita. Muokkaaminen on tehty hyvin käyttäjäystävälliseksi ja sen avulla voidaan saavuttaa hyvin erilaisia lopputuloksia. Käytimme partikkelisysteemiä esimerkiksi hirviön kuolinanimaatioissa, jolloin pystyimme luomaan näennäisen efektin siitä, että hirviö ikään kuin haihtuisi sen tuhoutuessa.

4.6 Audiovisuaalisen materiaalin luominen

Tässä luvussa kerromme audiovisuaalisen materiaalin luomisen vaiheista. Yritämme edetä mahdollisimman kronologisesti, sillä poikkeuksella, että ohjelmat käydään läpi yksi kerrallaan, eikä hypitä niiden välillä, kuten oikeasti projektin edetessä tehtiin. Materiaalia luotiin sykleissä kohtuullisen pitkällä aikavälillä ja ”kaksi askelta eteen - yksi taakse” tuntui olevan tämän osa-alueen toistuva teema lähes koko projektin ajan. Ongelmilta ei siis vältytty, mutta ratkaisut lopulta löydettiin.

Emme lähde käymään asioita läpi liian seikkaperäisesti, sillä esimerkiksi se, miten luotiin joku tietty 3D-malli, ei ole kovin relevanttia opinnäytetyön aiheen kannalta. Pyrimme kuitenkin näyttämään hieman eri vaiheita ja malleja, joiden kautta päädyttiin lopullisiin pelissä käytettyihin malleihin. Sisällytän tähän lukuun myös muutamia sellaisia kuvia, joita ei itse pelaaja pelissä näe, joko johtuen VR-tekniikan

rajoittuneisuudesta, tai siitä, että ne sijaitsevat varsinaisen pelialueen ulkopuolella. Olen tehnyt näitä siis muutamia, lähinnä oman osaamisen kehittämiseksi sekä näyttämiseksi.

4.7 Panoraamakuvauksen vaiheet

Panoraamakuvausta varten saimme hyvät ohjeet Vaasan ammattikorkeakoululta Yliopettaja Päivi Rajalan kautta. Ohjeissa oli käytetty pitkälti samaa panoraamalaitteistoa sekä ohjelmia kuin mekin käytimme.

Kappale on jaettu kahteen osaan, joissa ensimmäisessä käydään läpi hieman kuvaustekniikkaan liittyviä asioita, ja toisessa kerrotaan hieman kuvien yhdistämisestä valmiiksi pallopanoraamaksi. Lopussa kerrotaan vielä eteen tulleista ongelmakohdista ja niihin löydetyistä ratkaisuista.

4.7.1 Panoraamakuvauksen laitteisto ja kuvien ottaminen

Mietimme kuvauspaikkaa hetken, kunnes mieleen tuli vanha louhos Raippaluodon suunnalta, josta saattaisi sopiva paikka löytyä. Olimme yhtä mieltä siitä, että taustassa olisi hyvä näkyä vettä, mikäli se olisi mahdollista. Emme kuitenkaan halunneet minkäänlaista asutusta taustalle immersiota rikkomaan. Otimme siis laitteiston mukaan, kävimme katsomassa paikan, ja totesimme sen sopivan hyvin tarkoitukseemme.

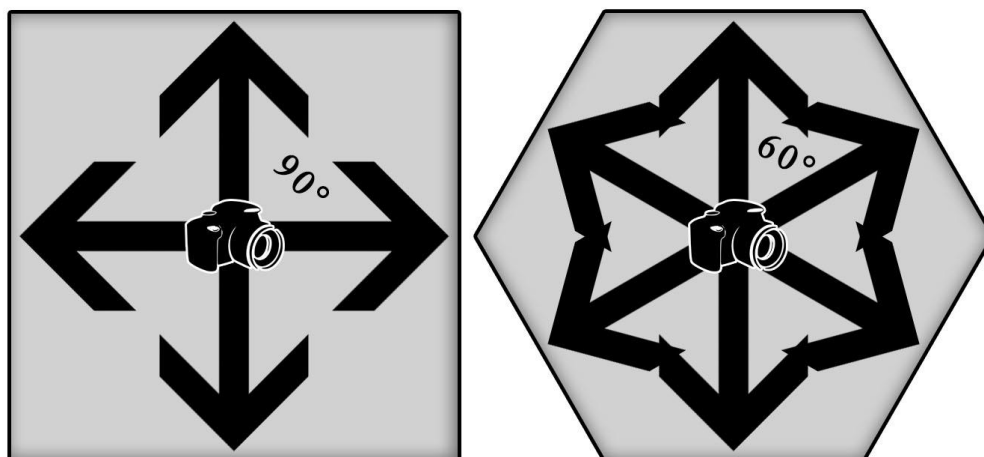
Kuvaus osui marraskuun puolivälin paikkeille. Maassa oli juuri pelin tunnelmaan sopiva pari päivää pysynyt ensilumi ja pihalla alkoi kohtuullisen aikaisin hämärtää. Asetelma siis vastasi aika hyvin sitä visiota, jota olimme ajatelleetkin. Pystyimme panoraamalaitteiston, teimme tarvittavat säädöt kolmijalkaan sekä panoraamapäähän, otimme käyttöön ja säädimme manuaaliasetukset, jonka jälkeen aloitimme kuvaamisen.

Otimme kuvat muutaman kerran vaihtelevilla asetuksilla:

- ❖ Aukon koko F11-F16
- ❖ Suljinaika 1/400-1/500

- ❖ ISO-alue 600-800
- ❖ .jpg & .png
- ❖ Manuaalitarkennus

Panoraamakuvauksen voi ko. laitteistolla toteuttaa kahdella eri tavalla, eli joko neljällä kuvalla tai kuudella kuvalla. Neljällä kuvalla kuvia otetaan: 1 edestä, 1 oikealta, 1 takaa, 1 vasemmalta. Kameraa siis käännetään jokaisen kuvan välissä 90° , jolloin viimeisen kuvan jälkeen on tehty täysi kierros. Toisessa tavassa, jossa otetaan kuusi kuvaa, kameraa käännetään kuvien välissä 60° ja kuvista muodostuu neliön sijaan heksagoni, eli kuusikulmio. Kuuden kuvan tekniikalla lopputulos on luonnollisesti parempi, joten päädyimme käyttämään sitä.



Kuvio 25. Neljän ja kuuden kuvan kuvaustekniikka

Kamera kiinnitettiin jalustaan lappelleen ja suunnattiin hieman yläviistoon. Panoraamapäässä oli astemitta, jonka avulla 60 asteen käännökset oli helppo tehdä ja laatu pysyi tasalaatuisena kuvien välillä. Alustan suoristamista sekä asettelua helpotti kolmijalkaan sekä panoraamapäähän integroidut vatupassit. Kuvioissa 26-27 muutama kuva panoraamakuvaukseen käytetystä järjestelmästä, joista on selkeyden vuoksi poistettu pihalla esiintyvä sinertävä sävy ja valoisuutta lisätty. Kuviossa 28 on esimerkkipicture panoraaman kahdesta ensimmäisestä kuvasta ilman sinertävyyden poistoa.



Kuvio 26. Kameran kiinnitys jalustaan



Kuvio 27. Kuva laitteiston takaa sekä kameran asetuksista

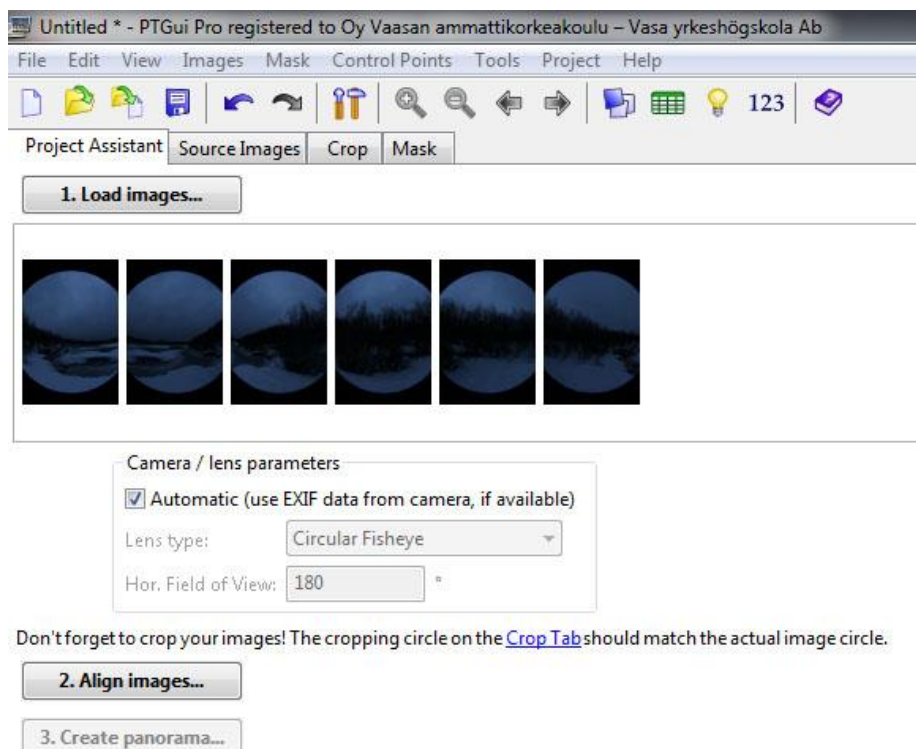


Kuvio 28. Panoraaman ensimmäiset kuvaa ennen suurempia muokkauksia

Sävyero kuvien välillä onkin melko suuri, ja pelissä käytetty tausta vastaakin huomattavasti enemmän Kuvio 28:n kuvia, joita on vielä tunnelman vuoksi tummennettu.

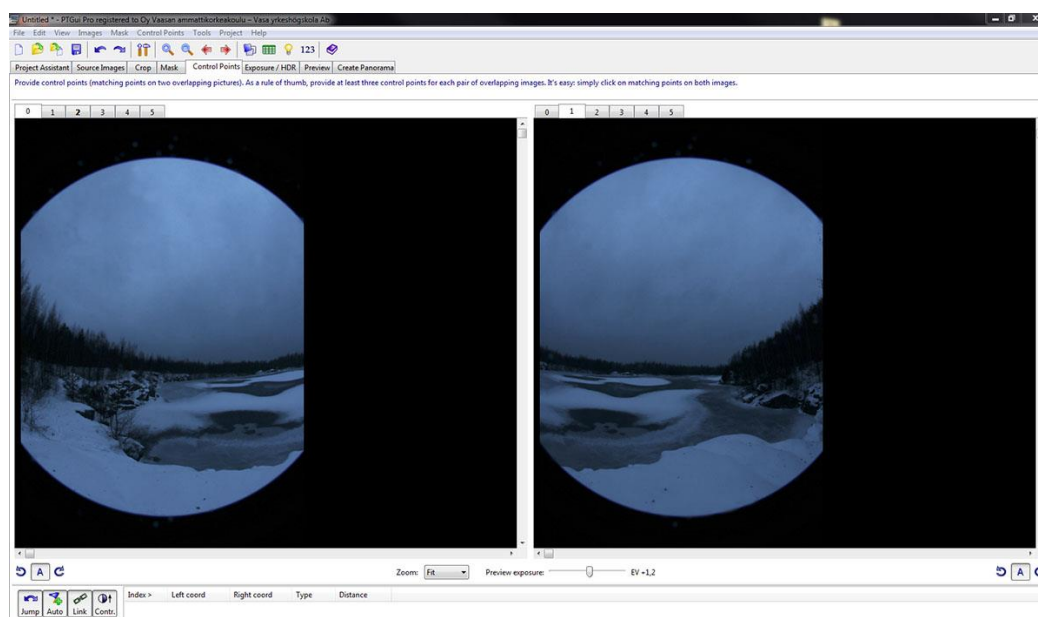
4.7.2 Kuvien yhdistäminen ja muuntaminen panoraamaksi

Kun kuvat on otettu, voidaan ne yhdistää panoraamakuvaksi PTGui-ohjelman avulla. Kuvat tuodaan ohjelmaan ”Load images” -painikkeen avulla. Sen jälkeen klikataan ”Align images” -painiketta, jolloin ohjelma etsii itsenäisesti ”kiintopisteitä” (control points) kuvien väliltä. Pisteinä voi toimia jokin kuvissa näkyvä objekti, eli vaikkapa: keppi, kivi, puu jne. Jos ohjelma ei löydä kiintopisteitä automaattisesti, tulee ne määrittää manuaalisesti.

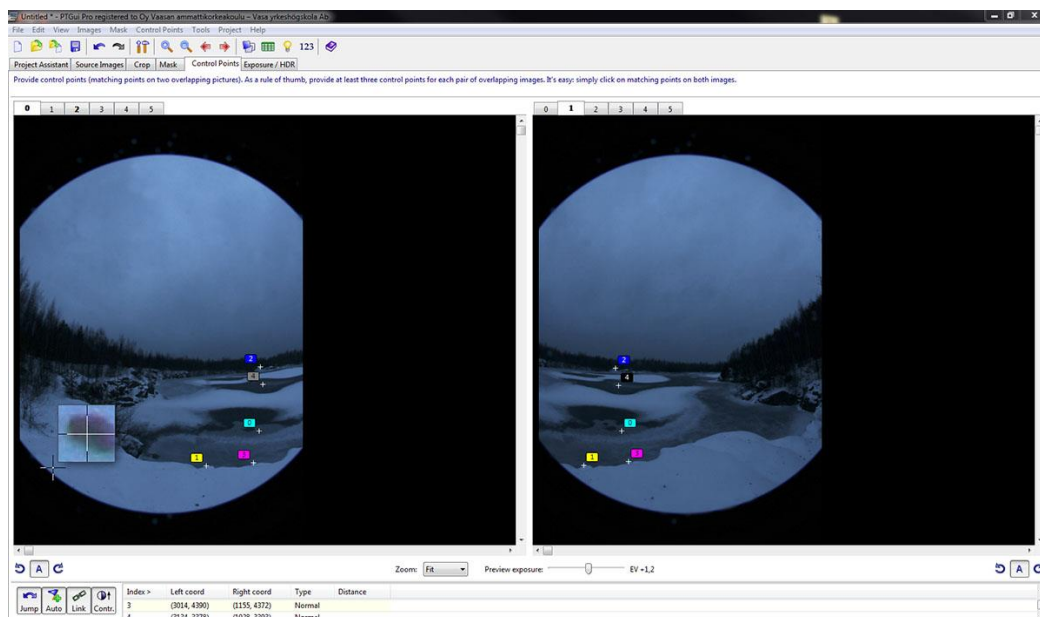


Kuvio 29. Perusnäkö PTGui ohjelmassa kuvien latauksen jälkeen

Tässä projektissa ohjelma löysi pisteet vain kahden kuvan väliltä automaattisesti, koska maasto oli hyvin samankaltaista kuvien välillä. Kun asetin kaksi kiintopistettä manuaalisesti kuvien väliin, ohjelma tuntui sen jälkeen ymmärtävän, missä saumakohtaan tulisi olla.

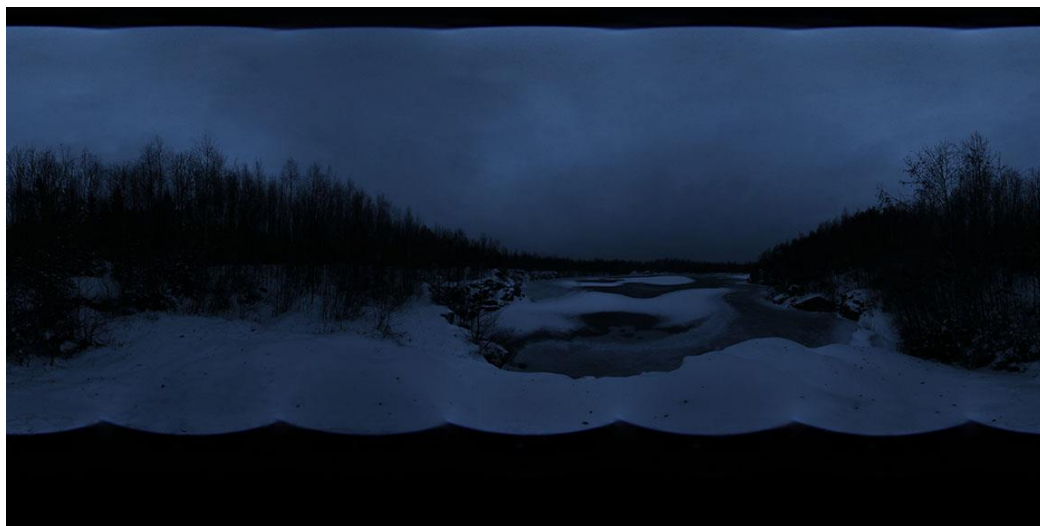


Kuvio 30. Näkö ilman kiintopisteitä



Kuvio 31. Näkymä kiintopisteiden lisäämisen jälkeen

Kun kaikkiin kuviin on määritelty riittävä määrä kiintopisteitä, eli vähintään kolmen, voidaan kuvasta koostaa panoraama, painamalla ”Create panorama” -painiketta. Silloin ohjelma koostaa määritellyjä kiintopisteitä apuna käyttäen saumattoman panoraaman kuvista.



Kuvio 32. Kuvista koostettu panoraama ennen muokkausta

Kun kuvat on muunnettu panoraamaksi, voidaan sitä helposti muokata yhtenäisenä kuvana. Kuvaan ei tehty kovin paljon suurempia muutoksia, kuin mustien reunojen poisto ylä- sekä alareunasta ”content aware” -työkalun avulla Photoshopissa.



Kuvio 33. Valmis panoraama reunojen poiston jälkeen

Panoraama ei ole täysin saumaton kohdista, joista musta reuna on poistettu. Se alue jää silti tässä projektissa täysin peittoon pelaajalta, sillä se sijaitsee varsinaisen pelialueen (mökin) lattian alapuolella. Emme siis nähneet tarvetta manuaaliseen korjailuun. Osasy syy saumoihin löytyy pienimuotoisesta ongelmasta, johon törmäsimme kuvausreissun jälkeen. Kuvauspaikalla satoi hieman lunta, ja lumihiutaleiden onnistui jättää kuviin tummia mustia pisteitä, jotka poistettiin myöhemmin manuaalisesti ”clone stamp” -työkalun avulla Photoshopissa. Sama tehtiin myös osalle jättämistämme jalanjäljistä, jotta ne eivät näkyisi vahingossakaan pelissä.



Kuvio 34. Kaksi täysin muokkaamatonta kuvaa mustine pisteineen

Muokkaamattomissa kuvissa näkyvät mustat pisteet muodostuivatkin pitkälti ainoksi oikeaksi ongelmaksi tässä osassa projektia. Olimme ajatelleet ottaa kuvat alun perin RAW-formaatissa, mutta alustavan testin perusteella huomasimme, ettei sille ole vielä vakiintunutta tiedostomuotoa, vaan tiedoston tyyppi vaihtelee kameran valmistajasta riippuen. Yritimme myös koostaa panoraamasta valmiin pallopanoraaman Pano2VR-ohjelman avulla. Kyseessä oli kuitenkin trial-versio, jossa on samat ominaisuudet kuin kokoversiossa, mutta pallopanoraamaksi muuttamisen jälkeen, kuva on täynnä pieniä vesileimoja, joissa lukee: ”Pano2VR – Trial”.



Kuvio 35. Pano2VR - Trial vesileimat pallopanoraamassa

Pienistä ongelmista huolimatta, kyseessä taisi olla koko projektin suoraviivaisin vaihe, ja valmis pelissä käytetty pallopanoraama koostettiin lopulta ilman erillistä ohjelmaa Unityn sisällä. (File-Extensions.org, 2017)

4.8 3D-mallien luominen

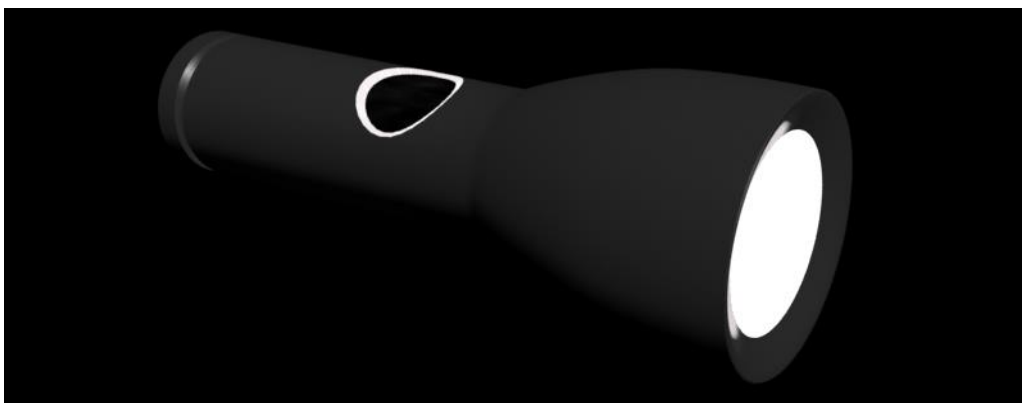
3D-mallien tekemisen aloitettiin oikeastaan ihan projektin alkuvaiheilla, ennen panoraamakuvan ottamista. Halusimme alussa lähinnä testata, kuinka helposti saisimme 3ds Maxin sekä Unityn toimimaan keskenään erilaisten mallien sekä materiaalien kanssa.

Tämä kappale on jaettu kahteen osaan selkeyden takia. Tavoitteena on käydä asiat läpi mahdollisimman kronologisesti, aloittaen ensin malleista, jotka luotiin peliin testejä varten, eli ns. prototyypeistä. Sen jälkeen kerromme hieman valmiista malleista, joita päädyimme pelissä lopulta käyttämään. Pyrimme molempien osioiden lopussa tuomaan myös esille merkittävimmät havainnot sekä pulmat, joita 3D-mallien kanssa kohdattiin.

4.8.1 Prototyypit

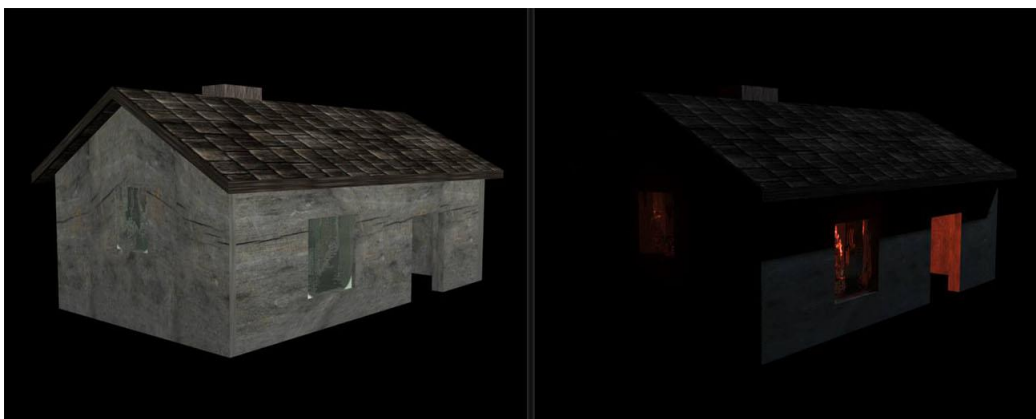
Tiesimme, että pelissämme tulisi olemaan joku valonlähde, eli taskulamppu, lyhty tms. Ensimmäinen malli oli siis yksinkertainen taskulamppu, joka luotiin tekemällä poikkileikkaus taskulampusta ”line” -työkalun avulla. Sen jälkeen tarvittavat modifikaattorit lisättiin poikkileikkaukseen, joita käyttämällä luotiin taskulampun

runko. Rungon ulkomuotoa muokattiin hieman, siihen lisättiin nappi sekä varsinaisen lamppu-osa. Sen jälkeen lamppu muutettiin polygoniksi, eri osille lisättiin numeroinnit ”multi/sub-object” -teksturoidintia varten. Kyseisessä tapaa käyttämällä on mahdollista sisällyttää esimerkiksi: napin, rungon, valon jne. materiaalit saman tekstuurin alle, jolloin jokaiselle niistä ei tarvitse varata/luoda omaa tekstuuria. Taskulampun materiaalit on otettu 3ds Maxin omasta materiaalikirjastoa, ja niitä on muokattu hieman. Lopputulos taskulampun 3D-mallista renderöitynä kuviossa 36.



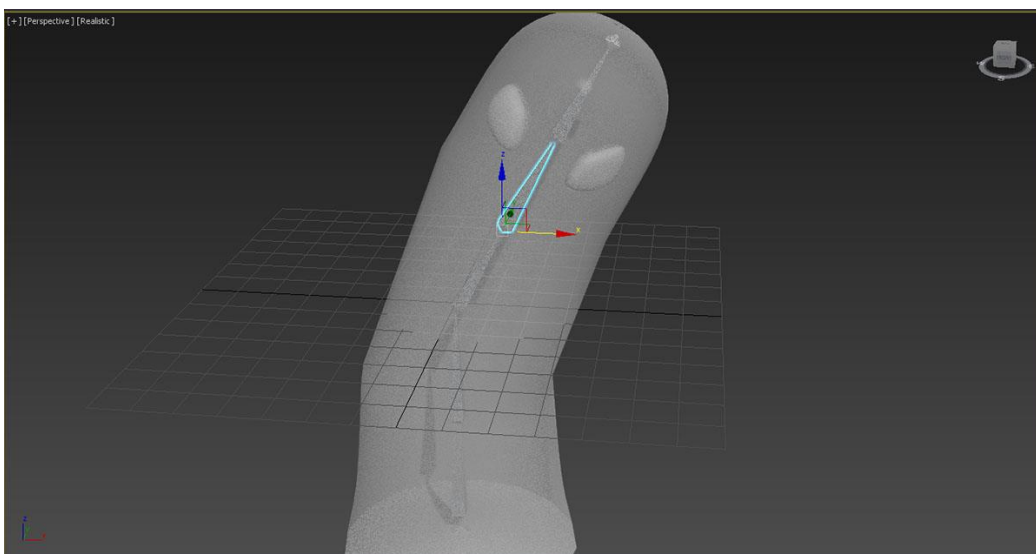
Kuvio 36. Projektin ensimmäinen 3D-malli

Seuraavaksi tiesimme tarvitsemamme mökin, eli varsinaisen pelialueen, jossa pelaaja pääsisi liikkumaan. Testimökiksi tehtiin todella yksinkertaisen malli paria objektia käyttämällä. Objektit muutettiin polygoneiksi, niistä muokattiin jotain mökin tapaista, ja valmista mallia käytettiin testitarkoituksiin projektin edetessä. Tekstuureina käytettiin yksinkertaisia saumattomia kuvapohjaisia tekstuureita, jotka etsimme googlesta. Kuviossa 37 kuva ensimmäisestä testimökistä 3ds Maxissa ilman valoeffektejä, ja valoeffektien kanssa.



Kuvio 37. Mökin prototyyppi

Seuraavaksi tarvitsimme peliin hirviön, jotta voisimme kokeilla pelin toiminnallista puolta sekä luoda pisteet, mihin hirviö pelissä ilmestyisi. Testitarkoitukseen luotiin yksinkertaisen perinteistä haamua muistuttava olento samaa poikkileikkaustyyliä käyttäen, jota käytettiin myös taskulampussa. Hirviön teksturointiin käytettiin 3ds Maxin omia materiaaleja. Pystyimme tässä samalla myös testaamaan, miten animoitu malli käyttäytyisi Unityn sisällä. Olentoon siis lisättiin myös yksinkertainen ”bone structure” ja se animoitiin heilumaan hitaasti puolelta toiselle 3ds Maxin avulla. Kuviossa 38 on kuva pelin prototyyppihirviöstä sekä sen sisäisestä yksinkertaisesta luustosta.

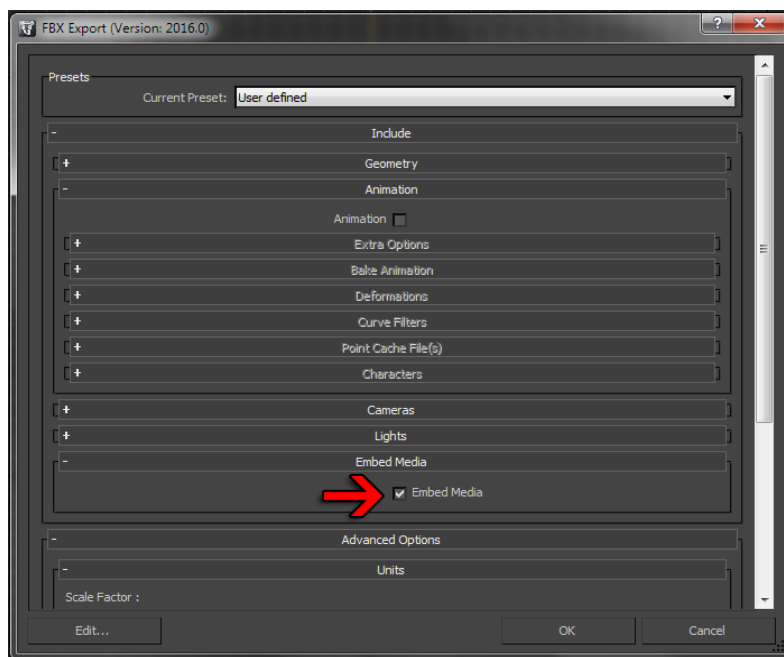


Kuvio 38. Hirviön ensimmäinen versio luineen osittain läpinäkyvänä

3ds Maxissa luotu animointi toimi hyvin Unityn sisällä, vaikkakin nopeus ei vastannut 3ds Maxissa käytettyä 0,25x arvoa. Nopeuden muuttaminen oli silti vaivastonta, jonka jälkeen animaatio toimi odotetusti. Toinen ongelma johon törmäsimme

jo ensimmäisessä taskulampun mallissa, oli joidenkin objektien läpinäkyvyys. Esimerkiksi taskulampun ”lamppu-osa”, joka oli luotu käyttämällä kartion muotoista objektia, näkyi taskulampun rungosta läpi. Syy tähän oli ilmeisesti se, että taskulampun rungon luonnin yhteydessä, olimme ikään kuin teksturoineet ko. mallin sisäpuolen. Tämä selvisi vasta myöhemmin projektissa salamavaloa tehdessä, jossa sama ongelma toistui. 3ds Maxin sisällä onkin mahdollista käyttää ”flip” -toimintoa, joka kääntää mallin ns. ”oikeinpäin”, jolloin tekstuuri jää mallin ulkopuolelle, kuten on yleensä tarkoituskin. Toinen ratkaisu jonka ongelmaan löysimme, oli Unity Asset Storen kautta ladattavissa oleva Double Sided Shaders niminen resursipaketti, jolla oli mahdollista asettaa mallissa käytetyt tekstuurit siten, että ne olivat sekä mallin sisä- että ulkopinnalla. Tämä tietysti luo ylimääräistä raskautusta tietokoneelle, koska objektien sisäpuoli ei monestikaan peleissä näy, joten sitä on myöskin turha teksturoida, sillä se vaatii ylimääräistä laskentatehoa. Tämän kokoisessa yksittäisessä objektissa kyseinen seikka ei silti ollut merkittävä.

Eräs merkittävimmistä seikoista, jonka 3ds Maxin sekä Unityn välillä huomattiin, oli materiaalien siirtyminen ohjelmien välillä. Käytimme 3ds Maxista exportatessa .fbx-formaattia, sillä se oli hyvin yhteensopiva Unityn kanssa, ja ko. tiedostomuotoon oli myös mahdollista ”sisäänkirjoittaa” materiaalitietoja, käyttäen ”embed media” -ominaisuutta exportin yhteydessä. Tällöin materiaalit kulkeutuvat fbx-tiedoston mukana ohjelmien välillä ja Unityyn importatessa Unity korvaa 3ds Maxin sisällä asetetut tekstuurit omilla lähes vastaavanlaisilla materiaaleillaan.



Kuvio 39. Embed media toiminto exporttauksessa

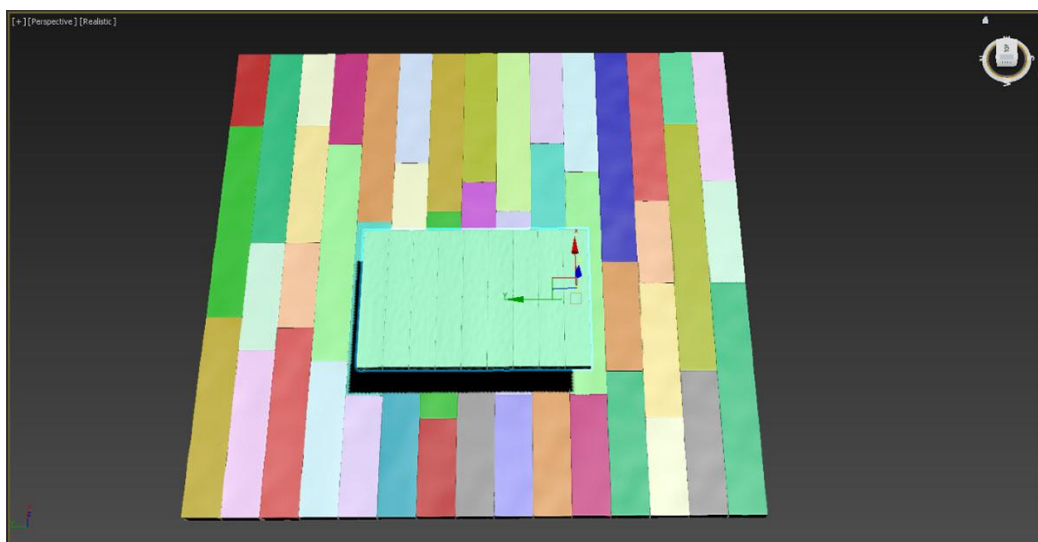
Prosessi ei ollut ihan saumaton, sillä materiaalit eivät ulkonäöltään täysin vastanneet 3ds Maxin sisällä valittuja tekstuureita. Niiden muokkaaminen sekä vaihtaminen olivat silti Unityn sisällä kohtuullisen yksinkertainen toimenpide, jolloin lopputulos saatiin mahdollisimman lähelle alkuperäistä.

4.8.2 Valmiit mallit

Kun testialue sekä pelin perustoiminnot olivat kasassa, aloimme luoda varsinaisia pelissä käytettäviä 3D-malleja. Työ aloitettiin mökin lattiasta, sillä se loisi pohjan koko pelialueelle ja toimisi hyvänä testialustana myös edistyneemmälle teksturoinnille Substance Painter 2:n sisällä.

Lattia luotiin muutaman laatikon avulla 3ds Maxin sisällä, jotka asetettiin lattian muotoon vierekkäin. Tämän jälkeen niihin lisättiin tarvittava määrän segmenttejä sekä tehtiin tarvittavat muokkaukset mm. FFD 4x4x4 modifikaattorin sekä ”noisen” avulla, joilla pyrittiin simuloimaan vanhan lankkulattian elämistä, rakoja ja puun hieman rosoista pintaa. Mittakaava ensimmäiseen versioon hahmoteltiin silmämääräisesti jo Unityssa testattujen peliä varten tehtyjen mallien avulla, ja lopputuloksena oli aivan liian pieni, pinta-alaltaan lähinnä leikkimökin lattiaa muistuttava tekele.

Testasimme ko. mallia silti Unityn sisällä ja kaikki vaikutti mittakaavaa lukuunottamatta toimivan hyvin. Seuraavan version päätettiin luoda käyttäen apuna 3ds Maxin sisällä olevaa metriikkaa sekä mittausta, joilla pystyttiin määrittämään lattian suurin piirteinen pinta-ala. Erilaisiin lankkuihin liittyvää tietoa etsittiin myös netistä, joita käytettiin antamaan suuntaa lopullisille mitoille. Päädyimme silti tekemään lankuista hieman oikean elämän vastineita suurempia, sillä lopullisen mallin jo ~30m²:n pinta-alalla yksittäisiä lankkuja tuli yhteensä 48 kappaletta, joista jokainen piti muokata erikseen, hyvän lopputuloksen aikaansaamiseksi. Viimeistä mallia ei myöskään tehty koko lattian pituisista lankuista, sillä se vastasi enemmän oikeaa lankkulattiaa ja elävöitti lopputulosta. Kuviossa 40 on esimerkkikuva vanhasta lattiasta uuden lattian päällä mittakaavan selventämiseksi.

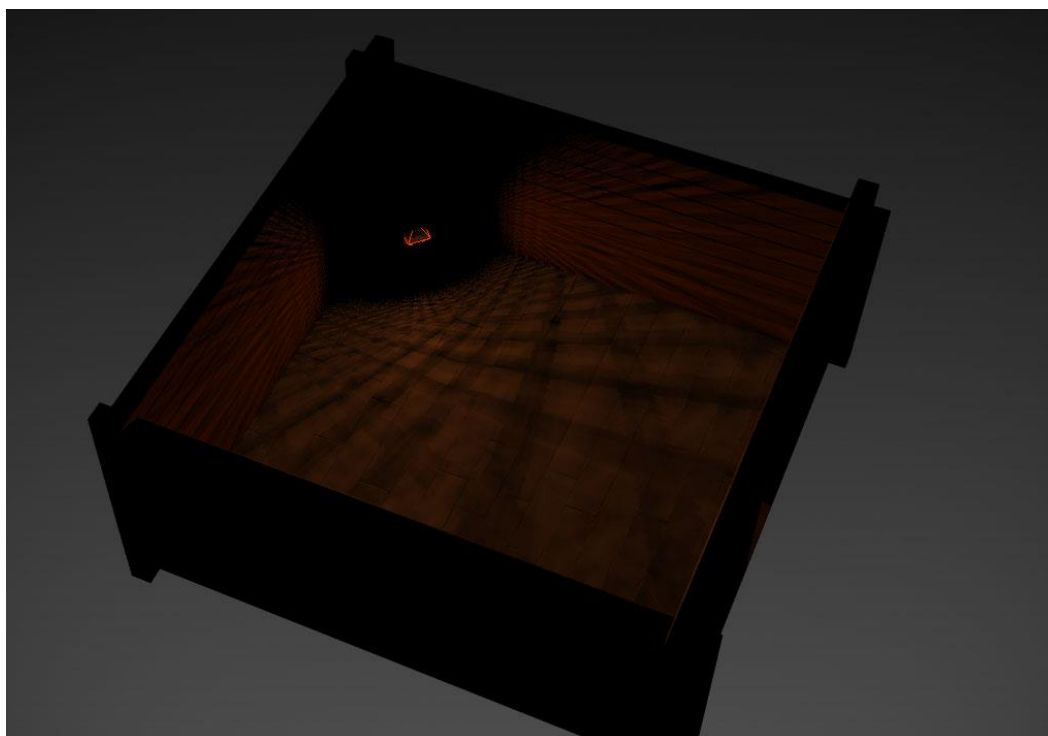


Kuvio 40. Lattiat päällekkäin 3ds Maxissa

Kun lattia oli valmis, luotiin mökkiin seinät. Tässäkin otettiin suuntaa koosta sekä ulkonäöstä käyttäen apuna kuvia oikeista hirsistä. Hirret luotiin käyttämällä laatikko-objekteja, joista viistettiin kulmat, kuten joissain hirsissä on tapana. Laatikot ladottiin päällekkäin, joista koostettiin seinät, ja lopuksi laatikot liitettiin siten toisiinsa, että jokaisesta seinästä tuli yhtenäinen pala. Tässä vaiheessa koitettiin samalla myös hieman hahmotella jotain takan tapaista, jota tulisimme peliin tarvitsemaan. Lopulta päädyttiin tekemään perinteisen takan sijaan vanhanaikaisen pönttö-

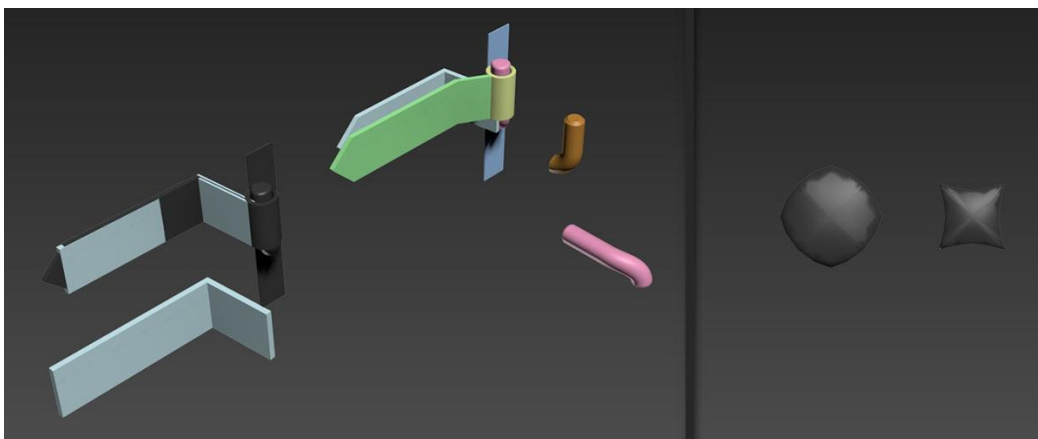
uunin, ja tässäkin käytettiin suuntaa antamaan kuvia vanhoista sekä uusista pönttöuuneista. Takkaan lisättiin testiksi 3ds Maxin sisällä myös pieni valo, jotta näkisimme hieman, miltä se voisi näyttää pelissä.

Pönttöuuni oli alun perin tarkoitettu lähinnä testiksi sille, jotta näkisimme miten se sopisi perinteisen takan sijaan peliin. Pidimme kuitenkin sen ulkonäöstä ja ratkaisu vaikutti siinä määrin toimivalta, että päätimme käyttää sitä lopullisessa versiossa.



Kuvio 41. Lattia, seinät ja pönttöuuni

Seuraavaksi suunniteltiin ikkunoiden sekä oven paikat, joille tehtiin reiät seiniin ”boolean” -työkalun avulla. Tässäkin käytettiin suurin piirtein oikeita mittoja, jotta lopulliset ikkunat sekä ovet vastaivat mittasuhteilta kohtuullisen hyvin oikean elämän vastineitaan. Sen jälkeen luotiin tarvittavat karmit, sisä- sekä ulkokaton, ovet, ikkunat jne. sekä määritettiin kaikille paloille omat materiaalinsa. Näiden lisäksi tehtiin myös muutamia ”ylimääräisiä” pienempiä objekteja, kuten saranat, lukko-helat, ovenkahvat, kiinnitysniitit jne. Ne eivät varsinaisesti pelissä pelaajalle näy, mutta luovat miellyttävän loppusilauksen, ja näyttävät hyvältä valmiiksi teksturoidussa mökissä.



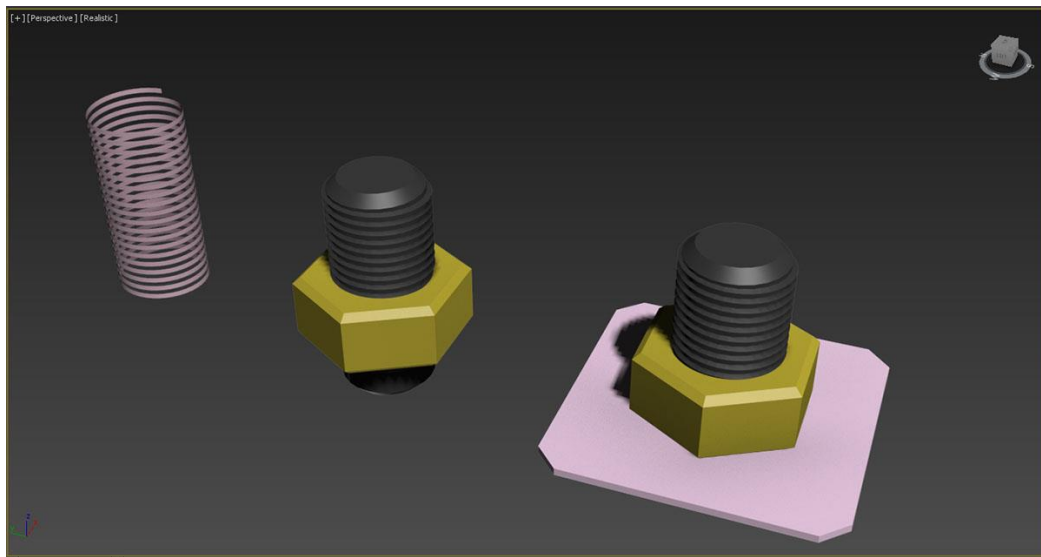
Kuvio 42. Saranoiden hahmottelua ja kiinnitysniitit



Kuvio 43. Valmis mökki 3ds Maxissa

Valmis mökki oli värimaailmaltaan vähintäänkin erikoinen 3ds Maxin sisällä. Sille oli kuitenkin syynsä, sillä teksturoidessa Substance Painter 2:n sisällä, objekteille tuli olla määritelty oma tekstuurinsa, jotta ongelmilta Unityn sisällä vältyttäisiin. Objektien värittäminen eri väreillä helpotti myös hahmottamaan sekä hallitsemaan tekstuureita 3ds Maxin sisällä, sillä valmis mökki ilman irtaimistoa koostui yli kahdestasadasta osasta.

Olimme toki päättäneet heti alusta alkaen, että mökki ei saisi olla täysin tyhjä. Irtaimistoksi peliin onkin siis tarkoitus tulla ainakin pöytä, tuoleja, makuualusta ja/tai -pussi, reppu. Projektin loppuvaiheilla päätettiin tehdä mökkiin myös kattoparrut sekä niiden kiinnikkeet, jotka rikkoivat mielestämme hyvin muuten hieman laatikomaista rakennetta.



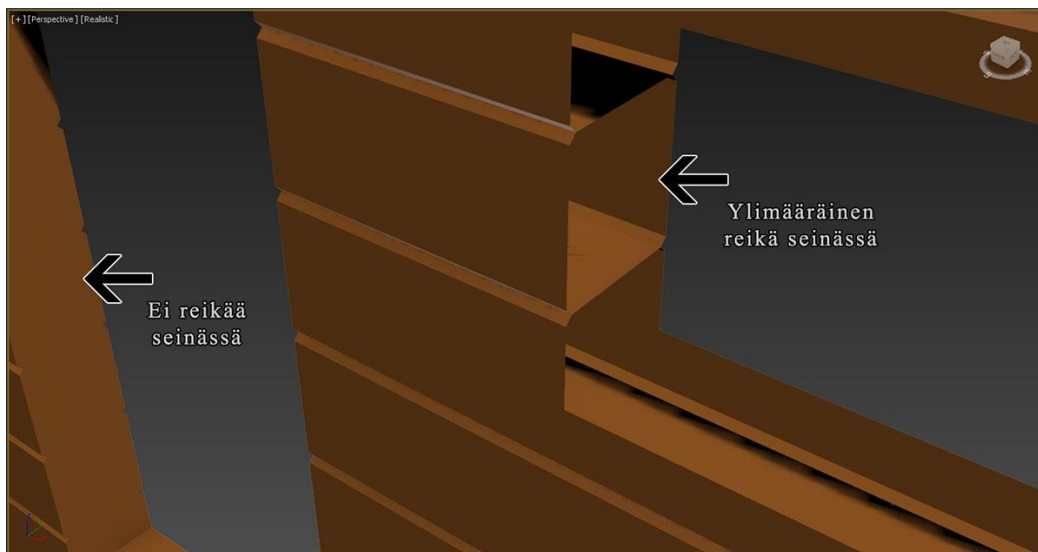
Kuvio 44. Kattoparrun kiinnikkeiden suunnittelua

Mökin luomisen aikana, idea pelin toiminnollisuudesta hieman muuttui, ja perinteisen taskulampun tilalle oli tarkoitus vaihtaa jonkinlainen salamavallo. Tässä kohtaa ilmeni muotoilun kannalta pieni dilemma. Tiesimme, että immersiosyistä halusimme lopullisen salamavalon muistuttavan hieman HTC Viven kapuloita, mutta niiden erikoinen muoto aiheutti haasteita. Lopputulos olikin ehkä liian futuristinen, vaikkakin esteettisesti kohtuullisen hieno



Kuvio 45. Salaman kolme vaihetta: neutraali, välähdys ja lataus

3D-mallien luominen on ollut oikeastaan koko projektin ylivoimaisesti työläin vaihe, erityisesti mökin. Tässä osassa projektia sai hyvän käsityksen siitä, miten työstä on mallien tekeminen oikeassa mittakaavassa. Pieniä ongelmia koko 3D-prosessin aikana oli oikeastaan jatkuvasti. Suurin osa johtui kuitenkin 3ds Maxin epävakaudesta tai erikoisista toiminnoista/toimimattomuudesta. Esimerkiksi joku tietty modifikaattori saattoi toimia pitkän aikaa juuri kuten pitääkin, kunnes se ei yhtäkkiä toiminutkaan. Tai kun seinää rei'itettiin oven ja ikkunoiden takia, saattoi ”boolean” -työkalu poistaa hieman ylimääräistä, vaikka se ei sitä edellisen kahden reiällä tehnyt. Ongelmat eivät siis olleet suuria, vaikkakin välillä äärimmäisen turhauttavia. Valtaosa onneksi ratkesi ohjelman uudelleenkäynnistyksellä, ja moneen pulmatilanteeseen löytyi myös hyviä tutoriaaleja, kunhan vain keksi oikeat sanat joilla etsiä. Kuviossa 46 vielä kuva em. ongelmatilanteesta, jossa reikää tehdessä mallista poistuu ylimääräisiä kohtia.



Kuvio 46. Ontto karmi seinän rei'ittämisen jälkeen

Ratkaisu tähän oli mallin muuttaminen polygoniksi, aukon reunojen valitseminen ja joko ”cap” tai ”bridge” -työkalun käyttäminen reiän paikkaamiseen. Kyseessä vaikutti olevan foorumeiden perusteella aika yleinen ongelma.

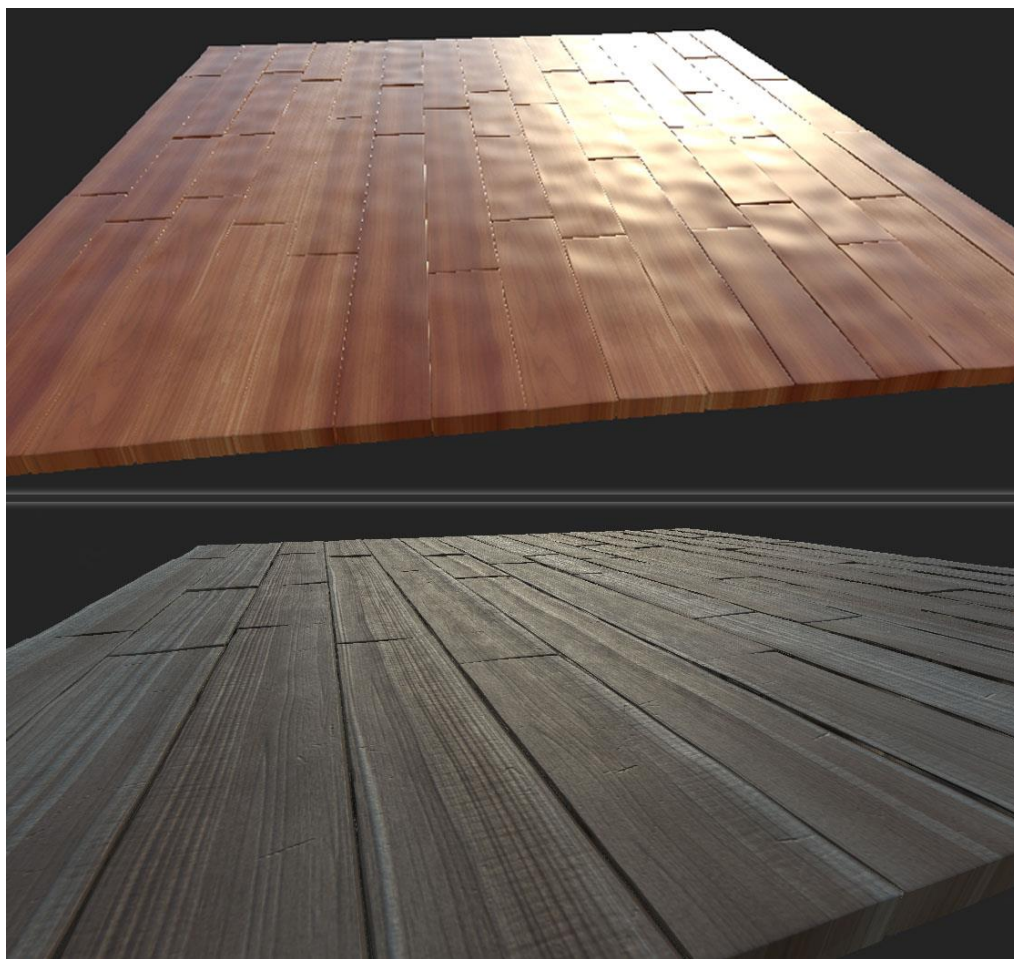
4.9 Substance Painter

3D-mallien valmistuttua, päätettiin niiden teksturointia kokeilla käyttämällä Substance Painter 2-nimistä ohjelmaa. Se soveltui hyvin mallien edistyneempään teksturointiin. Tämäkin kappale on jaettu kahteen osaan, jossa ensimmäisessä käsitellään pintapuolisesti mallien teksturointia, ja toisessa tekstuurien exporttaamista Substance Painter 2:sta. Lopussa myös keskeisimmät havainnot sekä mahdolliset ongelmat, joita tässä osiossa ilmeni.

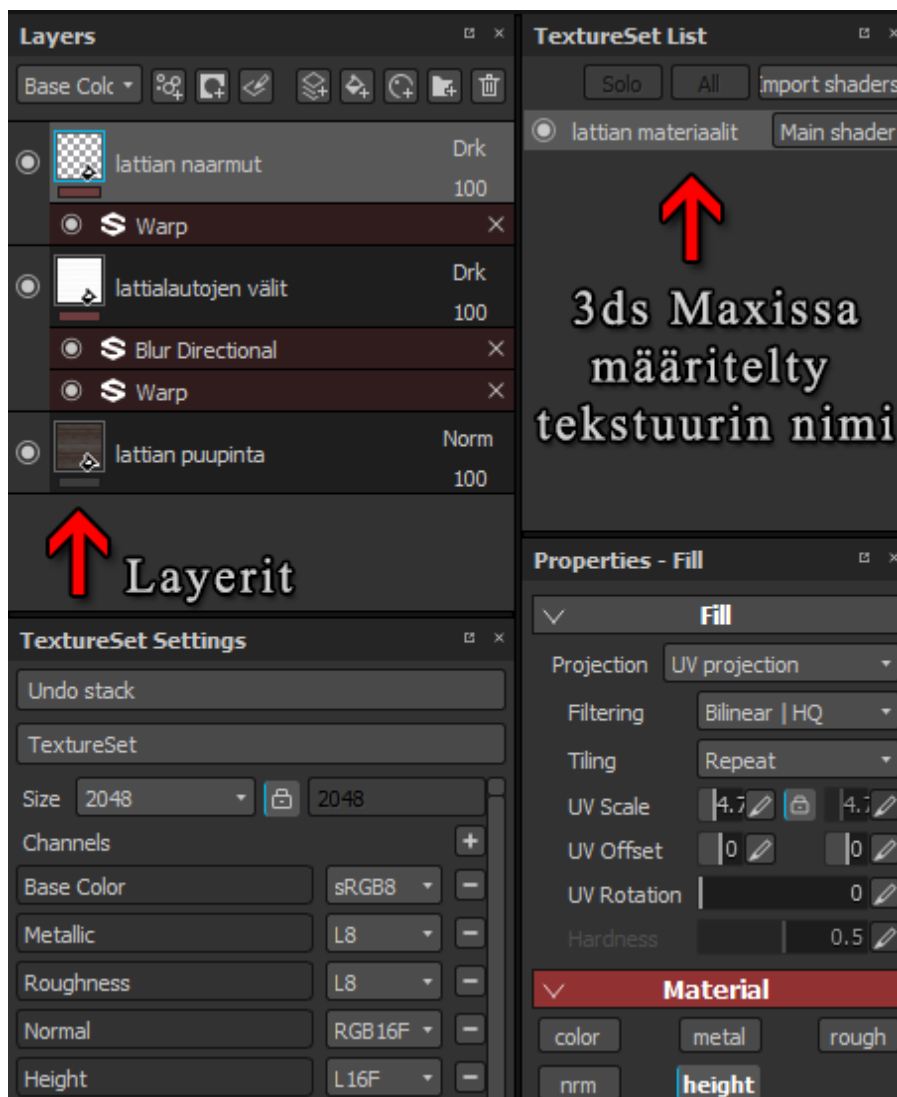
4.9.1 Mallien teksturointi Substance Painter 2:ssa

Ensimmäisenä vuorossa oli mökin lattia, sillä malli oli täysin valmis sekä muodoltaan riittävän yksinkertainen testitarkoitukseen. Mallit näkyvät listattuina Substance Painter 2:n perusnäkyvässä samoilla nimillä, jotka oli tekstuureille 3ds Maxin sisällä määritelty. Listasta valittiin lattia, johon lisättiin ”fill layer”, eli taso jolle voi asettaa koko mallin peittävän materiaalin ikään kuin pohjaksi. Materiaaliksi valittiin tarkoitukseen sopiva puu-materiaali Substance Painter 2:n materiaalikirjastosta. Lattia näytti aluksi tarkoituksiimme aivan liian kiillotetulta sekä kirkkaalta,

joten puu-materiaalin arvoja sekä väritystä muokattiin. Lattiaan lisättiin vielä autenttisuuden vuoksi muutamia naarmuja luomalla toinen fill layer, joka säädettiin siten, että se muuttaa pelkästään korkeutta. Tällöin naarmuille varattu fill layer ei vaikuta muiden tasojen väreihin, kirkkauteen, karheuteen jne. ollenkaan. Lattiaan lisättiin lopuksi vielä kolmas fill layer, johon laitettiin pitkittäiset raidat, jotka linjattiin lautojen väliin mukaan. Tämän layerin asetukset säädettiin siten, että raidoilla olisi oma värinsä sekä ne voisivat vaikuttaa myös korkeuteen. Tällä pystyttiin korostamaan lattialankkujen välejä ikään kuin kovertamalla välejä syvemmäksi sekä tummemmaksi. Lopuksi lisättiin raitojen fill layeriin vielä kaksi ”filteriä” nimeltään: ”warp” ja ”blur”. Näiden avulla pystyttiin hieman vääristämään sekä summentamaan raitojen liian kliinistä ilmettä. Myöhemmin sama warp-filteri lisättiin myös naarmuihin rikkomaan niiden rakennetta hieman.

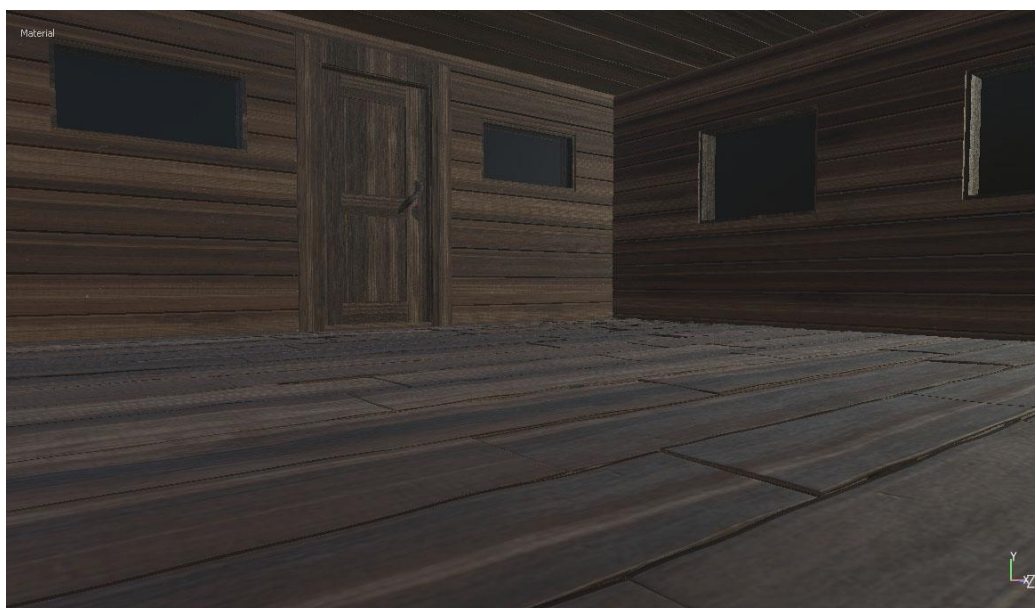


Kuvio 47. Lattia ennen ja jälkeen materiaalin muokkauksen

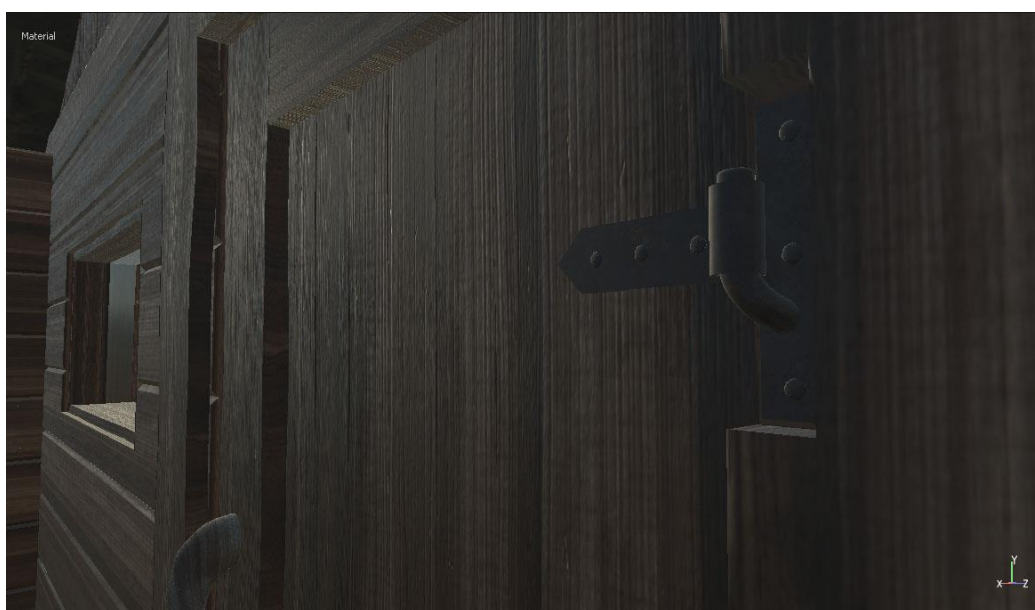


Kuvio 48. Oikealla alareunassa materiaalikartoista valittuna vain korkeus

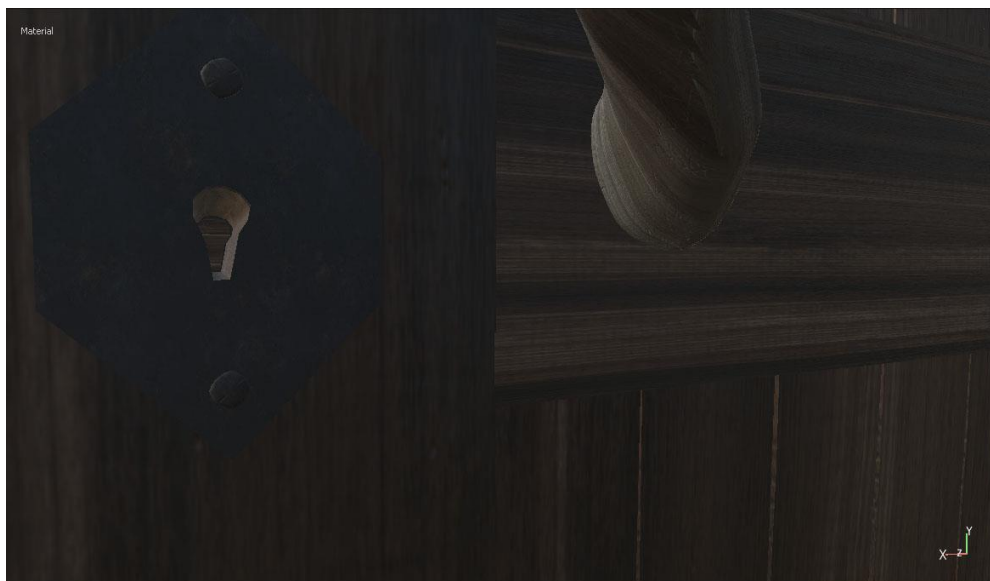
Siirsimme lattian Unityyn ja se näytti sopivan tarkoitukseemme hyvin. Seuraavaksi tehtiin loput mallit valmiiksi, jotka myös teksturoitiin Substance Painter 2:n avulla. Materiaalikirjasto on kohtuullisen kattava, ja esimerkiksi takkaan käytettiin kahta erilaista teräs-tekstuuria, joita muokattiin tarpeen mukaan, saadaksemme haluamme ilmeen. Kuvioissa 49-52 on nähtävissä muutama yksityiskohtaisempi kuva objekteista Substance Painter 2:n perusnäkyssä. Kuvien tarkkuus ei vastaa lopullista tarkkuutta, sillä tekstuuria voi käsitellä pienemmässä koossa Substance Painter 2:n sisällä, tietokeen rasituksen vähentämiseksi. Lopulliset tekstuurit ovat siis pääosin tarkempia kuin kuvissa.



Kuvio 49. Näkymä teksturoidun mökin sisältä



Kuvio 50. Saranat teksturoituna



Kuvio 51. Lukkopesästä näkee mökin sisälle



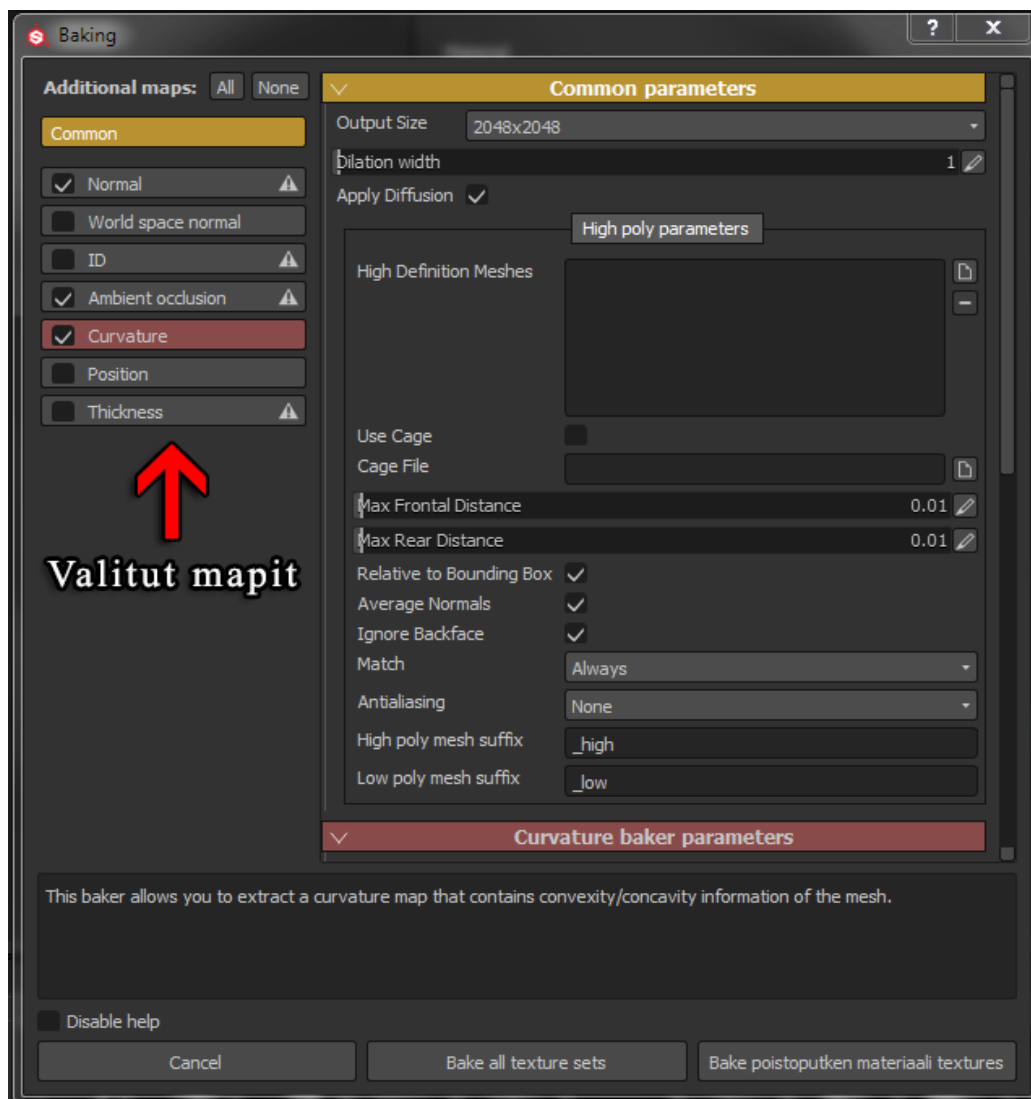
Kuvio 52. Terästekstuurit takassa ja kattoparrujen kiinnikkeissä

Kuten kuvistakin ilmenee, Substance Painter 2:lla hyvinkin yksityiskohtainen teksturointi on mahdollista, joka onkin osasy syy sen käyttöön myös suurissa kaupallisissa projekteissa.

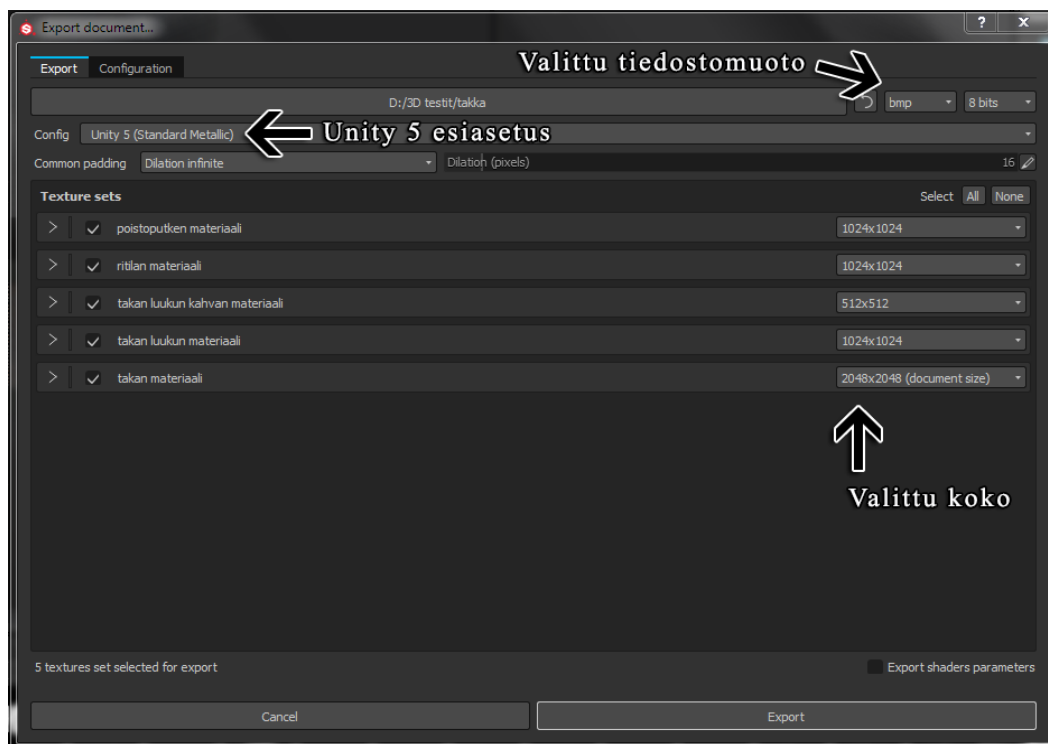
4.9.2 Materiaalien exporttaus

Kun osat on teksturoitu, voidaan materiaalit ”leipoa” (bake) objekteihin, jolloin ohjelma luo niistä tekstuurikartat, joissa on tarvittavat tiedot esimerkiksi väreistä sekä korkeuseroista. Yhden Substance Painter 2:n tekijän tutoriaalin mukaan, pakolliset mapit normaalitilanteessa olisivat: normal, ambient occlusion sekä diffuse. Jotka sisältävät siis arvot väreistä, kulumista sekä varjoista. Tässäkin projektissa käytettiin noita kolmea mappia bake-toiminnossa. Sen jälkeen kun bake on suoritettu, voidaan

tekstuurit exportata halutussa muodossa ulos Substance Painter 2:sta. Tässä käytettiin Substance Painter 2:n tekijän tutoriaalissa suosittelemia arvoja, ja materiaalit exportattiin käyttäen ”Unity 5 (Standard Metallic)” konfiguraatiota/esiasetusta, johon lisättiin ”ambient occlusion” mappi. Exportattavien tekstuurien koon voi itse määrittää väliltä 128x128 - 8192x8192. Tiedostomuotona käytettiin suositeltua 8-bitistä .bmp-tiedostoa.

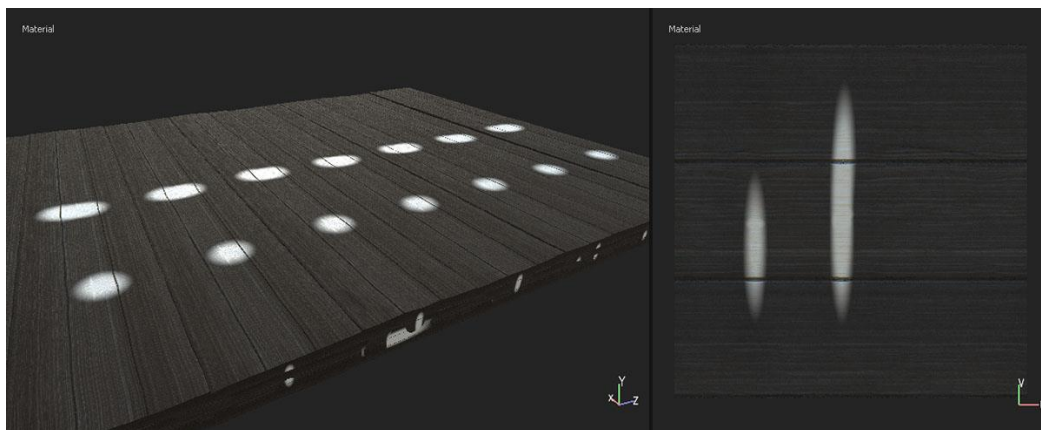


Kuvio 53. Baking-ikkuna Substance Painter 2:ssa



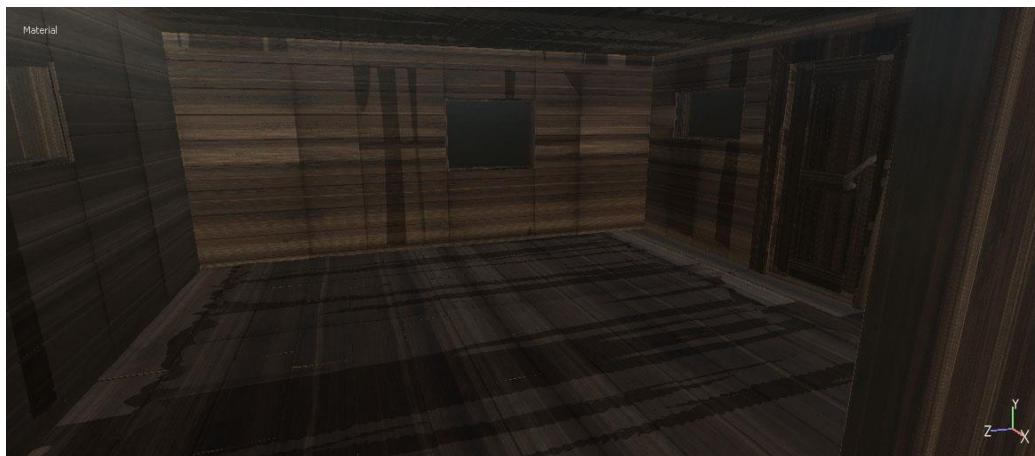
Kuvio 54. Export-ikkuna Substance Painter 2:ssa

Substance Painter 2:lla pystyy tekemään todella ammattimaista jälkeä ja se oli koh- tuullisen helppokäyttöinen muutamien tuntien tutoriaaleihin perehtymisen jälkeen. Suurimmat ongelmat tulivat heti alussa, kun UVW mapin toimintaperiaate oli vä- hän epäselvä. UVW map siis asetetaan 3D-malliin 3ds Maxin sisällä, ja sen tarkoi- tus on määrittää tekstuuri-koordinaatit sekä saumakohta mallissa. Projektin alussa ei siis ollut edes kunnon käsitystä siitä, mikä UVW map edes on. Väärinkäytettynä se rikkoikin tekstuuri-ai-ka tehokkaasti, bake-toimintoa käyttäessä Substance Painter 2:ssa (**Kuvio 56.**). Ihan ensimmäisessä testissä UVW mappia ei ollut lattiaan malliin asetettu ollenkaan, ja kun mallia koitettiin teksturoida, ohjelma oletti ilmeisesti jo- kaisen lautarivin olevan oma mallinsa, joka johti alla olevaan lopputulokseen.



Kuvio 55. Malli ilman UVW mappia

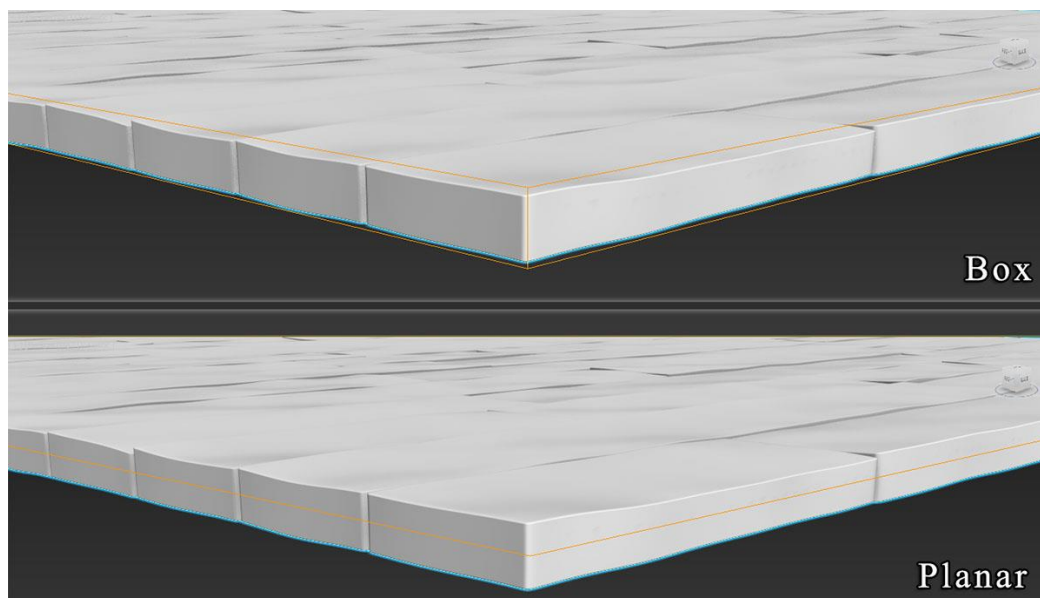
Lattiaan siis yritettiin maalata kahta pistettä siveltimellä, jolloin se monisti pisteet jokaiselle lautariville. Sanomattakin on selvää, ettei ko. ratkaisu olisi pelissä toiminut. Toinen ongelma, joka tuli esille bake-toimintoa käyttäessä, oli virheellisen muotoinen UVW map. Mökin ensimmäiseen versioon oli siis UVW mapit asetettu laatikkomuotoon objektien ympärille siten, että ne rajasivat koko objektin. Tämä toimi sinänsä hyvin, että maalasi mihin tahansa kohtaan seinästä, niin ohjelma ymmärsi sen koordinaatit juuri oikein. Ongelma tuli silti varjoissa, jotka levisivät minne sattuu baken yhteydessä. Alla näkymä sisäänpäin mökin ikkunasta, kun mökkiin oli ennen teksturointia asetettu väärän muotoiset UVW mapit.



Kuvio 56. Varjot ”valuneet” pitkin seinää

Muutaman testin jälkeen päätettiin käyttää ”planar” -muotoisia UVW mappoja objektien ympärillä. Eli mappoja, jotka ”halkaisevat” mallin keskeltä, ja luovat saumakohdan siihen. Tämäkään metodi ei ole täydellinen, sillä maalatessa esimerkiksi seinää, ohjelma kopioi maalin myös toiselle puolelle objektia. Pelissä ei kuitenkaan

käytetty sellaisia tekstuureita, jotta tässä yhteydessä voitaisiin puhua edes oikeasta ongelmasta, mutta täydellinen ratkaisu se ei siltikään ole. Kaupalliseen peliin ongelma tulisi siis ehdottomasti korjata, mutta tässä projektissa ja tämän aikataulun puitteissa, lopputulos on riittävän hyvä.



Kuvio 57. UVW mapin raja oranssilla viivalla

UVW mapin käyttö on nykyään kohtuullisen hyvin hallussa, mutta muutamissa malleissa tässäkin projektissa on puutteellisia kohtia. Hyvänä esimerkkinä (**Kuvio 51.**) näkyvä mutterin yläpuolella oleva metallilevy, jossa tekstuuri ei istu täydellisesti. Kyseessä on kuitenkin vain esteettinen vika, ja kohdat ovat sen verran huomaamattomia, ettei kiireellisen aikataulun vuoksi kaikkia niitä pystytä korjaamaan. Saranoiden UVW mapissa on myös vastaavanlainen vika, ja se korostui vääristyneinä varjoina väärissä paikoissa. Ongelma kuitenkin korjaantui ao-mapin poistamalla, eikä valmiissa mallissa näkynyt mainittavaa eroa. Osa ongelmakohtista on myös pelialueen ulkopuolella, josta niitä ei voi edes pelissä nähdä. Joka tapauksessa, UVW mapin käyttö on yksi projektissa ilmenneistä suuremmista kehitysaluista, johon aiotaan perehtyä myös jatkossa. Kuviossa 58 on vielä kuva lähes valmiiksi teksturoidusta mökistä Substance Painter 2:ssa.



Kuvio 58. Valmis mökki teksturoituna

Ikkunat teksturoimme lopulta Unityn avulla, välttääksemme mahdolliset ongelmat läpinäkyvyyden kanssa. Lopputulos oli tuossa tapauksessa oikeastaan lähes yhtä hyvä kuin Substance Painter 2:sta käyttäessä.

4.10 Adobe Photoshop

Photoshopin käyttö tässä projektissa jäi odotettua vähemmälle. Sitä käytettiin pääsääntöisesti logon luomiseen, sekä tekstiosiossa käytetyn kuvamateriaalin muokkaamiseen. Logoa varten ladattiin dafont.com-sivustolta muutamia erilaisia fontteja, joita sai ilmaiseksi käyttää ei-kaupalliseen toimintaan. Pelin nimi oli tuolloin vielä keksimättä, ja lyhyen mietinnän jälkeen nimeksi tuli ”Minera”. Sana on latinaa ja tarkoittaa kaivosta/louhosta tms. Panoraamakuva, joka pelin taustanamme toimii, on otettu vanhalla louhoksella, joten nimi oli looginen sekä helppo, ja olimme molemmat siihen tyytyväisiä. Päädyimme latinankieliseen nimeen oikeastaan vain siksi, että käytimme myös edellisessä peliprojektissamme latinankielistä nimeä, jolloin pelin nimi oli ”Tempus”.

Kun nimi oli päätetty, alkoi fonttien testailu. Mahdollisista muokkauksista tiedettiin vain se, että haluaisimme käyttää panoraamakuva jossain määrin myös logosamme. Lopulta päädyttiinkin luomaan tekstin taustalle ”layer maskin”, johon pystyi lisäämään mustavalkoisen kuvan tehosteeksi. Panoraamakuva laitettiin taustalle,

se sijoiteltiin paikalleen sekä tehtiin tarvittavat muutokset fonttiin. Lopulta logoja oli kahdeksan erilaista pienillä muutoksilla sekä kahdella eri fontilla. Päädyimme käyttämään hieman massiivisempaa logoa, sillä se vaikutti toimivan menuvideosamme hyvin sekä toi paremmin esille taustalla käytettyä panoraamakuvaa.



Kuvio 59. Alla logo jota käytimme lopulta pelissä

Tiedostomuotona käytimme .png-kuvia, sillä ne ovat häviöttömiä, eikä niiden laatu kärsi muokatessa ja tallentaessa kuten .jpg:ssä Photoshopin kanssa ei ollut oikeastaan mitään ongelmia.

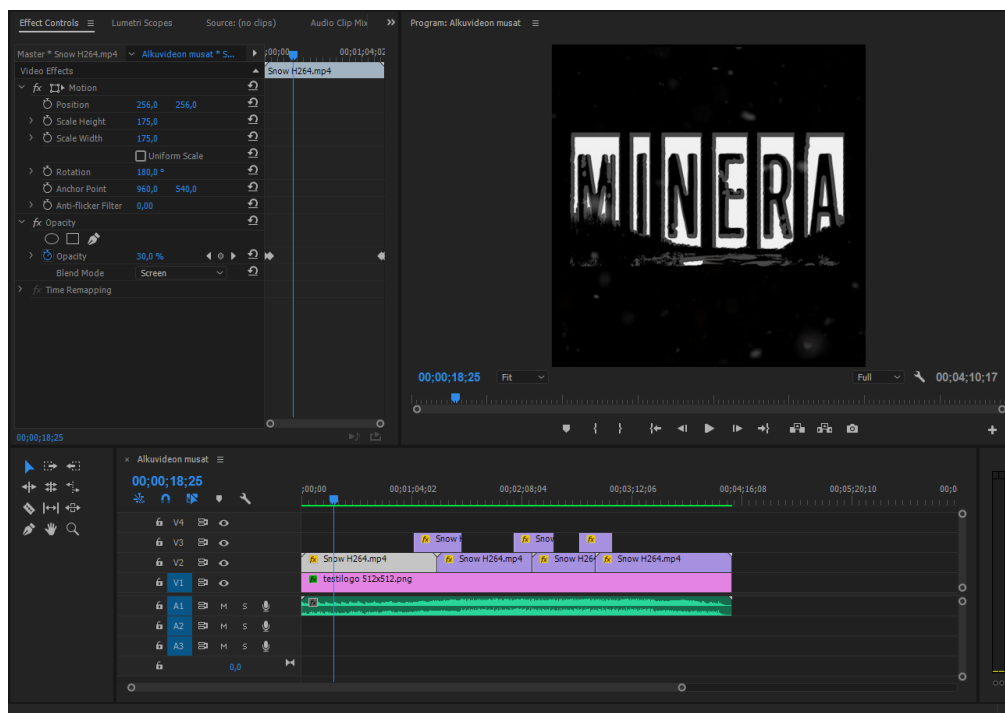
4.11 Adobe Premiere Pro

Premiereä käytimme pelissä menuvideon, alkuvideon sekä loppuvideon luomiseen. Alkuvideo on oikeastaan koostettu vain äänistä, joiden avulla ”maalataan” pelaajalle kuva lähtötilanteesta. Eli pelin alussa kuullaan, kuinka pelaajan hahmo menee mökkiin, tekee siellä muutamia asioita, kuten sytyttää takan ja menee nukkumaan. Ruutu on tässä vaiheessa vielä musta, lukuun ottamatta tekstiä, jossa kerrotaan pelaajalle kellonaika. Hetken kuluttua jotain tapahtuu, pelaajan hahmo herää ja peli alkaa. Loppuvideossa on vain musiikkia sekä tekstit; joissa rullaa tekijöiden nimet sekä lista henkilöitä, joiden materiaalia olemme pelissä käyttäneet ja joita haluamme kiittää.

Menuvideo oli ensimmäinen video, joka peliä varten tehtiin. Halusimme pitää sen kohtuullisen askeettisena käyttäen logoamme jollain tehosteella/tyylillä, jossa voisimme mahdollisesti hyödyntää VR-laitteistoa. Logo laitettiin videon taustalle ja päätimme kokeilla miten lumisade toimisi lisäefektinä, sillä se sopi hyvin pelin sekä freemusicarchive.org -sivustolta löytyneen valitun kappaleen tyyliin. Youtuben kautta hakemalla sanoilla ”snow overlay”, löytyi linkki sivustolle, jossa yksityishenkilö jakoi ilmaisia hyvälaatuisia videoita sekä kuvia. Sieltä löytyi kaksi videota, jotka näyttivät sopivan tarkoituksiimme. Lumisade-video laitettiin logomme päälle Premieren aikajanaan ja sen ”sekoitustilaksi” (blend mode) asetettiin arvo: ”screen”. Jolloin siis hiutaleet näkyvät videolla ikään kuin logomme päällä, mutta lumisade-videon musta tausta häviää sitä peittämästä. Visuaalisen materiaali monistettiin koko kappaleen pituiseksi, tarvittavat häivytykset sekä efektit lisättiin, ja menuvideo oli valmis.

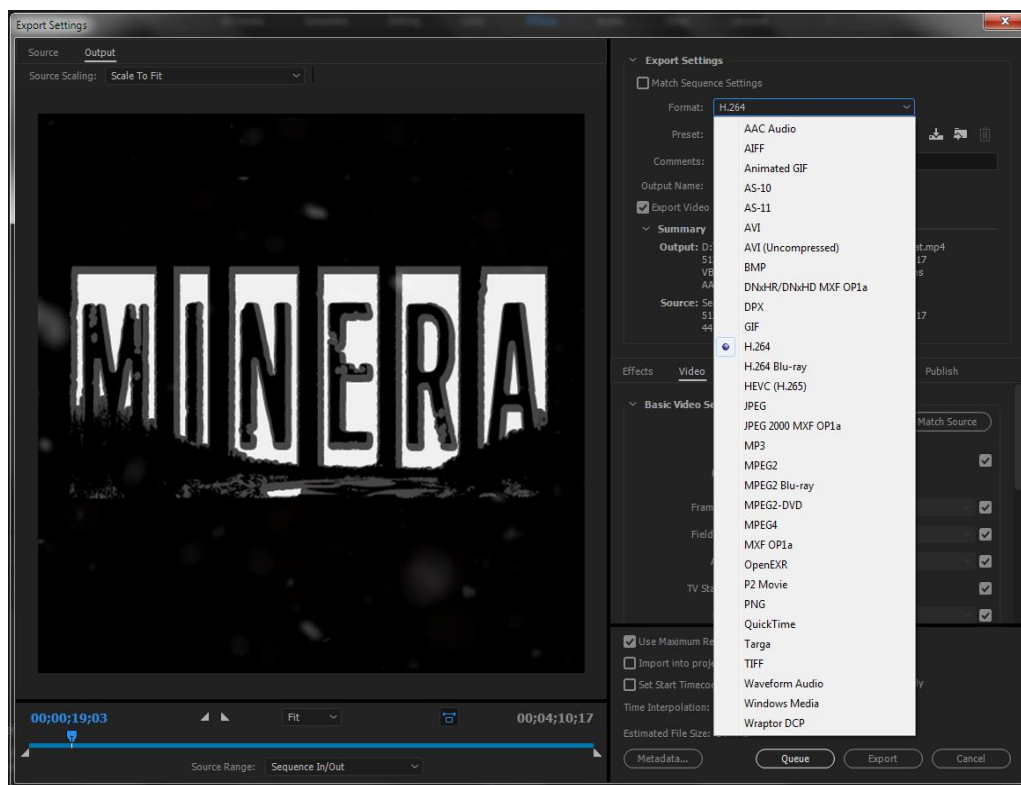
Tämän jälkeen video exportattiin Premierestä resoluutiolla 512x512, jonka olimme todenneet testien perusteella toimivaksi. Exportin valmistuttua videosta poistettiin logo, jolloin jäljelle jäi vain lumisade. Siitä exportattiin oma versio, jonka jälkeen vielä käännettiin lumisade peilikuvaksi, josta exportattiin kolmas video. Kolme videota siksi, että tarkoituksena oli käyttää niitä menussa, joka oli rakennettu kuutioon. Etummaisessa seinässä pyörisi video logolla, takaseinällä pyörisi sama video ilman logoa ja vasemmalla sekä oikealla pyörisi video, jossa lumisade oli peilattu

alkuperäisestä. Tällöin yhdessä vierekkäisessä ruudussa, jotka pelaaja voisi nähdä yhtäaikaaisesti, ei pyörisi täysin samaa lumisadetta. Päätimme vielä lopulta käyttää myös päällimmäisessä ruudussa suoraan kameraa kohti satavaa lumisadetta, joka oli ladattu samalta sivustolta. Alimmainen ruutu jätettäisiin tyhjäksi/mustaksi.



Kuvio 60. Premieren perusnäkö menuvideon luonnissa

Premiere tarjoaa todella kattavan valikoiman eri formaatteja sekä muokkausvaihtoehtoja exportin yhteydessä. Päädyimme käyttämään H.264 formaattia, sillä sen käyttöä suositeltiin, eikä mikään yhteensopivuusongelmia eteen tullutkaan. Tiedostomuoto videoissa on .mp4, joka sekin toimi moitteettomasti. Yhdessä vaiheessa tehtiin testiksi myös alle sekunnin mittaisen kuolinanimaation hirviöllemme videosta, jossa oli ulospäin ”räjähtävä” mustavalkoinen shokkiaalto. Sen tiedostomuoto oli .mov, joka oli myös yhteensopiva Unityn kanssa.

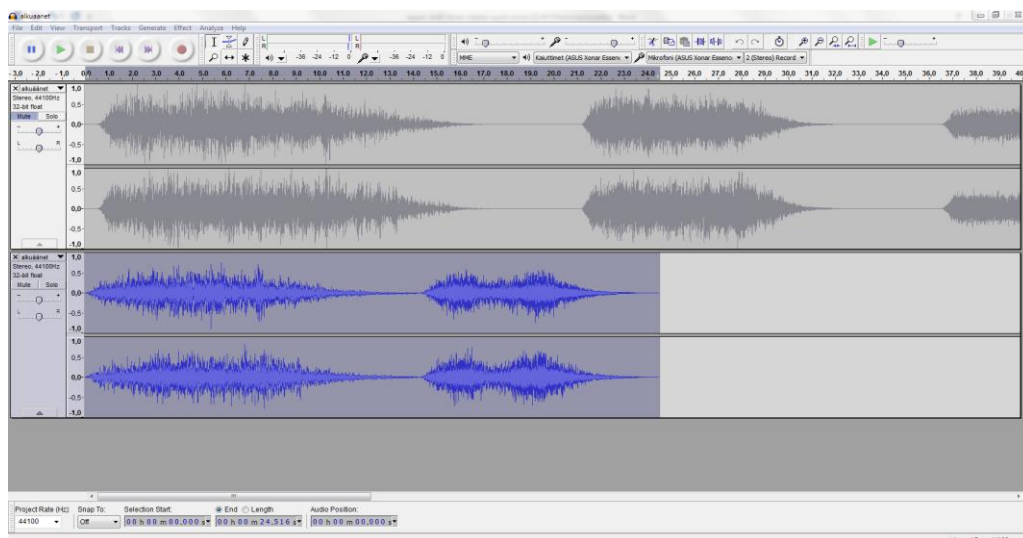


Kuvio 61. Vaihtoehdot eri formaateista exportin yhteydessä

Videoiden kanssa suurimmat ongelmat liittyivät resoluutioon. Emme aluksi tiedeneet, mikä olisi kuution yhden seinän resoluutio, jonka perusteella video tulisi tehdä. Ensimmäisen videon resoluutio olikin 2048x2048, joka näytti oletettavasti hyvältä tietokoneelta. Se myös skaalautui kohtuullisen hyvin menuvideon kuutioon, mutta viisi tuon kokoista videota pyörimässä yhtäaikaaisesti aiheutti todella pahaan pätkimistä. Päätimme siis kokeilla reilusti pienempää videota, jonka resoluutioksi lopulta tulikin 512x512. Se skaalautui vielä huomattavasti paremmin kuutioon, eikä aiheuttanut minkäänlaista pätkimistä, vaikka videoita pyörikin yhtäaikaaisesti viisi kappaletta.

4.12 Audacity

Audacityn käyttö oli tässä projektissa kohtuullisen vähäistä. Sen avulla muokattiin muutamia pelissä käytettyjä efektejä, jotka oli osittain ladattu freesound.org-sivustolta, ja osittain nauhoitettu itse M-Audio Microtrack II -audiotallentimella. Muokkaukset olivat lähinnä kohinanpoistoa, äänenvoimakkuuden tai nopeuden muuttamista, kaiun lisäämistä tai poistamista sekä efektien pätkimistä pienempiin osiin.



Kuvio 62. Aloitusäänen muokkausta

Suurimmat ongelmat äänien kanssa johtuivat freesound.org-sivustolta ladattujen äänien vaihtelevasta laadusta sekä audiotallentimen herkkyydestä. Freesound.org on sivusto, johon käyttäjät voivat tuoda ääniä ilmaiseksi muille ladattavaksi. Koska sivusto sekä materiaali ovat ilmaisia, sen laatu myös vaihtelee todella paljon. Jatkuvasti hyvää materiaalia etsiessä eteen tulee äänitteitä, joiden taustakohina/melu on lähes yhtä voimakasta kuin itse ääni, jota sieltä haetaan. Laadun vaihtelevuus näkyy toki myös äänittäjästä, ja onkin selvää, että osa äänitteistä on tehty vähän vähemmän tosissaan.

Audiotallentimen kanssa itse tallentaessa jaksaa ko. laitteen herkkyys yllättää kerta toisena jälkeen. Tätä projektia varten äänitettiin esimerkiksi muutaman tulitikun raapaisun äänet, ja samassa tilassa jossa äänitys tapahtui, oli seinäkello sekä toisessa kerroksessa oven takana puhuttiin puhelimeen. Kellon nakutus kuului taustalta vähän turhankin selvästi valmiissa äänitteessä, ja jos tarpeeksi laadukkailla kuulokkeilla äänitettä kuuntelee, pystyy siitä myös tunnistamaan kauempaa tulevan puheen. Laite on siis todella herkkä, ja sen huomaa niin hyvässä, kuin pahassakin. Saimme silti tallennettua hyvälaatuisia äänitteitä, joista osa tuli käyttöömmme pienen editoinnin jälkeen.



Kuvio 63. Microtrack II Audiotallennin



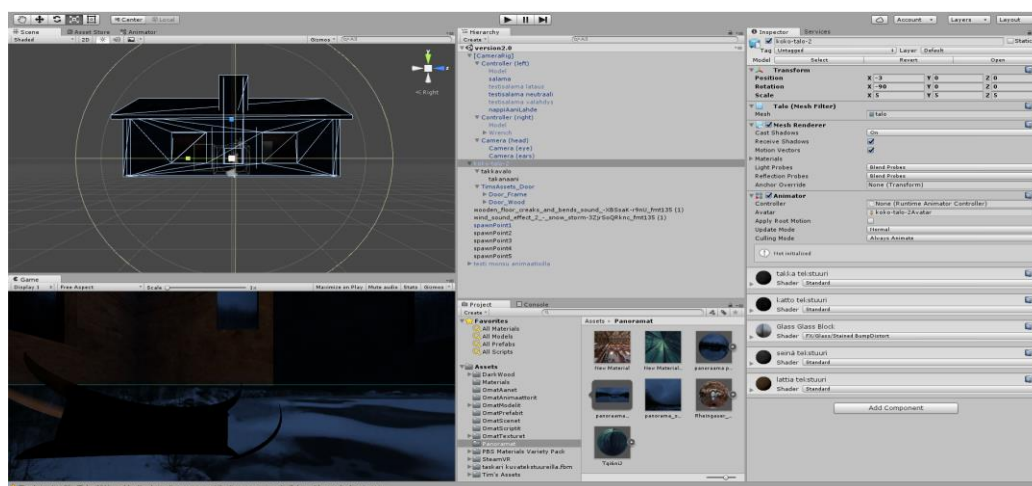
Kuvio 64. Audiotallentimen liitännät

Microtrack II tallentimen käyttö on todella intuitiivista, siinä on kohtuullisen kattavat liitännät sekä irrotettava mikrofoni.

4.13 Lopullisen pelin luominen ja digitaalisen materiaalin yhdistäminen

Tässä luvussa käymme läpi lopullisen pelin luonnin, digitaalisten resurssien tuomisen peliin ja aiemman prototyypin hyödyntämisen lopullisen pelin luomisessa.

Vanhoja materiaaleja oli helppo hyödyntää lopullisessa peliversiossa, sillä molemmat olivat samassa projektiympäristössä. Käytimme uuden pelin luomiseen vain uutta näyttämöä, jonne siirsimme kaikki prototyypin testaamisessa hyväksi ja hyödylliseksi todetut resurssit. Huonoiksi todetut resurssit ja skriptat poistimme selkeyden vuoksi. Rakennettuun versioon sisällytimme vain lopullisessa pelissä käyttämämme näyttämöt.



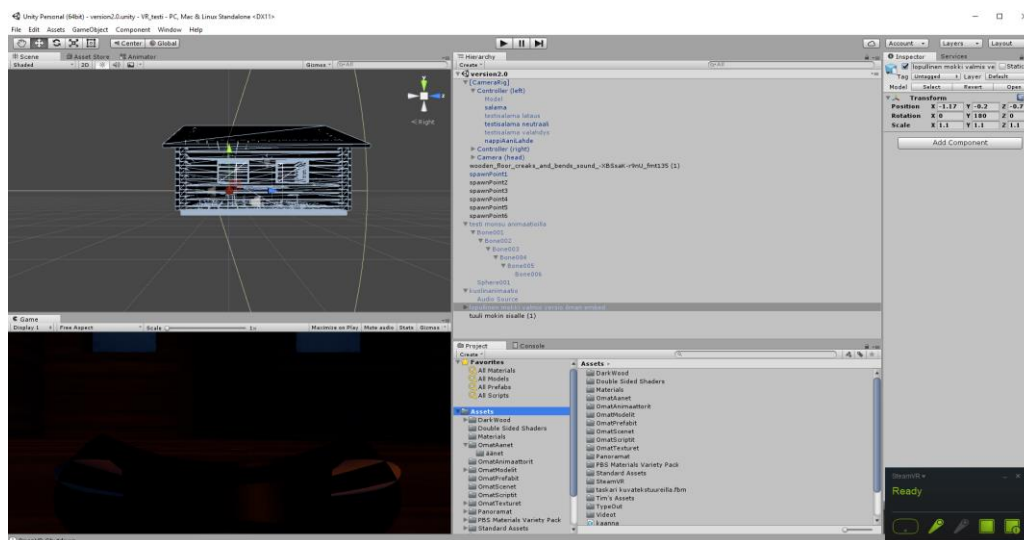
Kuvio 65. Lopullinen pelialue vanhoilla materiaaleilla

Jotkut vanhoista resursseista vaativat hiomista, kuten esimerkiksi aiemmat toimimattomat 3D-mallit. Nämä loimme uudestaan vanhojen pohjalta, jonka lisäksi otimme huomioon testaamisessa huomatuksi ongelmat. Jotkin vanhat resurssit oli tarkoituksella tehty hieman heikompileatuisiksi testaamisen nopeuttamiseksi ja ajan säästämiseksi. Hyödynsimme kuitenkin näissäkin vanhoja malleja luodessamme uusia hiotumpia ratkaisuja.

Käytimme äänimaailman luomiseen netistä löytämiämme satunnaisia ääniä ja videoita testatessamme satunnaisia videoita. Nämä kuitenkin korvattiin lopulliseen versioon joko omilla tai tekijänoikeusvapailta resursseilla.

Skybox-materiaalia testattiin prototyypissä netistä löydettyillä panoraamatoyksilla. Lopulliseen versioon rakensimme oman panoraamatekstuurin itseottamistamme kuvista rakennetulla panoraamakuvalla.

Joitakin malleja luotiin kokonaan tyhjästä pöydältä, esimerkiksi toiseen ohjaimeen liitetty taskulampun malli korvattiin uudella futuristisella Viven navigaatioasuvaa muistuttavalla salamavalon 3D-mallilla. Uusien resurssien luonti onnistui paremmin kuin prototyyppiä työstettäessä, sillä prototyypin myötä karttuneesta kehityksestä ja kokemuksesta johtuen työkalut ja -välineet löytyivät helpommin ja tiedettiin mitä kannattaa välttää ja mitä tehdä toisin.



Kuvio 66. Lopullinen pelinäyttämä uusilla resursseilla

Lopullisen version luominen oli huomattavasti helpompaa kaikin puolin, sillä enää ei tarvinnut opetella uudestaan käyttämään työvälineitä, joita oli jo prototyypin aikana harjoiteltu. Suurin osa lopullisen version tarvittavista resursseista oli tehty jo prototyyppiin, ja uudet työkalut oli helppo opetella, kun alustan toiminta oli jo ennestään tuttua.

4.14 Beetatestausta

Testautimme peliä kahdella kaverillamme. Vaikka mitään suuria muutoksia emme voineet tai aikoneet peliin enää loppumetreillä tehdä, testauksen idea oli saada mahdollisia kehitysideoita tulevaisuutta varten, sekä säätää pelin vaikeustaso sopivaksi. Olimme toki ensin itse testanneet peliä ja pyrkineet ajoittamaan pelin hirviöiden

ilmestymisen siten, että peli olisi sopivan vaikea sitä ensimmäistä kertaa koittavalle. Laadukasta dataa on kuitenkin todella vaikea saada, jos tietää tarkalleen, miten peli on rakennettu ja miten se toimii.

Emme myöskään halunneet uusia tilannetta, joka tapahtui viime peliprojektin kanssa vuonna 2014. Peliä demotessa kävi aika pian ilmi, että siinä oli selkeä epäkohta, joka johti siihen, ettei kukaan onnistunut peliä täysin läpäisemään. Pidimme itse pelin loppuratkaisua aika itsestäänselvyytenä, mutta todellisuudessa pelaajat olisivat tarvinneet vielä selkeämmän indikaattorin siitä, mitä pelaajan tulisi lopussa tehdä saavuttaakseen ns. ”hyvän lopun”. Vaikka nykyisestäkään pelistä ei ole tarkoitus tehdä liian helppoa, ei sen tulisi olla myöskään liian vaikea. Tästä syystä pidimmekin jonkinlaista testiä pakollisena ennen pelin lopullista ”buildausta”.

”Pätevä VR-peli ottaen huomioon realiteetit ja kehitysjajan. Tiukan haastava hyvällä tapaa. Simppeli ja toimiva. Paras asia: kamiinan elävä liekki. Tulevaisuudessa: monipuolisuutta, mitä ikinä se onkin” –Niko Leppänen

”Kokonaisuutena loistavasti toteutettu. Pelin tekemiseen oli selvästi käytetty aikaa, eikä tehty vähän sinnepäin. Pelissä olevat musiikit ja ”tunnelmallinen” jännitys oli loistavasti toteutettu. Myös VR-lasien käyttö peliä tehtäessä on onnistunut. Ensimmäistä kertaa kokeilin VR-laseille tehtyä peliä, eikä ollut pettymys ollenkaan. Isoa plussaa toi myös se, kun laittoi VR lasit päähän ja tunsu todella olevansa vanhassa mökissä ja mökin luontiin oli selvästi käytetty aikaa. Suosittelisin muitakin testamaan, 5/5.” –Pasi Ojala

”Pelin idea nopeasti ymmärrettävissä, pelaaminen onnistuisi varmasti ilman ohjeitakin. Äänitehosteita käytetty hyvin ja hirviöiden yksinkertaisesta ja tutusta ulkomuodosta plussaa. Kokonaisuutena hyvä alku kehittää peliä pidemmällekin.” –Johanna Latva-Valden

”Peli oli tunnelmaltaan vaikuttava, ja VR toimi loistavasti. Tykkäsin ja suosittelen.” –Hanna Ekblad

”Pelin idea oli helppo ymmärtää ja toteutus oli hyvin onnistunut.” –Tuomas Lammi

”Vastasi täysin odotuksia, ja osittain jopa ylitti ne. 5/5” –Jussi Loukiainen

4.15 Muita mahdollisia käyttötarkoituksia panoraamatekniikalle

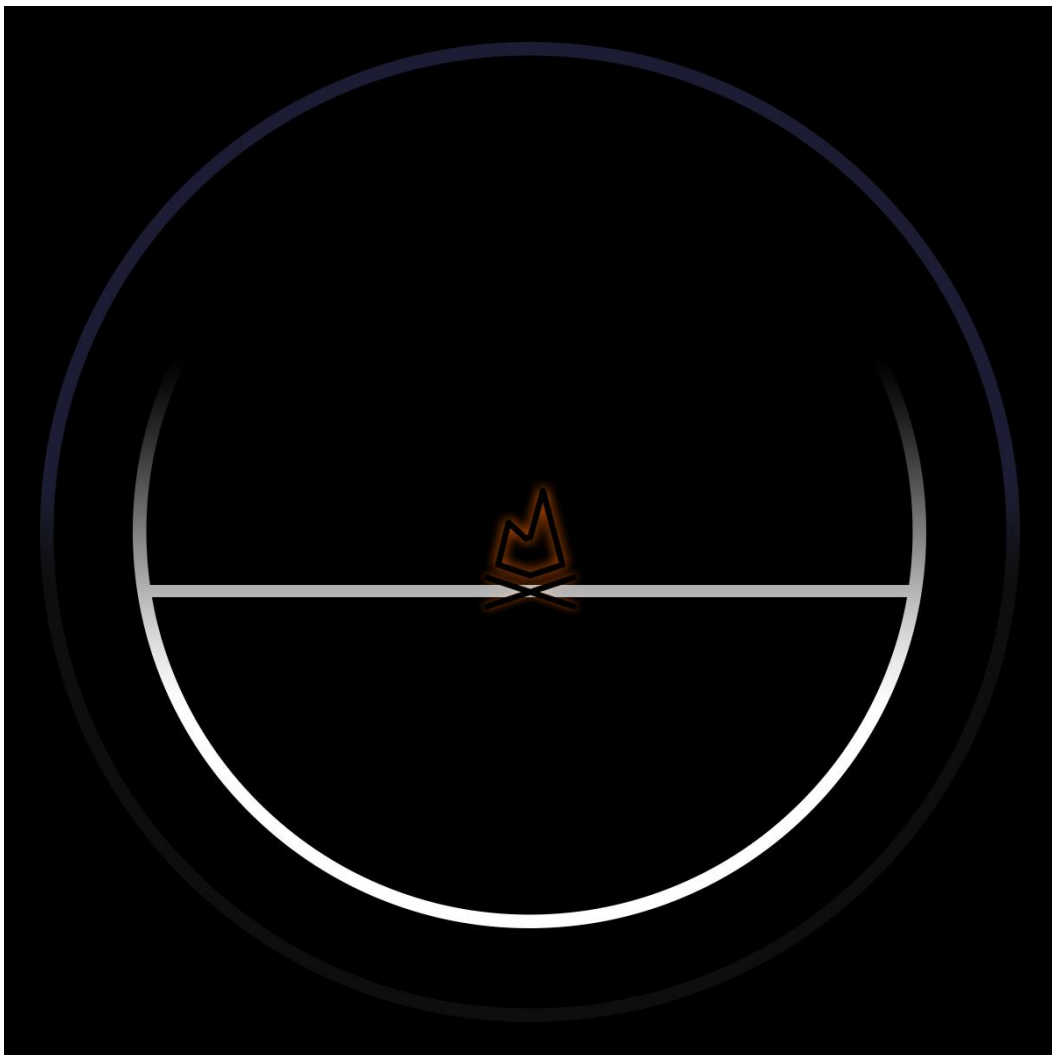
Koitimme projektin loppupuolella mieltä myös muita mahdollisia käyttötarkoituksia käyttämällemme tekniikalle. Loppujen lopuksi kyseisen tekniikan käyttö peleissä on kohtuullisen rajoittunutta muutamista syistä. Ensinnäkin horisonttinäkymän luominen panoraamakuvasta ei ole koskaan täysin saumaton. Joissakin tilanteissa saumakohdat ovat silti niin huomaamattomia, ettei se välttämättä häiritse kokemusta. Jos puhutaan kuitenkin pelillisestä käytöstä, täytyy huomioida immersion tärkeys lopullisessa tuotteessa. Kun horisonttinäkymä tehdään panoraamakuvasta, voidaan sillä osittain ”huijata” pelaajaa, mutta kyseessä on kuitenkin yksiulotteinen kuva, josta pallopanoraama on koostettu. Se muodostuu ongelmaksi silloin, jos siihen pitäisi esimerkiksi pystyä heijastamaan valoa ja/tai varjoja. Toinen ongelma, jonka kohtasimme täysin avoimessa skybox-näkymässä, oli ”leijumisen tunne”. Peli näytti siis siltä, että pelaaja olisi hyvin pienikokoinen ja leijuisi muutaman metrin korkeudella maasta. Tämä johtuu siitä, että skyboxin on tarkoituskin toimia horisonttina, jolloin näkymä on juuri tuon kaltainen.

Tulimmekin siis siihen lopputulokseen, että jonkinlainen ainakin osittain suljettu tila skyboxin sisällä toimii pelikäytössä parhaiten. Yksi idea, jota pyörittelimme ennen projektin varsinaista aloittamista, oli pelialueen luominen kahdesta sisäkkäisestä erikokoisesta pallosta. Ulommainen suurempi pallo (skybox) olisi toiminut horisonttinäkymänä ja sisempi pienempi pallo tavallaan rikkoisi rakennetta estäen osittain näköyhteyden ulompaan palloon ja sisältäisi kuvadataa, jossa olisi pelin teemaan liittyviä objekteja. Pelin alusta/maa luotaisiin erillisellä 3D-objektilla, jolla rikottaisiin näkyvyys skyboxin alaosaan ja poistettaisiin tunne leijumisesta.

Pelialueena ko. ideassa olisi toiminut öinen metsäaukea, jossa pelaaja istuisi nuotilla. Käytettävänä objektina pelissä olisi ollut jokin lyhty tms. jonka valo ei kuitenkaan ylittäisi sisempää palloa pidemmälle. Sisemmän pallon reuna koostuisi puskista, kivistä jne. ja se olisi yläosastaan läpinäkyvä. Ulommassa pallossa näkyisi puuraja sekä taivas. Tarkoituksena oli testata myös ”parallax scrolling” -tekniikan hyödyntämistä, jolloin siis sisemmälle pallolle olisi määritetty nopeampi arvo kuin

ulommalle pallolle kameraa liikuttaessa. Tällä saadaan aikaan keinotekoinen syvyys efekti, koska sisemmän pallon objektit liikkuvat eri nopeudella taustaan nähden. Näitä tekniikoita hyödyntämällä pitäisi teoriassa olla mahdollista luoda yksinkertainen peli jonkinlaisella syvyys efektilä. Mikäli se toimii kuten oletettu, pystyisi sillä oikomaan todella paljon taustan sekä pelaajaa ympäröivien objektien osalta. Toki lopputuloksena olisi kohtuullisen yksinkertainen peli, mutta mikäli pelissä haettaisiin tarkoituksellisesti vaikkapa ”low poly” -tyyliä, voisi se teoriassa toimiakin.

Kuviossa 67 vielä sivusta kuvattu poikkileikkaus tuosta suunnitelmasta, jota yhdessä vaiheessa pohdimme. Siinä kaksi sisäkkäistä palloa muodostaa pelialueen sisemmän ja ulomman reunan. Sisempi pallo on yläosastaan läpinäkyvä ja sen halaisee 3D-mallinnettu maa. Keskellä aluetta on nuotio, jonka vieressä pelaaja olisi.



Kuvio 67. Yksinkertaisen pelialueen rakenne

Toinen tapa, jossa tätä panoraamatekniikkaa voisi hyödyntää, olisi Vaasan ammattikorkeakoulussa eräällä kurssilla esille tullut maailmanperintö -projekti. Kyseisessä projektissa oli ilmeisesti tarkoitus tehdä audiovisuaalista materiaalia, jonka avulla luotaisiin tietopaketteja luonnonperintökohteista. Tällä tekniikalla voisi kohutuullisen hyvin sekä helposti luoda mielenkiintoista sisältöä tuollaiseen projektiin ja se voisi toimia erilaisena visuaalisena tapana esitellä joitain kohteita. Tuollaisessa käytössä immersio olisi myös toisarvoinen seikka, jota ei juurikaan tarvitsi huomioida.

5 YHTEENVETO

Tässä luvussa käymme läpi seikkaperäisesti projektin tulokset ja saavutukset, kuinka paljon projektin edetessä tulokset muuttuivat suunnitellusta ja miten teoria toimi projektin loppuunsaattamisen tukena.

5.1 Joni Hänninen; päätelmät ja lopputulosten analysointi

Projekti eteni erittäin hyvällä tahdilla projektin laajuuteen ja lopputulokseen suhteutettuna, lisäksi toimitusviiveistä selviydyttiin hyvin, vaikka projektin alku venyi useita kuukausia laitteistojen toimitusongelmien vuoksi. Opimme myös käyttämään hyvin meille entuudestaan tuntemattomia ohjelmistoja ja ymmärsimme miten ne toimivat. Opimme myös peliteollisuudesta, sen toimintatavoista ja tiimirakenteesta. Etenkin tiimin jäsenten roolit korostuivat projektissamme; tiimissä on hyvä olla paljon erilaisia ihmisiä, jotta saadaan uusia näkökulmia ja ajatuksia, jonka myötä saadaan entistä hiotumpi lopputulos.

Todistimme myös lopullisessa pelissämme, että panoraamakuvia voidaan hyödyntää myös peliteollisuudessa jonkun verran. Käyttömahdollisuuksia on lähes nyt hyvin rajallinen määrä, mutta tulevaisuudessa uskon niiden lisääntyvän. Vain mielikuvitus on rajana.

Lopputulos oli hyvä työn laajuuteen ja uusien ohjelmistojen käyttöön suhteutettuna. Opin myös paljon lisää ohjelmoinnista ja käytetyistä alustoista.

Valitettavasti emme päässeet koskaan hyödyntämään YLE:n tiloja ja heidän laitteistoaan ja tietotaitoaan, sillä tilojen remontointi ja vuokrasopimuksen kirjoitus venyi liian pitkälle. Myös sidosryhmien apu jäi valitettavan pieneksi aikataulujen riskiin menemisen vuoksi. Projektia hankaloitti myös laitteiston siirtely paikasta toiseen, sillä käytössämme oli ainut HTC Vive -laitteisto, jota hyödynnettiin Vaasan peleihin liittyvissä tapahtumissa.

Muutoksia projektin kulkuun toi erityisesti aikataulu, jonka vuoksi muutimme pelin juonta merkittävästi. Näin kuitenkin pystyimme pitämään projektin paremmin tavoiteaikataulussa, vaikka se hieman paisuikin loppua kohden. Loppujen lopuksi saimme kuitenkin projektin valmiiksi ja lopputulos oli molempia miellyttävä.

Projektissa työskentely toi paljon näkökulmaa pelintekemiseen, etenkin siihen, miten paljon tulee ottaa huomioon jo pelkässä suunnitteluvaiheessa. Huono suunnittelu voi romauttaa koko projektin kasaan, jos se esimerkiksi paisuu liian suureksi tai on liian monimutkainen. Kun pysytään yksinkertaisessa linjassa ja pyritään tekemään sillä mahdollisimman paljon, saadaan hyvä lopputulos.

5.2 Janne Harju; päätelmät ja lopputulosten analysointi

Projekti oli lopulta aika onnistunut kokonaisuus lukuisista vastoinkäymisistä huolimatta. Roolijako oli alusta asti selkeä, ja kuten edellisessäkin peliprojektissa, se toimi hyvin. Saimme molemmat keskittyä omiin alueisiimme ja oppiminen tuntui olevan jatkuvaa koko projektin ajan.

Valitettavaa on, ettei alkuperäinen visio tämän aikataulun puitteissa koskaan toteutunutkaan, mutta en ole myöskään varsinaisesti pettynyt nykyiseen lopputulokseen. Saimme kuitenkin koottua toimivan kokonaisuuden ja vaikkakin sisältö jäi pienemmäksi kuin olimme alun perin ajatelleet, oli se monilta osin hyvin hiottua. Panoraamakuvausta olisin pelissä hyödyntänyt mielelläni enemmänkin, sillä nyt sen osuus jäi hieman odotettua pienemmäksi.

Projektin aloitusta viivästyttivät toimitusviiveet ja kysymykset paikasta, johon laitteisto tulitaisiin kasaamaan. Emme ikävä kyllä päässeet vielä tämän projektin aikana Ylen tiloihin, mutta pärjäsimme toisaalta mainiosti nykyisissäkin tiloissa. Käytettävä laitteisto oli aluksi myös pitkään osaltamme täysi mysteeri, mutta onneksemme saimme toimeksiantajaksi Jussin, jolla oli tähän ratkaisu ja resurssit. Hänelle kuuluu siis erityiskiitos koko projektin mahdollistamisesta.

Mitä itse käytännön osuuteen tulee, niin uusia ohjelmia tuli odotettua enemmän ja sitä mukaa myös erilaisia pulmatilanteita eri ohjelmien välillä. Olenkin monta kertaa projektin aikana miettinyt, että jos nykyisellä tietotasolla olisi lähtenyt tätä alun

perin tekemään, aikataulusta olisi voinut leikata kolmasosan pois. Valtaosa ajasta on kuitenkin kulunut asioiden uudelleen tekemiseen ja jatkuvaan selvittämiseen sekä testailuun projektin edetessä.

Ihan loppusuoralla piti toki kokea vielä muutamat yllätykset laitteiston osalta. Tietokoneestani hajosi näytönohjain viimeisellä viikolla ja samaan aikaan kiintolevy ilmoitti, että elinikää on ~1% jäljellä. Onneksi varalle löytyi toinen näytönohjain, ja kiintolevykin jaksoi sinnitellä projektin loppuun asti. Olen siis valtaosan kaikesta materiaalistani luonut kotona omalla koneellani, joten tuo tapahtuma ei tullut ihan parhaaseen väliin.

Yhtä kaikki, onneksi vältyimme tällä kertaa ns. ”lumipalloepektiltä”, tajusimme tarpeeksi ajoissa resurssien sekä aikataulun rajat ja osasimme suhteuttaa toimintamme niihin. Peliprojektien työläys jaksaa kyllä yllättää varmasti jatkossakin, mikäli niitä tulee vielä tehtyä. Sitä ennen voisi opetella jakamaan oman tekemisensä tasaisemmin koko projektin aikavälille, nopeiden epätasaisten syklien sijaan.

Lopputuloksena oli onneksi molempien mielestä hyvä ja yhteistyö toimi loppuun asti. Edellisestä projektista tuntui olevan myös hyötyä ainakin suunnittelun osalta, ja vaikkakin näissä projekteissa tulee aina eriäviä mielipiteitä, yhteinen sävel löytyi ja molemmat hoitivat oman osuutensa hyvin loppuun asti.

5.3 Loppusanat

Tavoitteenamme oli luoda kokonaisuutena eheä ja toimiva tunnelmaltaan kauhu-genreen sopiva peli, sekä selvittää käyttämämme tekniikan tuoma mahdollinen lisäarvo pelikäytössä. Vaikkakin panoraamakuvien hyödyntäminen sekä mahdollisuudet ovat kohtuullisen rajattuja, oli se mielestämme hyvä tapa tuoda jotain erilaista mukaan peliin, ja ainakin yhdellä osa-alueella se säästi aikaa sekä resursseja, kuten oli tarkoituskin.

Tahdoimme myös projektin edetessä oppia lisää joidenkin ohjelmien käytöstä, ja muutamien täysin uusien ohjelmien sekä alustan käyttöä. Siltäkin osin tavoitteet toteutuivat vähintäänkin hyvin, sillä uutta opeteltavaa oli ehkä odotettua enemmän.

Taidot siis karttuivat projektin edetessä tasaisesti, ja nykyisellä tietotasolla vastaavan projektin toteuttaminen olisi huomattavasti helpompaa.

Toimeksiantajamme tavoitteena oli Vaasan VR-toiminnan kunnolla käynnistäminen, johon tarvittiin laitteistot sekä aiheesta kiinnostuneita tekijöitä. Projektiamme voikin jatkossa käyttää ikään kuin tekniikkademonia palvelemaan osittain tätä tarkoitusta. Tavoitteet toteutuvat siis siltäkin osin, sillä toimeksiantajamme oli peliimme erittäin tyytyväinen.

LÄHTEET

Adobe Systems 2017. Creatice Cloud sovellukset. Viitattu 23.01.2017.
https://creative.adobe.com/fi/plans?promoid=NV3KR7S1&mv=ot-her&store_code=fi

Allegorithmic 2017. Advanced channels painting. Viitattu 23.01.2017.
<https://support.allegorithmic.com/documentation/display/SPDOC/Advanced+channels+painting>

Allegorithmic 2016 a. Substance for Unity: Chapter 04-02 Exporting the textures from Substance Painter. Viitattu 23.01.2017.
<https://www.youtube.com/watch?v=6pbJgDwtEfw>

Allegorithmic 2016 b. UNCHARTED 4: This Game Has Substance. Viitattu 23.01.2017.
<https://www.allegorithmic.com/blog/uncharted-4-game-has-substance>

Autodesk 2017. 3ds Max Educational License & Maya Educational License. Viitattu 23.01.2017.
<http://www.autodesk.com/education/free-software/3ds-max>
<http://www.autodesk.com/education/free-software/maya>

Creative Commons 2017. About The Licenses. Viitattu 23.01.2017.
<https://creativecommons.org/licenses/>

D'Orazio, Savov, theverge.com 1.3.2015. Valve's VR headset is called the Vive and it's made by HTC. Viitattu 17.10.2016.
<http://www.theverge.com/2015/3/1/8127445/htc-vive-valve-vr-headset>

File-Extensions.org 2017. Digital camera RAW file extension list. Viitattu 23.01.2017.
<https://www.file-extensions.org/filetype/extension/name/digital-camera-raw-files>

Guiding Tech 2014. What is the Difference Between JPG, GIF, PNG, BMP Image Formats. Viitattu 23.01.2017.
<http://www.guidingtech.com/7729/difference-between-jpg-png-gif-bmp-image-formats/>

Kelion, bbc.com 1.3.2015. HTC reveals virtual reality headset with Valve at MWC. Viitattu 17.10.2016. <http://www.bbc.com/news/technology-31664948>

Kypreos, trustedreviews.com 2016. HTC Vive review. Viitattu 17.10.2016.
<http://www.trustedreviews.com/htc-vive-review>

Manninen Tony 2007. Pelisuunnittelijan käsikirja: ideasta eteenpäin. Viitattu 31.10.2016. Tallinna. Rajalla.

Microsoft 2016. Editing your code. Viitattu 18.10.2016. <https://www.visualstudio.com/vs/ide/>

Muotoilukeskus Muova, West Coast Startup 2016. Muotoilukeskus MUOVAn verkkosivut. Viitattu 13.12.2016. <http://www.muova.fi/fi/yhteisty/wcs/>

Oculus VR, oculus.com, Powering the Rift. Viitattu 23.01.2017. <https://www3.oculus.com/en-us/blog/powering-the-rift/>

Orland Kyle, arstechnica.com. 5.4.2016. Viitattu 17.10.2016. <http://arstechnica.com/gaming/2016/04/htc-vive-review-you-can-now-buy-your-own-holo-deck-simulator-v-1-0/>

PTGui, 2017. Buy a PTGui License. Viitattu 23.01.2017. <https://www.ptgui.com/order.html>

Roll D6 Games 2016. Pelisuunnittelun vaiheet. Viitattu 19.10.2016. <http://www.rolld6.com/pelikehittajalle/pelisuunnittelun-vaiheet/>

Russell, dentedpixel.com 20.11.2013. C# vs. UnityScript – Which is faster? Viitattu 18.10.2016. <http://dentedpixel.com/developer-diary/c-vs-unityscript-which-is-faster/>

Tom's Hardware 2016. SteamVR Performance Test. Viitattu 23.01.2017 <http://www.tomshardware.com/reviews/steamvr-performance-test-gpu-comparison,4489.html>

Unity Technologies 2016 a. Download Unity, Viitattu 18.10.2016. <https://unity3d.com/get-unity/download>

Unity Technologies 2016 b. Unity Asset Store. Viitattu 18.10.2016. <https://www.assetstore.unity3d.com/en/>

Unity Technologies 2016 c. Creating and using scripts. Viitattu 18.10.2016. <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>

Unity Technologies 2016 d. Unity download archive. Viitattu 18.10.2016. <https://unity3d.com/get-unity/download/archive>

Unity Technologies 2016 e. SteamVR plugin. Viitattu 18.10.2016. <https://www.assetstore.unity3d.com/en/#!/content/32647>

Valve Corporation 2017. Steam - Substance Indie Pack. Viitattu 23.01.2017. <http://store.steampowered.com/sub/98981/>

Vuorela 2007. Pelintekijän käsikirja. Viitattu 31.10.2016. Helsinki. BTJ Kustannus.

