

Tuomas Vaajamäki

Unity HTC Vive -virtuaalitodellisuussovelluksen käännöstyö Oculus Quest -alustalle

Tradenomi
Tietojenkäsittely
Syksy 2019



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä: Vaajamäki Tuomas

Työn nimi: Unity HTC Vive virtuaalitodellisuussovelluksen käännöstyö Oculus Quest-alustalle

Tutkintonimike: Tradenomi (AMK), tietojenkäsittely

Asiasanat: Virtuaalitodellisuus, sovelluskäännös, Oculus Quest, HTC Vive

Opinnäytetyössä dokumentoitiin Unityssä toteutetun virtuaalitodellisuussovelluksen käännöstyötä HTC Vive PC -alustalta Oculus Quest -mobiilialustalle. Pääpainoina olivat visuaalisen ilmeen ja hyväksyttävän suorituskyvyn tasapainottamisen haasteet sekä Input-systeemin muuttaminen alustasta riippumattomaksi. Tavoitteena oli luoda asiakkaalle eli Nokia Globe -tiimille dokumentaatio jo tehdystä käännöstyöstä ja antaa pohja mahdollisille tuleville käännöksille.

Opinnäyte esittelee yleisellä tasolla VR-alustojen perusteita, sillä aihepiirin suhteellisen uutuuden ja vähäisen kehitystyön takia se todettiin tarpeelliseksi dokumentin luettavuuden kannalta. Suoritetun työn aikana pyrittiin tutkia syvemmin varsinkin suorituskyvyn merkitystä virtuaalitodellisuuden kehityksen kannalta, sekä sitä miten kehittäjä voi varautua ja testata sovelluksia eri alustoilla.

Tärkeimmiksi käännöstyön aihealueiksi suorituskyvyn kannalta määriteltiin käytettyjen Unity-materiaalien määrän hallinta, shader-ohjelmien monimutkaisuuden vähentäminen sekä 3D-mallien yksinkertaistaminen tarvittaessa. Myös varsinaisten käyttäjien kanssa suoritettujen testauksien tärkeyttä korostettiin.

Abstract

Author(s): Vaajamäki Tuomas

Title of the Publication: Unity HTC Vive Virtual Reality Application Port to Oculus Quest

Degree Title: Bachelor's Degree in Business Information Technology

Keywords: Virtual reality, software porting, Oculus Quest, HTC Vive

The objective of this Bachelor's thesis was to document the porting of a Unity virtual reality application from the HTC Vive PC platform to the Oculus Quest mobile platform. The focus was on maintaining the balance between visuals and performance. Changing the input system to be platform independent was also important. The thesis was also to serve as documentation for the client, the Nokia Globe VR team, for the completed port work and to serve as a basis for further ports down the line.

The thesis attempts to introduce VR development concepts, as the field is new and not extremely well known even among software developers. During the project an emphasis was placed on the importance of performance and how developers might prepare for performance related issues during porting projects.

The most important measures for achieving desired performance targets were deemed to be controlling the Unity material, shader and model complexity, as well as the consistent use of end user-based testing.

Sisällys

1	Johdanto	1
2	Virtuaalitodellisuus yleisesti	2
2.1	Mitä virtuaalitodellisuus on?	2
2.2	Teknologiset perusteet.....	4
2.3	Käyttötarkoitukset.....	6
2.4	Markkinoiden tämänhetkiset VR-lasit.....	8
3	VR-sovelluskehitys	10
3.1	VR-kehityksessä huomioitavaa	10
3.2	Unity VR.....	11
3.3	Alustojen erot kehitysympäristössä	12
4	Sovelluksen käänös.....	13
4.1	Mikä on sovelluksen käänös.....	13
4.2	Käänöksen hyödyt ja riskit	13
4.3	Usean alustan tukeminen.....	14
5	Suorituskyvyn arviointi	16
5.1	Suorituskyky VR:ssä.....	16
5.2	Tiled rendering	16
5.3	Suorituskyvyn mittaustavat.....	17
6	Työtapausten perusteet.....	19
6.1	Nokia Globe	19
6.2	Lähtötilanne	20
6.3	Tavoitteet	21
6.4	Suunnitelma	21
7	Työtapausten toteutus.....	22
7.1	Oculus-ympäristön asennus	22
7.2	Unity Oculus- asetukset.....	23
7.3	Input-systeemin muutokset	25
7.4	Muutokset visuaaliseen sisältöön	25
7.5	Sovelluksen testaus Questilla.....	26

8	Työtapausten tulokset	28
9	Pohdinta työn suorituksesta.....	30
	Lähteet	31

1 Johdanto

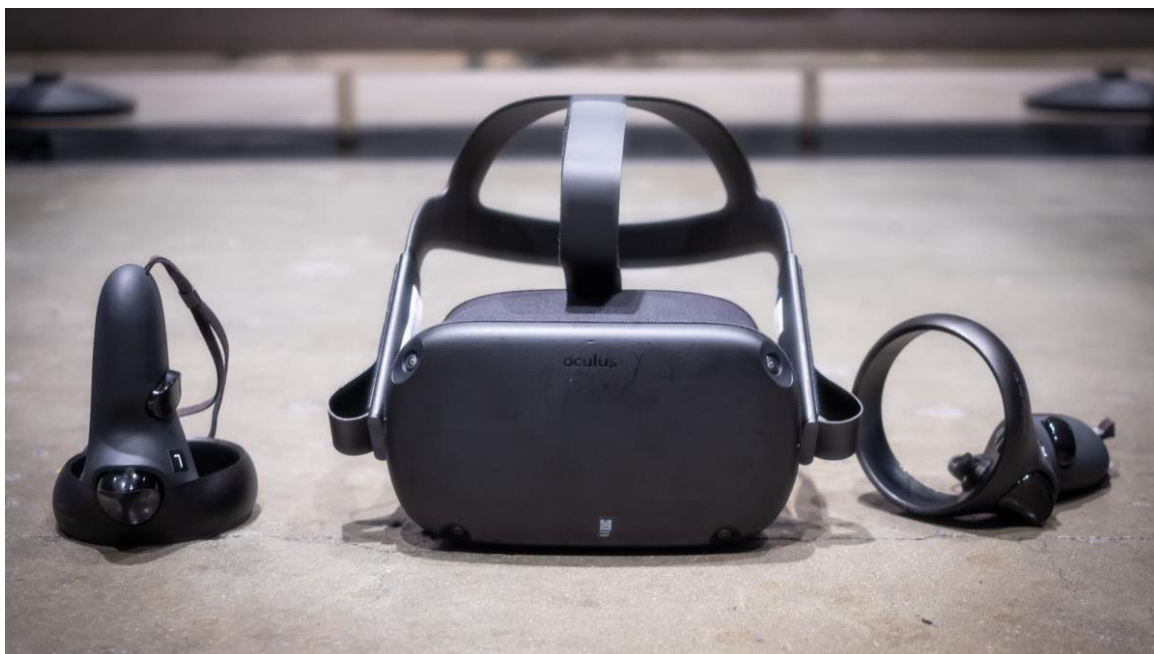
Tämän opinnäytteen tarkoituksena on toimia dokumentaationa kesällä 2019 Oulussa HTC Vive PC VR-alustalta Oculus Quest mobiili VR-alustalle tehdylle sovelluksen käännöstyölle Nokia Globe VR -tiimin tilaamana. Nokia Globe -tiimistä vastuuhenkilönä toimi työn aikana virtuaalitodellisuusinsinööri Sakari Naumanen. Työn ensimmäisen puolen tarkoitus on antaa riittävä teoriapohja varsinaisen toteutusosion ymmärtämiseen, mutta myös toimia tekijälle mahdollisuutena kerrata ja tiivistää tehdyn työn aikana kerättyä tietoa ja osaamista. Osiossa keskitytään selventämään virtuaalitodellisuuden perusteita, usean alustan samanaikaiseen kehitykseen liittyviä ongelmia sekä sovelluksen testaamiseen liittyviä toimintamalleja.

Toisen puoliskon toteutuksesta kertova osio antaa kuvan siitä, miksi ja miten varsinainen työ tehtiin sekä miten ja missä asioissa onnistuttiin. Osion tarkoituksena ei ole toimia tutoriaalina tai suorana ohjeena kehitykseen, vaan antaa osviittaa siitä, mihin mahdollisessa seuraavassa käännöstyössä tulisi alustavasti kiinnittää huomiota. Osa työn tiedosta on supistettu salassapitosopimuksen mukaisesti, mutta samalla on pyritty pitämään yleispätevä asiasisältö rikkoutumattomana ja selkeänä.

2 Virtuaalitodellisuus yleisesti

2.1 Mitä virtuaalitodellisuus on?

Virtuaalitodellisuus eli VR (Virtual Reality) on teknologinen tapa luoda immerstiivinen ja todentuntuinen kolmiulotteinen ympäristösimulaatio. Tämä simulaatio luodaan käyttämällä yhteistyössä tietokonelaitteistolla toimivaa VR-sovellusta ja ympäristön visualisoivia VR-laseja (kuva 1) eli ”Headsettejä”. Immersion eli simulaation ”uppoutumisen” helpottamiseksi ja vuorovaikutuksen mahdollistamiseksi virtuaalisen ympäristön kanssa käytetään lasien yhteydessä yleensä myös käsiin sopivia liikeohjaimia. [1.]



Kuva 1. Oculus Quest VR-lasit ja ohjaimet.

Kuten moni uusi teknologia, myös virtuaalitodellisuuden idea visioitiin ensin scifi-kirjallisuuden sivuilla. Vuonna 1930 julkaistussa tarinassa ”Pygmalion’s Spectacles” Stanley G. Weinbaum kuvaili laseja, joiden avulla käyttäjä pystyi kokemaan täysin immerstiivisen fiktiivisen maailman. 1900-luvun aikana kehitettiin useita erilaisia stereoskooppisia teknologioita sekä viihde- että hyötykäyttöön filmitoiminnasta sotilasvoimien simulaattoreihin. Jo 1960-luvulla kehitettiin ensimmäisiä varsinaisia päässä pidettäviä laseja, joista osa pystyi jopa seuraamaan käyttäjän pään liikkeitä

enemmän tai vähemmän reaaliajassa. Vuosikymmenen lopulla lisättiin lasien toimintaan ensimmäisen kerran myös tietokonesovelluksella tuotettu ympäristö, johon käyttäjä pystyi vaikuttamaan. [2.]

Nykyinen virtuaalitodellisuuden ”nousukausi” alkoi vuonna 2012, kun amerikkalainen teknologiafirma Oculus esitteli ensimmäistä kertaa julkisesti uusia ”Oculus Rift”-lasejaan. Tietotekniikka oli varsinkin älypuhelinien suosion nousun takia edeltävien vuosien aikana keskittynyt pienten mutta tehokkaiden laitteiden ja hyvälaatuisten näyttöjen kehittämiseen, joka antoi hyvän pohjan virtuaalitodellisuusteknologian mahdollistamiselle realistisilla kuluttajahinnoilla. Oculusen saadessa merkittävää huomiota myös muut teknologia-alan yritykset käänsivät huomionsa virtuaalitodellisuuskehitykseen, huomattavimpina näistä Google (Google Cardboard), Samsung (Galaxy Gear), Valve yhteistyössä HTC:n kanssa (HTC Vive), Microsoft (Hololens) sekä Playstation (Playstation VR). [2.]

Nykyään puhuttaessa virtuaalitodellisuudesta nousee esille usein VR:n pelikäyttö. Se on esimerkiksi Oculusen ja HTC Viven suuri painopiste sekä markkinoinnissa että puhuttaessa teknologian kehittämisestä eteenpäin, mutta muut yritysmaailman sektorit ovat myös ottaneet virtuaalisen todellisuuden käyttöönsä. Tämä kehitys tulee myös jatkumaan virtualisoinnin edetessä yhä syvemmälle kaikille yritysmaailman osa-alueista markkinoinnista kommunikaatioon, sekä suunnittelutyöstä myyntiin ja jopa oppimiseen. [3.]

Virtuaalitodellisuudesta puhuttaessa tulee usein esille myös kaksi muuta termiä, ”AR” ja ”MR”. AR eli ”Augmented Reality” tarkoittaa sovelluksen kautta esimerkiksi kädessä pidettävälle näytölle luotavaa tuotosta, joka liitetään todellisesta maailmasta otettavan kuvan tai videon päälle tai ”joukkoon”. Yksi suosituimmista AR esimerkeistä on Pokemon GO peli, jossa todellisen maailman paikkoihin perustuen luodaan keräily- sekä kevyitä roolipelaamisen elementtejä. MR eli ”Mixed Reality” on VR:n sekä AR:n välimuoto, jossa käyttäjä vuorovaikuttaa sekä todellisen maailman että virtuaalisen sovelluksen luoman sisällön kanssa, usein käyttäen kevyempää versiota VR-laseista. Näin käyttäjä voi esimerkiksi saada virtuaalisessa muodossa lisätietoja tai ohjeita todellisen maailman tehtävänsä menettämättä täysin kykyään nähdä sen hetkinen ympäristönsä. [1.]

2.2 Teknologiset perusteet

Näkyvimät virtuaalitodellisuuden luomiseen käytettävistä laitteista ovat varsinaiset VR-lasit. Vaikka eri valmistajat käyttävät eri teknologioita ja osia lasiensa valmistamiseen, ovat peruskomponentit ja niiden tarkoitukset samat. Lasit koostuvat käyttäjän kasvojen eteen aseteltavasta kuoresta, joka kiristetään paikalleen pään taakse kiertävillä hihnoilla. Lasien sisällä on näyttö, joka jaetaan stereoskooppisen periaatteen mukaan kahteen osaan fyysisellä linssikammilla, luoden kolmiulotteisuuden vaikutelman (kuva 2). Linssit myös muokkaavat näytöltä silmille tulevaa kuvaa ja helpottavat silmän tarkentumista virtuaaliseen ympäristöön. [4.]



Kuva 2. Oculus Rift lasien sisältö.

Näytön ollessa näin lähellä silmiä tarvitsee kuvan resoluution olla mahdollisimman korkea, jotta niin sanottu "screen door"-vaikutelma vähenisi. Vaikutelmalla viitataan ilmiöön, jossa käyttäjä huomaa näytön kuvan muodostuvan yksittäisistä pikseleistä rikkoen todellisuuden illusion. Paras keino tämän välttämiseen on suuremman resoluution näytön käyttäminen lasissa, mutta vaikka näyttöteknologia kehittyikin nopeasti, on vaadittava resoluutio edelleen liian kallis tuotettaviksi kuluttajatason VR-laseille. [4.]

Latensi eli tiedonsiirron viiveaika on toinen suuri ongelma VR-laitteiden kehityksessä. Kaikkeen tiedonsiirtoon liittyy tietty määrä viivettä, mutta VR-ympäristössä se on helposti huomattavissa, sillä ihmisen aivot ovat erityisen herkäät silmien ja tasapainoaistin välisille ristiriidoille. Valve yri-

tyksen VR-insinöörin mukaan täysin huomaamaton kuvan latenssiarvo olisi vain noin 10 millisekunnin luokkaa, mutta varsinaisessa käytössä luvut voivat olla jopa kaksinkertaisia tai suurempia, ilman että käyttäjälle aiheutuu viiveestä fyysisiä vaikutuksia. Eri lasien virkistystaajuus, eli kuinka monta kertaa sekunnissa näytön kuva päivitetään, vaihtelee valmistajien ja lasiversioiden mukaan. Tämänhetkisistä suurista markkinoilla olevista lasiversioista Oculuksen Quest-lasit toimivat hitaimmilla taajuuksilla, päivittäen näyttöään 72 kertaa sekunnissa. Suuri taajuus on tarpeen, kun halutaan vähentää käyttäjän latenssin tuntua, mutta se myös nostaa suorituskykyvaatimuksia VR-sovellusta pyörittävältä laitteelta. [4.]

VR-lasien lisäksi useimmat VR-tuotteet käyttävät simulaatiomaailman kanssa vuorovaikuttamiseen käsiohjaimia (kuva 3), joiden avulla käyttäjä voi ”koskettaa”, ottaa kiinni tai siirtää virtuaalisia esineitä. Ohjaimilla on useimmiten myös virtuaaliset versiot sovelluksen simulaatiomaailmassa, antaen käyttäjälle selkeän ja reaaliaikaisen kiintopisteen omaan ruumiiseensa. Nykyisten valmistajien ohjaimet perustuvat jokseenkin peliteollisuudesta tuttuihin malleihin, antaen käyttöön tietyn määrän eri toimintoihin soveltuvia nappeja ja liipaisimia. [4.]



Kuva 3. Vive Cosmos -ohjaimet.

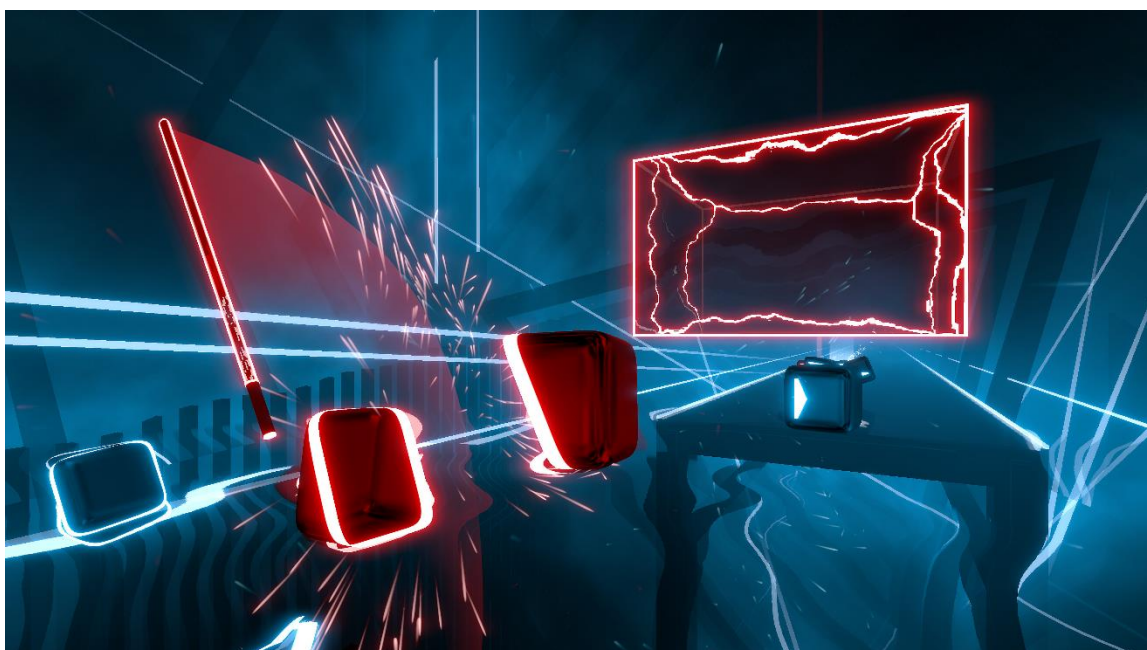
Koska VR-lasien näyttöjen virkistysaika on niin lyhyt, asettaa se myös kovat vaatimukset varsinaista VR-sovellusta ajavalle laitteistolle. Tämä on ongelma, kun halutaan luoda erityisen monimutkaista sisältöä visuaalisesti näyttävällä tavalla esitettynä. Virkistysajan vaatimukset tarkoittavat myös, että monimutkaisen sovelluksen pyörittämiseen tarvitaan sille varattu PC tai vastaava laite, josta luotu kuva siirretään kaapeleiden kautta lasihin. Jotkut VR-laitteiden valmistajista

ovat myös luoneet puhelinteknologiaan pohjautuvia ratkaisuja, kuten Oculus Quest. Tällöin käyttäjän ei tarvitse huolehtia liikkueensa kaapeleista tai investoinnista tehokkaaseen tietokoneeseen, mutta näistä eduista joudutaan maksamaan visuaalisessa laadussa ja sovelluksen monimutkaisuuden pienenemisessä. [5.]

Riippuen VR-sovelluksen suunnitellusta ”kokemuksesta”, voi käyttäjä myös tarvita tietyn määrän vapaata tilaa. Tämä voi olla ongelma varsinkin pienemmissä asunnoissa, joissa käyttäjillä ei välttämättä ole joidenkin sovellusten vaatimaa neljän neliömetrin tyhjää tilaa. [5.]

2.3 Käyttötarkoitukset

Tällä hetkellä pelaaminen ja ”hupikäyttö” ovat VR:n suurimmat käyttötarkoitukset. Alan suurimmat alustavalmistajat, Vive ja Oculus, ovat molemmat panostaneet markkinoinnissaan pelaajakokemuksen parantamiseen sekä VR-pelien kehityksen tukemiseen. VR-pelikehitys on vielä alalla uutta, ja käyttäjille tarjottavien kokemusten laatu on näin ollen erittäin vaihtelevaa. VR-peleistä tällä hetkellä parhaiten tunnettujen joukossa ovat rytmipeli Beat Saber (kuva 4), ajan manipulatioon pohjautuva ensimmäisen persoonan toimintapeli SuperHot VR ja humoristinen ”työpaikka-simulaatio” Job Simulator. Useista jo olemassa olevista peleistä on myös julkaistu täysiä VR-versioita, kuten suosittu Skyrim pelin HTC Vive -versio. [1.]



Kuva 4. Beat Saber peli.

Virtuaalitodellisuus tuo pelaamiseen uutta ulottuvuutta mahdollistamalla valtavasti lisäämmäsiota pelimaailmisiin ja lisäämällä kolmiulotteista fysikaalisuutta. Teknologian kehittyessä päästään myös eroon VR:n suurimmista haittatekijöistä pelaamisen suhteen tekemällä laseista mukavampia pitää päässä pitkiä yhtenäisiä ajanjaksoja, vähentämällä tarvittavaa laitteistoa sekä tuomalla uusia tapoja vaikuttaa virtuaalisiin maailmiin. Virtuaalitodellisuutta käytetään myös pelien ulkopuolella yhä laajemmin. Yksi ensimmäisistä hyvin tunnetuista VR-sovelluksista oli Googlen Tiltbrush piirto-ohjelma (kuva 5), joka antaa käyttäjille mahdollisuuden ilmaista itseään kolmiulotteisessa ympäristössä luomalla taidetta uusin tavoin. Käyttäjät voivat myös jakaa omia tuotoksiaan sekä katsella muiden jakamaa sisältöä. [6.]



Kuva 5. Google Tiltbrush VR-piirtosovellus.

Hyötykäyttötarkoituksia VR:lle löytyy myös lääketieteen, suunnittelun, markkinoinnin, koulutuksen ja kommunikaation alueilta. Virtuaalitodellisuutta on jo käytetty myös psykologisten ongelmien tutkimiseen ja hoitamiseen luomalla esimerkiksi posttraumaattisten stressihäiriöiden ahdistustiloja laukaisevia tilanteita VR-ympäristössä. Yrity maailmassa esimerkiksi autoteollisuus on laajasti ottanut käyttöön VR-alustoja suunnittelussa, testauksessa ja markkinoinnissa. Vaikka peliala onkin tällä hetkellä suurin VR-sovellusten tuottaja ainakin määrällisesti, tulee hyötykäyttöön tarkoitettujen VR- ja AR-sovellusten lisääntyminen olemaan varmaa lähivuosina. [7.]

2.4 Markkinoiden tämänhetkiset VR-lasit

Myytyjen VR-lasien osalta Sony on ollut viimeisen kahden vuoden aikana nousussa, julkaistuaan muihin headset-valmistajiin verrattuna hyvin hinnoitellut pelaamiseen suunnatut Sony VR-lasinsa. Vuonna 2018 Sony vei maailmanlaajuisesti jopa 43 % osuuden markkinoista, Oculuksen ollessa toisena noin 20 % osuudellaan ja HTC:n tullessa kolmanneksi noin 13 % osuudella. Vuoden 2019 aikana Oculuksen osuus kuitenkin tulee laskelmien mukaan nousemaan noin 10 % yksiköllä, suurin osin langattoman Quest-alustan julkaisun ansiosta. Headsetien tuoteperheistä tällä hetkellä aktiivisina ovat Sonyn PSVR, joka toimii ainoastaan Sonyn Playstation pelikonsolin yhteydessä, Oculuksen langallinen Rift ja langaton Quest, HTC:n Vive Cosmos sekä Valven tuottama Steam VR Index. Tämän opinnäytteen toteutusosiossa keskitymme lähinnä HTC Viveen ja Oculus Questiin. [8.]

Tällä hetkellä langattomuus on headsetien suurin kehitystyön alue, ainakin markkinoinnin näkökulmasta. Langalliset, raskaat ja teknisesti kyvykkäät lasimallit toimivat teknologian ollessa uusia uutta, kun headsettejä ostivat asiaa hyvin tuntevat ”early adopter”-tyyppiset käyttäjät. Nämä käyttäjät olivat jo suurilta osin ympäristöissä ja ekosysteemeissä, joissa he pystyivät jo vähentämään niin sanotun ”raskaan” headsetin tuomat ongelmat, kuten kaapeleiden määrän, lasien epämukavuustekijät sekä sovellusten vaatimat tehokkaat PC laitteistot. [9.]

Nyt valmistajat kuitenkin haluavat laajentaa käyttäjäkuntaansa, jolloin kevyempi mutta vähätehoisempi laite on helpompi markkinoida niin sanotulle keskivertokäyttäjälle. Näin esimerkiksi Oculus päätyi tuottamaan Android-alustaan pohjautuvat, puhelimissakin käytettävää teknologia-pohjaa hyödyntävät langattomat Oculus Quest -lasinsa. Quest-lasit eivät tarvitse erillistä ulkoista prosessointia vaan kaikki lasien sovellus suoritus tapahtuu itse lasien prosessoreissa. Tämä lukitsee Questin suorituskyvyn tietylle asteelle, joka on huomattavasti alhaisempi kuin langallisten headsettien sovellusten pyörittämiseen tarkoitettut tietokoneet. Se tarkoittaa, ettei laitteen kykyä voida nostaa samalla tavoin kuin esimerkiksi HTC Viven langallisilla laseilla. Uhraamalla tämän skaalautuvuuden langattomat lasit saavuttavat ketterämmän, kevyemmän ja vähemmän fyysisesti rajoittavan kokemuksen, joka on erityisen haluttava käyttäjille, jotka joutuvat käyttämään laseja pitkiä aikoja esimerkiksi ammattitilanteissa. [9.]

Vaikka tällä hetkellä suurimmat VR-alustoista ovat PSVR, Oculus sekä HTC, on markkinoille tulossa myös muita alustoja. Esimerkkinä näistä on suomalainen Varjo, joka tähtää erityisen korkeatasoisilla laseillaan yritysalan markkinoille (kuva 6). Kuluttajatason lasien sijaan Varjon headset on huo-

mattavasti kalliimpi, ja se on ensisijaisesti tarkoitettu yritysten suunnittelu-, simulaatio- ja tutkimuskäyttöön. Varjo onkin julkistanut yhteistyökumppaneihinsa kuuluvan esimerkiksi autovalmistaja Audin, joka on käyttänyt laseja autojensa fyysiseen testiajoon. Aika näyttää, mikä suunta on virtuaalitodellisuudelle tuottoisin. [10.]



Kuva 6. Varjo VR-2 ja VR-2 Pro headsetit.

3 VR-sovelluskehitys

3.1 VR-kehityksessä huomioitavaa

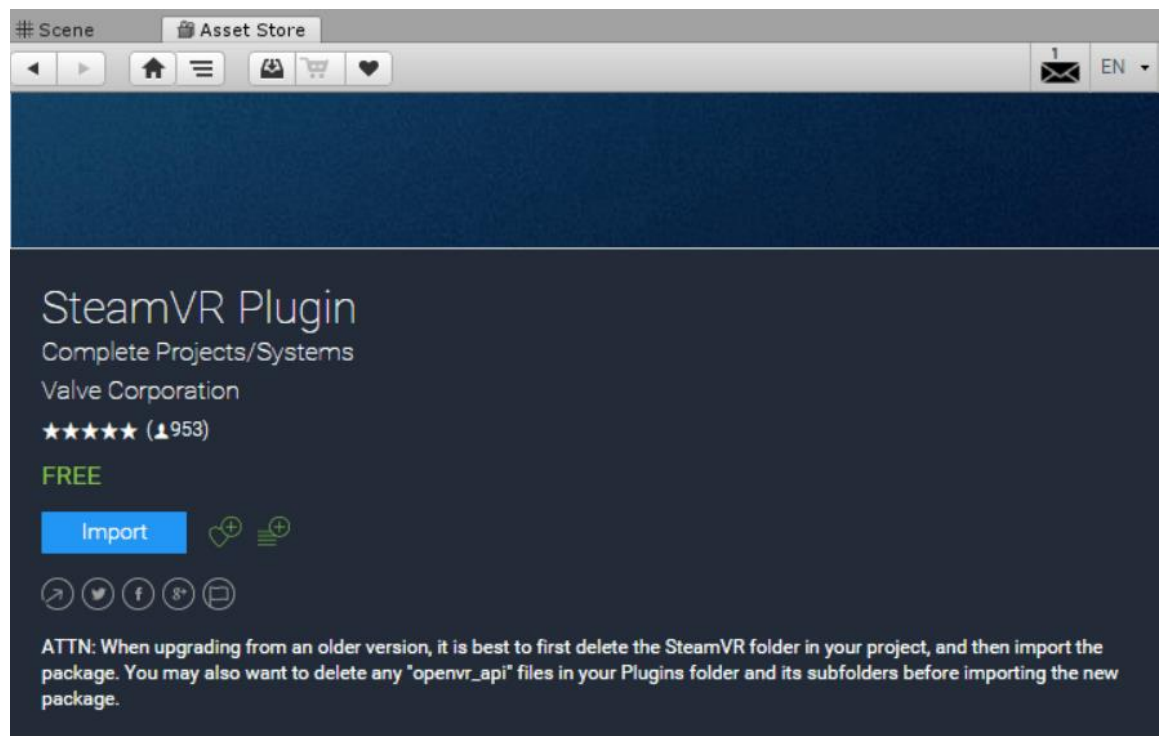
Edellisessä osiossa käytiin läpi virtuaalitodellisuuden mahdollisia teknisiä hankaluuksia, jotka tulee tietysti ottaa huomioon kehityksessä. Vaikka nopeasti ja tasaisesti toimiva sovellus on aina toivottava, ei VR:n ulkopuolella pienillä virkistystaajuuden häiriöillä ole käyttäjälle suurta merkitystä. Suorituskyvyn optimisaatio ei siis VR:n ulkopuolella ole aina välttämätöntä, riippuen sovelluksen kohdealustasta. Virtuaalitodellisuudessa pienikin latenssihäiriö on käyttäjälle huomattava, ja se voi jopa aiheuttaa fyysisiä reaktioita kuten pahoinvointia ja päänsärkyä. [11.]

Virtuaalitodellisuuteen tarkoitetuilla sovelluksilla on myös keskivertosovellusta korkeampi käyttäjän oppimiskynnys, sillä VR-laitteiston käyttö ohjaimineen tuo oman lisäoppimistarpeensa. Perinteisissä näyttöpohjaisissa sovelluksissa on usein jo olemassa oleva oletuskieli esimerkiksi hiiren tai näppäimistön käyttötavoille, mutta tällaisia valmiita tapoja ja standardeja ei vielä ole syntynyt VR-ympäristöille. VR-vuorovaikutuksen tavat on siis selitettävä käyttäjälle selkeästi mutta tarpeeksi nopeasti, mikä tuo lisää VR-kehitykseen niin sanotusti ylimääräisiä kerroksia. Koska VR-sovellusten kehitys on vielä kohtuullisen uutta, ei varsinaisia standardeja kehitykseen ole vielä löydetty suorituskyvyn optimisaatiovaatimusten ulkopuolella. Näin ollen uusien ratkaisujen löytämiseen tarvitaan paljon kokeilua, joka vie kehitysaikaa. VR-sovellukset myös ”vaativat” käyttäjiltään enemmän ja näin ollen heidän kokemuksissaan on enemmän vaihtelua, joten käyttäjätestaus on hankalampaa. [11.]

Sovelluksissa käyttäjä vaikuttaa ympäristöön jollakin määritetyllä välineellä, kuten hiiri, näppäimistö, peliohjain tai kosketusnäyttö. Riippuen sovelluksen kehitystavasta näiden välineiden ja sovelluksen välillä on syötejärjestelmä, joka vastaa siitä, miten käyttäjän komentoja käsitellään. Pelimoottoreiden yhteydessä tätä järjestelmää kutsutaan Input-systeemiksi. VR-kehityksessä Input-systeemit ovat huomattavia, sillä eri alustojen ohjaimet ovat erilaisia ja jokaisella alustalla on oma tapansa hoitaa käyttäjän komentojen käsittely. Pelimoottorien käsittelytapa myös muuttuu riippuen kohdealustasta. Kehityksessä on kuitenkin mahdollista luoda yleinen alustoja tiettyyn pisteeseen asti yhdistävä taso, vähentäen tarvetta alustakohtaiselle kehitykselle. Tämä on tärkeää sovelluksen ”notkeuden” takia. [12.]

3.2 Unity VR

Unity on alustariippumaton pelikehitysmoottori eli pohjasovellus, joka tarjoaa työkaluja ja valmiita toiminnallisuuksia kehitykseen. Se tarjoaa tukea muun muassa fysiikkalaskentaan, renderointiin eli kuvan piirtoon ruudulle sekä saman koodin käyttöön useilla eri alustoilla. Unity itsessään tukee kehitystä yli 25:llä eri alustalla, kuten PC, iOS, Android, Mac, Playstation 4, Xbox sekä useimmilla VR-alustoilla. Unity on tällä hetkellä pelialan suurin VR-kehityksen alusta noin 60 % sovelluksista käyttäessä moottoria. Yritys onkin panostanut virtuaalitodellisuuskehityksen tukemiseen ja on vuoden 2019 lopulla uudistamassa osia VR-työkaluistaan. Eri VR-alustojen valmistajat myös tarjoavat omia Unity-pakettejaan kehityksen aloittamisen avuksi (kuva 7). [13.]



Kuva 7. Unity SteamVR-lisäosa.

Virtuaalitodellisuuskehitys voi olla haasteellista Unityssä sillä kehityksen työkalut ovat vielä uusia ja muuttuvat muita pelimoottorin osia useammin. Osa moottorin valmiista ominaisuuksista ei myöskään toimi eri VR-alustoilla, ja kehittäjä joutuu luomaan räätälöityjä ratkaisuja riippuen laitteistosta. [13.]

3.3 Alustojen erot kehitysympäristössä

Kehittäjä voi ladata VR-kehityksen aloittamista varten alustasta riippuen erilaisia paketteja Unityyn. Esimerkiksi Oculus Quest kehityksen voi Unityssä aloittaa lataamalla Unityn Asset Storesta Oculuksen pluginin eli lisäosan, joka tuo mukanaan joukon esimerkkejä erilaisista tavoista toteuttaa VR-vuorovaikutuksia. Tärkeänä osiona lisäosa myös sisältää Oculus-ohjainten Input-systeemin yhdistämiseen Unityyn tarvittavat komponentit. Oculuksen Plugin sisältää myös tarvittavat ohjelmaosiot muun muassa niin sanottuun sovelluksen ”allekirjoittamiseen” eli siihen, miten Oculuksen omat järjestelmät varmistavat valmiin sovelluksen alkuperän luotetuksi lähteeksi. HTC Viven kehitys vaatii Valven omistavan Steam-pelikaupan asentamista ja Valven kehittämän SteamVR-paketin lataamista. Samalla tavoin kuin Oculuksen luoma paketti Valven ratkaisu tuo mukanaan valinnaisia esimerkkejä sekä apua input-systeemin käyttöön. [14.]

Minkä tahansa teknologian käyttöönoton yhteydessä kehitystä usein nopeuttaa hyvä dokumentaatio, jonka perusteella kehityksessä päästään alkuun. Tämä on varsinkin tarpeellista, kun kyseessä on uudempi teknologia kuten VR, eikä aiheesta ole yhtä suurta yhteisöllistä ja yleistä tietopohjaa. Oculus ja Vive molemmat antavat kehittäjille pääsyn ilmaiseen dokumentaatioon, mutta niiden laatu ei ole täysin samalla tasolla, Oculuksen materiaalin ollessa selkeämpi ja laajempi kokonaisuus. Alustojen virallisen dokumentaation ja tukipalveluiden ulkopuolella VR-kehityksen yhteisöt ovat yrittäneet myös tuottaa omia ratkaisujaan yleisiin ongelmiin. Yksi näistä työkaluista on VRTK eli ”Virtual Reality Toolkit”, jonka lataamalla ja liittämällä Unity VR -projektiin voidaan saada tiettyjä valmiita ratkaisuja esimerkiksi liikkumiseen, vuorovaikutuksiin sekä erilaisiin käyttöliittymäongelmiin. [14.]

4 Sovelluksen käännös

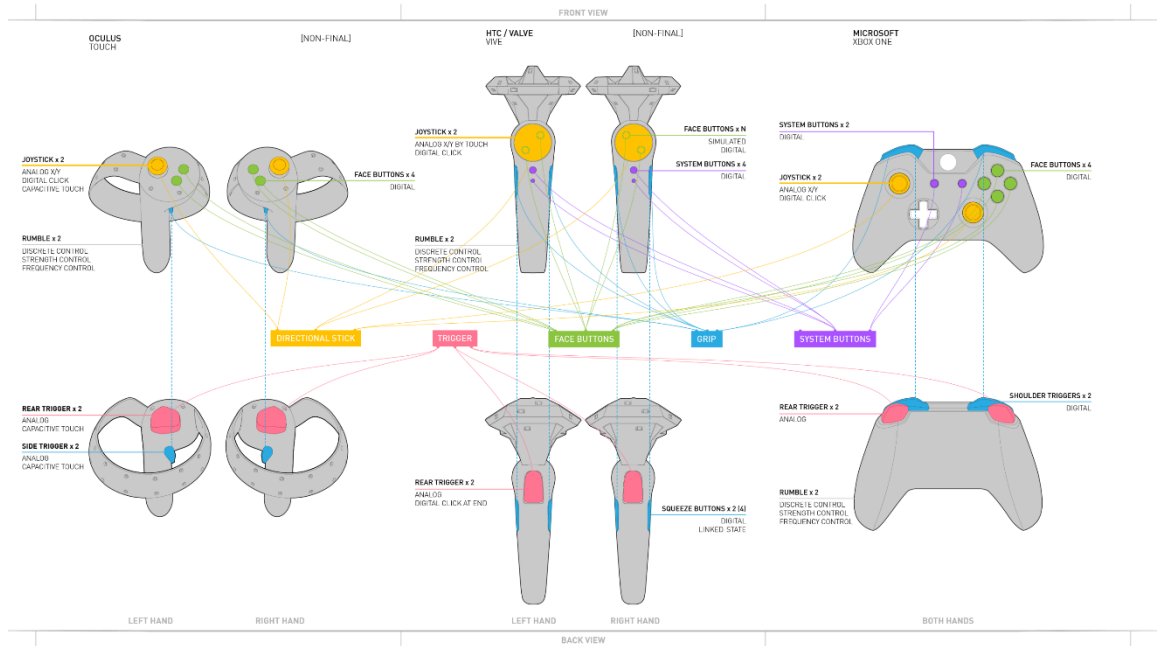
4.1 Mikä on sovelluksen käännös

Sovelluskäännöksellä tarkoitetaan ohjelman toiminnallisuuksien siirtämistä toiseen ympäristöön, esimerkiksi Android-mobiilialustalla toimivan pelin tuomista iOS-alustalle ilman, että käyttäjäkokemus kärsii tai muuttuu huomattavasti. Riippuen alustojen eroista tämä ei kuitenkaan aina ole mahdollista, varsinkin puhuttaessa VR-laitteista, joiden välillä ei ole samanlaista standardointia kuin esimerkiksi puhelimien tai PC-ohjainlaitteiden. Käännöstä suunnitellessa tulee ottaa huomioon, mitkä alustojen välisistä eroista vaikuttavat eniten sovelluksen toimintaan. Onko kyseessä laitetyyppien välinen käännös, jolloin paino on käyttäjän interaktiotoivoilla ohjelmiston kanssa, vai onko alustojen välillä suuri suorituskykyero? [15.]

Ohjelmallisia eroja tulee myös tutkia, kuten tukeeko kohdealusta kaikkia alkuperäisessä sovelluksessa käytettyjä kehityksen keinoja. Näihin kuuluvat muun muassa käytetty ohjelmointikieli ja sen mahdolliset laajennukset tai lisätyt koodikirjastot. Unityä käytettäessä tulee ottaa huomioon mitkä moottorin osioista toimivat kohdealustalla, joko täydellisesti tai osittain, ja miten mahdollisia vajavaisuuksia voidaan paikata. Unity on yleisesti hyvin usean alustan samanaikaiseen kehitykseen soveltuva pelimoottori, sillä se antaa kehittäjälle hyvät työkalut olemassa olevan koodin ja komponenttien automaattiseen siirtoon useiden alustojen välillä. Tämä valmius kuitenkin perustuu siihen, kuinka paljon kunkin alustan kehitysvalmiuksiin on Unityn puolesta ehditty panostaa. VR-valmiuksien osalta tämä prosessi on vielä osin kesken. [16.]

4.2 Käännöksen hyödyt ja riskit

Suurin käännöksen hyödyistä on laajemman mahdollisen käyttäjämäärän tavoittaminen. Uusien alustojen tutkiminen ja tukeminen voi myös tuoda uusia näkökulmia sekä ideoita kehitykseen ja auttaa löytämään ennen huomaamattomia heikkouksia tai vikoja aikaisemmin tehdyissä ratkaisuissa. Varsinkin VR-kehityksessä uusien ohjainratkaisujen ja vuorovaikutustapojen löytäminen helpottuu, kun uusi alusta ”pakottaa” etsimään uusia näkökulmia (kuva 8). [17.]



Kuva 8. Eri VR-alustojen ohjainvaihtoehtoja.

Suuri hankaluus ja mahdollinen riskitekijä käännöstyössä on uusien teknologioiden tutkimiseen ja opettelemiseen käytetyn ajan määrä. Tuntemattomia systeemeitä on myös hankalaa arvioida etukäteen, joten käännöstyössä voi tulla vastaan ennalta-arvaamattomia ongelmia. Riippuen kohdeteknologiasta voidaan myös joutua luomaan räätälöityjä sovellusratkaisuja, jotka pahimmassa tapauksessa hidastavat tai estävät muita kehityshaaroja. [17.]

4.3 Usean alustan tukeminen

Kun kehityksessä halutaan tukea uutta alustaa, joudutaan miettimään, mitä riskitekijöitä kohdelaitteisiin tai teknologioihin liittyy. Huomattavin haitta on usein ajan tarve, sillä jokainen uusi alusta lisää oman tarpeensa suunnitteluun, toteutukseen ja testaukseen. Jokaista uutta sovellukseen lisättävää osaa tai teknologiaa pitää myös jatkossa miettiä kaikkien olemassa olevien versioiden kannalta. [18.]

Usein sovelluskehityksessä valitaan tärkein alusta, jolle uudet toiminnot kehitetään ensimmäisenä ja siirretään valmiina muille tuetuille alustoille. Päätettäessä pääalustaa projektille tulee harvita kehityksen suorittavan tiimin kykyjä, kohdeyleisön tai asiakkaan tarpeita sekä markkinatodellisuutta eli minkälainen nykyinen ja mahdollinen tulevaisuuden näkymä on esimerkiksi tuettujen

alustojen kohdalla. Ottaen huomioon käyttäjien tarpeet ja asiakkaan toiveet päätetään, mitä alustoja halutaan tukea. VR-kehityksessä alustat ovat kohtuullisen hyvin eroteltavissa, mikä auttaa pääalustan valinnassa. Suurin tämänhetkisistä huomioon otettavista tekijöistä on se, halutaanko tukea ”Stand-alone”-headsettejä, eli laitteita kuten Oculus Quest, joka ei käytä PC-laitteistoa sovelluksen suoritukseen. Heikomman suorituskyvyn omaavat laitteet vaativat kehitykseltä erityistä huolta, sillä ne rajoittavat sovelluksen monimutkaisuutta. Tuotteen mukaan onkin tarkasti harkittava, onko johdottomuuden tuoma vapaus sen vaatimien rajoitusten arvoista. [18.]

5 Suorituskyvyn arviointi

5.1 Suorituskyky VR:ssä

Yleisesti sovelluskehityksessä ”suorituskyvyllä” viitataan sovelluksen tai alustan kykyyn toimia tietyllä määritetyllä nopeuden tasolla. Peli- ja VR-kehityksessä tämä taso usein määritellään näytön virkistysnopeuden tasaisuuden perusteella. Varsinkin virtuaaliodellisuuden alustoilla tasainen latenssitaso on erityisen tärkeää, sillä vaihtelevainen virkistysnopeus voi aiheuttaa fyysisiä oireita käyttäjässä, ja tästä syystä optimisaatioon tulisi panostaa erityisesti. VR-alustoilla virkistysnopeuden kohdenopeus määrittyy käytetyn näyttötekniologian mukaan. HTC Viven maksiminopeus on 90 Hertsiä (Hz), eli näytön kuva päivittyy ylimmillään 90 kertaa sekunnissa tai noin 11 millisekunnin välein. Sama luku Oculus Questin kohdalla on 72 Hz. [19.]

Pelimoottorit ja niiden perusteelle tehdyt sovellukset usein käyttävät näitä virkistysnopeuksia päivitystaajuuksina, eli esimerkiksi komponenttien ja objektien tiloja lasketaan uudelleen jokaisen virkistyskohdan aikana. Jokaiselle laskentakerralle pitää myös saada mahtumaan varsinaisen näytölle piirretyn kuvan laskenta ja muodostus. Tämä on VR-kehityksessä yksi suurimmista ongelmakohdista, testauksen kohteista ja optimisaation alueista. [19.]

5.2 Tiled rendering

Tiled rendering eli niin sanottu paloitettu piirto on mobiilialustoilla käytettävien grafiikkaprosessoreiden tapa piirtää näytölle muodostettava kuva, joka eroaa PC-alustoilla käytettävästä piirtotai ”renderointi”-tavasta. Peruserona tavoissa on se, että Tiled renderingiä käytettäessä kuva jaetaan ennalta määritettyihin paloihin, joiden piirrosmatematiikka lasketaan pala kerrallaan ja laskennan tulos siirretään laitteen muistiin odottamaan kaikkien palojen valmistumista. Tämä tehdään, koska mobiilialustojen grafiikkapiireillä on käytettävissään huomattavasti vähemmän nopeaa muistia, joten koko näytön kuvan kerralla piirtämiseen tarvittava tieto olisi liian hidasta siirtää prosessoriin. [20.]

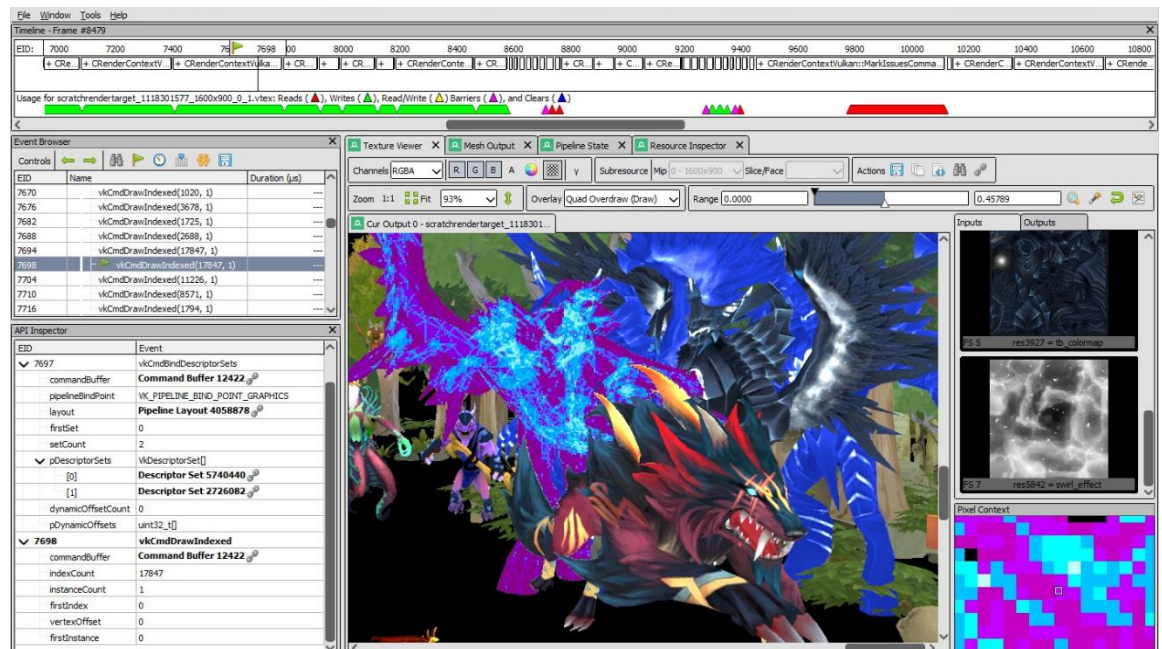
Suorituskyvyssä tämä tulee esille esimerkiksi, kun halutaan lisätä post-processing eli jälkikäsitteilyefektejä kuvaan. PC-alustalla piirrettäessä kuvaa kukin efekti lasketaan vain kerran, mutta tile-

pohjaisessa piirroksessa efektiin tieto tulee välittää uudestaan jokaiselle palalle. Tämä moninkertaistaa efektiin ”käsittelyhinnan”. Kehityksessä tulee siis olla tarkkana ja varmistaa, mitkä halutut efektit tai kuvankäsittelytavat saavat aikaan ylimääräisiä piirtokutsuja. [20.]

5.3 Suorituskyvyn mittaustavat

Monilla alustoista kuten PC, Android ja iOS on useita tapoja seurata suorituskykyä sekä reaaliajassa että nauhoittaa suorituskyvyn vaihtelua analysoitavaksi myöhemmin. Näitä työkaluja voidaan sovelluskehityksessä käyttää hyödyksi, kun halutaan selvittää esimerkiksi, mikä laitteiston osa-alue on suurimmassa käytössä, ja sen kautta päätellä, mitä ohjelmiston osaa tai komponenttia tulisi uudistaa, optimoida tai karsia suorituskyvyn parantamiseksi (kuva 9). [21.]

PC-pohjaisilla VR-alustoilla voidaan suurimmaksi osaksi käyttää samoja analyysiohjelmiä kuin muussakin PC-kehityksessä testattavan sovelluksen toimiessa tietokoneella. Langattomien VR-laitteiden kuten Oculus Questin hoitaessa kaiken prosessoinnin itse lasien sisällä tarvitaan ratkaisuja, jotka pystyvät välittämään tietoa suorituskyvystä testaajalle. Quest on Android pohjainen alusta, joten osa Android-kehityksessä yleisesti käytettävistä työkaluista toimii myös Quest-kehityksessä. Oculus tarjoaa kehittäjien käyttöön omia työkalujaan tarkoitukseen. [21.]



Kuva 9. RenderDoc-suorituskykymittaustyökalu.

Oculuksen oma OVR Metrics Tool, joka on ilmaiseksi ladattavissa Oculuksen kotisivustolla asennetaan Questille, jonka jälkeen työkalua voidaan käyttää näyttämään reaaliajassa tietoa esimerkiksi keskussuorittimen tai grafiikkasuorittimen tilasta. Tämä tieto voidaan myös tallentaa ja siirtää myöhemmin analysoitavaksi kehitysympäristössä. Quest voidaan myös liittää usb-kaapelilla tai lähiverkkoyhteydellä kehityskoneeseen ja käyttää yleistä Android-alustan kehitykseen tarkoitettua Android Device Monitor ohjelmaa suorituskyvyn analysointiin. [22.]

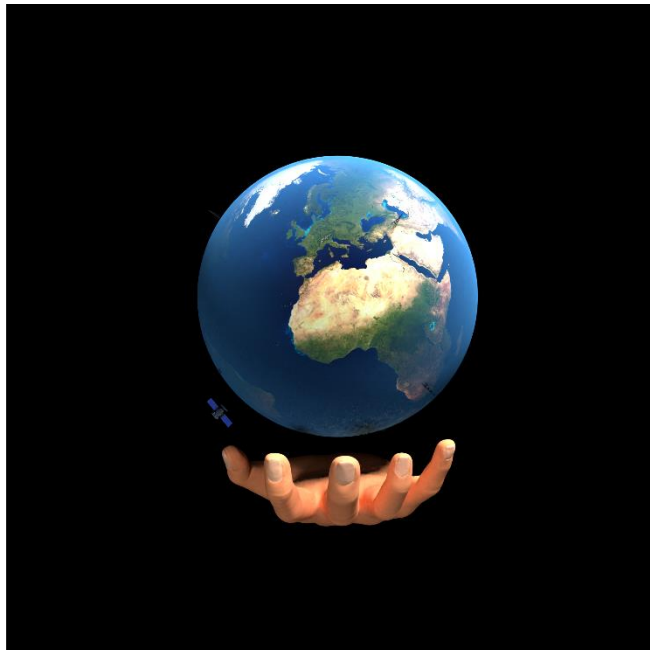
Unity antaa kehittäjille mahdollisuuden käyttää ”Profiler”-työkalua, mutta sen toimiminen on rajoitettua langattomilla VR-alustoilla, sillä se vaatii suoran yhteyden kehitystilassa olevaan Unity moottorisovellukseen toimiakseen tarpeeksi laajasti. Sovelluksen optimisaatiotestausta tehtäessä on tärkeää pitää selkeästi kirjaa siitä, mitä kussakin yksittäisessä testissä on muutettu ja miten suorituskyky on sen mukaan muokkaantunut. Jos käytetään useita eri työkaluja, on kulloinkin käytetty työkalu tärkeää listata. Jos testausta on tekemässä suurempi määrä kehittäjiä, on tarkkojen testauskriteerien määrittely välttämätöntä, jotta kerättyä tietoa voidaan soveltaa kehitysuunnan määrittelyyn.

6 Työtapausten perusteet

6.1 Nokia Globe

Nokia Globe on virtualisointiympäristöä kehittävä projektitiimi (kuva 10), jonka tarkoituksena on tällä hetkellä tarjota Nokian sisäisesti lisää työkaluja muun muassa suunnittelun, kehityksen ja markkinoinnin alueilla. Tähän pyritään tutkimalla ja kehittämällä ratkaisuja uusien teknologioiden, kuten esimerkiksi virtuaalitodellisuussovellusten sekä tekoälykehityksen kautta. Tiimin käyttämät teknologiat ovat nopeasti eteneviä, joten kehityksessä pyritään käyttämään ketteriä työtapoja.

Esimerkkinä kehitetyn VR-sovelluksen käyttötarkoituksista toimii sen mahdollistama "Smart Product Catalog"-toiminnallisuus, joka antaa käyttäjälle tavan esitellä Nokian radio- ja antennituotteita virtuaalisesti, jopa jo prototyyppivaiheessa, ilman että uudesta suunnitellusta tuotteesta tarvitsisi valmistaa fyysistä mallia.



Kuva 10. Nokia Globen logo.

Huomion arvoista on, että olen työn kirjoitushetkellä Nokian vakituksena työntekijänä salassapitovelvollisuuden alainen, joten työn toteutuksen joitakin yksityiskohtia joudutaan jättämään tästä dokumentista pois. Yritän kuitenkin kertoa yleisemmistä itse tuotteeseen suoraan liittyvistä asioista mahdollisimman yksityiskohtaisesti. Työtä valvova Nokia Globe -tiimin VR-

insinööri Sakari Naumanen on tarkastanut dokumentin ja todennut sen olevan salassapitosopimuksen mukainen.

Olin kesän 2019 aikana työharjoittelussa Globe-tiimissä VR-kehittäjänä. Tänä aikana Oculus julkaisi Quest-headsettinsä, jota oli suunniteltu mahdolliseksi seuraavaksi alustaksi VR-sovelluksen kehitykselle, ja sen käyttöönottoa varten tarvittiin selvitys- ja tutkimustyötä mahdollisen sovelluskäännöksen vaatimuksista.

Selvitystyö oli laaja mutta erittäin kiinnostava prosessi, jonka jälkeen aloitettiin varsinainen käännöstyö. Olin vastuussa lähinnä visuaalisten elementtien käännöksestä ja siihen liittyvän optimisaatiotyön suunnittelusta ja toteutuksesta. Kokonaisuutena käännöstyöstä saatiin paljon uutta kokemusta ja tietoa VR-kehitykseen liittyen, mutta informaation kerääminen ja dokumentaatio oli osittain puutteellista. Halusin kirjoittaa opinnäytteeni aiheesta voidakseni kerrata saatuja oppoja itselleni sekä toivottavasti kiteyttää tietoa tietyssä määrin myöhempää käyttöä kuten seuraavan alustan käännöstyötä varten.

6.2 Lähtötilanne

Ennen käännöstyötä tuotettu sovellus toimi vain HTC Vive PC-ympäristössä, mikä rajoitti sen mahdollista käyttöä esimerkiksi nopean markkinoinnin työkaluna. Myös PC-käyttöön tarvittavan laitteiston hankintahinta on paljon Oculus Quest ratkaisua korkeampi, joten siirtämällä sovellus Questille laajennetaan mahdollista käyttäjämäärää selvästi.

HTC Vive -ympäristössä suorituskyky riippuu sovellusta suorittavan tietokoneen tehosta, joten suorituskykyongelmista voitiin päästä eroon niin sanotulla raa'alla koneteholla. Kehityskoneilla käytettynä sovellus toimi tarvittavan 90 Hz:n virkistysnopeuden tasolla lähes aina, mutta heikommilla kannettavilla PC-laitteilla ei aina yletty tarvittaviin lukuihin.

VR-sovelluksen osana halutaan esittää Nokian tuotteiden 3D-malleja eli kolmiulotteisia representaatioita joko jo markkinoilla tai vielä suunnitteluasteella olevista radiotuotteista. Nämä 3D-mallit ovat kuitenkin erittäin monimutkaisia ja näin ollen vaativat grafiikkaprosessorilta paljon laskentatehoa. PC-ympäristössä grafiikkaprosessorit on usein suunniteltu tämäntyyppiseen laskentaan, mutta sovelluksen suorituskyky huononee helposti siirryttäessä käyttämään mobiiliarkkitehtuurin prosessoria. Miten näitä 3D-malleja siis käsitellään Questilla oli tärkeä selvityksen aihe käännöstä suunniteltaessa ja tehtäessä.

6.3 Tavoitteet

Työssä haluttiin siirtää kaikki valmiin sovelluksen toiminnallisuus ja käyttötarkoitukset mahdollisimman rikkoutumattomana Quest-alustalle. Suorituskykyvaatimuksia pidettiin kuitenkin niin tärkeinä, että joistakin PC-version ominaisuuksista oltiin valmiita luopumaan, jotta vaadittava virkistystaajuuden tasainen nopeus saavutettaisiin.

Samalla haluttiin myös tarkemmin määritellä alustan suorituskyvyn rajoituksia tulevaisuuden varalle, jotta pystyttiin muodostamaan kuva siitä, miten alustojen eriävyyksiin voitaisiin sovelluksen sisällön suunnittelussa ja toteutuksessa varautua. Koska HTC Vive- ja Oculus Quest-alustojen Input-systeemien välillä on eroavaisuuksia, tuli myös löytää ratkaisu siihen, miten interaktiosysteemit jatkossa suunniteltaisiin.

Työ antoi myös mahdollisuuden tutkia jo olemassa olevia sovelluksen visuaalisia komponentteja ja määritellä, tulisiko niiden luomiseen ja muokkaamiseen käytettyihin järjestelmiin ja käytäntöihin tehdä muutoksia. Haluttiin myös luoda dokumentaatio sille, miten Oculus-kehitysympäristö saadaan tehokkaasti ja helposti toimintaan niitä tiimin jäseniä varten, jotka käännöstyön aikana suorittivat muita sovellukseen liittyviä tehtäviä.

6.4 Suunnitelma

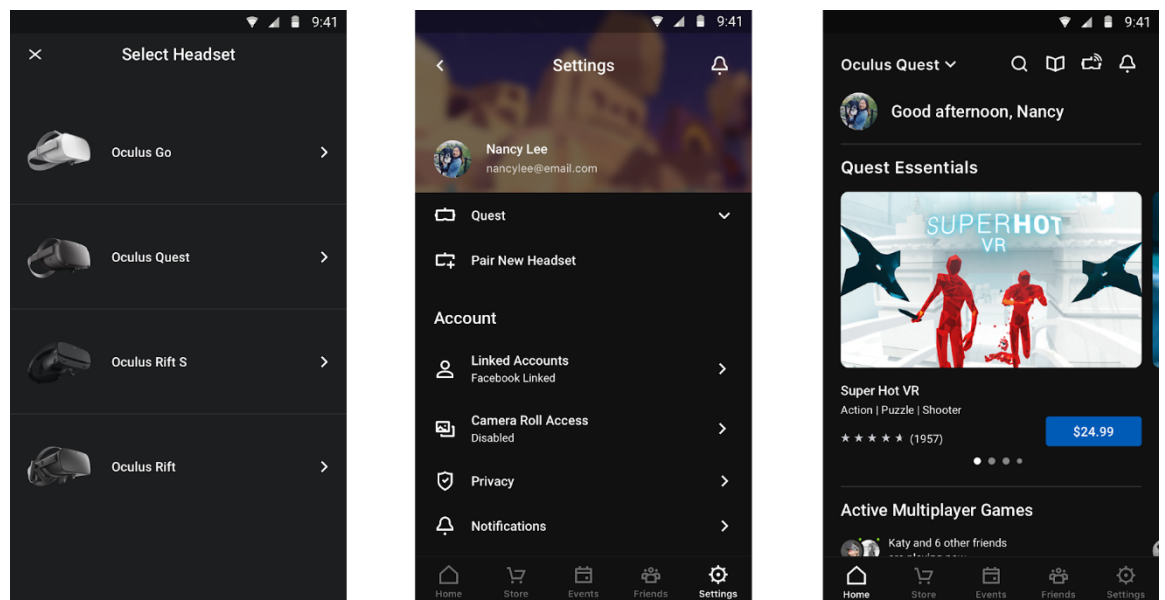
Työlle määriteltiin resursseiksi yksi henkilö tutkimaan ja toteuttamaan visuaalista osuutta sekä kaksi henkilöä tutkimaan sovelluksen olemassa olevaa koodirakennetta ja Input-systeemin muutostarvetta. Minut nimettiin visuaalisen osuuden vastuhenkilöksi.

Alustavasti laskettiin tutkimustyön osuudelle noin kuukauden mittainen aikataulus, jonka jälkeen tarvittavien toteutuksien suunnitteluun ja tekoon varattiin viiden viikon mittainen aikaväli. Aikataulussa otettiin huomioon ennalta-arvaamattomien ongelmien ja täysin uusien asioiden opetteluun tarvittava aika. Prosessin aikana opitulle tiedolle määriteltiin dokumenttipohja, johon kaikki käännöstööhön osallistuvat lisäsivät hyödylliseksi tuntemansa tiedon tai tiedonlähteet.

7 Työtapausten toteutus

7.1 Oculus-ympäristön asennus

Oculus-kehitys vaatii liittymisen Oculusin systeemiin luomalla ilmaisen tilin tai käyttämällä olemassa olevaa Facebook-tiliä. Varsinaisen Quest-laitteen asennuksen ohessa tili liitetään puhelimelle tai muulle mobiililaitteelle ladattavaan Oculus-sovellukseen (kuva 11), jonka kautta sekä tilin että Quest-laitteen asetuksia voidaan hallita. Puhelinsovelluksen kautta tulee myös asettaa Quest-laite ”Developer” eli kehittäjätilaan.



Kuva 11. Oculus-mobiilisovellus.

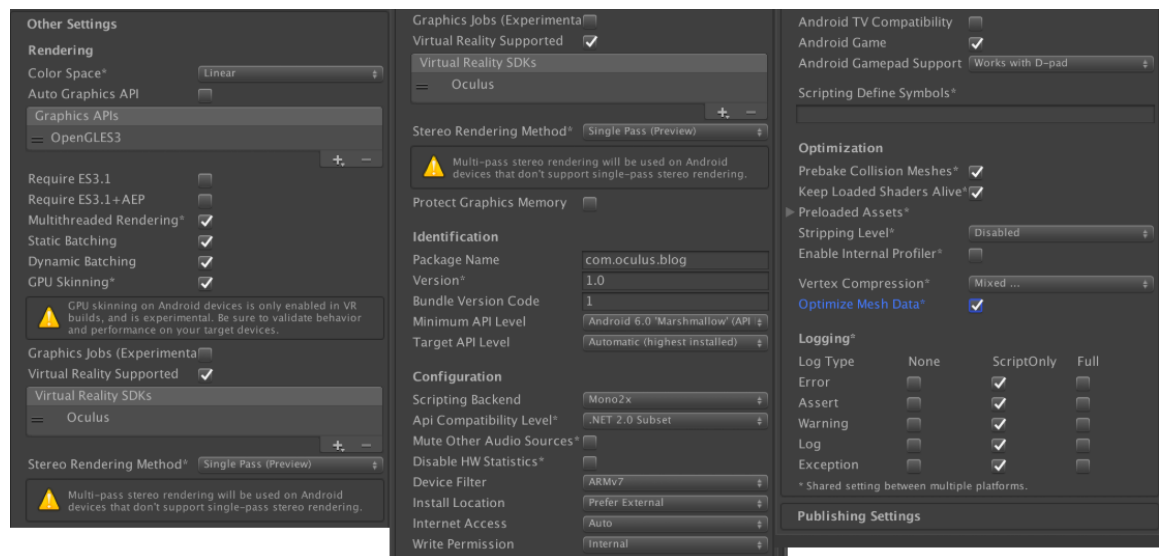
Unityssä Quest-kehityksen aloittamiseksi tarvitaan Unityn oma Android-paketti, jonka voi liittää olevassa olemaan pelimoottorin asennukseen sen aloitusvalikosta. Kun Android-ympäristö on valmiina, lisäämään Unityn Package Manager palvelun kautta moottoriin lisätoimintona Oculusin tuottama ”Oculus Integration”. Quest-laitteen liittämiseksi kehitystietokoneeseen tarvitaan myös Android Debug Bridge-niminen ohjelma sekä sen tarvitsemat Windows-ajurit. Nämä löytyvät muun muassa Oculusin kehittäjä sivustoilta ilmaiseksi ladattavina.

Android Debug Bridge eli ADB on Android-laitteiden liittämiseen PC-ympäristöön tarkoitettu ohjelmisto, joka toimii Windowsin komentorivin kautta. Sen avulla voidaan esimerkiksi asentaa tai poistaa kehitettyjä sovelluksia Quest-laitteelta.

7.2 Unity Oculus- asetukset

Unityssä on rajoitetusti varsinaisesti VR:ään suoraan liittyviä asetuksia, mutta tärkeimpänä niistä on VR-laitteiden tukemisen asetus (kuva 12). Tämä löytyy seuraamalla Unityn Project-asetuksissa polkua Player Settings -> XR Settings -> Oculus. Unity listaa tämän asetuksen alle kaikki tukemansa VR-alustat, mutta Oculus-kehityksen aikana on tärkeää, että Oculus SDK on ainut valittu alusta. Saman Player Settings -> Other Settings polun alta löytyy myös joitakin muita tärkeitä asetuksia liittyen Quest-kehitykseen.

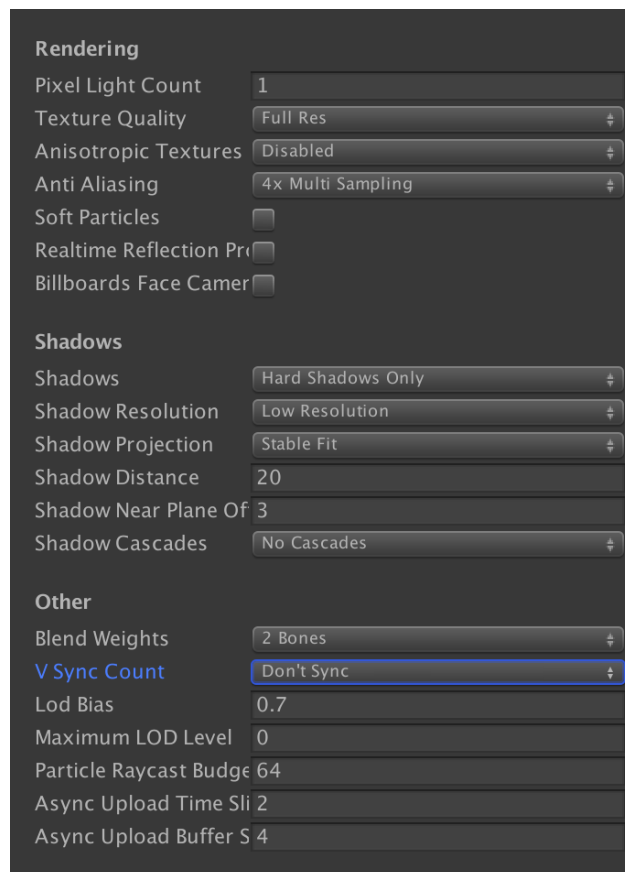
Yksi tällä hetkellä tärkeimmistä asetuksista on Unityn Vulkan-grafiikka API:iin liittyvä asetus, sillä Unity ei tällä hetkellä vielä tue sitä VR-kehityksen osalta ja asetus voi aiheuttaa hankalasti diagnosoitavia virheilmoituksia. Vulkan-tuen poistaminen "Graphics API"-asetuksista on siis ainakin tämän työn kirjoitushetkellä suositeltavaa kehitettäessä Quest-sovellusta, mutta tuen tilanne tulisi tarkistaa myöhemmin suoritettavan kehityksen aikana uudestaan.



Kuva 12. Unity Project-asetuspaneeli.

Unityn Project-asetuksista löytyy myös Quality (kuva 13) eli grafiikan prosessoinnin yleiseen tasoon liittyvä paneeli, johon voidaan nimetä ja määritellä erilaisia asetuskokoelmia. Nämä kokoelmat ovat hyödyllisiä kehitettäessä samaan aikaan sekä PC että mobiiliversiota sovelluksesta, sillä ne antavat mahdollisuuden nopeasti tehdä tarvittavia laatumuutoksia riippuen testattavasta alustasta. Näiden laatuasetusten tarkat halutut arvot riippuvat niin vahvasti kunkin sovelluksen sisällöstä, ettei niitä varten voida antaa yleisiä ohjeita, mutta pyrin listaamaan ne asetukset, joiden muutoksilla oli suurin vaikutus sovelluksemme Quest-versioon.

Yksi tärkeimmistä asetuksista on Stereo Rendering Mode, eli miten Unityn grafiikkamoottori hallitsee kuvan piirtämisen kahdelle eri silmälle. Questille kehitettäessä tämän tulisi olla asetettuna Single Pass vaihtoehdolle, jolloin Unity laskee piirrännän ensin yhdelle silmälle ja käyttää peilaa-mismatematiikkaa kääntääkseen saman kuvan toiselle silmälle laskematta sitä uudestaan.



Kuva 13. Unity Quality-asetuspaneeli.

Scripting Backend -asetus tulisi asettaa käytettäväksi tarkoitetuissa sovellusversioissa IL2CPP-vaihtoehdolle, sillä se nopeuttaa hieman koodin suoritusta. Mono2x-vaihtoehto on hitaampi, mutta sitä voidaan käyttää iteraatio- ja testausvaiheissa, sillä se lyhentää Unityn sovelluksen valmistusaikaa.

Riippuen sovelluksen sisällöstä asetukset Prebake Collision Meshes, Keep Loaded Shaders Alive ja Optimize Mesh Data voivat nopeuttaa toimintaa hieman, mutta ne suurentavat sovelluksen asennuskokoa. V Sync Count -asetus tulee asettaa Don't Sync -vaihtoehtoon, sillä Quest tekee toiminnon itse ilman Unityn apua.

7.3 Input-systeemin muutokset

Input-systeemillä viitataan siihen, miten Unity ja sovelluksessa käytetty koodi tulkitsee laitteistolta saatuja komentoja. Lähtökohtaisesti sovelluksen Input-systeemi käytti HTC Vive -kehityksessä usein käytettävää OpenVR-ratkaisua. Oculuksen OVR-paketti tuo mukanaan oman ratkaisunsa, mutta kumpikaan näistä ei toimi suoraan molemmille alustoille. Tämän takia päädyttiin toteuttamaan ratkaisu, joka toimii molemmilla alustoilla ja johon voidaan myöhemmin lisätä tuki uusille laitteille.

Toteutus tapahtuu lisäämällä koodiin välitaso, joka tarkistaa, mikä alusta on kullakin hetkellä käytössä, käsittelee laitteilta vastaanotetut komennot ja jakaa ne määritetyillä tavoilla koodia kutsuville komponenteille. Toteutus ei ole erityisen monimutkainen, mutta se vaatii kehittäjältä hyvän käsityksen siitä, millaisia vuorovaikutustapoja halutaan tukea. Itse tehty ratkaisu antaa myös mahdollisuuden määritellä tarkemmin vuorovaikutusten parametrit.

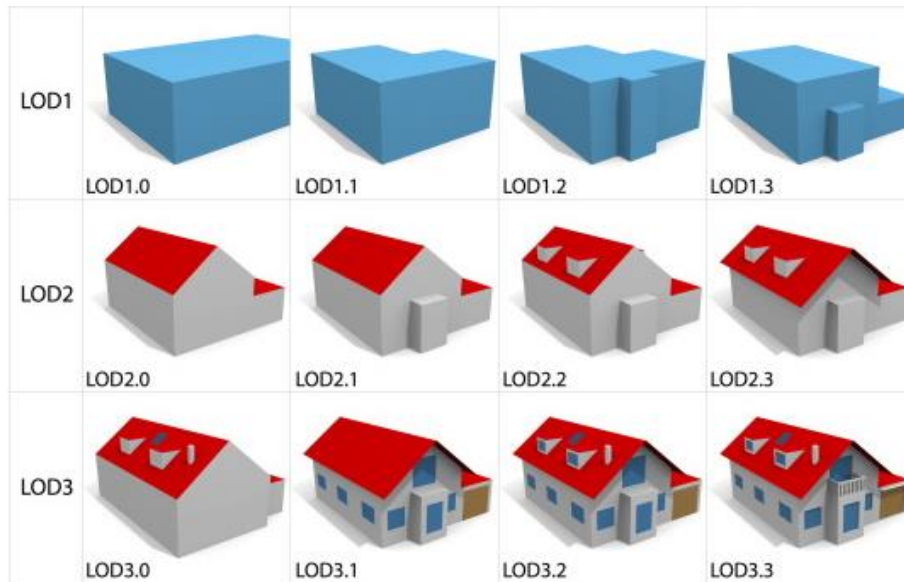
7.4 Muutokset visuaaliseen sisältöön

Unity käyttää materiaalisysteemiä pintojen piirtämisasetusten määrittelyyn. Kehittäjä voi muuttaa materiaalien asetuksia saadakseen aikaan halutun visuaalisen ilmeen. Jokainen näytölle piirrettävä materiaali kuitenkin käyttää tietyn määrän prosessorin laskemisaikaa riippuen monimutkaisuudestaan.

Quest-käännöstä valmisteltaessa huomattiin, että osassa sovelluksen elementeistä käytettiin erittäin suurta määrää eri materiaaleja. Tämä ei aiheuttanut ongelmaa PC:llä, mutta useiden materiaalien aiheuttama objektien uudelleen piirtäminen hidasti sovellusta Questilla suoritettaessa huomattavasti. 3D-malleista luotiin Questia varten uudet versiot, joista karsittiin pois liiat materiaalit, ilman että mallin ulkonäkö muuttui merkittävästi.

3D-mallien kohdalla jouduttiin myös tarkistamaan vertiisien määrää. Vertiisi on 3D-mallinnuksessa käytetty termi, joka tarkoittaa mallin muotoa määrittelevää pistettä. Mitä enemmän vertiisejä mallissa on, sitä tarkempi mallin muoto on, mutta vertiisien määrä lisää myös tarvittavan laskennan määrää. Osa käännökseen liittyvästä testaustyöstä liittyi Questilla suoritettavan sovelluksen yksittäisen laskentakerran maksimivertiisimäärän määrittelyyn. Oculuksen suosituksen mukaan tulisi kerrallaan Questilla piirtää korkeintaan noin 300000 kolmiota eli vertiisien muodostamaa muotoa.

Sovelluksen PC-versiossa raskaimmissa käytetyistä 3D-malleista oli jopa 1,5 miljoonaa kolmiota. Tämän takia päätettiin luoda malleista LOD eli Level of Detail -versioita (kuva 14). Näissä versioissa Unityä varten luodaan yksinkertaistettuja malleja, joita Unity automaattisesti vaihtaa piirrettäväksi riippuen objektin etäisyydestä kameran näkökulmaan. Näin kauempana olevat 3D-mallit ovat vielä tunnistettavia, mutta vaativat vähemmän laskentatehoa.



Kuva 14. Esimerkki LOD-malleista.

7.5 Sovelluksen testaus Questilla

Kun halutaan asentaa kehitysasteella oleva sovellus Questille, tulee se ensin "Builddata" eli muuttaa varsinaiseen sovellusmuotoon. Tämä tapahtuu muuttamalla Unityn Build Settings-asetuksista Platform kohta Androidiksi ja valitsemalla Build. Tämä luo valittuun kohteeseen .apk muotoisen tiedoston, joka voidaan asentaa Questille käyttäen ADB-ohjelman "Install"-komentoa ja antaen oikean .apk:n lähdeosoitteen. Sovelluksen asennettu versio tulee poistaa Questilta ennen kuin uusi versio voidaan asentaa. Tämä toimenpide voidaan toteuttaa käyttäen "Uninstall"-ADB-komentoa ja antamalla sovelluksen Android paketti-nimen, joka määritellään Unityn Project -asetuksissa.

Toteutuksen aikana käytettiin enimmäkseen Oculuksen OVR Metrics toolia, jonka avulla kirjattiin keskimääräisiä virkistystaajuuden nopeuksia, jotka kirjattiin keskeisten asetusmuutosten kanssa ylös vertailua varten. Työkalusta saatiin myös tarkempaa tietoa siitä, mikä alustan komponenteista oli heikon suorituskyvyn aiheuttaja.

Ongelmana testauksessa oli sovellusten siirtäminen PC:ltä Questille, sillä kun pienikin muutos tehtiin, tuli koko ohjelmisto koota uudestaan, poistaa edellinen versio Questilta ja asentaa se uudestaan ennen kuin uusi testi voitiin suorittaa. Oculus on ilmoittanut antavansa kehittäjille ja muille käyttäjille mahdollisuuden yhdistää Quest-laitteensa myöhemmin tänä vuonna USB-johtoa käyttäen PC-alustaan varsinaista suoraa sovelluskäyttöä varten. Tähän asti USB-yhdistäminen on toiminut vain kehityskäytössä ja silloinkin rajoitetusti.

8 Työtapausten tulokset

Toiminnallisuudeltaan sovelluskäännös onnistui täysin, eikä mitään käytössä ollutta komponenttia tarvinnut poistaa Quest-versiosta. Suorituskyvyn tavoitteisiin pääsemiseksi tehtiin paljon töitä, ja suurimmaksi osaksi visuaaliset muutokset pystyttiin tekemään tyydyttävällä tasolla. Työn aikana saatiin myös hyvä kuva Questin kyvykkyydestä ja pystyttiin luomaan dokumentaatio tulevia mahdollisia käännöstöitä varten.

Input-systeemin muutokset antoivat hyvän pohjan uusien vuorovaikutusten suunnitteluun ja kuvan siitä, miten tulevia alustoja varten tulisi valmistautua. Itse ohjaimiin liittyvän koodipohjan tutkiskelu myös antoi hyvää tuntumaa laitteiston varsinaisiin toimintatapoihin, ja sai tiimin keskustelemaan siitä, mitkä vuorovaikutustavoista olivat oikeasti merkityksellisimpiä.

Käännöstyö toi esiin ongelmia 3D-mallien käsittelyssä ja prosessissa, jolla malleja tuodaan tiimin ulkopuolisista lähteistä sovellukseen. Tämä kuitenkin saattoi tiimin suunnittelemaan uuden pipeline-järjestelmän tarpeita ja toteutuksen, joka tehosti työtä kaikilla alustoilla. Mallien kanssa työskentely myös antoi minulle mahdollisuuden harjoitella lisää Blender-ohjelman käyttöä, ja useiden suoritettujen muokkausten jälkeen olen huomattavasti itsevarmempi 3D-mallien työskentelyssä.

Työn perusteella käännöksen tärkeimpiä kohtia olivat Unity-materiaalien yksinkertaistaminen ja karsiminen, 3D-mallien LOD-mallien luonti ja yleinen ympäristöjen visuaalinen yksinkertaistaminen. Myös yksinkertaisilta tuntuvat asiat kuten Unityn automaattisesti luomat ”Standard” shaderit eli materiaalien piirrosta käytetyt ohjelmat huomattiin yllättävän raskaiksi mobiilialustalla. Projektissa käytettävien shadereiden vaihto kevyempiin versioihin kuten Unityn valmiisiin Mobile-tyyppin shadereihin vaikuttaa pieneltä optimisaatioitoimelta, mutta riippuen sovelluksen ympäristön monimutkaisuudesta tämäkin toimenpide voi parantaa suorituskykyä huomattavasti.

Yksi tärkeimmistä työn aikana opituista asioista on testauksen suorittaminen varsinaisilla käyttäjillä. Kehittäjät usein keskittyvät liiankin pitkäjänteisesti ja itsepäisesti yksittäisten komponenttien kehitykseen tai esimerkiksi pieniin optimisaatioihin. Nämä kehitysalueet ovat tietysti tärkeitä, mutta kun sovellus annetaan testattavaksi käyttäjille, jotka eivät välttämättä tunne kehityksessä ilmenneitä ongelmia, he usein keskittyvät varsinaiseen sisältöön ja käyttötarkoituksiin huomatta optimisaatio-ongelmia. Tällä on tietysti rajansa ja huonon suorituskyvyn ei ikinä pitäisi antaa tulla sovelluksen käyttötarkoituksen tai mukavuuden edelle. Tehdyn kehityksen ja sen aikana

kohdattujen ongelmien pohjalta luotiin myös kattava ohjeistus Oculus-kehityksen aloittamista varten ja tätä dokumentaatiota on jo käytetty uusien tiimin jäsenten koulutukseen hyvin tuloksin.

9 Pohdinta työn suorituksesta

Työ oli aiheen opiskelun kannalta erinomainen ja antoi syyn hakea tietoa laajasti. Tutkimuksen ja toteutuksen suhde oli myös sopiva ja niiden välissä tehty suunnitteluosio loi hyvän linkityksen teoriapohjan ja käytännön tekemisen välille. Opin myös paljon eri prosesseista ja niiden hallinnasta sekä eri tavoista tehokkaasti dokumentoida tutkittua tietoa muuta tiimiä tai laajempaa yleisöä varten. Koska työhön liittyi myös tiimin ulkopuolelta tulevan tiedon ja materiaalin käsittelyä, antoi se mahdollisuuden oppia kommunikaatitaitoja ison yrityksen sisällä toimimista varten.

Ongelmakohtina olivat tiettyssä määrin oman ajankäytön hallinta varsinkin optimisaatiovaiheessa, sillä visuaalisia elementtejä olisi voinut hioa lähes loputtomasti ilman varsinaista huomattavaa hyötyä. Sen rajan hahmottaminen, missä jokin komponentti tai osa on tarpeeksi hyvä, oli hankalaa.

Yleisesti olen suoritettuun työhön tyytyväinen, ja se oli mielestäni arvokasta sekä itselleni että koko tiimille ja projektille. Saamani palautteen mukaan myös asiakas eli muu tiimi oli työhön tyytyväinen. Kokonaisuutena se antoi hyvän pohjan sekä tulevalle VR-kehitykselle että yleisesti alan töissä toimimiselle. Opinnäytteen tekoon ja tämän dokumentin kirjoittamisen kannalta olen aiheen valintaan ja sitä varten tehtyyn tutkimustyöhön tyytyväinen.

Lähteet

- 1 Joe Bardi. (2019). Marxent Labs. Saatavilla 17.11.2019.
<https://www.marxentlabs.com/what-is-virtual-reality/>
- 2 Virtual Reality Society. (2015). VRS. Saatavilla 17.11.2019.
<https://www.vrs.org.uk/virtual-reality/history.html>
- 3 Sophie Thompson. (2019). Virtual Speech. Saatavilla 17.11.2019.
<https://virtualspeech.com/blog/vr-applications>
- 4 Andrew Wheeler. (2016). Engineering.com. Saatavilla 17.11.2019
<https://www.engineering.com/Hardware/ArticleID/12699/Understanding-Virtual-Reality-Headsets.aspx>
- 5 Kyle Orland. (2013). Ars Technica. Saatavilla 17.11.2019.
<https://arstechnica.com/gaming/2013/01/how-fast-does-virtual-reality-have-to-be-to-look-like-actual-reality/>
- 6 Gadget Flow. (2018). Virtual Reality Pop. Saatavilla 17.11.2019.
<https://virtualrealitypop.com/will-vr-change-the-world-of-gaming-d8d0e56d3092>
- 7 D. Freeman, S. Reeve, A. Robinson, A. Ehlers, D. Clark, B. Spanlang, M. Slater. (2017).
<https://www.cambridge.org/core/journals/psychological-medicine/article/virtual-reality-in-the-assessment-understanding-and-treatment-of-mental-health-disorders/A786FC699B11F6A4BB02B6F99DC20237>
- 8 Shanhong Liu. (2019). Statista. Saatavilla 17.11.2019.
<https://www.statista.com/statistics/755645/global-vr-device-market-share-by-vendor/>
- 9 Kei Studios. (2019). Kei-Studios. Saatavilla 17.11.2019.
<https://kei-studios.com/wireless-vr-headset-the-next-generation/>

10 Varjo PR. (2019). Varjo. Saatavilla 17.11.2019.

<https://varjo.com/press-release/varjo-announces-two-new-human-eye-resolution-hmds-for-professional-vr-incorporating-support-for-steamvr-content-at-industrys-highest-fidelity-and-integrated-hand-tracking/>

11 Nick Statt. (2016). The Verge. Saatavilla 17.11.2019.

<https://www.theverge.com/2016/10/13/13261342/virtual-reality-oculus-rift-touch-lone-echo-robo-recall>

12 Essinger, K. (2018). A comparison of interaction models in Virtual Reality using the HTC Vive.

<http://bth.diva-portal.org/smash/get/diva2:1260974/FULLTEXT02.pdf>

13 Unity. (2018). Unity. Saatavilla 17.11.2019.

<https://web.archive.org/web/20181107004229/https://unity3d.com/public-relations>

14 Oculus developer blog. (2019). Oculus. Saatavilla 17.11.2019.

<https://developer.oculus.com/documentation/quest/latest/concepts/book-unity-gsg/>

15 Elena Sotiri. (2017). PC Steps. Saatavilla 17.11.2019.

<https://www.pcsteps.com/14266-video-game-porting-console-to-pc-ports-suck/>

16 Unity User Manual. (2019). Unity. Saatavilla 17.11.2019.

<https://docs.unity3d.com/Manual/CrossPlatformConsiderations.html>

17 Michael J. Garbade. (2018). Codeburst. Saatavilla 17.11.2019.

<https://codeburst.io/native-vs-cross-platform-app-development-pros-and-cons-49f397bb38ac>

18 Iryna Pototska. (2017). Upwork. Saatavilla 17.11.2019.

<https://www.upwork.com/hiring/for-clients/cross-platform-development-with-react-native/>

19 Fredrik Tall. (2017). Luleå University.

<http://ltu.diva-portal.org/smash/get/diva2:1112571/FULLTEXT01.pdf>

20 Raevenlord. (2017). TechPowerUp. Saatavilla 17.11.2019.

<https://www.techpowerup.com/231129/on-nvidias-tile-based-rendering>

21 Oculus developer blog. (2019). Oculus. Saatavilla 17.11.2019.

<https://developer.oculus.com/documentation/quest/latest/concepts/book-perfanalysis/>

22 Oculus developer blog. (2019). Oculus. Saatavilla 17.11.2019.

<https://developer.oculus.com/documentation/quest/latest/concepts/mobile-ovrmetricstool/>

23 Oculus developer blog. (2019). Oculus. Saatavilla 17.11.2019.

<https://developer.oculus.com/documentation/unity/latest/concepts/unity-profiler-panel/>

Kuvat

Kuva 1: Oculus Quest. (2019). PC World.

<https://www.pcworld.com/article/3441097/oculus-connect-6-significant-oculus-quest-upgrades-could-render-the-rift-obsolete.html>

Kuva 2: Oculus Rift. (2013). Gizmodo.

<https://gizmodo.com/whats-inside-the-oculus-rift-virtual-reality-headset-5994263>

Kuva 3: Vive Cosmos. (2019). SlashGear.

<https://www.slashgear.com/htc-vive-pro-eye-cosmos-dev-kit-infinity-subscription-expand-vr-ecosystem-07560688/>

Kuva 4: Beatsaber. (2018). Destructoid.

<https://www.destructoid.com/rhythm-slice-em-up-beat-saber-hits-playstation-vr-this-month-530523.phtml>

Kuva 5: Google Tiltbrush. (2016) Google Youtube kanava.

<https://www.youtube.com/watch?v=TckqNdrdbgk>

Kuva 6: Varjo. (2019). VarjoVR.

<https://varjo.com/press-release/varjo-announces-two-new-human-eye-resolution-hmds-for-professional-vr-incorporating-support-for-steamvr-content-at-industrys-highest-fidelity-and-integrated-hand-tracking/>

Kuva 7: Unity SteamVR. (2019). itse otettu kuvankaappaus.

Kuva 8: Alustojen Ohjaimet. (2015). Metanaut VR.

<https://www.oculus-pour-porte.html>

Kuva 9: RenderDoc. (2019). Renderdoc.

<https://renderdoc.org/>

Kuva 10: Globe logo. (2019). Nokia Globe, kuva itse tuotettu.

Kuva 11: Oculus mobiili. (2019). Itse otettu kuvankaappaus.

Kuva 12: Unity Project. (2019). Itse otettu kuvankaappaus.

Kuva 13: Unity Quality. (2019). Itse otettu kuvankaappaus.

Kuva 14: LOD-mallit. (2019). Itse otettu kuvankaappaus.