

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2023

Stefan Frunza

PC VR -sovelluksen muunnos kannettavalle VR -laitteelle



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikka

2023 | 24 sivua

Stefan Frunza

PC VR -sovelluksen muunnos kannettavalle VR -laitteelle

Opinnäytetyön tavoitteena oli toteuttaa terveysalan projektin kohtauskääntötyö kannettavalle VR -laitteelle. Tämän opinnäytetyön toimeksiantajana on Turun ammattikorkeakoulun tutkimusryhmä Tulevaisuuden Interaktiiviset Teknologiat. VR -laitteet tunnetusti sisältävät teholtaan alhaisempaa laitteistoa, ja siksi tutkimuskysymyksenä oli, kuinka paljon alkuperäistä käyttäjäkokemusta voisi säilyttää käännetyllä alustalla. Sen lisäksi sekundaarisena tutkimuskysymyksenä oli tutkittu, millä työkaluilla optimoinnin kautta kokemuksen säilytys olisi mahdollista toteuttaa.

Työn aluksi tutkittiin projektin pelattavuutta ja suorituskykyä. Sitten kokeiltiin eri optimointiyhdistelmiä ja käyttäjäkokemusprosesseja. Niiden tulokset mitattiin piirtokutsujen ja kuvataajuden kautta. Näin varmistettiin, että laitteiston potentiaalia maksimoitiin samalla ja vältettiin pullonkaulausten syntymistä tai alhaista visuaalisen kokemuksen laatua, sillä ne voivat tuottaa terveyshaittoja.

Tuloksena oli, että projektia saatiin onnistuneesti hieman optimoitua piilopintojen poiston avulla. Laajuudeltaan projektilla oli rajoitettu lähestymistapa kannettavan VR -laitteen optimoinnissa. Yleensä kääntötyöhön vaaditaan kokonaisia kehitysryhmiä, joissa jokainen henkilö on erikoistunut omaan alueeseensa sen sisällä ja kääntötyössä tällainen haaste vaatisi projektin ominaisuuksien uudelleen suunnittelua tai jopa kokonaisien projektiosoiden poistoa.

Asiasanat:

Virtuaalinen todellisuus, optimointi, kääntötyö, alusta

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Technology

2023 | 24 pages

Stefan Frunza

PC VR -application conversion to a portable VR - device

The aim of this thesis was to fulfill a healthcare project scene conversion to a portable VR headset, which was commissioned by the research group Futuristic Interactive Technologies of Turku University of Applied Sciences. Since VR devices generally consist of lower performance hardware the primary question was to determine how much of the original user experience can be converted to another platform and as the secondary question, what the tools can be used to enable this conversion through optimization.

To achieve the objectives, the conversion had to first be examined from the gameplay and performance perspective. After which by applying different combinations of optimization and user experience processes, the conversion was measured through draw calls and frames per second ensuring that the VR - headset resources were used to the maximum without bottlenecking issues or low-quality visual experience which could cause a health hazard to the user.

The conclusion was that although the conversion could be slightly optimized mainly through occlusion culling, its scope limits the approach to portable VR - optimization. The reason for this limitation is that such a conversion usually requires complete teams with each specializing in their own area and re-designing or even sacrificing features that heavily impact performance.

Keywords:

Virtual reality, conversion, optimization, platform

Sisältö

Käytetyt lyhenteet ja sanasto	6
1 Johdanto	7
2 Virtuaalinen todellisuus	9
3 Kääntötyön filosofia ja käytäntö	10
3.1 Rajoitukset	11
3.2 Ohjelmistoratkaisuja	11
3.3 Kääntötyön työkalu	13
4 Alustoiden tutkinta	14
4.1 Lähtöalusta	14
4.2 Päätelaitteiston tarpeet	15
4.3 Tavoite	16
5 Käännöstyö	17
5.1 Mittaukset ja testaukset	17
5.2 Lopputulos	21
6 Yhteenveto ja johtopäätökset	22
6.1 Skaala	22
6.2 Resurssit	22
Lähteet	24

Kuvat

Kuva 1. Oculus Performance Tracker OVR Metrics Tool Oculus Quest 2:n ympäristössä	13
Kuva 2. Vasemmalla piilopintojen poiston tulos kun kääntyy poispäin. Oikealla piilopintojen poiston tulos kun katsoo hahmoa suoraan kohti.	19
Kuva 3. Vasemmalla ilman reunojen pehmenystä. Oikealla reunojen pehmentämisellä.	20

Taulukot

Taulukko 1. Kohtauksen suoritus tietokoneen ja Oculus Quest 2:n alustalla	17
Taulukko 2. Piilopintojen poiston käyttö kohtauksessa	18
Taulukko 3. Reunojen pehmenys ilman ja PP eli piilopintojen poiston kanssa.	20

Käytetyt lyhenteet ja sanasto

FFR	Tekniikka sivualueen alhaisemmalle kuvantamiselle kuin näytön keskellä.
FPS	Frames Per Second, kuvataajuus.
Kuvantaminen	Pikselien luonti.
Leipominen	Etukäteen simuloitu tai kuvannettu data, joka sitten tallennetaan muistiin.
Piilopintojen poisto	Prosessi, jossa asioita, joita ei näy näytöllä, jätetään kuvantamatta säästääkseen näytönohjaimen tehoa.
Piirtokutsu	Käskeytys prosessorilta näytönohjaimelle datan kuvantamiselle.
Pullonkaulaus	Ilmiö, missä kuvantamisen prosessi hidastuu, koska yksi komponentti on täydellä teholla, ja sen osion suoritusta joudutaan odottamaan ennen kuin prosessia voi jatkaa.
Pikseli	Kuvapiste näytöllä.
RAM	Random Access Memory, välimuisti.
Ylipiirto	Piirtokutsujen päällekkäistä piirtämistä.
VRAM	Volatile RAM.
VR	Virtual Reality.

1 Johdanto

Turun ammattikorkeakoulun tutkimusryhmä Tulevaisuuden interaktiiviset teknologiat on osallistunut kehittämään kansainvälistä Safe4Child-projektia. Aiheena on aggressionhallintakoulutusta hoitajille, aggressiivisten lapsien vuorovaikutuksessa. Laajuudeltaan projekti jakaantuu neljään eri vaiheeseen, ja sen kolmas vaihe sisälsi virtuaalisen todellisuuden ympäristön luontia (Turku AMK 2022).

Viimeisen kymmenen vuoden aikana virtuaalista todellisuutta on alettu harkitsemaan todellisena vaihtoehtona interaktiivisen teknologian maailmassa sekä tutkimusalueena. Laitteiston edistymisen kautta virtuaalisen todellisuuden simuloinnin hyödyllisyys nousi ja sen käyttötarkoitus laajeni. Yhdeksi käyttötapaueksi kehittyi pedagogia (Hamad & Jia 2022). Opetusmateriaalin simulointi toi uusia vahvuuksia perinteiseen opettamiseen. Käyttäjä kykeni esimerkiksi harjoittelemaan kokemalla tilanteita, joita oli vain kuvailtu teksteissä tai näytelty videoissa.

Safe4Child-projekti halusi käyttää näitä kokemisen ja materiaalin lähestymisen vahvuuksia opettamalla projektissa kehitettyä opetusmateriaalia simulaation kautta. Tarkoituksena oli siis kääntää hoitajaopiskelijoille suunniteltua realistista simulaatio-opetusmateriaalia laitteistolle toimivaksi. Parhaiten se toteutuisi laitteessa sovelluksena, koska itsenäisenä, laite ei vaatisi muuta kuin pelaajan, laitteiston käyttöönoton ja sovelluksen käynnistämisen valikosta. Materiaalista siis tulisi kannettava, ja sitä voitaisiin skaalata opettamaan jopa kokonaisia luokkia kerralla.

Sovelluksen toteuttamiseen, täytyi projekti ensin muuttaa kääntötyönä laitteiston alustalle toimivaksi. Lisäksi, jotta kääntötyön voitiin ajatella onnistuneen, täytyi varmistaa, että käyttäjäkokemus säilyisi optimoinnin kautta. Laitteistoksi valittiin Oculus Quest 2, koska se oli projektin kehitysvaiheessa johtavin tuote markkinoilla, jolloin se myös tarjosi laajimman alustan kehittäjille ja heidän tuotteilleen (Raj 2023).

Tavotteisiin päästiin tutkimalla, millä tekniikoilla projektia voitiin optimoida laitteelle sopivaksi, ja kuinka lähelle alkuperäistä käyttäjäkokemusta voitiin päästä näillä työkaluilla.

2 Virtuaalinen todellisuus

Virtuaalinen todellisuus (tästä lähin VR eli virtual reality), on tietokoneen luoma kolmiulotteinen ympäristö, joka on olemassa reaaliajassa. Tarkoitus on, että ihminen pystyisi osallistumaan vuorovaikutuksessa ympäristön kanssa esim. tietokoneen kautta (Vince 2004).

Alalle oli myös luotu oma laitteistonsa, jotta vuorovaikutus olisi lähempänä todellisen tuntuista. VR -lasien tarkoitus on heijastaa kuvannettua näyttöä ihmisen koko näkökentälle. Sen lisäksi lasien mukana tuli VR -ohjaimet, jotka toimivat virtuaalisina käsinä luoden näin käyttäjälle syvennetyn virtuaalisen kokemuksen. Teknologia on keksitty 1960-luvulla ja noin 2010-luvulle asti sitä pidettiin markkinointitemppuna. Tämä näkemys kuitenkin alkoi muuttumaan 2012, kun ensimmäinen Oculus-prototyyppi julkaistiin (Hamad & Jia 2022). Laitteisto alkoi pikkuhiljaa olemaan tarpeeksi tehokasta, jotta se voisi kompaktisti suorittaa simulaatioita. Näin yhtiöitä kannustettiin kehittämään omia VR -laitteitaan ja VR:stä tuli menestykseäs ala (Hamad & Jia 2022).

Alun perin VR oli käytössä vain videopelialalla, ja siksi teknologia parantui pelkästään alan tarpeiden mukaan, mutta nykyään VR -laitteistolle löytyy laajempaa hyötykäyttöä. Se on jo sellaisella tasolla, että sillä pystytään simuloimaan todellisen tuntuista ympäristöjä. Näin VR -laitteistoa voidaan hyödyntää uusissa käyttötarkoituksissa, kuten opetuksessa, harjoittamisessa tai valmistautumisessa. (Hamad & Jia 2022.)

3 Kääntötyön filosofia ja käytäntö

Kääntötyö (tunnetaan myös nimellä “porttaaminen”) perustuu projektin kokonaisuuden siirtämiseen yhdeltä alustalta toiselle (Starloop Studios 2023). Ideaalisesti toteutus tapahtuisi yksi suhde yhteen, mutta realistisesti ja pragmaattisesti se ei ole mahdollista. Se ei kumminkaan tarkoita, ettei siihen yritettäisi pyrkiä. Tavoitteena on siis säilyttää käyttäjäkokemusta mahdollisimman paljon kohdealustalla ja muita tärkeitä tekijöitä tähän tehtävään ovat esim. ohjainkartoitus ja saman tason suorituskyky (Denisyuk 2021).

Pelien perspektiivistä tuote itsessään ei saa tuntua eri peliltä vaan lisäykseltä tai pidennykseltä alkuperäisestä mediasta, kuten esim. kirjan kääntäminen elokuvaksi. Eli materiaali pysyy, mutta sen ilmaiseminen voi hieman muuttua. Esimerkiksi siirtämällä PC-pelin konsolille, pelaajalla ei ole käytössä näppäimistöä tai hiirtä vaan konsoliohjain, jolloin se luo tarvetta syötteen kartoituksen uudelleen konfigurointiin. Kääntötyön tarve siis yleensä onkin tuotteen saatavuuden laajentamiseen. Peli, joka on saatavana useammalla alustalla varmistaa enemmän myyntiä ja huomiota (Denisyuk 2021). Henkilö, joka ei omista konsolia voisi ostaa saman pelin mobiiliversiona, sillä melkein kaikki omistavat älypuhelimien nykyään.

VR -pelien maailmassa se tuo enemmän haasteita. Perinteisissä alustoissa huono tai epäonnistunut kääntötyö haittaa vain käyttäjäkokemusta ja voi tehdä tuotteesta epäkäytännöllisen, mutta VR:ssä se voi myös tuottaa terveys- ja turvallisuushaittoja. Päänsäryn ja pahoinvoinnin riskit ovat liitettynä huonoon suorituskykyyn (Mason 2022). Tämän välttämiseen on keksitty muutama metodi, kuten tietokoneen kautta suorittaminen tai VR -laitteen optimointi. Tietokoneen avulla VR -lasit voivat toimia ulostulona ja laskelmat, kuvantaminen sekä muut suoritukset tapahtuisivat tietokoneen kautta ja se lähettäisi vain tarpeelliset tiedot laitteelle. Näin varmistuen, että tehoa riittää raskaankin pelin suorittamiseen eikä kokemus kärsisi. Mutta, VR -lasitkin sisältävät tehokasta koneistoa ja siksi nykyään ne voivat myös suorittaa sovelluksia, ns. mobiilina. Jolloin toinen vaihtoehto olisi optimoida käyttäjäkokemus laitteelle sopivaksi.

3.1 Rajoitukset

Yleensä laitteisto tulee olemaan suurin este kääntötyössä ja myös määrittää valmiiksi työn parametreja. Tietokoneisiin verrattuna konsolit tai mobiililaitteet aina sisältävät vähemmän RAM- sekä VRAM -muistia, näytönohjaimet ja prosessorit eivät ole yhtä tehokkaita ja tallennuslaitteiden kapasiteetti kärsii (Starloop Studios 2023). Kun laitteiston jonkin osan teho on pienempi kuin ohjelmiston tarve, syntyy pullonkaulausta. Eli laitteiston muut osat eivät kykene suorittamaan laskelmiaan täydellä potentiaalilla, sillä puutteellinen komponentti hidastaa koko prosessia (Intel 2023). Seurauksena ohjelman suoritus kärsii ja se ilmenee käyttäjäkokemuksessa.

VR:n houkutus on sen mobiilikäyttö, mutta tehokkaan laitteiston pyörittäminen kompaktissa ja kevyessä muodossa esittää haasteita. Oculus Quest 2 on ollut mullistava tuotos näiden tarpeiden toteuttamisessa. Sisäiset näytöt tukevat 1832 x 1920 resoluutiota per silmä eli yhteensä 3664 x 3840, ja näytönohjain on muistiltaan 6 GB, joka on yleinen alan vaatimus (Starloop Studios 2023). Muistin määrä ei ole kuitenkaan PC-tasolla.

Tuote siis tietokoneversiona pitää optimoida "alaspäin", jotta se suoriutuisi VR -laitteen standardien mukaisesti. VR alana on ollut olemassa jo jonkin aikaan, jolloin ratkaisua on keretty kehittämään ja ovat asettuneet alan käytännöiksi (Hamad & Jia 2022).

3.2 Ohjelmistoratkaisuja

Piirtokutsu on yksi ydinkonsepteista suorituksen lisäämiseen ja loogisesti myös vähentämiseen. Piirtokutsu tarkoittaa prosessorin datan lähettämistä näytönohjaimelle kuvantavaksi ja ilmestyy näytöllä pikseleinä (Jukić 2015). Mitä enemmän piirtokutsuja laite saa, sitä enemmän laskelmallista voimaa se vaatii. Ensimmäisiä askelia kääntötyössä onkin piirtokutsu-kutsujen tarkistus ja tarpeen mukaan vähennys. Sen lisäksi pitää varoa, ettei synny ylipiirtotilannetta eli kuvannettujen pikseleiden päällekkäistä "piirtämistä". Se rasittaa näytönohjainta

ja vie enemmän VRAM -muistia (Bonet 2020b). Jos muisti loppuu kesken suoritusta, syntyy taas pullonkulausta.

Periaatteessa kaikki laskennallinen osuus, jota voidaan poistaa ohjelman suorittamisen aikana laskisi piirtokutsuja, ja seurauksena vähentää suoritustehon käyttöä (Oculus 2023b). Tähän kategoriaan kuuluvat valaistus, piilopintojen poisto ja tekstuurit.

Ohjelmissa valaistus, tarkemmin dynaaminen valaistus, on reaaliajassa simuloitua valoa. Laite jatkuvasti laskee ja simuloi valonsäteitä tuottaakseen realistista valaisua (El-Nasr et al. 2007). Sen voi korvata ns. leipomalla valoja. Simuloimalla valaistusta etukäteen, sen data voidaan leipoa eli tallentaa omalle valokartalle ja ohjelman suorituksessa laite käyttää kyseistä karttaa skenaarion valaisemiseen (Unity 2023b).

Piilopintojen poisto on prosessi, jossa kameran näkökentän ulkopuoliset asiat jätetään kuvantamatta. Esineitä, huoneita, hahmoja jne. on turha kuvantaa, kun niitä ei ole tarpeen huomioida sillä hetkellä. Vasta esim. pelaajan kääntyessä kyseiseen suuntaan, sallii piilopintojen poisto tarvittavien asioiden laskelmallisesti ilmestyä. (Unity 2023a)

Kuvantamisen määrää voi myös vähentää tiivistämisen kautta. Tekstuurit vievät enemmän muistia mitä tarkempia yksityiskohdiltaan ja näin suurempia ovat. Jokainen tekstuuritiedosto myös vaatii oman piirtokutsun. Niitä voi tiivistää luomalla tekstuuri atlaksen eli kuvan, joka sisältää useampia tekstuureja kerrallaan. Laite tunnistaa, mikä tekstuuri otetaan käyttöön atlaksesta x- ja y-koordinaattien avulla. Yhdellä piirtokutsulla voisi siis ladata kaikki tarvittavat tekstuurit (Sciutteri 2016).

VR -laitokohtainen huomio optimoinnin suhteen on kiinteä fovea-kuvantaminen (FRR eli fixed foveated rendering) Laite normaalisti heijastaa linssien kautta käyttäjän silmiin kuvan koko näköalueelle, mutta ihmisen silmänrakenteen takia tärkeimmät reseptorit ovat keskitetty ja siksi emme näe silmän sivuista yhtä hyvin. FFR toimii kuvantamalla kuvan keskiosaa korkeimmalla resoluutiolla ja sen ympäriltä alemmalla resoluutiolla, emmekä huomaa niiden eroja. Tämä tekniikka

skaalautuu sen mukaan, kuinka paljon tehoa kuvannus vie näytönohjaimelta, sillä hetkellä (Oculus 2023a).

3.3 Kääntötyön työkalu

Jotta piirtokutsujen ja muiden suoritukseen liittyvien arvojen määrää voidaan määrittää ja tulkita, on tarvetta ohjelmalle, joka suorittaa mittauksia ja tulostaa tiedot tilastona. Näitä ohjelmia tunnetaan nimellä Performance Tracker (Kuva 1). Piirtokutsujen lisäksi ne mittaavat kuvataajuutta (fps), näytönohjaimen ja prosessorin hyötykäyttöä (utilization), muistin käyttömäärää, laitteen komponenttien lämpötilaa jne. (Oculus 2023c). Niitä tulkiten pystytään puuttumaan puutteisiin ja tilanteen mukaan ottamaan käyttöön optimointitekniikoita.



Kuva 1. Oculus Performance Tracker OVR Metrics Tool Oculus Quest 2:n ympäristössä

4 Alustoiden tutkinta

4.1 Lähtöalusta

Projektia on tuotettu Unity-pelimoottorin kanssa ja ennen kääntötyötä pitää huomioida suoritusta vaativat alueet, joilla on vaikutusta optimointiin eli valaistus, hahmot, pelimekaniikat ja ympäristö.

Skenaario sisältää neljä ei-pelaaja-hahmoa (tästä lähin lyhennetty EPH), joista kaksi ovat tapahtumakutsuilla ja toiset kaksi toistavat heille spesifisiä animaatioita. Tapahtumakutsuttomat EPH:t eivät ole interaktiivisia.

Tapahtumakutsulla tarkoitetaan sitä, että lepotilassa olevaa koodiosuutta ei aloiteta suorittamaan ennen kuin tietynlainen ehto A toteutuu. Tämä on optimoinnin suhteen kannattavaa ja sitä pitäisi toteuttaa aina kuin mahdollista. Toisena vaihtoehtona olisi käyttää ehtolauseita, jotka ovat aktiivisia, koska ne jatkuvasti tarkastaisivat, onko ehto jo täyttynyt. Niitä yleensä vältetään muissa, kuin pakollisissa tilanteissa, sillä sen väärinkäyttö on huonoa käytäntöä ja voi vaikuttaa suorituskykyyn huomattavasti (Khandaker 2021). Tästä voimme päätellä, että taustalapset ovat ensimmäisenä aktiivisuuden hierarkiassa niiden animaatiokutsujen takia. Hahmojen laskelmallinen taakka ei kumminkaan ole huomattava pelimekaniikkojen ansiosta.

Simulaation päämekanismeina toimivat ponnahdusikkunat ja pelaajan hahmon liikuttaminen. Ponnahdusikkunat aktivoituvat tapahtumakutsulla ja pysäyttävät skenaarion muut toiminnat niiden aukiolon aikana. Pelaajan liikkuminen ei myöskään tule vaatimaan paljon tehoja sillä skenaariossa on vain muutama määritelty alue mihin hahmoa voi siirtää. Ei siis tarvitse kuvantaa hahmon liikkumista tai seurata koordinaatioita yhtä tarkasti. Näin sallien kohtauksen ympärille rakentamista ja EPH:t voivat kävellä tiettyjä ratoja sen sijaan, että niille luotaisiin tekoälyä ja kyky liikkua mihin vain.

Viimeisenä on tärkeä katsoa valaistusta. Tässä esimerkkikohtauksessa suurin tehoa vaativa osuus. Sen optimointia oli jo projektin aikana ennakoitu leipomalla

valaistuksen valmiiksi kohtaukseen. Sitä tarvitsee ainoastaan uudelleen simuloida Android-alustalla, sillä asetukset ovat valmiiksi säädettyjä. Tämä siksi, koska yleisesti ei ole kannattavaa tuoda valokarttaa käännöksenä tietokonealustalta Android:lle vaan luoda kokonaan uusiksi Android-alustaa varten.

4.2 Päätelaitteiston tarpeet

Oculus Meta Quest 2:n laitteiston päämääränä oli imitoida tehokkaan tietokoneen suorituskykyä kannettavassa muodossa ja se onnistuu siinä melko hyvin. Näytönohjaimen muistikaista (6 GB) on kirjoituksen aikana keskiarvoltaan suositeltu määrä tietokonepeleille (Starloop Studios 2022). Sen avulla sovellukset pystyvät suoriutumaan, jopa 90 kehysnopeudella (fps), mutta tehdasasetuksena raja on 72. Korkean kuvataajuuden ylläpito oli tärkeää tuottaakseen mahdollisimman tasaisen käyttäjäkokemuksen ja välttääkseen aiheuttamasta päänsärkyä ja pahoinvointia pelaajalle (Mason 2022).

Laitteen teknologia ei ole kumminkaan vielä saavuttanut tietokoneen tasoista tehoa täysin, jolloin sen rajat toimivat parametreina ja sen mukaan täytyy tiivistää. Pitää myös huomioida, että Oculus Quest 2:n käyttöjärjestelmänä toimii Android. Se ei siis käynnistä .exe-muotoisia ohjelmia vaan .apk-tyyppisiä sovelluksia, jolloin alusta ja sen kautta projektin koodi pitää muuttaa laitteelle luettavaksi.

Tekstuurien koon ylärajaksi on asetettu 2K (2048x2048 pikseliä) oletusarvona ja sitä kannattaa noudattaa optimoinnissa, koska se on hyvä välimaasto suorituksen ja laadun kanssa. Ylärajaa ylittävät tekstuurit täytyy siis tiivistää 2K-kokoon. Todellisuudessa mikään ei estä laitetta kuvantamasta, jopa 8K:n (8192x8192 pikseliä) tekstuureja. Laitte sallii sisäisten tehdasasetusten ja reunojen muuttamista tietokoneen kautta kehittäjätilassa ja sieltä voi vaikka tuon kuvataajuuden arvoa tai tekstuurin maksimi sallittu kokoa muuttaa, mutta se itsessään tuottaa muutamia ongelmia.

Ensinnäkin, vain koska se on mahdollista, ei tarkoita, että se olisi toiminnallista. Suoritustehon käyttö nousee moninkertaisesti kuvannetun tekstuurin koosta ja

niiden määrästä eli monia 2K:n tai muutama 8K:n tekstuuria voi luoda samaa pullonkaulausta piirtokutsujen kautta. Parempi vaihtoehto näistä olisi 2K:n kokoiset tekstuurit, koska silloin käytössä oleva kapasiteetti olisi selvästi suurempi ja niitä voisi vain vähentää tai tiivistää tekstuuri atlaksella, korjatakseen ongelman.

Toiseksi, isompi laskelmallinen teho tuottaa enemmän lämpöä ja kuluttaa enemmän sähköä ja, koska käsittelemme kompaktia mobiililaitetta, kumpikaan näistä ei ole toivottuja vaikutuksia. Yksi opinnäytetyön tavoitteista oli nimenomaan skaalattavuus ja se, että laitteissa on suurempi akun kulutus tai huomattava riski simulaation pätkiytymisestä, pahimmillaan jäätymisestä, on vältettävä. Pullonkaulaus sen lisäksi kuluttaa itsessään enemmän sähköä, koska se on kuvantamassa laitteen täydellä teholla.

4.3 Tavoite

Yleisellä tasolla toimeksiantajan tehtävänanto oli tuottaa hyvin suoriutuva VR -sovellus Unity-projektin ensimmäisestä opetusmateriaalikohtauksesta skaalautuvasti.

Mitä käytännössä se tarkoittaa on, että ensiksi kohtaaus täytyy kääntää sovellukseksi. Sovelluksena siksi, jotta sen voi helposti asentaa uudelle tyhjälle Oculus Meta Quest 2:lle ja laitteisto kykenee käynnistämään sen suoraan omasta käyttöjärjestelmästä. Jotta tämä onnistuisi, täytyy sovelluksen olla optimoitu laitteelle sopivaksi, ylläpitääkseen tietokoneversion standardeja suoriutumisessa. Skaalautuvuus tulee tässä esille helpon asennus- ja korkean suorituskyvyn kautta. Tavoitteiden toteuttaminen voidaan näin todeta yhden onnistuneen sovelluksen asennuksen ja suorituksen jälkeen. Sillä jos tämä prosessin toteutus onnistuu kerran, se onnistuu joka kerta. Näin esim. kolmellekymmenelle hoitajaopiskelijalle voidaan ojentaa VR -lasit, joihin on asennettu opetusmateriaali ja he voivat oppia käymällä simulaatiota.

5 Käännöstyö

5.1 Mittaukset ja testaukset

Aluksi mitattiin tietokoneversion lukemia suorituksen aikana, jotta saatiin pohjalukemat. Niitä käyttäen luotiin tavoite suoritustaso kääntötyölle. Suorituksen mittauksiin käytetty tietokone sisälsi Intel i5-10600K -prosessorin, Nvidia GTX 1660 Super -näytönohjaimen ja 32 GB:n DDR4 RAM -muistia. Sen jälkeen rakennettiin .apk-sovellus ”raakana” eli ilman asetuksien muuttamista tai optimointiprosessien lisäyksiä. Ensimmäinen askel, eli alustan kääntäminen, toteutettiin Unity-pelimoottorin avulla. Yksi Unityn ominaisuuksista on se, että se on monialustainen pelimoottori. Se siis tukee pelinkehitystä useammalle alustalle, joista yksi on Android (Denisyuk 2021). Ominaisuutta hyödyntäen projekti siirrettiin kohdealustalle kokoamalla sen koodi ja projektiasetukset, sekä luonnokset uudelleen. Mittaukset suoritettiin kohtauksen aikana ja taulukossa 1 ilmenee lähtökohtainen piirtokutsu määrä ja kuvataajuden nopeus.

Taulukko 1. Kohtauksen suoritus tietokoneen ja Oculus Quest 2:n alustalla

Kohtaus	Kuvataajuus (fps)	Piirtokutsujen määrä
Tietokonealusta	72	275
Oculus Quest 2-alusta	70	275

Tietokoneversion mittaus ilmeni yläraja-arvolla, mikä oli odotettua, sillä simulaatio suoritettiin tietokoneen teholla. Oculus Quest 2 myös suoriutui lähes samoin kohtauksesta (Taulukko 1). Syitä tähän on monta. Valot olivat leivottu valokartalle ja FFR oli asetettu kohtalaiselle tasolle. Näin kuvantamisen taakka oli tarpeeksi kevyt, jotta se voisi suorittaa maksimiavolla. Sen lisäksi laskelmallinen teho oli kohtalainen kohtauksen minimaalisuuden ansiosta. Piirtokutsujen määräkin oli hyvin alhainen sen suoritustasoon verrattuna. Esimerkkikohtauksen tyyppinen simulaatio luokitellaan kevyeksi simulaatioksi ja sille suositeltu alue toimivaan suoritukseen olisi n. 400–600 piirtokutsua (Oculus 2023d).

Se ei silti tarkoita, ettei sitä voisi vielä optimoida lisää. Kohtaus kuvantuu simulaation alussa kokonaan ja se ei ole tarpeellista. Pelaaja näkee vain osia kohtauksesta kerrallaan, jolloin siihen voidaan lisätä piilopintojen poistoa. Taulukossa 2 näkyy ensimmäinen lisätty optimointimetodi ja sen vaikutus suoritukseen.

Taulukko 2. Piilopintojen poiston käyttö kohtauksessa

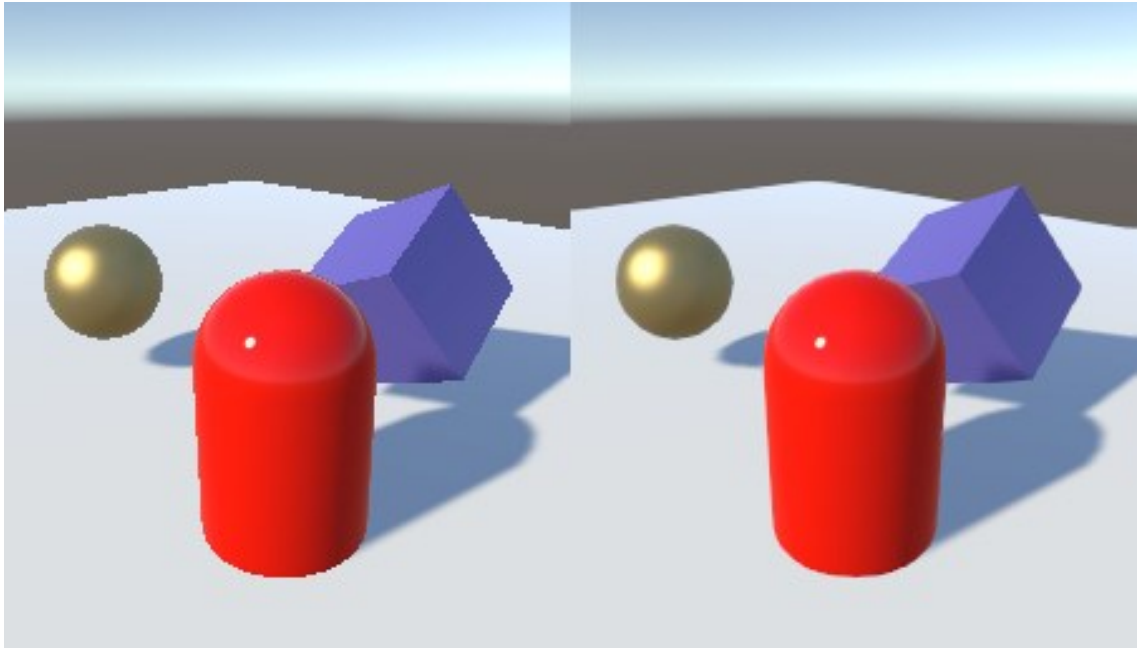
Kohtaus	Kuvataajuus (fps)	Piirtokutsujen määrä
Ilman piilopintojen poistoa	70	275
Piilopintojen poiston kanssa	71	239

Tuloksena saatiin marginaalinen parannus suoritukselle. Sitä ei voida huomata käyttäjäkokemuksen kautta, eikä sen käyttö tule ilman vaatimuksia. Koska Piilopintojen poisto on itsessään ylimääräinen laskelmallinen prosessi, joka simuloidaan suorituksen aikana, vaati sekin muistia ja suoritustehoa prosessorilta. Sitä yleensä käytetäänkin näytönohjaimen suoriutumisen parannukseen piirtokutsu- ja ylipiirtotilanteissa (Unity 2023a). Se myös toimii paremmin staattisten kameroiden kanssa, siksi VR:ssä, missä pelaaja pystyy kääntämään päätään 360 astetta, se joskus tulkitsee ja kuvantaa väärin (Kuva 2).



Kuva 2. Vasemmalla piilopintojen poiston tulos kun kääntyy pois päin. Oikealla piilopintojen poiston tulos kun katsoo hahmoa suoraan kohti.

Taakkaa täytyi kumminkin keventää mahdollisimman paljon, kun lisätiin käyttäjäkokemuksen optimoinnille tärkeä asetus, reunojen pehmenys. Yleisesti peleissä ja etenkin VR -peleissä voi esiintyä sahalaitaisuutta objektien reunoille. Syy tähän on se, että pikseleitä kuvannetaan miljoonina pieninä neliöinä näytölle. Näistä neliöstä muodostuu kuvannetut objektit ja luonnollisesti objektien reunoilla ne muodostavat yhtenäisen suoran viivan. Kun reunojen pitäisi esim. kaartua nämä viivat voivat kuvantua väärin ja erottua toisistaan, jolloin se näyttää sahalaitaiselta (Kuva 3). Reunojen pehmenys on filteri, joka muokkaa kuvantamisessa pikseleiden esiintymistä oikein reunoilla (Marak 2023).



Kuva 3. Vasemmalla ilman reunojen pehmennystä. Oikealla reunojen pehmentämisellä.

Filtterin vaikutusta mitataan kertoimella ja korkeampi kerroin tarkoittaa tarkempaa tulosta. Sen lisäksi reunojen pehennys aiheuttaa paljon ylimääräistä laskelmointia, sillä se vaikuttaa koko kuvantamisprosessiin (Taulukko 3). Tästä syystä on hyvä optimoida sovellus ennen sen lisäämistä.

Taulukko 3. Reunojen pehennys ilman ja PP eli piilopintojen poiston kanssa.

Kohtaus	Kuvataajuus (fps)	Piirtokutsujen määrä
Reunojen pehennys 2x, ilman PP	71	275
Reunojen pehennys 4x, ilman PP	58	275
Reunojen pehennys 2x PP:n kanssa	69	239
Reunojen pehennys 4x PP:n kanssa	60	239

5.2 Lopputulos

Vaikka joitakin asetuksia tai prosesseja olisi voinut säädellä korkeammalle, se ei olisi tarkoittanut positiivista korrelaatiota. Esimerkiksi kiinteä fovea-kuvantamista olisi voinut lisätä, mutta se olisi vain keventänyt näytönohjaimen taakkaa ja lisännyt sitä prosessorille. Seurauksena se olisi voinut aiheuttaa prosessorin pullonkaulausta. Tasapaino on siis tärkeintä optimoinnissa ja hyöty-haittasuhdetta pitäisi aina punnita ennen lisäämistä.

Käännöstyön lopullisiksi asetuksiksi valittiin kaksinkertainen reunojen pehmennys piilopintojen poiston kanssa, joka tarkoittaa että kohtausta vaatii 15% vähemmän suoritustehoa. Tulosten mukaan se onnistui parhaiten toteuttamaan korkeaa suoritustasoa ja samalla säilytti alkuperäisen tietokoneversion käyttäjäkokemuksen, sovelluksen muodossa. Se myös tarkoittaa sitä, että se onnistuu joka kerta suoriutumaan yhtä hyvin, eli skaalautuvuuden ongelma on myös ratkaistu. Tehtävänannon tavoitteet siis toteutuivat.

6 Yhteenveto ja johtopäätökset

Projektin tarkoitus oli kääntää simulaatiokohtaus PC -alustalta VR -laitteelle, jotta materiaali olisi skaalattavassa muodossa pedagogisien tarpeiden mukaisesti. Huomioimalla laitteen tarpeita ja mittaamalla sekä optimoimalla kohtausta, saatiin suoritustehontaakkaa laskettua 15%:lla. Samalla käyttäkokemus ylläpidettiin, eikä pelaaja huomaa eroa kääntötyössä alustasta riippumatta. Tällä lähestymisellä voidaan onnistuneesti tuottaa pieniä tehokkaita projekteja VR -laitteille ilman ongelmia, kuten pullonkaulausta.

6.1 Skaala

Vaikka kohtaaminen on itsessään onnistunut ja se luo hyvän pohjan tulvaisuuden kehitystä varten, se ei välttämättä riitä. Kohtauksen ja kokonaisen pelin optimointi toimivat eri tekniikoilla ja lähestymisenkin eroa suuresti. Piirtokutsujen määrä voi olla hyvä yhdelle kohtaukselle, mutta useampi ja toisistaan eroavat kohtaukset aikovat vaatia eri määriä piirtokutsuja.

Sisältökin voi myös vaikuttaa suuresti. Pelkästään suuren optimoidun ympäristön piirtokutsujen määrä voi olla jo 650 (Oksanen 2022). Tällöin Oculus Quest 2:n tyyppiselle laitteelle se ei ehkä olisi mahdollista kääntää, vaan se jouduttaisiin poistamaan kokonaan kääntötyössä. Se on hyvin yleistä peliyrityksille, jotka haluavat kääntää työnsä mobiilialustoille tai VR:ksi. He joutuvatkin usein uudelleen suunnittelemaan pelinsä (Ohannessian 2020).

6.2 Resurssit

Optimointi ei koskaan ole yhden henkilön vaan kokonaisen kehittäjäryhmän urakka. Jokainen henkilö on erikoistunut omaan alueeseensa, jolloin he myös tietävät parhaiten, miten optimoida, ja mitä työkaluja käyttää siihen. Siksi peliyritykset itse kääntävät heidän projektiaan muille alustoille. Sen lisäksi työkalujen määräkin nousee. Opinnäytetyön tekijä olisi voinut esimerkiksi käyttää

verkkoleipuria, joka olisi tehnyt kaikkien mallien kehyksistä yhden tekstuuriatlaksen vähentääkseen piirtokutsuja. Jos opinnäytetyön tekijä olisi ollut perehtyneempi 3D-mallinukseen (Bonnet 2020a).

Kääntötyö on myös oma alansa. Pelinkehittäjiä ja pelinkehitysryhmiä tai jopa pelistudioita erikoistuu ainoastaan projektien kääntämiseen tietyille alustoille, ja tarjoavat palvelujaan niitä vaativille yrityksille. Opinnäytetyön tulokset kuitenkin osoittavat, että esim. pienyrityksen voisivat hyödyntää näitä yleisiä optimointitekniikoita heidän VR -kääntötyössään, sillä pienyrityksillä ei yleisesti ole resursseja palkata erikseen kääntökehittäjiä. Kääntötyön tiedon yleistämisellä he voisivat toteuttaa sen itse. Autokoulut voisivat opettaa auto-onnettomuuskäyttäytymistä tai kokkikoulut paineen alla työskentelyä ja keittiössä koordinoitua VR -laseja hyödyntäen. Lastentarhoissa, lapsien olisi joskus helpompi ymmärtää VR -kokemusopetuksen kautta, verbaalisen sijasta.

Opinnäytetyötä voi käyttää pohjana tulevaisuudessa ja laajentaa käyden läpi aihekohtaiset optimoinnit. Näin luodaan kattava ohjeistus optimointiin joita pienyritykset voivat käyttää mallina omassa kääntötyössään.

Lähteet

Bonet, R. 2020a. Mesh Baking Your Assets (With Ian Deane). Viitattu 4.10.2023

<https://thegamedev.guru/unity-cpu-performance/mesh-baker-interview-ian-deane/>

Bonet, R. 2020b. Unity Overdraw: Improving the GPU Performance of Your Game. Viitattu 6.11.2023

<https://thegamedev.guru/unity-gpu-performance/overdraw-optimization/>

Denisyuk, Y. 2021. The 8 Important Stages of Successful Game Porting Process. Viitattu 4.10.2023

<https://pinglestudio.com/blog/porting/the-8-important-stages-of-successful-game-porting-process>

El-Nasr, M. et al. 2007. Dynamic Lighting for Tension in Games. Viitattu 2.12.2023

https://gamestudies.org/0701/articles/elnasr_niedenthal_knez_almeida_zupko

Hamad, A. & Jia, B. 2022. How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations. Viitattu 26.11.2023

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9517547/>

Intel, 2023. How to Properly Balance Your Components. Viitattu 26.11.2023

<https://www.intel.com/content/www/us/en/gaming/resources/what-is-bottlenecking-my-pc.html>

Jukić, T. 2015. Draw calls in a nutshell. Viitattu 26.11.2023

<https://toncijukic.medium.com/draw-calls-in-a-nutshell-597330a85381>

Khandaker, S. 2021. If and Only If – Performance Cost of Using If Statements and Branch Predictor. Viitattu 17.11.2023

<https://codeburst.io/if-and-only-if-2d58ef7ef027>

Marak, S. 2023. What is Anti-Aliasing? Ultimate Guide. Viitattu 21.11.2023

<https://www.selecthub.com/resources/what-is-anti-aliasing/>

Mason, P. 2022. Porting to Meta Quest. Viitattu 29.11.2023

<https://starloopstudios.com/porting-to-meta-quest/>

Oculus, 2023a. Fixed Foveated Rendering (FFR). Viitattu 30.11.2023

<https://developer.oculus.com/documentation/native/android/mobile-ffr/>

Oculus, 2023b. Guidelines for VR Performance Optimization. Viitattu 16.11.2023

<https://developer.oculus.com/documentation/native/pc/dg-performance-guidelines/>

Oculus, 2023c. Performance Analyzer and Metrics. Viitattu 15.11.2023

<https://developer.oculus.com/documentation/unity/ts-odh-logs-metrics/>

Oculus, 2023d. Testing and Performance Analysis. Viitattu 16.11.2023

<https://developer.oculus.com/documentation/unity/unity-perf/>

Ohannessian, K. 2020. Porting To Oculus Quest: How To Optimize And Maximize Your VR Game. Viitattu 26.11.2023

<https://www.uploadvr.com/porting-oculus-quest-VR -optimize/>

Oksanen, M. 2022. 3D INTERIOR ENVIRONMENT OPTIMIZATION FOR VR. Viitattu 27.11.2023

https://www.theseus.fi/bitstream/handle/10024/788282/Oksanen_Mika.pdf?sequence=2&isAllowed=y

Raj, A. 2023. Meta headset sales outperform other brands in 2023. Viitattu 26.11.2023

<https://techwireasia.com/2023/07/meta-headset-sales-outperform-other-brands-in-2023/>

Sciutteri, M. 2016. Using a Texture Atlas to Optimize Your Game. Viitattu 26.11.2023

<https://code.tutsplus.com/using-texture-atlas-in-order-to-optimize-your-game--cms-26783a>

Starloop Studios, 2023. The How's and Why's of Game Porting. Viitattu 27.11.2023

<https://starloopstudios.com/the-hows-and-whys-of-game-porting/>

Turku AMK, 2022. Safe4Child – Trauma informed approaches to support staff working with children and adolescents exhibiting behaviours that challenge. Viitattu 23.11.2023

<https://www.tuas.fi/en/research-and-development/projects/safe4child-caring-violent-child-safely-in-child-ps/>

Unity, 2023a. Light mode: Baked. Viitattu 22.11.2023

<https://docs.unity3d.com/Manual/LightMode-Baked.html>

Unity, 2023b. Occlusion Culling. Viitattu 22.11.2023

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

Vince, J. (2004). Virtual Reality. In: Introduction to Virtual Reality. Springer, London. Viitattu 8.12.2023