

Timo Nivala

CAVE-ohjelmiston kehitys Unity-pelimoottorille

**Opinnäytetyö
CENTRIA AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Toukokuu 2014**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieskan yksikkö	Aika Toukokuu 2014	Tekijä/tekijät Timo Nivala
Koulutusohjelma Tietotekniikan koulutusohjelma		
Työn nimi CAVE-ohjelmiston kehitys Unity-pelimoottorille		
Työn ohjaaja Hannu Puomio		Sivumäärä 21 + 7 liitettä
Työelämäohjaaja Mika Sorvisto		
<p>Opinnäytetyössä kehitettiin ohjelmistolisäosat, joiden avulla Unity-pelimoottoria voidaan käyttää sisällön tuottamiseen Satavision-CAVE -järjestelmiin.</p> <p>Teoriaosassa tutkittiin kuinka CAVE-järjestelmä toimii ja miten sen vaatimat ominaisuudet olisi mahdollista toteuttaa Unity-pelimoottorilla.</p> <p>Käytännön osuus koostui vaadittavien lisäosien ohjelmoinnista. Lisäosien ominaisuuksina toteutettiin virtuaalikameroiden perspektiivin korjaus, stereoskooppisen kuvan muodostus, käyttäjän optinen paikannus sekä liikkuminen virtuaaliympäristössä langattomilla syöttölaitteilla.</p>		
Asiasanat CAVE, Unity, stereoskopia, virtuaaliympäristö, 3D		

ABSTRACT

Unit Department of Ylivieska	Date May 2014	Author Timo Nivala
Degree programme Bachelor of Engineering, Information Technology		
Name of thesis CAVE software development for Unity game engine		
Instructor Hannu Puomio		Pages 21 + 7 appendices
Supervisor Mika Sorvisto		
<p>The subject of this thesis was to develop CAVE software add-ons for Unity game-engine to create content for Satavision-CAVE systems.</p> <p>The theoretical part of the work was to investigate how CAVE system works and how to implement necessary software features using Unity game engine.</p> <p>The practical part of the work was to program software add-ons with required features. Features of the add-ons were perspective correction of virtual cameras, stereoscopic output, optical tracking of user and navigation in virtual world using wireless input devices.</p>		

<p>Key words CAVE, Unity, stereoscopy, virtual reality, 3D</p>

KÄSITTEIDEN MÄÄRITTELYT

CAVE	Cave Automatic Virtual Environment. Ulkoasultaan huonetta muistuttava järjestelmä, jolla luodaan todentuntuinen, keinotekoinen ympäristö käyttäjän ympärille.
Rekursiivinen akronyymi	Itseensä viittaava lyhenne, joka toistuu lyhenteen auki kirjoitetussakin muodossa.
Virtuaaliympäristö	Tietokoneella luotu keinotekoinen ympäristö.
Stereoskopia	Yhteisnimitys tekniikoille, joilla voidaan luoda syvyysvaikutelma kaksikulotteiseen kuvaan.
Binokulaarinen vihje	Syvyysnäön hahmottamiseen liittyvä vihje, joka perustuu molempien silmien yhtäaikaan aistimukseen.
Polarisaatio	Aaltoliikkeen värähtelyjen amplitudin suuntariippuvuus aaltoliikkeen etenemisuuntaan nähden kohtisuorassa tasossa.
OpenGL	Monialustainen ohjelmointirajapinta interaktiivisen tietokonegrafiikan tuottamiseen.
Direct3D	Microsoft-yhtiön luoma ohjelmointirajapinta interaktiivisen tietokonegrafiikan tuottamiseen Windows-käyttöjärjestelmissä.
Debugaus	Ohjelmistojen virheiden paikallistaminen ja korjaaminen.
GPU	Graphics Processing Unit. Tietokonegrafiikan käsittelyyn tarkoitettu mikroprosessori.

DLL	Dynamic Link Library. Windows-käyttöjärjestelmässä käytettävä jaettu ohjelmakirjasto.
Heuristinen päättely	Epätäsmällinen, arvioiva päättelytyyli, jonka tarkoituksena on nopeasti päätellä asia riittävän lähelle oikeaa lopputulosta.
Interpoloiminen	Uusien arvojen laskemista jo olemassa olevien raja-arvojen avulla.

**TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELYT
SISÄLLYS**

1 JOHDANTO	1
2 CAVE-JÄRJESTELMÄ	2
2.1 Yleisesti	2
2.2 Historia	2
2.3 Tekniikka	3
2.4 Satavision-CAVE	6
3 UNITY-PELIMOOTTORI	9
3.1 Ominaisuudet	9
3.2 CAVE-valmius	10
4 OHJELMISTOLISÄOSIEN KEHITYS	12
4.1 Perspektiivin korjaus	12
4.2 Stereoskooppinen kuva	14
4.3 Käyttäjän paikannus	17
4.4 Navigointi virtuaaliympäristössä	19
4.5 Lisäosien käyttö	19
5 YHTEENVETO	21
LÄHTEET	22
LIITTEET (poistettu julkisesta versiosta)	

1 JOHDANTO

Opinnäytetyön toimeksiantajana toimiva Satavision Oy valmistaa CAVE-järjestelmiä ja virtuaaliloja. Vaikka CAVE-järjestelmät ovat yleistyneet maailmalla vuosien varrella, ei monissa nykyisissä sisällöntuotanto-ohjelmistoissa ole suoraa CAVE-tukea. Sisällön näyttäminen CAVE-järjestelmissä on perinteisesti toteutettu erillisillä virtuaalitodellisuusohjelmistoilla. Kuitenkin monet nykyiset virtuaalitodellisuusohjelmistot eivät toimeksiantajan kokemusten mukaan vastaa tehokkuudeltaan, graafisilta ominaisuuksiltaan, käyttömukavuudeltaan tai hinnaltaan nykyisiä vaatimuksia.

Opinnäytetyön lähtökohtana oli tarve toteuttaa vaihtoehto perinteisille virtuaalitodellisuusohjelmistoille sisällön näyttämiseksi CAVE-järjestelmissä. Pelimoottorien käyttöä vaihtoehtona vanhoille virtuaalitodellisuusohjelmistoille käsittelee Asmo Jussila opinnäytetyössään ”Unity-pelimoottorin hyödyntäminen CAVE-järjestelmässä”. Kyseisessä opinnäytetyössä tehtiin vertailua eri pelimoottorien suhteen ja saatiin hyviä kokemuksia Unity-pelimoottorin käytöstä CAVE-järjestelmässä. Myös Satavision Oy:llä oli aikaisempia hyviä kokemuksia Unity-pelimoottorin käyttämisestä visualisointien alustana, joten tälle pelimoottorille päätettiin kehittää ohjelmistolisäosat, jotka mahdollistaisivat helpon sisällön tuottamisen Satavision Oy:n CAVE-järjestelmiin.

Ensimmäisessä pääluvussa käsitellään CAVE-järjestelmää yleisesti sekä sen historiaa ja tekniikkaa. Lisäksi käsitellään Satavision Oy:n CAVE-järjestelmän erityispiirteitä ja ominaisuuksia. Toisessa pääluvussa käsitellään Unity-pelimoottoria ja sitä, kuinka siihen on mahdollista kehittää vaadittavat ohjelmistolisäosat. Kolmannessa pääluvussa kerrotaan ohjelmistolisäosien toteutuksesta sekä niiden käytöstä Unity-pelimoottorissa. Neljäs pääluku sisältää yhteenvedon sekä pohdintaa työn onnistumisesta.

2 CAVE-JÄRJESTELMÄ

2.1 Yleisesti

CAVE-järjestelmä on laitteistokokonaisuus, jonka avulla on tarkoitus luoda todentuntuinen virtuaaliympäristö käyttäjän ympärille. Järjestelmä muistuttaa yleensä pientä huonetta, jonka seinät muodostuvat kuvapinnoista. Kuvapintojen kuva muodostetaan yleensä projektoreilla etu- tai takaprojisoitina. Myös näyttöpaneelleja voidaan käyttää kuvapintojen muodostamiseen. Kuvapinnoille muodostetaan stereoskooppinen kuva virtuaaliympäristöstä käyttäjän perspektiivistä, jolloin virtuaaliympäristö ja sen kohteet näkyvät katselijalle luonnollisessa koossa. Paikannusjärjestelmän avulla näkymien perspektiiviä voidaan muuttaa käyttäjän liikkuaessa tilassa. Tällöin käyttäjä voi esimerkiksi kyykistyessään nähdä virtuaaliympäristössä olevan esineen alle. (Wikipedia 2014.)

2.2 Historia

CAVE:n kehittivät Thomas A. DeFanti ja Daniel J. Sandin johtama tutkimusryhmä Illinoisin yliopistossa Chicagossa. Se esiteltiin ensimmäisen kerran vuonna 1992 tietokonegrafiikka-aiheisessa SIGGRAPH -tapahtumassa. Nimitys CAVE on rekursiivinen akronyyymi, joka muodostuu sanoista Cave Automatic Virtual Environment. Nimi viittaa myös Platonin luolavertaus -metaforaan, joka käsittelee todellisuuden käsitystä ja illuusiota. CAVE™ -nimi on Illinoisin yliopiston rekisteröimätön tavaramerkki ja se viittaa nimenomaan alkuperäiseen CAVE™ virtuaalitalaan. Nykyään sanaa on kuitenkin alettu käyttää yleissanana alkuperäisen CAVE™:n kaltaisista virtuaalituloista. Monet nykyiset kaupalliset virtuaalitaratkaisut viittaavat nimissään alkuperäiseen CAVE™ -nimeen kuitenkin niin, että nimitys eroaa CAVE™ -tavaramerkistä. Vuonna 1994 yhdysvaltalainen Mechdyne-yhtiö lisensoi Illinoisin yliopistolta alkuperäisen CAVE™-tekniikan ja näin ollen käyttää alkuperäistä nimeä CAVE™ -nimeä kaupallisissa toteutuksissaan. Tässä opinnäytetössä CAVE -nimitystä käytetään yleissanana kuvaamaan huoneen kaltaisia virtuaalituloja. CAVE™ -nimellä viitataan

nimenomaan alkuperäiseen, rekisteröimättömään tuotemerkkiin. Satavision-CAVE –nimellä tarkoitetaan opinnäytetyössä Satavision Oy:n CAVE-järjestelmiä. (Dave Pape 2001.)



Kuva 1. CAVE™ Illinoisin yliopistossa vuonna 2001. (Wikipedia 2014.)

2.2 Tekniikka

CAVE-järjestelmien tekninen toteutus vaihtelee riippuen muun muassa järjestelmän koosta, ominaisuuksista sekä käytetystä budjetista. Järjestelmään kuuluu kuitenkin yleensä laitteen runko, näyttöpinnat, projektorit, tietokonejärjestelmä ohjelmistoineen, paikannusjärjestelmä, 3D-lasit sekä navigointilaite.

Näyttöpinnat toteutetaan yleensä projektoreilla, koska niillä on helppo toteuttaa isoja yhtenäisiä kuvapintoja. Kuvapintojen toteutuksissa yritetään välttää saumoja tai kuva-alueen katkoksia, sillä ne haittaavat stereoskooppisen kuvan syvyyden havainnointia. Näyttöpintojen kokoa, suuntaa tai määrää ei ole tarkkaan määritelty,

mutta yleensä tarkoituksena on peittää mahdollisimman paljon käyttäjän näkökentästä, jotta illuusio ympärillä olevasta virtuaaliympäristöstä säilyisi mahdollisimman hyvin. Projektoreilla toteutetut näyttöpinnat voidaan toteuttaa joko etu- tai takaprojisoinnilla. Etuprojisoinnissa projektorin kuva heijastetaan läpinäkymättömään pintaan katselupuolelta. Takaprojisoinnissa kuva heijastetaan puoliläpinäkyvään pintaan katselupinnan toiselta puolelta. Se kumpaa projisointitekniikkaa käytetään riippuu yleensä siitä, kuinka monella näyttöpinnalla CAVE-järjestelmä on toteutettu. Useampiseinäiset järjestelmät toteutetaan yleensä takaprojisoinnilla, jotta projektorien sijoittelusta ja niiden käyttäjistä ei aiheutuisi varjoja kuvapinnoille. (Wikipedia 2014.)

Stereoskopian avulla näyttöpintojen kuville saadaan tarvittava syvyysvaikutelma. Yleensä tämä tapahtuu jäljittelemällä syvyysnäön binokulaarisia syvyysvihjeitä. Tällöin kummallekin silmälle näytetään hieman toisistaan eroavaa kaksiulotteista kuvaa, jolloin ihmisen aivot tulkitsevat ne yhdeksi kolmiulotteiseksi kuvaksi. CAVE-järjestelmissä stereoskopia toteutetaan yleensä 3D-laseilla, joiden avulla saadaan määriteltyä minkälaista kuvaa kumpikin silmä näkee. Tällainen stereoskopia voidaan toteuttaa joko aktiivi- tai passiivitekniikalla. Aktiivitekniikassa 3D-lasit sisältävät elektroniikkaa, joka pimentää lasien linssijä vuorotellen siten, että vain toinen silmä näkee kerrallaan laseista läpi. Tämä vuorottainen linssien pimentäminen tapahtuu laseista riippuen useista kymmenistä kerroista satoihin kertoihin sekunnissa. Kun aktiivilasien toimintaan yhdistetään näyttöpaneeli tai projisoitu kuva, joka vaihtuu samassa tahdissa lasien vuorottelun kanssa, saadaan kummallekin silmälle näytettyä eri kuva. (Wikipedia 2014.)

Passiivitekniikassa 3D-lasit eivät sisällä elektroniikkaa. Passiivitekniikassa lasien linssit toimivat suodattimena valonsäteille päästäten vain tietyn ominaisuuden omaavan valon läpi. Linssit voivat suodattaa muun muassa valon eri taajuuksia eli värejä tai valon aaltoliikkeen värähtelyn suuntaa eli polarisaatiota. Kun vasemman ja oikean linssin suodatus eroavat toisistaan, on mahdollista näyttää erilaista kuvaa kummallekin silmälle suodattamalla kuvalähteen valonsäteet linssijä vastaavalla tavalla. CAVE-järjestelmien passiivitekniikassa käytetään yleensä valon polarisointia. Projektoreilla toteutettaessa tämä vaatii kaksi projektorista yhtä kuvapintaa kohden. Toinen projektori näyttää oikealle ja toinen vasemmalle

silmälle tarkoitettua kuvaa. Kummankin projektorin linssin edessä on toisistaan eroava polarisaatiosuodatin. Kun heijastettua kuvaa katsotaan passiivilaseilla, joissa on projektoreita vastaavat suodattimet linseissä, näkevät kummatkin silmät vain niille tarkoitetun kuvan. Polarisationitekniikka vaatii toimiakseen valon polarisaation säilyttävän näyttöpinnan. (Wikipedia 2014.)

Näyttöpinoilla näytettävä stereoskooppinen kuva tuotetaan tietokonejärjestelmällä. Tietokonejärjestelmä laskee ja piirtää kullekin näyttöpinnalle näkymän virtuaaliympäristöstä käyttäjän perspektiivistä. Tietokonejärjestelmä muodostuu järjestelmäkokonaisuudesta riippuen yhdestä tai useammasta koneesta. Yhden tietokoneen järjestelmässä kaikkien kuvapintojen kuvasignaali tuotetaan yhdeltä koneelta useasta eri kuvaulostulosta. Järjestelmän etuna on järjestelmän yksinkertaisempi käyttö ja varmempi toiminta verrattuna useaan tietokoneeseen. Huonona puolena on isompi laskentatehon tarve. Usean tietokoneen järjestelmässä tietokoneet on verkotettu toisiinsa niin sanotuksi klusteriksi. Tällöin yksi koneista toimii pääkoneena, joka ohjaa muita koneita verkon välityksellä. Koneiden kuvaulostulot ovat yleensä tahdistettuja keskenään, jotta stereoskooppinen kuva pysyisi kaikilla kuvapinoilla samassa vaiheessa. Klusterin etuna on tietokoneiden yhdistetty laskentateho, mutta huonona puolena järjestelmän monimutkaisuus sekä epävarmempi toiminta verrattuna yhteen ohjauskoneeseen. (Bruno Raffin & Luciano Soares 2006.)

Paikannusjärjestelmää käytetään käyttäjän katselusijainnin ja -suunnan paikantamiseen. Tietokonejärjestelmä laskee sijainnin ja suunnan perusteella kuvapintojen näkymän virtuaaliympäristöstä käyttäjän perspektiivistä. Käyttäjän liikkuesssa järjestelmä mukauttaa näkymien perspektiivin vastaamaan katselijan sijaintia. Yleinen paikannustekniikka on optinen, jossa CAVE-järjestelmän runkoon kiinnitetty yksi tai useampi kamera kuvaa käyttäjää, ja tietokone laskee kuvan perusteella käyttäjän sijainnin ja suunnan. Yleensä käytetään 3D-laseihin kiinnitettyjä markkereita, joiden avulla käyttäjän silmien sijainti ja suunta saadaan tarkemmin laskettua. (Wikipedia 2014.)

2.3 Satavision-CAVE

Satavision-CAVE on toimeksiantajan, Satavision Oy:n valmistama kaupallinen CAVE-järjestelmä, jota voidaan käyttää muun muassa rakennettavan ympäristön suunnitteluun ja havainnointiin, koulutuksiin sekä erilaisten tuotteiden ja palveluiden myyntiin ja markkinointiin. Järjestelmä yksilöidään asiakkaan tarpeiden mukaan, joten sen rakenne ja ominaisuudet vaihtelevat. Opinnäytetyössä kehitetyt ohjelmistolisäosat ovat yhteensopivia käytännössä kaikkien Satavision-CAVE -kokoonpanojen kanssa, mutta käytössä olleen rajallisen ajan vuoksi kehityksessä ja testauksessa keskityttiin opinnäytetyön kirjoitushetkellä yleisimpään kokoonpanoon, CAVE-M -malliin. (Satavision Oy 2014.)

(kappale poistettu julkisesta versiosta)

3 UNITY-PELIMOOTTORI

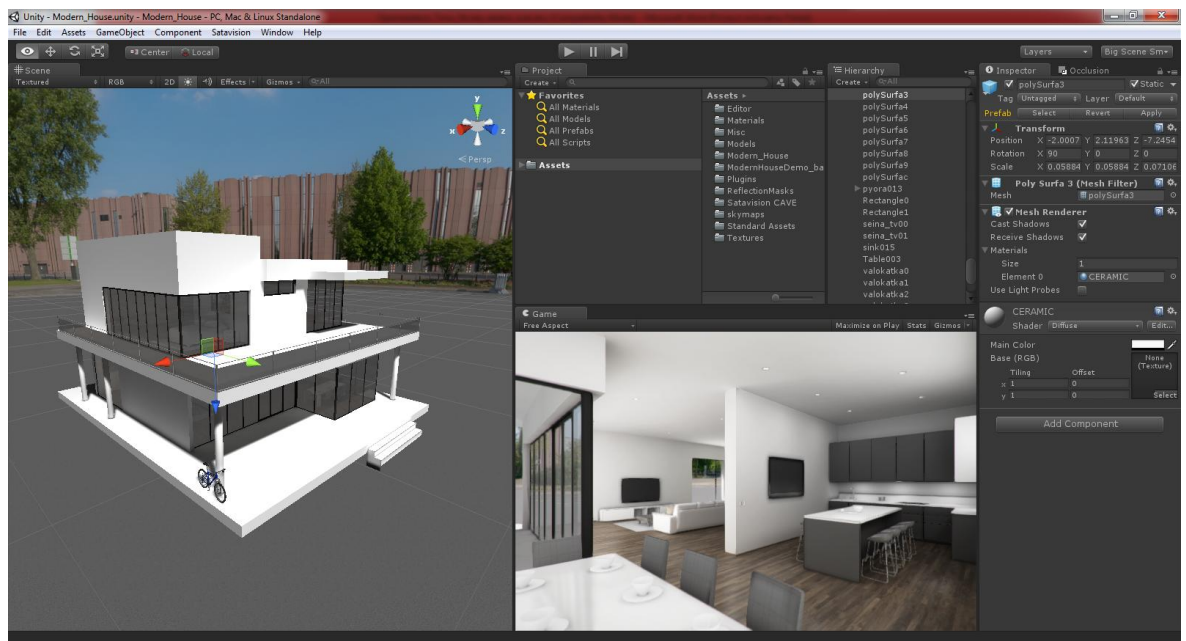
Unity on Unity Technologiesin kehittämä monialustainen pelimoottori, jolla voidaan kehittää pelejä ja interaktiivisia sovelluksia tietokoneille, mobiililaitteille, konsoleille sekä verkkoselaimille. Pelimoottorin kehitys aloitettiin vuonna 2001 ja ensimmäinen versio julkaistiin vuonna 2005, jolloin sen kehitys- ja julkaisualustana toimi OS X -käyttöjärjestelmä. Myöhemmin kehitys- ja julkaisualustat ovat lisääntyneet merkittävästi ja nykyisin tuettuja julkaisualustoja ovat muun muassa Windows, OS X, Linux, iOS, Android, Windows Phone 8, Xbox One, Play Station 4 ja Wii. Kehitysalustoina voidaan käyttää Windows- ja OS X –käyttöjärjestelmiä. (Unity Technologies 2014.)

Opinnäytetyön kirjoitushetkellä Unitystä on saatavilla versionumero neljä, mutta seuraavan version tulevasta julkaisusta on jo ilmoitettu. Unity on jaettu lisensoinnin perusteella kahteen versioon. Unity Free on ilmainen perusversio, jota voi käyttää yksityiseen ja pienen kokoluokan kaupalliseen käyttöön. Unity Pro on maksullinen versio, jota voi käyttää yksityiseen ja kaupalliseen käyttöön ilman rajoituksia. Unity Pro sisältää lisäominaisuuksia, joita perusversiossa ei ole. (Unity Technologies 2014.)

3.1 Ominaisuudet

Unity:n ominaisuuksiin kuuluu tyypilliset pelimoottorin perusominaisuudet, kuten renderöinti-, fysiikka- ja äänimoottori, sisällön tuonti, ohjelmointirajapinta ja verkko-ominaisuudet. Unity sisältää myös muun muassa maastoeditorin, Mechanim-animaatiotyökalun, reittilaskennan, näkyvyysoptimoinnin, partikkelieditorin sekä Beast-valaistusmoottorin. Unityllä tehtävä peli- ja sovelluskehitys tapahtuu graafisessa kehitysympäristössä Unity Editorissa. Editori tarjoaa visuaalisen käyttöliittymän, jossa editoriin tuodut sisällöt, kuten 3D-mallit, tekstuurit, äänet ja skriptit yhdistetään halutuksi kokonaisuudeksi. (Unity Technologies 2014.)

Unityssä toiminnallisuudet luodaan skripteillä. Unityn skriptijärjestelmän perustana on Mono, joka on Microsoftin .NET Frameworkin kaltainen monialustainen ohjelmistokehitysympäristö. Skripteillä voidaan luoda toiminnallisuuksia sovelluksen objekteille, mutta niillä voidaan myös luoda lisäosia ja toiminnallisuuksia itse editoriin. Unityn skriptejä voidaan kirjoittaa C#, JavaScript tai Boo –kielillä. Skriptejä voi kirjoittaa millä tahansa tekstieditorilla, mutta debuggaus toimii ainoastaan Unityn mukana tulevalla Mono Develop –ohjelmointiympäristöllä. Unityssä on tuki myös C, C++ ja Objective-C -kielillä kirjoitetuille liitännäisille. Näiden liitännäisten funktioita voidaan kutsua skripteillä, jolloin saadaan käyttöön suorituslustan alemman tason ominaisuuksia. (Unity Technologies 2014.)



Kuva 3. Unity Editor.

3.2 CAVE-valmius

Unity-pelimoottorissa ei ole valmiina suoraa CAVE-tukea, mutta tarvittavat perustoiminnallisuudet on mahdollista toteuttaa skripteillä sekä ohjelmistoliitännäisillä. Unity-pelimoottorille on saatavilla kolmannen osapuolen tekemiä kaupallisia CAVE –käyttöön tarkoitettuja lisäosia, kuten i'm in VR –yhtiön MiddleVR for Unity. Lisäosasta oli opinnäytetyön kirjoitushetkellä ladattavissa

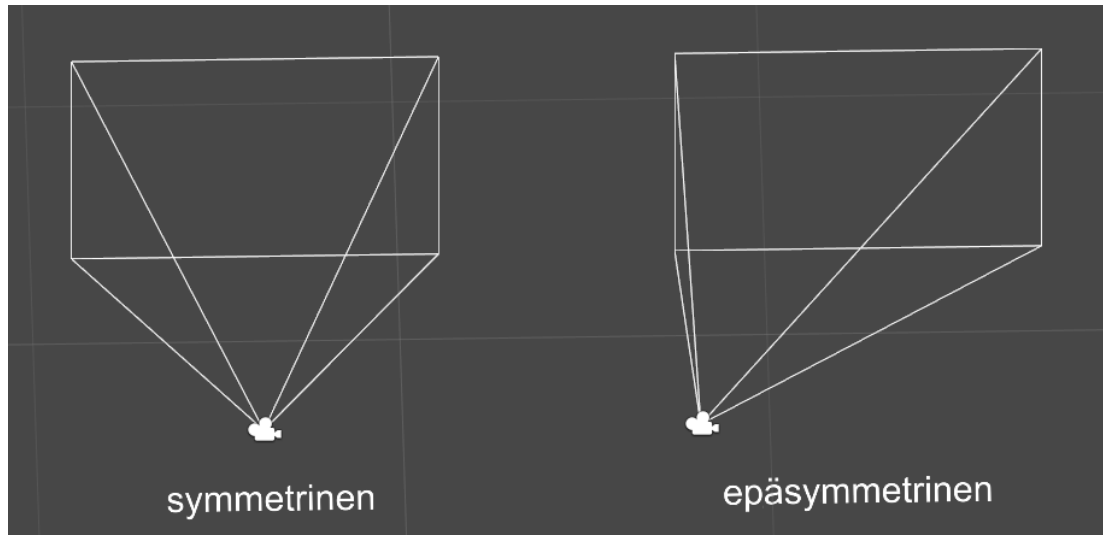
kokeiluversio, jonka toimivuutta kokeiltiin toimeksiantajan CAVE-järjestelmissä. Kokeilussa selvisi että lisäosan stereoskopia ei toimi halutulla tavalla. MiddleVR for Unity –lisäosan aktiivinen stereoskopia vaati toimiakseen OpenGL –rajapinnan käytön, mikä ei ole täysin tuettuna Windows-pohjaisissa Unity-sovelluksissa. Tämä aiheutti sovelluksissa muun muassa turhaa tehohäviötä ja jälkikäsittelyefektien toimimattomuutta. (i'm in VR. 2014.)

4 OHJELMISTOLISÄOSIEN KEHITYS

Tarkoituksena oli kehittää Unity-pelimoottorille lisäosat, joiden avulla toimeksiantaja ja heidän asiakkaansa voisivat tuottaa sisältöä SataVision-CAVE -järjestelmiin. Haluttuja ominaisuuksia olivat perspektiivin korjaus, stereoskooppinen kuva, käyttäjän optinen paikannus sekä navigointi virtuaaliympäristöissä langattomilla syöttölaitteilla. Lisäosia oli tarkoitus pystyä jakamaan toimeksiantajan omana tuotteena, joten kehityksessä ei ollut mahdollista käyttää kolmansien osapuolien maksullisia ratkaisuja.

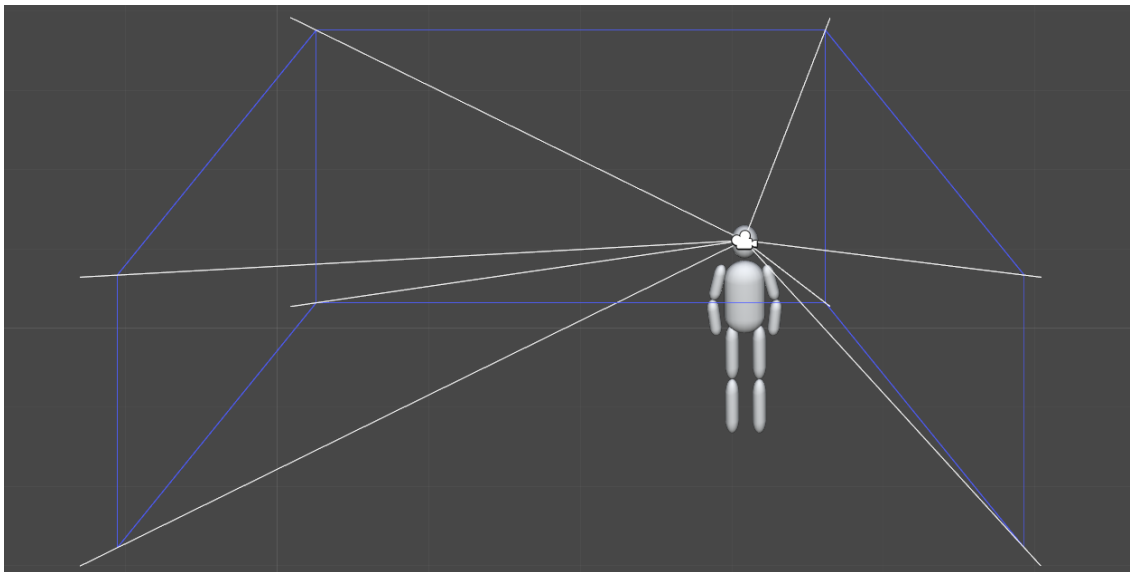
4.1 Perspektiivin korjaus

Ohjelmistolisäosien toteutus aloitettiin perspektiivin korjauksesta. Perspektiivin korjauksella tarkoitetaan virtuaalikameroiden piirron muokkaamista oikeaan muotoon niin, että näyttöpintoihin muodostetun kuvan perspektiivi on oikea suhteessa katselijaan nähden. Perspektiivin korjaus tehdään muuttamalla virtuaalikameroiden perspektiiviprojektiota. Perspektiiviprojektion määrittelee virtuaalikameran näkymäkartio. Englanninkielisessä kirjallisuudessa näkymäkartiosta käytetään nimitystä *view frustum*. Näkymäkartio määrittelee kuinka laajan ja minkä muotoisen näkymän virtuaalikamera näkee virtuaalimaailmaan. Näkymäkartioiden voivat olla symmetrisiä tai epäsymmetrisiä. Yleensä visualisointiohjelmien virtuaalikameroiden näkymäkartioiden ovat symmetrisiä, jolloin kameran muodostaman kuvan perspektiivi vastaa keskeltä katsottua kohtisuoraa näkymää kuvapinnalle. CAVE-järjestelmissä kaikkia kuvapintoja ei yleensä katsella kohtisuoraan keskeltä, minkä vuoksi virtuaalikameroissa on käytettävä epäsymmetrisiä näkymäkartioiden. Kuvassa 4 on esitettyä symmetrisen ja epäsymmetrisen näkymäkartioiden.



Kuva 4. Symmetrinen ja epäsymmetrinen näkymäkartio.

Jotta virtuaalikameroiden muodostama perspektiivi olisi oikea katselijalle, on näkymäkartiot muodostettava siten, että niiden tahkot mukailevat CAVE-järjestelmän näyttöpintoja vastaavia virtuaalisia näyttöpintoja. Kuvassa 5 on esitetty kolmeseinäistä CAVE-järjestelmää vastaavat virtuaaliset näyttöpinnat sekä niitä mukailevat näkymäkartiot. Virtuaaliset näyttöpinnat on merkitty kuvaan sinisellä ja näkymäkartiot valkoisilla viivoilla. Jokaista näyttöpintaa kohden on oma virtuaalikameransa, jonka näkymäkartion muoto riippuu katselijan sijainnista näyttöpintaan nähden.



Kuva 5. Kolmeseinäisen CAVE-järjestelmän virtuaaliset näyttöpinnat sekä niitä vastaavat näkymäkartiot visualisoituna.

Unity-pelimoottorissa kameroiden näkymäkartiot ovat oletusarvoisena symmetrisiä. Editorin käyttöliittymällä voidaan muokata näkymäkartion avautumiskulmaa, mutta ei epäsymmetristä muotoa. Epäsymmetrisen näkymäkartion muodostamiseksi on kameran ominaisuuksia muokattava scriptillä. Kameran näkymäkartion muoto määritellään kameran projektiomatriisilla. Projektiomatriisin alkioden arvoja voidaan muuttaa scriptillä, jolloin saadaan muodostettua halutun muotoinen näkymäkartio. Unityn dokumentaatiossa on yksinkertainen esimerkki projektiomatriisin muokkaamisesta. Esimerkin pohjalta luotiin C# -kielellä scripti, joka laskee tarvittavan projektiomatriisin scriptissä määritellyn virtuaalisen näyttöpinnan ja siihen linkitetyn kameran sijainnin perusteella. Käytön helpottamiseksi scripti piirtää editorinäkymään virtuaalisen näyttöpinnan sekä sitä vastaavan kameran näkymäkartion. Scripti löytyy liitteestä yksi.

CAVE-M -järjestelmän vaatiman näkymän luomiseksi scripti asetettiin neljään objektiin, joiden sijainti, koko ja asento editorinäkymässä asetettiin vastaamaan järjestelmän näyttöpintoja. Scripteihin linkitettiin neljä kameraa, jotka asetettiin yhden emo-objektin alle. Emo-objektia siirtelemällä voitiin kameroille valita käyttäjän pään sijaintia vastaava kohta virtuaalimaailmassa, jolloin kameroiden tuottama kuva saatiin näkymään käyttäjälle oikeassa perspektiivissä. Ilman käyttäjän paikannusta perspektiivin korjaus toimii kuitenkin vain yhdestä määrittelystä katselukohdasta. Useimmat Satavision-CAVE -järjestelmät ovat tarkoitettu usean henkilön yhtäaikaiseen käyttöön, jolloin on järkevää käyttää yhtä staattista katselukohtaa. Usean henkilön käyttötilanteissa perspektiivin korjauspisteeksi valitaan yleensä kohta tilan keskeltä keskiarvoiselta katselukorkeudelta, jolloin saadaan kohtuullisen oikea perspektiivi kaikille keskialueella oleville käyttäjille.

4.2 Stereoskooppinen kuva

Toisena ominaisuutena toteutettiin stereoskooppisen kuvan piirto.

Stereoskooppisen kuvan piirto Satavision-CAVE -järjestelmässä oli mahdollista toteuttaa joko OpenGL Quad Buffer Stereo, Direct3D Quad Buffer Stereo tai

NVIDIA 3D Vision –tekniikalla. Koska OpenGL –rajapinta ei ollut kunnolla tuettuna Windows-pohjaisissa Unity-sovelluksissa, jäivät mahdollisiksi toteutusvaihtoehdoiksi Direct3D Quad Buffer Stereo ja NVIDIA 3D Vision - tekniikat. Aluksi testattiin NVIDIA 3D Vision –tekniikan käyttöä, koska sen avulla oli mahdollista saada stereoskooppinen piirto ilman lisäohjelmointia. Tekniikka toimi, mutta sen käyttö aiheutti niin paljon rajoituksia Unityn ominaisuuksissa, että sen käytöstä päätettiin luopua. NVIDIA 3D Vision –tekniikkaa käytettäessä muun muassa reaaliaikaiset varjot, jälkikäsitteilyefektit ja heijastukset eivät toimineet oikein, eikä tekniikka antanut tarpeeksi vapauksia stereoskopisen kuvan säätöön. Tekniikkaa käytettäessä ei voinut olla varma stereoskooppisen piirron lopputuloksesta, koska stereokuvaparien muodostukseen ei voinut itse kunnolla vaikuttaa. 3D Vision –tekniikan rajoituksista johtuen päätettiin stereoskooppisen kuvan muodostus tehdä Direct3D Quad Buffer Stereo –tekniikkaa käyttäen. Tässä tekniikassa stereoskooppinen kuvapari muodostetaan itse, jolloin voidaan tarkasti vaikuttaa stereoskooppisen kuvan lopputulokseen. Lisäksi tekniikka ei rajoita Unityn piirto-ominaisuuksia samalla tavalla kuin NVIDIA 3D Vision -tekniikka.

(kappale poistettu julkisesta versiosta)

4.3 Käyttäjän paikannus

(kappale poistettu julkisesta versiosta)

4.4 Navigointi virtuaaliympäristössä

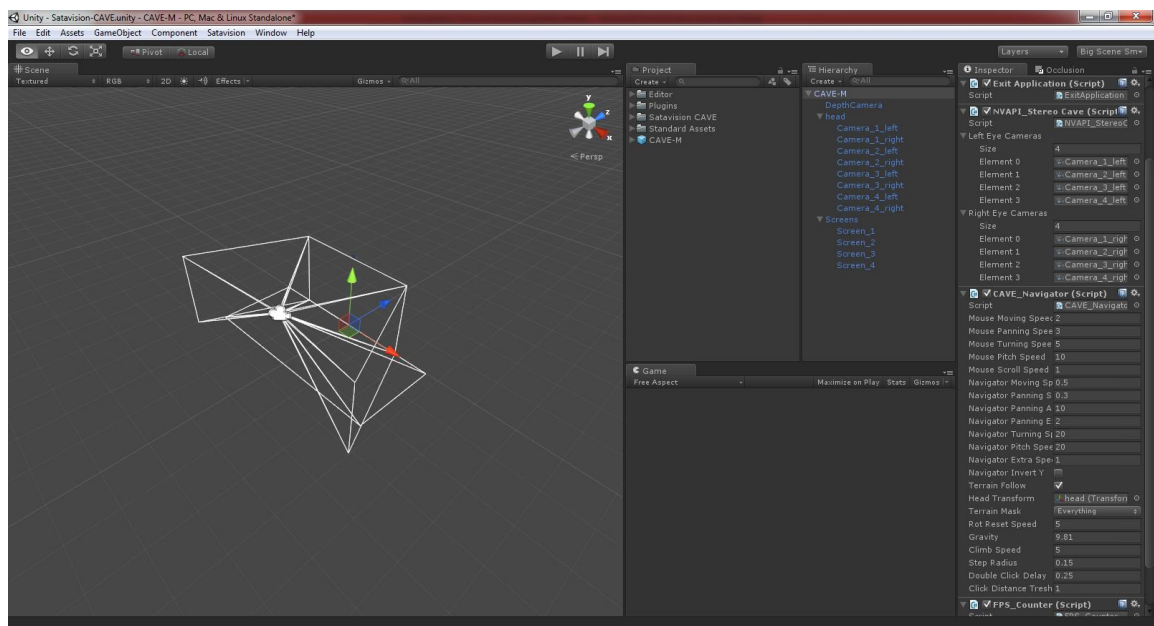
Viimeisimpänä ominaisuutena lisäosiin kehitettiin CAVE-käyttöön sopivat navigointiscriptit, joiden avulla olisi mahdollista navigoida virtuaaliympäristössä langattomilla syöttölaitteilla. Satavisionin-CAVE -järjestelmiin kuuluvia langattomia syöttölaitteita ovat näppäimistö ja hiiri sekä optiona valittava PlayStation Move -navigaatio-ohjain. Unity-pelimoottorin mukana tulee useita valmiita navigointiin tarkoitettuja scriptejä. Testeissä huomasimme, että nämä scriptit toimivat hyvin pelikäytössä, mutta eivät kovin hyvin CAVE-järjestelmien visualisointitarpeissa. Pelikäyttöön tarkoitetut navigointimallit olivat yleensä nopealiikkeisiä, eikä nopeuden säätelyä ollut mahdollista tehdä portaattomasti. CAVE-järjestelmien visualisointitavasta johtuen on tärkeää, että virtuaaliympäristöissä pystytään liikkumaan hyvin hitaasti nopeita liikkeitä välttämällä, jolloin minimoidaan käyttäjien mahdollinen liikehuimaus. Koska sopivia navigointiscriptejä halutuilla ominaisuuksilla ei löytynyt valmiina, päätettiin sellainen tehdä itse C# -kielellä.

Unity-pelimoottori sisältää valmiin syöttölaitteiden lukujärjestelmän, joten navigointiscriptillä oli helppo lukea syöttölaitteiden komentoja ja muuttaa tämä virtuaaliympäristössä tapahtuvaksi liikkeeksi. Scriptiin ohjelmoitiin liikkuminen joko hiirtä tai PlayStation Move -navigaatio-ohjainta käyttäen. Näppäimistöllä tapahtuva navigointi jätettiin pois, koska painonapeilla toteutettu portaaton nopeudensäätö navigoinnissa olisi ollut hankalakäyttöinen. Scriptin navigointiominaisuuksiksi toteutettiin liikkuminen eteen- ja taaksepäin, kääntyminen vaaka- ja pystysuunnassa sekä siirtyminen vaaka- ja pystysuunnassa. Navigointiscriptiin ohjelmoitiin myös muita käyttöä helpottavia ominaisuuksia, kuten alkutilaan palautuminen, katselusuunnan normalisointi, perspektiivipisteen korkeuden säätö sekä tasopinnan mukaan säätävä korkeussijainti. Navigointiscriptin toteutus on esitetty liitteessä seitsemän.

4.5 Lisäosien käyttö

Jotta lisäosien käyttö Unity-pelimoottorissa olisi mahdollisimman helppoa, oli lisäosat pakattava helposti käsiteltäväksi kokonaisuudeksi. Unity Editorissa luotiin

CAVE-järjestelmää kuvaava objekti, joka sisälsi kaikki toteutettujen ominaisuuksien tarvitsemat alaobjektit ja scriptit. Objekti asetukseineen tallennettiin Unity:n prefab-resurssiksi. Prefab on yhden tai useamman objektin ja niihin linkitettyjen scriptien ja resurssien muodostaman kokonaisuuden näköismalli. Prefab-resurssit helpottavat objektikokonaisuuksien käsittelyä ja käyttöä Unity Editorin sisällä. Jotta CAVE-lisäosat olisivat käytettävissä muissa Unity-projekteissa, pakattiin tarvittavat projektiasetukset, CAVE-prefab ja kaikki siihen linkitetty resurssit yhdeksi unitypackage-paketiksi. Unitypackage on pakattu tiedosto, jonka avulla Unity-resursseja on helppo siirtää eri Unity-projektien välillä. Kun CAVE-lisäosia halutaan käyttää uudessa Unity-projektissa, avataan unitypackage-tiedosto, jolloin paketin CAVE-prefab, resurssit ja projektiasetukset saadaan käyttöön projektissa. Kuvassa 7 on esitetty Unity Editor –näkyvä, jossa on CAVE-järjestelmää kuvaava objektikokonaisuus.



Kuva 7. CAVE-järjestelmää kuvaava objektikokonaisuus Unity Editor –näkyvässä.

5 YHTEENVETO

Visualisoinnin tarve eri teollisuudenaloilla kasvaa koko ajan. Visualisoinneilla saatavien tietojen ja kokemusten avulla on mahdollista tehostaa prosesseja ja säästää rakennus- ja ylläpitokustannuksissa. CAVE-järjestelmät tarjoavat hyvän työkalun monimutkaisten asiakokonaisuuksien visualisointiin ymmärrettävämmässä muodossa. Erikoisesta visualisointimuodosta johtuen on esitysalustana yleensä käytettävä juuri CAVE-järjestelmille suunniteltuja visualisointiohjelmistoja. Nykyaikaiset pelimoottorit tarjoavat vaihtoehdon näille ohjelmille visualisointisisällön näyttämiseksi CAVE-järjestelmissä.

Opinnäytyössä kehitettiin Unity-pelimoottorille ohjelmistolisäosat, joiden avulla on mahdollista tuottaa sisältöä CAVE-järjestelmiin. Lisäosien loppukäyttäjänä olivat Satavision Oy:n lisäksi heidän asiakkaansa, joten toteutuksessa oli kiinnitettävä erityishuomiota toimintavarmuuteen ja helppokäyttöisyyteen. Lisäosien kehitys sujui pääosin ilman ongelmia. Ainoastaan stereoskooppisen piirron toteutus aiheutti hieman hankaluuksia johtuen Unity:n tavasta suorittaa lisäosien ohjelmakoodia.

Opinnäytetyö onnistui kokonaisuudessaan hyvin. Käytännön ohjelmointityöhön ja testaukseen kului enemmän aikaa kuin oli alunperin suunniteltu. Opinnäytetyön tavoitteet täyttyivät kuitenkin hyvin ja kaikki vaadittavat lisäosat ja niiden ominaisuudet saatiin toteutettua ja koottua helposti käytettävään muotoon.

LÄHTEET

- Bruno Raffin & Luciano Soares. 2006. PC Clusters for Virtual Reality. Www-dokumentti. Saatavissa: <http://www.computer.org/csdl/proceedings/vr/2006/0224/00/02240215.pdf>. Luettu: 14.5.2014.
- Dave Pape. 2001. The CAVE Virtual Reality System. Www-dokumentti. Saatavissa: <http://www.evl.uic.edu/pape/CAVE/>. Luettu: 14.5.2014.
- Electronic Visualization Laboratory (EVL). 2002. The CAVE™ Virtual Reality Theater. Www-dokumentti. Saatavissa: <http://www.evl.uic.edu/core.php?mod=4&type=1&indi=161>. Luettu: 14.5.2014.
- i'm in VR. 2014. MiddleVR for Unity. Www-dokumentti. Saatavissa: <http://www.imin-vr.com/middlevr-for-unity/>. Luettu: 14.5.2014.
- Mechdyne. 2014. CAVE™. Www-dokumentti. Saatavissa: <http://www.mechdyne.com/cave.aspx>. Luettu: 14.5.2014.
- Mechdyne. 2013. CAVE2™ licensing. Www-dokumentti. Saatavissa: <http://www.mechdyne.com/article.aspx?id=238>. Luettu: 14.5.2014.
- NVIDIA. 2011. NVIDIA 3D VISION AUTOMATIC. Www-dokumentti. Saatavissa: http://developer.download.nvidia.com/whitepapers/2010/3D_Vision_Best_Practices_Guide.pdf. Luettu: 14.5.2014.
- NVIDIA. 2010. NVIDIA 3D VISION PRO AND STEREOSCOPIIC 3D. Www-dokumentti. Saatavissa: http://www.nvidia.com/docs/IO/40505/WP-05482-001_v01-final.pdf. Luettu: 14.5.2014.
- Paul Bourke. 1999. Calculating Stereo Pairs. Www-dokumentti. Saatavissa: <http://paulbourke.net/stereographics/stereorender/>. Luettu: 14.5.2014.
- Satavision Oy. 2014. Satavision-CAVE-järjestelmät. Sisäinen materiaali. Luettu: 14.5.2014.
- Unity Technologies. 2014a. Custom projection matrix. Www-dokumentti. Saatavissa: <https://docs.unity3d.com/Documentation/ScriptReference/Camera-projectionMatrix.html>. Luettu: 14.5.2014.
- Unity Technologies. 2014b. Unity AI. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/quality/ai>. Luettu: 14.5.2014.
- Unity Technologies. 2014c. Unity animation. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/animation>. Luettu: 14.5.2014.

Unity Technologies. 2014d. Unity editor. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/workflow/integrated-editor>. Luettu: 14.5.2014.

Unity Technologies. 2014e. Unity multiplatform. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/multiplatform>. Luettu: 14.5.2014.

Unity Technologies. 2014f. Unity performance. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/performance>. Luettu: 14.5.2014.

Unity Technologies. 2014g. Unity Plugins. Www-dokumentti. Saatavissa: <http://docs.unity3d.com/Documentation/Manual/Plugins.html>. Luettu: 14.5.2014.

Unity Technologies. 2014h. Unity public relations. Www-dokumentti. Saatavissa: <http://unity3d.com/company/public-relations>. Luettu: 14.5.2014.

Unity Technologies. 2014i. Unity rendering. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/quality/rendering>. Luettu: 14.5.2014.

Unity Technologies. 2014j. Unity scripting. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/workflow/scripting>. Luettu: 14.5.2014.

Unity Technologies. 2014k. Unity terrains. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/quality/terrains>. Luettu: 14.5.2014.

Wikipedia, vapaa tietosanakirja. 2014a. Active shutter 3D system. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/Active_shutter_3D_system. Luettu: 14.5.2014.

Wikipedia, vapaa tietosanakirja. 2014b. Cave automatic virtual environment. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment. Luettu: 14.5.2014.

Wikipedia, vapaa tietosanakirja. 2014c. Daniel J. Sandin. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/Daniel_J._Sandin. Luettu: 14.5.2014.

Wikipedia, vapaa tietosanakirja. 2014d. OpenNI. Www-dokumentti. Saatavissa: <http://en.wikipedia.org/wiki/OpenNI>. Luettu: 14.5.2014.

Wikipedia, vapaa tietosanakirja. 2014e. Stereoscopy. Www-dokumentti. Saatavissa: <http://en.wikipedia.org/wiki/Stereoscopy>. Luettu: 14.5.2014.