

Asmo Jussila

Unity-pelimoottorin hyödyntäminen CAVE-järjestelmässä

Opinnäytetyö

Kevät 2013

Tekniikan yksikkö

Tietotekniikan koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Tietoverkkotekniikka

Tekijä: Asmo Jussila

Työn nimi: Unity-pelimoottorin hyödyntäminen CAVE-järjestelmässä

Ohjaaja: Petteri Mäkelä

Vuosi: 2013

Sivumäärä: 39

Liitteiden lukumäärä: 2

Opinnäytetyö toteutettiin Seinäjoen ammattikorkeakoulun Tekniikan yksikön virtuaalilaboratorion CAVE-tilaan. Työn tarkoituksena oli uudistaa CAVE:ssa käytössä olevia vanhahtavia virtuaalitodellisuusohjelmia. Työssä selvitettiin kolmen eri pelimoottorin soveltuvuutta CAVE-järjestelmään.

Työn pelimoottorivertailuun valittiin suositut ja ilmaiseksi saatavilla olevat Cryengine 3 SDK, Unreal Development Kit sekä Unity-pelimoottorit. Pelimoottoreita vertailtiin keskenään graafisten ominaisuuksien, helppokäyttöisyyden, laajennettavuuden sekä CAVE-yhteensopivuuden kannalta. Pelimoottoritutkimuksen lopputuloksena Unity valittiin CAVE-järjestelmään parhaiten soveltuvaksi pelimoottoriksi.

Työn käytännön osuus koostui Unityllä luodusta demonstraatio-sovelluksesta. Unityä muokattiin siten, että se tuki useimpia CAVE-järjestelmän ominaisuuksia ja laitteita. Sovellus hyödynsi CAVE-järjestelmän stereoskooppista 3D-kuvaa, langattomia syöttölaitteita sekä PC-koneiden synkronointia lähiverkon kautta. Optisen paikannuksen tukea ei toteutettu, sillä se olisi vaatinut liian paljon resursseja.

Avainsanat: virtuaalitodellisuus, pelimoottori, stereoskooppinen 3D, CAVE, Unity

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Information Network Technology

Author: Asmo Jussila

Title of thesis: Utilizing Unity game engine in the CAVE-system

Supervisor: Petteri Mäkelä

Year: 2013

Number of pages: 39

Number of appendices: 2

This thesis was carried out at Seinäjoki University of Applied Sciences, School of Technology. The purpose of this work was to modernize the dated virtual reality programs of the CAVE system in the virtual laboratory. Three different game engines were researched for compatibility with the CAVE system.

Popular and free CryEngine 3 SDK, Unreal Development Kit, and Unity game engines were chosen for the game engine comparison. Game engines were compared with each other in graphical features, ease of use, scalability and compatibility with the CAVE. As a result of the game engine study, Unity was found out to be the most suitable game engine for the CAVE system.

The practical part of the work consisted of a demonstration application created in Unity. Unity was modified so that it supported the CAVE's features and devices. The application used the CAVE system for stereoscopic 3D effect, wireless input devices, and PC LAN synchronization. Optical tracking was not implemented because it would have required too many resources.

Keywords: virtual reality, game engine, stereoscopic 3D, CAVE, Unity

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
Kuvio- ja taulukkoluetelo.....	6
Käytetyt termit ja lyhenteet	7
1 JOHDANTO	8
1.1 Työn tausta	8
1.2 Työn tavoite	8
1.3 Työn rakenne	8
2 CAVE	9
2.1 CAVE-tila yleisesti.....	9
2.2 Laitteisto ja ohjelmisto.....	10
2.3 NaturalPoint OptiTrack, optinen paikannusjärjestelmä.....	14
2.4 Stereoskooppinen 3D-kuva.....	16
3 3D-PELIMOOTTORIT	18
3.1 CAVE-pelimoottorin valintakriteerit.....	18
3.2 Cryengine 3 SDK	18
3.2.1 Graafiset ominaisuudet	19
3.2.2 Sandbox-editori.....	20
3.3 Unreal Development Kit	22
3.3.1 Graafiset ominaisuudet	23
3.3.2 UnrealEd-editori	23
3.4 Unity.....	26
3.4.1 Graafiset ominaisuudet	27
3.4.2 Unityn editori	27
3.5 Projektiin sopivan pelimoottorin valinta	30
4 Unity-pelimoottori CAVE-järjestelmässä.....	31
4.1 Stereoskooppinen 3D-kuva Unityssä	31
4.2 Kamera- ja paikannusjärjestelmä	33
4.3 Synkronointi verkon kautta.....	34

5 Yhteenveto.....	37
LÄHTEET.....	38
LIITTEET.....	40

Kuvio- ja taulukkoluetelo

Kuvio 1. CAVE-tila	9
Kuvio 2. NaturalPoint OptiTrack V100:R2 -kamera.....	15
Kuvio 3. Sandbox-editorin pääikkuna.....	21
Kuvio 4. UnrealEd-editorin pääikkuna.....	25
Kuvio 5. Unityn pääikkuna	28
Kuvio 7. NVIDIA Control Panel	32
Kuvio 6. Demonstraatiosovelluksen aloitusruutu.....	35

Käytetyt termit ja lyhenteet

CAVE	Cave Automatic Virtual Environment. Järjestelmä, jolla luodaan todentuntuinen ympäristö virtuaalisesti.
Pelimoottori	Videopelien ohjelmarunko, joka kattaa yleisen pelimekaniikan ja vastaa muun muassa kappaleiden piirrosta näyttölaitteelle.
Stereoskooppinen 3D-kuva	Kolmiulotteinen kuva luodaan muodostamalla kummallekin silmälle oma kuva eri perspektiivistä.
PC-klusteri	Ryhmä PC-tietokoneita, joita käytetään yhtenä tietokone-laitteistona. PC-tietokoneet ovat toisiinsa yhteydessä lähiverkon kautta.
VR	Virtual Reality. Tietokonesimulaation tuottamien aistimusten avulla luotu keinotekoinen ympäristö.
VRPN	Virtual Reality Peripheral Network. Laitteistoriippumaton järjestelmä, joka tarjoaa rajapinnan virtuaalitodellisuuden oheislaitteisiin.
Renderöinti	Kuva, joka on luotu tietokoneohjelman avulla kolmiulotteisesta mallista.

1 JOHDANTO

1.1 Työn tausta

Opinnäytetyön teko alkoi halusta uudistaa ja parantaa Seinäjoen ammattikorkeakoulun Tekniikan yksikössä sijaitsevan CAVE-järjestelmän virtuaalitodellisuusohjelmia. CAVE-järjestelmässä oli käytössä nykystandardeilla jo vanhahtavia ja kömpelöitä ohjelmia. Vaihtoehtoisia ilmaisia, parempia ja helppokäyttöisempiä vaihtoehtoja oli saatavilla internetistä.

1.2 Työn tavoite

Opinnäytetyön tavoitteena oli muokata sopivaa 3D-pelimoottoria siten, että se toimisi sulavasti CAVE-järjestelmässä ja hyödyntäisi järjestelmän laitteistoa. Työssä oli tarkoitus hyödyntää CAVE:n paikannusjärjestelmää, 3D-stereografiikkaa, audiojärjestelmää sekä 3D-grafiikan piirtoa käyttäjän perspektiivistä. 3D-pelimoottorin käyttöönotolla pyrittiin parantamaan ja helpottamaan 3D-mallien ja virtuaalimaailmojen luomista CAVE-tilassa, sekä parantamaan grafiikan tasoa.

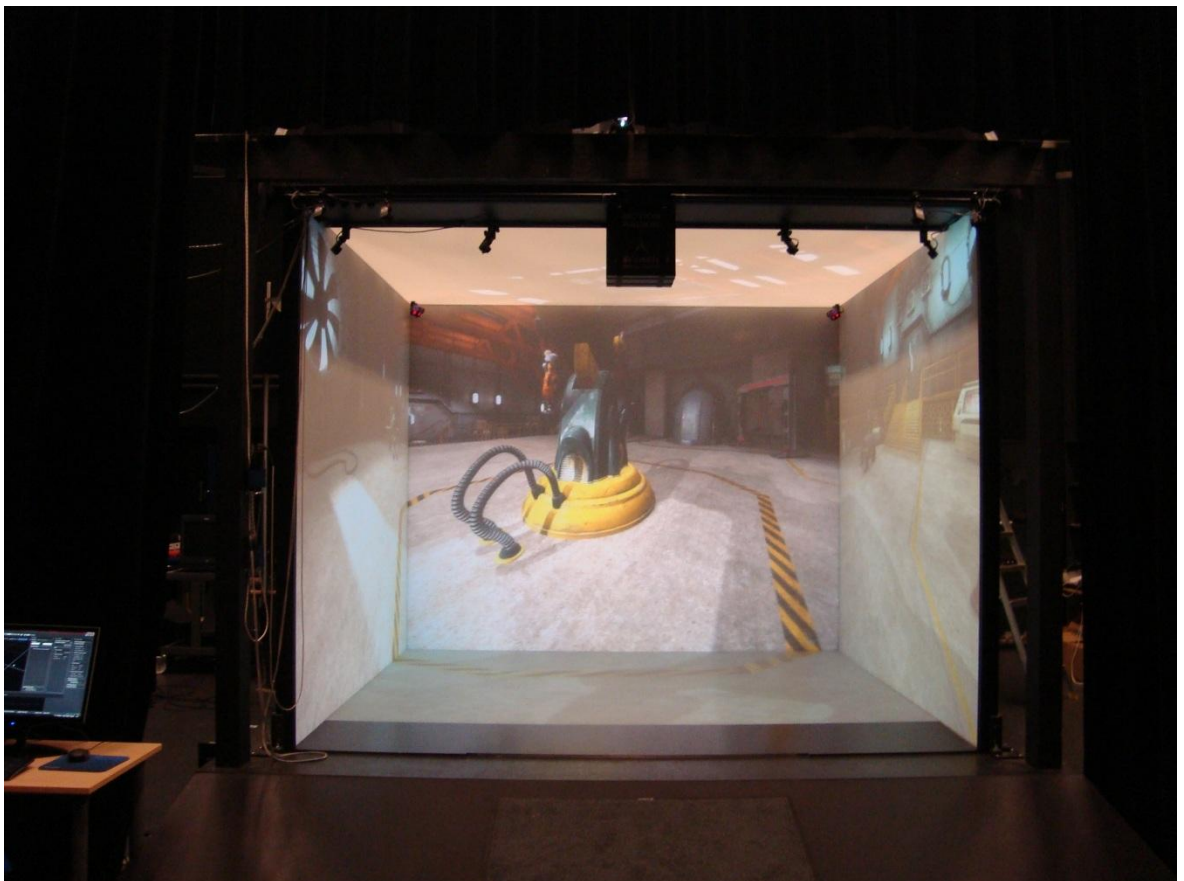
1.3 Työn rakenne

Toisessa luvussa kerrotaan CAVE-järjestelmän toiminnasta. Luvussa käsitellään järjestelmän oheislaitteita, näyttölaitteita, 3D-stereografiikan muodostusta, paikannusjärjestelmiä sekä PC-klusteria. Kolmannessa luvussa vertaillaan kolmea pelimoottoria ja arvioidaan niiden soveltuvuutta CAVE-järjestelmään. Luvussa perustellaan myös Unityn valintaa työssä käytettäväksi pelimoottoriksi. Neljännessä luvussa käydään läpi Unity-pelimoottorin muokkaus CAVE-järjestelmään sopivaksi.

2 CAVE

2.1 CAVE-tila yleisesti

Seinäjoen ammattikorkeakoulun Tekniikan yksikön VR-laboratoriossa sijaitseva CAVE-tila koostuu viidestä valkoisesta lasisesta projektioseinästä, jotka ympäröivät käyttäjän edestä, ylhäältä, alhaalta, vasemmalta ja oikealta. CAVE:n seinät ja katto ovat kooltaan 3,0 m x 2,4 m. Lattian projektiopinnan koko on 3150 x 2550 mm. Seinäjoen ammattikorkeakoulun CAVE on Suomen ainoa CAVE-tila, jossa on käytössä viisi projektiopintaa. (Virtuaalilaboratorio 2011.)



Kuvio 1. CAVE-tila.

CAVE-tilaa käytetään tietokoneella luotujen kolmiulotteisten mallien visualisoimiseen. Virtuaalitodellisuustekniikkaa pystytään soveltamaan monella osa-alueella, kuten tehdassuunnittelussa, matematiikassa, simuloinnissa, muotoilussa, arkkiteh-

tuurissa, rakennustekniikassa, lääketieteessä, geologiassa, tähtitieteessä, kulttuurissa, historiassa ja viihtessä. Järjestelmää voidaan käyttää lähes minkä tahansa kolmiulotteisen tiedon havainnollistamiseen. (Virtuaalilaboratorio 2011.)

Tekniikan yksikön CAVE-järjestelmällä on toteutettu muun muassa Seinäjoen kaupungin tilaama Virtuaali-Seinäjoki. Malli sisältää keskustan alueelta rakennusten 3D-malleja sekä Framin rakennusten 3D-mallit. Maaston tekstuuri on saatu ilmakuvasta ja rakennusten tekstuurit on saatu valokuvaamalla kyseisiä kohteita. Virtuaali-Seinäjokea kehitetään eteenpäin VR-laboratoriossa työharjoittelussa olevien opiskelijoiden toimesta. (SeAMK 2009.)

2.2 Laitteisto ja ohjelmisto

Tietokonejärjestelmät. CAVE:ssa on käytössä kaksi tietokonejärjestelmää. Uudemmassa järjestelmässä on kuusi Fujitsu Siemensin Celsius R670 -työasematietokonetta. Vanhemmassa järjestelmässä on viisi Fujitsu Siemensin Celsius R630 -työasematietokonetta.

Uudemman Celsius R670:n kokoonpano on:

- Intelin Xeon 3.33 GHz:n prosessori
- 12 GB DDR2 RAM-muistia
- NVIDIA Quadro 6000 -näytönohjain.

Vanhemman Celsius R630:n kokoonpano on:

- Intelin Xeon 3.2 GHz:n prosessori
- 3 GB DDR2 RAM-muistia
- NVIDIA Quadro FX 5500 -näytönohjain.

Nämä toisiinsa verkotetut tietokoneet muodostavat niin kutsutun PC-klusterin. Tietokoneet on sijoitettu päällekkäin laitekaappiin, joka sijaitsee ilmastoidussa huoneessa. Jokaisella PC-laitteella on omat CRT-monitorit, jotka ovat kytketty Extron CrossPoint 128HVA -matriisikytkimen kautta. Matriisikytkimellä voi vaihtaa kuvan lähdettä kahden PC-klusterin välillä. Kaikki järjestelmän PC-koneet ovat yhdistet-

tyinä internetiin sekä lähiverkkoon. Klusterissa yksi tietokone on niin sanottu konsolikone, josta ohjataan muita klusterin tietokoneita. Konsolikone myös synkronoi stereokuvan tahdistuksen. Klusterin tietokoneiden näytönohjaimien kuvan synkronointi tapahtuu erillisellä tietokoneisiin liitetyllä synkronointikortilla. Oheislaitteet ovat myös yhdistetty konsolikoneeseen. Klusterin etuina on muun muassa tietokoneiden prosessorien tai näytönohjaimien laskentatehojen yhdistäminen.

Projektiio. Kuva luodaan viidelle projektiioseinälle viidellä Christie Digitalin Mirage 4000 DLP -projektorilla. Projektorit toimivat 1280 x 1024 resoluutiolla ja ne pystyvät näyttämään aktiivista stereoskooppista 3D-kuvaa korkeintaan 108 Hz:n taajuudella. Projektorit painavat 40 kg. Projektorien näyttämä kuva heijastuu kaikille projektiopinnoille peilipintojen kautta. Tällä saavutetaan pienempi projektorien asennusetäisyys ja parempi säädettävyys.

Projektorissa on seuraavia ominaisuuksia:

- valoteho 4000 ANSI lumenia
- 300:1 ANSI kontrasti
- 700 W CERMAX Xenon -lamppu
- linssin siirto
- SXGA resoluutio
- 3D-stereomoduuli
- 3-piirinen digitaalinen mikropeilipiiri
- etäohjaus PC-laitteesta COM-portin avulla.

(Christie Digital Systems 2013.)

3D-lasit. Jotta stereoskooppisen 3D-efektin pystyy havainnoimaan, täytyy CAVE-tilan käyttäjällä olla käytössään stereografiikkaa tukevat aktiiviset 3D-lasit. CAVE:ssa on käytössä RealD CrystalEyes 4s -lasit, joihin on kiinnitetty neljä merkkiä. Optinen paikannusjärjestelmä pystyy paikantamaan nämä merkit.

CrystalEyes 4s -laseissa on seuraavia ominaisuuksia:

- paino 67 grammaa
- USB-liitännän kautta ladattava sisäänrakennettu akku
- 35 tunnin käyttöaika

- lasit sammuttavat itsensä 5 minuutin kuluttua, jos signaalia ei ole saatavilla
 - ulkoisten valonlähteiden signaalinhäirinnän sieto
 - vaihdettava kuminen nenäpidike
 - 1000:1 kontrasti
 - 1.7 millisekunnin vasteaika
 - tuki maksimissaan 144 Hz:n taajuudelle.
- (RealD [viitattu 11.3.2013].)

Äänijärjestelmä. CAVE:n tiläänijärjestelmä koostuu kahdeksasta JBL Control 28AV -pääkaiuttimesta, kahdesta JBL-subwooferista, Yamaha 01V96 -esivahvistimesta, pääkaiuttimien vahvistimesta ja subwooferien vahvistimesta.

Yamaha 01V96 -esivahvistimen ominaisuuksia ovat:

- 40 kanavan miksaus
 - 16 mikrofoni/linjasisääntuloa
 - 24 bittinen A/D- ja D/A-muunnin
 - digitaalinen sisääntulo
 - moottoroidut äänenvoimakkuuden säätimet
 - ekvalisaattori
 - 320 x 240 pikselin LCD-näyttö.
- (Yamaha Corporation 2013.)

Pääkaiuttimet on sijoitettu CAVE-tilan ulkopuolelle. Neljä pääkaiutinta on sijoitettu etunäytön vasemmalle ja oikealle puolella. Loput neljä pääkaiutinta sijaitsevat avoimen seinän vasemmalla ja oikealla puolella. Konsolikone lähettää analogisen äänisignaalin esivahvistimelle, joka puolestaan lähettää signaalin optisena pääkaiuttimien vahvistimelle ja subwoofer-vahvistimelle.

Syöttölaitteet. CAVE:n käyttäjä pystyy liikkumaan virtuaalimaailmoissa käyttämällä Playstation Move navigation controller -peliohjainta. Peliohjain toimii langattomalla Bluetooth-tekniikalla ja sisältää ladattavan akun. Peliohjaimen kärkeen on lisätty jäykkä kappale, johon on kiinnitetty viisi merkkiä optista paikannusta varten.

Lisäksi käytössä on myös Logitech Wireless Gamepad F710 -peliohjain, johon on myös kiinnitetty viisi merkkiä.

Ohjelmisto. Kolmiulotteisen datan visualisointiin on käytössä seuraavia Windows PC-ohjelmistoja: VRCO VRScene, TreeC VR4MAX Extreme ja 3DVIA Virtools 5. Ohjelmille ominaista on 3D-mallien lataaminen sekä toimintojen luominen 3D-ympäristöön. Kaikissa ohjelmissa on sisäänrakennettu tuki CAVE-järjestelmälle.

CAVE:ssa eniten käytetty ohjelma on VR4MAX Extreme. Ohjelma tukee passiivista ja aktiivista stereoskooppista 3D-kuvaa. Ohjelma tukee kattavaa määrää eri paikannus- ja osoitinlaitteita. Graafisilta ominaisuuksilta VR4MAX Extreme on vaatimaton ja käyttöliittymä nykystandardeilla vanhahtava ja hankalakäyttöinen.

VR4MAX Extreme tukee prosessorin monisäikeisyys-tekniikkaa, joka nopeuttaa esimerkiksi fysiikan laskentaa, tilaäänen prosessointia ja simulaatioita. Kuvan piirto voidaan toteuttaa mille tahansa projektiio-asennuksille. Ohjelma tukee VR konferenssia, jossa usea käyttäjä voi navigoida reaaliaikaisesti virtuaalitodellisuusympäristössä mistä päin maapalloa tahansa. (Tree C Technology [viitattu 25.4.2013].)

CAVE:ssa käytetään myös Virtools 5-ohjelmaa. 3DVIA Virtools 5-ohjelma muistuttaa käytöltään suuresti myöhemmin vertailtavia pelimoottoreita. Sen graafiset ominaisuudet ja käyttöliittymä ovat hieman VR4MAX Extremeä paremmat. Ohjelma tukee aktiivista stereoskooppista 3D-kuvaa ja kattavaa määrää eri paikannus- ja osoitinlaitteita. (Dassault Systèmes [viitattu 24.4.2013]).

Virtoolsin editorilla pystyy navigoimaan, luomaan, editoimaan, valitsemaan ja manipuloimaan 3D-malleja, valoja kameroita ja reittejä. Virtools 5 sisältää Behavior Engine -työkalun, jolla voi hallita tapahtumia ja objekteja kuten kameroita, hahmoja, törmäystunnistusta, kontrollereita, valoja, logiikkaa, materiaaleja, tekstuureja, partikkeleja, ääniä, varjostimia ja 3D-mallien muokkausta. (Dassault Systèmes [viitattu 24.4.2013].)

2.3 NaturalPoint OptiTrack, optinen paikannusjärjestelmä

CAVE:ssa on käytössä kaksi paikannusjärjestelmää. Ascension Technology Corporationin Flock of Birds magneettinen paikannusjärjestelmä sekä NaturalPointin OptiTrack optinen paikannusjärjestelmä. Vanhempaa magneettipaikannusta käytetään varajärjestelmänä.

CAVE-tilan optinen paikannusjärjestelmä koostuu kahdestatoista NaturalPoint OptiTrack V100:R2 -kamerasta, jotka lähettävät infrapunavaloa ja kuvaavat kennoonsa merkeistä takaisin heijastuvaa valoa. Neljä kameraa on sijoitettu CAVE-tilan etuosaan ylävasemmalle ja yläoikealle. Kahdeksan kameraa on sijoitettu tilan taaksaosaan ylävasemmalle ja yläoikealle. Kamerat on suunnattu alaviistoon ja ne ovat käännettyinä siten, että ne näkevät mahdollisimman kattavasti CAVE-tilan.

Kameroiden ohjausyksikkö on liitettyä USB-johdolla tietokoneeseen, jolla käytetään Naturalpointin Motive-ohjelmaa. Motivella pystyy esimerkiksi luomaan paikannetuista merkeistä jäykkiä kappaleita tai kaappaamaan luuranko-animaatiota. Motive lähettää dataa lähiverkkoon käyttäen VRPN- ja Unicast-protokollaa.

OptiTrack paikannusjärjestelmän kamerat havaitsevat parhaiten pallon muotoiset valoa heijastavat merkit. Pienempien merkkien halkaisija on noin 1 cm ja isompien 2 cm. Käytännössä mikä tahansa valoa heijastava pinta sopii paikannettavaksi järjestelmälle.

Kameran lähettämä infrapunavalokimppo heijastavasta merkistä takaisin kameraan. Kamerat voivat parhaimmillaan kuvata palloja sata kertaa sekunnissa. Tämän tiedon perusteella Motive-ohjelma pystyy laskemaan kolmiulotteisen paikan kyseiselle merkille. Vähintään kahden kamerasa täytyy nähdä merkki tai laskentaa ei voida suorittaa.



Kuvio 2. NaturalPoint OptiTrack V100:R2 -kamera.
(NaturalPoint 2013)

Joskus paikannettavat merkit voivat kadota kameroiden näkökentästä. Käyttäjä voi omalla kehollaan peittää merkkien näkyvyyden tai tiellä voi olla jokin muu este. Tästä syystä kamerat pyritään sijoittelemaan siten, että ne ympäröisivät käyttäjän mahdollisimman hyvin joka suunnalta, jotta merkkien näkyminen parane.

Jäykkiä paikannettavia esineitä kutsutaan rigidbody-objekteiksi. Esineiden paikannukseen käytetään merkkejä, jotka pysyvät varmasti paikoillaan. Jos rigidbody-objektista halutaan saada suunta-, paikka- ja orientaatio-dataa, täytyy esineeseen kiinnittää vähintään kolme merkkiä.

Merkit kannattaa sijoitella epäsymmetrisesti, jotta laskenta ei sekoita lähekkäin olevia jäykkiä kappaleita toisiinsa. Jos jokin esineen merkki jostain syystä vaihtaa paikkaa tai siirtyy, ei paikannus ole enää tarkkaa ja rigidbody-objektia pitää korjata Motive-ohjelmassa joko luomalla objekti uudelleen tai korjaamalla vanhaa.

CAVE-tilassa käytettäviin 3D-stereolaseihin ja Playstation Move navigation controller -peliohjaimeen kiinnitetyistä merkeistä on luotu Motive-ohjelmassa rigidbody-objektit, joiden paikka-, suunta- ja orientaatio-dataa voi lähettää lähiverkon kautta VRPN- tai Multicast-protokollaa käyttäen.

Optisen paikannuksen hyötyjä ovat:

- datan tarkkuus
- reaaliaikaisuus
- langattomuus
- mahdollistaa monen eri esineen paikannuksen samaan aikaan
- merkkejä voi käyttää useita samaan aikaan
- paikannettavat esineet voivat liikkua vapaasti paikannusalueella.

Optisen paikannuksen haittoja ovat:

- paikannus vaatii laskentatehoa
- paikannettava merkki vaatii suoran kamerayhteyden
- merkit saattavat hävitä kameroiden katvealueisiin
- kamerat saattavat sekoittaa ulkopuolisia heijastavia pintoja merkkeihin.

2.4 Stereoskooppinen 3D-kuva

Kolmiulotteisuus on yksi olennaisimmista osista CAVE-käyttökokemuksessa. Stereoskooppinen 3D-kuva sallii CAVE:n käyttäjän eläytyä virtuaalitodellisuusympäristöön aivan kuin hän olisi oikeasti läsnä luodussa ympäristössä.

CAVE-järjestelmä käyttää aktiivista stereoskooppista 3D-kuvaa. Aktiivisessa järjestelmässä illuusio kolmiulotteisuudesta luodaan näyttämällä kuvia vuorotellen käyttäjän vasemmalle ja oikealle silmälle (Samuel Gateau 2011). Kuvien piirto tapahtuu kymmeniä kertoja sekunnissa (Samuel Gateau 2011).

3D-ohjelmat luovat kolmiulotteisuuden vaikutelman siirtämällä, vaihtamalla tai luomalla uudelleen virtuaalisen kameran vasemmalle ja oikealle silmälle. Tällöin 3D-malli tai maailma piiryy hieman eri perspektiivistä. Virtuaalisen kameran liikkeen täytyy vastata ihmisen silmien etäisyyttä toisistaan tai stereoeefektistä muodostuu epäselvä. (Samuel Gateau 2011.)

CAVE:ssa käytössä olevat CrystalEyes 4s -lasit käyttävät nestekide-tekniikkaa. Lasit saavat synkronointisignaalin useasta infrapunalähettimestä, jotka on sijoitettu projektiopintojen taakse. Lähetin lähettää signaalin aina kun kuva piirretään. 3D-

lasien infrapunavastaanotin vastaanottaa signaalin ja pimentää joko oikean tai vasemman silmän nestekidekalvon.

Kuvan piirto tapahtuu CAVE:ssa maksimissaan 108 kertaa sekunnissa, jolloin sekunnin aikana kumpikin silmä näkee 54 kuvaa. Vertailun vuoksi kaksiulotteisissa elokuvissa näytetään yleensä 24 kuvaa sekunnissa.

3 3D-PELIMOOTTORIT

3.1 CAVE-pelimoottorin valintakriteerit

CAVE:n käytössä olevat VRCO VRScene-, TreeC VR4MAX Extreme- ja 3DVIA Virtools -ohjelmat ovat nykymittapuulla vanhahtavia ja hankalakäyttöisiä. Vaihtoehtoista ohjelmaa lähdettiin etsimään pelimoottoreista. Projektin toteutukseen haettiin ilmaiseksi saatavilla olevia 3D-pelimoottoreita.

3D-pelimoottori ja 3D-moottori eroavat yleensä siten toisistaan, että 3D-pelimoottoreissa tulee mukana erillinen editori sekä muita työkaluja kehittämistyöhön, kun taas 3D-moottorista voi olla saatavilla pelkkä koodirunko, johon voi itse ohjelmoida kehittämistyötä helpottavat editorit ja työkalut. (Alex Galuzin 2012.)

Pelimoottorin valinnassa kriteereinä olivat muun muassa helppokäyttöinen graafinen editori ja helposti muokattavissa tai lisättävissä olevat toiminnot. Pelimoottorin täytyy myös pystyä näyttämään stereoskooppista 3D-grafiikkaa. Erityistä huomiota kiinnitettiin pelimoottorin graafisiin ominaisuuksiin ja näyttävyyteen. Lisäksi pelimoottorin täytyi olla ilmaiseksi saatavilla.

Julkisia ilmaisia 3D-pelimoottoreita on olemassa noin 30 kappaletta. Ilmaisella tässä tarkoitetaan sellaista pelimoottoria, jolla voi kehittää 3D-sovelluksia omaan käyttöön ilman erillisiä lisenssimaksuja. Kolme 3D-pelimoottoria esikarsiutui selvästi edukseen yli muiden:

- Cryengine 3 SDK, tekijänä Crytek
- Unreal Development Kit, tekijänä Epic Games
- Unity, tekijänä Unity Technologies.

3.2 Cryengine 3 SDK

Crytekin tekemän Cryengine-pelimoottorin kehitys alkoi vuonna 2000. Far Cry oli ensimmäinen kaupallinen peli, joka hyödynsi pelimoottorin ensimmäistä versiota. Vuonna 2007 Crytek julkisti Cryengine 2:n jota oli kehitetty Cryengine 1:n pohjalta.

Vuonna 2009 Crytek julkisti Cryengine 3:n, joka on opinnäytetyön kirjoitushetkellä kaikista uusien versio kyseisestä pelimoottorista. (Nicholas Werner 2011.)

Crytek julkaisi elokuussa vuonna 2011 Cryengine 3 SDK:n. Tämä paketti sisältää itse Cryengine 3 pelimoottorin sekä Sandbox-editorin, joka sisältää itsessään erinäisiä editoreja ja työkaluja. Cryengine 3 SDK:n voi kuka tahansa yksityinen henkilö ladata, käyttää ilmaiseksi ja luoda pelejä tai 3D-sovelluksia käyttäen samanlaisia työkaluja, joilla Crytek on työstänyt omia pelejään. Cryengine 3 SDK on tällä hetkellä tarkoitettu ainoastaan PC-käyttöön. (Nicholas Werner 2011.)

3.2.1 Graafiset ominaisuudet

Cryengine 3 SDK sisältää edistyneitä graafisia ominaisuuksia. Sen kehittynyt valaistusmoottori pystyy muun muassa näyttämään lähes realistista luonnollista valoa. Valaistusmoottori kykenee luomaan pehmeitä varjoja, jotka dynaamisesti ja reaaliaikaisesti vastaavat luonnolliseen liikkeeseen. (Crytek 2013.)

Valaistusmoottori sisältää myös dynaamisen ja reaaliaikaisen globaalin valaistuksen, joka on optimoitu nykyisille ja tuleville kehitysalustoille. Käyttäjä näkee reaaliajassa valon pomppimisen, värivuodot ja heijastustehosteet ilman esilaskentaa tai geometrisia rajoituksia niin staattisissa kuin dynaamisissakin kappaleissa. (Crytek 2013.)

Säteilyvoimakkuusjärjestelmä pystyy luomaan suorituskykyisen valaistusalueen, joka kykenee simuloimaan valon säteilyä ja lisäämään väriä heijastettuun valoon. Kehittäjät voivat lisätä loputtomasti valoja tämän valaistusalueen sisään. (Crytek 2013.)

Valon etenemisen volyyymi on skaalautuva tekniikka epäsuoran valaistuksen näyttämiseen. Tekniikka käyttää ristikoita ja palloharmoniaa näyttääkseen spatiaalisen ja kulmanjakautumissuodattuneen valon 3D-ympäristössä. Se on tarkoitettu käytettäväksi pääasiallisesti kerran kimmonneeseen epäsuoraan valaistukseen, mutta sitä voi laajentaa käsittelemään useita kertoja kimmonnutta valoa. (Crytek 2013.)

Partikkelijärjestelmä tukee täyttä liikkeensumentumisefektiä. Partikkelijärjestelmä voi myös vastaanottaa varjoja ja valaistustietoa ympäristön antureista ja globaalilta valaistukselta, mikä parantaa partikkelien yhdenmukaisuutta. (Crytek 2013.)

Cryengine 3 SDK tukee DirectX11-yhteensopivaa laitteistokiihdytettyä tesselaatiota. Tesselaation avulla 3D-muoto voidaan tehdä bittikartasta. SDK tukee kolmea erityyppistä tesselaatiota, joita voi sekoittaa keskenään. Tesselaation käyttäminen parantaa graafista ulkonäköä. (Crytek 2013.)

Reaaliaikaiset heijastukset ovat yksi suurimpia haasteita reaaliaikaisessa kuvanpiirroksessa, jos käytetään viivästettyä piirtoa ja valaistusmoottoria. Cryengine 3 SDK on pyrkinyt ohittamaan tämän ongelman käyttämällä reaaliaikaista paikallista heijastusta. Tämä tekniikka antaa likiarvoisen sädeseuratun HDR-heijastuksen paikallisesti kappaleille. Tämä tekniikka sallii minkä tahansa kaarevan pinnan heijastaa ympäristöään reaaliajassa. (Crytek 2013.)

Limittäinen näkyvyyskartta on jälkikäsitteleyefekti, joka lisää kappaleisiin omia varjoja ja lisää pintojen syvyysvaikutelmaa. Tämä tekniikka on kuvanpiirrolle edullisempi vaihtoehto verrattuna tesselaatioon. (Crytek 2013.)

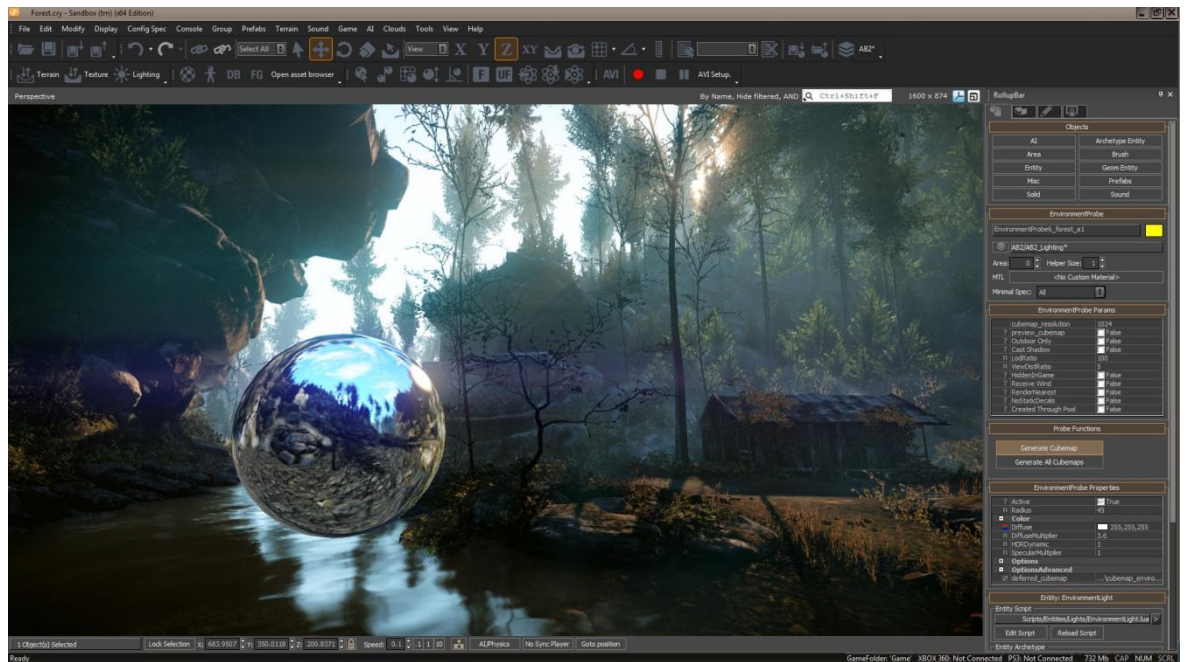
Silmien sopeutumistekniikkaa käytetään simuloimaan ihmisen silmän reaktiota äkillisille ja suurille valaistuksen muutoksille, kuten siirtymiselle pimeistä sisätiloista yllättäen aurinkoiseen ulkotilaan. Laajan dynamiikan renderöinti sallii realistisen renderöinnin maisemissa, joissa on äärimmäisiä kontrastin ja kirkkauden vaihtelevuutta. (Crytek 2013.)

Cryengine 3 SDK:n varjostimilla voidaan näyttää esimerkiksi kappaleiden pintoja märkänä, mutaisina tai jäätyneinä. Tuettuja ovat myös reaaliaikainen pikselintarkka valaistus, kuhmuiset heijastukset, valon taittuminen, hohto-efekti ja animoidut tekstuurit. (Crytek 2013.)

3.2.2 Sandbox-editori

Sandbox-editori on niin sanottu WYSIWYP-editori (What You See Is What You Play) eli käyttäjän tekemät pelimaailman muutokset tapahtuvat reaaliajassa ja gra-

fiikka näyttää editorissa samalta kuin se näyttäisi itse pelin sisällä. Sandbox-editorista voi hypätä peliin saumattomasti ilman lataustaukoja. (Crytek 2013)



Kuvio 3. Sandbox-editorin pääikkuna. (Crytek 2013)

Cryengine 3 SDK:n Sandbox-editori sisältää muun muassa seuraavia ominaisuuksia ja työkaluja:

- fysiikkamoottori
- Flowgraph - visuaalinen editor, joka poistaa tarpeen ohjelmoinnille
- stereoskooppinen 3D-tuki
- materiaalieditori
- hahmojen tekoälyeditori
- animaatioeditori
- maastoeditori. (Crytek 2013.)

Sandbox-editorin ehdottomia vahvuuksia ovat sen sisäänrakennettu helpokäyttöinen maastoeditori sekä Flowgraph-editori. Maastoeditorilla pystyy vaivattomasti luomaan maksimissaan 256 neliökilometrin kokoisen maaston. Maaston voi myös tuoda editoriin mustavalkoisena korkeuskarttana, joka on bitmap-muodossa. Maastoeditori toimii siten, että kursorin kohdasta maasto joko nousee tai laskee riippuen valitusta työkalusta. (Crytek 2013.)

Flowgraph-editori vähentää C++-ohjelmoinnin ja skriptien kirjoittamisen tarvetta. Erityisesti graafikot ja sisällön suunnittelijat suosivat tämänlaisia editoreja, koska niillä voidaan pienellä vaivalla kontrolloida ja luoda pelin tai sovelluksen toimintoja, pelilogiikkaa sekä kuva- ja ääniefektejä. Flowgraph-editorilla voi suunnitella jopa kokonaisia kenttiä. Editori toimii vedä ja pudota -menetelmällä. (Crytek 2013.)

3D-mallien tuonti Sandbox-editoriin tapahtuu Crytekin oman CryExport-liitännäisen kautta, joka on saatavilla Autodesk 3ds Maxiin, Autodesk Mayaan ja Autodesk Softimageen. 3D-mallin tekstuuritiedostot pitää tuoda Sandbox-editoriin erikseen Adobe Photoshopiin tehdyn CryTIF-liitännäisen kautta. (Crytek 2013.)

Syvällisempi kehittäminen Cryengine 3 SDK:lla vaatii C++-ohjelmoinnin osaamista. C++-ohjelmointi on välttämätöntä tilanteissa, joissa käyttäjä haluaa luoda ominaisuuksia joita ei Flowgraph-editorilla pysty tekemään. (Crytek 2013.)

3.3 Unreal Development Kit

Epic Games julkaisi vuonna 1998 Unreal 1 -pelin, joka toimii yrityksen kehittämän Unreal-pelimoottorin päällä. Unreal-peli oli suosittu etenkin pelinmuokkaajien keskuudessa, koska sitä pystyi muokkaamaan skriptien avulla. Unreal-pelin mukana tuli myös UnrealEd-editori, jolla pystyi luomaan ja muokkaamaan kenttätiedostoja. (Chris Plante 2012.)

Unreal-pelimoottori kehittyi vuosien varrella, kunnes vuonna 2009 Epic julkaisi ei-kaupalliseen ja opetustarkoitukseen vapaasti käytettävissä olevan kehitystyökalun nimeltä Unreal Development Kit, lyhennettynä UDK. UDK pystyy kääntämään sillä luodut 3D-sovellukset Windows- ja iOS-käyttöjärjestelmille. (Chris Plante 2012.)

UDK:n etuina muihin vastaaviin kehitystyökaluihin verrattuna on sen saatavilla oleva laaja dokumentaatio ja suuri foorumiyhteisö. UDK on tämän tyyppisistä kehitystyökaluista vanhin.

3.3.1 Graafiset ominaisuudet

UDK:n grafiikan renderöijänä toimii prosessorin monisäikeistystä hyödyntävä Gemini-järjestelmä. Gemini antaa käyttäjälle käyttöön 64-bittisen HDR-renderöintilinjan, jonka avulla on mahdollista ajaa fotorealistisia simulaatioita. Gammakorjattu, lineaarisen väriavaruuden tarjoama renderöijä näyttää tarkasti värit ja tukee samalla suurta määrää jälkikäsitteilyefektejä, kuten liikkeensumentumista, syvyysterävyysaluetta, hehkua ja ympäristön okklusiota. (Epic Games 2013.)

UDK tukee kaikkia moderneja pikselintarkkuudella suoritettavia valaistus- ja renderöintitekniikoita, kuten normaalikartoitettua parametristä Phong-valaistusta, artistien tekemiä kustomoituja valaistumalleja anistrooppisella suodatuksella, virtuaalista siirtokarttaa, valon heikentymisen funktioita ja esilaskettuja varjomaskeja ja suuntaavia valokarttoja. (Epic Games 2013.)

Kamerat, volyymit ja läpinäkyvien objektien toiminnot käsitellään kaikki pikselin tarkkuudella. Luodut maailmat voivat sisältää monikerroksisia, globaaleita korkeuden mukaan määräytyviä sumu- ja sumuvolyymi-efektejä. Tuettuna on myös erittäin suorituskykyinen tekstuureiden suoratoisto. (Epic Games 2013.)

Valaistuksen osalta UDK tarjoaa tekstuureihin sisällytettyä staattista valaistustietoa Lightmass-valaistuksenlaskijalla. UDK-pelimooottori tukee täyttä dynaamista varjostusta ja suuria määriä eri valonlähteitä. (Epic Games 2013.)

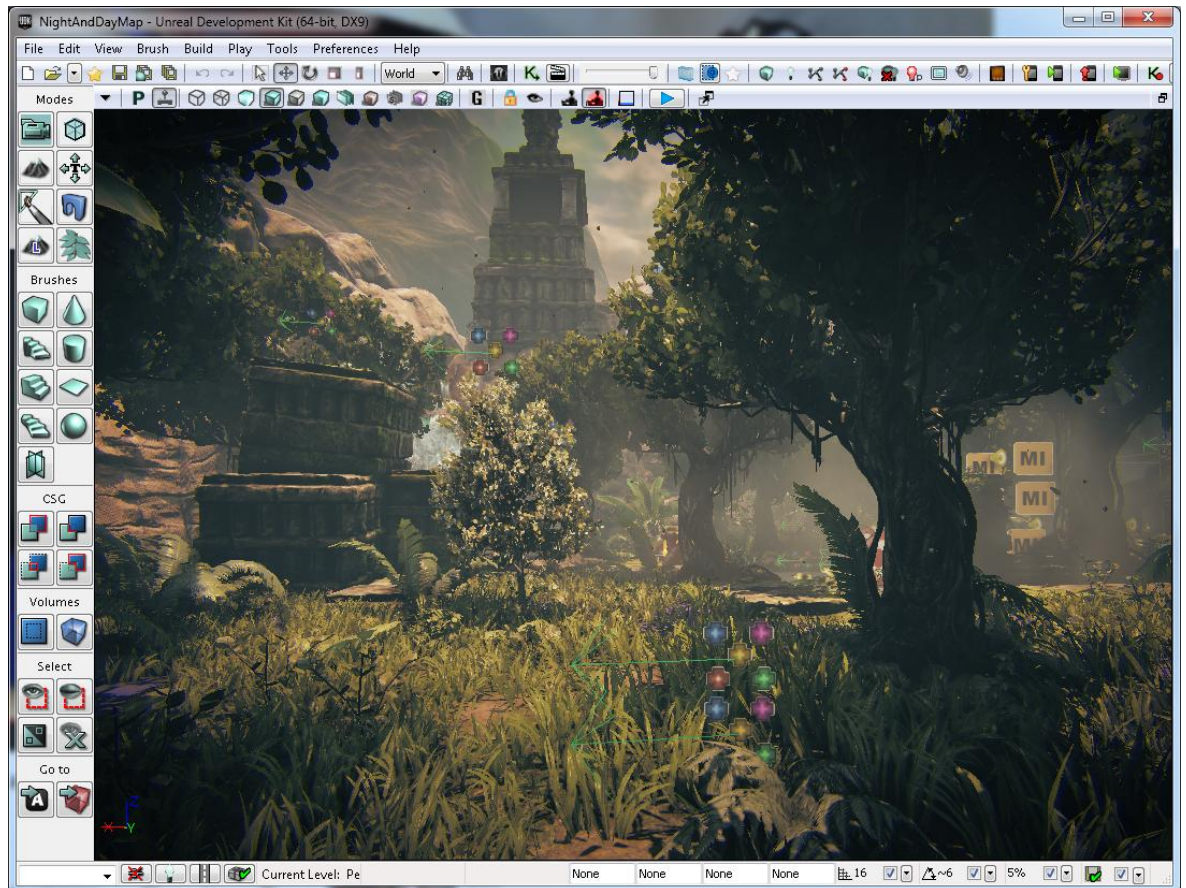
UDK:n DirectX 11 -ominaisuuksina on muun muassa viivästetty renderöinti, jonka avulla suuret määrät valaistuksen laskentaa voidaan toteuttaa resursseja säästäen. Lisäksi DirectX 11 -ominaisuuksina on tauluheijastumiset, joiden avulla voidaan laskea monimutkaisia heijastumisia staattisista ja dynaamisista varjoista prosessoria juurikaan kuormittamatta. (Epic Games 2013.)

3.3.2 UnrealEd-editori

UDK sisältää Unreal-pelimooottorin päälle rakennetun editorin nimeltä UnrealEd. Editori sisältää muun muassa seuraavia työkaluja ja ominaisuuksia:

- fysiikkamoottori
- Kismet – visuaalinen skriptaja
- stereoskooppinen 3D-tuki
- materiaalieditori
- hahmojen tekoälyeditori
- animaatioeditori
- maastoeditori
- partikkelieditori
- resurssieditori
- Lightmass - valaistuksen laskijatyökalu. (Epic Games 2013.)

UnrealEd-editori toimii samalla tavalla kuin Cryengine 3 SDK:n Sandbox-editori. Käyttäjän tekemät muutokset tapahtuvat reaaliajassa ja grafiikka näyttää editorissa samalta kuin se näyttäisi itse 3D-sovelluksen tai pelin sisällä. UnrealEdistä voi hypätä peliin tai sovellukseen saumattomasti ilman lataustaukoja. (Epic Games 2013.)



Kuvio 4. UnrealEd-editorin pääikkuna.

UnrealEdin vahvuuksiin kuuluu sen sisältämä Kismet-editori, joka toimii samalla periaatteella kuin Cryengine 3 SDK:n Flowgraph-editori.

Kismet-editorilla voi periaatteessa suunnitella tai luoda prototyypejä kokonaisista peleistä tai sovelluksista, jotka sisältävät käyttäjän interaktion ja toiminteita. Kismet-editori toimii vedä ja pudota -menetelmällä. (Epic Games 2013.)

UDK sisältää nopean ja tarkan Lightmass-valaistuksen laskijan. Lightmass laskee simuloidun auringon fotonien kimpoamista eri materiaaleista objektien pinnoilla. Lightmass tallentaa tämän tiedon erillisiin valaistustekstuuritiedostoihin. Laskenta-prosessi ei toimi reaaliajassa. (Epic Games 2013.)

3D-mallien tuonti UnrealEdiin tapahtuu Autodeskin luoman FBX-liitännäisen avulla. Liitännäinen on saatavilla Autodesk 3ds Max:iin, Autodesk Mayaan ja Autodesk Softimageen. FBX-liitännäinen tallentaa alkuperäisen 3D-tiedoston FBX-tiedostoformaattiin, jonka jälkeen sen voi viedä UnrealEdin resurssieditoriin. FBX-

tiedostosto sisältää 3D-mallin, mallin mahdollisen animaatiotiedoston, materiaalit ja tekstuurit. (Epic Games 2013.)

UDK:lla tehtävään syvällisempään kehitystyöhön vaaditaan UnrealScript-skriptikielen osaamista. UnrealScript-skriptiaamista tarvitaan etenkin tilanteissa, joissa käyttäjä haluaa luoda ominaisuuksia ja toiminteita, joita ei Kismet-editorilla pysty tekemään, tai se olisi liian aikaa vievää ja hankalaa. UnrealScriptin sulava käyttö edellyttää ohjelmointiosaamista Java-kielestä. (Epic Games 2013).

3.4 Unity

Unity on Unity Technologiesin kehittämä monialustainen pelimoottori. Pelimoottori on ollut kehityksessä vuodesta 2001 asti. Unity on opinnäytetyön kirjoitushetkellä versionumerossa neljä. Unityllä kehitystä voi tehdä Windows- tai OS X -käyttöjärjestelmillä. Unity tukee Android-, Windows- Linux-, iOS-, PlayStation 3-, Xbox 360-, WiiU- ja OS X -järjestelmille kehittämistä. (Unity Technologies 2013.)

Unity-pelimoottorista on saatavilla perusversio ja Pro-versio. Unityn perusversio on ilmainen ja vapaasti ladattavissa. Unityn perusversiota voi käyttää kaupalliseen ja ei-kaupalliseen käyttöön. Perusversiolla käännetyt pelit ja sovellukset näyttävät Unityn logon käynnistyksen yhteydessä. (Unity Technologies 2013.)

Pro-versio on maksullinen. Pro-lisenssistä on myös saatavilla 30 päivän kokeiluversio. Pro-versiossa on seuraavia ominaisuuksia, joita ei perusversioon saa:

- renderöinti suoraan tekstuuriin
- jälkikäsitteilyefektit
- mahdollisuus käyttää liitännäisiä
- kameralle näkymättömien 3D-pintojen piirtämättä jättäminen
- dynaamiset varjot. (Unity Technologies 2013.)

3.4.1 Graafiset ominaisuudet

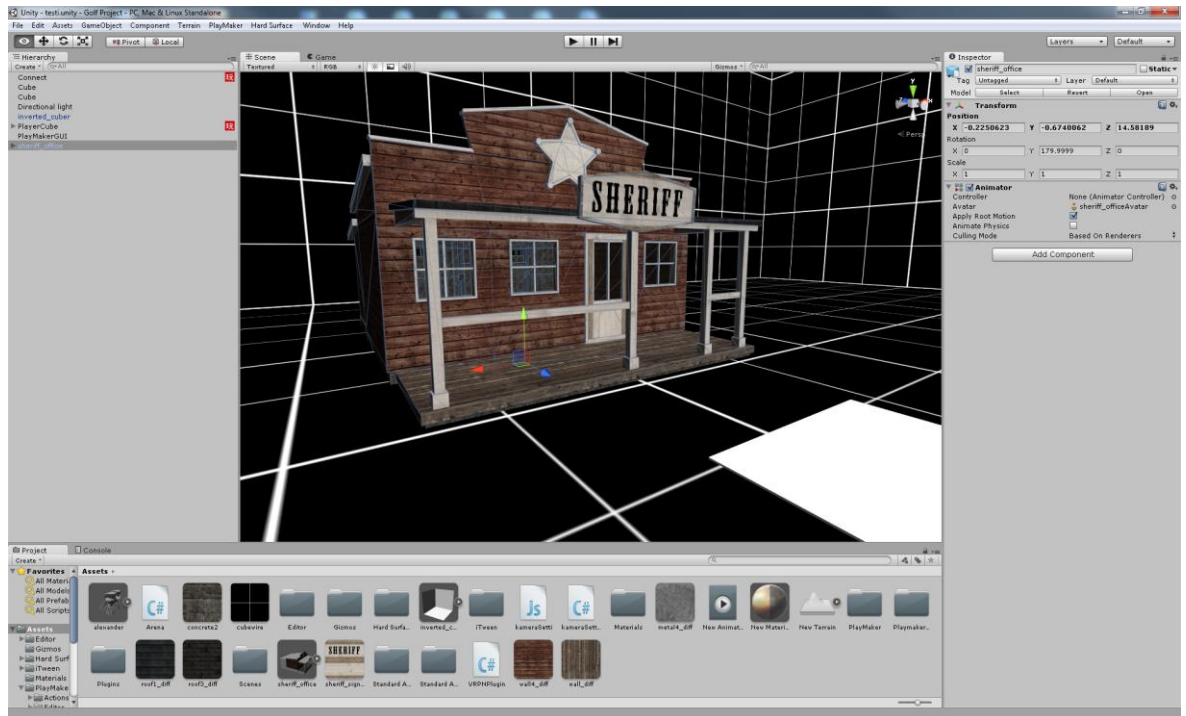
Graafisia ominaisuuksia voi luoda tai muokata rajoitetusti kirjoittamalla skriptejä. Unity tukee DirectX 11-versiota sekä OpenGL:ää. Unityllä voi muokata sen olemassa olevia varjostimia tai luoda uusia varjostimia. (Unity Technologies 2013.)

Unityn mukana tulevia jälkikäsittelyefektejä ovat kameran liikkeensumentuminen, syvyysterävyysalue bokeh-ominaisuudella, kuvan rakeisuus, kuvan kohina, hehku, linssiefektit, värinkorjaus, reunanhavainnointi ja ympäristön okklusio. (Unity Technologies 2013.)

Unity tarjoaa Pro-versiossa täydet dynaamiset valot ja varjot niin PC- kuin mobiili-järjestelmiin. Valoja voi olla lähes rajattomasti käytössä. Kuvan renderöinti tapahtuu monisäikeistystä hyödyntämällä. (Unity Technologies 2013.)

3.4.2 Unityn editori

Käyttäjän tekemät muutokset Unityn editorissa tapahtuvat reaaliajassa ja grafiikka näyttää editorissa samalta kuin se näyttäisi 3D-sovelluksen sisällä. Unityn editorista voi hypätä peliin tai sovellukseen saumattomasti ilman lataustaukoja. (Unity Technologies 2013.)



Kuvio 5. Unityn pääikkuna

Unity sisältää Asset Storen, jonka laajasta valikoimasta voi ostaa tai ladata ilmaiseksi muun muassa scene-tiedostoja, skriptejä, laajennoksia, 3D-malleja, animaatioita, tekstuureita, ääniä ja materiaaleja. (Unity Technologies 2013.)

Unity Pro-versiossa on muun muassa seuraavia työkaluja ja ominaisuuksia:

- Mecanim-animaatiotyökalu
- fysiikkamoottori
- maastoeditori
- partikkelieditori
- Beast, valaistuksen laskija. (Unity Technologies 2013.)

Unityn editori ei sisällä toimintojen luomiseen ja muokkaamiseen tarkoitettua visuaalista editoria, kuten UDK:n Kismet-editori tai Cryengine 3 SDK:n Flowgraph-editori. Tällaisia editoreja on kuitenkin saatavilla Unityn Asset Storesta ja ne asennuvat Unityn oman editorin lisäosiksi. (Unity Technologies 2013.)

Unity sisältää nopean ja tarkan Illuminate Labsin luoman valaistuksen laskijatyökalun Beastin. Beast laskee simuloidun auringon fotonien kimpoamista eri materiaa-

leista. Beast tallentaa valotiedon tekstuureihin. Valaistuksenlaskentaprosessi ei toimi reaaliajassa. (Unity Technologies 2013.)

Unityn Pro-versiossa pystyy hyödyntämään Umbra Softwaren tekemää kameralle näkymättömien 3D-mallien piilottajaa. Piilottaja toimii esilasketulla mallilla. Se tuo näkyviin ja piilottaa ympäristön 3D-malleja kamerasiirtojen mukana. Piilottajan tuoman hyödyn huomaa etenkin suurissa ympäristöissä, jotka sisältävät paljon 3D-malleja. (Unity Technologies 2013.)

3D-mallien tuonti Unityyn tapahtuu parhaiten Autodeskin luoman FBX-liitännäisen avulla. Liitännäinen on saatavilla Autodesk 3ds Maxiin, Autodesk Mayaan ja Autodesk Softimageen. FBX-liitännäinen tallentaa luodun 3D-skenen FBX-tiedostoformaattiin, jonka jälkeen FBX-tiedoston voi pudottaa Unityn projektin Assets-kohtaan. FBX-tiedostoon voi sisällyttää 3D-mallin, materiaalit, tekstuurit ja animaatiotiedot. Unity tukee myös seuraavia 3D-mallitiedostoja: dae, 3DS, dxf ja obj. (Unity Technologies 2013.)

Unity tukee seuraavien kuvatiedostomuotojen suoraa tuomista editoriin: JPEG, PNG, GIF, BMP, TGA ja TIFF. Kuvatiedostot voidaan asettaa esimerkiksi difuusitekstuureiksi, normaalikartoiksi, heijastuskartoiksi, taivaskartaksi tai kuutiokartoiksi. (Unity Technologies 2013.)

Unity-pelimoottoria käyttäessä C#-kielen tai Java-ohjelmointikielen osaaminen on välttämätöntä, jos käyttäjä haluaa luoda toiminnallisuutta ja interaktiivisuutta 3D-sovellukseen tai peliin.

Toiminnallisuutta luodaan skripteillä. Skriptin voi liittää mihin tahansa objektiin. Skriptin kirjoittaminen on nopeampaa kuin ohjelmointi, mutta skriptit ovat rajattuja valmiiksi tehtyihin kutsuihin 3D-pelimoottorista. Skripti käskyttää pinnan alla käytössä olevaa 3D-pelimoottoria, jonka ohjelmakoodi on ulospäin näkymätöntä ja suljettua. Unityn lähdekoodin saa käyttöönsä lisensoimalla Unityn täyden lisenssin, jonka hintaa ei ole yleisesti ilmoitettu. (Unity Technologies 2013.)

3.5 Projektiin sopivan pelimoottorin valinta

Cryengine 3 SDK vakuutti erityisesti edistyksellisillä graafisilla ominaisuuksillaan. Jotta Cryengine 3 SDK:n olisi saanut toimimaan CAVE-järjestelmässä, olisi siihen täytynyt ohjelmoida lähiverkon kautta toimiva synkronointijärjestelmä PC-klusterin synkronoimiseen. Tämän lisäksi olisi täytynyt toteuttaa perspektiivin korjaus kameroihin sekä paikannusdatan käsittelijä.

Kaikkia CAVE-järjestelmän ominaisuuksia ei olisi voinut Flowgraph-editorilla toteuttaa. Kehitystyö olisi vaatinut C++-ohjelmointitaitoa, eikä valmiita CAVE-toteutuksia ollut saatavilla. Näistä syistä Cryengine 3 SDK:ta ei valittu CAVE-järjestelmän pelimoottoriksi.

UDK:n CAVE-yhteensopivuus olisi vaatinut UnrealScriptillä luotua synkronointijärjestelmää PC-klusterin synkronoimiseen, kameroiden perspektiivin korjauksen luomista sekä paikannusdatan käsittelijän luomista. Valmiita sovelluksia tai skriptejä CAVE-yhteensopivuuteen ei ollut saatavilla ja UnrealScript-skriptaustaidon puutteellisuuden johdosta ei UDK:ta valittu CAVE:n pelimoottoriksi.

Unity-pelimoottori ei ole niin graafisesti kehittynyt, kuin UDK tai Cryengine 3 SDK. Kuitenkin valmiiden tai helposti muokattavissa olevien skriptien saatavuudessa Unity-pelimoottori oli ylivoimaisesti paras. Unityn editori oli vertailun kolmesta pelimoottorista paras ja helppokäyttöisin. CAVE-käyttöön soveltuvia helposti muokattavissa olevia C#- ja javaskriptejä oli ilmaiseksi saatavilla. Näistä syistä Unity valittiin pelimoottoriksi CAVE-käyttöön.

4 Unity-pelimoottori CAVE-järjestelmässä

Unity-pelimoottori oli tarkoitus saada tukemaan seuraavia CAVE:n ominaisuuksia:

- stereoskooppinen 3D-kuvan piirto
- PC-klusterin tietokoneiden synkronointi lähiverkossa
- 3D-grafiikan piirto käyttäjän perspektiivistä
- optinen paikannusjärjestelmä.

Unitystä aktivoitiin kolmenkymmenen päivän Pro-kokeiluversio. Tarkoituksena oli luoda demonstraatiosovellus, jota pystyisi ajamaan klusterin tietokoneilla ja näyttämään sovellusta CAVE-tilassa.

Työ aloitettiin Unityssä tyhjästä projektista. Unityn editoriin sisäänrakennetusta Asset Storesta ladattiin ilmaiseksi saatavilla oleva 572.4 Mt:n kokoinen Unity Labs 01: Robot Lab -tiedosto. Tiedosto purettiin Unityn editorissa ja otettiin käyttöön.

RobotLabin scene-tiedosto sisälsi valmiin ympäristön grafiikoineen ja toimintoi-
neen. Scene-tiedosto sisälsi yhden ison tehdashallin näköisen 3D-mallin, joka toi-
mi ympäristönä. Neljätoista erilaista 3D-mallia oli hajautettu tai kopioitu ympäris-
töön. Malleihin lisättiin fysiikkamoottorin törmäysmuotoja.

4.1 Stereoskooppinen 3D-kuva Unityssä

Työn alussa käytettiin zSpacen tekemää Unity-lisäosaa, joka mahdollisti Quad Buffer Stereon käytön. Lisäosan käytössä tuli myöhemmin ongelmia. Lisäosa ei välttämättä aina toiminut Unityn päivitysten kanssa, ja usein joutui odottamaan viikkoja yhteensopivan lisäosan saatavuutta.

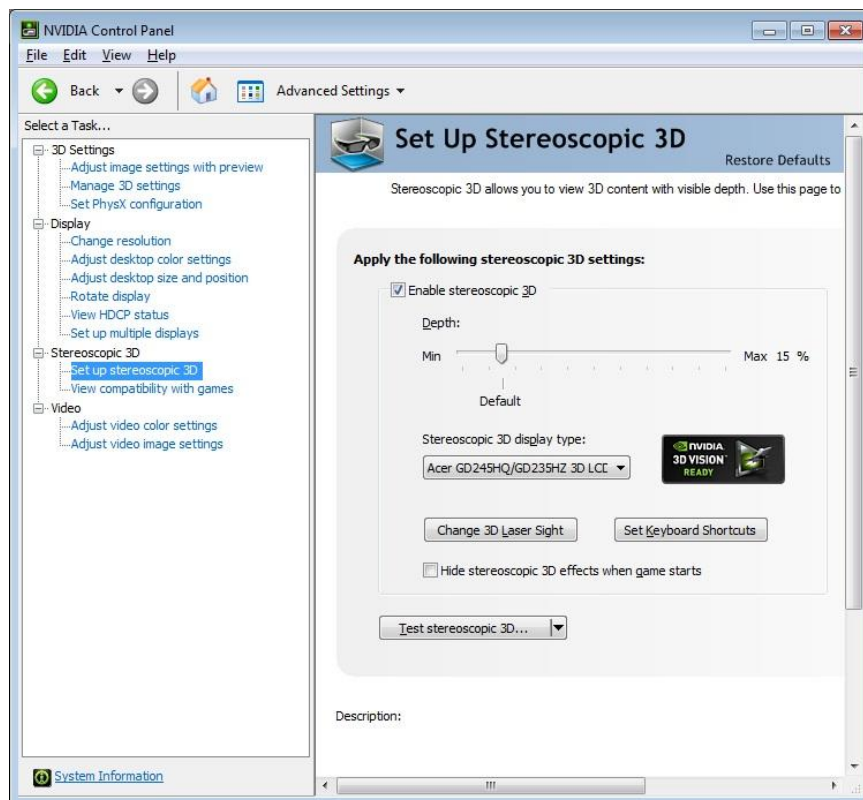
Lisäosa vaati myös OpenGL graafisen rajapinnan käytön, jota Unity ei suoranai-
sesti tue Windows-käyttöjärjestelmissä. Lisäksi lisäosa kaappasi Unityn kameran,
jonka jälkeen perspektiivin korjaus olisi ollut mahdotonta toteuttaa muille kameroil-
le. Näistä syistä zSpacen Unity-lisäosa poistettiin käytöstä.

Vaihtoehtoa zSpacen lisäosalle etsittäessä kävi ilmi, että NVIDIA 3D Vision kykenee kääntämään Unity-pelimoottorista tulevan kuvan stereoskooppiseksi 3D-kuvaksi. Unityssä ei ole erikseen tukea 3D Visionille, mutta 3D Visionin ajurit osaavat muodostaa stereoskooppisen kuvan itsenäisesti kaappaamalla kameran ohjauksen.

CAVE:n PC-klusterin Quadro 6000 -näytönohjaimet sisältävät 3D Vision -tuen. Nämä ammattilaisnäytönohjaimet sallivat 3D Visionin käytön millä tahansa aktiivisereolaseilla ja näyttölaitteilla. Tämä ei olisi mahdollista kuluttajille suunnatuissa NVIDIA:n näyttönohjaimissa.

3D Visionin käyttöön liittyy tiettyjä rajoitteita:

- valmis sovellus täytyy piirtää DirectX-ohjelmointirajapinnalla
- sovellus täytyy ajaa kokoruututilassa
- varjostinefekteissä saattaa näkyä virheitä
- 3D-efektiä ei pääse muokkaamaan suoraan Unitystä. (Unity3D Wiki 2013.)



Kuvio 6. NVIDIA Control Panel

Stereoskooppinen 3D kytketään päälle NVIDIA Control Panelista valikosta ”Set up stereoscopic 3D”. Liukusäätimestä voi säätää 3D-efektin voimakkuutta. Stereoskooppisen 3D:n syvyysasetus oli parhaimman näköinen, kun säädön jätti yhteen prosenttiin. Jos syvyysefektiä säädettiin lisää, CAVE-tilassa projisoitu kuva muuttui epätarkaksi ja silmiä rasittavaksi.

Stereoskooppisen 3D:n päälle kytkemisen jälkeen lähes kaikki Direct3D-ohjelmat ja pelit näyttävät kuvan stereoskooppisena. Tämän ominaisuus edellyttää sovelluksien näyttämistä kokoruututilassa. Lisäksi 3D Visionin ajurin täytyy päästä kaappaamaan sovelluksen kamera-objekti.

4.2 Kamera- ja paikannusjärjestelmä

CAVE-yhteensopivan Unityn kamerajärjestelmä koostuu viidestä kamerasta, jotka on sijoitettu päällekkäin samaan kohtaan 3D-maailmassa. Kamerat ovat orientoitu kohti kuvitteellisia CAVE-tilan projektiopintoja.

Haasteena oli saada kameroiden kuvan piirto toimimaan keskenään siten, että sovellusta CAVE-tilassa käytettäessä eri kameroiden kuvat näkyisivät saumattomasti eri projektiopinnoilla. Ongelma ratkaistiin käyttämällä kustomoitua siirto-projektio-skriptiä, joka löytyy liitteestä kaksi.

Siirtoprojektio-skripti lisättiin jokaiselle viidelle kameraobjektille erikseen. 3D-maailmaan täytyi luoda oikean kokoiset ja oikeilla paikoilla olevat 3D-pinnat kameroiden ympärille, jotka edustivat viittä CAVE:n seinää. Skripti tarvitsi näiden luotujen 3D-pintojen reunusten koordinaatteja kuvan siirtoon ja piirtämiseen.

Skriptissä kuitenkin havaittiin pian ongelmia. Se ei tukenut dynaamisia varjoja, dynaamista valaistusta ja muutamia jälkikäsitteleyefektejä. Unityn foorumeilta löytyi erään käyttäjän tekemä siirtoprojektio-skripti, joka salli edellä mainitut ominaisuudet. Tämä siirtoprojektio-skripti löytyy liitteestä yksi.

Paikannusjärjestelmä. CAVE:n paikannusjärjestelmää käyttämällä oli tarkoitus saada Unityn kamerajärjestelmä liikkumaan reaaliajassa CAVE:n käyttäjän pään liikkeiden mukaisesti. Tätä tekniikkaa käyttämällä kuva piirtyisi käyttäjän perspek-

tiivistä. Käyttäjän perspektiivistä piirretty kuva sallii virtuaaliympäristön tarkastelun käyttäjän omien liikkeiden mukaisesti. Käyttäjä voi esimerkiksi tarkastella ympäristön objekteja eri puolilta liikuttamalla päätään CAVE-tilassa eri suuntiin.

Käyttäjän perspektiivistä piirretyn kuvan ongelmaksi muodostui paikannusjärjestelmän lähiverkossa lähettämän datan käsittely. NaturalPointin foorumeilla oli saatavilla paikannusdatan käsittelyä varten kolme skriptiä, jotka eräs NaturalPointin työntekijä oli kirjoittanut Unity-käyttöä varten.

Skriptien käyttöönotossa todettiin, että ne eivät pystyisi suoraan ilman uudelleenkirjoittamista hyödyntämään CAVE:n paikannusjärjestelmää. Tämän johdosta paikannusjärjestelmän hyödyntämisestä luovuttiin.

4.3 Synkronointi verkon kautta

CAVE:ssa kuvan piirtoon tarvitaan jokaista PC-klusterin tietokonetta, jotta kaikille projektiopinnoille saadaan kuva näkyviin. Kaikkien viiden projektiopinnoin käyttö on välttämätöntä, jotta käyttäjä saisi tunteen kolmiulotteisesta virtuaalilasta.

Jokaisella PC-klusterin tietokoneella ajetaan sama ohjelma, joka tässä tapauksessa on Unityllä luotu demonstraatio-sovellus. Tämä sovellus suunniteltiin monikäyttöiseksi. Sovellus voi käyttäytyä palvelimena tai asiakasohjelmana. Koska klusterin konsolikoneella hallinnoidaan muita klusterin tietokoneita, oli järkevää asettaa demonstraatio-sovellus toimimaan palvelimena tälle tietokoneelle.

PC-klusterin tietokoneiden synkronointi tapahtuu lähiverkon kautta. Unity-pelimoottorista löytyy sisäänrakennettu nettikoodi, joten suurta vaivaa ei tämän ominaisuuden toteuttamiseen vaadittu. Nettikoodia piti kuitenkin pystyä kutsumaan ja käyttämään.

Synkronoinnin toteutus aloitettiin lataamalla Unityyn Unity Storesta saatavilla oleva PlayMaker-lisäosa. PlayMaker-lisäosalla voi luoda monimutkaisia toimintoja visuaalisesti ilman ohjelmointitaitoa. PlayMaker sisälsi muun muassa nettikoodin kutsuja ja toimintoja.

PlayMakerilla luotiin sovelluksen graafinen käyttöliittymä. Käyttöliittymä aukeaa heti näytön vasempaan yläkulmaan kun demonstraatio-sovelluksen käynnistää. Käyttöliittymässä syötetään palvelimena toimivan koneen IP-osoite ja portin numero. CAVE-käytössä käytetään PC-klusterin konsolikoneen IP-osoitetta. Tämän jälkeen valintoina on joko palvelimeen yhdistäminen asiakkaan roolissa tai palvelimen käynnistäminen.



Kuvio 7. Demonstraatio-sovelluksen aloitusruutu

Nappien toiminnallisuus luotiin PlayMaker-lisäosalla. Start Server -napin painallus käynnistää lyhyen toiminnan, joka kutsuu Unityn sisäänrakennetusta nettikoodista palvelimen luontia. Connect As Client -napin painallus kutsuu asiakaskoneen nettikoodia.

Palvelimen täytyy olla käynnissä ensimmäisenä. Muilla neljällä klusterin tietokoneella käynnistetään sama demonstraatio-sovellus ja yhdistetään asiakkaan roolissa konsolikoneella käytettävään palvelinohjelmaan.

Liikettä sisältävät objektit pitää synkronoida muiden asiakaskoneiden kanssa, jotta liike olisi mahdollisimman sulavaa. Objekteissa ei ole oletuksena päällä verkon kautta tapahtuvaa synkronointia, koska suurien 3D-maailmoiden jokaisen objektin synkronointi tukkisi verkkoliikenteen. Synkronoinnin käyttöönotto tapahtuu lisäämällä Unityn editorista Network View -komponentti haluttuun objektiin.

5 Yhteenveto

Pelimoottorien ja virtuaaliohjelmistojen kehitys etenee huimaa vauhtia. Käyttäjien saataville tulee jatkuvasti uusia 3D-ohjelmia ja pelimoottoreita. Seinäjoen ammatti-
korkeakoulun Tekniikan yksikön CAVE-järjestelmässä oli käytössä nykytekniikkaan nähden vanhahtavia 3D-ohjelmia. Tämä opinnäytetyö toi CAVE:n ohjelmistoon huomattavan parannuksen valjastamalla suosituksen ja helppokäyttöisen Unity-pelimoottorin CAVE-yhteensopivaksi.

Unity-pelimoottorin CAVE-toteutuksesta jäi uupumaan tuki optiselle paikannukselle. Paikannus on yksi oleellisista osista CAVE:n käyttöä. Kehitystyö kuitenkin toivottavasti jatkuu tulevien opiskelijoiden toimesta. Valmiudet optisen paikannuksen hyödyntämiseen ovat olemassa. Paikannuksen toteuttamiseen vaaditaan edistynyttä ohjelmointitaitoa ja Unity-pelimoottorin käytön osaamista.

Unity-pelimoottorin muokkaus CAVE-yhteensopivaksi toteutui työn tavoitteiden mukaisesti, paitsi optisen paikannuksen osalta. Projekti eteni sopivalla tahdilla. Sain apua projektin edetessä muun muassa Unityn foorumeilta ja Unityn käyttöohjeista. Projektin teko oli erittäin mielekästä ja haastavaa. Oli myös mukava huomata, kuinka pitkälle projektia pystyi Unityllä tekemään ilman ohjelmoinnin tarvetta.

Unity-pelimoottorin CAVE-käyttöönoton myötä graafinen laatu parani huomattavasti verrattuna muiden CAVE:ssa käytettyjen ohjelmien grafiikkaominaisuuksiin. Unityn editorilla ympäristöjen rakentaminen 3D-malleista on huomattavasti helpompaa ja nopeampaa kuin muilla CAVE:ssa käytössä olevilla ohjelmilla.

LÄHTEET

Virtuaalilaboratorio. 2011. [Verkkosivu]. SeAMK, Tekniikka. [Viitattu 19.3.2013]. Saatavissa: <http://www.seamk.fi/?deptid=1374&preview=1>

SeAMK. 2009. Hankkeita. [Verkkosivu]. SeAMK, Tekniikka. [Viitattu 19.3.2013]. Saatavissa: http://www.seamk.fi/Suomeksi/SeAMK_Info/Yksikot/SeAMK_Tekniikka/Laboratorio_t/Virtuaalitekniikka/Hankkeet.iw3

Christie Digital Systems. 2013. Mirage 4000. [Verkkosivu]. Christie Digital Systems USA. [Viitattu 9.3.2013]. Saatavissa: <http://www.christiedigital.com/en-us/product-support/discontinued-products/Pages/Christie-Mirage4000-Digital-Projector.aspx>

RealD. Ei päiväystä. CrystalEyes 4s. [Verkkosivu]. RealD Inc. [Viitattu 11.3.2013]. Saatavissa: <http://www.reald.com/content/crystal-eyes-4.aspx>

Yamaha Corporation. 2013. Main Features. [Verkkosivu]. Yamaha Corporation. [Viitattu 12.3.2013]. Saatavissa: <http://www.yamahaproaudio.com/global/en/products/mixers/01v96vcm/features.jsp>

Tree C Technology. Ei päiväystä. VR4MAX Extreme. [Verkkosivu]. Tree C Technology B.V. [Viitattu 25.4.2013]. Saatavilla: <http://www.tree-c.nl/vr4max/vr4max-extreme>

Dassault Systèmes. Ei päiväystä. 3DVIA Virtools 5. [Verkkosivu]. Dassault Systèmes. [Viitattu 24.4.2013]. Saatavilla: http://www.3ds.com/fileadmin/PRODUCTS/3DVIA/3DVIAVirtools/Resources/datasheets/ds_virtools5_3DVIA_eng_LR.pdf

Samuel Gateau. 2011. Stereoscopic 3D Demystified: From Theory to Implementation in Starcraft 2. [Verkkosivu]. NVIDIA Corporation. [Viitattu 26.4.2013]. Saatavilla: <http://www.nvidia.com/content/PDF/GDC2011/Stereoscopy.pdf>

Alex Galuzin. 2012. What Level Editor and Game Engine Should You Use - (How to Choose). [Verkkosivu]. [Viitattu 11.4.2013]. Saatavilla:

http://www.worldofleveldesign.com/categories/level_design_tutorials/what-level-editor-game-engine-should-you-use-how-to-choose.php

Nicholas Werner. 2011. All your history: Crytek. [Verkkosivu]. [Viitattu 1.3.2013]. Saatavilla: <http://www.insidegamingdaily.com/2011/11/30/all-your-history-crytek-part-1>

Crytek. 2013. Cryengine 3: Visuals. [Verkkosivu]. Crytek GmbH. [Viitattu 2.3.2013]. Saatavilla: <http://mycryengine.com/index.php?conid=8>

Crytek. 2013. Sandbox. [Verkkosivu]. Crytek GmbH. [Viitattu 3.3.2013]. Saatavilla: <http://mycryengine.com/index.php?conid=53>

Chris Plante. 2012. Better with age: A history of Epic Games. [Verkkosivu]. Polygon. [Viitattu 4.3.2013]. Saatavilla: <http://www.polygon.com/2012/10/1/3438196/better-with-age-a-history-of-epic-games>

Epic Games. 2013. Unreal Engine 3 Game Development Tools. [Verkkosivu]. Epic Games. [Viitattu 15.4.2013]. Saatavilla: <http://www.unrealengine.com/en/features/>

Unity Technologies. 2013. Public Relations. [Verkkosivu]. Unity Technologies. [Viitattu 24.4.2013]. Saatavilla: <http://unity3d.com/company/public-relations/>

Unity Technologies. 2013. What's New. [Verkkosivu]. Unity Technologies. [Viitattu 12.4.2013]. Saatavilla: <http://unity3d.com/unity/whats-new/unity-4.0>

Unity Technologies. 2013. Unity rendering. [Verkkosivu]. Unity Technologies. [Viitattu 15.4.2013]. Saatavilla: <http://unity3d.com/unity/quality/rendering>

Unity3D Wiki. 2013. Stereoscopic 3D. [Verkkosivu]. [Viitattu 17.4.2013]. Saatavilla: http://wiki.etc.cmu.edu/unity3d/index.php/Stereoscopic_3D

Unity Technologies. 2013. Unity scripting. [Verkkosivu]. Unity Technologies. [Viitattu 17.4.2013]. <http://unity3d.com/unity/workflow/scripting>

LIITTEET

LIITE 1 Kameran skripti

```

using UnityEngine;
using System.Collections;

[ExecuteInEditMode]
public class OffProjection : MonoBehaviour {
    public float left = -0.2F;
    public float right = 0.2F;
    public float top = 0.2F;
    public float bottom = -0.2F;

    void LateUpdate() {
        Camera cam = camera;
        float width = (right - left);
        float height = (top - bottom);
        cam.aspect = width/height;
        float fov = (Mathf.Atan(top/cam.nearClipPlane) - Mathf.Atan(bottom/cam.nearClipPlane)) * 57.2957795f;
        cam.fieldOfView = fov;

        Matrix4x4 m = PerspectiveOffCenter(left, right, bottom, top, cam.nearClipPlane, cam.farClipPlane);
        cam.projectionMatrix = m;
    }

    static Matrix4x4 PerspectiveOffCenter(float left, float right, float bottom, float top, float near, float far) {
        float x = 2.0F * near / (right - left);
        float y = 2.0F * near / (top - bottom);
        float a = (right + left) / (right - left);
        float b = (top + bottom) / (top - bottom);
        float c = -(far + near) / (far - near);
        float d = -(2.0F * far * near) / (far - near);
        float e = -1.0F;
        Matrix4x4 m = new Matrix4x4();
        m[0, 0] = x;
        m[0, 1] = 0;
        m[0, 2] = a;
        m[0, 3] = 0;
        m[1, 0] = 0;
        m[1, 1] = y;
        m[1, 2] = b;
        m[1, 3] = 0;
        m[2, 0] = 0;
        m[2, 1] = 0;
        m[2, 2] = c;
        m[2, 3] = d;
        m[3, 0] = 0;
        m[3, 1] = 0;
        m[3, 2] = e;
        m[3, 3] = 0;
        return m;
    }
}

```

LIITE 2 Kameran toinen skripti

```
// This script should be attached to a Camera object
// in Unity. Once a Plane object is specified as the
// "projectionScreen", the script computes a suitable
// view and projection matrix for the camera.
// The code is based on Robert Kooima's publication
// "Generalized Perspective Projection," 2009,
// http://csc.lsu.edu/~kooima/pdfs/gen-perspective.pdf
#pragma strict
```

```
public var projectionScreen : GameObject;
public var estimateViewFrustum : boolean = true;
```

```
function LateUpdate() {
    if (null != projectionScreen)
    {
        var pa : Vector3 =
            projectionScreen.transform.TransformPoint(
                Vector3(-5.0, 0.0, -5.0));
            // lower left corner in world coordinates
        var pb : Vector3 =
            projectionScreen.transform.TransformPoint(
                Vector3(5.0, 0.0, -5.0));
            // lower right corner
        var pc : Vector3 =
            projectionScreen.transform.TransformPoint(
                Vector3(-5.0, 0.0, 5.0));
            // upper left corner
        var pe : Vector3 = transform.position;
            // eye position
        var n : float = camera.nearClipPlane;
            // distance of near clipping plane
        var f : float = camera.farClipPlane;
            // distance of far clipping plane

        var va : Vector3; // from pe to pa
        var vb : Vector3; // from pe to pb
        var vc : Vector3; // from pe to pc
        var vr : Vector3; // right axis of screen
        var vu : Vector3; // up axis of screen
        var vn : Vector3; // normal vector of screen

        var l : float; // distance to left screen edge
        var r : float; // distance to right screen edge
        var b : float; // distance to bottom screen edge
        var t : float; // distance to top screen edge
        var d : float; // distance from eye to screen

        vr = pb - pa;
        vu = pc - pa;
        vr.Normalize();
        vu.Normalize();
        vn = -Vector3.Cross(vr, vu);
            // we need the minus sign because Unity
            // uses a left-handed coordinate system
        vn.Normalize();

        va = pa - pe;
        vb = pb - pe;
        vc = pc - pe;
```

```

d = -Vector3.Dot(va, vn);
l = Vector3.Dot(vr, va) * n / d;
r = Vector3.Dot(vr, vb) * n / d;
b = Vector3.Dot(vu, va) * n / d;
t = Vector3.Dot(vu, vc) * n / d;

var p : Matrix4x4; // projection matrix
p[0,0] = 2.0*n/(r-l);
p[0,1] = 0.0;
p[0,2] = (r+l)/(r-l);
p[0,3] = 0.0;

p[1,0] = 0.0;
p[1,1] = 2.0*n/(t-b);
p[1,2] = (t+b)/(t-b);
p[1,3] = 0.0;

p[2,0] = 0.0;
p[2,1] = 0.0;
p[2,2] = (f+n)/(n-f);
p[2,3] = 2.0*f*n/(n-f);

p[3,0] = 0.0;
p[3,1] = 0.0;
p[3,2] = -1.0;
p[3,3] = 0.0;

var rm : Matrix4x4; // rotation matrix;
rm[0,0] = vr.x;
rm[0,1] = vr.y;
rm[0,2] = vr.z;
rm[0,3] = 0.0;

rm[1,0] = vu.x;
rm[1,1] = vu.y;
rm[1,2] = vu.z;
rm[1,3] = 0.0;

rm[2,0] = vn.x;
rm[2,1] = vn.y;
rm[2,2] = vn.z;
rm[2,3] = 0.0;

rm[3,0] = 0.0;
rm[3,1] = 0.0;
rm[3,2] = 0.0;
rm[3,3] = 1.0;

var tm : Matrix4x4; // translation matrix;
tm[0,0] = 1.0;
tm[0,1] = 0.0;
tm[0,2] = 0.0;
tm[0,3] = -pe.x;

tm[1,0] = 0.0;
tm[1,1] = 1.0;
tm[1,2] = 0.0;
tm[1,3] = -pe.y;

tm[2,0] = 0.0;
tm[2,1] = 0.0;
tm[2,2] = 1.0;
tm[2,3] = -pe.z;

```

```

tm[3,0] = 0.0;
tm[3,1] = 0.0;
tm[3,2] = 0.0;
tm[3,3] = 1.0;

// set matrices
camera.projectionMatrix = p * rm * tm;
camera.worldToCameraMatrix = Matrix4x4.identity;
// we put everything into the projection matrix:
// because our "viewing matrix" might look at a
// point that is off the screen.

if (estimateViewFrustum)
{
    // rotate camera to screen for culling to work
    var q : Quaternion;
    q.SetLookRotation((0.5 * (pb + pc) - pe), vu);
    // look at center of screen
    camera.transform.rotation = q;

    // set fieldOfView to a conservative estimate
    // to make frustum tall enough
    if (camera.aspect >= 1.0)
    {
        camera.fieldOfView = Mathf.Rad2Deg *
            Mathf.Atan((vu.magnitude + vr.magnitude)
                / va.magnitude);
    }
    else
    {
        // take the camera aspect into account to
        // make the frustum wide enough
        camera.fieldOfView =
            Mathf.Rad2Deg / camera.aspect *
            Mathf.Atan((vu.magnitude + vr.magnitude)
                / va.magnitude);
    }
}
}
}
}

```