

Sami Salo

**KOLMIULOTTEISEN KUVAN LUOMINEN KINECT-
LIIKETUNNISTIMELLA UNITY 3D -OHJELMISTOSSA**

**Opinnäytetyö
CENTRIA AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Maaliskuu 2013**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika Maaliskuu 2013	Tekijä/tekijät Sami Salo
Koulutusohjelma Tietotekniikka		
Työn nimi KOLMIULOTTEISEN KUVAN LUOMINEN KINECT-LIIKETUNNISTIMELLA UNITY 3D -OHJELMISTOSSA		
Työn ohjaaja Hannu Puomio		Sivumäärä 31
Työelämäohjaaja		
<p>Tämän opinnäytetyön tavoitteena oli luoda vaikutelma kolmiulotteisesta tilasta käyttäen Kinect-liiketunnistinta Unity 3D -ohjelmistossa. Tarkoituksena oli saada tietoa Kinect-sensorilta tietokoneelle ja liikuttaa Unity kameraa xyz-koordinaatistossa käyttäjän pään sijainnin mukaan.</p> <p>Teoriaosuus käsittelee liiketunnistimien historiaa, Kinectin sovelluksia, laitteistoa, toimintaa, sekä ajureita. Käytännön osuudessa käydään läpi Kinectin asennus, käyttö Unity 3D ohjelmistossa, sekä skenen luominen. Pääpaino opinnäytetyössä oli 3D-vaikutelman luominen Unityssä.</p> <p>Työ rajattiin Kinect for Xbox 360 liiketunnistimen ja työhön parhaiten soveltuvien ajureiden käyttöön, sekä 3D-vaikutelman luomiseen Unityssä.</p>		
Asiasanat Kinect, kolmiulotteinen, liiketunnistus, Unity 3D		

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES	Date March 2013	Author Sami Salo
Degree programme Information Technology		
Name of thesis CREATING A THREE-DIMENSIONAL VIEW WITH A KINECT MOTION SENSOR IN UNITY 3D SOFTWARE		
Instructor Hannu Puomio		Pages 31
Supervisor		
<p>The objective of this thesis was to create an impression of a three-dimensional space by using the Kinect motion sensor in Unity 3D software. The purpose was to get information from the Kinect sensor to a computer and to move Unity camera in the xyz -coordinate system according to the position of the user's head.</p> <p>The theory part focused on the history of motion sensors, Kinect applications, hardware, functions and drivers. In the practical part the focus was on the installation of Kinect, its usage in Unity 3D software and creating a scene. The main focus of the thesis was to create a 3D impression in Unity.</p> <p>The work was limited to the usage of Kinect for Xbox 360 motion sensor and to the most suitable drivers as well as creating a 3D impression in Unity.</p>		
Key words Kinect, motion detection, three-dimensional, Unity 3D		

**TIIVISTELMÄ
ABSTRACT
SISÄLLYS**

1 JOHDANTO	1
2 MICROSOFT KINECT -LIIKETUNNISTIN	2
2.1 Liiketunnistimien historiaa	2
2.2 Kinect sovellukset	6
2.3 Laitteisto	8
2.4 Toiminta	11
2.5 Ajurit	14
3 KINECT UNITY 3D -OHJELMISTOSSA	17
3.1 Unity 3D	17
3.2 Ajurien valinta	18
3.3 Asennus ja testaus	19
3.4 Zigfu Development Kit	22
3.5 Skenen luominen Unityssä	23
3.6 Tarvittavat skriptit	24
4 TULOKSET JA POHDINTA	28
LÄHTEET	30
LIITTEET	
KUVIOT	
KUVIO 1. NES Power Glove	3
KUVIO 2. Nintendo Wii-Mote	3
KUVIO 3. PlayStation Move ja Eye	4
KUVIO 4. Microsoft Kinect	5
KUVIO 5. Kinectin kamerat ja mikrofonit	8
KUVIO 6. Kinectin piirilevyt	9
KUVIO 7. Infrapunakameran näkymä	11
KUVIO 8. Kinectin luurankomalli	13
KUVIO 9. OpenNI-ohjelmiston toiminta	15
KUVIO 10. Unity 3D -käyttöliittymä	17
KUVIO 11. Laittehallinta näkymä	20
KUVIO 12. Kinectin syvyys- ja videokuva	20
KUVIO 13. Käyttäjän tunnistus ja luuranko	21
KUVIO 14. Käyttäjän käden seuranta	21
KUVIO 15. Objektit skenessä	24
KUVIO 16. Valmiin skenen ajo Unityssä	26
KUVIO 17. Muokatun skenen ajo Unityssä	27

1 JOHDANTO

Työn tavoitteena on luoda näytölle vaikutelma kolmiulotteisesta kuvasta Microsoft Kinect -liiketunnistimen avulla. Tarkoituksena on seurata käyttäjän pään sijaintia Kinectillä ja vastaavasti liikuttaa kuvaa näytöllä xyz-koordinaatistossa. Käyttäjän liikkua esimerkiksi vasemmalle, näytöllä tarkasteltavaa kohdetta katsotaan vasemmalta puolelta. Vastaavasti liikuttaessa oikealle, ylös tai alas, tarkastellaan kohdetta kyseisestä suunnasta. Koska kuvakulma vaihtuu seurattavan käyttäjän liikkua, vaikutelma 3D-tilasta näkyy vain kyseiselle käyttäjälle. Kuvan liikuttelu ja 3D-objektien tarkastelu tapahtuu Unity 3D -ohjelmistossa.

Kiinnostuin Kinectin käyttämisestä tietokoneella nähtyäni monia innovatiivisia käyttötarkoituksia kinecthacks-sivustolla. Käyn työssä aluksi läpi liiketunnistimien historiaa, sekä Kinectille tehtyjä käyttäjäsovelluksia. Tämän jälkeen perehdyn Kinectin sisällä olevaan laitteistoon ja toimintaan, sekä tarvittaviin ajureihin. Kolmannessa osiossa käyn vaiheittain läpi Kinectin asennuksen, Unityyn liittämisen ja skenen, eli kohtauksen luomisen. Käyttöympäristöksi valitsin Unity 3D -ohjelmiston, koska sen avulla voi helposti luoda yksinkertaiset 3D-objektit, sekä liittää näihin toimintoja varten tarvittavat koodit. Lisäksi minulla on Unityn käytöstä aiempaa kokemusta.

Työssä käytin pelkästään Kinect for Xbox 360 -mallia, joten en puutu Kinect for Windows -mallin toimintaan. Viitatessa Windows-malliin, mainitsen asiasta erikseen. Koska Kinect on tarkoitettu peli- ja viihdekäyttöön, rajasin historiaosuuden vain näihin käyttötarkoituksiin kuuluviin liiketunnistimiin.

2 MICROSOFT KINECT -LIIKETUNNISTIN

2.1 Liiketunnistimien historiaa

Pelialalla liiketunnistimet ja liikeohjaimet ovat nousseet suosituksi vaihtoehdoksi perinteisten ohjaimien rinnalle. Liiketunnistimet mahdollistavat pelien pelaamisen ja sovellusten käytön kehon liikkeiden avulla. Liikkeen havaitsemisessa käytetään laitteiden sisällä tiettyjä antureita. Nykyiset liikeohjaimet ja -tunnistimet käyttävät videokameroita ja infrapunasensoreita käyttäjän tunnistuksessa ja etäisyyden mittauksessa, kiihtyvyyssantureita (accelerometer) kiihtyvyyden mittauksessa, gyrokoopiantureita (gyroscope sensor) kiertoliikkeen tunnistamiseen ja magneto metrejä (magnetometer) suunnan määrittelyssä (VGCL 2013).

Normaaleista konsoliohjaimista eroten, liikeohjaimet vaativat mahdollisten napinpainallusten lisäksi ohjaimen tai kehon liikuttamista haluttujen toimintojen suorittamiseen. Ennen Nintendon Wii-Motea, ohjaimissa kuten vuonna 1998 esitelty Microsoft SideWinder Freestyle Pro, käytettiin ”two-axis” kiihtyvyyssanturia, jonka liikeseuranta rajoittui kallistusliikkeisiin (Champy 2007). Pääpaino vastaavissa ohjaimissa oli liikkeentunnistuksen sijaan silti normaali ohjaustyylissä. Nintendon Wii-Mote oli ensimmäinen suosittu liikeohjain, jossa käytettiin ”three-axis” kiihtyvyyssanturia ohjaimen 3D-liiketunnistukseen.

NES Power Glove

Ensimmäisiä peleissä käytettyjä liiketunnistuslaitteita myytiin jo 1980-luvun lopulla. Power Glove julkaistiin Nintendolle 1989 ja se käytti liikkeentunnistukseen ultraääntä. Käsine lähettää ultraäänisignaaleja, joita television päälle ja viereen asetettavat mikrofonit vastaanottivat. Lähetettävä signaali muuttui riippuen siitä, mitkä sormet olivat koukistettuina. Peleille pystyi siis antamaan eri komentoja käsinettä liikuttamalla ja sormia koukistamalla. Erityyisiä pelejä varten käsineen ohjelmointiin oli 14 eri näppäinyhdistelmää, joilla muutettiin ohjaustapaa. Laite oli vaikea-

käyttöinen ja se ei yleensä toiminut kunnolla, joten sitä pidetään epäonnistuneena. (Retro Gaming Nation 2012.)



KUVIO 1. NES Power Glove (Lukiegames 2013).

Nintendo Wii-Mote

Liikeohjainten suosio lähti nousuun Nintendon julkaistua vuonna 2006 Wii-konso-
lin, joka käyttää langattomia liikkeen tunnistavia Wii-Mote -ohjaimia. Ulkonäöltään
kaukosäädintä muistuttavat ohjaimet sisältävät kiihtyvyyssanturin lisäksi kärjessä
sijaitsevan optisen anturin, joka kommunikoi television ylä- tai alapuolelle sijoitet-
tavan Wii Sensor Bar:n kanssa. Anturin avulla määritellään suunta, johon ohjain
osoittaa (VGCL 2013). Liikkeentunnistuksen lisäksi ohjainta voidaan siis käyttää
tietokoneen hiiren tavoin osoittamalla näytölle haluttuun kohtaan ja painamalla
nappia.



KUVIO 2. Nintendo Wii-Mote (Gdgt 2013).

PlayStation Eye ja Move

Sony julkaisi 2007 PlayStation 3:lle lisälaitteen nimeltä PlayStation Eye, jolla on mahdollista kaapata 60 kuvaa sekunnissa 640x480 resoluutiolla ja 120 kuvaa sekunnissa 320x240 resoluutiolla. Videokaappauksen lisäksi ominaisuuksiin kuuluu liikkeen ja värin seuranta, reunojen ja eleiden tunnistaminen, sekä käyttäjän pään seuranta ja kasvojen tunnistus. Sisäänrakennettu mikrofoni voi paikallistaa äänen lähteen ja tunnistaa puhetta sekä vaimentaa taustamelun ja kaiun vaikutusta. (VGCL 2013.)

Syyskuussa 2010 Sony julkaisi PlayStation Move:n, joka toiminnaltaan muistuttaa Nintendon Wii-Motea. Move sisältää samat ominaisuudet ja anturit kuin PlayStation 3:n vakio-ohjain ja lisäksi myös magnetometrin. Ohjaimen kärjessä oleva pallo voi hohtaa RGB-LED:ejä käyttäen monella eri värillä. Move suunniteltiin toimimaan yhdessä PlayStation Eye:n kanssa siten, että Eye pystyy määrittelemään ohjaimen sijainnin ja etäisyyden sen kärjessä olevan pallon muodon ja värin perusteella. Eri värejä käyttämällä voidaan seurata neljää eri ohjainta yhtä aikaa. Ohjaimen magnetometrin, gyroskooppi- ja kiihtyvyyssanturin ja Eye:n seuranta-ominaisuuksien ansiosta ohjaimen liikkeestä saadaan tarkkaa tietoa konsolille. (VGCL 2013.)



KUVIO 3. PlayStation Move ja Eye (Ebay 2013).

Microsoft Kinect

Microsoft vei kehitystä eteenpäin julkaisemalla marraskuussa 2010 Kinect-liike-tunnistimen Xbox 360:lle. Kilpailevista ohjainmalleista poiketen, Kinectin käyttö ei vaadi erillistä kädessä pidettävää liikeohjainta. Näytön ylä- tai alapuolelle sijoitettavassa Kinectissä on videokamera, joka tuottaa 30 kuvaa sekunnissa 640x480 resoluutiolla. Mikrofonirivi laitteen sisällä mahdollistaa äänen lähteen paikallistamisen, puheen tunnistuksen ja taustamelun ja kaiun vaimennuksen. Samat toiminnot löytyvät myös PS Eye:stä, mutta Kinectissä on lisäksi infrapunakamera ja infrapunälähetin, joiden avulla se näkee edessä olevan alueen kolmiulotteisena. Kinect on ohjelmoitu tunnistamaan käyttäjien ruumiinosat ja kasvot, sekä seuramaan koko kehon liikettä. Samanaikaisesti voidaan tunnistaa kuusi käyttäjää. (VGCL 2013.)



KUVIO 4. Microsoft Kinect (Gdgt 2013).

2.2 Kinect sovellukset

Kinect on noussut tunnetuksi laitteeksi erityisesti robotiikan-, 3D-mallinnuksen- ja animaation harrastajien parissa. Koska laite on edullinen ja sitä voidaan käyttää tietokoneella, voivat harrastelijat hyödyntää Kinectin tarjoamia ominaisuuksia omissa projekteissa, sekä sovelluksissa.

Kinect Effect

Microsoft julkaisi Kinectin marraskuussa 2010 ja pian tämän jälkeen Adafruit Industries tarjosi 1000 dollarin palkkiota ensimmäiselle henkilölle, joka saisi rakennettua Kinectille avoimen lähdekoodin ajurit. Kuudessa päivässä eräs hakkeri voitti 3000 dollariin nousseen palkkion, rakennettuaan toimivat ajurit. (Lowensohn 2011.)

Kinectin laaja saatavuus ja matala hinta mahdollistivat ohjelmistokehityksen peliteollisuuden ulkopuolella. Monet tietotekniikan, elektroniikan ja robotiikan ammattilaiset alkoivat kehittää laitteelle uusia käyttötarkoituksia. Microsoft kutsuu tätä ilmiötä nimellä Kinect Effect. Kuukauden kuluessa KinectHacks.com sivustolla oli listattuna jo noin 90 eri projektia, jonka jälkeen määrä on kasvanut tasaiseen tahtiin. (Zhang 2012.)

KinectHacks

KinectHacks-sivusto tarjoaa uutisia ja tietoa käyttäjien luomista Kinect-sovelluksista. Muutamille ohjelmistoille löytyy myös asennusohjeita, mutta näissä olevat tiedot ovat osittain vanhentuneita. Sivustolle on kertynyt yhteensä yli 500 esittelyä projekteista, sekä sovelluksista. Suurin osa sovelluksista on vain esittelyssä, mutta on myös muutamia ilmaisohjelmistoja. Tästä esimerkkinä ohjelmisto nimeltä Flexible Action and Articulated Skeleton Toolkit (FAAST), joka mahdollistaa tietokonepelien pelaamisen Kinectillä. Ohjelmiston avulla käyttäjä voi määrittellä näppäimistön tai hiiren napinpainalluksen tapahtumaan tietyn asennon tai kehon

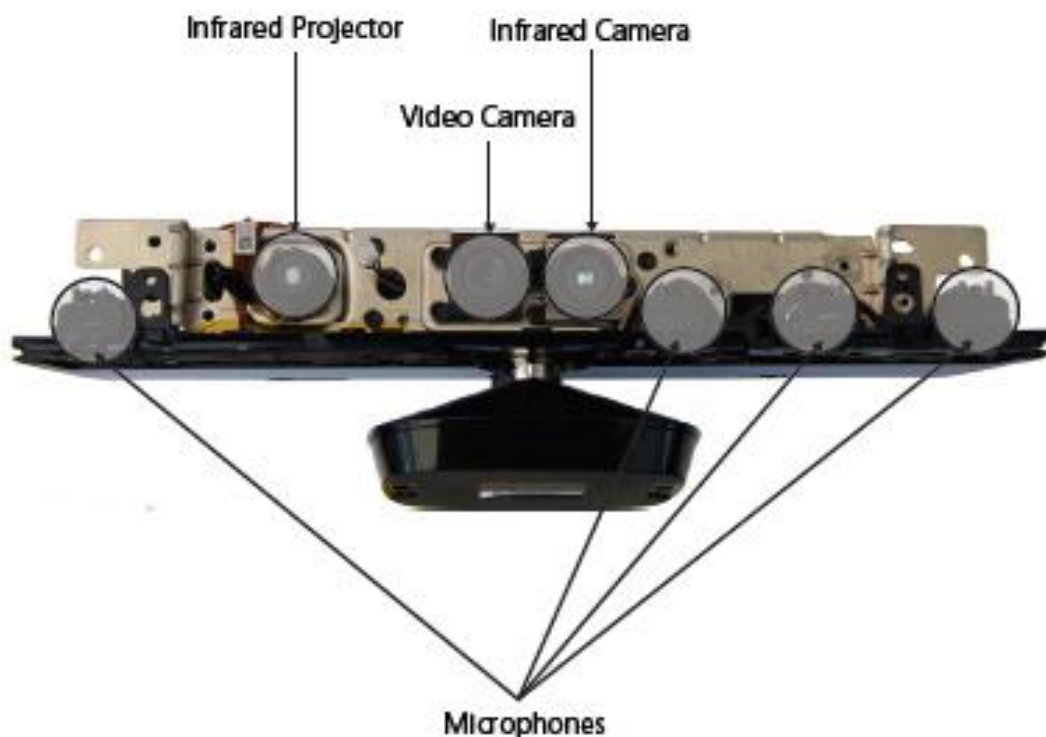
liikkeen seurauksena, jolloin esim. kättä heilauttamalla eteenpäin, pelihahmo iskee miekalla (KinectHacks 2013).

Toinen mainitsemisen arvoinen ohjelmisto on Brekel Kinect, jonka avulla voidaan tuottaa kuvattavasta kohteesta 3D-mallinnus tai animaatio. Ohjelmiston ilmaisella versiolla voidaan tehdä 3D-mallinnus ja kaapata liikkeet BVH-tiedostoon tai Autodesk Motionbuilder -ohjelmistoon. KinectHacks-sivustolta löytyy käyttöohjeistus nimellä MoCap Tutorial, jossa neuvotaan vaiheittain animaation vieminen Motionbuilderiin.

2.3 Laitteisto

Komponentit

Kinect-liiketunnistin käyttää sensoreina videokameraa, infrapunakameraa, infrapunälähetintä sekä neljää mikrofonia. Kameroita kutsutaan myös complementary metal-oxide-semiconductor (CMOS) -sensoreiksi. Sensoreilta saatavaa tietoa laitteen sisällä käsittelee PrimeSense:n valmistama PS1080 -proessori. Infrapunälähttimen ja videokameran välissä on Kinectin tilaa kuvaava status-ledi. Alustassa sijaitsevan hammasvaihhteiston ja sähkömoottorin avulla säädetään Kinectin sensorien kulma pystysuunnassa. Laitteen alaosassa sijaitsevaa neljää mikrofonia käytetään äänikomentojen poiminnassa.



KUVIO 5. Kinectin kamerat ja mikrofonit (Miles 2012, 4).

Laitteen sisällä on kaksi piirilevykerrosta, joista suurimmassa on:

- WM8737G, analogi-digitaalimuuntimet mikrofoneille
- NEC uPD720114, USB 2.0 kontrolleri

- Marvell AP102 System on a Chip (SoC) kamerakontrolleri
- Hynix H5PS5162FF 512 megabit (64 Mt) DDR2 SDRAM muisti. (Ifixit 2010.)

Toisen piirilevyn sisältö, johon kamerat ja infrapunalähetin on kiinnitettyinä:

- AD8694 analogipiiri
- TI ADS78301 analogi-digitaalimuunnin
- Allegro Microsystems A3906, alustan sähkömoottorin ajuri
- ST Microelectronics M29W800DB, 8 Mt flash-muisti
- PrimeSense PS1080 SoC prosessori. (Ifixit 2010.)

Kolmas ja pienin piirilevy sisältää TI TAS1020B USB audio-ohjaimen, sekä Kionix KXSD9 kiihtyvyyssanturin kallistuksen havaitsemiseen. Osa komponenteista tuottaa paljon lämpöä, joten jäähdytystä varten pätyyn on laitettu tuuletin puhaltamaan kuuma ilma ulos. (Ifixit 2010.) Kinect käyttää erillistä AC muuntajaa, jonka arvot ovat 12V, 1.08A.



KUVIO 6. Kinectin piirilevyt (Miles 2012, 5).

Liittäminen tietokoneeseen

Liitettäessä Kinectiä tietokoneen USB -porttiin on näiden välissä käytettävä erillistä virta/USB -jatkojohtoa, koska USB-portin antama virta ei riitä Kinectille ja johdossa käytetään Xbox 360:n erikoisliitintä. Jatkoakaapeli tulee paketin mukana, kun Kinect ostetaan erillään konsolista. Xbox 360 -konsolin mukana tulleelle Kinectille on

jatkokaapeli tilattava netin kautta, tai tehtävä itse yhdistämällä Kinectin johto, USB -kaapeli ja sopiva 12V AC muuntaja.

Tarkkoja laitteistovaatimuksia Kinectillä ei ole, mutta seuraavia Kinect for Windows SDK Beta 2 –ohjelmiston vaatimuksia voidaan pitää suuntaa antavina:

- Windows 7 tai Windows 8
- 32 bit (x86) tai 64 bit (x64) käyttöjärjestelmä
- Dual-core 2.66GHz tai parempi prosessori
- USB 2.0 portti
- 2GB RAM muistia (Microsoft 2012).

2.4 Toiminta

Kinect näkee esineet ja käyttäjät kolmiulotteisena rakentamalla etäisyystietoja sisältävän syvyyskartan (depth map) sen edessä olevasta alueesta infrapunalähttimen ja -kameran avulla. Kamera sisältää näkyvän valon suodattimen, joten se näkee vain infrapunapisteet (KUVIO 7.) (Miles 2012, 7).



KUVIO 7. Infrapunakameran näkymä (Miles 2012, 6).

Syvyyskuva

Infrapunalähtetin suuntaa eteensä tuhansista pienistä infrapunapisteistä koostuvan pseudo-random (suom. näennäissatunnainen) kuvion. Kyseisen kuvion pisteet eivät siis ole satunnaisesti sijoiteltuja, vaan tiedot on mekaanisesti tuotettu ja helppo toistaa. Koska ohjelmisto tietää, miten kuvio on muodostettu, se voi vertailla infrapunakameralta saatavaa kuvaa ja alkuperäistä kuviota, sekä näissä olevista eroista laskea jokaisen pisteen etäisyyden sensorista. Kinectin laitteisto suorittaa

laskennan tuhansille infrapunapisteille monta kertaa sekunnissa. Saaduista tiedoista luodaan syvyyskartta, joka lähetetään USB -kaapelia pitkin analysoitavaksi tietokoneelle tai Xboxille. (Miles 2012, 7-8.)

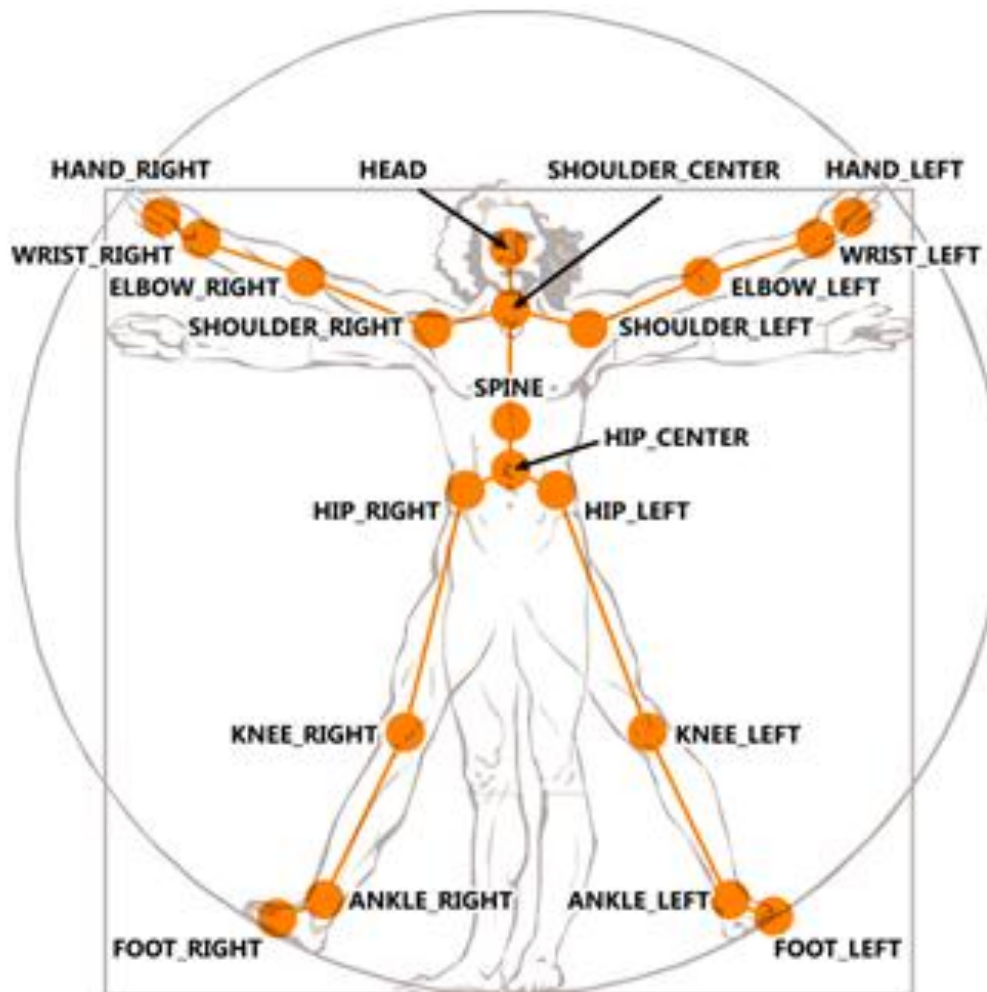
Mikrofonit

Neljän mikrofonin avulla voidaan määritellä mistä suunnasta huonetta käyttäjän ääni tulee. Mikrofonit sijaitsevat eri puolilla Kinectin alaosaa, joten ääni saapuu eri ajanhetkinä jokaiselle mikrofonille. Ohjelmisto pystyy laskemaan äänen lähteen sijainnin saapumisaikojen perusteella. Äänen sijaintitietojen perusteella mikrofonit ohjataan kuuntelemaan käyttäjän ääntä ja muualta tuleva taustamelu voidaan jättää huomioimatta. (Miles 2012, 8-9). Tällä tavalla puheen sisältö saadaan helpommin ymmärrettäväksi ja Kinect voi suorittaa käyttäjän äänikomentoa vastaavan toiminnon taustamelusta huolimatta.

Liiketunnistus

Käyttäjän tunnistaminen ja seuraaminen tapahtuu vasta tietokoneella/Xboxilla. Isäntäkoneen ohjelmisto analysoi Kinectiltä saapuvaa syvyyskarttaa ja tunnistaa ihmisen muotoja muistuttavia osia. Ohjelmisto käyttää tunnistamiseen ja seuraamiseen apuna suuntia, johon ruumiinosat osoittavat, sekä niiden tapoja liikkua. (Miles 2012, 9).

Käyttäjän liikkeiden seuraamista varten ohjelmistoilla luodaan käyttäjästä luurankomalli, joka seuraa käyttäjän liikettä. Luuranko muodostetaan yhdistämällä 20 tiettyä kohtaa käyttäjän kehosta. (KUVIO 8.) Kaikilla luurangon pisteillä on 3D-koordinaatit metreinä, joita voidaan hyödyntää ohjelmoitaessa sovelluksia (Microsoft 2012).



KUVIO 8. Kinectin luurankomalli (Microsoft 2012).

Rajoitukset

Kinectin kallistuskulma on välillä -27 ja $+27$ astetta pystysuunnassa, jossa 0 -aste vastaa vaakatasoa. Field of View (FOV), eli sensorien näkökenttä on vaakatasossa 57 astetta ja pystysuunnassa 43 astetta. Infrapunakameralla käyttöetäisyys on välillä $1.2\text{m} - 3.5\text{m}$. Kinect pystyy tunnistamaan samanaikaisesti yhteensä kuusi käyttäjää, mutta vain kahden käyttäjän liikkeitä voidaan seurata luurankomallilla yhtä aikaa. (Plunkett 2010).

2.5 Ajurit

Kinectiä varten on asennettava sopivat ajurit, jotta USB-porttiin liitettäessä tietokone tunnistaa laitteen. Liiketunnistinta käytettäessä luodaan sensorilta saadusta tiedosta syvyyskartta, jonka jälkeen käytetään väliohjelmistoa (middleware) käsittelemään syvyyskartta ja tunnistamaan käyttäjän ruumiinosat, sekä seuraamaan liikettä (OpenNI 2011). Käytettävät sovellukset voivat tämän jälkeen hyödyntää väliohjelmiston tuottamia tietoja.

Kinect for Windows SDK

Kinect for Windows Software Development Kit (SDK) on Microsoftin kehittämä SDK Kinectin Windows-mallille, joka sisältää myös Kinectin ajurit. Versio 1.6 on tällä hetkellä uusin ja tarjoaa Kinect for Windows-mallin tukemia ominaisuuksia, kuten "Near mode", "skeletal tracking control" ja Application Programming Interface (API) parannuksia. SDK tukee C++, C# ja Visual Basic ohjelmointikieliä, Microsoft Visual Studio 2010 ja 2012 -ympäristössä. (Microsoft 2012).

Xbox 360 -mallin Kinectin saa toimimaan edellä mainitulla SDK:lla, mutta kaikkia toimintoja ei voi käyttää. Viimeisin Kinect for Xbox 360 -laitteen käyttöön tarkoitettu SDK on Kinect for Windows SDK Beta 2 -versio, jolla voidaan kehittää ohjelmistoja ei-kaupalliseen käyttöön. Asennukseen sisältyy Kinectin ajurit, API:t, dokumentoinnit sekä esimerkkisovelluksia. Ohjelmistovaatimuksina on Visual Studio 2010 ja NET Framework 4.0. Skeletal Viewer -esimerkkisovelluksia varten tarvitaan Microsoft DirectX 9 ja puheentunnistukseen Microsoft Speech Platform Runtime, versio 10.2. (Microsoft 2012).

OpenKinect

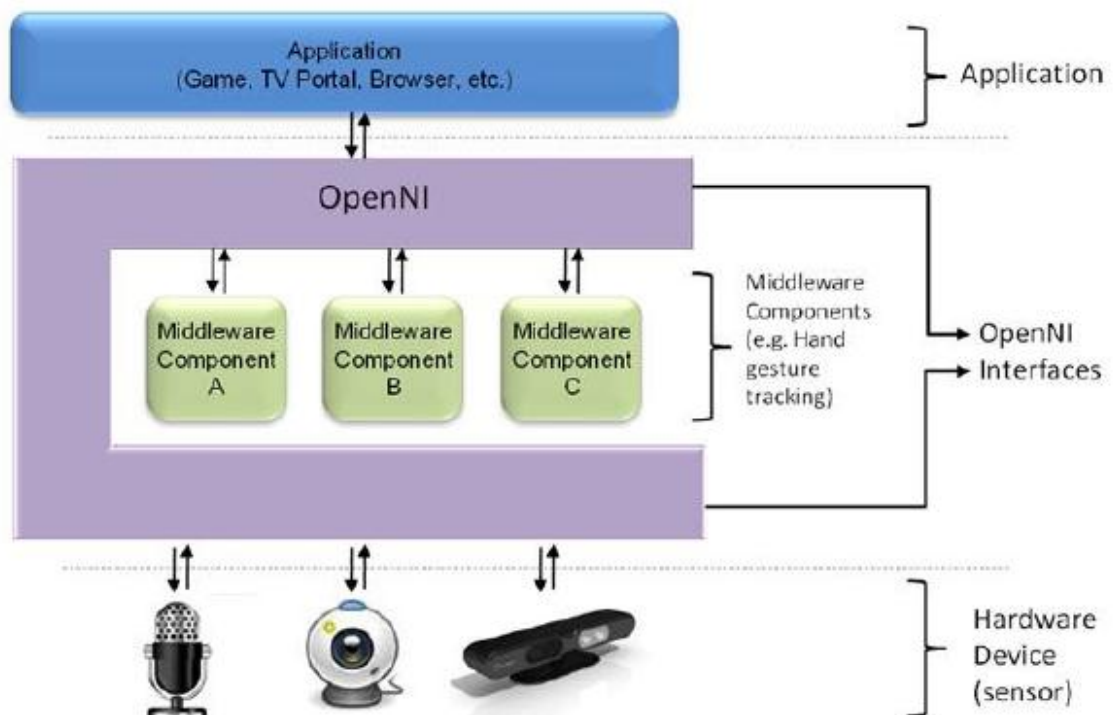
OpenKinect on avoin kommuuni, joka on kehittänyt avoimen lähdekoodin Kinect-ajureita, nimeltä "libfreenect". Ohjelmisto on kehitetty ensimmäisen toimivan avoimen lähdekoodin ajurin pohjalta, joka valmistui pian Kinectin julkaisun jälkeen

(Openkinect 2011). Libfreenect tukee Kinectin video- ja syvyyskuvan, moottorin, kiihtyvyyssanturin, sekä ledin käyttöä Windows, Linux ja Mac OS X –käyttöjärjestelmillä. (Openkinect 2012).

OpenNI

Open Natural Interaction (OpenNI) tarjoaa avoimen lähdekoodin SDK:n ja API:n, sekä toimii rajapintana (interface) liiketunnistimen, väliohjelmistojen ja käytettävien sovellusten välillä. Ohjelmiston tarkoituksena on helpottaa liiketunnistusta hyödyntävien ohjelmistojen kehitystä. Sovellus- ja pelikehittäjät, väliohjelmistojen kehittäjät ja sensoreiden valmistajat voivat OpenNI:a käyttämällä tehdä sovelluksia ja sensoreita yhteensopiviksi toisistaan riippumatta. (OpenNI 2011).

Uusin versio on tällä hetkellä 2.1, jossa on uudistettu API ja tuki Kinect for Windows SDK:ta käyttäville Kinect-liiketunnistimille. Uutta, sekä vanhempia 1.x ohjelmistoversioita voidaan käyttää Windows, Linux ja OS X -käyttöjärjestelmillä (OpenNI 2013). Asennukseen sisältyy esimerkkisovelluksia sekä käyttöohjeet.



KUVIO 9. OpenNI-ohjelmiston toiminta (OpenNI 2011).

NiTE

Natural Interface Technology for End-User (NiTE) on väliohjelmisto, joka liiketunnistimesta saaduista syvyys-, kuva- ja äänitiedoista muodostaa algoritmien avulla toimintoja, kuten käden paikantamista ja seuraamista, käyttäjän nivelten seuraamista ja eleiden tunnistamista. Ohjelmiston on kehittänyt PrimeSense, jonka prosessoria Kinect käyttää. Yhtiö valmistaa myös Kinectiä muistuttavia liiketunnistimia. (PrimeSense 2013.)

NiTE toimii yhdessä OpenNI-ohjelmiston kanssa ja uusin versio on 2.0. Käyttöjärjestelmäksi uudelle, sekä vanhemmille 1.x versioille sopii Windows, Linux tai OS X. (OpenNI 2013). Käyttöohjeet sekä esimerkkisovellukset sisältyvät asennukseen.

SensorKinect

SensorKinect on Kinectin ajuri, joka on kehitetty PrimeSense Sensor Module -ajurin pohjalta. Vanhat 1.x version OpenNI- ja NiTE-ohjelmistot eivät virallisesti tue Kinect-liiketunnistimia, joten SensorKinectin avulla Kinect saadaan toimimaan OpenNI- ja NiTE-ympäristössä. Uusin versio on 0.93 ja se on tarkoitettu käytettäväksi OpenNI 1.5.4.0- ja NiTE 1.5.2.21-versioiden kanssa. Windowsin lisäksi SensorKinect tukee Linux- ja OS X-käyttöjärjestelmiä. (Sensorkinect 2012).

3 KINECT UNITY 3D -OHJELMISTOSSA

3.1 Unity 3D

Unity on suosittu pelinkehitysalusta, joka soveltuu hyvin itsenäisille pelikehittäjille sekä studioille. Pelikehitykseen Unity tarjoaa monipuoliset työkalut mm. kohtaus-ten, eli skenejen rakentamiseen, koodaukseen ja animointiin. Käyttöliittymä koostuu eri ikkunoista, joita ovat project, hierarchy, inspector, scene, game ja toolbar. Näiden avulla hallitaan projektiin ja skeneen kuuluvia tiedostoja, peliobjektien ominaisuuksia, muokataan pelin skeneä sekä ajetaan ja testataan itse peliä.



KUVIO 10. Unity 3D -käyttöliittymä (Unity 2012).

Unityn Asset Store -sivustolla on tarjolla monia ilmaisia sekä maksullisia esimerk- kipelejä, peliobjekteja, tehosteita ja animaatioita. Foorumi-osiosta löytyy monia kymmeniä tuhansia käyttäjien keskusteluja Unityn eri aihealueista. Lisäksi Unityn dokumentaatio-sivuilla on laaja ohjeistus kaikista toiminnoista sekä pelien koo- dauksessa auttava scripting reference -osio, joka tarjoaa esimerkkejä koodin käytöstä.

Koodaukseen voi käyttää asennuksen mukana tulevaa MonoDevelop-ohjelmistoa ja ohjelmointikieli on JavaScript, C# tai Boo. Unitystä on julkaistu 4.0 versio, joka sisältää uuden animaatiotyökalun, reaaliaikaisen varjostuksen, DirecX 11 renderöinnin, päivitetyn partikkelisysteemin ja uusina julkaisualustoina Adobe Flash sekä Linux. (Unity 2013). Unity on saatavilla ilmaisena sekä maksullisena Pro-versiona, joita voidaan käyttää Windows- ja OS X-käyttöjärjestelmillä. Ilmainen versio riittää monelle, sillä se sisältää Unityn pääominaisuudet.

3.2 Ajurien valinta

Ajurivaihtoehtoja on neljä erilaista:

- Kinect for Windows SDK Beta 2 tai v1.6
- libfreenect
- SensorKinect 0.93 + OpenNI 1.5.4.0+ NiTE 1.5.2.21
- Kinect for Windows SDK 1.6 + OpenNI 2.1 + NiTE 2.0.

Työssä käytän Kinectiä Unity 3D:ssä, joten ajurien pitää olla tähän ympäristöön sopivat. Jotta Kinect saadaan toimimaan Unityssä, käytetään Zigfu Development Kit (ZDK) for Unity 3D -unitypakettia yhdessä OpenNI:n ja NiTE:n tai Kinect for Windows SDK:n kanssa (Zigfu 2013). Libfreenect ei siis sovellu käyttöön Unityssä.

Uusien OpenNI 2- ja NiTE 2 -ohjelmistojen käyttö Kinectin kanssa vaatii ajureiksi Kinect for Windows SDK:n, mutta Unity-käytössä pelkkä Kinect for Windows SDK riittää. Testatessa Kinect for Windows SDK 1.6 -ohjelmistoa Unityn 3.5.7 -versiossa, yhdessä ZDK 1.1 -version kanssa, lähes kaikki testiskenet toimivat. SimpleViewer-testiskenen avaaminen kaataa Unityn. Tämä johtuu Kinect for Windows-mallille tarkoitetusta Near-modesta, joka on oletuksena päälle kytkettynä kyseisessä skenessä. Unityn 4.0 versiossa skene voidaan avata ja Near-mode saadaan kytkettyä pois päältä, jolloin skene voidaan ajaa onnistuneesti Kinect for Xbox 360:lla.

ZDK:n uudessa 1.1 -versiossa lisätty Microsoft Kinect SDK -tuki koskee uusinta SDK 1.6 -versiota, joten vanhan Kinect SDK Beta 2:n käyttöä ei tueta. Sen sijaan

1.5-version OpenNI- ja NiTE-käyttö onnistuu ZDK:lla. Käytettäessä SensorKinect 0.93, OpenNI 1.5.4.0 ja NiTE 1.5.2.21 -ohjelmistoja, ZDK:n testiskenet toimivat samalla tavalla, kuten Kinect for Windows SDK 1.6:n kanssa. Kinect for Xbox 360 -liiketunnistimen käyttöön Kinect for Windows SDK 1.6, sekä OpenNI ja NiTE -yhdistelmä ovat molemmat hyviä vaihtoehtoja. Windows-mallin Kinectiä käytettäessä Kinect for Windows SDK 1.6 on parempi vaihtoehto, koska sillä voidaan hyödyntää Kinect for Windows -mallin Near-modea Unityssä.

3.3 Asennus ja testaus

Käytin opinnäytetyössä Windows 7 -käyttöjärjestelmää ja Unityn 3.5.7-versiota, koska uusille Unity 4.0:n ominaisuuksille ei ollut tarvetta työn suorituksen kannalta. Testausten perusteella valitsin työhön seuraavat ajurit ja ohjelmistot:

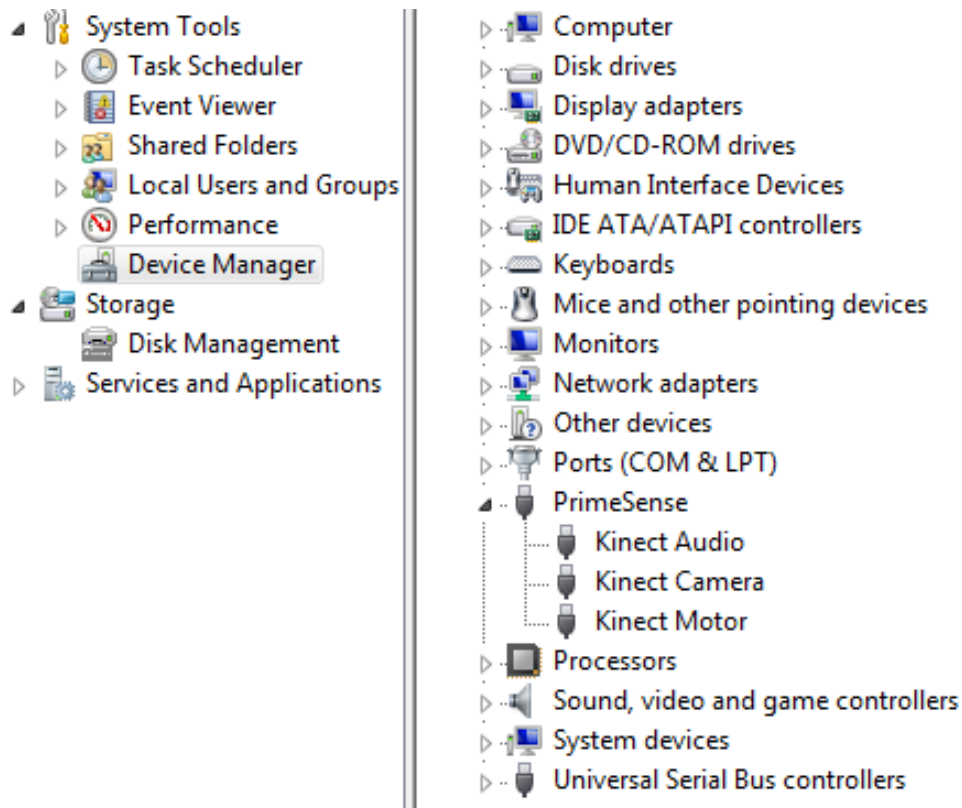
- OpenNI 1.5.4.0
- NiTE 1.5.2.21
- SensorKinect 0.93

Unityn asennuksessa valittavista komponenteista vain Unity tarvitaan, mutta MonoDevelop on myös hyvä asentaa ohjelmointipuolta varten. Tämän jälkeen valitaan asennuspolku, joka on oletuksena C-aseman Program Files. Asennukseen sisältyy myös Microsoft .NET Framework 4, ellei sitä ole vielä koneella.

User Account Control (UAC) valittaa tuntemattomasta julkaisijasta OpenNI- ja NiTE-asennuksissa, jos sitä ei ole kytketty pois päältä tietokoneen käyttäjienhallinnasta. SensorKinectin asennuksessa myös Windows Security antaa varoituksen, joka johtuu digitaalisen varmenteen (digital signature) puuttumisesta, sillä SensorKinect on käyttäjyhteisön luoma avoimen lähdekoodin ohjelmisto. Kyseiset ohjelmistot eivät ole siis haitallisia ja ne voidaan asentaa koneelle. OpenNI, SensorKinect ja NiTE kysyvät vain asennuspolkua, jonka annetaan olla oletusarvoilla.

OpenNI, SensorKinect ja NiTE asennuksien jälkeen voidaan testata Kinectin toimintaa. Liitetään Kinect USB-porttiin, jonka jälkeen tietokone tunnistaa laitteen

automaattisesti. Laitehallinnasta voidaan tarkistaa, että tietokone on tunnistanut Kinectin ja käyttää asennettuja ajureita. Kohdan PrimeSense avaamalla nähdään laitteet: Kinect Audio, Camera ja Motor. (KUVIO 11.)



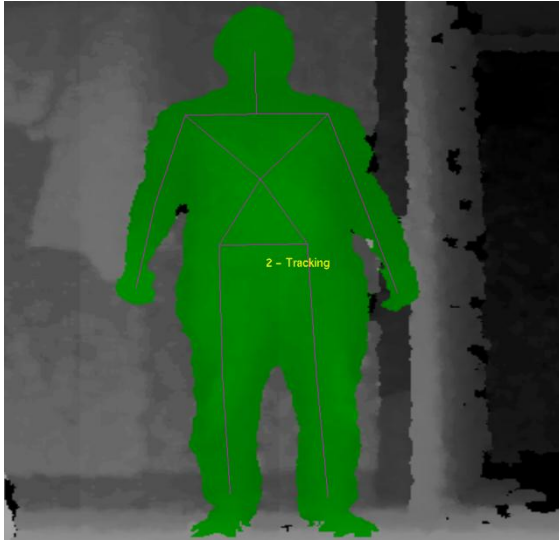
KUVIO 11. Laitehallinta näkymä.

Ajurien mukana tulevien esimerkkien avulla voidaan testata Kinectin eri toimintoja. OpenNI esimerkeissä on NiViewer.exe-tiedosto, jonka avaamalla nähdään molempien Kinectin kameroiden tuottamaa kuvaa. (KUVIO 12.) Kyseinen exe-tiedosto löytyy polusta: *C:\Program Files\OpenNI\Samples\Bin\Release*



KUVIO 12. Kinectin syvyys- ja videokuva.

Samassa kansiossa on myös NiUserTracker.exe, jonka avaamalla nähdään käyttäjän tunnistus, sekä luuranko. (KUVIO 13.) Kinect määrää automaattisesti jokaiselle seurattavalle käyttäjälle oman numeron, värin ja luurangon, muu ympäristö näkyy harmaana.



KUVIO 13. Käyttäjän tunnistus ja luuranko.

Kolmas testattava tiedosto on Sample-PointViewer.exe, jonka avaamalla voidaan testata käden seuranta. (KUVIO 14.) Seuranta aktivoituu, kun käyttäjä vilkuttaa kädellä Kinectin edessä. Xbox 360 -konsoli käyttää samantapaista aktivointia, kun sen päävalikkoa halutaan ohjata Kinectillä. Tiedosto löytyy PrimeSensen esimerkeistä, polusta: *C:\Program Files\PrimeSense\NITE\Samples\Bin\Release*.



KUVIO 14. Käyttäjän käden seuranta.

3.4 Zigfu Development Kit

Zigfu Development Kit (ZDK) on tarkoitettu Kinect-liiketunnistinta hyödyntävien ohjelmistojen kehitykseen. Saatavilla on HTML5 ja Javascript sekä Unity 3D -sovelluksia tukevat tiedostopakettit ja tulossa on myös Adobe Flash -tuki. ZDK for Unity3D tukee käyttöä Unity-sovelluksissa ja Unityn Web Player -sovelluksissa. Unity-paketti sisältää tarvittavien C# -tiedostojen lisäksi valmiita skenejä, joita voi käyttää pohjana omissa sovelluksissa. ZDK:n ilmainen trial-versio sisältää kaikki lisensoidun version ominaisuudet, mutta lisää käytettäviin sovelluksiin vesileiman ja on tarkoitettu vain ei-kaupalliseen käyttöön. Trial-version käyttöä ei ole aikarajotettu. (Zigfu 2013.)

ZDK:n käyttöä varten luodaan Unityssä aluksi uusi projekti valitsemalla yläpalkista File ja New Project. Aukaistuun Project Wizardiin määritellään projektille tallennuspolku, valitaan halutut unitypaketit ja painetaan Create. Paketteja voi lisätä projektiin myös myöhemmin, joten tässä vaiheessa niiden valitseminen ei ole välttämätöntä. Tämän jälkeen aukeaa luotuu projektiin tyhjä skene. Valitaan yläpalkista Assets, Import Package ja Custom Package. Haetaan koneelta ZDK-unitypaketti ja painetaan Open, josta aukeaa Importing package -ikkuna. Paketin sisällöstä kaikki tiedostot on valittu oletuksena, joten painetaan Import. Tyhjä skene voidaan tallentaa omaa projektia varten valitsemalla yläpalkista File, Save Scene. Tallennuspolku on oletuksena projektin Assets-kansio.

Tämän jälkeen Project-ikkunassa nähdään Unityyn tuotu ZDK-paketti, jonka SampleScenes-kansiosta voidaan testausta varten aukaista valmiita skenejä. Esimerkkejä on mm. 3D-hahmon liikuttamisesta, käden seurannan hyödyntämisestä, video- ja syvyys-kuvan käytöstä sekä kahden käyttäjän jaetun ruudun tilasta. Kun jokin yhdeksästä testiskenestä on aukaistu, liitetään Kinect tietokoneen USB-porttiin, ellei sitä ole vielä liitetty. Ajetaan testiskene painamalla Toolbar:sta Play-nappia, jonka jälkeen Kinect aloittaa käyttäjän seuraamisen. Jos SimpleViewer-skene halutaan ajaa, on käytettävä Unityn 4.0 versiota ja kytkettävä Near-mode pois päältä ZigFu-objektin Zig-skriptistä.

3.5 Skenen luominen Unityssä

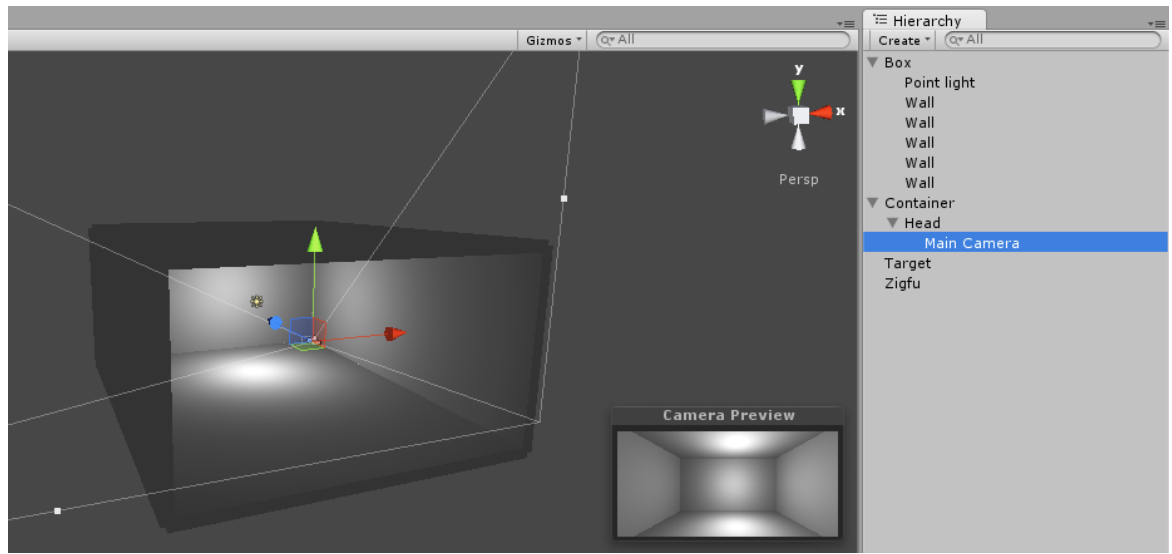
Tavoitteena on luoda kolmiulotteinen kuva, liikuttamalla kameraa skenessä käyttäjän pään sijainnin mukaan xyz-koordinaatistossa. Päätin luoda tätä varten skenen, jossa tarkastellaan kyljellään olevan laatikon sisältöä. Näytölle on tarkoitus luoda efekti, jossa laatikko näyttäisi oikealta fyysiseltä tilalta, joka ulottuu näytön taakse. Efektiä voisi demonstroida esim. tarkastelemalla eri kulmista pöydällä kyljellään olevan laatikon sisältöä, jossa laatikon reunat vastaisivat näytön kehystä. Skenessä kamera asetetaan seuraamaan tiettyä pistettä laatikon pohjalla ja liikkuminen tapahtuu laatikon reunojen sisäpuolella.

Laatikko voidaan tehdä skeneen yhdistämällä viisi Cube-objektia, joita luodaan valitsemalla Hierarchy-ikkunan yläosasta Create ja Cube. Inspector-ikkunan Scale-arvoja muuttamalla, kuutiosta tehdään laatikon seinä, jonka nimeksi laitetaan Wall. Käyttämäni näytön kuvasuhde on 16:9, joten laitetaan laatikon arvoiksi korkeus 9, leveys 16 ja syvyys 20. Inspector-ikkunasta muuttamalla Position- ja Rotation-arvoja, saadaan seinät siirrettyä paikoilleen. Valitaan yläpalkista GameObject ja Create Empty, joka luo skeneen tyhjän objektin. Muutetaan objektin nimeksi Box ja siirretään seinät Hierarchy ikkunassa Box-objektin sisälle. Wall-objektit ovat nyt Box-objektin child-objekteja, joten laatikkoa on tarvittaessa helpompi käsitellä, koska kaikki seinät liikkuvat Box-objektin mukana.

Lisätään skeneen valonlähde, valitsemalla Hierarchy-ikkunan Create-valikosta Point Light -objekti. Asetellaan skenessä valonlähde ja Main Camera laatikon sisälle. Valon etäisyyttä saadaan lisättyä Inspector-ikkunasta, nostamalla Range-arvoa. Siirretään valonlähde Hierarchy-ikkunassa Box-objektin sisään. Tämän jälkeen luodaan tyhjä GameObject, joka myöhemmin asetetaan kameran kohteeksi. Vaihdetaan nimeksi Target ja asetellaan objekti skenessä keskelle laatikon pohjaa.

Luodaan käyttäjän seuranta varten tyhjä GameObject, jonka sisällä pidetään käyttäjän luurankomallia seuraavat objektit. Vaihdetaan nimeksi Container ja asetellaan se skenessä samaan kohtaan Main Cameran kanssa. Luodaan toinen tyhjä GameObject ja vaihdetaan nimeksi Head. Työssä käytetään vain käyttäjän pään

seuranta, joten muille ruumiinosille ei tarvita seuraavia objekteja. Asetellaan Head samaan kohtaan skenessä Main Cameran ja Containerin kanssa. Hierarchy-ikkunassa siirretään Head-objekti Containerin sisään ja Main Camera Head-objektin sisään. Zigfun skriptien käyttöä varten luodaan Zigfu niminen tyhjä Game Object. Tämän jälkeen skenessä ovat kaikki tarvittavat objektit, eli Box, Point light, Container, Head, Main Camera, Target ja Zigfu. (KUVIO 15.)



KUVIO 15. Objektit skenessä.

3.6 Tarvittavat skriptit

ZDK-unitypaketti sisältää muutamia C#-skriptejä, jotka ovat Kinectin toiminnan kannalta välttämättömiä. Käytetään skenessä Zig-, ZigEngageSingleUser- ja ZigSkeleton-skriptejä, jotka löytyvät Project-ikkunassa ZigFu, Scripts -kansioden alta. Liitetään Zigfu-objektiin Zig-skripti, jonka asetuksista on mahdollista säätää Kinect- ja ajuriasetuksia. Samaan objektiin liitetään myös ZigEngageSingleUser-skripti, joka huolehtii käyttäjän seuraamisesta. Lisätään skriptiin Engaged Users -kohtaan Hierarchy-ikkunasta Container-objekti, jolla luurangon seuraaminen saadaan yhdistettyä Containerin alla oleviin objekteihin.

Liitetään Container-objektiin ZigSkeleton-skripti, joka sisältää kaikki käyttäjän luurankomallia seuraavat objektit, sekä seurantaan liittyvien asetusten säädöt.

Lisätään skriptin Head-kohtaan Hierarchy-ikkunasta Head-objekti, jolloin ohjelman ajossa Head-objekti ja Main Camera liikkuvat käyttäjän pään sijainnin mukaan. Main Cameraan liitetään ExitOnEscape-skripti, jonka avulla voidaan ohjelman ajo keskeyttää painamalla Escape-näppäintä ja ottaa kuvakaappauksia painamalla F10. Skripti löytyy Project-ikkunassa ZigFu, SampleScenes, _Data, Scripts -kansioiden alta.

Tämän jälkeen skeneä voidaan testata Kinectillä, painamalla Toolbar:sta Play-nappia. Tarvittavat skriptit ovat käytössä, mutta näiden asetuksia ei vielä ole muutettu, joten kamera kääntyy käyttäjän kääntyessä. Testillä varmistetaan vain käyttäjän tunnistuksen ja Kinectin toiminta skenessä. Seuraavaksi säädetään ZDK-skriptien asetuksia, sekä luodaan uusi skripti Unityn MonoDevelop-ohjelmistolla. Uuden skriptin luominen tapahtuu painamalla Project-ikkunassa Create ja C# Script.

Main Camera on asetettava tarkkailemaan laatikon pohjalla olevaa Target-objektia. Unityssä tämä onnistuu helposti yksinkertaisella LookAt-komennolla.

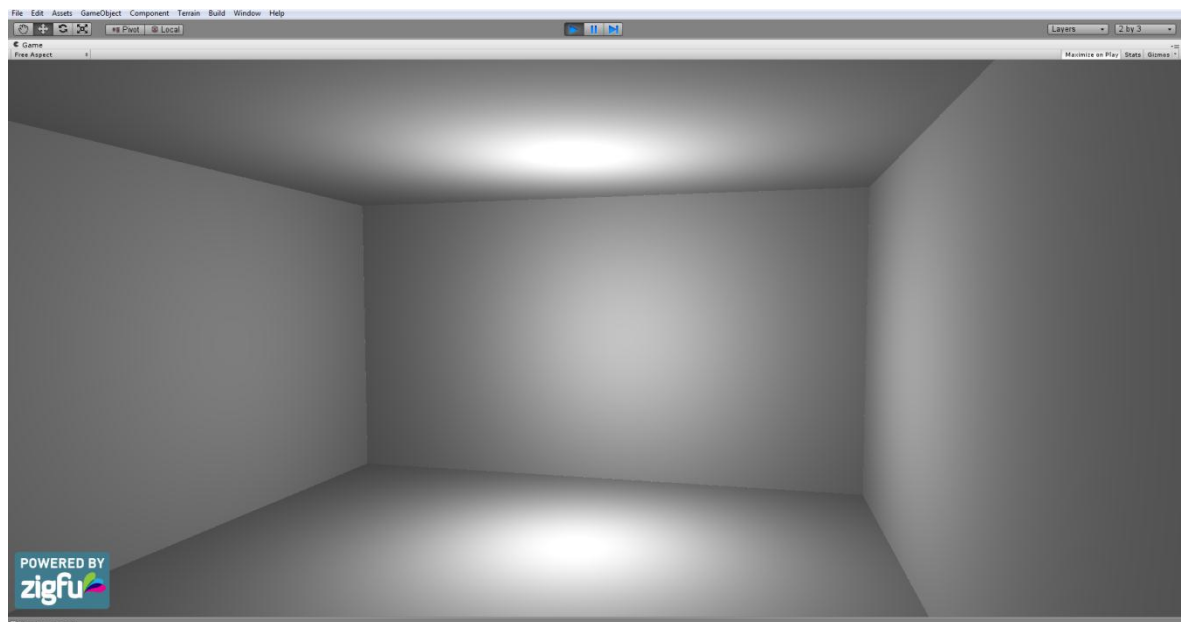
```
private Transform target;

void Start () {
    target = GameObject.Find("Target").transform;
}

void LateUpdate () {
    transform.LookAt(target);
}
```

Alussa on target-muuttujan määrittely, johon Start-funktiossa asetetaan laatikon pohjalla olevan Target-objektin Transform-arvot. Start-funktio suoritetaan vain yhden kerran ohjelman ajon alussa ennen Update-funktioita. LateUpdate-funktiossa käytetään transform.LookAt -komentoa, jonka kohteeksi on asetettu target-muuttuja. Koska Update-funktioita käytettäessä objektien seuraamisessa ilmenee häiritsevää kameran tärinää, käytetään LateUpdate-funktioita, joka ajetaan viimeisenä kaikkien Update-funktioiden jälkeen. Liitettäessä LookAt-skripti Main Cameraan, pysyy kamera aina Target-objektiin kohdistuneena.

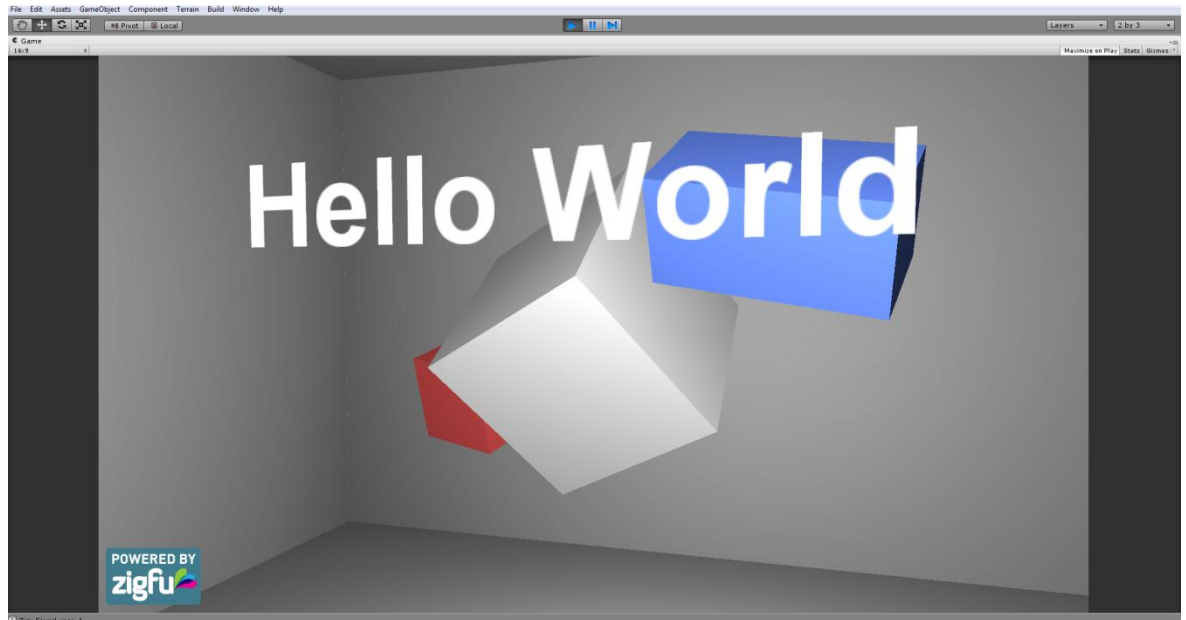
Valitaan Container-objektiin liitetystä ZigSkeleton-skriptistä käyttöön Update Joint Position- ja Update Root Position ja poistetaan käytöstä Update Orientation. Näillä asetuksilla Main Camera liikkuu skenessä käyttäjän pään sijainnin mukaan, eikä yritä kääntyä käyttäjän mukana. Saman skriptin Scale arvoista muutetaan X- ja Y-arvoja suuremmaksi, jolloin kamera liikkuu enemmän skenessä. Näiden arvojen sopiva suuruus riippuu käytettävän näytön koosta siten, että suurempaa näyttöä varten pitää käyttää isompia arvoja. Liian suurilla Scale arvoilla kamera liikkuu laatikon ulkopuolelle. Z-arvo voidaan asettaa nolnaan, sillä kameralle ei tarvita liikettä syvyys-suunnassa. (LIITE 1.) Skenessä on nyt kaikki tarvittavat objektit ja skriptit, sekä asetukset, joten ajetaan skene painamalla Toolbar:sta Play. (KUVIO 16.)



KUVIO 16. Valmiin skenen ajo Unityssä.

Tehdään skeneen vielä jonkin verran muutoksia, lisäämällä uusia objekteja ja vaihtamalla seurattavaa kohdetta. Luodaan kolme Cube-objektia, sekä 3D Text-objekti ja asetellaan nämä laatikon keskelle. Vaihdetaan myös Target-objektin sijainti skenessä laatikon keskelle. Käännellään Cube-objekteja jonkin verran muuttamalla Inspector-ikkunassa näiden Rotation arvoja. Muutetaan kuutioiden väriä luomalla Project-ikkunassa Create, Material. Vaihdetaan Inspector-ikkunassa materiaalin Main Color halutuksi ja vedetään materiaali värjättävään objektiin. Box-objektia voidaan suurentaa muuttamalla Inspector-ikkunassa Scale arvoja isom-

maksi. Luodaan vielä uusi Point Light -objekti, joka asetellaan valaisemaan kuu-
tioita. (LIITE 2.) Container-objektin ZigSkeleton-skriptistä muutetaan Scale arvoja
lisäämällä Z-arvo, joka antamaan kameralle liikkeen syvyys-suunnassa. Ajetaan
skene painamalla Toolbar:sta Play. (KUVIO 17.)



KUVIO 17. Muokatun skenen ajo Unityssä.

4 TULOKSET JA POHDINTA

Kinectin ajurivaihtoehtoja oli useita, joista vain muutamat olivat yhteensopivia ZDK:n kanssa Unity 3D käyttöä varten. Eri ajuriyhdistelmien testaaminen oli aikaa vievää, mutta asennukset sujuivat ongelmitta ja niiden tarjoamat testisovellukset toimivat hyvin. Kinectin käyttö Unity 3D -sovelluksessa on helppo aloittaa, sillä ZDK-unitypaketti sisältää monipuolisia esimerkkejä, joita voi hyödyntää pohjana omia sovelluksia kehittäessä. Valmiissa ZDK-skripteissä on useita eri säätömahdollisuuksia, joita voi tarvittaessa muokata Unityn Inspector-ikkunassa. Käyttäjän liikkeiden seuraaminen Unityssä toimi hyvin ja vain nopeasti liikuttaessa oli kamerassa havaittavissa pientä viivettä.

Skenessä oli tarkoitus saada aikaan kolmiulotteinen vaikutelma, joka laatikkoa käytettäessä jäi melko heikoksi. Koska kameran annetaan liikkua vapaasti, laatikon reunat eivät pysy kiinni näytön kehyksessä. Tästä syystä laatikko näyttää näytön taakse ulottuvan tilan sijaan irralliselta, eikä oikeaa 3D-vaikutelmaa synny. Käyttäjän katsellessa näyttöä pienessä kulmassa, kuva näyttää ihan hyvältä, mutta liikuttaessa vasemmalle tai oikealle kulma kasvaa ja 3D-vaikutelma heikenee. Testasin kameran liikutteluun eri menetelmiä ja asetuksia, mutta nämä eivät tuottaneet parempia tuloksia.

Päätin muokata skeneä ja hakea 3D-vaikutelmaa, luomalla laatikon keskelle kameralla seurattavia objekteja. Muokatussa skenessä 3D-vaikutelma oli paljon parempi. Laatikon sijaan huomio kiinnittyi keskellä oleviin objekteihin ja käyttäjä pystyi arvioimaan niiden etäisyyksiä, tarkastelemalla kohteita eri kuvakulmista. Lisäämällä skriptiin Z-arvon, objekteja pystyi tarkastelemaan myös lähietäisyydeltä, joka lisäsi syvyysvaikutelmaa. Vastaavasta skenestä olisi mahdollista luoda näyttävämpi käyttämällä erillistä 3D-mallinnusohjelmaa objektien luonnissa.

Sopivana jatkokehityksenä voisi olla käyttäjän vuorovaikutuksen lisääminen, esim. liikuttamalla skenessä olevia objekteja käsien avulla. Kinectiä olisi mahdollista hyödyntää myös ohjaimena peliprojektissa. Ohjaamisessa Kinectiä voi käyttää

yksinään tai yhdessä käsissä pideltävän peliohjaimen kanssa ja myös puheentunnistuksen hyödyntäminen olisi mahdollista.

Opinnäytetyössä sain hyvän kuvan liiketunnistimissa käytettävästä sensoritekniikasta ja Kinectin toiminnasta, sekä laitteistosta. Kinectin ajurit, käyttö Unityssä, sekä käyttöön liittyvät rajoitukset tulivat myös tutuksi. Vähän oli myös koodausta C#-kielellä, mutta suurin osa oli valmiiksi saatavilla ZDK-paketissa.

LÄHTEET

Champy, A. 2007. Elements of Motion: 3D Sensors in Intuitive Game Design. Pdf-tiedosto. Saatavissa: http://www.newarkinone.thinkhost.com/brands/promos/leading_edge/Analog_Devices_Accelerometer_4104.pdf. Luettu 9.2.2013.

Ebay. 2013. Playstation Move. Www-dokumentti. Saatavissa: <http://www.ebay.com/itm/PS3-Playstation-Move-Starter-Pack-2-Brand-New-/140689455238>. Luettu 11.2.2013.

Gdgt. 2013. Nintendo Wii Remote controller. Www-dokumentti. Saatavissa: <http://gdgt.com/nintendo/wii/remote-controller/>. Luettu 11.2.2013.

Gdgt. 2013. Microsoft Kinect. Www-dokumentti. Saatavissa: <http://gdgt.com/microsoft/kinect/>. Luettu 11.2.2013.

Ifixit. 2010. Microsoft Kinect Teardown. Www-dokumentti. Saatavissa: <http://www.ifixit.com/Teardown/Microsoft+Kinect+Teardown/4066/2>. Luettu 8.2.2013.

KinectHacks. 2013. FFAST on your PC. Www-dokumentti. Saatavissa: <http://www.kinecthacks.com/guides/install-faast-on-your-pc/>. Luettu 8.2.2013.

Lowensohn, J. 2011. Timeline: A look back at Kinect's history. Www-dokumentti. Saatavissa: http://news.cnet.com/8301-10805_3-20035039-75.html. Luettu 7.2.2013.

Lukiegames. 2013. NES Nintendo Power Glove Set. Www-dokumentti. Saatavissa: http://www.lukiegames.com/NES-Nintendo-Power-Glove-Set_p_16206.html. Luettu 11.2.2013.

Microsoft. 2012. Kinect Sensor. Www-dokumentti. Saatavissa: <http://msdn.microsoft.com/en-us/library/hh438998.aspx>. Luettu 11.2.2013.

Microsoft. 2012. Kinect for Windows SDK. Www-dokumentti. Saatavissa: <http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx>. Luettu 9.2.2013.

Microsoft. 2012. Kinect for Windows SDK Beta 2. Www-dokumentti. Saatavissa: <http://www.microsoft.com/en-us/download/details.aspx?id=27876>. Luettu 8.2.2013.

Miles, R. 2012. Start Here - Learn the Kinect API. Sebastopol: O'Reilly Media, Inc.

Openkinect. 2012. OpenKinect libfreenect. Www-dokumentti. Saatavissa: <https://github.com/OpenKinect/libfreenect>. Luettu 9.2.2013.

Openkinect. 2011. OpenKinect History. Www-dokumentti. Saatavissa: <http://openkinect.org/wiki/History>. Luettu 9.2.2013.

OpenNI. 2011. OpenNI Documentation. Www-dokumentti. Saatavissa: <http://kinectcar.ronsper.com/docs/openni/introduction.html>. Luettu 9.2.2013.

OpenNI. 2013. OpenNI SDK. Www-dokumentti. Saatavissa: <http://www.openni.org/openni-sdk/>. Luettu 10.2.2013.

OpenNI. 2013. NiTE 2. Www-dokumentti. Saatavissa: <http://www.openni.org/files/nite/>. Luettu 10.2.2013.

Plunkett, L. 2010. Here Are Kinect's Technical Specs. Www-dokumentti. Saatavissa: <http://kotaku.com/5576002/here-are-kinects-technical-specs>. Luettu 11.2.2013.

PrimeSense. 2013. NiTE Middleware. Www-dokumentti. Saatavissa: <http://www.primesense.com/solutions/nite-middleware/>. Luettu 9.2.2013.

Retro Gaming Nation. 2012. The Power Glove. Www-dokumentti. Saatavissa: <http://retrogamingnation.blogspot.fi/2012/08/the-power-glove.html>. Luettu 7.2.2013.

Sensorkinect. 2012. SensorKinect. Www-dokumentti. Saatavissa: <https://github.com/avin2/SensorKinect>. Luettu 9.2.2013.

Unity. 2012. Learning the Interface. Www-dokumentti. Saatavissa: <http://docs.unity3d.com/Documentation/Manual/LearningtheInterface.html>. Luettu 11.2.2013.

Unity. 2013. Unity scripting. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/workflow/scripting>. Luettu 11.2.2013.

Unity. 2013. Unity 4.0.1. Www-dokumentti. Saatavissa: <http://unity3d.com/unity/whats-new/unity-4.0.1>. Luettu 11.2.2013.

VGCL. 2013. History of the Game Controller – Wireless Controllers - Advanced. Www-dokumentti. Saatavissa: <http://www.videogameconsolelibrary.com/art-controller.htm#page=wireless2>. Luettu 7.2.2013.

VGCL. 2013. History of the Game Controller – Motion Sensing. Www-dokumentti. Saatavissa: <http://www.videogameconsolelibrary.com/art-controller.htm#page=motion>. Luettu 7.2.2013.

Zhang, Z. 2012. Microsoft Kinect Sensor and Its Effect. Pdf-tiedosto. Saatavissa: <http://research.microsoft.com/en-us/um/people/zhang/Papers/Microsoft%20Kinect%20Sensor%20and%20Its%20Effect%20-%20IEEE%20MM%202012.pdf>. Luettu 8.2.2013.

Zigfu. 2013. ZDK for Unity3D. Www-dokumentti. Saatavissa: <http://zigfu.com/en/zdk/unity3d/>. Luettu 10.2.2013.

