

Teemu Patana

RAEPUHALLUSROBOTIN SIMULOINTIYMPÄRISTÖ

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteiden ala
Tietojenkäsittelyn koulutusohjelma
30.5.2012

| | |
|--|---|
| Koulutusala Luonnontieteiden ala | Koulutusohjelma Tietojenkäsittelyn koulutusohjelma |
| Tekijä(t) Teemu Patana | |
| Työn nimi Raepuhallusrobotin simulointiympäristö | |
| Vaihtoehtoiset ammattiopinnot Peliohjelmointi | Ohjaaja(t) Mikko Romppainen Toimeksiantaja Blastman Robotics Oy |
| Aika 30.5.2012 | Sivumäärä ja liitteet 31 |
| <p>Tässä opinnäytetyössä kerrotaan raepuhallusrobottien toiminnasta ja simuloinnista. Aluksi kerrotaan lyhyesti robotiikasta ja robottien historiasta, jonka jälkeen siirrytään robottien käyttötarkoituksiin sekä esitellään opinnäytetyön toiminnallisessa osuudessa käytettävän raepuhallusrobotin käyttämiä tiloja.</p> <p>Graafista simulointia varten kolmiulotteisesta grafiikasta ja sen animoinnista selitetään lyhyesti tärkeimmät asiat kuten piirtoputki ja varjostimet. Animointia varten kerrotaan hieman muutamista yleisimmistä interpolointimenetelmistä. Myös käyttöliittymistä sekä parista yleisesti käyttöliittymien toteuttamiseen tarkoitettua kirjastosta kerrotaan olennaisimmat asiat.</p> <p>Toiminnallisessa osuudessa käytettäviä tiedostomuotoja ja kirjastoja tutkitaan syvällisemmin käymällä läpi niiden pääasialliset käyttötarkoitukset. Vertauksena toiminnallisen osuuden työhön käytetään muutamia valmiita simulointiohjelmistoja, joista kerrotaan myös tärkeimmät ominaisuudet.</p> <p>Opinnäytetyön toiminnalliseen osuuteen kuului toteuttaa yksinkertainen simulointiohjelma robotteja varten. Pää-tarkoituksena simulointiohjelmalla on toteuttaa ohjelmia, joita on mahdollista ajaa oikeassa elämässä toimivilla raepuhallusroboteilla.</p> | |
| Kieli | Suomi |
| Asiasanat | Kajak3D, Qt, robotiikka |
| Säilytyspaikka | <input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto |

| | |
|--|---|
| School Business | Degree Programme Business Information Technology |
| Author(s) Teemu Patana | |
| Title Simulation Environment for Abrasive Blasting Robots | |
| Optional Professional Studies Game programming | Instructor(s) Mikko Romppainen |
| | Commissioned by Blastman Robotics Ltd |
| Date May 30th 2012 | Total Number of Pages and Appendices 31 |
| <p>The topic of this thesis is about the functionalities and simulations of abrasive blasting robots. The thesis will first briefly explain robotics and the history of robots, and then it will go through the most common usages for robots and the robot's working environment which is used in the practical part.</p> <p>For graphical simulations, the main points of three-dimensional graphics and its animation will be briefly explained. This includes the graphics pipeline, shaders and a few common interpolation methods which are used in generating computer animations. This thesis will also explain about user interfaces and some of the most commonly used development libraries for creating user interfaces.</p> <p>The libraries which are used in the practical part are explained in detail by describing the main features, what kind of programs and applications they are used in. And as a comparison to the practical part, this thesis will go through some of the most common simulation programs by explaining their most important features.</p> <p>The practical part of this thesis is about designing and developing a simple simulation software for robots. The main purpose of the simulation software is to create programs which are capable of being run in real life abrasive blasting robots.</p> | |
| Language of Thesis | Finnish |
| Keywords | Kajak3D, Qt, robotics |
| Deposited at | <input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences |

SISÄLLYS

| | |
|---|----|
| 1 JOHDANTO | 1 |
| 2 TAUSTATEORIAA | 2 |
| 2.1 Robotit | 2 |
| 2.2 3D-grafiikka | 5 |
| 2.2.1 Piirtoputki | 6 |
| 2.2.2 Animointi | 8 |
| 2.3 Käyttöliittymät | 11 |
| 2.4 Kirjastot | 11 |
| 2.4.1 Kajak3D | 12 |
| 2.4.2 Qt | 13 |
| 2.5 Tiedostomuodot | 16 |
| 3 YLEISESTI KÄYTETTYJÄ ROBOTISIMULOINTIOHJELMISTOJA | 17 |
| 3.1 RobotStudio | 17 |
| 3.2 Microsoft Robotics Developer Studio | 19 |
| 4 PROJEKTI | 20 |
| 4.1 Simulointi | 20 |
| 4.2 Vaatimukset | 21 |
| 4.3 Suunnittelu | 21 |
| 4.4 Toteutus | 21 |
| 4.5 Näkymän rakenne | 24 |
| 4.6 Testaus | 25 |
| 4.7 Tulosten vertailu vaatimuksiin | 27 |
| 5 POHDINTA | 28 |
| LÄHTEET | 29 |
| LIITTEET | |

SYMBOLILUETTELO

| | |
|-------------|---|
| CAD | Computer Aided Design |
| Direct3D | Microsoftin kehittämä grafiikkarajapinta |
| GTK+ | GIMP Toolkit, alun perin GIMP-kuvankäsittelyohjelmaa varten luotu käyttöliittymäkirjasto |
| HTML | HyperText Markup Language, verkkosivuilla käytettävä merkintäkieli |
| Kajak3D | Kajaanin ammattikorkeakoulussa kehitettävissä oleva pelimoottori |
| LERP | Linear interpolation, lineaarinen interpolaatio |
| M3G | Mobile 3D Graphics, Java-ohjelmointikielelle tarkoitettu rajapinta grafiikkaa tuottavia ohjelmia varten |
| OpenGL | Open Graphics Library, monella alustalla toimiva grafiikkarajapinta |
| Qt | Nokian kehittämä alustariippumaton sovelluskehitysalusta |
| Renderointi | Kuvan luomista mallista tietokoneohjelman avulla |
| SLERP | Spherical linear interpolation, yleinen kvaternioina esitettyjen asentojen interpolointitapa |
| STL | Standard Tessellation Language, 3D-malleille tarkoitettu tiedostomuoto |
| XHTML | eXtensible HyperText Markup Language |

1 JOHDANTO

Tässä opinnäytetyössä kerrotaan pääasiassa robotiikasta, tietokoneella toteutettavasta kolmiulotteisesta grafiikasta ja sen animoinnista sekä graafisten käyttöliittymien toteuttamisesta. Opinnäytetyön aihe löytyi Blastman Robotics Oy:n kautta, sillä heillä on suuri tarve omiin tarpeisiin räätälöidylle helppokäyttöiselle simulointiohjelmalle raepuhallusrobotteja varten.

Tietokonegrafiikka on tärkeässä osassa tämän opinnäytetyön toiminnallisen osuuden ohjelman toteutuksessa, joten kolmiulotteisesta grafiikasta sekä sen animoinnista kerrotaan tärkeimmät asiat. Käyttöliittymiä varten kerrotaan lyhyesti parista yleisesti käytetystä käyttöliittymäkirjastosta käymällä läpi niiden olennaisimmat ominaisuudet ja käyttötarkoitukset.

Toiminnallisessa osuudessa tässä opinnäytetyössä toteutettiin ohjelma joka mahdollistaa yksinkertaisen raepuhallusrobottien simulaation. Vertauksena toteuttavaan ohjelmaan kerrotaan muutamista yleisimmistä roboteille tarkoitetuista simulointiohjelmistoista tärkeimmät ominaisuudet.

Graafisten käyttöliittymien toteuttamisesta ei ollut paljoa aikaisempaa kokemusta, joten perehtyminen niiden työkaluihin sekä niiden oppimiseen vaati aikansa. Myös itse robotiikkaan ei ollut entuudestaan kovin tuttua, joten robottien toimintaan tutustuminen vei jonkin verran aikaa.

Valitsin aiheen koska projekti vaikutti kiinnostavalta ja siihen sisältyi monia asioita, jotka vaikuttivat mielenkiintoisilta sekä hyviltä aiheilta opiskelun kannalta. Graafisia käyttöliittymiä vaativat työkalut ovat tarpeellista myös tietokonepelien kehityksessä, sillä esimerkiksi kenttien suunnittelu on yleensä melko hidasta ilman tehokkaita työkaluja.

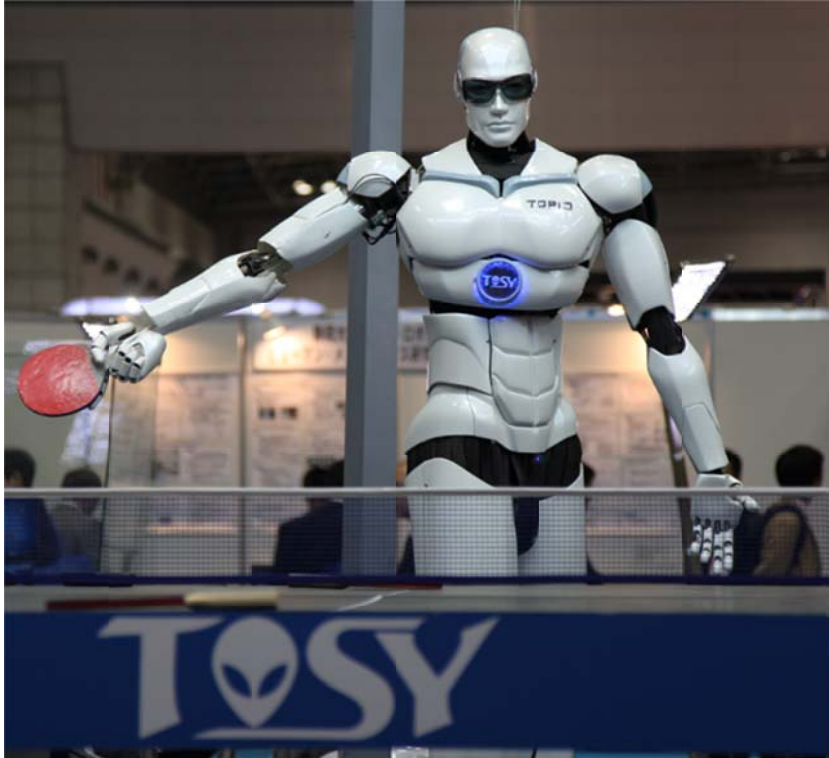
2 TAUSTATEORIAA

Robottiikka on teknologian tieteenhaara joka käsittelee robottien suunnittelua, rakentamista ja käsittelemistä (Oxford Dictionaries 2012). Yhdysvaltalaisen robotiikkayhdistys, The Robot Institute of America, määrittelee robotin monipuolisena uudelleen ohjelmoitavana mekaanisena laitteena, joka on suunniteltu pääasiassa liikuttamaan kappaleita tai osia (Pikkarainen, Räsänen, Pikkarainen & Heikkinen 2012).

2.1 Robotit

Teollisuusrobottien kehitys alkoi Yhdysvalloissa 1960-luvulla, mutta vasta 1990-luvun jälkeen niitä on saatu tuotua enemmän massatuotantoon. Syynä teollisuusrobottien yleistymiseen on hintojen laskeminen, sillä robottien hinnat puolittuivat siirtyessä 1990-luvun alusta 2000-luvulle. (Wikipedia 2012 a.)

Teollisuusrobotit ovat käytössä erityisesti metalli-, muovi- ja elintarviketeollisuudessa. Teollisuusrobotteja on nykyään maailmassa noin miljoona ja niitä liikkuu markkinoilla noin 100 000 kappaletta vuosittain. Suomessa markkinaluvut ovat kuitenkin vielä hieman alhaisia, sillä teollisuusrobotteja asennetaan noin 300 kappaletta vuodessa. (Suomen Robotiikkayhdistys Ry 2012 a.)



Kuvio 1. TOPIO 3.0 -robotti Tokion kansainvälisessä robottinäyttelyssä (TOPIO 2009.)

Humanoidirobotit ovat robotteja joiden ulkonäkö muistuttaa ihmiskehoa. Humanoidirobotit ovat suunniteltu käyttäytymään ihmismäisesti ennalta määrätyissä tehtävissä. Kuviossa 1. näkyy TOPIO 3.0, jonka tarkoituksena on matkia tapaa jolla ihminen pelaa pingistä. TOPIO 3.0:lla on korkeutta 188 senttimetriä, painoa 120 kilogrammaa sekä vapausasteita 39 kappaletta. (TOPIO 2009.)

Androidit ovat humanoidirobotteja, joilla tarkoitetaan robotteja jotka pyrkivät näyttämään entistä enemmän ihmismäiseltä. Vaikka androidi tulee kreikan kielen sanoista ”miehen kaltainen”, käytetään sitä yleensä kuvaamaan molempia sukupuolia. Naispuolisista androideista käytetään joskus sanaa gynoidi. (Wikipedia 2012 b.)



Kuvio 2. Työstettävä kappale ja raepuhallusrobotti kammiossa (Blastman Robotics Oy 2012.)

Raepuhallusrobotit toimivat pääasiassa omassa eristetyssä työskentelytilassa työstökappaleen parissa (Kuvio 2.). Robottien tarkoituksena on poistaa vaaralliset ja tapaturma-alttiit työtävät. Tällaisiin työtehtäviin kuuluvat esimerkiksi raskaiden tai kuumien kappaleiden käsittely sekä sellaiset työtehtävät, joissa käsitellään terveydelle vaarallisia aineita. Myös toistuvasti tapahtuvat työtehtävät saattavat aiheuttaa vammoja pitkällä aikavälillä. (Pikkarainen, Räsänen, Pikkarainen & Heikkinen 2012.)

Vaikka robottien tarkoituksena on vähentää mahdollisia vaaratilanteita, tuovat ne kuitenkin mukanaan uusia turvallisuusongelmia. Esimerkiksi huoltotöiden yhteydessä saattaa aiheutua vakaviakin vaaratilanteita. Robotti saattaa tehdä nopeita käännöksiä ja liikkeitä, joten sen käyttäytyminen saattaa vaikuttaa arvaamattomalta, varsinkin henkilöille jotka eivät ole entuudestaan työskennelleet kyseisen robotin parissa. Vaaratilanteisiin voi kuulua esimerkiksi iskujen saaminen tai puristuksiin joutuminen. (Pikkarainen ym. 2012.)



Kuvio 3. Raepuhallusrobotin kammio ulkopuolelta (Blastman Robotics Oy 2012.)

Raepuhallusrobotteja varten on olemassa omat työskentelykammiot. Kuviossa 3. näkyy Renforsin Rannan tiloissa sijaitseva kammio, joka on käytössä pääasiassa raepuhallusroboteille. Työskentelylle tarkoitettujen kammioiden tarkoituksena on parantaa turvallisuutta, jotta ulkopuoliset ihmiset eivät aiheuta vaaratilanteita joutumalla vahingossa työskentelemässä olevan robotin läheisyyteen. Kammio estää myös raepuhalluksessa käytettävän hiekan pölyn leviämisen ulkopuolelle. (Blastman Robotics Oy 2011.)

2.2 3D-grafiikka

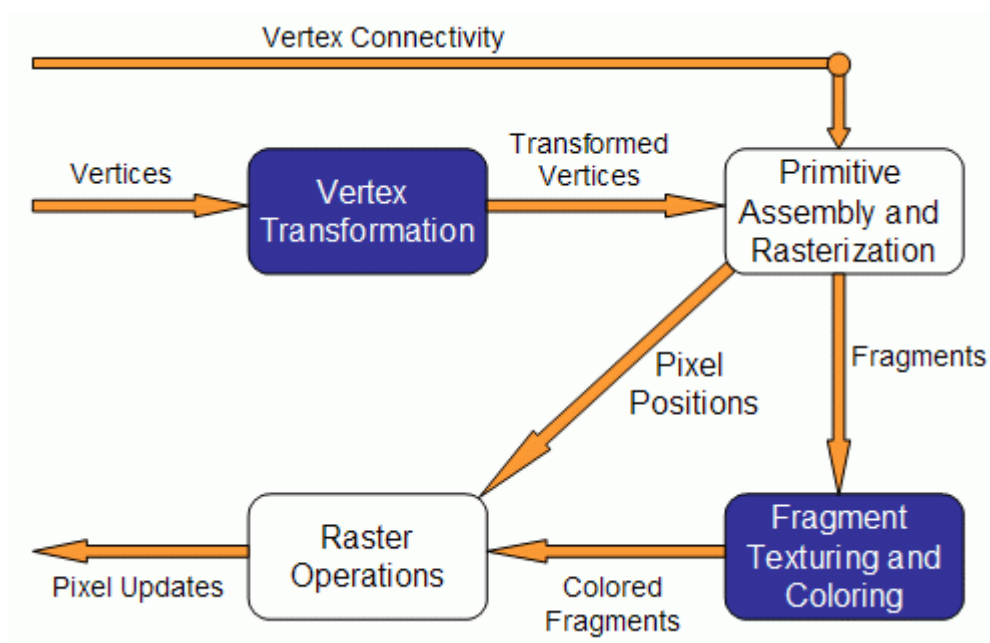
Tietokoneilla toteutetun grafiikan historia on vielä nuori, sillä ensimmäiset tietokonegrafiikkaa hyödyntävät sovellukset johtavat 1950-luvun vaihteeseen. Ja vaikka tietokonegrafiikka on tieteenalana vielä suhteellisen nuori, on sen aikana ehditty keksiä monia nerokkaita teknisiä ratkaisuja ja algoritmeja. Vasta parin viime vuosikymmenien aikana tietokonegrafiikan sovellukset ovat tulleet tavallisten ihmisten ulottuville. (Puhakka 2008.)

Tietokonegrafiikassa kolmiulotteiset mallit koostuvat yleensä kulmapisteistä, joiden välille muodostetaan halutut primitiivit. Kyseiset primitiivit voivat olla esimerkiksi kolmioita, viivoja tai vaikka pelkkiä pisteitä. Kulmapisteisiin voidaan liittää paikkatiedon lisäksi myös muutaakin tietoa. Esimerkiksi kulmapistekohtaiset pintakuvakoordinaatit ja normaalivektorit ovat tärkeässä osassa luomassa graafista ilmettä nykyisissä tietokonepeleissä. (Puhakka 2008.)

Normaalivektorilla tarkoitetaan vektoria, joka on kohtisuorassa tasoon nähden. Normaalivektorilla voidaan selvittää esimerkiksi kumpi puoli tasosta osoittaa eteen ja kumpi taakse. Normaalivektoreita käytetään useasti avuksi valaistuksessa. (Puhakka 2008.)

2.2.1 Piirtoputki

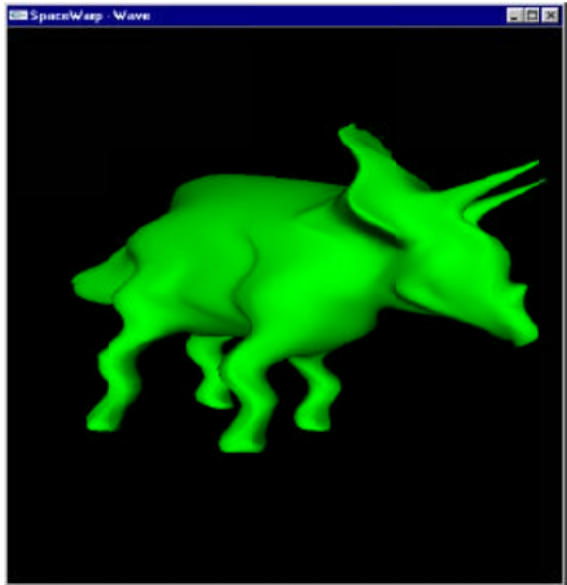
Piirtoputken tarkoituksena on luoda kaksiulotteinen kuva ennalta määritettyjen tietojen perusteella. Kyseisiä tietoja ovat esimerkiksi virtuaalinen kamera, kolmiulotteiset objektit, valonlähteet, varjostinyhtälöt sekä pintakuvat. Piirtoputki on siis reaaliajassa toimivan renderoinnin perustan toimintaväline. (Akenine-Möller, Haines & Hoffman 2008)



Kuvio 4. Yksinkertaistettu kaavio piirtoputkesta (Lighthouse 3D 2012 a.)

Kuviossa 4. on yksinkertaistettu kaavio piirtoputken toiminnasta. Piirtoputki koostuu useasta eri vaiheesta, joilla jokaisella on oma tarkoituksensa lopputuloksen saavuttamiseksi. (Akenine-Möller, Haines & Hoffman 2008.)

Varjostimet ovat näytönohjaimella suoritettavia ohjelmia, joiden tarkoituksena on korvata piirtoputken eri vaiheita. Varjostimet ovat tuoneet mukanaan paljon dynaamisuutta piirtämiseen, sillä ohjelmoijat voivat toteuttaa huomattavasti yksityiskohtaisemmin eri piirtoputken vaiheet. (Puhakka 2008.)



Kuvio 5. Kulmapistevarjostin toiminnassa (NVIDIA 2010.)

Kulmapistevarjostimen tarkoituksena on ottaa vastaan kulmapisteen tiedot ja muuttaa niitä haluttujen tietojen perusteella. Kulmapisteen paikkaa voidaan liikuttaa esimerkiksi kameran ja projektiomatriisin avulla. Kuviossa 5. on käytössä kulmapistevarjostin, joka muuttaa kulmapisteiden paikkaa x-akselilla y-akselin suhteen. (NVIDIA 2010.)



Kuvio 6. Geometriatesselaatio Unigine-moottorissa (Unigine 2009.)

Geometriavarjostin tuli mahdolliseksi Direct3D-rajapinnassa version 10:n myötä (Windows Dev Center 2012). Vastaava tuki geometriavarjostimelle sisällytettiin OpenGL-rajapintaan versiossa 3.2, mutta se oli jo aikaisemmin käytettävissä OpenGL-rajapinnan laajennosten avulla. (OpenGL Wiki 2010.)

Yleinen käyttötapa geometriavarjostimelle on esimerkiksi geometriatesselaatio, jossa tarkoituksena on luoda ajon aikana lisää kulmapisteitä kameralle näkyviin malleihin saadakseen näyttävämpiä objekteja tuomalla lisää tarkkuutta. (NVIDIA 2012.) Kuviossa 6. näkyy oletusmalli sekä tesseloitu versio samasta mallista. Mitä lähempänä kamera on tesseloitavaa mallia, sitä enemmän mallin muodot ja yksityiskohdat tulevat geometriatesselaatiossa esille. (Unigine 2009.)

Geometriavarjostimen tarkoituksena on luoda uusia grafiikkaprimitiivejä vastaanotetuista primitiiveistä, kuten pisteistä, viivoista tai kolmioista. Geometriavarjostin suoritetaan kulmapistevarjostimen jälkeen. (NVIDIA Developer Zone 2007 a.)



Kuvio 7. Kuvapistevarjoitin toiminnassa (Lighthouse 3D 2012 b.)

Kuviossa 7. on käytössä yleisesti käytetty sarjakuvamainen efekti, joka toteutetaan pääasiassa kuvapistevarjoitimella (Lighthouse 3D 2012 b.). Kuvapistevarjoitimella tapahtuvat kuvapistekohtaiset laskutoimitukset, joilla on tarkoituksena laskea lopullinen väriarvo kyseiselle kuvapisteelle. Kuvapistevarjoitimella on mahdollista käsitellä vain yhtä kuvapistettä kerrallaan. (Akenine-Möller, Haines & Hoffman 2008.)

2.2.2 Animointi

Tietokoneanimaatioiden toteuttaminen on lainannut suurimman osan ideoista tavanomaisesta animaatioiden toteuttamisesta. Avainkehysien käyttö on tullut perustekniikaksi tietoko-

neanimoinnissa. Avainkehysten ideana on luoda animaatiolle alku- ja loppupisteet, joiden väliltä halutut kehykset lasketaan erikseen. Haluttujen kehyksien laskemiseen sulavasti käytetään interpolointia. (Parent 2008.)

Avainkehykset voidaan luoda suoraan kulmapistekohtaisiksi. Kyseistä tekniikkaa käytetään useasti esimerkiksi tietokonepeleissä, mutta tällöin ongelmana tulee muotojen vääristyminen, koska interpolointi pelkkien kulmapisteiden paikkojen välillä ei ota huomioon ollenkaan kolmiulotteisen mallin laajempaa rakennetta. (Puhakka 2008.)



Kuvio 8. Overgrowth-pelin jänishahmo sekä sen luurakenne (Wolfire Games 2009.)

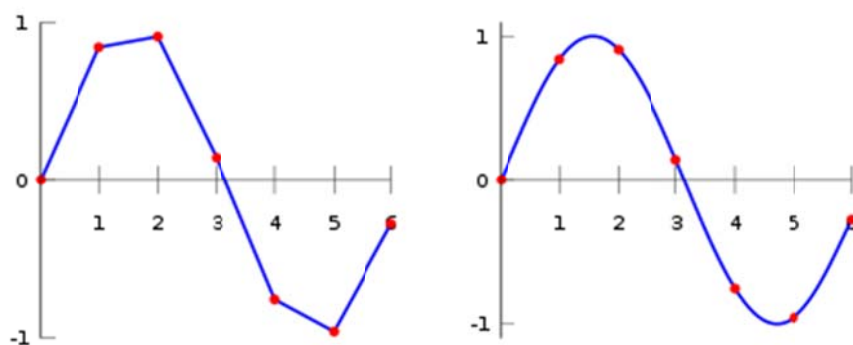
Luurankomallilla sen sijaan saavutetaan huomattavasti parempia tuloksia. Luurankomallissa on ideana luoda kuviossa 8. olevan mallin kaltainen luurankorakenne, joka toimii pääasiassa samalla tavoin kuin ihmisen luusto. Luurakenteen avulla voidaan toteuttaa luonnollisempia animaatioita kuin pelkän kulmapisteinterpoloinnin avulla, sillä kulmapisteet eivät suoraan liiku paikasta toiseen vaan kulkevat määritettyjen luiden mukana. (Puhakka 2008.)

Luurankomallissa jokainen kulmapiste on liitetty yhteen tai useampaan luuhun, ja jokaiselle liitetylle luulle on omat painoarvonsa, joilla määritetään miten paljon kyseisen luun liike vaikuttaa kulmapisteeseen. (Puhakka 2008.)

Luurakenne koostuu nivelistä, joiden arvoja muuttamalla luodaan hahmon asennot. Yksittäisten nivelten liikuttamista kutsutaan etuperoiseksi kinematiikaksi (Forward Kinematics). Käänteiskinematiikalla (Inverse Kinematics) sen sijaan tarkoitetaan tilannetta, jossa lasketaan tiettyjen nivelten arvot niin että ne kohdistuvat johonkin haluttuun pisteeseen. Esimerkiksi

hahmon käden kurkottaminen esinettä kohti saadaan näyttämään luonnollisemmalta huomattavasti helpommin, koska tällöin ei tarvitse toteuttaa erillisiä animaatioita esineiden poimimiselle mallinnusohjelmistossa. (Puhakka 2008.)

Yksi tietokoneilla toteutetun animoinnin tärkeistä perustoista on arvojen interpolointi. Yksinkertainen esimerkki animoinnista on pisteen paikan interpolointi avaruudessa. Tämän tekeminenkään ei ole triviaalia ja tarvitsee muutamien asioiden ottamista huomioon: sopivan interpolointifunktion valinta, funktion parametrisointi kuljetulle matkalle sekä interpoloidun paikan ylläpitäminen ajasta riippumatta. (Parent 2008.)



Kuvio 9. Lineaarinen interpolointi ja splini-interpolointi (Wikipedia 2007.)

Kuviossa 9. nähdään miten lineaarinen interpolointi ja splini-interpolointi eroavat. Lineaarinen interpolointi kulkee suoraan pisteestä pisteelle ja splini-interpoloinnin tarkoituksena on antaa huomattavasti luonnollisempia tuloksia. Linearisesta interpoloinnista käytetään useasti lyhennettä lerp (Linear Interpolation). (Puhakka 2008.)

Kvaternioita käytetään yksinkertaisen interpolointinsa vuoksi useasti asentojen esittämiseen 3D-avaruudessa. Kvaterniot ovat 4-komponenttisia kompleksilukuja, joiden interpolointia varten on olemassa slerp (Spherical Linear Interpolation). Kyseinen interpolointitapa antaa huomattavasti luonnollisempia arvoja kuin lerp. Slerp toimii kulkemalla yksikköpallon pintaa pitkin yhtä sen isoympyrää seuraten. 3D-kierrot on mahdollista esittää myös matriiseina, mutta kvaterniot ovat huomattavasti kompaktimpia, sillä matriisit tarvitsevat 16 komponenttia ja kvaterniot vain 4 komponenttia. (Puhakka 2008.)

2.3 Käyttöliittymät

Käyttäjät ovat vuorovaikutuksessa ohjelman kanssa käyttöliittymän avulla. Käyttöliittymäsuunnittelussa on tarkoituksena tehdä ohjelmasta helppokäyttöinen sekä nopeasti opittava. Ensimmäinen merkittävä suorassa vuorovaikutuksessa toimiva graafinen ohjelma oli Sutherlandin kehittämä Sketchpad. Sketchpad oli graafiseen suunnitteluun tarkoitettu ohjelma, joka toteutettiin vuonna 1963. Yleensä graafisella käyttöliittymällä tarkoitetaan menetelmiä, joissa käytetään vuorovaikutuksen saavuttamiseen grafiikkaa, näppäimistöä ja hiirtä. (AMIA 2007.)

GTK+

GTK+ (GIMP Toolkit) on alun perin GIMP-kuvankäsittelyohjelmalle tarkoitettu käyttöliittymäkirjasto, jolla on mahdollista toteuttaa pieniä ohjelmia sekä isojakin ohjelmistokokonaisuuksia. GTK+ on kirjoitettu C-ohjelmointikielillä, mutta se on toteutettu muita kieliä silmällä pitäen, joten se toimii myös esimerkiksi C++-, Java-, JavaScript- sekä Python-ohjelmointikielillä. GTK+-kirjastoa käyttävät mm. GIMP-kuvankäsittelyohjelma, Pidgin-pikaviestinohjelma sekä Firefox-www-selain. (The GTK+ Team 2012.)

wxWidgets

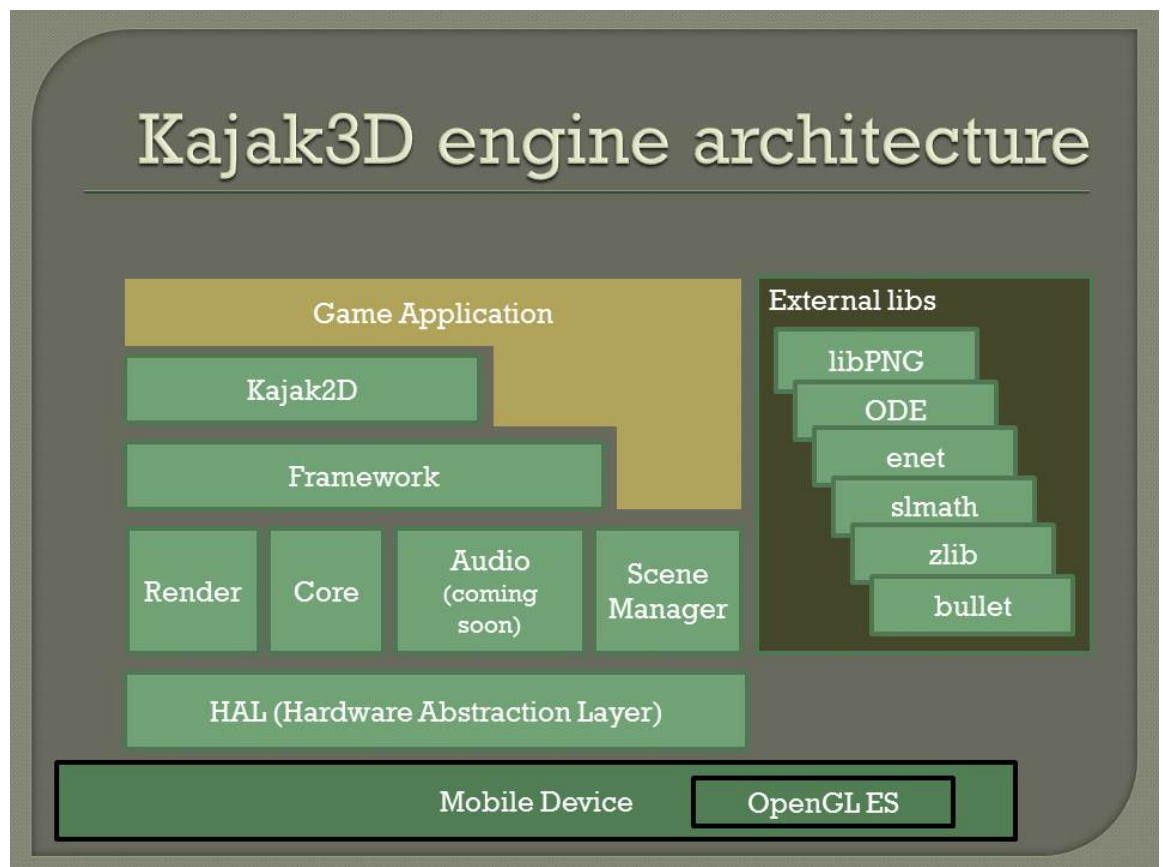
wxWidgets on avoimen lähdekoodin käyttöliittymäkirjasto, jonka kehitys on jatkunut jo parin kymmenen vuoden ajan. wxWidgets-kirjastoa käyttävät mm. Code::Blocks-kehitysympäristö sekä Audacity-äänieditori. wxWidgets-kirjaston tuettuihin alustoihin kuuluvat mm. Windows ja Linux. (wxWidgets 2012.)

2.4 Kirjastot

Tietotekniikassa kirjastolla tarkoitetaan resurssikokoelmaa, jonka avulla toteutetaan ohjelmia. Kyseessä on yleensä valmiiksi kirjoitetusta koodista, jonka tarkoituksena on nopeuttaa ohjelmien toteutusta. (Wikipedia 2012 c.)

2.4.1 Kajak3D

Kajaanin ammattikorkeakoulun kehittämä Kajak3D on sovelluskehitysalusta, joka on suunniteltu toimimaan pääasiassa älypuhelinlaitteilla. Kajak3D toimii älypuhelimien lisäksi myös PC- ja konsoliympäristöissä. Kajak3D-pelimoottorilla tehdään sovelluksia pääasiassa iOS-, Android- ja Windows-alustoille. Suurimpana tarkoituksena Kajak3D-pelimoottorilla on helpottaa sovelluksien tekemistä monelle alustalle ilman suuria muutoksia koodin suhteen. (Kajak3D 2012.)



Kuvio 10. Kajak3D-pelimoottorin arkkitehtuurikaavio (Kajak3D Wiki 2012.)

Kajak3D-pelimoottorin kehitys alkoi vuonna 2008 ja sen arkkitehtuuri (Kuvio 10.) perustui tuolloin Java-spesifikaatioon. Kajaanin ammattikorkeakoulun Kajak3D-pelimoottorin piirto-ominaisuudet hyödyntävät OpenGL- sekä OpenGL ES -rajapintoja. (Kajak3D Wiki 2012.)

Kajak3D-pelimoottori siirtyi syksyllä vuonna 2012 avoimeksi lähdekoodiksi LGPL:n alaisuuteen, mutta yhteistyökumppaneiden on mahdollista saada Kajak3D käyttöönsä myös BSD-lisenssillä. Avoin lähdekoodi mahdollistaa ottamaan vastaan ulkopuolisten ohjelmoijien muu-

tokset sekä laajemman käytön itse pelimoottorille, sillä aikaisemmin Kajak3D ei ollut ulko-
puolisten käytössä. (YouTube 2012.)



Kuvio 11. Ruudunkaappaus Kajak Rally -pelistä (YouTube 2010.)

Kajaanin ammattikorkeakoulun pelipuolen opiskelijat ovat toteuttaneet Kajak3D-pelimoottorilla useita eri peliprojekteja kurssien suorittamista tai erilaisia tutkimusprojekteja varten. Tällaisista projekteja ovat esimerkiksi suurin osa Dibidogs-sarjan peleistä, Kinect Stroke -aivohalvauspotilaiden kuntoutuspeli sekä Kajak Rally (Kuvio 11.). (Kajak3D 2012.)

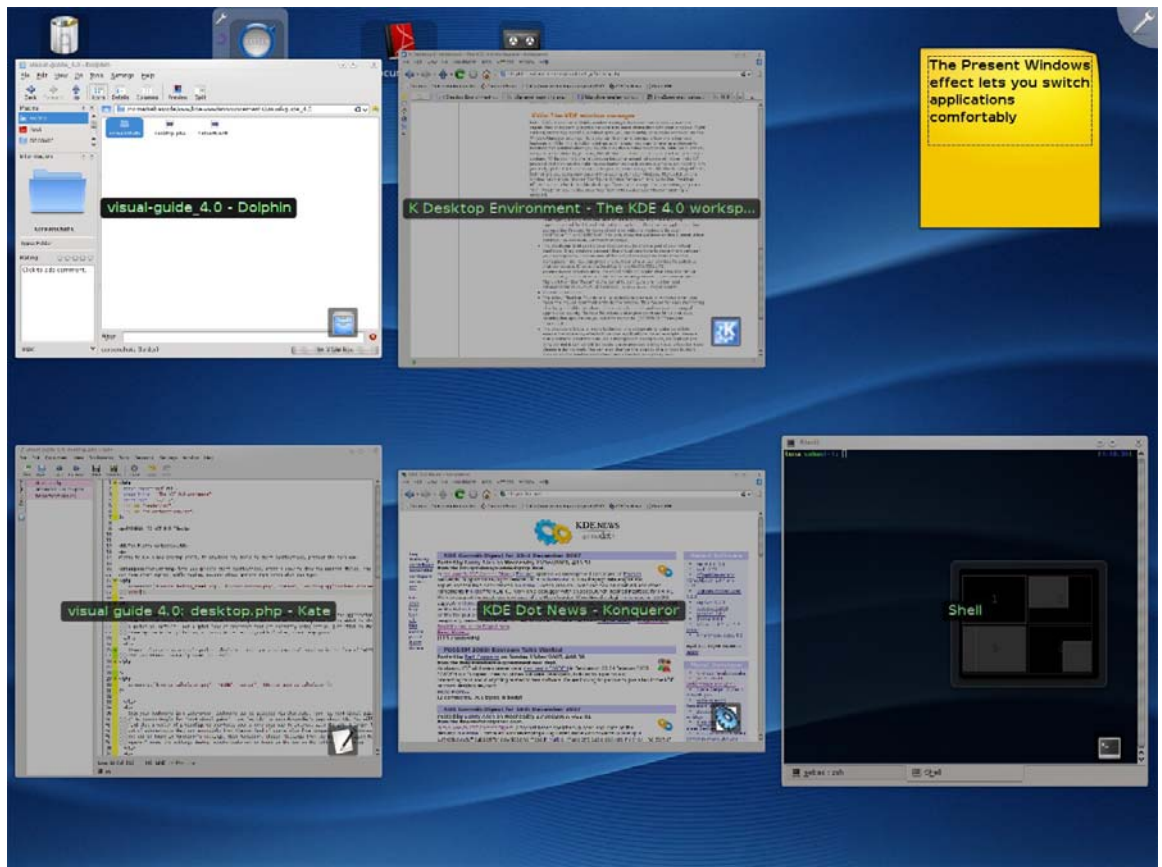
Kajak3D-pelimoottori on suuressa käytössä Kajaanin ammattikorkeakoulun pelialan opetuksessa. Sitä käytetään mm. opiskelualustana pelien toteuttamiseen mobiililaitteille. Esimerkiksi Kajak Rally on kolmen Kajaanin ammattikorkeakoulun pelipuolen opiskelijan toteuttama rallipeli, jota käytettiin tekoälyä ja verkkopeliohjelmointia vaativien kurssien suorittamiseen. (Kajak3D 2012.)

2.4.2 Qt

Qt on sovelluksille ja käyttöliittymille tarkoitettu avoimen lähdekoodin sovelluskehys, joka sisältää alustariippumattoman luokkakirjaston, integroidut kehitystyökalut ja alustariippumat-

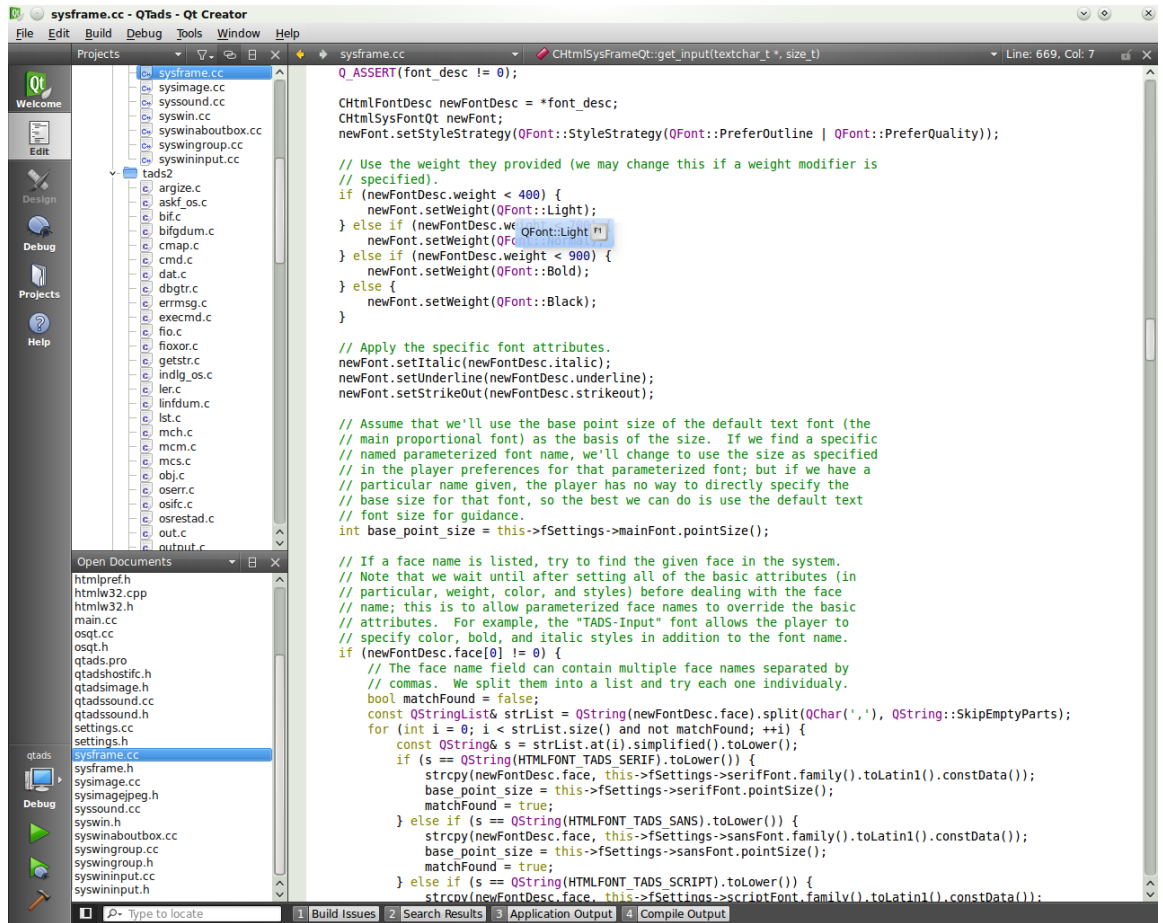
toman kehitysympäristön. Qt:n avulla voidaan toteuttaa sovelluksia, jotka on mahdollista ottaa käyttöön usealla eri alustalla ilman että joutuu tekemään muutoksia sovelluksen lähdekoodiin. Yli 450 000 kehittäjää yli 70 teollisuusalalla käyttää Qt:ta toteuttaakseen kehittyneitä sovelluksia ja laitteita. (Qt 2012.)

Qt:ta on kehitetty jo vuodesta 1992 lähtien. Qt on kirjoitettu C++-ohjelmointikielellä, mutta Qt:ta on mahdollista käyttää myös useilla muilla eri ohjelmointikielillä niiden omien rajapintatoteutuksien avulla. (Qt 2012.)



Kuvio 12. KDE-työpöytäympäristö. (Qt 2012)

Qt:ta käytäviä ohjelmistoja ovat mm. KDE-työpöytäympäristö (Kuvio 12.), Skype-pikaviestin sekä VLC-mediasoitin. (Qt 2012.)



Kuvio 13. Ruudunkaappaus Qt Creator -ohjelman käyttöliittymästä (Wikipedia 2012 d.)

Qt Creator on alustariippumaton kehitysympäristö, joka on tarkoitettu Qt:ta käyttävien ohjelmien toteuttamiseen C++-ohjelmointikielillä. Qt Creator ei kuitenkaan rajoita käyttämään pelkästään Qt:ta, joten se sopii myös muidenkin C++-pohjaisten projektien toteuttamiseen. Kuviossa 13. on Qt Creator -kehitysympäristön ohjelmointinäkömä. (Qt 2012.)

2.5 Tiedostomuodot

STL

STL (Standard Tessellation Language) on yksinkertaisuutensa vuoksi yleinen tiedostomuoto monissa CAD-mallinnusohjelmistoissa. STL-tiedostomuodosta on olemassa ASCII- ja binäärimuotoiset versiot. Binäärimuotoista versiota käytetään yleensä tiedostoformaateissa pienen koon takia, sillä tekstimuodossa olevien tiedostojen koot kasvavat nopeasti todella suuriksi. Binäärimuotoisia tiedostoja käytetään pääasiassa silloin, kun ihmisen ei tarvitse kyetä lukemaan tiedostoa suoraan, vaan sen lukemisen suorittaa tietokone. STL-tiedostomuodossa oleva geometria koostuu pelkistä kolmioista, joten kolmiot eivät voi käyttää samoja kärkipistekoordinaatteja. Tästä syystä johtuen niillä ei voi olla kärkipistekohtaisia normaalikoordinaattejakaan, vaan ne on tallennettu suoraan kolmioihin. (Burkardt 2001.)

XML

XML (Extensible Markup Language) on merkintäkieli, jolla määritetään helposti luettavia dokumentteja sekä tietokoneille että ihmisille. XHTML on yksiesimerkki XML-merkintäkielen käyttötavoista, mutta sitä ei kuitenkaan ole tarkoitettu korvaajaksi HTML-merkintäkielelle. (Refsnes Data 2012.)

3 YLEISESTI KÄYTTYJÄ ROBOTTISIMULOINTIOHJELMISTOJA

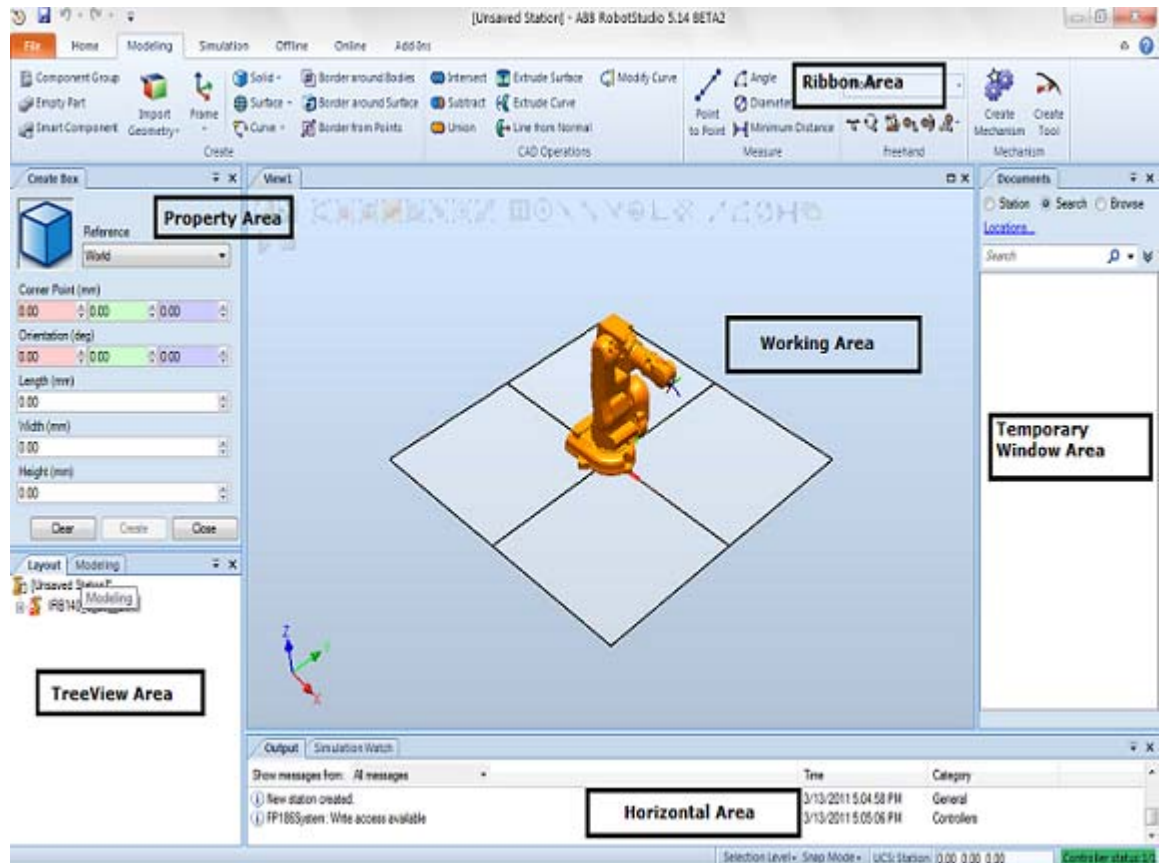
On olemassa useita valmiita robottien simulointiin tarkoitettuja ohjelmia, joiden avulla voidaan toteuttaa roboteilla suoritettavia ohjelmia. Useimmat simulointiohjelmistot ovat isojen yritysten toteuttamia ohjelmistoja, mutta myös monet käyttäjäyhteisöt ovat kehittäneet omia vapaan lähdekoodin toteutuksia robottien simulointia varten.

3.1 RobotStudio

RobotStudio on ABB:n kehittämä simulointi- ja etäohjelmointityökalu, joka mahdollistaa robottien ohjelmoinnin tietokoneella. RobotStudion avulla ohjelmat voidaan toteuttaa toimituksessa ilman että varsinaisia robotin työtehtäviä keskeytetään. RobotStudio toimii perinteisillä tietokoneilla ja sisältää Visual Basic for Applications -tuen, jonka avulla mahdollistetaan monipuolisten toimintojen toteuttaminen. (ABB 2012 a.)

RobotStudion tarkoituksina ovat mm. riskien vähentäminen sekä tuotettavuuden parantaminen. Tarkkuuden parantamista varten RobotStudio mahdollistaa esimerkiksi CAD-mallinnusohjelmilla toteutettujen yleisimpien tiedostomuotojen tuomisen ohjelmaan simulointia varten. Riskien vähentäminen tapahtuu visualisoimalla ja hyväksymällä ratkaisut ja suunnitelmat. Tuotettavuutta pyritään parantamaan optimointimahdollisuuksien avulla. (ABB 2012 a.)

RobotStudio on rakennettu VirtualController-ohjelmiston avulla, joka on tarkka kopio oikeassa tuotannossa käytettävän robotin ohjelmistosta. Tämän ansiosta RobotStudio mahdollistaa todella realistisia simulaatioita. (ABB 2012 a.)



Kuvio 14. RobotStudion käyttöliittymä (ABB 2012 a.)

Kuviossa 14. on RobotStudion pääikkuna, jossa on näkyvissä työskentelynäköm, mallinnustyökalujen painikkeet sekä muut robottien ohjelmointia helpottavat toiminnot. (ABB 2012 a.)

3.2 Microsoft Robotics Developer Studio

Robotics Developer Studio (RDS) on Microsoftin kehittämä robottien ohjelmien toteuttamiseen tarkoitettu simulointiohjelmisto. RDS on Windows-ympäristössä toimiva harrastajille, opiskelijoille sekä kaupalliseen käyttöön tarkoitettu ympäristö, jolla on mahdollista luoda ohjelmia useille eri laitealustoille. Uuden version myötä RDS tukee myös Kinect-sensoreita, joita voidaan myös simuloida oikeiden laitteiden sijasta. (MSDN 2012.)



Kuvio 15. Robotics Developer Studion visualisointinäkymä (MSDN 2012.)

RDS-ohjelmiston ominaisuuksiin kuuluu myös tuki usean eri robotin simulointiin samaan aikaan (Kuvio 15.). RDS mahdollistaa ohjelmien ohjelmoinnin C#-kielen lisäksi VPL:n (Visual Programming Language) avulla. (MSDN 2012.)

4 PROJEKTI

Käytännönsuutena tässä opinnäytetyössä oli toteuttaa yksinkertainen simulointiohjelma raepuhallusrobotteja varten. Simulointiohjelman päävaatimuksina oli ladata käytettävän robotin tiedot, antaa sille tilapisteitä ja lopuksi tulostaa toteutettu tilapisteistä koostettu ohjelma raepuhallusrobotille sopivaan tiedostomuotoon. Lopullisen tiedoston suorittavan raepuhallusrobotin tulisi käyttäytyä oikeassa elämässä samalla tavoin kuin se on suunniteltu käyttäytymään simulointiohjelmassa olevassa näkymässä.

4.1 Simulointi

Graafinen simulointi helpottaa robottien työtehtävien suunnittelussa. Simuloitaessa saadaan selville tärkeitä asioita robotin työtehtävän tapahtumista. Esimerkiksi törmäystarkastelulla selvitetään onko tapahtumassa mahdollisia törmäyksiä robotin ja työkappaleen välillä, ja täten vältetään mahdolliset robotin aiheuttamat vauriot.

Koska simuloitaessa tarvitaan ainoastaan tiedot oikean robotin toiminnasta, ei oikealle robotille ole tarvetta ohjelmoitaessa. Simulointi helpottaa siis uuden työstökappaleen tuontia tuotantoon, sillä tuotanto ei keskeydy kun robotille ohjelmoidaan etänä uudet toimenpiteet. Esimerkiksi samalle työstettävissä olevalle kappaleelle voidaan toteuttaa uudet työtehtävät ilman että varsinaista robottia joudutaan pysäyttämään.

Simuloinnissa on myös haittapuolia, sillä simulointi tekee ainoastaan sen mitä on käsketty tekemään. On mahdollista että ohjelman toteutusvaiheessa saattaa unohtua esimerkiksi asettaa kitkan arvot tai painovoima, tai ne arvot voivat olla muuten vain väärät. Ohjelma ei toimikaan tämän jälkeen odotetulla tavalla oikeassa elämässä. Yleensä simulointiohjelmissä käytetyt robotit ovat kuitenkin mahdollisimman yksinkertaisia, esimerkiksi kitkat poistettu, ei viivettä syötteessä ja kaikki objektit ovat pyritti toteuttamaan mahdollisimman pienellä geometrialla.

4.2 Vaatimukset

Simulointiohjelman päävaatimukset ovat suhteellisen yksinkertaiset. Ohjelmalta vaaditaan vähintään seuraavan listan määrittelemät asiat.

1. Mallin lataus tiedostosta
2. Mallin esittäminen ruudulla
3. Ohjelman lataaminen tiedostosta
4. Ohjelman tallentaminen tiedostoon
5. Akseleiden arvojen vaihtaminen

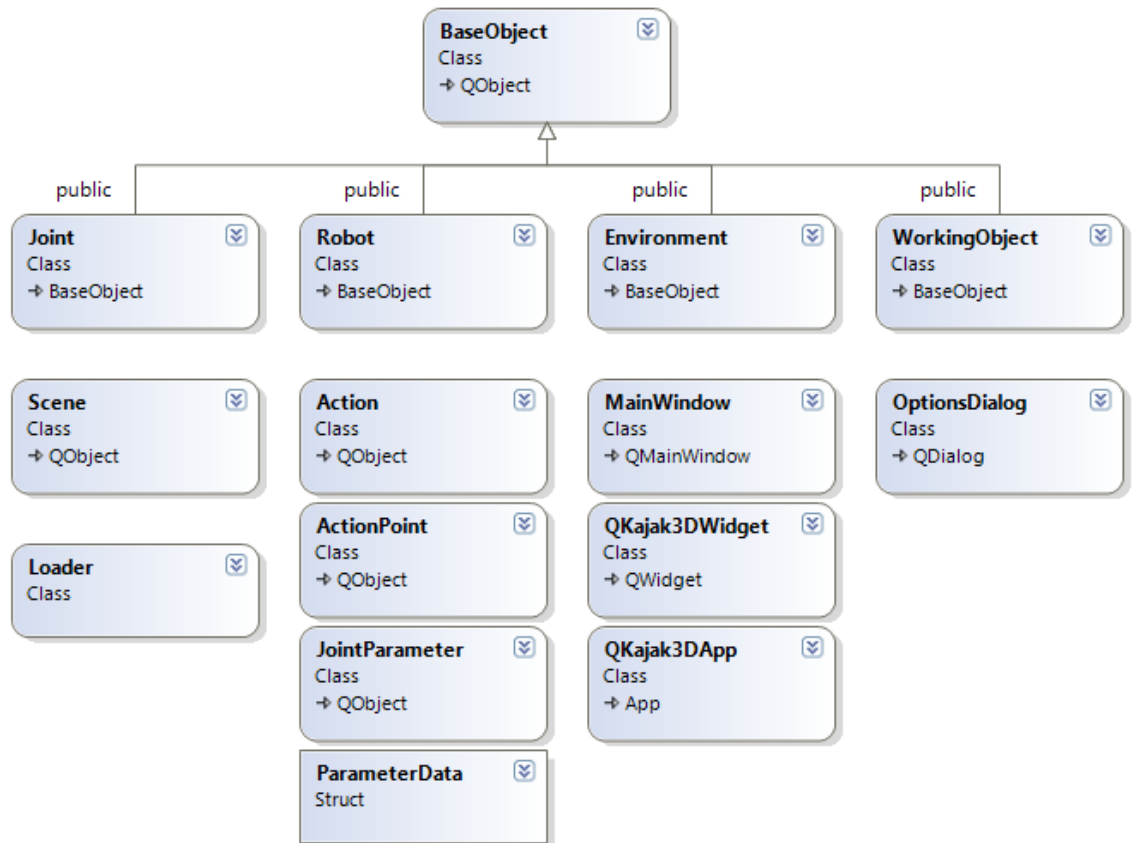
4.3 Suunnittelu

Projektin suunnittelu aloitettiin muutamaa viikkoa aikaisemmin itse varsinaisen toteutuksen aloittamisesta. On vaikeaa tietää kaikkea toiminnallisuutta käytettävistä roboteista etukäteen, joten suunnittelua jatkettiin myös toteutuksen ja testauksen aikana tulleiden havaintojen vuoksi. Yksi esimerkki testauksen aikana tulleesta havainnoinnista virheellisen toiminnallisuuden suhteen on puhaltimen tilan vaihtaminen. Puhallin vaihtaa tilaansa vasta seuraavassa pisteessä, eikä otakaan puhallusasetusta käyttöön heti nykyisestä jo saavutetusta pisteestä. Kuten muussakin ohjelmistosuunnittelussa, simulointiohjelmalla toteutettuja ohjelmia olisi hyvä siis testata mahdollisimman useasti ja monipuolisesti, jotta tämän tyylistä ohjelmistovirheistä päästäisiin eroon mahdollisimman nopeasti heti alkuvaiheissa.

4.4 Toteutus

Projektissa käytössä olevat tiedostomuodot ovat STL- ja XML-muodoissa. STL-tiedostomuoto valittiin mallien latausta varten, sillä se on tuettuna useissa eri CAD-mallinnusohjelmistoissa. Objektien ja animaatioiden tallentamista varten kehitettiin oma XML-pohjainen tiedostomuoto.

Projektissa käytettiin C++-ohjelmointikieltä ja kehitysympäristönä Qt Creatoria. Käyttöliittymät toteutettiin Qt:n avulla ja kolmiulotteisen näkymän esittämiseen käytettiin Kajak3D-pelimoottorin piirto-ominaisuuksia.



Kuvio 16. Simulointiohjelman luokkakaavio

Kuviossa 16. näkyvästä luokkakaaviosta näkyy projektin pääluokat ilman periytyksiä Qt:n luokista.

Action

Action-luokka pitää sisällään robotin komennot. Action-luokan kautta tapahtuu mm. akselien arvojen asetukset sekä puhallustilan vaihtaminen.

BaseObject

BaseObject on kantaluokka, josta kaikki näkyvät objektit peritään. BaseObject-luokan kautta tapahtuu mm. paikan, asennon sekä näkyvyyden asettamiset.

Environment

Environment on ympäristölle tarkoitettu luokka. Environment-luokka periytyy BaseObject-luokasta.

Joint

Joint-luokka esittää yhtä robotin niveltä. Joint-luokan kautta tapahtuu parametrien asettaminen nivelelle. Joint-luokka periytyy BaseObject-luokasta.

Robot

Robot-luokan tarkoituksena on pitää yllä robotin asetuksia sekä Joint-luokasta koottua nivel-listaa. Robot-luokka periytyy BaseObject-luokasta.

Scene

Scene-luokka huolehtii näkymässä olevien objektien päivittämisestä ja animoinnista. Scene-luokan kautta tapahtuu tilapisteen vaihtaminen kaikkien objektien hallinta.

WorkingObject

WorkingObject on BaseObject-luokasta peritty objekti, joka huolehtii työstökappaleen esittämisen.

JointParameter

JointParameter-luokka ylläpitää nivelen asetuksia.

Loader

Loader on luokka joka hoitaa tarvittavien tiedostojen lataukset ja tallennukset.

MainWindow

MainWindow-luokka on Qt:lla toteutettu ohjelman pääikkuna, jonka tarkoituksena on ylläpitää ohjelman ulkoasua.

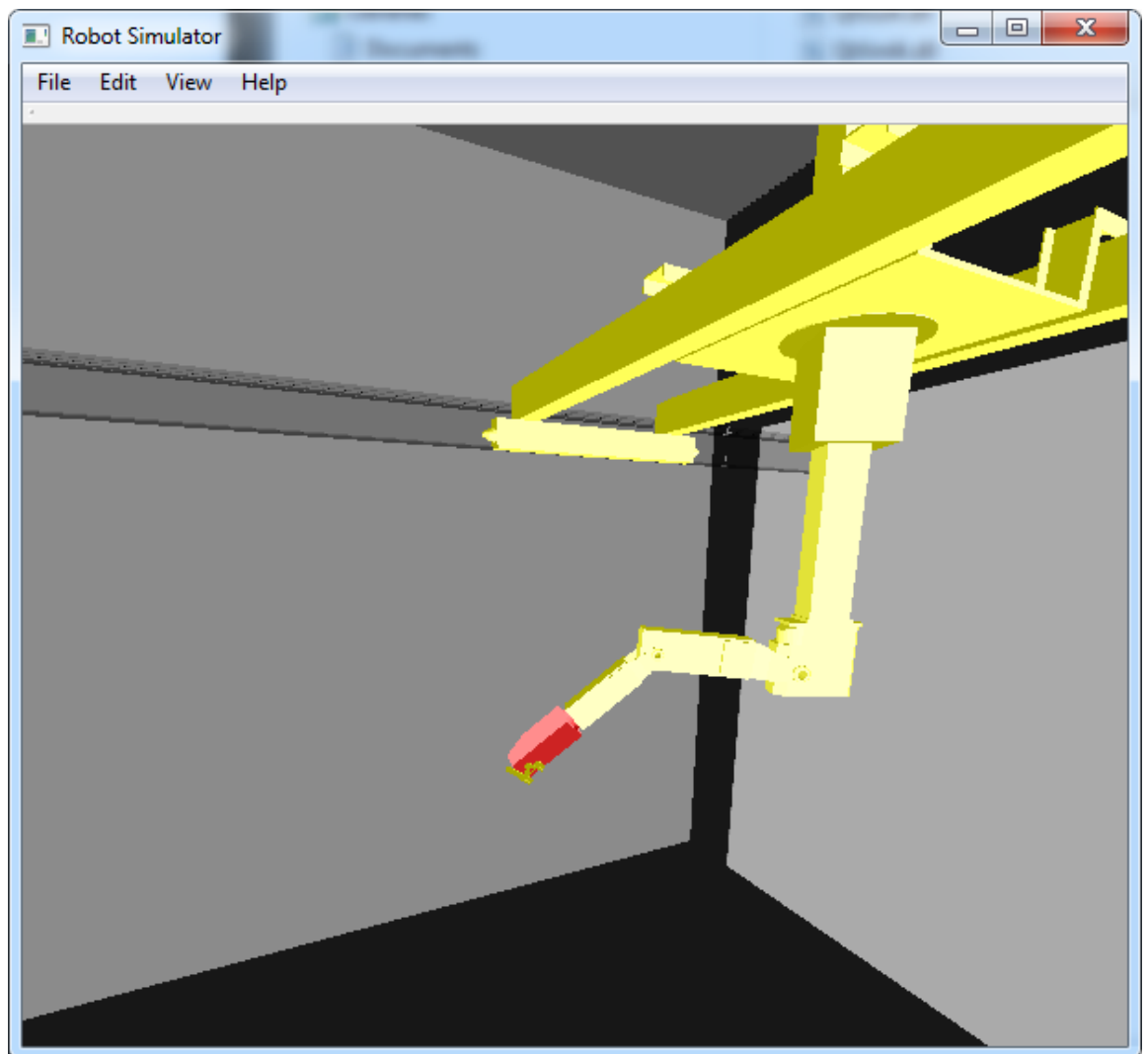
QKajak3DApp

QKajak3DApp-luokka hoitaa tarvittavien tietojen välittämisen Kajak3D-pelimoottorille.

QKajak3DWidget

QKajak3DWidget on käyttöliittymäkomponentti, joka hoitaa QKajak3DApp-luokan liittämisen Qt:lla luotuun ikkunaan.

4.5 Näkymän rakenne



Kuvio 17. Robotti simulointiohjelman päänäkymässä

Simulointiympäristön näkymä (Kuvio 17.) koostuu pääasiassa kolmesta eri objektista: työympäristöstä, työkappaleesta ja robotista. Objekteilla on perusominaisuuksina paikka, asento, skaalaus sekä näytettävä 3d-malli.

Robotti eroaa muista objekteista enemmän, sillä sen malli koostuu useista eri palasista. Robotille asetetaan myös työstöpisteet ja sen eri osat animoidaan kyseisten työstöpisteiden avulla.

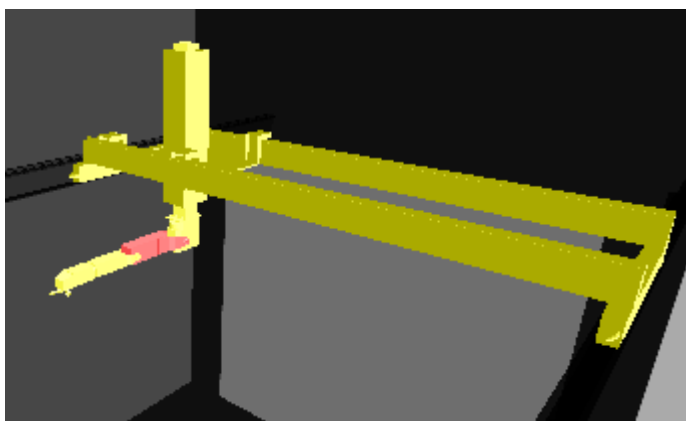
4.6 Testaus

Testausta tehtiin jatkuvasti ohjelman kehittyessä. Testausta toteutettiin myös itse robotin parissa, jotta robotin toiminnallisuutta ymmärtäisi helpommin. Testauksesta oli huomattavasti hyötyä, sillä esimerkiksi nivelten liikkumis- ja pyörimissuunnat eivät olleet helposti ennalta arvattavissa.

Toteutimme oikealla robotilla muutamia ohjelmia, jonka jälkeen tehtyjä ohjelmia verrattiin simulointiympäristön näkymään. Robotilla tehdyt ohjelmat helpottivat merkittävästi simulointiohjelman toteuttamista, sillä ongelmien kohdatessa viat olivat helpommin havaittavissa. Ongelmat saattoivat ilmetä esimerkiksi virheellisenä nivelen suuntana, joka voi johtua monesta eri asiasta. Esimerkiksi robotin asetukset tai laskuvirheet saattavat aiheuttaa väärän suunnan nivelen liikkeelle.

Testitapaus 1

Testataan ohjelman latausta tiedostosta.

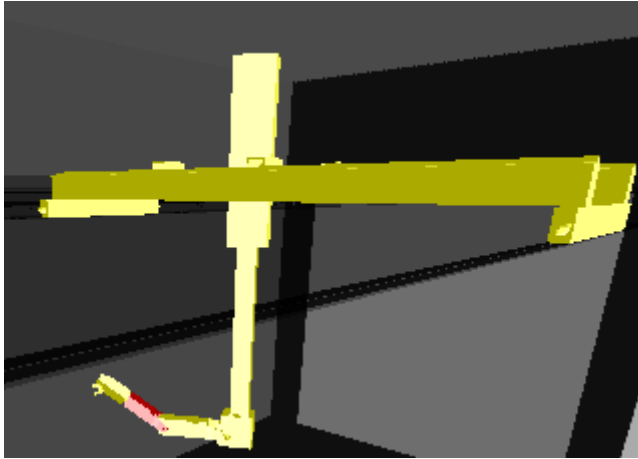


Kuvio 18. Robotti oletusasennossa

Testitapauksen perusteella havaitaan että ohjelman lataus onnistuu ja ympäristö näkyy ruudulla. Kuviossa 18. näkyy testitapauksen saavuttama robotin oletustila.

Testitapaus 2

Testataan robotin liikeakseleita antamalla sille muutamia liikkeitä sekä asetetaan ne omiin tilapisteisiin.



Kuvio 19. Robotti uudessa tilapisteessä

Testitapauksen perusteella havaitaan että robotin akselit ovat liikutettavissa, sekä tilapisteiden tallentaminen onnistuu. Kuviossa 19. näkyy robotin muutettu asento.

Testitapaus 3

Lisätään ohjelmaan muutamia tilapisteitä ja testataan sen tallentamista tiedostoon.

Testin perusteella havaitaan että ohjelma toimii halutulla tavalla. Tämä ohjelman osa-alue vaatii kuitenkin vielä lisää kehitystä ja testausta, sillä koodin rakenne tältä osin ei ole riittävän hyvä ja siellä saattaa olla vielä joitakin virheitä. Mahdollinen koodin uudelleen kirjoitus on tarpeen.

4.7 Tulosten vertailu vaatimuksiin

Vertailukuvion avulla käy ilmi mitkä testitapaukset vastaavat mitäkin vaatimusta.

| | Vaatus 1 | Vaatus 2 | Vaatus 3 | Vaatus 4 | Vaatus 5 |
|---------------|----------|----------|----------|----------|----------|
| Testitapaus 1 | X | X | X | | |
| Testitapaus 2 | | | | | X |
| Testitapaus 3 | | | | (X) | |

Kuvio 20. Vaatimuksia vastaavat testaustapaukset

Kuviossa 20. nähdään mitkä testitapaukset testaavat eri vaatimuksia. Kaikki vaatimukset täyttyivät, mutta kolmannen testitapauksen kohdalla on parannettavaa, sillä testi ei läpäissyt täydellisesti.

5 POHDINTA

Robotiikkaa esiintyy Suomessa vielä nykyään melko vähän, mutta toivottavasti tulevaisuudessa robottien markkinat kasvavat. Aiheena robotiikka on kiinnostava, ja varsinkin robotiikassa käytettävät tekniikat ovat mielenkiintoisia.

Tarvittavat ohjelmat ja käyttöoikeudet ohjelmien kehittämistä varten ovat erittäin tärkeitä. Työskentelin kesän ajan pääasiassa koulun pelilaboratoriossa, jossa nykyään ei ole opiskelijoilla pääkäyttäjän oikeuksia tietoturvaongelmien takia. Jotkin ohjelmista olivat myös vanhentuneita, mikä häiritsi huomattavasti työskentelyä. Kotona tehdyt muutokset eivät välttämättä toimineetkaan koulussa olevilla vanhemmilla ohjelmien versioilla.

Projektiin olisi myös ollut hyvä saada vähintään yksi ohjelmoija lisää, jotta erilaisten asioiden suunnittelusta ja toteutuksesta olisi voinut keskustella paremmin. Jatkuvassa keskustelussa ajatuksetkin pysyisivät luultavasti paremmin tietyssä ohjelman osa-alueessa, tai ainakin ajatukset voisivat olla huomattavasti selkeämmät. Aihe on muutenkin laaja ja erilaiset ohjelman osa-alueet olisi voinut jakaa helposti muutamien ohjelmoijien kesken.

Projekti onnistui mielestäni kohtuullisen hyvin. Varsinkin oppimisen kannalta projekti toi paljon uutta tietoa käyttöliittymien toteuttamisen suhteen. Vaikka ohjelmassa on vielä paljon puutteita, toimii se yksinkertaisuutensa vuoksi tällä hetkellä tulevaisuutta ajatellen hyvänä koodikantana jatkokehitystä varten.

LÄHTEET

ABB 2012 a. Saatavilla:

<http://www.abb.com/product/seitp327/78fb236cae7e605dc1256f1e002a892c.aspx>.

(Luettu 2.5.2012).

ABB 2012 b. RAPID Reference Manual. Saatavilla:

<http://rab.ict.pwr.wroc.pl/irb1400/overviewrev1.pdf> (Luettu 2.5.2012).

Akenine-Möller, T., Haines, E. & Hoffman, N. 2008. Real-Time Rendering, Third Edition.
A K Peters, Ltd.: Wellesley, Massachusetts.

AMIA 2007. Saatavilla: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655855/> (Luettu 2.5.2012).

Blastman Robotics Oy 2011. Keskustelut vierailujen aikana Renforsin Rannassa.

Blastman Robotics Oy 2012. Saatavilla: http://www.blastman.com/sivu/fi/kestavaa_laatua/
(Luettu 2.5.2012).

Burkardt, J. 2012. Saatavilla: <http://orion.math.iastate.edu/burkardt/data/stl/stl.html> (Luettu 2.5.2012).

Kajak3D 2012. Saatavilla: <http://kajak3d.com> (Luettu 2.5.2012).

Kajak3D Wiki 2012. Kehittäjien sisäinen wiki. (Luettu 2.5.2012).

Lighthouse 3D 2012 a. Saatavilla: <http://www.lighthouse3d.com/tutorials/glsl-tutorial/pipeline-overview/> (Luettu 2.5.2012).

Lighthouse 3D 2012 b. Saatavilla: <http://www.lighthouse3d.com/tutorials/glsl-tutorial/toon-shading/> (Luettu 2.5.2012).

Microsoft Robotics 2012. Saatavilla: <http://www.microsoft.com/robotics/> (Luettu 2.5.2012).

NVIDIA 2010. Saatavilla: http://www.nvidia.com/object/feature_vertexshader.html (Luettu 2.5.2012).

- NVIDIA 2012. Saatavilla: <http://www.nvidia.com/object/tessellation.html> (Luettu 2.5.2012).
- NVIDIA Developer Zone 2007. Saatavilla: http://developer.download.nvidia.com/opengl/specs/GL_EXT_geometry_shader4.txt. (Luettu 2.5.2012).
- OpenGL Wiki. 2010. Saatavilla: http://www.opengl.org/wiki/Geometry_Shader. (Luettu 2.5.2012).
- Oxford Dictionaries 2012. Saatavilla: <http://oxforddictionaries.com/definition/robotics>. (Luettu 2.5.2012).
- Parent, R. 2008. Computer Animation: Algorithms and Techniques. Burlington: Morgan Kaufman Publishers.
- Pikkarainen, E., Räsänen, S., Pikkarainen L. & Heikkinen T. 2012. Opetusmateriaali. Kajaanin ammattikorkeakoulu.
- Puhakka, A. 2008. 3D-grafiikka. Helsinki: Talentum Media.
- Qt 2012. Saatavilla: <http://qt.nokia.com/>. (Luettu 2.5.2012).
- RAPID Reference Manual 2012. Saatavilla: <http://rab.ict.pwr.wroc.pl/irb1400/overviewrev1.pdf>. (Luettu 2.5.2012).
- Refsnes Data 2012. Saatavilla: <http://www.w3schools.com/xml/>. (Luettu 2.5.2012).
- Suomen Robotiikkayhdistys Ry 2012. Saatavilla: http://www.robayhd.fi/index.php?option=com_docman&gid=55. (Luettu 2.5.2012).
- The GTK+ Team. 2012. Saatavilla: <http://www.gtk.org/>. (Luettu 2.5.2012).
- TOPIO 2009. Saatavilla: <http://topio.tosy.com>. (Luettu 2.5.2012).
- Unigine. 2009. Saatavilla: <http://unigine.com/press-releases/091001-directx11/>. (Luettu 2.3.2012).
- Wikipedia 2007. Saatavilla: <http://en.wikipedia.org/wiki/Interpolation>. (Luettu 2.5.2012).

Wikipedia 2012 a. Saatavilla: <http://fi.wikipedia.org/wiki/Teollisuusrobotti>. (Luettu 2.5.2012).

Wikipedia 2012 b. Saatavilla: <http://fi.wikipedia.org/wiki/Androidi>. (Luettu 2.5.2012).

Wikipedia 2012 c. Saatavilla: [http://en.wikipedia.org/wiki/Library_\(computing\)](http://en.wikipedia.org/wiki/Library_(computing)). (Luettu 2.5.2012).

Wikipedia 2012 d. Saatavilla: <http://en.wikipedia.org/wiki/File:QtCreator-Linux-2.0.1.png>. (Luettu 2.5.2012).

Windows Dev Center 2012. Saatavilla: [http://msdn.microsoft.com/en-us/library/windows/desktop/bb509609\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb509609(v=vs.85).aspx). (Luettu 2.5.2012).

Wolfire Games 2009. Saatavilla: <http://blog.wolfire.com/2009/11/volumetric-heat-diffusion-skinning/>. (Luettu 2.5.2012).

wxWidgets 2012. <http://www.wxwidgets.org/>. (Luettu 2.5.2012).

YouTube 2010. Kajak Rally. Saatavilla: http://www.youtube.com/watch?v=wrxv-I_0sYA. (Katsottu 2.5.2012).

YouTube 2012. Kajak3D, an open source multiplatform game engine. Saatavilla: <http://www.youtube.com/watch?v=aDQ61RorSDo>. (Katsottu 2.5.2012).

