LAPPEENRANNAN TEKNILLINEN YLIOPISTO
TEKNILLISTALOUDELLINEN TIEDEKUNTA
TIETOTEKNIIKAN OSASTO

Interaktiivinen NURBS-sovelma opetuskäyttöön

Kandidaatintyön aihe on hyväksytty 29.1.2009
Työn ohjaajana ja tarkastajana toimii lehtori TkT Arto Kaarna

Luumäellä 18.4.2009

Jori Niemi
Junttolantie 56
54530 Luumäki
Puh. +358 40 837 6543

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tietotekniikan osasto

Jori Niemi

Interaktiivinen NURBS-sovelma opetuskäyttöön

Kandidaatintyö

2009

34 sivua, 4 kuvaa, 2 taulukkoa ja 1 liite.

Tarkastaja: TkT Arto Kaarna

Hakusanat: NURBS, Java, JOGL, sovelma, interaktiivinen

Parametriset NURBS-käyrät ja -pinnat ovat yleisiä työkaluja graafisessa mallinnuksessa, tietokoneavusteisessa suunnittelussa ja muissa sovelluksissa. NURBS-mallinnus on pääosin melko intuitiivista, mutta siinä on joitakin vaikeammin ymmärrettäviä osia. Ihminen oppii parhaiten tekemällä, minkä takia päätettiin tehdä interaktiivinen NURBS-opetussovelma, joka tutustuttaa käyttäjän NURBSien toimintaan. Sovelma toimii www-selaimessa Java-liitännäisen avulla ja hyödyntää Java OpenGL -grafiikkarajapintaa kolmiulotteisen grafiikan piirtämiseen. Sovelma on helppokäyttöinen ja opastaa käyttäjää NURBS-mallinnuksen eri osa-alueissa.

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Department of Information Technology

Jori Niemi

Interactive NURBS Tutorial Applet

Thesis for the Degree of Bachelor of Science in Technology

2009

34 pages, 4 figures, 2 tables and 1 appendix.

Examiner: D.Sc. (Tech.) Arto Kaarna

Keywords: NURBS, Java, JOGL, applet, interactive

Parametric NURBS curves and surfaces are widely used tools in graphical and computer aided design and other applications. NURBS modeling is relatively intuitive but has some aspects that are more difficult to understand. Humans learn by doing, which is why it was decided that an interactive NURBS tutorial should be implemented. The tutorial introduces different aspects of NURBS to the user. The tutorial is an applet that runs in a Web browser with a Java plugin. The applet uses Java OpenGL library for drawing three-dimensional graphics. The applet is easy to use and guides the user through different aspects of NURBS modeling.

ALKUSANAT

Tämä työ tehtiin Lappeenrannan teknilliselle yliopistolle keväällä 2009. Haluan kiittää ohjaaja Arto Kaarnaa avusta ja ideoista ja opiskelija Antti Karjalaista avusta LaTeX-pohjan kanssa. Työ oli opettavainen ja tarjosi sopivasti monenlaisia haasteita.

Jori Niemi

SISÄLLYSLUETTELO

1 JOHDANTO **3**
1.1 Tausta . . . 3
1.2 Tavoitteet ja rajaukset . . . 3
1.3 Työn rakenne . . . 4

2 TEORIAOSUUS **5**
2.1 Splinit . . . 5
2.1.1 NURBS . . . 5
2.1.2 NURBSien ominaisuuksia . . . 7
2.2 Java . . . 8
2.3 OpenGL . . . 9
2.4 Tutkimusongelman ja sen ratkaisutavan esittely . . . 10

3 KÄYTÄNNÖN OSUUS **11**
3.1 Käytännön toteutus . . . 11
3.1.1 Suunnittelu . . . 11
3.1.2 NURBS-toteutus . . . 11
3.1.3 Eclipse-ohjelmointiympäristö . . . 12
3.1.4 Java ja Java OpenGL . . . 13
3.2 Tulokset . . . 14
3.2.1 Aihealueet . . . 14
3.2.2 Käyttöliittymä . . . 16
3.2.3 Luokkarakenne . . . 19
3.2.4 Valmiit mallit . . . 21
3.2.5 Testatut tietokoneet . . . 22
3.3 Tulosten yhteenveto ja tulkinta . . . 23

4 POHDINTA **24**

5 YHTEENVETO **26**

LÄHDELUETTELO **27**

Liite 1. Vaatimusmäärittely

LYHENTEET

`NURBS`	Non-Uniform Rational B-Spline
`WWW`	World Wide Web
`JOGL`	Java OpenGL
`JDK`	Java Development kit
`GCJ`	GNU Compiler for Java
`JNI`	Java Native Interface

1 JOHDANTO

1.1 Tausta

Paloittain määritellyt polynomifunktiot, splinit, ovat keskeinen väline graafisessa mallinnuksessa, kuten tietokoneavusteisessa suunnittelussa ja grafiikkasovelluksissa [1, 2]. NURBS (Non-Uniform Rational B-Spline) on Bezier-käyrien ja B-splinien yleistys. NURBS-malleilla voidaan esittää sekä käyriä että pintoja, ja yksi niiden tärkeimmistä ominaisuuksista on se, että ne mahdollistavat sekä analyyttisten että vapaamuotoisten kappaleiden mallintamisen yhdellä menetelmällä.

Tietokonegrafiikan peruskurssilla käsitellään NURBSeja, mutta niiden toiminnan ymmärtäminen on osoittautunut haastavaksi. Ihminen oppii hyvin tekemällä. Tämän vuoksi päätettiin tehdä sovellus, joka auttaa käyttäjää ymmärtämään NURBSien toimintaa ja päästää käyttäjän kokeilemaan NURBSeja käytännössä.

1.2 Tavoitteet ja rajaukset

Tämän työn tavoitteena on luoda NURBS-sovellus tukemaan Lappeenrannan teknillisen yliopiston Introduction to Computer Graphics -kurssin (ennen Tietokonegrafiikan perusteet) splinejä käsittelevää osaa. Koska kurssi on englanninkielinen, myös sovelluksen käyttöliittymä tulee olemaan englanniksi.

Sovelluksen tulee olla käytettävissä helposti ja ilman monimutkaisia asennustoimenpiteitä, minkä lisäksi sen on tarkoitus toimia sekä Linuxissa että Windowsissa. Tämä onnistuu parhaiten kun siitä tehdään www-selaimessa toimiva sovelma. Työssä päädyttiin käyttämään Java-alustaa selainpohjaisuuden toteuttamiseen.

Sovelma opastaa käyttäjää miten NURBS-mallin parametrit vaikuttavat lopulliseen käyrään tai pintaan. Lisäksi käyttäjä pääsee vaikuttamaan malliin interaktiivisesti. Käyttöliittymän tulee olla visuaalinen ja havainnollinen.

Olennainen osa ihmiskeskeistä suunnittelua on saada monimutkaiselta vaikuttava työkalu ymmärrettäväksi ja käytettäväksi työkaluksi, joka toimii siinä tehtävässä johon se on suunniteltu [3]. Tämän sovelman käyttötarkoitus on rajattu tiukasti NURBSien perusolemuksen havainnollistamiseen, eli sovelmasta ei olla tekemässä täysipainoista mallintamistyökalua. Tällä tavoin työmäärä saadaan pidettyä kohtuullisena ja käyttöliittymä yksinkertaisena. Yksinkertaista muokkausta edistyneempiä NURBS-työkaluja ei tulla toteuttamaan ainakaan interaktiivisina. Toteuttamatta jäävät esimerkiksi usean NURBS-mallin liittäminen yhteen ja trimmaaminen, eli reikien muodostamisen NURBS-pintoihin.

1.3 Työn rakenne

Luvussa 2 tutustutaan NURBS-malleihin ja muihin työssä käytettäviin tekniikoihin. Lisäksi esitellään tutkimusongelma.

Luvussa 3 esitellään, kuinka tutkimusongelman ratkaisu toteutettiin, minkälaisia haasteita tuli vastaan, ja mitä saatiin tulokseksi.

Luvussa 4 arvioidaan työn tulosten merkitsevyyttä ja tulevaisuutta.

Luku 5 on yhteenveto koko työstä.

2 TEORIAOSUUS

2.1 Splinit

Splinit ovat paloittain määriteltyjä polynomifunktioita, joilla voidaan esittää käyriä ja pintoja. Niiden muotoa muokataan ensisijaisesti siirtämällä kontrollipisteitä, joita splinin muoto seuraa. Splinit määritellään parametrisesti ja niiden parametriavaruus jaetaan paloihin solmuilla [2, s. 5].

Ennen tietokoneiden keksimistä suunnittelijat piirsivät suunnitelmia käsin käyttäen erilaisia apuvälineitä, kuten viivoittimia, astemittoja ja harppeja [4]. Näillä saatiin tehtyä halutunlaisia suoria linjoja, kulmia ja ympyräkaaria. Vapaamuotoisten kaarien tekemiseen tarvittiin toisenlaisia apuvälineitä. Pienien kaarien piirtäminen onnistui osaavalta tekijältä luotettavasti vapaalla kädellä, mutta esimerkiksi laivanrakennuksessa tarvittiin erityisen suuria piirustuksia, joiden koko teki käsin piirtämisen mahdottomaksi. Apuvälineinä käytettiin splineiksi kutsuttuja pitkiä, kapeita taipuisia kappaleita. Splinejä pidettiin paikoillaan ja niiden käyryyttä hallittiin lyijypainoilla.

Splinien käyttö tietokoneavusteisessa suunnittelussa sai alkunsa 1950-luvun lopulla ja 1960-luvun alussa Paul de Casteljaun ja Pierre Beziérin töiden pohjalta [5, s. 609]. Splinejä käytettiin rungon muotojen suunnitteluun autoteollisuudessa.

2.1.1 NURBS

Tässä työssä käsitellään splineistä erityisesti NURBSeja. NURBS on Bezier-käyrien ja B-splinien yleistys. Lyhenne NURBS tulee sanoista “nonuniform rational basis spline”. Nonuniform tarkoittaa, että splinin solmujen välin pituus voi vaihdella, eli splini on jaettu paloihin, jotka eivät ei ole välttämättä tasavälisiä. Rationaalisuus puolestaan tarkoittaa käytännössä sitä, että kullakin kontrollipisteellä on painokerroin ω. Painokerroin mahdollistaa splinin muodon tarkemman hallinnan. Sana basis tarkoittaa kantaa ja tulee siitä, että splini on määritelty kantafunktioiden avulla.

Les A. Pieglin ja Wayne Tillerin kirja The NURBS Book [6] on yksi kattavimmista NURBSeja käsittelevistä teoksista. Teoksessa käsitellään laajasti NURBS-käyrien ja -pintojen muodostusta ja erilaisia niihin liittyviä algoritmeja. Les Pieglin artikkeli On NURBS: a survey [1] käy läpi NURBSien erilaisia käyttötarkoituksia ja ominaisuuksia, ja on muutenkin varsin hyvä katsaus aihealueesta. NURBS on keskeinen käsite tietokonegrafiikassa, ja monet laajat tietokonegrafiikka-aiheiset kirjat käsittelevät myös NURBSeja [5, 7].

NURBS-käyrän muodostamiseen tarvitaan käyrän kertaluku, solmuvektori ja painotetut kontrollipisteet. Solmuvektori U jakaa käyrän parametrisen avaruuden solmuväleihin.

Käyrän kertaluku kertoo kuinka monta kontrollipistettä on aktiivisena kerrallaan kullakin solmuvälillä. Käyrän aste on käyrän kertaluku vähennettynä yhdellä. Esimerkiksi kertaluvun kaksi NURBS-käyrän aste on yksi, ja se on murtoviiva, joka kulkee kontrollipisteiden kautta.

Painokerroin vaikuttaa siihen, kuinka voimakkaasti kontrollipiste vetää käyrää puoleensa [1]. Painokertoimen kasvattaminen saa käyrän kulkemaan lähempänä pistettä ja terävämpänä. Painokertoimen pienentäminen vastaavasti loiventaa käyrää ja vähentää kontrollipisteen vaikutusta. Painokertoimien vaikukset riippuvat niiden keskinäisestä suhteesta. Mikäli kaikkia painokertoimia muutetaan samassa suhteessa, lopullinen käyrä pysyy samana.

Käyrän solmuvektori on joukko reaalilukuja, joita kutsutaan solmuiksi. Vektorin luvut ovat suuruusjärjestyksessä pienimmästä suurimpaan. Käyrän arvoja laskettaessa kunkin solmun kohdalla aktivoituu uusi kontrollipiste ja vastaavasti yksi poistuu aktiivisten kontrollipisteiden joukosta. Ainoastaan kulloinkin aktiiviset kontrollipisteet vaikuttavat yksittäiseen käyrän pisteeseen, ja kaikkien muiden kontrollipisteiden kantafunktio on nolla. Solmuvektorin absoluuttiset arvot eivät vaikuta lopputulokseen, vaan niiden keskinäiset suhteet ratkaisevat käyrän muodon. Solmuvektorin siirtäminen (vakion lisääminen jokaiseen vektorin alkioon) tai skaalaaminen (vektorin kertominen skalaarilla) eivät vaikuta lopulliseen käyrään.

NURBS-käyrän jatkuvuus riippuu ensisijaisesti sen asteesta, mutta siihen voidaan vaikuttaa myös solmuvektorin valinnalla. Jatkuvuus voidaan jakaa parametriseen C^n ja geometriseen G^n jatkuvuuteen [8]. Geometrisessa jatkuvuudessa kappaleen muoto on sileä, ja parametrisessa jatkuvuudessa käyrän parametrisointi on sileä. Kun käyrällä mallinnetaan kappaleen liikettä, ja liikeradan muodon lisäksi etenemisnopeuden on oltava jatkuva, tarvitaan parametrista jatkuvuutta. Parametrinen jatkuvuus on geometrisen jatkuvuuden erityistapaus, ja yleensä ainoastaan geometrinen jatkuvuus on tärkeää kappaleiden muodon mallinnuksessa. Asteen nolla käyrä koostuu yksittäisistä pisteistä, joilla ei ole jatkuvuutta. Asteen yksi käyrä on murtoviiva, jossa sijainti pysyy jatkuvana solmuvälin muuttuessa. Käyrällä on tällöin geometrinen G^0-jatkuvuus, eli käyrän muodon nollas derivaatta (sijainti) on jatkuva. Astetta kaksi olevassa käyrässä käyrän suuntavektori tangentti pysyy jatkuvana solmupisteissä. Tällöin saavutetaan G^1-jatkuvuus. Vastaavasti kolmannen asteen käyrässä käyrän suuntavektorin muutosnopeus on jatkuva solmupisteissä, ja tällöin käyrällä on G^2-jatkuvuus. Myös korkeamman asteen jatkuvuudet ovat mahdollisia, mutta niiden tarve on usein vähäinen käytännön sovelluksissa.

NURBS-käyrä voidaan määritellä seuraavanlaisesti [1, 5, 6, 9]:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\omega_i P_i}{\sum_{i=0}^{n} N_{i,p}(u)\omega_i} \qquad a \leq u \leq b \tag{1}$$

Kaavassa 1 esiintyvät P_i ovat käyrän kontrollipisteet, ja ω_i ovat kontrollipisteitä vastaavat

painokertoimet. Käyrän aste on p. $N_{i,p}(u)$ ovat käyrän kantafunktiot:

$$N_{i,0}(u) = \begin{cases} 1 & \text{jos } u_i \leq u < u_{i+1} \\ 0 & \text{muutoin} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+i} - u_{i+1}} N_{i+1,p-1}(u) \tag{2}$$

jossa u_i on solmuvektori: $\mathrm{U} = \{u_0, u_1, ..., u_m\}$

NURBSeilla voi esittää käyrien lisäksi myös pintoja, jolloin käyrästä tulee kahden parametrin funktio. Pinnan muodostamiseen käytetään kahden käyrän tensorituloa:

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{k} N_{i,p}(u) N_{j,q}(v) \omega_{i,j} P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{k} N_{i,p}(u) N_{j,q}(v) \omega_{i,j}} \tag{3}$$

Kaavassa 3 esiintyvät p ja q ovat parametrien u ja v suuntaiset asteluvut. Kantafunktiot $N(u)$ ja $N(v)$ on määritelty kaavassa 2.

2.1.2 NURBSien ominaisuuksia

NURBSeilla on lukuisia hyödyllisiä ominaisuuksia, minkä takia niitä käytetäänkin yleisesti tietokoneavusteisessa suunnittelussa.

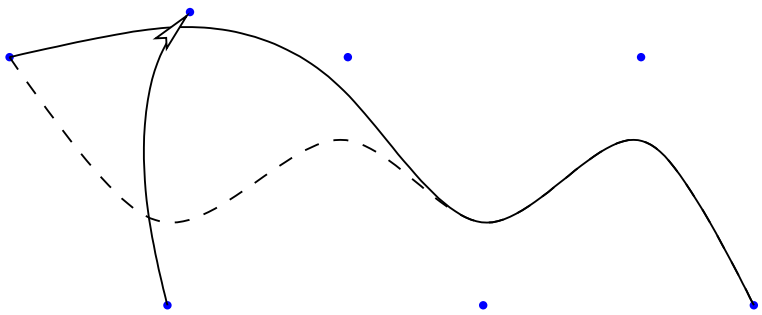
NURBSit soveltuvat sekä analyyttisten että vapaamuotoisten kappaleiden esittämiseen. Rationaalisuuden ansiosta NURBS-käyrillä voidaan esittää kartioleikkauksia tarkasti, kun epärationaalisilla B-splineillä niitä voidaan ainoastaan approksimoida. Toinen NURBSien erityisominaisuus on se, että ne ovat invariantteja perspektiivimuunnosten suhteen [1]. NURBS-käyrän laskeminen ja perspektiivimuunnoksen tekeminen erikseen jokaiselle käyrän pisteelle antaa saman tuloksen kuin se, että tekee perspektiivimuunnoksen ainoastaan kontrollipisteille ja laskee käyrän vasta sitten. B-splinien tapauksessa nämä eri tavat antaisivat eri tuloksen.

Epätasaiset solmuvälit mahdollistavat käyrän jatkuvuuden hallinnan. Yhden moninkertaisen solmun lisääminen, eli solmun laittaminen kohtaan, jossa on ennestään solmu, pienentää käyrän jatkuvuutta kyseisessä pisteessä yhdellä. Sijoittamalla päätepisteisiin käyrän kertaluvun verran solmuja käyrä saadaan alkamaan ensimmäisestä ja loppumaan viimeiseen kontrollipisteeseen, mikä on usein haluttu ominaisuus.

NURBSeilla on myös muita hyödyllisiä ominaisuuksia, joista monet ovat yhteisiä myös muiden splinien kanssa. B-splini tai Bézier-käyrä voidaan esittää NURBSina, koska se on

niiden yleistys.

Eräs splinien tärkeimmistä ominaisuuksista on paloittaisen määrittelyn mahdollistama paikallinen kontrolli. Kontrollipisteen siirtäminen vaikuttaa vain siihen splinin väliin, joissa kontrollipistettä vastaava kantafunktio ei ole nolla. Kyseistä väliä kutsutaan kantafuntion tueksi (support) [5]. Splineistä poiketen esimerkiksi Beziér-käyrällä kantafunktioiden tuki kattaa koko määrittelyalueen, mikä tarkoittaa sitä, että minkä tahansa kontrollipisteen liikuttaminen vaikuttaa aina koko käyrän muotoon. Lokaalia kontrollia on havainnollistettu kuvassa 1.



Kuva 1. Lokaali kontrolli mahdollistaa käyrän osien paikallisen muokkaamisen. Mustat pisteet ovat käyrän kontrollipisteitä. Käyrän muoto ennen kontrollipisteen siirtoa on merkitty katkoviivalla.

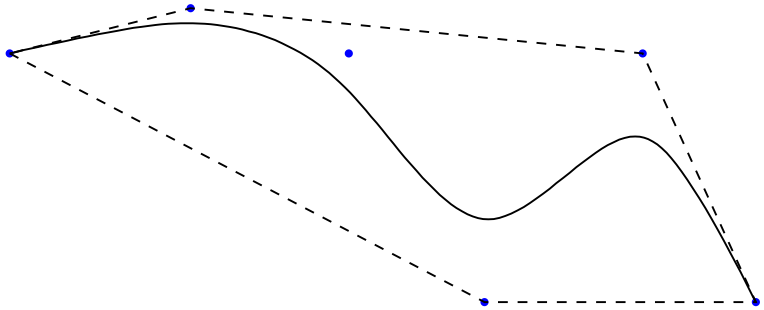
NURBS-käyrä muodostuu aina sen kontrollipisteiden konveksin verhon (convex hull) sisäpuolelle. Kontrollipisteiden konveksi verho on niiden kaikkien konveksien kombinaatioiden muodostama joukko, eli kaikki pisteet, jotka saadaan kaavalla [5, s. 615]:

$$\sum_i \alpha_k P_k, \qquad \sum \alpha_k = 1$$

Kaksiulotteisessa tapauksessa tämä vastaa sitä, että asetetaan kuminauha kontrollipisteiden ympärille ja annetaan sen kiristyä. NURBS-mallinnuksessa on hyödyllistä tietää, että käyrä tai pinta ei voi koskaan ulottua rajatun alueen ulkopuolelle. Konveksia verhoa on havainnollistettu kuvassa 2.

2.2 Java

Java on monipuolinen yleiskäyttöinen oliopohjainen ohjelmointikieli. Javalla voidaan kehittää sekä www-selaimessa toimivia sovelmia (applet) että itsenäisesti ajettavia sovelluksia (application). Sovelman ajamiseen selaimessa tarvitaan selainliitännäinen, joka on vapaasti ladattavissa Sun Microsystemsin verkkosivustolta [10]. Java-kehitystä varten tarvitaan JDK (Java Development Kit), joka on myös ladattavissa vapaasti Sunin sivustolta. Javasta on olemassa myös useita avoimen lähdekoodin toteutuksia, kuten OpenJDK [11]



Kuva 2. NURBS-käyrä on aina konveksin verhon sisäpuolella. Mustat pisteet ovat käyrän kontrollipisteitä. Kontrollipisteiden muodostama konveksi verho on merkitty katkoviivalla.

ja GCJ [11]. Suurin osa Java-teknologioista onkin saatavilla vapaan lähdekoodin ohjelmistoina, joita voi muokata ja jakaa edelleen vapaasti.

Java-ohjelma käännetään alustariippumattomaksi tavukoodiksi, joka puolestaan voidaan suorittaa missä tahansa tuetussa ympäristössä virtuaalikoneella [12]. Virtuaalikone muuntaa tavukoodin käskyt käytettävän ympäristön ymmärtämään muotoon.

Toinen suosittu vaihtoehto selaimessa toimivien ohjelmienn toteuttamiseen on kaupallinen Adobe Flash [13], jonka kehitystyökalut ovat maksullisia, mutta toisto-ohjelma ilmainen. Flashissa ei ole ollut 3D-ominaisuuksia ennen uusinta 10-versiota, ja sen painopiste on multimediassa.

2.3 OpenGL

Kolmiulotteisten NURBS-mallien esittämiseen tarvitaan 3D-grafiikkarajapinta. Javassa ei ole sellaista valmiina, joten grafiikan piirtoa varten on käytettävä ulkoista kirjastoa. OpenGL on suosittu 2D/3D-grafiikkarajapinta, joka on saatavilla lukuisille eri järjestelmäalustoille. JOGL (Java OpenGL) [14] on matalan tason OpenGL-rajapinta Javalle ja mahdollistaa laitteistokiihdytetyn kolmiulotteisen grafiikan luomisen Javalla. JOGL:n etuihin kuuluu sen tehokkuus, mutta se ei käytännössä hyödynnä Javan oliopohjaisuutta [12, s. 13-20], koska se on lähes suora käännös OpenGL:n C-kielisestä toteutuksesta. JOGL tarjoaa käytännössä suoran pääsyn OpenGL:n funktioihin, joihin kuuluvat myös NURBS-käyrien ja -pintojen piirto. Voidakseen hyödyntää näytönohjaimen grafiikkaominaisuuksia JOGL käyttää C-kielistä OpenGL-rajapintaa, jonka toimintoja se kutsuu JNI-rajapinnan (Java Native Interface) kautta. Tätä varten tietokoneessa on oltava OpenGL-tuki ja asennettuna JOGL:n JNI-kirjastot, jotka ovat laitteisto- ja käyttöjärjestelmäkohtaisia.

2.4 Tutkimusongelman ja sen ratkaisutavan esittely

NURBS on monipuolinen ja suosittu työkalu käyrien ja pintojen mallinnukseen. NURBS-kappaleen kontrollipisteiden toiminta on kohtalaisen helppo ymmärtää, mutta kantafunktioiden ja solmuvektorin merkitysten ymmärtäminen on vaikeampaa. Sen takia olisi hyvä kehittää järjestelmä, joka auttaisi ymmärtämään NURBSien toimintaa. Järjestelmän tulisi olla helposti käytettävissä ja mielellään myös riippumaton laitteistosta tai käyttöjärjestelmästä ja helposti saatavilla. Www-ympäristö tarjoaa mahdollisuuden rakentaa laitteistoriippumattomia ohjelmistoja, jotka ovat saatavilla kaikkialla, missä on Internet-yhteys. Näiden pohjalta muotoillaan tutkimusongelma:

Kuinka toteutetaan www-ympäristössä toimiva ohjelma, joka esittelee NURBSit intuitiivisesti ja havainnollisesti?

Ratkaisuna tutkimusongelmaan esitellään www-selaimessa toimiva Java-sovelma. Sovelma esittelee NURBSien eri ominaisuuksia vaiheittain ja antaa sanallisen kuvauksen siitä, mitä kulloinkin ollaan esittelemässä. Samalla käyttäjä pääsee muokkaamaan mallia. Käyttöliittymästä tulee hiirikäyttöinen ja se perustuu kappaleiden suoraan manipulointiin painamalla niitä tai vetämällä niitä osoittimella. Interaktiivisuus tekee oppimisesta konkreettisempaa toimintaa, ja tällä konkreettisella tekemisellä oppija saa välitöntä palautetta toiminnastaan [15, s.180-181]. Kohteiden suoraan manipulointiin pohjautuva käyttöliittymä on helppo oppia ja kannustaa kokeilemaan [16, s.72].

Ratkaisussa hyödynnetään soveltuvin osin aikaisempien ratkaisujen perusideoita. Hyvänä esimerkkinä toimii Evgeny Demidovin verkkosivusto An Interactive Introduction to Splines [17], joka käsittelee erilaisia splinejä, mukaanlukien NURBSit. Sivustolla on useita erilaisia interaktiivisia Java-sovelmia, joissa voi muokata kaksiulotteisia splinikäyriä. Käyriä on havainnollistettu väreillä ja myös niiden kantafunktiot ovat nähtävissä hiirellä liikuteltavina käyrinä, mikä on lähtökohtana myös tässä työssä. Käyrien astetta eikä kontrollipisteiden määrää voi muuttaa. Sovelmien Java-lähdekoodit ovat vapaasti ladattavissa. Myös muita interaktiivisia NURBS-esittelysovelmia on olemassa [18, 19, 20], mutta niiden ongelmana on usein epäintuitiivinen käyttöliittymä, tai puuttuvat ominaisuudet, kuten pintojen mallintaminen.

3 KÄYTÄNNÖN OSUUS

3.1 Käytännön toteutus

NURBS-sovelma tehtiin Eclipse-ohjelmointiympäristössä Java-kielellä. Koska Introduction to Computer Graphics -kurssi, jolle sovellusta tehdään, on englanninkielinen, myös sovelma on englanninkielinen. Tässä kappaleessa käsitellään projektin toteutusta, työvälineitä ja vastaan tulleita ongelmia.

3.1.1 Suunnittelu

Projektin suunnittelu alkoi etsimällä aiheeseen liittyvää kirjallisuutta ja erilaisia valmiita NURBS-toteutuksia. Valmiiden toteutusten hyviä ja huonoja puolia vertailtiin, ja niiden pohjalta mietittiin, minkälainen sovelman tulisi olla. Samalla pohdittiin, mitä kaikkea sovelmalta vaaditaan. Näiden perusteella tehtiin vaatimusmäärittely listaamalla sovelmalta halutut ominaisuudet ja se, kuinka tärkeää kunkin ominaisuuden toteutuminen on. Vaatimusmäärittelyn perusteella luonnosteltiin alustava luokkakaavio, jotta sovelman rakenne olisi tiedossa, ennen kuin sitä aletaan ohjelmoida. Vaatimusmäärittely on esitetty liitteessä 1.

Rinnakkain vaatimusmäärittelyn teon kanssa ohjelmoitiin tärkeimmät perusominaisuudet sisältävä sovelman esiversio, jotta saatiin selville, onko projekti toteutettavissa järkevästi valituilla menetelmillä ja työkaluilla. Ennen kaikkea Java OpenGL -kirjaston toimiminen luotettavasti erilaisilla tietokoneilla epäilytti etukäteen. Esiversion toimintaa kokeiltiin eri tietokoneilla ja sekä Linux- että Windows-ympäristöissä, ja se toimi riittävän hyvin, jotta projektia voitiin jatkaa luottavaisin mielin. Esiversiossa oli yksittäinen NURBS-käyrä, jonka kontrollipisteitä pystyi liikuttamaan ja jonka kantafunktiot olivat näkyvissä graafisina, mutta ei muita toimintoja.

3.1.2 NURBS-toteutus

Projektin alussa pohdittiin mahdollisuutta käyttää sellaisenaan tai soveltaa valmista NURBS-kirjastoa. JOGL:n tarjoamat NURBS-piirtofunktiot eivät riittäneet ominaisuuksiltaan, ja lupaavin NURBS-kirjasto oli Jgeom [21]. Ajatus oli, että koska Jgeom on julkaistu avoimen lähdekoodin lisenssillä, sitä voitaisiin muokata sovelman tarpeisiin. Jgeomille ei kuitenkaan löytynyt kunnollista dokumentaatiota eikä kunnolla esimerkkikoodia, joten ajatuksesta luovuttiin turhan hankalana.

Sovelmaa varten ohjelmoitiin kaksi eri NURBS-luokkaa, joista toinen on NURBS-käyrille ja toinen NURBS-pinnoille. NURBS-käyrä toteutettiin aiemmin esiteltyjen kaa-

vojen 1 ja 2 pohjalta. NURBS-pintojen toteutukseen käytettiin käyriä varten luotua pohjaa, joka muokattiin ja laajennettiin kaavan 3 mukaiseksi. Pintojen tapauksessa melko iso osa ohjelmakoodista on käytännössä toistettu kahdesti, kerran parametrin u suuntaan ja kerran parametrin v suuntaan. Toiston määrään vähentäminen olisi ollut mahdollista käyttämällä taulukoita, mutta nykyinen ratkaisu toimii riittävän hyvin. Sovelman NURBS-toteutuksen periaatteena on, että sen sisäinen toiminta pystytään esittämään graafisesti, vaikkakin suorituskyvyn kustannuksella. Sovelmassa käytettävät mallit ovat kuitenkin sen verran pieniä, että niiden laskemisessa ei pitäisi olla suorituskykyongelmia.

NURBS-luokkien toteuttaminen oli mielenkiintoinen haaste. Luokkien toteutuksessa selvästi hankalimmin ymmärrettävä suoraan NURBSeihin liittyvä osa-alue oli solmuvektorin toiminta. Sovelman NURBS-toteutuksessa solmuvektoreita pystyy muuttamaan vapaasti ja myös moninkertaiset solmut on huomioitu. Mikäli solmujen moninkertaisuus jossain pisteessä ylittää NURBSin asteen, syntyy epäjatkuvuus. Muut osa-alueet oli kohtuullisen helppo toteuttaa, ja hankaluudet liittyivät lähinnä kosmeettisiin yksityiskohtiin. Toinen hankalammin toteutettavissa ollut osuus oli kontrollipisteiden ja solmujen valinta ja liikuttaminen hiirellä, mutta ne saatiin toteutettua hyödyntämällä OpenGL:n omaa kappaleenvalintajärjestelmää ja erilaisia projektioita.

NURBS-luokkien antamia tuloksia verrattiin OpenGL:n NURBS-piirtofunktioiden antamiin tuloksiin, ja ne muistuttivat hyvin paljon toisiaan, niin kuin niiden pitäisikin. Yhtenä eroavaisuutena sovelman NURBS-luokat käyttävät euklidisia koordinaatteja painokertoimella, kun OpenGL käyttää homogeenisiä koordinaatteja NURBSien renderointiin. Käytännössä ero näkyy siinä, että koordinaatit on kerrottava painokertoimella, ennen kuin ne syötetään OpenGL:n NURBS-funktiolle.

3.1.3 Eclipse-ohjelmointiympäristö

Projektin toteutuksessa käytetty Eclipse [22] on pääosin Javalla kirjoitettu monipuolinen ohjelmointiympäristö. Eclipse on tehty ensisijaisesti Java-kehitystä varten, mutta eri liitännäisten avulla siihen on saatavilla tuki myös monille muille ohjelmointikielille. Eclipse on saatavilla sekä Linuxille että Windowsille, mutta tässä projektissa käytettiin ainoastaan Linux-versiota. Käytetty Eclipsen versio oli 3.2.2.

Projektin edetessä Eclipse havaittiin varsin hyväksi työkaluksi. Se hallitsee ja osaa hyödyntää monia Javan erityispiirteitä, mikä tekee siitä tehokkaamman kuin yleiskäyttöisempi ohjelmoijan tekstieditori olisi. Esimerkiksi muuttujan uudelleennimeäminen useammasta tiedostosta kerrallaan sujui täysin ongelmitta. Myös projektin siirtäminen kehityskoneesta toiseen onnistui kohtalaisen helposti.

Täydellinen ympäristö Eclipse ei kuitenkaan ole, ja ajoittain se yrittää ennakoida käyttäjän aikeita turhan paljon. Lisäksi se on joissain asioissa melko raskas: kohtalaisen modernilla kehityskoneella automaattinen tekstin täydennys saattoi pysäyttää ohjelman usean sekunnin ajaksi kesken kirjoittamisen. Tämä oli hyvin turhauttavaa silloin, kun tekstin

olisi saanut kirjoitettua nopeammin suoraan ilman täydennystä. Ongelma ratkesi ottamalla automaattisen täydennyksen pois päältä.

Varsinaisia suuria ongelmia Eclipsen kanssa ei ollut. Kerran Eclipsen työympäristön asetustiedosto korruptoitui jostain syystä, ja asetukset jouduttiin nollaamaan. Työympäristön asetuksia täytyi säätää uudestaan, mutta varsinaiset projektitiedot kuitenkin pysyivät tallessa.

3.1.4 Java ja Java OpenGL

Javasta on olemassa useita erilaisia toteutuksia, jotka kuitenkin toteuttavat pääpiirteittäin samat ominaisuudet. Projektissa käytettiin Sun Microsystemsin virallisen JDK-kehitysympäristön versiota 1.5. Myös kahta avoimen lähdekoodin Java-toteutusta kokeiltiin, mutta ne eivät toimineet toivotulla tavalla. OpenJDK on Sun Microsystemsin käynnistämä projekti, jonka tarkoitus on luoda kokonaan vapaaseen ja avoimeen lähdekoodiin perustuva JDK-toteutus. OpenJDK toimi muuten hyvin, mutta JOGL:n tekstien renderointi ei toiminut sen kanssa ollenkaan. Toisella avoimella Java-kääntäjällä GCJ:llä sovelma ei kääntynyt ollenkaan. Ainakin nykyisellään on siis turvauduttava Sunin omaan toteutukseen.

Java-kielellä ohjelmoinnista ei ollut etukäteen kokemusta muuten kuin hyvin pienillä kokeiluluontoisilla ohjelmilla, mutta se ei kuitenkaan aiheuttanut pahoja ongelmia. Joitain ratkaisuja kannattaisi tehdä eri tavoin, jos sovelma pitäisi tehdä nyt uudestaan, mutta suuria ongelmakohtia ei ole tiedossa nykyisessä ratkaisussa. Eniten epävarmoja kohtia on sovelman luokkarakenteessa ja luokkien välisessä viestinnässä, kuten oli odotettavissakin. Java-kielessä miellytti esimerkiksi C++:n verrattuna ohjelmoijan kannalta huomattavasti yksinkertaisempi muistinhallinta, joka huolehtii muistin vapauttamisesta automaattisesti.

Järjestelmää suunniteltaessa oli vielä vähän epävarmaa, kuinka hyvin JOGL toimii erilaisissa ympäristöissä. Koska JOGL tarvitsee suoran pääsyn OpenGL-rajapintaan, tietokoneeseen täytyy olla asennettu JOGL:n JNI-kirjastot, jotta JOGL voi toimia. Www-sovelman kannalta tämä voisi olla hankalaa, mutta asennuksen voi automatisoida käyttämällä Javan JNLPAppletLauncher-luokkaa. JNLPAppletLauncher lataa sovelmaa ajettaessa tarvittavat järjestelmäkohtaiset JNI-kirjastot, mikäli niitä ei ole vielä asennettu. Lataamiseen kuluva aika riippuu käytettävän Internet-yhteyden nopeudesta. Asennuksen todettiin toimivan hyvin, eikä se tarvitse montaa napin painallusta käyttäjältä. JOGL toimii kohtalaisen tasalaatuisesti eri järjestelmissä. Joissain yksityiskohdissa on pieniä eroja, mutta kun ne huomioidaan ohjelmakoodissa, sovelma saadaan näyttämään kohtalaisen samanlaiselta riippumatta alustasta.

Eräs JOGL:n etu on se, että se mahdollistaa OpenGL:n valmiiden NURBS-piirtofunktioiden käytön. Valmiiden piirtofunktioiden ominaisuudet eivät kuitenkaan riittäneet siihen, että niitä olisi voinut käyttää järkevästi lopullisessa sovelmassa. Kehitysvaiheessa ne olivat kuitenkin hyvin hyödyllisiä ja auttoivat omien ratkaisujen tulosten

tarkistamisessa.

JOGL ei tuonut suurempia ongelmia itse ohjelmointiin, mutta oli selvästi kohtalaisen suora käännös C-kielisestä rajapinnasta, eikä käytännössä hyödynnä Javan oliopohjaisuutta. Matalan tason rajapintana JOGL kuitenkin tarjoaa ohjelmoijalle hyvät mahdollisuudet toteuttaa ideoitaan vapaasti. Matalasta abstraktiotasosta huolimatta JOGL:n käyttäminen on melko helppoa ja yksinkertaista. Jotkin yksityiskohdat poikkeavat hieman normaalista C-kielisestä OpenGL-toteutuksesta, ja voivat olla huonosti dokumentoituja, mutta useimmiten olemassaolevilla esimerkeillä pääsee hyvin eteenpäin.

3.2 Tulokset

Tässä kappaleessa esitellään millainen itse sovelmasta tuli.

Projektin tuloksena tuotettiin Java-sovelma ja verkkosivu. Verkkosivu ja sovelma siirrettiin yliopiston palvelimelle. Verkkosivu sisältää yksinkertaiset ohjeet ohjelman käyttöön, ja painopiste on itse sovelmalla. Sovelman suorittamiseen tarvitaan www-selain, jossa on asennettuna Java-liitännäinen. Koska sovelmassa käytetään kolmiulotteista grafiikkaa, sen sujuvaan suorittamiseen tarvitaan kohtuullisen nykyaikainen näytönohjain. Sovelman testattiin toimivan sekä Linux- että Windows-käyttöjärjestelmissä.

Sovelma on interaktiivinen NURBS-opas. Sovelma opastaa käyttäjää NURBSien eri aihealueissa ja päästää käyttäjän samalla muokkaamaan malleja. Kukin aihealue on jaettu useaan vaiheeseen, joissa käyttäjä pääsee eteenpäin painamalla seuraava-painiketta.

3.2.1 Aihealueet

Kukin aihealue on pieni kokonaisuus, joka käsittelee NURBSien yhtä osa-aluetta. Yksittäinen osa-alue on jaettu useaan vaiheeseen, joissa käyttäjä pääsee eteenpäin painamalla seuraava-painiketta. Sovelmassa on seuraavat aihealueet:

0) Esittely

Toivottaa käyttäjän tervetulleeksi ja näyttää esimerkin NURBS-käyrästä.

1) Kontrollipisteet

Käyttäjä tutustuu käyrän kontrollipisteisiin ja siihen, kuinka niitä voi liikuttaa. Myös kameran liikuttamiseen tutustutaan. Lisäksi sovelma kertoo, kuinka käyrän kontrollipisteiden painoja voi muuttaa. Lokaalin kontrollin tarkoitus selitetään.

2) Kantafunktiot

Käyrän rationaaliset kantafunktiot esitellään graafisina käyrinä. Sovelma kertoo, kuinka kantafunktioiden avulla lasketaan käyrä. Kertaluvun ja asteluvun merkitys esitellään näyttämällä, miten kertaluvun muuttaminen vaikuttaa käyrän muotoon. Lopussa käyttäjä pääsee muuttamaan kertalukua itse.

3) Solmuvektori

Kantafunktioiden esitykseen tulee mukaan solmuvektori, jota käyttäjä voi muuttaa, ja myös solmuvälit esitetään graafisesti. Käyttäjä tutustutetaan erityyppisiin solmuvektoreihin ja moninkertaisiin solmuihin. Lopussa esitetään ympyrä esimerkkinä solmuvektorin ja painojen käytöstä analyyttisten kappaleiden muodostuksessa.

4) Vapaa käyrän muokkaus

Käyttäjä voi muokata käyrää vapaasti, lisätä siihen uusia kontrollipisteitä tai poistaa vanhoja. Kontrollipisteiden lisäys ja poisto tapahtuvat käyrän loppupäässä, koska kontrollipisteen lisäys ja poisto mielivaltaisessa käyrän kohdassa olisi vaatinut monimutkaisemman toteutuksen. Käyttäjä voi myös ladata ohjelmaan sisäänrakennettuja valmiita käyrämalleja. Kaikki käyrän parametrit ovat muutettavissa.

5) NURBS-pinnat

NURBS-pinnat esitellään. Nyt käytettävissä on sekä U- että V-suuntainen solmuvektori. Tutustutaan siihen, miten eri solmuvektorien muuttaminen vaikuttaa pintaan, ja kuinka pinnan kantafunktiot muodostetaan. Käyttäjä pääsee muuttamaan pinnan U- ja V-suuntaista kertalukua. Lopuksi esitellään sylinteri esimerkkinä analyyttisestä NURBS-pinnasta.

6) Vapaa pinnan muokkaus

Käyttäjä voi muokata pintaa vapaasti, lisätä siihen uusia kontrollipisterivejä tai poistaa vanhoja. Käyttäjä voi myös ladata ohjelmaan sisäänrakennettuja valmiita pintamalleja. Kaikki käyrän parametrit ovat muutettavissa.

3.2.2 Käyttöliittymä

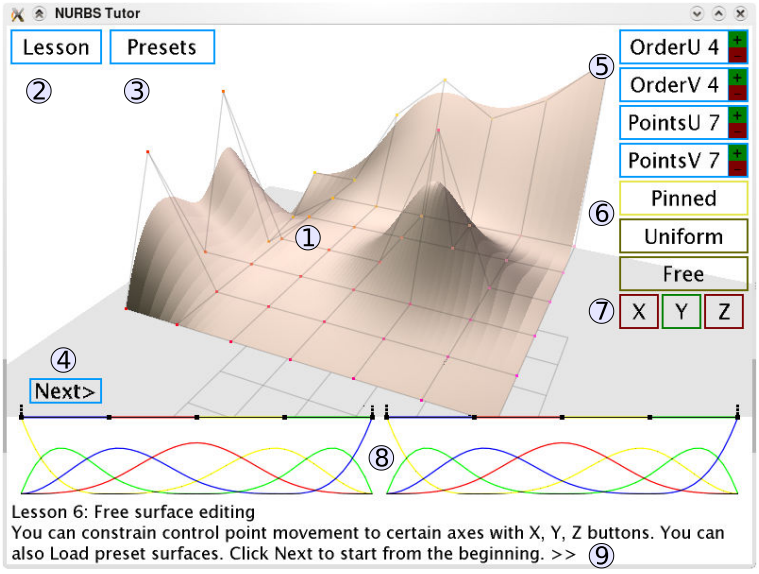
Sovelman käyttöliittymä on englanninkielinen. Käyttöliittymän perusajatus on, että käyrät ja niiden kantafunktiot esitetään graafisesti, ja niitä voidaan liikuttaa hiirellä. Tässä on otettu mallia Evgeny Demidovin verkkosivustosta An Interactive Introduction to Splines [17]. Sivusto käsittelee erilaisia splinejä, mukaanlukien NURBSeja. Sivustolla on useita erilaisia interaktiivisia Java-sovelmia, joissa voi liikuttaa kaksiulotteisia splinikäyriä. Käyriä on havainnollistettu väreillä ja myös niiden kantafunktiot ovat nähtävissä hiirellä liikuteltavina käyrinä.

Kaikki sovelman toiminnot ovat käytettävissä tavallisella kaksipainikkeisella hiirellä. Hiiressä on hyvä olla vieritysrulla, koska sillä saa muutettua kontrollipisteiden painoja ja lähennettyä tai loitonnettua kameraa helposti ja käytännöllisesti. Kaikissa hiirissä rullaa ei ole, eikä myöskään kannettavan tietokoneen tuntolevyssä, mutta ne on huomioitu sovelman toteutuksessa, ja samat toiminnot ovat käytettävissä myös normaaleilla osoitinlaitteen painikkeilla. Sovelman NURBSien muokkaamiseen tarkoitetut ominaisuudet ovat tarkoituksella yksinkertaisia. Yksinkertaisuus rajoittaa, mitä sovelmalla voi tehdä mielekkäästi, mutta kehittyneempien työkalujen lisääminen kasvattaisi työmäärää ja tekisi käyttöliittymästä monimutkaisemman, mikä ei ole mielekästä.

Käyttöliittymän elementit muuttavat ulkoasuaan osoittimen ollessa niiden päällä. Tämän tarkoitus on antaa käyttäjälle suoraa palautetta siitä, mikä kohde on milloinkin aktiivisena. Lisäksi painikkeet on pyritty pitämään kohtalaisen suurina, mutta kuitenkin ilman, että ne menevät häiritsevästi tielle. Fittsin lain [23, s. 442] mukaan kohteen osoittaminen on sitä nopeampaa, mitä suurempi kohde on.

Osa sovelman toiminnoista on aluksi piilossa, jotta ne eivät veisi huomiota käyttäjältä eivätkä hämmentäisi paljoudellaan. Toiminnot tulevat käytettäviksi vasta sitten, kun ne tulevat käyttäjälle olennaisiksi. Esimerkiksi solmuvektoria ei näytetä eikä sitä pääse muokkaamaan kuin vasta siinä vaiheessa, kun solmuvektoria aletaan käsitellä.

Käyttöliittymä on jaettu kolmeen perusosaan. Ylimmässä osassa on itse NURBS-käyrä tai -pinta ja painikkeita sovelman hallintaan. Seuraavassa osassa on käyrän solmuvektorista muodostettu kantafunktio. Pinnan tapauksessa esitellään sekä u- että v-suuntaiset kantafunktiot, joiden tensoritulosta pinnan lopullinen kantafunktio muodostuu. Alimmassa osassa on tekstikenttä, jossa sovelma kertoo käyttäjälle tietoa siitä, mitä kulloinkin ollaan havainnollistamassa. Sovelman käyttöliittymä ja sen osat on esitelty kuvassa 3.



Kuva 3. Sovelman käyttöliittymän osat.

1) NURBS-malli

NURBS-malli esitetään graafisesti ja on joko käyrä tai pinta riippuen siitä, missä tilassa ollaan. Myös mallin kontrollipisteet ja ne yhdistävä murtoviiva tai verkko ovat näkyvissä. Kun käyttäjä vie osoittimen kontrollipisteen päälle, ruudun vasemmassa laidassa näkyy kontrollipisteen tiedot, eli numerotunniste, koordinaatit ja paino.

Käyttäjä voi liikuttaa kontrollipisteitä painamalla osoitinlaitteen ensimmäistä painiketta pohjassa. Kontrollipisteiden painon muuttaminen tapahtuu joko vierittämällä osoitinlaitteen rullaa tai painamalla sen painikkeita samalla, kun osoittaa kontrollipistettä osoittimella. Kuvakulmaa voi kiertää liikuttamalla osoitinta osoitinlaitteen toinen painike pohjassa. Kuvakulman lähentäminen ja loitontaminen mallista tapahtuu osoittamalla hiirellä tyhjään kohtaan ja vierittämällä osoitinlaitteen rullaa.

2) Aihealueen valintapainike

Kun aihealueen valintapainiketta painetaan, valittavien aihealueiden luettelo tulee näkyviin. Tämän jälkeen käyttäjä voi siirtyä haluamaansa aihealueeseen painamalla vastaavaa painiketta. Luettelo poistuu näkyvistä, kun aihealue vaihdetaan.

3) Valmiiden NURBS-mallien valintapainike

Näkyvissä ainoastaan vapaissa käyrien tai pintojen muokkaustiloissa. Kun valmiiden NURBS-mallien valintapainiketta painetaan, sovelmaan valmiiksi tallennettujen NURBS-mallien luettelo tulee näkyviin. Uusi painallus piilottaa luettelon. Käyttäjä voi ladata mallin painamalla luettelosta haluamaansa painiketta. Luettelo näyttää joko ohjemaan tallennetut käyrät tai pinnat riippuen siitä, kummassa tilassa ollaan.

4) Seuraava-painike

Seuraava-painikkeella käyttäjä voi edetä aihealueen seuraavaan vaiheeseen. Mikäli aihealueen viimeinen vaihe on meneillään, seuraava-painikeella siirrytään seuraavaan aihealueeseen.

5) Kertalukupainikkeet

Näkyvissä ainoastaan osassa sovelman tiloista. Kertalukupainikkeilla käyttäjä voi muuttaa käyrän kertalukua ja astetta, aste on kertaluku - 1. Pinnan tapauksessa kertalukupainikkeita on kaksi, eli sekä u- että v-suunnalla on omansa.

6) Kontrollipisteiden määrä -painike

Näkyvissä ainoastaan osassa sovelman tiloista. Painikkeella käyttäjä voi joko lisätä kontrollipisteitä tai poistaa niitä. Pinnan tapauksessa painikkeita on kaksi, ja ne lisäävät uuden pisterivin joko u- tai v-suuntaan.

7) Solmuvektorin tyyppi -painike

Näkyvissä ainoastaan osassa sovelman tiloista. Painikkeita käyttämällä käyttäjä voi muuttaa solmuvektorin tyyppiä. Eri solmuvaktori tyyppejä on kolme: tasavälinen (uniform), tasavälinen toistetuilla päädyillä (pinned uniform) ja vapaa (free).

8) Kantafunktioiden graafinen esitys

Kantafunktiot esitetään graafisina käyrinä. Pinnan tapauksessa u- ja v-suuntaiset kantafunktiot esitetään erikseen. Kantafunktioiden yläpuolella on nähtävillä solmuvektori graa-

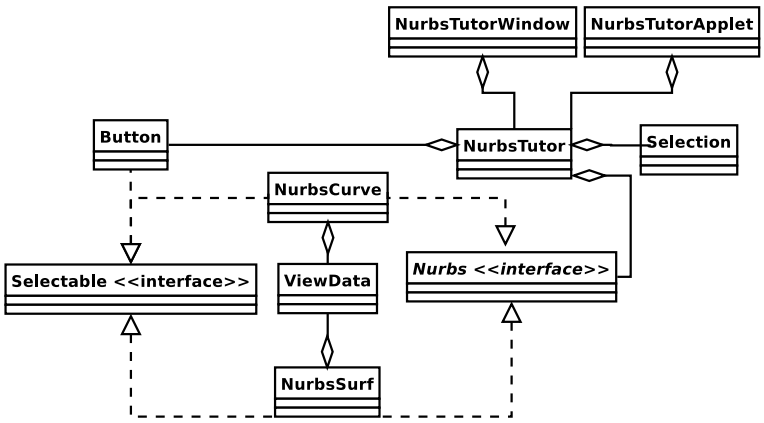
fisina pisteinä ja sen muodostamat solmuvälit väritettyinä. Solmuvektoria voi muuttaa vetämällä solmuja osoitinlaitteen ensimmäinen painike pohjassa. Solmuvektori on näkyvissä ainoastaan osassa ohjelman tiloista.

9) Tekstialue

Sovelma kertoo tekstialueella tietoa meneillään olevasta aihealueesta ja opastaa käyttäjää.

3.2.3 Luokkarakenne

Olio-ohjelmointi on keskeinen osa Java-kieltä. Olio-ohjelmoinnissa ohjelmointiongelmien ratkaisut esitetään olioiden välisenä yhteistoimintana. Kukin olio kuuluu johonkin luokkaan. Työssä käytettyjen luokkien väliset suhteet on esitetty kuvassa 4.



Kuva 4. Sovelman luokat ja niiden väliset suhteet.

NurbsTutorApplet

Sovelluksen aloituskohta, silloin kun se on käännetty selaimessa toimivaksi sovellukseksi. Luo www-sovelmalle piirtoalueen ja tekee muita alustustoimenpiteitä.

NurbsTutorWindow

Sovelluksen aloituskohta silloin kun sitä käytetään itsenäisesti toimivana ohjelmana. Vastaava kuin NurbsTutorApplet, mutta luo uuden ikkunan.

NurbsTutor

Luokka, joka sisältää ohjelman pääasiallisen toimintalogiikan. NurbsTutor luo käyttöliittymän ja huolehtii vuorovaikutuksesta käyttäjän kanssa välittämällä käyttäjän syötteen kulloinkin tarvittavalle oliolle, ja huolehtii kappaleiden piirtämisestä ruudulle.

Nurbs

Rajapinta, jonka avulla NurbsTutor pääsee käyttämään NURBS-luokkia samalla tavalla riippumatta siitä, ovatko ne pintoja vai käyriä.

NurbsCurve

Luokka NURBS-käyrien toteutukseen. Sisältää funktiot NURBS-käyrän luomiseen, piirtämiseen ja muokkaamiseen. Toteuttaa Nurbs- ja Selectable-rajapinnat.

NurbsSurf

Luokka NURBS-pintojen toteutukseen. Sisältää funktiot NURBS-pinnan luomiseen, piirtämiseen ja muokkaamiseen. Toteuttaa Nurbs- ja Selectable-rajapinnat.

ViewData

Luokka, johon tallennetaan NURBSien piirrossa käytetty projektiomatriisi. Matriisia hyödynnetään kontrollipisteiden ja solmupisteiden valinnassa hiirellä.

Selection

Luokka, jossa säilytetään kohdistimen valintaan liittyvää dataa, kuten hiiren sijainti painallushetkellä ja tämänhetkinen valittu kohde.

Selectable

Rajapinta, jonka toteuttavat kaikki luokat, joita voi valita ja käyttää osoitinlaitteella. Näitä ovat Button, NurbsSurf ja NurbsCurve.

Button

Hiirellä käytettävä painike, jota käytetään erilaisten toimintojen toteuttamiseen. Toteuttaa Selectable-rajapinnan.

3.2.4 Valmiit mallit

Sovelmassa on useita sisäänrakennettuja valmiita NURBS-malleja. Opetustilassa käytetään valmiita NURBS-malleja esimerkkeinä erityyppisistä kappaleista. Vapaissa muokkaustiloissa käyttäjä voi ladata haluamansa valmiin NURBS-mallin muokkaamista varten. Mallit on esitelty taulukossa 1. Malleista ympyrä ja sydän ovat suljettuja. Suljettu käyrä loppuu samaan pisteeseen kuin mistä se alkaa. Suljettu käyrä muodostetaan toistamalla käyrän lopussa käyrän alun kontrollipisteitä ja valitsemalla solmuvektori sopivasti. Pinnoista sylinteri on suljettu toisen parametrin suuntaan.

Taulukko 1. Sovelman sisäänrakennetut NURBS-mallit

Nimi	Kuvaus
Käyrät	
Suora	Suora käyrä. Oletuskappale vapaan käyrän muokkauksen tilassa.
Ympyrä	Sopivalla kontrollipisteiden, painokertoimien ja solmuvektorin valinnalla rakennettu ympyrä. Toimii esimerkkinä analyyttisestä NURBS-käyrästä.
Aalto	Jaksollinen aaltokäyrä.
Sydän	Vapaamuotoisesti piirretty sydän.
Spiraali	Aloitusruudussa näkyvä spiraalimainen kierre. Muista valmiista käyristä poiketen spiraali on kolmiulotteinen.
Piste	Yksittäinen kontrollipiste, jota voi käyttää lähtökohtana käyrän muodostuksessa.
Pinnat	
Tasainen	Tasainen pinta. Oletuskappale vapaan pinnan muokkauksen tilassa.
Aaltoileva	Aaltoileva kappale, esimerkki kumpuilevasta alustasta.
Sylinteri	Ympyrää toistamalla luotu sylinteri. Toimii esimerkkinä analyyttisestä NURBS-pinnasta.

Sovelmaan suunniteltiin myös mahdollisuutta omien NURBS-mallien tallentamiseen käyttäjän levyasemalle, mutta sitä ei toteutettu. Java-sovelma ei voi turvallisuussyistä normaalisti lukea käyttäjän levyasemaa, vaan sovelma ajetaan rajoitetussa ympäristössä. Tiedostojen lukemista varten sovelmaan on liitettävä sertifikaatti, jonka käyttäjä joutuu erikseen hyväksymään. Sovelman tarjoamat mahdollisuudet NURBS-mallien luomiseen ovat muutenkin hyvin rajalliset ja tallennusmahdollisuus olisi vaatinut lisätyötä sekä ohjelmoijalta että käyttäjiltä, minkä takia tallennustoimintoa ei toteutettu.

3.2.5 Testatut tietokoneet

Normaalin Java-sovelman suorituskyky ei riipu kovinkaan paljon näytönohjaimesta, ja sovelmat toimivat melko samalla tavalla eri laitteistoissa, mikäli muistia ja suoritintehoa on riittävästi. Kolmiulotteisen grafiikan esittämiseen tarvitaan kuitenkin käytännössä tuki näytönohjaimelta, ja myös näytönohjaimen ja käyttöjärjestelmän välillä toimivat ajurit voivat vaikuttaa merkittävästi sovelman suorituskykyyn ja yleiseen toimivuuteen. Reaaliaikainen vuorovaikutus käyttäjän kanssa tuo myös omat vaatimuksensa ohjelman suorituskyvylle. Näiden seikkojen takia pidettiin tärkeänä, että sovelman toimivuutta kokeillaan erilaisissa ympäristöissä. Sovelman toimintaa ja suorituskykyä testattiin eri tietokoneilla ja käyttöjärjestelmillä. Tulokset ovat taulukossa 2.

Taulukko 2. Sovelman kanssa testatut tietokoneet ja käyttöjärjestelmät

Suoritin / muisti / näytönohjain	Käyttöjärjestelmä	Sovelman toiminta
AMD Athlon 64 3200+ / 2 Gt / Nvidia GeForce 9500 GS	Ubuntu 8.10, Windows XP	Toimii ongelmitta
AMD Athlon 64 X2 TL-56 / 2 Gt / Nvidia GeForce 7200 Go	Ubuntu 8.10, Windows XP	Toimii ongelmitta
AMD Athlon XP 1.0 GHz / 1 Gt / Integroitu SIS 741	Ubuntu 8.10, Windows XP	Kaataa Javan
AMD Athlon XP 2800+ GHz / 1 Gt / Ati Radeon 9550	Debian 4.0	Hidas, 3D-kiihdytys puuttunee ajurista
Intel Core 2 6600 / 2 Gt / Nvidia Quadro FX 3450 SDI	Windows XP	Toimii ongelmitta
Intel Core 2 E8400 / 4 Gt / Ati Radeon HD 3450	Windows XP	Toimii ongelmitta
AMD Athlon 64 2800+ / 1 Gt / Matrox G400/G450	Ubuntu 8.10	Hidas, pahoja piirto-ongelmia.

Useimmilla testatuista järjestelmistä sovelma toimi täysin sujuvasti ja ongelmitta. Suoritinteho ja muistin määrä eivät olleet pullonkaulana millään laitteistolla, vaan sovelman toiminta riippui näytönohjaimesta. Yhdellä tietokoneella, jossa oli integroitu SIS 741

-näytönohjain, sovelmaa ei saatu toimimaan ollenkaan, vaan se kaatoi Javan riippumatta siitä, oliko käyttöjärjestelmänä Windows vai Linux. Yhdessä järjestelmässä sovelma toimi muuten moitteetta, mutta oli käytännössä liian hidas käytettäväksi, mikä luultavammin johtui tietokoneessa käytetyistä ajureista.

Integroitujen näytönohjainten 3D-ominaisuudet ovat usein varsin heikot, mikä myös selittänee sen, miksi Matroxin integroitu näytönohjain toimi huonosti ja SIS 741 -näytönohjainta ei saatu toimimaan ollenkaan sovelman kanssa. Yliopiston Linux-luokan 6325 tietokoneissa Radeon 9550 -näytönohjaimella sovelma toimi liian hitaasti ollakseen käytettävä. Radeon 9550:n teho luultavammin riittäisi kunnolliseen sovelman suorittamiseen, jos käytetäisiin 3D-kiihdytystä tukevia ajureita. Toimiessaan kunnolla sovelma näytti graafisesti lähes samalta kaikissa kokeilluissa ympäristöissä, ja ilmi tulleet häiritsevät eroavaisuudet saatiin pois pienellä sovelman hienosäädöllä.

3.3 Tulosten yhteenveto ja tulkinta

NURBS-sovelma toteutettiin onnistuneesti Javan ja Java OpenGL:n avulla. Sovelma on sekä havainnollinen että interaktiivinen, kuten oli tarkoituskin. Käyttäjää opastetaan NURBSien erilaisten ominaisuuksien kanssa, ja sovelma on jaettu aihealueittain. Käyttäjä voi halutessaan muotoilla omia NURBS-malleja tai muokata valmiita. Selaimessa toimiva sovelma on helposti saatavilla, kunhan käytettävissä on Internet-yhteys.

NURBS-sovelmassa päädyttiin käyttämään omaa NURBS-toteutusta, koska olemassa olleet ratkaisut eivät sopineet sovelman tarpeisiin. Useimmat NURBSien osa-alueet on kohtalaisen intuitiivisia ja helppoja ymmärtää ainakin periaatetasolla. Selvästi hankalin osa-alue on solmuvektorin toiminta, eli kuinka se vaikuttaa käytännössä NURBS-malliin. Tyypillisesti NURBS-mallinnuksessa ei kuitenkaan tarvita solmuvektorin syvällistä ymmärtämistä, vaan paino on enemmän kontrollipisteiden ja niiden painojen muokkaamisella. Solmuvektoreista tarvitsee yleensä tietää vain muutama perusasiaa, jotta NURBSeja voi hyödyntää kohtalaisen sujuvasti.

Projektin perusteella Java itsessään soveltuu hyvin tietokone- ja käyttöjärjestelmäriippumattomaan ohjelmointiin. Tämän sovelman kannalta pahin ongelmakohta oli JOGL, jonka tuki erilaisten tietokonejärjestelmien ja Java-toteutusten välillä vaihtelee paljon, mikä voi tuottaa joitain ongelmia suorituskyvyn ja yhteensopivuuden suhteen. JOGL:n pitäisi kuitenkin toimia hyvin, mikäli tietokoneen näytönohjain on kohtalaisen nykyaikainen ja sille on asennettu 3D-kiihdytystä tukevat ajurit. JOGL ei näytä nykyisellään toimivan kunnolla kuin Sun Microsystemsin virallisen Java-julkaisun kanssa. JOGL tarvitsee myös tietokoneeseen asennetut natiivikirjastot, mutta niiden lataamisen ja asentamisen saa automatisoitua hyvin, eikä se tuota kummempia ongelmia.

4 POHDINTA

Tässä luvussa pohditaan työssä käytettyjen ratkaisujen toimivuutta, saatuja tuloksia ja niiden merkitsevyyttä. Lisäksi mietitään ohjelman tulevaisuutta.

Projektin tutkimusongelmana oli luoda www-ympäristössä toimiva ohjelma, joka esittelee NURBSit intuitiivisesti ja havainnollisesti, ja siinä myös onnistuttiin. Parannettavaa kuitenkin olisi varmasti, ja olisi hyvä saada ohjelman varsinaisia käyttäjiä testaamaan sovelmaa ja esittämään parannusehdotuksia. Käyttäjätestaus ei kuitenkaan sopinut tämän kandidaatintyön aikatauluun ja työmäärään.

Yksi mahdollinen parannus sovelmaan olisi se, että käyttäjää ohjattaisiin enemmän. Nykyisellään sovelma luottaa siihen, että käyttäjä itse kokeilee mallin muokkaamista pyydettäessä ja painaa sen jälkeen seuraava-nappia. Tätä edistyneemmässä mallissa sovelma päästäisi käyttäjän automaattisesti eteenpäin silloin, kun käyttäjä on tehnyt pyydetyn asian.

Sovelma on muokattavissa, ja uusien aihealueiden lisääminen on mahdollista. Eräs pohdittu kehityssuunta sovelmalle olisi, että NURBSeihin, niiden käyttötarkoituksiin ja niiden teoriaan perehdyttäisiin syvällisemmin. Tällainen lähestymistapa kuitenkin paisuttaisi sovelmalta vaadittavaa ominaisuuslistaa paljon, ja toimisi varmaan paremmin verkkokurssin tapaisena sivustona, jossa olisi useita sovelmia, joista kukin havainnollistaisi yksittäistä osa-aluetta, kuten An Interactive Introduction to Splines-sivustolla [17]. NURBSit ovat kuitenkin vain yksi osa-alue Introduction to Computer Graphics -kurssia, ja sivuston kehittämiseen tarvittava työmäärä voisi olla turhan suuri kandidaatintyöksi.

Sovelman tarjoamat työkalut NURBSien muokkaamiseen ovat varsin yksinkertaisia, mikä rajoittaa niiden käyttömahdollisuuksia, mutta pitää myös käyttöliittymän yksinkertaisena. Työkalut kuitenkin riittävät yksinkertaisten perusideoiden ymmärtämiseen. Mikäli haluaa oppia NURBS-mallinnusta kunnolla, on myös varauduttava siihen, että pitää opetella käyttämään täysipainoista mallinnusohjelmistoa.

Työ osoitti, että Javan ja JOGL-rajapinnan avulla voidaan tehdä interaktiivisia www-selaimessa toimivia sovelmia, jotka hyödyntävät 3D-grafiikkaa. Mahdollisia tulevia projekteja ajatellen on kuitenkin hyvä miettiä, olisiko jokin muu alusta soveltunut paremmin. Www-selaimen käyttäjä on tottunut siihen, että selaimessa toimivat sovellukset toimivat joko suoraan sellaisenaan tai hyvin pienellä vaivalla, mikä rajoittaa käytettäviä tekniikoita hyvin paljon. Valitun tekniikan pitäisi olla yleisessä käytössä ja hyvin tuettu. Selaimessa toimivan sovelman alustalle käytännössä on kolme pääasiallista vaihtoehtoa, joista ensimmäinen on tässä työssä käytetty Java. Toinen vaihtoehto on JavaScript, joka on selaimessa ajettava komentosarjakieli. JavaScriptin etuihin kuuluu se, että nykyaikaisissa www-selaimissa on sisäänrakennettu JavaScript-tulkki, eikä sellaista tarvise asentaa erikseen. JavaScript kuitenkin soveltuu huonosti grafiikan luomiseen, ja etenkin 3D-grafiikan esittäminen on sillä hankalaa. Lisäksi JavaScriptin ominaisuuksien tuki vaihtelee selainkohtaisesti. Käytännössä JavaScript ei siis ole varteenotettava vaihtoehto. Kol-

mas vaihtoehto olisi Adoben kaupallinen Flash-alusta, jonka toisto-ohjelma on ilmainen. Flash on viime vuosina kasvattanut suosiotaan paljon ja useimmat selainpelit ja monet multimediasivustot perustuvat nykyään siihen. Flashille voi tehdä ohjelmia ActionScript-komentosarjakielen kanssa, ja Flashin uusimmassa 10-versiossa on tuki kolmiulotteiselle grafiikalle, mikä siitä on ennen puuttunut. Flash 10 saattaisikin olla siis toimiva ratkaisu tämän kaltaisessa projektissa, mutta asian tarkistamiseen vaadittaisiin tarkempaa perehtymistä. Ilmaisuus ja avoimuus kuitenkin puoltavat Javaa.

Projektin suurimmat ongelmat Javan kanssa liittyivät lähinnä JOGL-rajapintaan, eivätkä suoraan Javaan itseensä. Yksi JOGL:n ongelmista on natiivien kirjastojen tarve, minkä takia sovelma ei ole normaalisti suoraan käytettävissä verkkosivun auetessa, vaikka Java olisi jo asennettuna tietokoneessa. Toinen merkittävä JOGL:n ongelma on laitteistotuen vaihtelevuus. Tunnetuin vaihtoehto JOGL:lle olisi hyvin erilainen Java3D [24], joka on oliopohjainen ja perustuu renderoitavan näkymän esittämiseen graafina. Java3D:hen vaihtamalla menetettäisiin matalan tason grafiikan hallinta, mikä tekisi ohjelmoinnista erilaista. Java3D tarvitsee JOGL:n tavoin koneeseen asennetun ajonaikaisen kirjaston, eli siinä suhteessa sillä on samat ongelmat kuin JOGL:lla. Javalle on myös useita harrastajien luomia 3D-kirjastoja, mutta ne ovat yleensä peliorientoituneita. Näiden pohjalta mietittynä JOGL lienee nykytilanteessa ongelmistaan huolimatta paras Java-pohjainen ratkaisu tällaiseen projektiin, tai ei ainakaan muita vaihtoehtoja huonompi.

5 YHTEENVETO

NURBS on Bezier-käyrien ja B-splinien yleistys, jota käytetään yleisesti graafisessa mallinnuksessa ja monissa muissa tietokonesovelluksissa. NURBSeilla voidaan esittää sekä käyriä että pintoja, ja ne tarjoavat mahdollisuuksia sekä analyyttisten että vapaamuotoisten kappaleiden esittämiseen. NURBS-mallien ymmärtämistä helpottamaan tehtiin englanninkielinen interaktiivinen Java-sovelma Introduction to Computer Graphics -kurssille. Sovelmalle luotiin yksinkertainen www-sivu, joka sijoitettiin yliopiston palvelimelle.

Sovelman käyttöliittymän kaikki ominaisuudet ovat käytettävissä tavallisella kaksipainikkeisella hiirellä. Sovelma esittelee NURBS-käyrien ja -pintojen ominaisuuksia aihealueittain ja päästää käyttäjän muokkaamaan mallia. Sovelmaan ohjelmoitiin valmiita NURBS-malleja, joita käyttäjä pääsee muokkaamaan. Mahdollisuudet NURBS-mallien muokkaamiseen on pidetty yksinkertaisina, jotta sovelman totetus ja käyttöliittymä eivät tulisi liian monimutkaisiksi.

Kolmiulotteisen grafiikan piirtämiseen tarvittiin ulkoinen kirjasto, koska Javassa ei ole sellaista valmiina. Kirjastoksi valittiin JOGL. Java ja JOGL todettiin toimivaksi alustaksi www-selaimessa toimivan kolmiulotteisen grafiikan luontiin, mutta JOGL:n laitetuki voi olla huono etenkin vanhemmilla tietokoneilla tai käyttöjärjestelmillä. Lisäksi muut kuin Sun Microsystemsin oma virallinen Java-jakelu eivät tällä hetkellä näytä toimivan kunnolla JOGL:n kanssa. Sovelma toimii sekä Linux- että Windows-käyttöjärjestelmissä.

LÄHDELUETTELO

[1] Les Piegl, 1991. On NURBS: A Survey. *Computer Graphics and Applications, IEEE*, Vol. 11:55–71.

[2] Jaakko Tuosa, 2006. T-splinit. Pro gradu, Helsingin yliopisto.

[3] Don Norman, 10. 2008. Simplicity Is Not the Answer. *Interactions, ACM*, 5:45–46. Saatavilla: http://jnd.org/dn.mss/simplicity_is_not_the_answer.html. Viitattu 5.4.2009.

[4] Philip J. Schneider, 1996. NURB Curves: A Guide for the Uninitiated. *Develop, The Apple Technical Journal*, 25.

[5] F. S. Hill, Jr., 2001. *Computer Graphics Using Open GL*, Chapter 11: Curve and Surface Design, s. 597–671. Prentice Hall, Upper Saddle River, New Jersey, USA, toinen laitos.

[6] Les Piegl ja Wayne Tiller, 1995. *The NURBS Book*. Springer-Verlag, New York, NY, USA.

[7] Jim X. Chen ja Chunyang Chen, 2006. *Foundations of 3D Graphics Programming*, Curved Models, s. 191–226. Springer London.

[8] T.D. Barsky, B.A. DeRose, 1989. Geometric continuity of parametric curves: three equivalent characterizations. *Computer Graphics and Applications, IEEE*, 9(6):60–69.

[9] John Lowther John Fisher ja Ching-Kuang Shene, 2004. If you know B-splines well, you also know NURBS! Kirjassa *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, s. 343 – 347.

[10] Sun Microsystems, 2009. Java se downloads - sun developer network. Saatavilla: `http://java.sun.com/javase/downloads/index.jsp`. Viitattu 4.2.2009.

[11] Sun Microsystems, 2009. OpenJDK. Saatavilla: `http://openjdk.java.net/`. Viitattu 5.4.2009.

[12] Hong Zhang ja Y. Daniel Liang, 2007. *Computer Graphics Using Java 2D and 3D*. Pearson Prentice Hall, Upper Saddle River, New Jersey, USA.

[13] Adobe, 2009. Adobe® Flash® CS4 Professional. Saatavilla: `http://www.adobe.com/products/flash/`. Viitattu 5.2.2009.

[14] Sun Microsystems, 2009. Java bindings for OpenGL. Saatavilla: `https://jogl.dev.java.net/`. Viitattu 5.4.2009.

[15] Ilpo Vuorinen, 2001. *Tuhat tapaa opettaa: Menetelmäopas opettajille, kouluttajille ja ryhmän ohjaajille*. Kirjatalo Resurssi, Naantali, 6. laitos.

[16] Ben Shneiderman, 1998. *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*. Pearson Education.

[17] Evgeny Demidov, 2003. An Interactive Introduction to Splines. Verkkosivusto. Saatavilla: `http://www.people.nnov.ru/fractal/Splines/Intro.htm`. Viitattu 1.2.2009.

[18] Tomáš Hráský, 2008. SurfaceApp. Java OpenGL -rajapinnan esimerkkisovellus. Saatavilla: `https://jogl.dev.java.net/`. Viitattu 1.2.2009.

[19] University of Cambridge. NURBS demo. Advanced Graphics 2008–09 kurssiaineisto.
Saatavilla: `http://www.cl.cam.ac.uk/teaching/0809/AdvGraph/demos/Nurbs2d/index.html`. Viitattu 1.2.2009.

[20] Bob Fraser, 2007. Non-uniform Rational B-Spline (NURBS) Applet.
Saatavilla: `http://www.cs.uwaterloo.ca/~r3fraser/splines/asst4/asst4q4.html`. Viitattu 29.1.2009.

[21] Samuel Gerber, 2007. Jgeom: Geometry Library for Java.
Saatavilla: `https://jgeom.dev.java.net/`. Viitattu 1.2.2009.

[22] Eclipse Foundation, 2009. Eclipse.
Saatavilla: `http://www.eclipse.org/`. Viitattu 5.4.2009.

[23] Alan Dix et al., 2004. *Human-Computer Interaction*. Prentice Hall, Upper Saddle River, New Jersey, USA, kolmas laitos.

[24] Sun Microsystems, 2009. Master project for Java 3D projects.
Saatavilla: `https://java3d.dev.java.net/`. Viitattu 5.4.2009.

Liite 1. Vaatimusmäärittely

Vaatimusmäärittelyssä olleet ominaisuudet. Prioriteetti kertoo kuinka tärkeänä ominaisuuden toteuttamista on pidetty ja on välillä 5 (erittäin tärkeä) - 1 (ei kovinkaan tärkeä). Toteutumattomat ominaisuudet on merkitty sulkuihin.

Järjestelmä

Ominaisuus	Prioriteetti
Järjestelmä toimii selaimella.	5
Järjestelmä on toteutettu Javalla.	4
Järjestelmä käyttää Javan versiota 5.	3
Järjestelmässä hyödynnetään Java OpenGL -kirjastoa grafiikan tuottamiseen.	3
(Järjestelmässä käytetään valmiita OpenGL-kirjastofunktioita NURBS-mallien renderointiin.)	3
Järjestelmä toimii sekä Linuxissa että Windowsissa	4
(Järjestelmä on käytettävissä Linux-luokan 6325 koneilla.)	3
Järjestelmä on siirrettävissä.	3
Järjestelmä on asennettuna osaston palvelimella.	5
Järjestelmä on valmis viimeistään 30.4.2009.	3
Järjestelmä on helposti korjattavissa ja päivitettävissä.	1
Dokumentaatio on kattava sekä kirjallisena että ohjelmakoodin seassa.	2
Dokumentaatio on englanniksi.	3

NURBS-mallit

Ominaisuus	Prioriteetti
Järjestelmä tuottaa sekä NURBS-käyriä että -pintoja.	5
Käyrät ja pinnat ovat kolmiulotteisia.	5
Järjestelmä osaa piirtää käyrän eri segmentit eri väreillä.	3
Järjestelmä osaa piirtää mallin kantafunktiot graafisesti.	5
Järjestelmään on ohjelmoituna valmiita NURBS-malleja.	5

Käyttöliittymä

Ominaisuus	Prioriteetti
Järjestelmällä on interaktiivinen ja visuaalinen käyttöliittymä.	5
Käyttöliittymä on englanninkielinen.	5
Käyttöliittymä on yksinkertainen ja helppo käyttää.	3
Järjestelmässä on vapaa muokkaustila sekä käyrille että pinnoille.	4
Järjestelmässä on opetustila.	5
Kaikki opetustilan kannalta olennaiset toiminnot ovat käytettävissä hiirellä.	4
Opetustilassa järjestelmä esittelee NURBSien ominaisuuksia aihealueittain.	5
Opetustila antaa sanallisen kuvauksen esiteltävästä aihealueesta.	5
Opetustilassa on NURBSeja havainnollistavia animaatioita.	2
Opetustila on interaktiivinen.	4
Opetustilassa aihealueen kannalta epäolennaiset toiminnot ovat pois käytöstä.	2

Käyttäjän toiminnot

Ominaisuus	Prioriteetti
Käyttäjä voi liikuttaa NURBSin kontrollipisteitä.	5
Käyttäjä voi muuttaa NURBSin solmuvektoreita vapaasti.	5
Käyttäjä voi asettaa NURBSin solmuvektorit ennalta määrättyyn muotoon (tasavälinen, toistetut päädyt).	5
Käyttäjä voi muuttaa NURBSin astetta/kertalukua sekä u- että v-suuntaan.	5
Käyttäjä voi lisätä uusia kontrollipisteitä.	3
Käyttäjä voi poistaa kontrollipisteitä.	3
Käyttäjä voi vaihtaa opetustilassa käsiteltävää aihealuetta.	5
Käyttäjä voi vaihtaa opetustilasta vapaaseen muokkaustilaan ja päinvastoin.	5
Käyttäjä voi kääntää kuvakulmaa.	4
(Käyttäjä voi tallentaa NURBS-mallin tiedostoon ja ladata sen tiedostosta.)	2

Opetustilassa käsiteltävät aiheet

Ominaisuus	Prioriteetti
Opetustilassa on alkuesittely.	4
Opetustilassa käsitellään käyrän kantafunktioita.	5
Opetustilassa käsitellään käyrän astetta ja kertalukua.	5
Opetustilassa käsitellään käyrän solmuvektoria.	5
Opetustilassa käsitellään paikallista kontrollia.	5
Opetustilassa käsitellään NURBSin jatkuvuutta.	5
Opetustilassa käsitellään NURBSin painokertoimia.	5
Opetustilassa esitellään analyyttisia kappaleita.	3
Opetustilassa käsitellään pinnan kantafunktioita.	5
Opetustilassa käsitellään pinnan astetta ja kertalukua, sekä u- että v-suuntaan.	5
Opetustilassa käsitellään pinnan solmuvektoria, sekä u- että v-suuntaan.	5