SAVONIA

Piirtokomponentti web-käyttöliittymään

Julkinen versio

Mikko Korhonen

Opinnäytetyö

__. __. ____ _____

Ammattikorkeakoulututkinto

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä Mikko Korhonen	
Työn nimi Piirtokomponentti web-käyttöliittymään	
Päiväys 4.5.2012	Sivumäärä 33
Ohjaaja Lehtori Jussi Koistinen	
Toimeksiantaja Insinööri Ari-Pekka Raudaskoski	

Tiivistelmä

Tämän insinöörityön aiheena oli suunnitella piirtokomponentin toteutusta asiakkaan verkkosovellukseen. Sovellus koostui käyttöliittymästä ja tästä komponentista. Käyttäjän tuli voida suunnitella käyttöliittymän avulla kolmiulotteisia tuotteita ja piirtokomponentti näytti tuotteesta kuvan.

Projekti aloitettiin tutkimalla käytettävissä olevien tekniikoiden soveltavuutta projektin tarpeeseen. Valintaan vaikutti käytettävän tekniikan yhteensopivuus käyttöliittymän kanssa, kehittäjien aiempi kokemus tekniikasta sekä käytännöllisyys piirtotehtäviin.

Ensin suunniteltiin kynällä ja paperilla, miltä lopullinen kuva tulisi näyttämään. Kolmiulotteisen kuvan ajatuksesta luovuttiin sen vaikean toteutuksen vuoksi ja tilalle vaihdettiin kolme kaksiulotteista näkymää eri kuvakulmista. Jokainen näkymä sijoitettiin omalle alueelleen kuvassa ja näiden piirtoalueiden sisältöä alettiin tarkentaa.

Lopullinen piirtokomponentti toimi käyttöliittymän kanssa, mutta siinä oli puutteensa. Tuotantokäyttöön tarkoitettuna sovelluksena sitä pitäisi kehittää vielä paljon; tällä hetkellä toimintoja on liian vähän. Ohjelmakoodi kannattaa kirjoittaa uudestaan, mutta tästä ensimmäisestä versiosta voidaan oppia paljon.

Avainsanat
WWW, PHP, GD, grafiikka, suunnittelutyökalu

Luottamuksellisuus
julkinen

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author Mikko Korhonen			
Title of Thesis Drawing Component for Web Interface			
Date	4, March 2012	Pages	33
Supervisor Mr Jussi Koistinen, Lecturer			
Client Organisation Mr Ari-Pekka Raudaskoski, Engineer			

Abstract

The aim of this thesis was to design and produce a drawing component for a customer's web application. The application consisted of the interface and this component. A user was supposed to be able to design three dimensional products by this interface and the drawing component showed an image of the product.

The project was started by research on available technologies suitable for the needs of this project. The choice was affected by the compatibility between the chosen technology and the interface, developers' previous experience of the technology and practicality for drawing tasks.

First, the outcome of the final image was designed with a pen and paper. The idea of a three-dimentional image was abandoned for its difficult implementation and was replaced by three two-dimentional views from different angles. Each view was assigned to its own area in the image and the content of these drawing areas were defined more closely.

The final drawing component worked with the interface, but it had its flaws. As an application meant for production it should be developed further, at the moment it lacks some features. The code should be rewritten, but much can be learnt from this first version.

Keywords
WWW, PHP, GD, Graphics, Designing tool

Confidentiality
public

ALKUSANAT

Haluan kiittää työn tilaajaa Ari-Pekka Raudaskoskea mahdollisuudesta tehdä insinöörityöni ja muutenkin mukavasta yhteistyöstä. Lisäksi haluan kiittää työni ohjaajaa ja lukuisten kurssieni opettajaa lehtori Jussi Koistista saamastani ohjauksesta ja hyvästä opetuksesta sekä lehtoreita Maija Lötjästä ja Ulla Huttunen-Fintaa raportin kieliasun tarkastamisesta ja hyvistä neuvoista.

Kuopiossa 4.5.2012

Mikko Korhonen

SISÄLTÖ

1 JOHDANTO

Viimeisen vuosikymmenen aikana moni palvelu on siirtynyt Internetiin. Tämä on vapauttanut ihmisiä ajasta ja paikasta, kun palveluja on voinut käyttää juuri silloin, kun itse haluaa ja etäisyydellä palvelun sijaintiin ei ole ollut enää merkitystä. Lisäksi tekniikan kehitys on johtanut siihen, että uudenlaisia palveluita on mahdollista tarjota.

Asiakkaalla on liikeidea, jossa ihmisille tarjotaan mahdollisuutta suunnitella erikokoisia, -muotoisia ja haluamastaan materiaaleista olevia tuotteita. Suunnittelua varten käyttäjä tarvitsee WWW-pohjaisen käyttöliittymän. Käyttöliittymästä valitaan halutunlainen tuotetyyppi ja annetaan sille numeerisina arvoina mitat. Käyttäjä voi lisätä tuotteeseen haluamiinsa paikkoihin muotoiluja. Tuotteesta esitetään havainnollistava kuva kolmesta eri suunnasta ja tehdyt muokkaukset päivittyvät välittömästi kuvaan.

WWW-pohjainen suunnitteluohjelma koostuu kahdesta osasta: käyttöliittymästä ja kuvan muodostamisen hoitavasta piirtokomponentista. Opiskelija Simo Jokela toteutti opinnäytetyönään käyttöliittymän (Jokela, 2011) ja tässä työssä toteutettiin käyttöliittymässä käytettävä piirokomponentti.

Tämä on salaisen opinnäytetyön julkinen raportti.

2 TOTEUTUSTEKNIIKAT JA -VÄLINEET

Projektin alussa asiakas antoi suhteellisen vapaat kädet valita toteutustekniikan. Valintaan vaikuttivat esimerkiksi

- tekniikan tarjoamat mahdollisuudet
- yhteensopivuus käyttöliittymän kanssa
- kuormittavuus palvelimen ja asiakasohjelman välillä
- tekniikan tuttuus.

Asiakkaan toive oli, että piirrosta aiheutuva kuormitus tulisi pääasiassa asiakasohjelmalle. Näin yrityksen ei tarvitsisi panostaa niin paljoa palvelinkapasiteettiin. Tekniikoita olisivat mm. JavaScript ja Flash, sillä ne toimivat pääasiassa asiakasohjelman puolella. PHP (Personal Home Page) puolestaan muodostaa kuvan palvelimen puolella ja lähettää sen valmiina asiakasohjelmalle.

JavaScriptin ongelma on sen olemattomat piirtometodit. Tekniikka mahdollistaa kyllä omien piirtotoimintojen toteuttamisen, mutta se on erittäin epäkäytännöllinen keino, kun parempiakin ja valmiita tekniikoita on olemassa. Toinen JavaScriptin heikko kohta on sen hitaus, erityisesti piirrettäessä monimutkaisia kuvia itse tehdyillä piirtometodeilla.

Vähäisen JavaScript-kokemuksen ja huonosti tehdyn selvitystyön vuoksi jäi virheellinen mielikuva siitä, että JavaScriptille ei olisi valmiita piirtotoimintoja. Tämän mielikuvan valossa JavaScript ei vaikuttanut hyvältä vaihtoehdolta, koska tiedossa oli parempia, erityisesti piirtämiseen tarkoitettuja tekniikoita.

Flash on puolestaan kuuluisa näyttävän grafiikan piirtämisen ominaisuuksista. Tekijöiden kokemukset tekniikasta olivat tosin vähäiset ja yhteensopivuudesta käyttöliittymän kanssa ei ollut tietoa.

PHP:n melkein ainoaksi huonoksi puoleksi todettiin se, että se on hieman raskaampaa palvelimen kannalta. Sen sijaan PHP oli tuttu toteuttajille ja erityisesti runsaat piirto-ominaisuudet olivat tiedossa. PHP:llä muodostuvat kuvat tiedettiin voitavan liittää vaivatta käyttöliittymään valmiilla metodeilla ja lisäksi PHP:n yleisyyden vuoksi lisätietoa oli aina saatavilla.

Muita mahdollisia tekniikoita olisivat olleet esimerkiksi ASP (Active Server Pages) ja JSP (JavaServerPages). Kokemus näistä tekniikoista oli kuitenkin olematonta, joten paremmaksi vaihtoehdoksi nähtiin käyttää jo tuttua tekniikkaa.

Puntaroitaessa toteutustekniikoita päädyttiin asiakkaan alkuperäisestä toiveesta huolimatta PHP:hen. Kyseisen tekniikan mahdollisuudet voittivat ylivoimaisesti sen ainoan huonon puolen eli palvelimen kuormituksen. Todennäköisesti palvelimen kuormitus ei olisi kuitenkaan niin suuri, että siitä aiheutuisi ongelmia.

2.1 JavaScript

JavaScript on alun perin Netscape Communications Corporation -yrityksen kehittämä komentosarjakieli WWW-sivuja varten. JavaScript mahdollistaa dynaamisia toimintoja WWW-selaimessa eikä vaadi erillistä palvelinta kuten moni muu dynaamisuuden mahdollistava tekniikka. (Negrino 2007.)

Nykyinen JavaScript perustuu DOM-malliin (Document Object Model) eli se on oliokieli. Javascript tulkitsee mm. selainikkunan, näytön ja varsinaisen verkkosivun olioina, joiden ominaisuuksien ja metodien avulla dynaamisuus luodaan. (Negrino, 2007)

Esimerkki JavaScript-koodista:

```
<script type="text/javascript">
document.write("Hei, maailma!");
</script>
```

Esimerkki tulostaa HTML -sivun dokumenttiosaan tekstin “Hei, maailma!”.

2.2 Adobe Flash

Flash on Adobe Systemsin kehittämä ympäristö multimediasisällön tuottamiseen yleensä WWW-sivuille, joskin kyseistä tekniikkaa on mahdollista käyttää myös muissakin ympäristöissä. Flashia käytetään usein animaatioiden tuottamiseen esimerkiksi mainoskäyttöön, mutta sisäänrakennettu komentosarjakieli ActionScript mahdollistaa myös erittäin monipuolisten interaktiivisten pelien toteutuksen. (Adobe.)

Flashia on sekä kehuttu että moitittu. Sillä saa toteutettua erittäin näyttäviä interaktiivisia multimediaesityksiä. Eräänä esimerkkinä Flash-esityksestä on monien verkkokauppojen käyttämät sähköiset mainoskuvastot, jotka vastaavat hyvin paljon painettuja kuvastoja.

Eräs kritiikin aihe on Flash-esitysten erittäin heikko käytettävyys. Flash-elementti toimii yhtenä graafisena elementtinä, johon ei sisälly ollenkaan olennaisia metatietoja. Tämä aiheuttaa suuria vaikeuksia esimerkiksi näkörajoitteisille käyttäjille tai niille, joiden verkkoselaimessa ei ole käytössä Flashin liitännäistä.

Flash–esitysten katsominen on mahdollista Adobe Flash Player -liitännäisen avulla, jonka saa ladata ilmaiseksi käytännössä jokaiselle tunnetulle käyttöjärjestelmille ja verkkoselaimille.

2.3 PHP ja GD-kirjasto

PHP on alun perin Rasmus Lerdorfin vuonna 1995 kehittämä C-kieleen pohjautuva skriptikieli. Ensimmäinen PHP:n versio osasi vain kirjata verkkosivulla vierailut ja näyttää niiden lukumäärän. Muiden kehittäjien suuri kiinnostus sai hänet lisäämään PHP:hen ominaisuuksia ja pian kehittäjäjoukko laajeni. Vuonna 1997 PHP:n lyhenne vaihdettiin tarkoittamaan sanoja PHP: Hypertext Preprocessor. Vuonna 1998 PHP:tä käytettiin jo 50 000 WWW-sivulla ja 1999 rikottiin miljoonan sivun rajan. (Gilmore 2005, 1.)

Vaatimusten kasvaessa kaksi ydinkehittäjää eli Zeez Suraski ja Andi Gutmans kirjoittivat PHP:n jäsentimen kokonaan uusiksi, josta tuli PHP:n versio 4. Se toi mukanaan parannuksia, joiden vuoksi PHP:stä tuli vakavasti harkittava ratkaisu yritystason sivustoillekin. Näitä ominaisuuksia olivat mm. parannettu resurssien hallinta, osittainen oliopohjaisuus, istuntojen hallinta, salakirjoitus ja Java-tuki. Näiden lisäksi PHP:hen lisättiin satoja metodeita, jotka laajensivat entisestään kielen tehoa. (Gilmore 2005, 2.)

PHP 4:ssä oli erinomaisuudestaan huolimatta runsaasti puutteita, erityisesti olio-ohjelmoinnin toteutuksessa. Tästä syystä kehitettiin versio 5, jonka parannuksina olivat mm. kehittyneemmät olio-ohjelmoinnin työkalut, try/catch–poikkeusten hallinta ja kehittyneempi merkkijonojen hallinta. (Gilmore 2005, 2.)

Esimerkki PHP–koodista:

```
<?php
$muuttuja = ”Hei, maailma!”;
echo $muuttuja;
?>
```

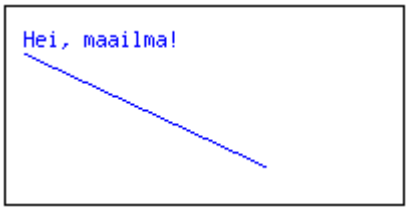
Esimerkki tulostaa HTML-sivulle tekstin “Hei, maailma!”.

GD–kirjasto on avoimen lähdekoodin C-kieleen pohjautuva ohjelma grafiikan luomiseen ja muokkaamiseen. Kirjasto sisältyy nykyiseen PHP-tulkkiin, joten yhteensopivuus PHP:llä luotuihin verkkosivuihin on taattu. GD-kirjasto sisältää laajat valikoimat grafiikkametodeita, joita yhdistelemällä voidaan toteuttaa monimutkaisiakin toimintoja (Image Processing and GD, 2012).

Esimerkki GD–koodista:

```
<?php

header("Content-type: image/png");

$kuva=imagecreate(200,100);

$tausta=imagecolorallocate($kuva, 255, 255, 255);
$musta=imagecolorallocate($kuva,0,0,0);
$sininen=imagecolorallocate($kuva,0,0,255);

imagerectangle($kuva,0,0,199,99,$musta);
imagestring($kuva,2,10,10,"Hei, maailma!",$sininen);
imageline($kuva,10,24,130,80,$sininen);

imagepng($kuva);
imagedestroy($kuva);

?>
```

Esimerkki muodostaa laatikon, jossa on viiva ja teksti “Hei, maailma!” (kuvio 1).

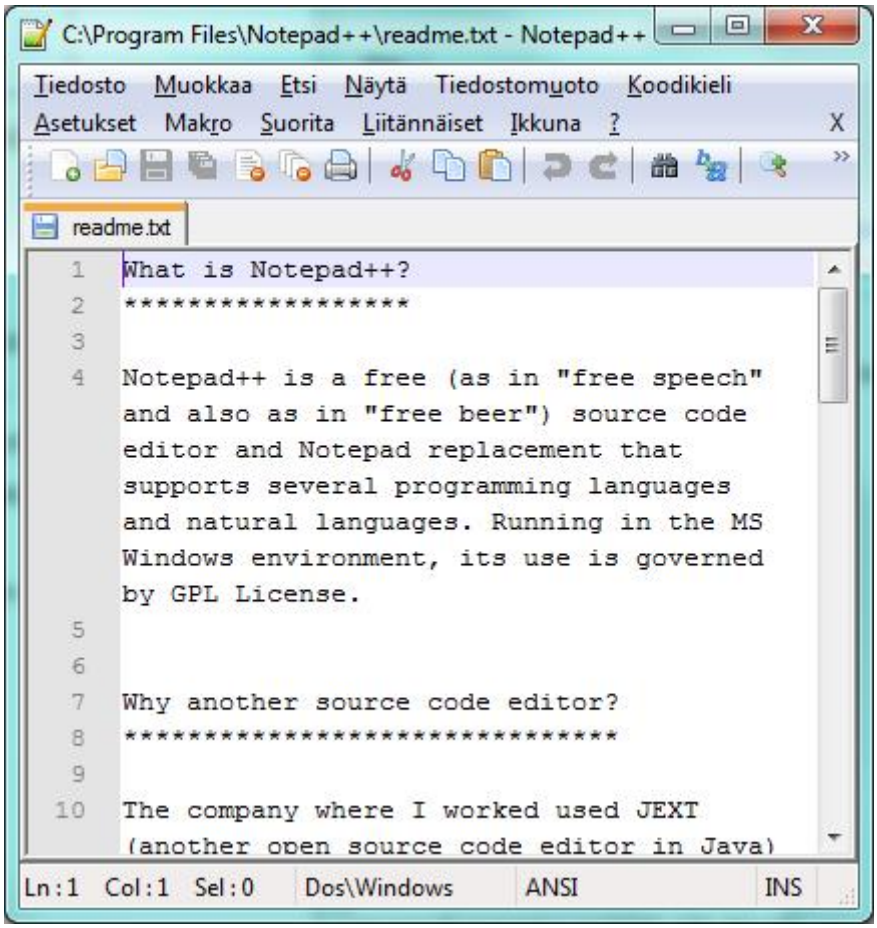


KUVIO 1. GD-kirjastolla muodostettu kuva

2.4 Notepad++ v.5.9.8

PHP:n valinta käytettäväksi tekniikaksi on edullinen vaihtoehto siinä mielessä, että se ei vaadi kalliita ohjelmistoja. Ohjelmointikoodin kirjoitus onnistuu millä tahansa tekstinmuokkausohjelmalla ja kyse on enemmänkin siitä, mikä ohjelma miellyttää itseä eniten. Tärkeimmät ominaisuudet löytyvät kuitenkin suuresta osasta ohjelmia.

Käytettäväksi valittiin Notepad++ -ohjelma, koska siitä oli myönteisiä kokemuksia entuudestaan (Notepad++). Eräs hyödyllinen ominaisuus ohjelmassa on monipuoliset merkkijonojen korvaustoiminnot. Jos sama merkkijono toistuu useassa kohtaa tiedostoissa, voi nämä korvata helposti toisilla merkkijonoilla tai erikoismerkeillä.



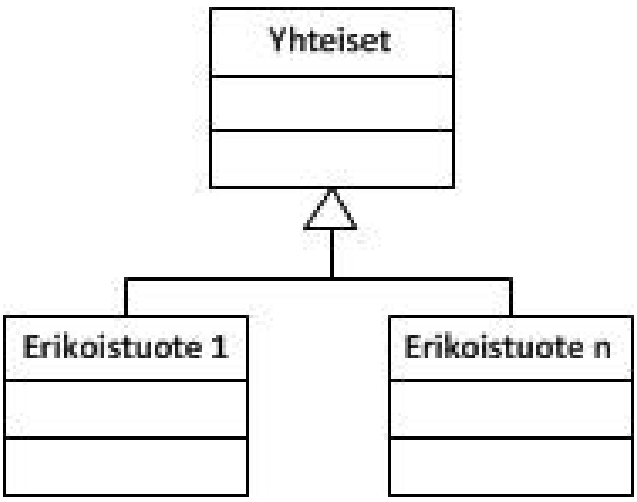
KUVIO 2. Notepad++

3 PIIRTOKOMPONENTIN RAKENNE JA SUUNNITTELU

Komponentin huolellinen suunnittelu helpottaa kehitystä nyt ja myöhemmässä vaiheessa. Huonosti suunniteltu ohjelmakoodi saattaa johtaa siihen, että kokonaisia osia ohjelmasta joudutaan toteuttamaan uudestaan, kun jatkokehitystä ei olla huomioitu ajoissa. Komponentin suunnitteluun käytettiin Microsoft Office Visio 2007 -nimistä ohjelmaa.

3.1 Luokkarakenne

Komponentin toiminnallisuus on jaettu eri luokkiin, koska se on tarkoitettu jatkossa laajennettavaksi. Yhteiset-luokka sisältää kaikkien tuoteryhmien yhteiset piirtotoiminnot, kuten piirtoalueen ja väreihin ja tekstuureihin liittyvät toiminnot. Jokaisella eri tuoteryhmällä on oma luokkansa, jotka perivät Yhteiset-luokan ja täydentää sitä omilla ominaisuuksillaan ja toiminnoillaan. Tässä projektissa on vain yksi tuoteryhmä, joka on suorakulmaisen särmiön muotoinen kappale.



KUVIO 3. Tuotteiden hierarkia

3.2 Muuttujat

Kuvaluokka sisältää paljon muuttujia tiedon tallennukseen. Muuttujat on määritelty protected-näkyvyydellä, eli niihin ei voi viitata suoraan luokan ulkopuolelta. Tiedot ovat kuitenkin käytettävissä kaikkialta luokan sisältä.

3.3 Metodit

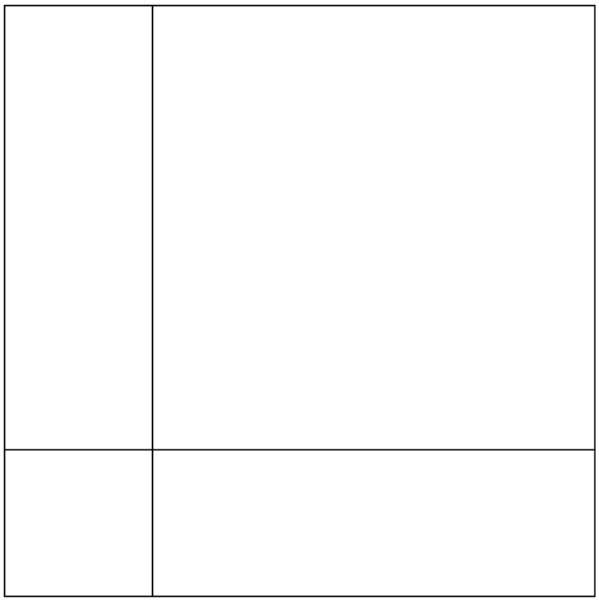
Metodien yhtenä käyttötarkoituksena on kirjoittaa usein tarvittava toiminnallisuus yhteen paikkaan, josta tätä voidaan tarvittaessa kutsua. Toinen käyttötapa on kapseloida ohjelmakoodia pienempiin, helposti ymmärrettäviin ja ylläpidettäviin kokonaisuuksiin. Tämän projektin komponentissa ei ole suoraviivaisen toimintalogiikan vuoksi montaa usein tarvittavaa toimintoja, mutta ymmärrettävyyden vuoksi kapseloivia metodeja olisi voinut käyttää enemmänkin.

3.4 Kuvien sijoittelu

Koko kuvakomponentti koostuu kuudesta piirtoalueesta: ylänäkymästä, sivunäkymästä, päätynäkymästä, pituuden mittajanasta, leveyden mittajanasta ja oikeaan alareunaan jäävästä tyhjästä alueesta, jonne voi tulevaisuudessa lisätä muotoiluja kuvastavia tarkempia kuvia. Komponentin jakamisella erillisiin piirtoalueisiin on se etu, että kuvan osat pysyvät paremmin hallitulla alueella ja kuvasta tulee muutenkin esteettisempi. Piirtoalueet on kehystetty reunaviivoilla, jotta komponentin tuottama kuva erottuu paremmin.

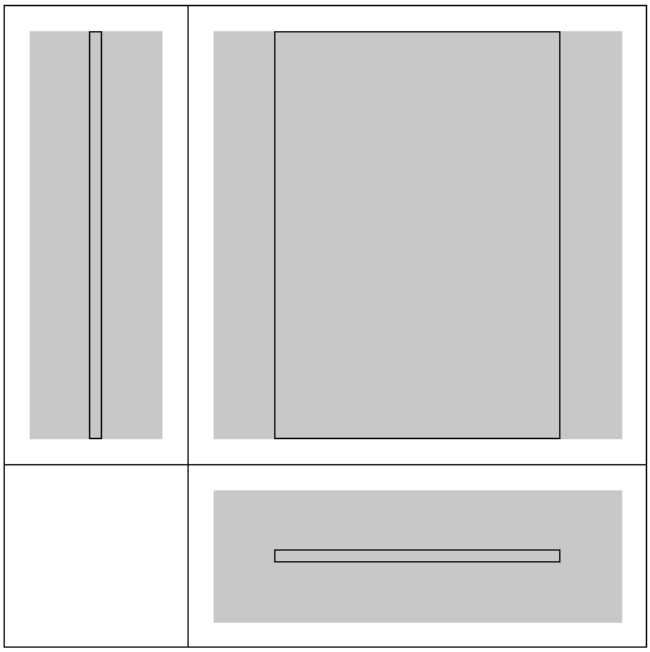
Alapuolella on havainnollistettu elementtien sijoittumisesta kuvassa. Samalla kuvioista näkee millaisia vaiheita ja muutoksia komponentissa tapahtui projektin edetessä.

Kuviossa 4 piirrettävä kuva on jaettu neljään eri alueeseen. Tärkein kuva eli ylänäkymä on oikeassa yläkulmassa. Vasemmassa yläkulmassa on sivunäkymä ja oikeassa alakulmassa päätynäkymä. Oikeaan alakulmaan suunniteltiin kuvaa kulmapyöristyksistä.



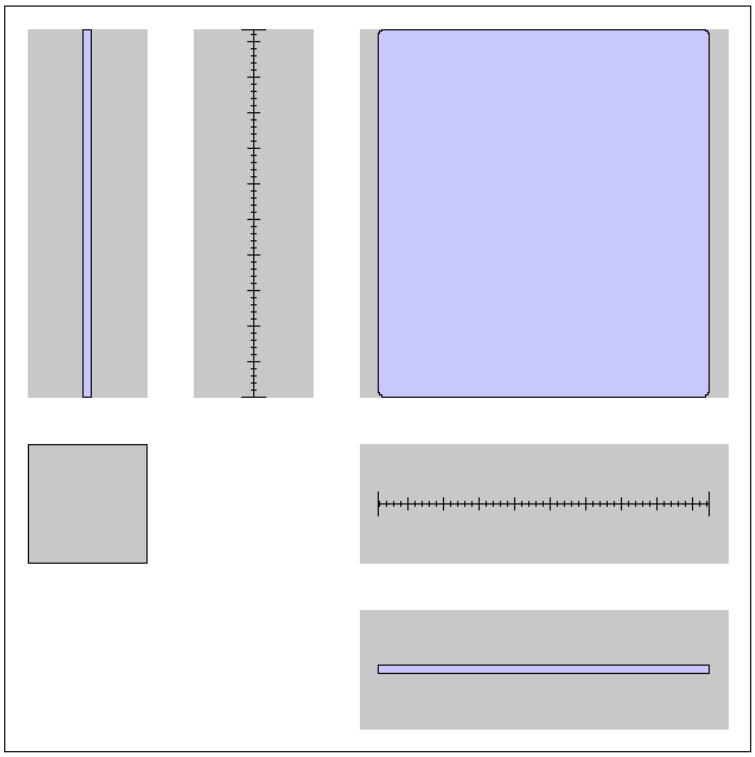
KUVIO 4. Piirtoalueiden sijoittelu

Kuviossa 5 nähdään jaetuille alueille sijoitetut piirtoalueet. Näille piirtoalueille tapahtuu tuotteen piirtäminen.



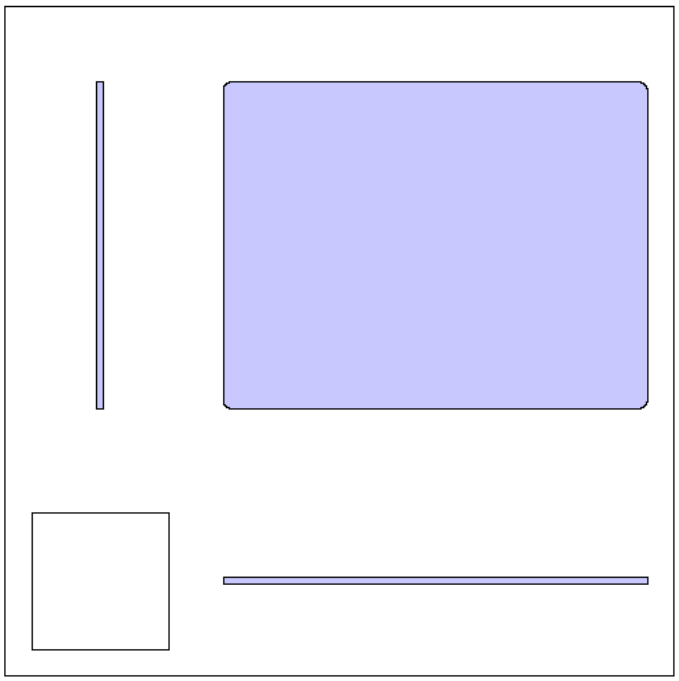
KUVIO 5. Kappaleiden sijoittuminen piirtoalueille

Kuvioon 6 on lisätty ensimmäiset versiot mittajanoista ja niissä olevat mitta-asteikot.



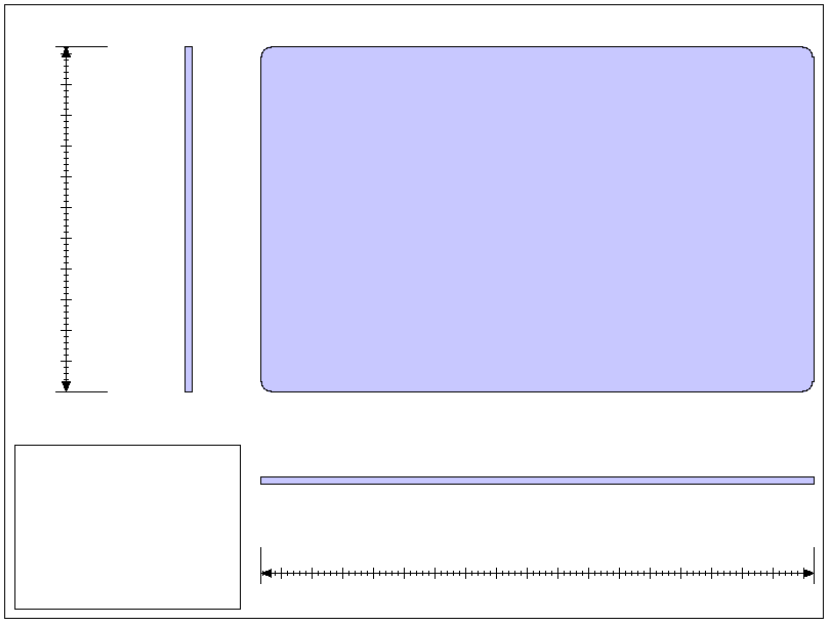
KUVIO 6. Ensimmäinen versio mittajanoista

Kuviossa 7 sekä piirtoalueet että mittajanat on piilotettu.



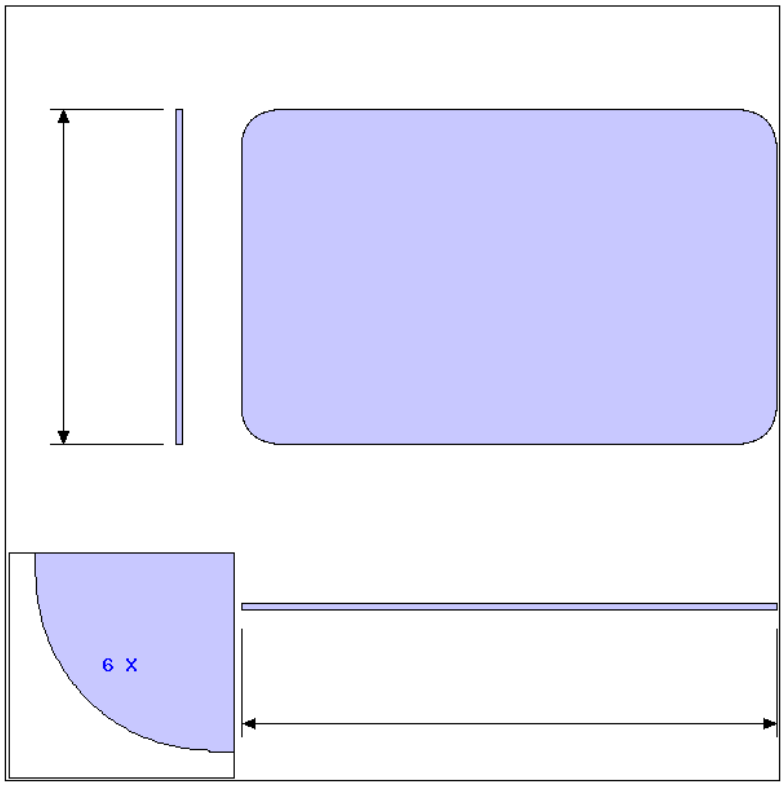
KUVIO 7. Piirtoalueet piilotettuina

Kuviossa 8 mittajanat on siirretty uuteen paikkaan.



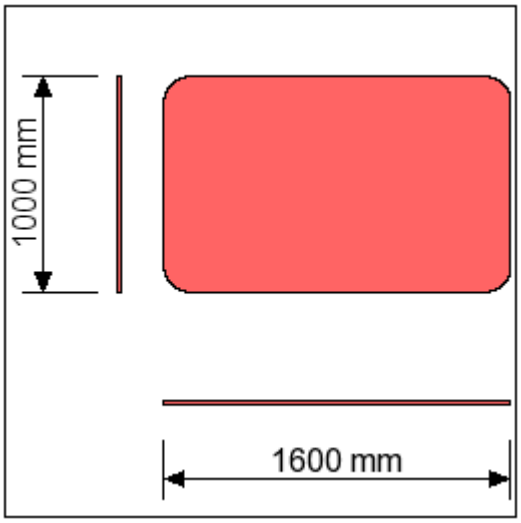
KUVIO 8. Mittajanan koekäyttöä

Kuviossa 9 mittajanoja on siistitty poistamalla pienet mittaviivat ja kulmapyöristyksen kuvan on toteutettu oikeaan alakulmaan.



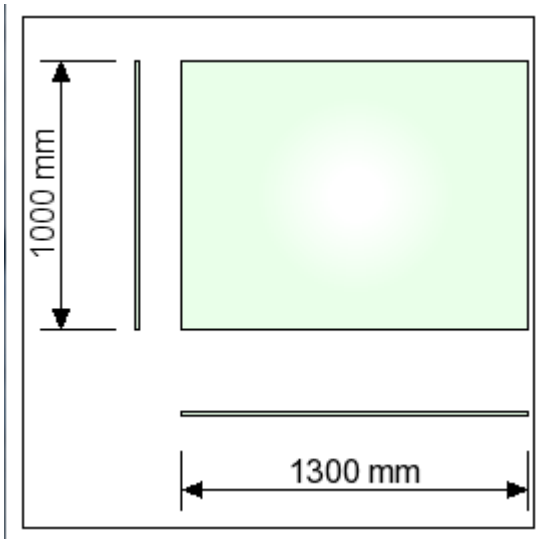
KUVIO 9. Kulmanäkymän testausta

Kuviossa 10 kulmapyöristyksen kuva on poistettu vähäisen lisäinformaation vuoksi ja mittajanoihin on lisätty numeeriset arvot.



KUVIO 10. Mittojen testausta

Kuviossa 11 on testattu graafisen ilmeen lisäämistä tuotteen pintaan.



KUVIO 11. Erilaisten tekstuurien testausta

Kuviossa 12 on testattu tekstuurien käyttöä kuvassa. Kokeilu jäi kuitenkin kesken, mutta kokeilua voidaan käyttää ehkä tulevissa versioissa.

KUVIO 12. Piirtokomponentti tekstuurin kanssa

3.5 Näkymä ylhäältä, sivusta ja päädystä

Oikeaan yläkulmaan sijoitetussa ylänäkymässä piirretään kuva tuotteesta niin kutsutusti ylhäältä katsottuna. Ylänäkymää voidaan pitää myös tärkeimpänä kolmesta näkymästä, sillä sen perusteella määritetään piirrettävän tuotteen mittasuhde. Tuotteen pituutta ja leveyttä verrataan ylänäkymän piirtoalueen pituuteen ja leveyteen. Tämän jälkeen voidaan piirtää tuote mahdollisimman isona, mutta kuitenkin niin, ettei se ylitä piirtoalueen rajoja. Samalla voidaan laskea mittasuhdekerroin, jolla muutkin näkymät piirretään. Kuvan piirrossa käytetään mittasuhteen lisäksi tuotteen pituutta ja leveyttä.

Ylänäkymän vasemmalle puolelle sijoitetussa sivunäkymässä piirretään kuva tuotteesta niin kutsutusti sivusta katsottuna. Näkymässä erottuu tuotteen pituus ja paksuus, ja se piirretään samassa mittakaavassa kuin ylänäkymän kuva.

Ylänäkymän alapuolelle sijoitetussa päätynäkymässä piirretään kuva tuotteesta niin kutsutusti päädystä katsottuna. Näkymässä erottuu tuotteen leveys ja paksuus ja se piirretään samassa mittakaavassa kuin ylänäkymän kuvan.

3.6 Pituuden ja leveyden mittajana

Sivunäkymän vasemmalla puolelle piirretään mittajana, joka ilmaisee tuotteen pituuden. Mittajanan vasemmalle puolelle tulostuu tekstimuodossa tuotteen pituus. Päätynäkymän alapuolelle piirretään mittajana, joka ilmaisee tuotteen leveyden. Mittajanan yläpuolelle tulostuu tekstimuodossa tuotteen leveys.

Molempien mittajanojen tarkoitus on havainnollistaa tuotteen mittoja. Muista näkymistä kokoa ei voi päätellä, sillä niissä piirrettävän kappaleen koko suhteutetaan piirtoalueelle, jotta siitä saisi mahdollisimman tarkan kuvan.

4 PIIRTOKOMPONENTIN LOGIIKKA

Tuotteesta luotavan kuvan koostaminen on melko suoraviivainen prosessi. Käyttäjän syötettyä haluamansa mitat ja lisäominaisuudet piirtokomponentti laskee näiden tietojen perusteella lukuisia laskelmia ja piirtää kuvat paikoilleen oikeassa mittakaavassa.

4.1 Muuttujien esittely

Aivan piirtoluokan alussa esitellään komponentin sisäisessä käytössä olevat muuttujat:

1. kuvaolio
2. sivunäkymän korkeus ja leveys
3. ylänäkymän korkeus ja leveys
4. päätynäkymän korkeus ja leveys
5. mittajanojen korkeudet ja leveydet
6. tuotteen pituus, leveys ja paksuus
7. Tuotteen muut ominaisuudet
8. piirtoväri kahdessa eri muodossa
9. marginaalin mitta
10. piirtosuhdeluku
11. piirtoalueiden reunojen koordinaatit
12. piirtoalueiden korkeudet ja leveydet
13. tuotteen suhteellinen pituus, leveys ja paksuus.

4.2 Tuotteen ja piirtoalueen tietojen välittäminen muuttujiin

Komponentin muodostinmetodi saa joukon parametreja, joiden sisältö otetaan talteen olion muuttujille. Parametrit ovat järjestyksessä:

1. kuvan koko (pikseleinä)
2. kuvan leveys (pikseleinä)
3. tuotteen pituus (millimetreissä)
4. tuotteen leveys (millimetreissä)
5. tuotteen paksuus (millimetreissä)
6. tuotteen muut ominaisuudet
7. piirtoväri (arvo 0-3).

Seuraavassa kuviossa 14 suoritetaan laskutoimituksia ja sijoitetaan arvoja muuttujiin.

KUVIO 14. Arvojen tallennus

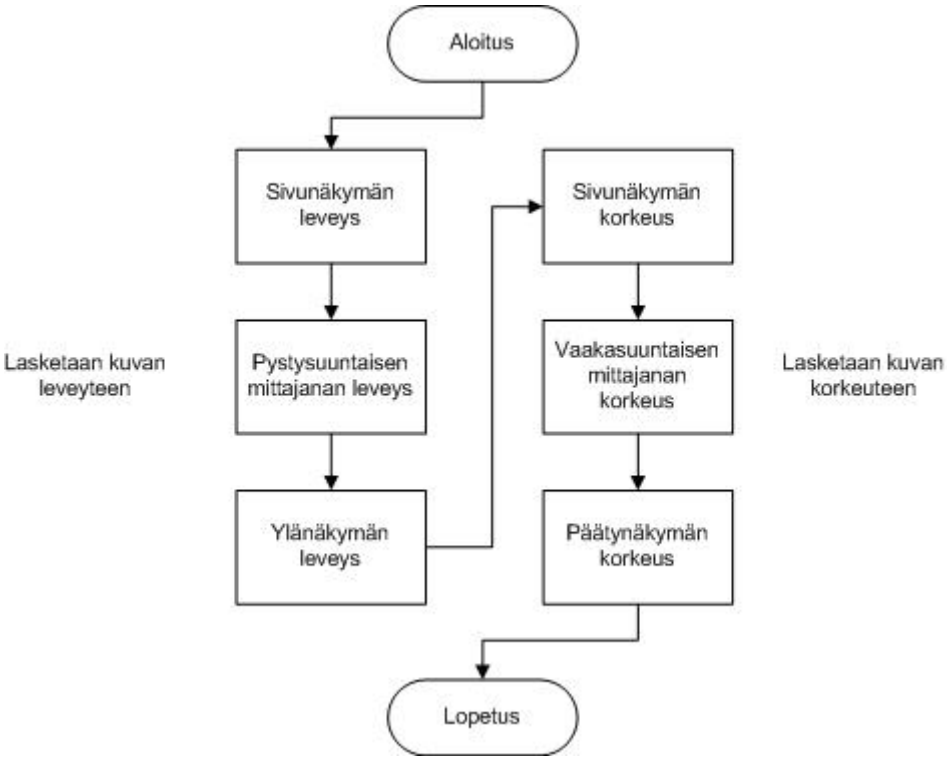
4.3 Piirtometodin suoritus

Komponentissa on vain yksi suoritettava metodi ja sen tehtävä on muodostaa kuva käyttäjän antamien tietojen avulla. Kuva muodostuu itse tuotteesta ja sen ympärillä olevista kehyksistä ja mittajanoista. Metodin vaiheet ovat:

1. luodaan piirto-olio, johon kaikki piirtämiset liitetään
2. määritellään piirtovärit
3. tyhjennetään koko kuva
4. lasketaan tuotteen piirtosuhde
5. piirretään tuote ylänäkymään
6. piirretään tuote sivunäkymään
7. piirretään tuote päätynäkymään
8. piirretään mittajanat ja viereen mitat
9. palautetaan piirto-olio, josta muodostetaan Jpeg –kuva.

4.4 Piirto-olion luominen

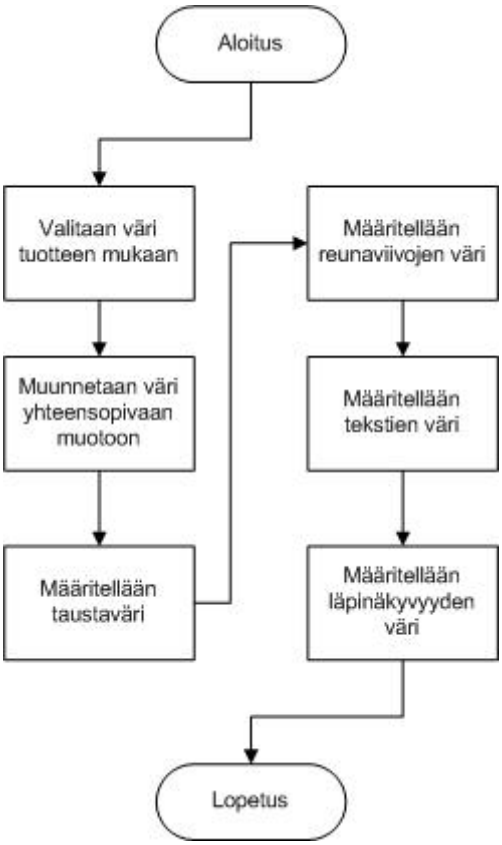
Piirto-olion muodostetaan itse toteutetulla *imagecreate*-metodilla. Metodi tarvitsee syöttötietoinaan muodostettavan kuvan leveyden ja korkeuden. Mitat lasketaan dynaamisesti muiden piirtoalueiden arvoista. Muodostettavan kuvan koko vaakasuunnassa saadaan laskemalla yhteen sivunäkymän, pystysuunnassa olevan mittajanan sekä ylänäkymän piirtoalueiden koko sivusuunnassa. Muodostettavan kuvan koko pystysuunnassa saadaan laskemalla yhteen sivunäkymän, sivusuunnassa olevan mittajanan sekä päätynäkymän piirtoalueiden koko pystysuunnassa. Metodin palauttaman piirto-olion tallennetaan *$image* –muuttujaan, jota melkein jokainen tämän komponentin kuvametodi käyttää.



KUVIO 15. Piirto-olion luominen

4.4.1 Piirtovärien määritys

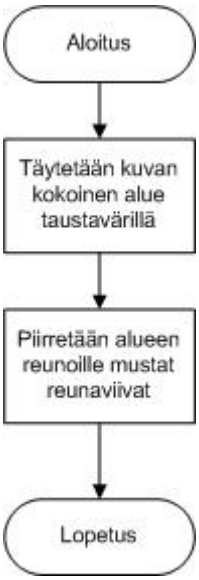
Värien määrittely on olennainen vaihe komponentissa, koska kaikki piirtotoiminnot vaativat jonkin värin. Värit tallennetaan omaan resurssimuuttujaansa ja yhdistetään kuvan resurssimuuttujaan. Määritys tehdään metodilla *imagecolorallocate($kuvaresurssi, $R, $G, $B)*, jossa $kuvaresurssi on kuva, johon väri liitetään ja $R, $G ja $B ovat värin RGB-arvot (0-255). Käyttäjä ei valitse tuotteen väriä, vaan se riippuu käytetystä materiaalista.



KUVIO 16. Värien määrittäminen

4.4.2 Kuvan tyhjentäminen

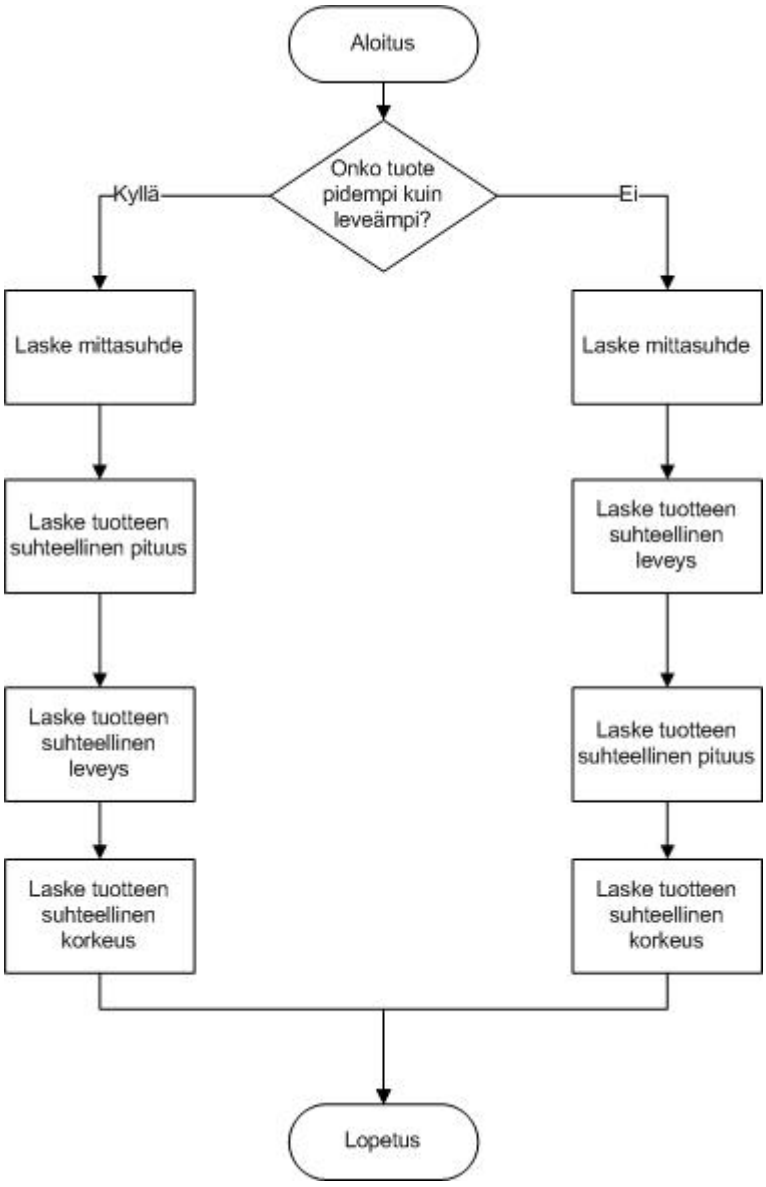
Ennen varsinaista kuvaan piirtämistä se on tyhjennettävä. Tämä tapahtuu piirtämällä siihen ensiksi kuvan kokoisen suorakaiteen muotoisen täytetyn alueen taustavärillä. Tämän jälkeen reunoihin piirretään yhden pikselin paksuiset reunaviivat. Nyt piirtoalue on alustettu ja sille voidaan alkaa piirtämään.



KUVIO 17. Kuvan tyhjentäminen

4.4.3 Tuotteen piirtosuhteen laskeminen

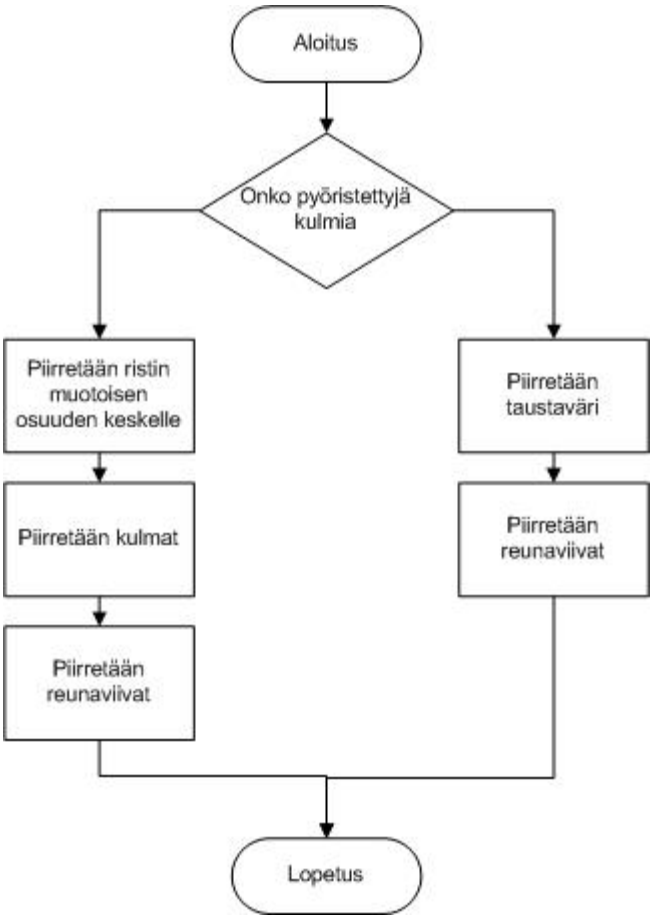
Piirrettävä kappale saattaa olla minkä kokoinen tahansa, mutta piirtoalue on kuitenkin rajallinen. Tuotetta joudutaan kutistamaan sen verran, jotta se mahtuu piirtoalueelle. Tämän kutistussuhteen määrä lasketaan vertaamalla kappaleen pituutta ja leveyttä piirtoalueen pituuteen ja leveyteen. Tämän jälkeen kappaleen voidaan piirtää sopivassa mittasuhteessa.



KUVIO 18. Piirtosuhteen laskeminen

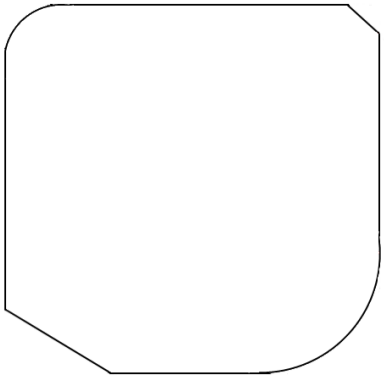
4.4.4 Ylänäkymän piirtäminen

Suorakulmaisen särmiön muotoista kappaletta muodostaessa riittää, että sille varatulla metodilla piirretään suorakulmio kappaleen pituuden ja leveyden avulla. Jos kuitenkin yhdessäkin kulmassa on muotoiluja, käytetään toista metodia. Tällä metodilla ensin piirretään keskelle ristin muotoinen alue ja sen jälkeen jokaiseen kulmaan pyöristys tai viiste ja täytetään ylimääräiset alueet suorakulmioilla.



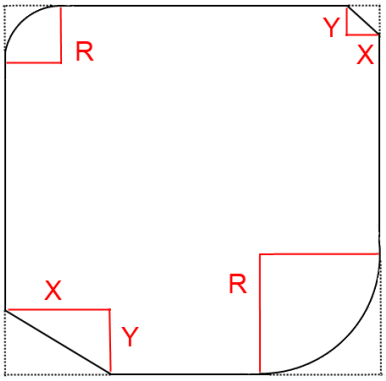
KUVIO 19. Ylänäkymän piirtäminen

Seuraavaksi kuvataan edellä mainittu kulmien muotoilu. Kuviossa 20 on esitetty, millaisiin muotoiluihin esimerkissä pyritään: Kahdessa vastakkaisessa kulmassa on pyöristykset ja kahdessa jäljelle jääneessä ovat viisteet.



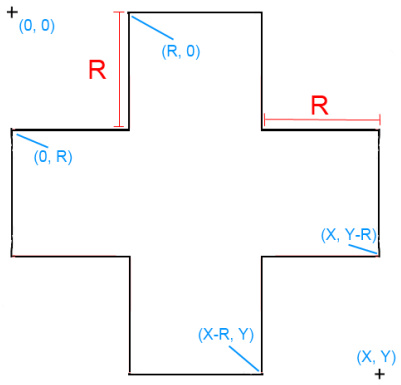
KUVIO 20. Kulmamuotoilut

Kuviossa 21 kaikista kulmista leikataan palat pois. Pyöristyksissä leikattavan palan koko on pyöristyksen säde R ja viisteissä korkeus ja leveys X ja Y. Leikattujen palojen arvot säilytetään kuitenkin tallessa, koska niitä käytetään myöhemmin.



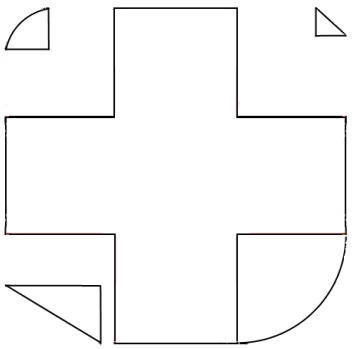
KUVIO 21. Kulmien leikkaaminen

Kuviossa 22 jokaisesta kulmasta on leikattu pois yhtä suuri pala. Palan koko on suurin arvo pyöristyksen säteistä tai viisteiden korkeuksista tai leveyksistä. Jäljelle jää ristin muotoinen kappale.



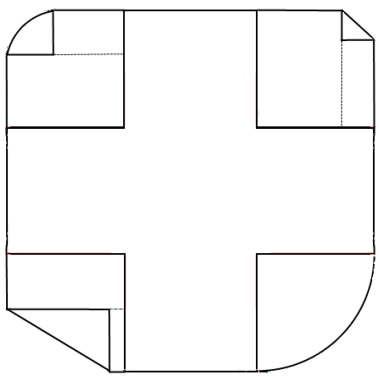
KUVIO 22. Ristin muodostaminen

Kuviossa 23 aiemmin poistetut kulmapalat palautetaan takaisin paikoilleen. Kuvioon jää kuitenkin useita tyhjiä alueita kappaleen riippuen siitä, millaiset kulmamuotoilut käyttäjä on määritellyt.



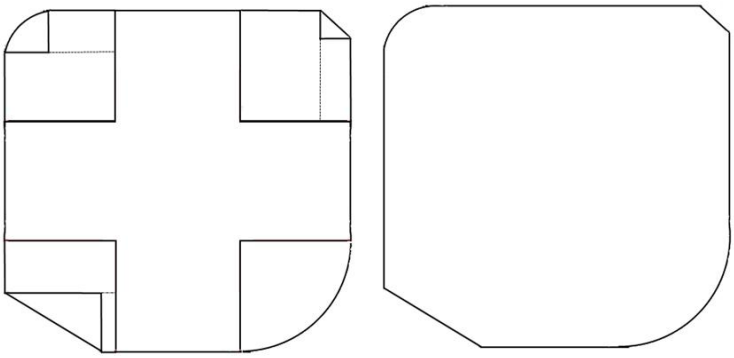
KUVIO 23. Kulmien palauttaminen

Kuviossa 24 kappaleen keskellä olevat tyhjät aukot on täytetty suorakaiteen muotoisilla alueilla. Lopuksi kappaleen reunustetaan reunaviivan värillä, joka yleensä on valittu mustaksi.



KUVIO 24. Tyhjien alueiden täyttäminen

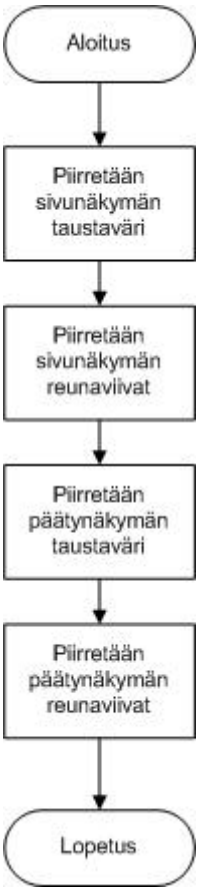
Kuten kuviossa 25 huomataan, ovat alun perin suunniteltu kuvio ja lopputulos samanmuotoisia. Lopputuloksessa näkyvät ristin, suorakaiteen ja kolmion muotoiset reunaviivat ovat vain havainnollistamista varten, käytännön toteutuksessa ne ovat kaikki samaa väriä kuin itse kappale.



KUVIO 25. Lopullinen ja alkuperäinen kappale rinnakkain

4.4.5 Sivu- ja päätynäkymän piirtäminen

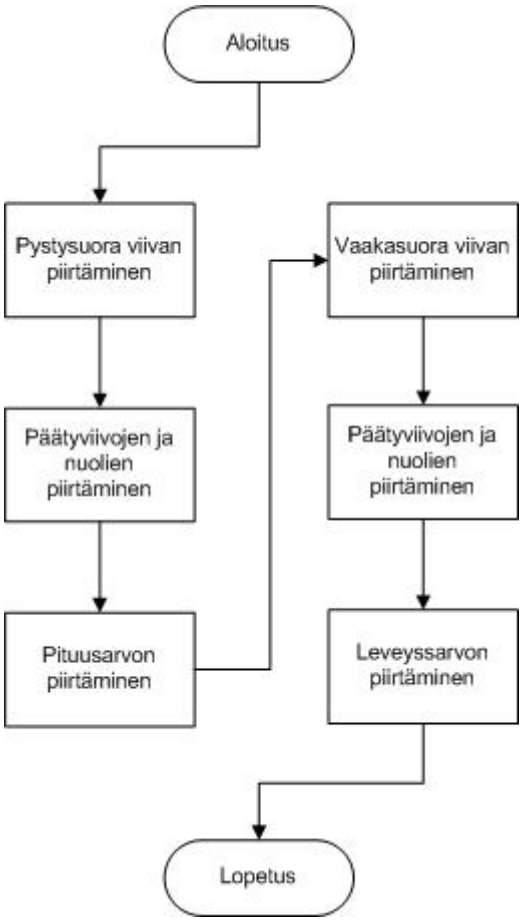
Sivunäkymässä piirretään pystysuunnassa yhtä pitkän ja päätynäkymässä vaakasuunnassa yhtä leveän suorakulmion tuotteesta, kuin mitä ylänäkymässä on. Toisena mittaarvona toimii kuitenkin kappaleen paksuus ja näin saadaan kappaleen sivu- ja päätyprofiili.



KUVIO 26. Sivu ja päätynäkymän piirtäminen

4.4.6 Mittajanojen piirtäminen

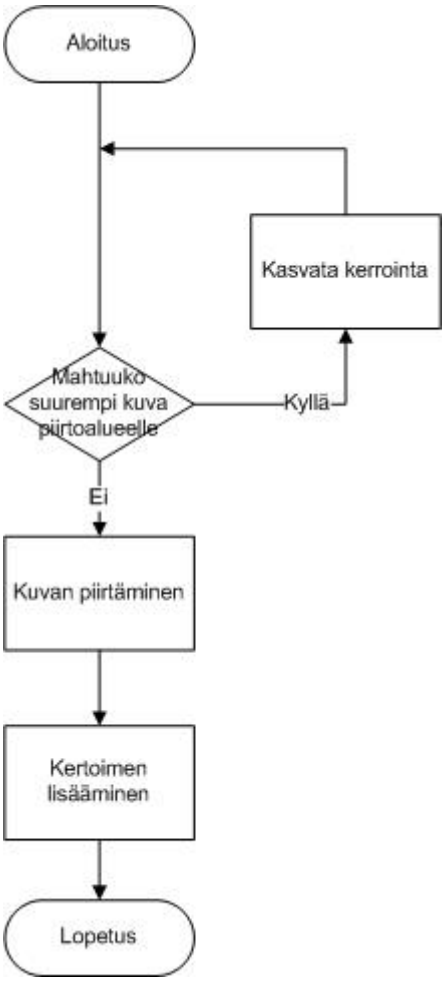
Mittajanat piirretään yhtä pitkiksi, kuin sivu- ja päätynäkymät. Päätyihin merkataan kohtisuorilla viivat merkiksi pituuden loppumisesta ja viivan rinnalle tekstinä pituuden ja leveyden määrä.



KUVIO 27. Mittajanojen piirtäminen

4.4.7 Nurkkakuvan muodostaminen

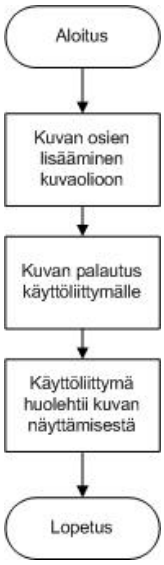
Alkuperäisten suunnitelmien mukaan komponentissa oli suorakulmion muotoinen piirtoalue, joka antoi tarkennetun kuvan mahdollisista kulmien muotoilusta. Kulmaa suurennettiin niin kauan, kun muotoiltu kulma vielä mahtui kuvaan. Nurkkakuvan antaman lisätiedon todettiin kuitenkin myöhemmin vähäiseksi, joten sen jätettiin ainakin tällä hetkellä pois toteutuksesta (Kuvio 28).



KUVIO 28. Nurkkakuvan muodostaminen

4.4.8 Kuvan palauttaminen ja Jpeg-kuvan muodostaminen

Kuvakomponentti ei toimi itsenäisesti, vaan se tarvitsee aina käyttöönsä käyttöliittymän. Käyttöliittymä kutsuu komponenttia, joka generoi syötteiden avulla kuvan ja palauttaa sen. Käyttöliittymä huolehtii lopulta palautetun kuvan näyttämisestä käyttäjälle.



KUVIO 29. Kuvan palauttaminen ja Jpeg-kuvan muodostaminen

5 YHTEENVETO

Projektilla oli useita tavoitteita. Pelkkien ohjelmointitehtävien lisäksi projektissa oppi suunnittelu- ja vuorovaikutustaitoja. Tällaisessa monimutkaisessa projektissa olisi suotavaa, että suunniteltaisiin ohjelmat kunnolla ja vasta sen jälkeen ruvettaisiin ohjelmoimaan. Tämän projektin aikana tuli kuitenkin edettyä enemmän kokeilemalla asioita käytännössä. Oman haasteen työhön toi myös ryhmän jäsenten erilaiset taustat ja elämäntilanteet. Kommunikointi toimi kuitenkin hyvin ja pitkät välimatkatkaan eivät olleet esteenä, kiitos Skype-keskusteluohjelman. Tässä työryhmässä oli erittäin viihtyisää työskennellä.

Projekti ei toteutunut alun perin suunnitellulla tavalla. Vastaan tuli useita ongelmia, joiden vuoksi asiat päädyttiin toteuttamaan eri tavalla kuin oli suunniteltu. Esimerkiksi piirtokomponentti suunniteltiin alun perin vapaasti 3-ulotteisesti pyöriteltäväksi vektorikuvaksi, jota käyttäjä voisi tarkastella haluamaltaan suunnalta. Pian työmäärä huomattiin kuitenkin turhan suureksi ja komponentti päädyttiin toteuttamaan kolmena 2-ulotteisena kuvana teknisten piirustusten tapaan.

Osa komponentin ominaisuuksista toteutettiin, mutta jätettiin vielä taustalle tulevaa kehitystä varten. Yksi tällainen ominaisuus on kuvan vasempaan alakulmaan tehty piirtoalue, jossa voidaan ilmoittaa erilaisilla symboleilla muotoiluja, joita on vaikeampi havainnollistaa kuvassa.

Vastoinkäymisistä huolimatta työstä oppi runsaasti uusia asioita, joita tulee ottaa tulevissa projekteissa huomioon jo suunnitteluvaiheessa. Huolellinen suunnittelu osoittautui tärkeäksi osaksi projektia, sillä huonosti suunniteltujen ongelmien korjaaminen vie moninkertaisesti aikaa ja vaivaa myöhemmissä vaiheissa. Ohjelmakoodin laajentuessa yli 1500 rivin huomasi, miten sekavaksi koodin seuraaminen tulee. Virheitä oli hankala paikallistaa ja parempien ohjelmointivälineiden käyttö olisi ollut suotavampaa. Lisäksi koodissa käytettiin niin monia eri muuttujia, että varmasti vähemmälläkin määrällä olisi tullut toimeen. Todennäköisesti nämä ongelmat olisi voinut välttää alun perin paremmalla suunnittelulla ennen varsinaisen ohjelmointityön aloittamista.

Jos tämän ohjelman kehitystä jatketaan, kannattaa jossain vaiheessa kirjoittaa koko ohjelmakoodin uusiksi, jotta siitä saisi huonosti toteutetut asiat korjattua. Ohjelman kehittämisen jatkaminen on vielä avoin kysymys, mutta kehitysideoita ainakin riittää.

LÄHTEET

Adobe Systemsin WWW-sivu [viitattu 4.5.2012]. Saatavissa: http://www.adobe.com

Gilmore, W. J. *PHP & MySQL Ohjelmointi*. Jyväskylä: Gummerus.

Negrino, T. 2007. *JavaScript, tehokas hallinta*. Helsinki: Readme.fi.

Jokela, S. 2011. *RIA-käyttöliittymän suunnittelu ja toteutus*. Savonia-ammattikorkeakoulu, Tietotekniikan koulutusohjelma. Opinnäytetyö.

Notepad++ [viitattu 4.5.2012]. Saatavissa: http://www.notepad-plus-plus.org.

Php. *Image Processing and GD* [verkkosivu]. Php. [viitattu 4.5.2012]. Saatavissa: http://php.net/manual/en/book.image.php.


SAVONIA