

# Utveckling av Prisjaks jämförelsesystem

Visualisering av produkter



LUNDS  
UNIVERSITET

Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg  
Datateknik

Examensarbete:  
Kristina Brandt

© Copyright Kristina Brandt

LTH Ingenjörshögskolan vid Campus Helsingborg  
Lunds universitet  
Box 882  
251 08 Helsingborg

LTH School of Engineering  
Lund University  
Box 882  
SE-251 08 Helsingborg  
Sweden

Tryckt i Sverige  
Media-Tryck  
Biblioteksdirektionen  
Lunds universitet  
Lund 2009

## Sammanfattning

Prisjakt är en prisjämförelsesida på nätet där man, förutom att jämföra produkters priser, kan jämföra produkters egenskaper.

Efter att ha kommit i kontakt med Prisjakt utarbetades förslaget till examensarbetet om att förbättra deras jämförelsesystem med inriktning på produkters storlek.

Idag anges informationen om en produkts storlek endast av måtten i millimeter (bredd x höjd x djup). För att underlätta storleksjämförelse av produkter ville Prisjakt att det för varje produkt skulle ritas ut en skalenlig bild för att illustrera storleken.

För själva implementeringen valdes JavaScript eftersom det ger bra möjligheter till interaktion.

## **Abstract**

The company Prisjakt is a price comparison site on the Internet where you, except from compare products prices, can compare product details.

After getting in contact with Prisjakt, the suggestion for improving their comparison system with concentration on the size of products was created. Today the measures of a product are shown by width, height and depth in millimetres (WxHxD).

To make this comparison easier, Prisjakt wanted a picture to be drawn for each product to illustrate the real size in a suitable scale.

For the implementation we chose JavaScript because it yields good possibilities for interaction.

# Innehållsförteckning

<b>1 Inledning .....</b>	<b>2</b>
<b>1.1 Bakgrund och syfte .....</b>	<b>2</b>
<b>1.2 Företagspresentation .....</b>	<b>2</b>
<b>1.3 Mål.....</b>	<b>2</b>
<b>2 Javascript .....</b>	<b>4</b>
<b>2.1 Historia .....</b>	<b>4</b>
<b>2.2 Syntax.....</b>	<b>4</b>
<b>2.3 Objekt och funktioner.....</b>	<b>4</b>
<b>2.4 jQuery .....</b>	<b>5</b>
<b>3 Tillvägagångssätt .....</b>	<b>6</b>
<b>3.1 Efterforskning.....</b>	<b>6</b>
<b>3.2 Genomförandet .....</b>	<b>7</b>
<b>4 Slutresultat .....</b>	<b>13</b>
<b>5 Reflektioner .....</b>	<b>16</b>
<b>6 Utvecklingsmöjligheter .....</b>	<b>17</b>
<b>7 Terminologi .....</b>	<b>18</b>
<b>8 Referenser .....</b>	<b>19</b>



## **Förord**

Jag vill tacka alla på Prisjakt som hjälpt mig under arbetets gång, speciellt tack till min handledare Jonas Birgander. Dessutom vill jag passa på att tacka min examinator Mats Lilja för all hjälp, upptäckten av CRT-problemet hade antagligen inte hittats utan dig.

# 1 Inledning

Följande kapitel är en introduktion till arbetet och en presentation av företaget Prisjakt.

## 1.1 Bakgrund och syfte

Syftet med detta examensarbete är att implementera bildgenereringen av skalenliga bilder till Prisjakts produkter och framför allt till själva jämförelsefunktionen. Prisjakt är en prisjämförelsesida på internet där man kan jämföra priser, egenskaper och mått på produkter.

Det finns idag bilder på vissa av produkterna men dessa är inte skalenliga utan är endast tänkta att visa hur produkten ser ut.

Tanken är att det utöver denna bild ska genereras en skalendig bild av produkten med hjälp av måtten (bredd, höjd och djup) från Prisjakts databas. För att sedan kunna jämföra med andra produkter ska man kunna lägga till flera objekt i samma fönster, och på så sätt snabbt se hur produkternas mått skiljer sig från varandra.

Prisjakts konkurrenter har idag inga liknande sätt att illustrera produkterna på.

## 1.2 Företagspresentation

Prisjakt är en prisjämförelsesida på nätet där man inför ett eventuellt köp kan jämföra produkters egenskaper och priser.

Arbetet med företaget startades 2000 av Jonas Bonde. Iden var att skapa en lättanvänd prisjämförelsetjänst och community för hifi- och hembiointresserade. 2002 lanserades sidan och blev snabbt populär bland dessa intressenter.

2003 lades Sveriges största hembiocommunity MinHembio.com ner pga tidsbrist hos dess skapare, Prisjakt tog då över dess diskussionsforum, galleri och filmdatabas. Sedan dess har både MinHembio och Prisjakt vidareutvecklats och nya funktioner tillkommer ständigt.

Idag har Prisjakt ett 30-tal anställda och deras huvudkontor ligger i Ängelholm.

## 1.3 Mål

Prisjakt vill att man ska kunna jämföra produkters mått visuellt på ett enkelt och smidigt sätt. På varje produkts sida ska det genereras en skalendig bild av produkten som man ska kunna se i fyra olika vyer – i 3D, framifrån, från sidan



och uppifrån. Produkterna ska representeras i form av modeller med olika färger.

När det gäller mindre produkter som t.ex. mobiltelefoner är tanken att om man t.ex. har en viss modell av en mobil uppritad på skärmen, ska man kunna sätta motsvarande mobil i verkligheten mot skärmen och måtten ska då överensstämma exakt.

För större produkter, som t.ex. tvättmaskiner, ska produkten ritas upp i lämplig skala på skärmen.

Bilgeneratorn ska utvecklas i JavaScript som är ett objektorienterat skriptspråk. Största anledningen till varför just JavaScript valdes var för att det är lätt att integrera det i Prisjaks nuvarande system.

## 2 Javascript

Här följer en kort beskrivning av JavaScript.

### 2.1 Historia

JavaScript är ett procedurellt och objektorienterat skriptspråk som exekveras i webbläsarens JavaScript-motor. JavaScript utvecklades av Brendan Eich på Netscape och introducerades 1995. Från början kallades språket Mocha, namnet JavaScript kom först senare då Netscape lade till stöd för Java i sin webbläsare. Trots namnen är de två språken inte besläktade, det finns dock en del likheter, exempelvis att de båda använder en syntax influerad av programspråket C. Även många reserverade ord är lika i de båda språken, detta är ett resultat av att Netscape och Sun Microsystems samarbetade med marknadsföringen av Netscapes dåvarande webbläsare.

Några bidragande faktorer till JavaScripts popularitet är att det finns inbyggt i de flesta webbläsare, det är ett väldigt flexibelt språk och det är enkelt att lära sig.

### 2.2 Syntax

Mycket av syntaxen i JavaScript ser likadan ut som i C, t.ex. if-satser, while-loopar, switch-satser och så vidare.

När det gäller variabler är hanterar JavaScript detta, som i de flesta andra skriptspråk, genom att man inte anger variabeltyp. Man deklarerar en variabel och beroende på vad man ger den för värde blir variabeltypen därefter.

Här följer ett exempel på JavaScript-kod:

```
<script type="text/javascript">

    var x = "123"; //skapar en sträng med innehållet "123"
    var y = Number(x) +2; //konverterar x till ett tal, adderar 2. Resultatet sparas i y
    document.write("X: " + x + " Y: " + y); //skriver ut x och y på skärmen

</script>
```

### 2.3 Objekt och funktioner

Objektorienterade programspråk är en utveckling från procedurella programspråk. JavaScript har objekt, men inte klasser och arv på det sätt som t.ex. Java har, därför är det både objektorienterat och procedurellt. Med procedurellt menas att man strukturerar upp koden enligt procedurer, eller funktioner.

Istället för klasser har JavaScript prototyper för att definiera ett objekts egenskaper, på detta sätt kan man så att säga simulera klasser.

## **2.4 jQuery**

För att göra arbetet med JavaScript enklare och mer effektivt har det skapats olika JavaScript-bibliotek. Dessa bibliotek ska förhindra att man i onödan skriver sådan kod som någon annan redan har skrivit, ett av dessa bibliotek är jQuery och det är detta bibliotek som jag till viss del använt mig av i mitt arbete.

### 3 Tillvägagångssätt

I december 2008 tog jag kontakt med Prisjakt angående examensarbete våren 2009. Jag träffade kort därefter VD Jonas Bonde och utvecklare Marcus Liedholm och vi diskuterade olika förslag till examensarbete. Vi kom överens om att arbetet skulle handla om att på något sätt generera skalenliga bilder av deras produkter på hemsidan.

Nedan följer en beskrivning av hur arbetet från början till slut gick till.

#### 3.1 Efterforskning

I slutet av januari fick jag via mail kontakt med Marcus Liedholm på Prisjakt lite mer detaljer kring vad arbetet skulle gå ut på, de hade tänkt sig något i stil med Postens ”skickaguiden”[1]. Den fungerar så att en 3d-bild av ett paket genereras genom att man fyller i mått (bredd, höjd och djup).

Efter sökningar på internet hittades även sidan Sizeazy[2], där kan man lägga till/ta bort objekt och dessa visas då skalenligt i en ruta på skärmen.

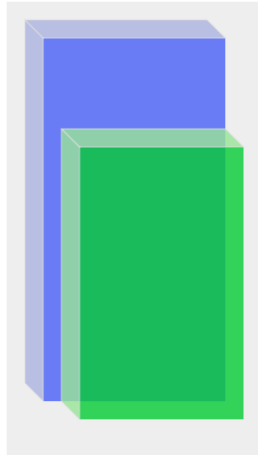
Detta var den enda lösning på internet som hade de egenskaper vi var ute efter. Tyvärr fungerar inte denna sida korrekt längre, 3d-modellerna visas fel. Efter att ha fått inspiration från dessa sidor började undersökningen om möjligheten för utveckling i JavaScript.

Marcus tipsade om HTML-elementet Canvas, som är en del av HTML5, där det finns möjligheter att presentera grafiskt material på en hemsida.

Det finns givetvis många andra sätt att presentera något grafiskt på en hemsida, men fördelen med Canvas är att det kräver väldigt lite av datorns prestanda och internetuppkoppling.

En annan viktig faktor att just JavaScript och Canvas valdes var att det skulle vara lätt att integrera det med Prisjakts nuvarande system.

Grundläggande kunskaper av Canvas erhöles genom olika internet-sidor[3][4] och därefter testades några enkla funktioner för att se om det var möjligt att fortsätta utvecklingen med hjälp av Canvas. Resultatet av dessa var att det helt klart fanns potential att kunna utveckla bildgenereringen i JavaScript med hjälp av Canvas.



*Första modellen av rätblock gjorda på en Canvas.*

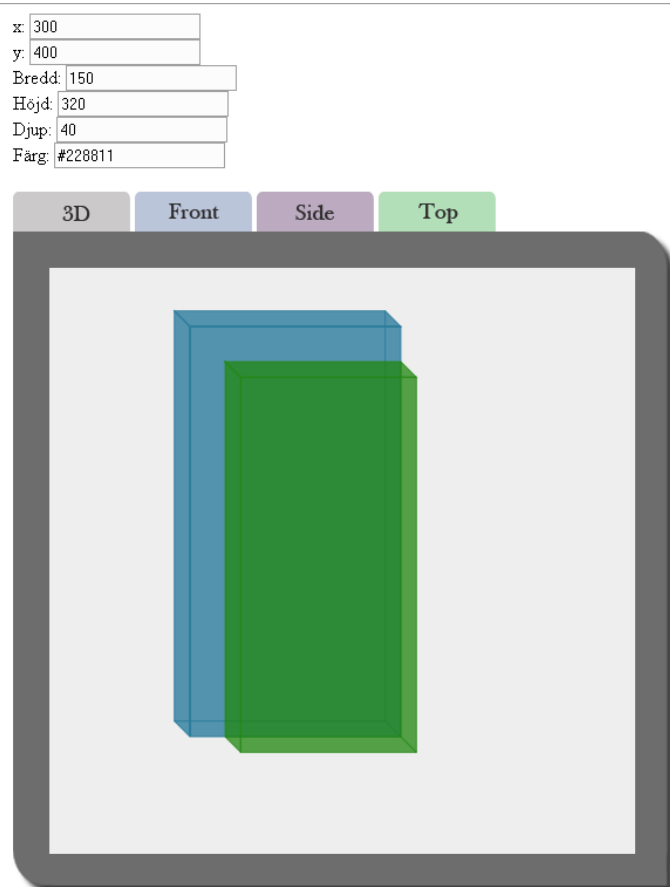
### **3.2 Genomförandet**

I början av februari träffade jag Jonas Birgander och Marcus Liedholm på Prisjakt för att gå igenom lite mer ingående vad som skulle göras och då visade jag även min testsida för dem. Jag hade där hårdkodat två rätblock i olika storlek och färg efter varandra, så att man tydligt skulle se hur resultatet i Canvas kunde se ut. De var nöjda med förslaget jag lagt fram och tyckte att det var på rätt spår.

Jag fick tillgång till egen datorplats på Prisjakts kontor och de såg till så att de program som behövdes fanns tillgängliga på datorn.

Mycket av arbetet i början gick ut på att fundera hur man på bästa sätt kunde presentera bildgenereringen av produkterna så att användaren enkelt skulle se storleksskillnaderna. Jag diskuterade och bollade idéer med Jonas Birgander och Marcus Liedholm om vad som var viktigt och vad som var mindre viktigt.

En fundering var om man skulle ha referensobjekt eller ej, t.ex. om man vid jämförelse av mobiltelefoner skulle ha en kortlek som referens. Men vid större objekt är det svårt att hitta ett passande universal-referensobjekt. Så en lösning där man själv väljer referensobjekt verkade bättre, att användaren antingen själv kan skriva in måtten på valfritt objekt eller välja befintligt objekt från Prisjakts databas som t.ex. en gammal mobiltelefon man har hemma.



*Första testsidan där man kunde lägga till objekt med hjälp av att fylla i uppgifter om mått och startpunkt av objekt i textfälten.*

För att få en bättre överblick över arbetet gjorde jag en lista med alla olika funktioner som skulle finnas, så att man efterhand som man blev klar med dem kunde bocka av dem.

Med testsidan som utgångspunkt började jag nu dela upp koden i olika funktioner för att få det dynamiskt, och även lägga till flikar i applikationen för olika vyer. Vyerna som skulle finnas var - 3d, sidan, framifrån och ovanifrån.

Nu kunde man lägga till ett rätblock genom att själv skriva in mått(bredd, höjd, djup) och färg i textfält. Även startpunkt på canvasen i form av x- och y-koordinat(räknat från övre vänstra hörnan av canvasen) skulle anges.

Nästa steg var att få rätblocken att vändas så att största måttet alltid är höjden och djupet är det minsta.

När t.ex. två tangentbord ska jämföras är det svårt att jämföra dem om man lägger in dem direkt utan att ändra om måtten eftersom ett tangentbord är minst på höjden och störst på bredden. Det man vill göra då är att ställa det på högkant, så att bredden blir höjden, djupet blir bredden och höjden blir djupet.

Exempel på produkt som måste vändas:

Bredd: 300mm

Höjd: 15mm

Djup: 90mm

Produkten vänds då enligt följande:

Bredd → Höjd

Höjd → Djup

Djup → Bredd

Resultat:

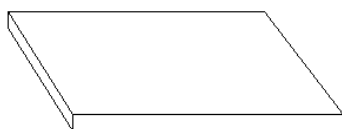
Bredd: 90mm

Höjd: 300mm

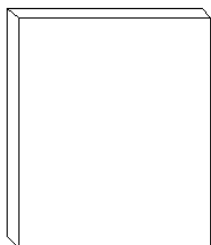
Djup: 15mm

Det svåra med detta var att ta fram en bra kodföljd som testade objektens mått och därefter ändrade måtten. Det blev en del skissande på papper i detta skede för att få det hela rätt.

Objekten läggs i 3D-vyn in efter varandra beroende på varandras djup, så skulle man lägga in t.ex. två bärbara datorer utan att först vända dem skulle det bli väldigt svårt att jämföra dem överhuvudtaget.



*Så här skulle en bärbar dator presenteras om den inte först vänds.*



*Så här presenteras en bärbar dator efter att den vänts.*

Efterföljande steg var att få rätblocken sorterade, så att man lätt kan se alla objekt man lagt till för jämförelse. Om objekten inte sorteras skulle det innebära att om man först lägger till ett litet objekt och sedan ett stort skulle vyn av det lilla objektet blockeras. Bästa resultat blev det om man sorterade efter största bredd, eftersom man ser objekten i en sidovinkel.

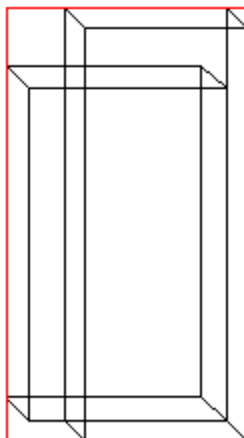
Nästa steg var att få rätblocken skalenliga, detta var en utmaning men efter lite forskande på nätet kom jag fram till en väldigt enkel lösning som jag kunde implementera i koden. Formeln jag kom fram till är följande:

$$\frac{\sqrt{\text{skärmbredd}^2 + \text{skärnhöjd}^2}}{\text{skärmstorlek i tum} * 25,4}$$

Skärmstorleken i tum multipliceras med 25,4 för att få skärmstorleken i millimeter (2,54cm = 1 tum). Alltså behöver användaren skriva in sin skärmstorlek för att få det skalenligt, skärmupplösningen kan man ta reda på genom JavaScript.

Men det jag inte upptäckte först var att för en CRT-skärm gäller inte den angivna storleken i tum, står det att det är t.ex. en 17" är den egentligen bara 16". Jag fick därför även lägga till två st. knappar (radiobuttons) på sidan där man anger om man har CRT- eller TFT-skärm.

I samband med att rätblocken gjordes skalenliga kom uppgiften med att få dem att hålla sig inom canvasen, alltså när objekt som är för stora läggs till ska de och alla eventuellt tidigare inlagda objekt automatiskt skalas ner till största möjliga storlek för aktuell canvas. Detta var ganska problematiskt, har man ett objekt på canvasen är det inte svårt att hålla detta innanför men har man flera blir det värre. Detta löstes genom att ha en stor virtuell rektangel, vars mått baserades på den sammanlagda yta som objekten tog upp på canvasen.



*I bilden ovan är två objekt uppritade, den röda markeringen visar den virtuella rektangel som skapas för att hålla reda på hur mycket plats objekten tar upp på canvasen.*



När sedan objekten ska ritas ut på canvassen ska de först centreras, detta gjordes enkelt genom följande formel:

$(\text{storlek på canvas} - (\text{storlek på canvas} - \text{rätblockens sammanlagda storlek})/2)$

Med hjälp av denna formel fick man fram vad den nya startpunkten i form av x- och y-koordinat skulle vara för objekten.

Objekten sorteras, som beskrivs tidigare, baserat på bredd innan de ritas ut, men det skulle även finnas möjlighet att sortera dem manuellt. Detta löste jag med hjälp av en sorterbar lista där man kan dra runt elementen i listan och på så sätt ändras även rätblockens position på canvassen.

Denna lista är gjord med hjälp av jQuery UI[5].

I denna lista presenteras objekten som man lagt till i form av rektangulära element, med objektets färg i textform för att kunna skilja objekten åt. Detta kommer senare att bytas ut mot objektets namn, t.ex. Apple Iphone 3G 16GB, då bildgenereringen kopplas samman med Prisjakts databas.



*Exempel på förflyttning av objekt med hjälp av listan.*

Nästa steg var alltså då att koppla ihop scriptet med Prisjakts databas, detta gjordes med hjälp av jQuery[6] och Ajax. Jag använde en Ajax/jQuery callback-funktion[7] för att hämta information om en vald produkt, genom att skicka med produkt-id fick jag tillbaka produktens namn, bredd, höjd, djup samt färg. Med dessa uppgifter kunde sedan ett objekt skapas på canvassen och produktens namn lades till i sorteringslistan till höger.

Ett problem som upptäcktes tidigt var kompatibilitetsproblem med webbläsaren Internet Explorer.

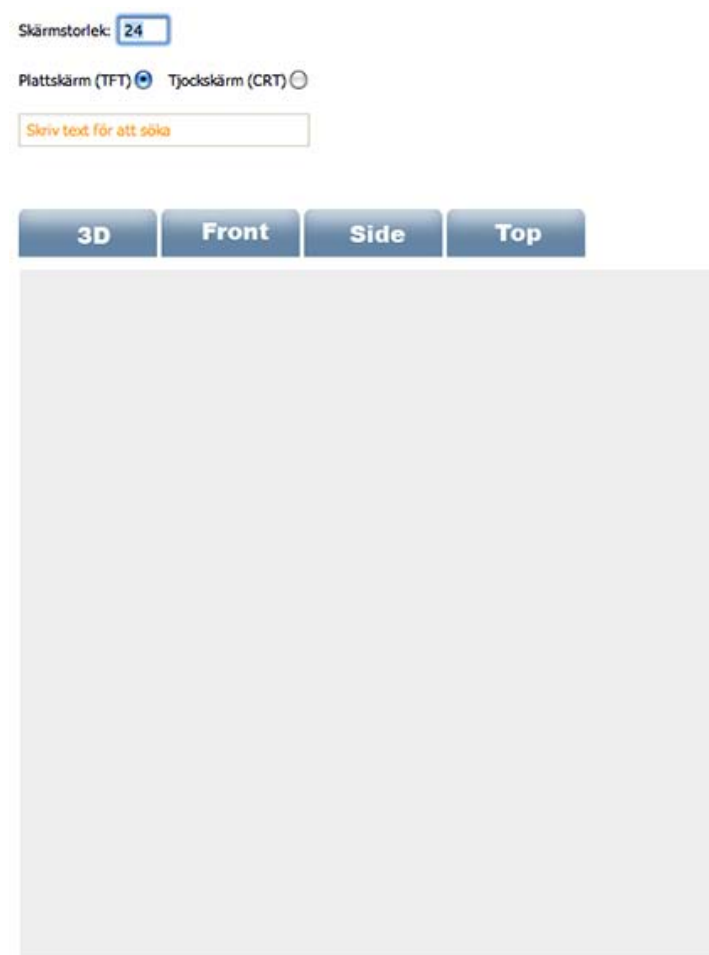
Internet Explorer använder sig utav VML (Vector Markup Language) istället för Canvas för att visa grafik, men det här problemet löstes snabbt genom Googles script ExplorerCanvas. ExplorerCanvas är helt enkelt ett script som Google tillhandahåller, detta skript översätter Canvas till VML. Genom att man inkluderar en scriptfilen "excanvas.js" på sin sida kan Canvas visas även i Internet Explorer.

## 4 Slutresultat

Här följer ett antal bilder med förklarande bildtext som visar slutresultatet av bildgeneratorn.

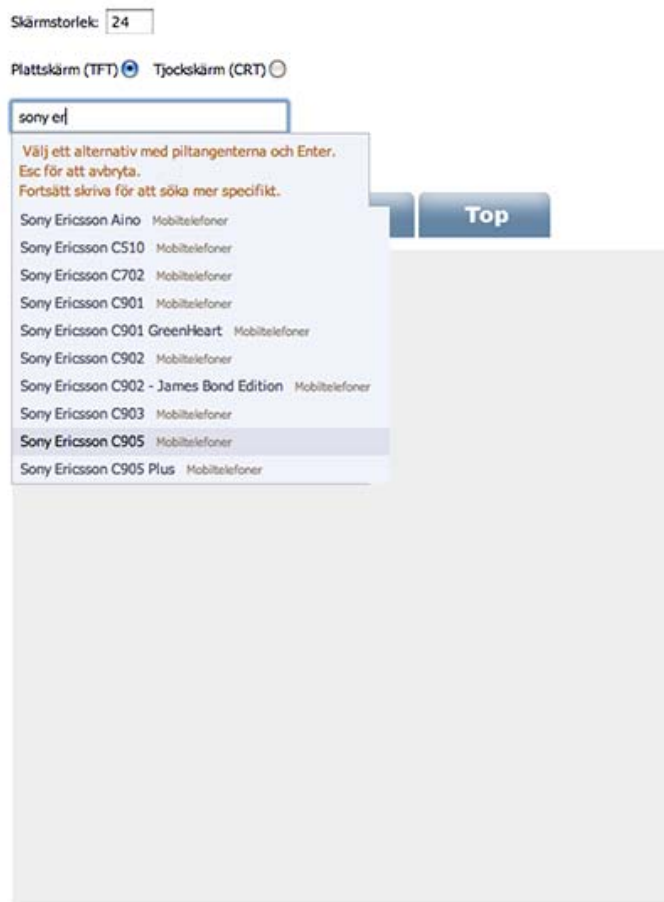
Bildgeneratorns placering på Prisjakt's sida är ännu inte klar, så i nuläget finns den på en egen adress i ett eget webbläsarfönster.

Adressen dit är <http://kristina.prisjakt.nu/Compare.html>

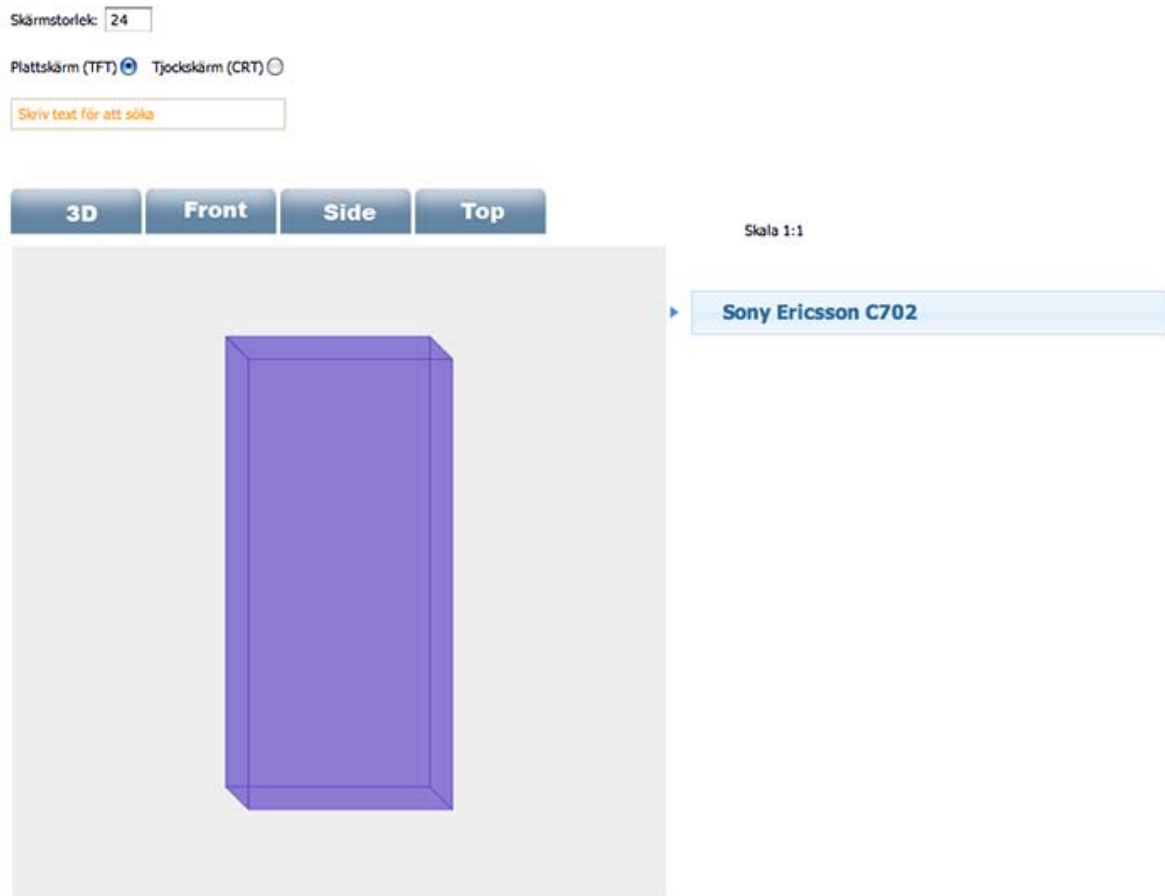


The screenshot shows a web interface for selecting screen specifications. At the top, there is a label 'Skärmstorlek:' followed by a text input field containing the number '24'. Below this, there are two radio button options: 'Plattskärm (TFT)' which is selected, and 'Tjockskärm (CRT)'. Underneath these options is a search input field with the placeholder text 'Skriv text för att söka'. At the bottom of the interface, there are four buttons labeled '3D', 'Front', 'Side', and 'Top', which are used to view the product from different perspectives. The main content area below these buttons is currently blank.

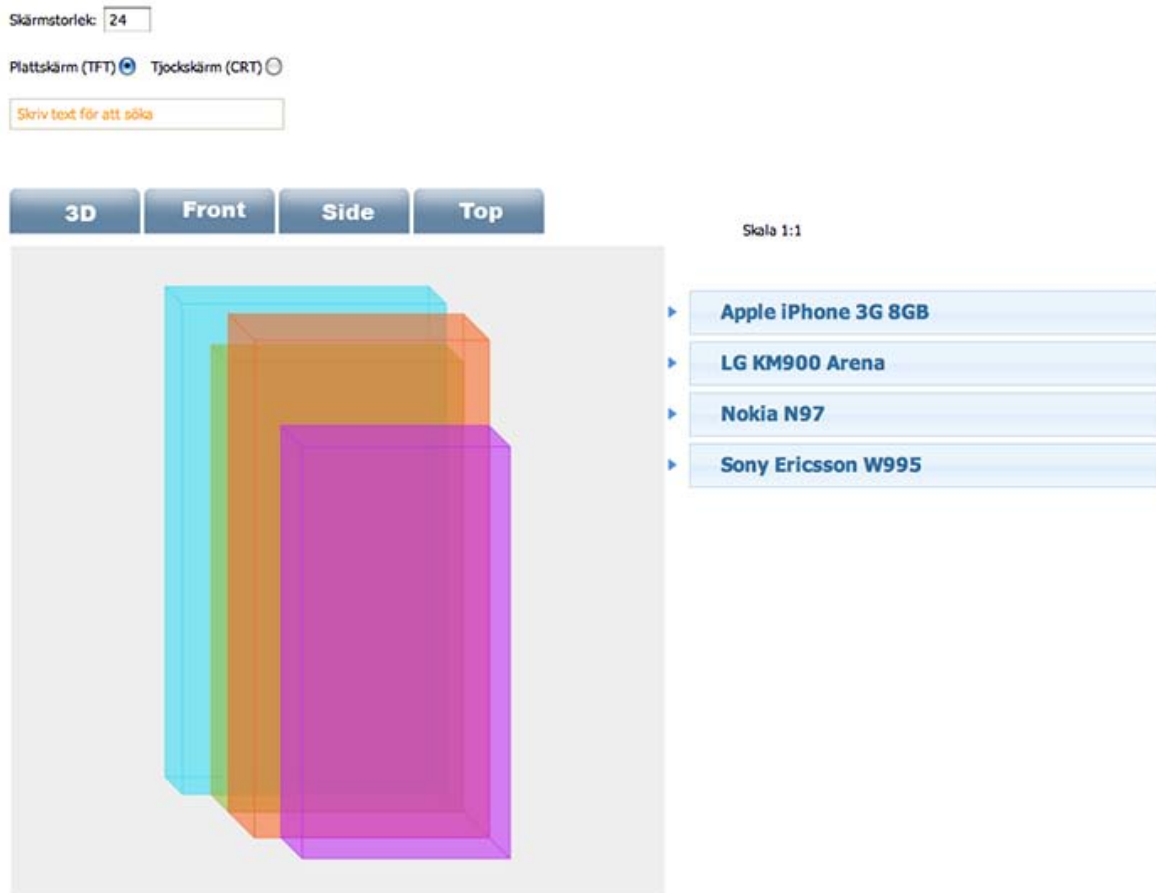
*Fig. 1. Man börjar med att fylla i sin skärmstorlek i översta fältet, sedan väljer man om man har TFT- eller CRT-skärm.*



*Fig.2. När du skriver något i sökrutan får du upp en lista med ett antal produkter som matchar din sökning. Genom att klicka på någon av dessa läggs den till som ett objekt i canvasen.*



**Fig. 3.** Här har vi lagt till mobilen Sony Ericsson C702. För att byta vy klickar man på någon av de 4 flikarna 3D (nuvarande vy,), Front, Side eller Top. Vill man lägga till fler produkter är det bara att skriva i sökrutan igen.



*Fig 4. Här har vi lagt till fyra mobiltelefoner och här ser man också att de sorteras efter bredd när man lägger in dem. Genom att dra i listelementen kan man enkelt placera om objekten efter hur man själv vill ha dem.*

## 5 Reflektioner

Redan från start har jag tyckt att arbetet har varit spännande och intressant just för att man skapar något där man direkt kan se ett grafiskt resultat på skärmen. Kombinationen grafik och teknik har jag alltid tyckt varit intressant då jag är intresserad av både konst och datorer. Att då kunna kombinera dessa intressen har för mig varit väldigt roligt och stimulerande.

Jag har under arbetets gång sökt information på nätet och läst om liknande lösningar och möjligheter kring Canvas, det tillkommer ständigt nya möjligheter för Canvas. Men jag har även läst mycket om grundläggande Javascript eftersom jag tidigare inte hållit på med det mer än enklare script för hemsidor. Ett ovärderligt plugin till webbläsaren Firefox som jag använde under princip hela utvecklingen var Firebug. Detta plugin gör att man direkt i Firefox kan upptäcka fel i bl.a. JavaScript- och HTML-kod. Detta har

verkligen varit till stor hjälp i mitt arbete då det annars inte är så lätt att felsöka HTML och JavaScript.

Med tanke på detta tycker jag att arbetet i stort har fungerat bra och samarbetet med Prisjakt har varit till stor hjälp.

Det som kunde gjorts bättre är att man från början skulle haft klara mål vad gäller de olika funktionerna i bildgeneratorn. Vad som skulle finnas med och inte, vad som var nödvändigt och inte samt vad som skulle göras i mån av tid. Istället blev det att man efterhand såg vad som skulle behövas och implementerade det därefter. Det blev därför svårt att avgöra hur mycket som var kvar på arbetet och därmed också svårt att uppskatta tidsåtgången till det hela.

Samtidigt var det svårt att från början ha en klar bild av hur det skulle se ut, efterhand som man testade märkte man vad som saknades och vad som skulle vara bra att ha med.

## 6 Utvecklingsmöjligheter

Det finns många utvecklingsmöjligheter både för bildgeneratorn i sig och även när det gäller att använda den för andra syften. T.ex. skulle man kunna använda den för att räkna ut volymen på böcker när de ska transporteras, så att man vet hur mycket packningsmaterial som behövs för de böcker man valt. På så sätt skulle man spara in på materialkostnad och även miljön.

Möjliga utvecklingar för bildgeneratorn åt prisjakt är att:

1. Vidareutveckla 3d-vyn så att man kan vrida på objekten i realtid på canvasen.
2. Att ha bilder av produkterna till objekten i de olika vyerna.
3. Byta plats på objekten genom att dra i objekten på canvasen istället för bara i listan.
4. Att namnet/information på/om produkten visas när man för musen över motsvarande objekt i canvasen.
5. Att ett fält med samma färg som objektet på canvasen visas tillsammans med objektets listelement.

Dessa är de möjligheter som vi har funderat över men som inte varit möjliga p.g.a. tidsbrist, det finns säkert många fler.

## 7 Terminologi

Ord

Förklaring

JavaScript	Objektorienterat skriptspråk
Ajax	(Asynkron JavaScript och XML) Används för att bygga applikationer för World Wide Web.
Canvas	HTML-element som skapar en yta som man kan rita på.
jQuery	Javascript-bibliotek.
VML	(Vector Markup Language) ett XML-språk för att presentera vektor-grafik.
HTML	(HyperText Markup Language) Webbstandard för strukturering av bl.a. text och media på exempelvis webbsidor och i e-postmeddelanden.
Firebug	Plugin till webbläsaren Firefox. Gör så att man kan redigera, felsöka och övervaka bl.a. HTML och JavaScript på valfri webbsida.
Netscape	Företag som bl.a. utvecklade webbläsaren Netscape.
Sun Microsystems	Företag som bl.a. tillverkar processorer, serverdatorer och mjukvara som t.ex. Java.
Google	Amerikanskt internetföretag, mest känt för sin sökmotor med samma namn.



## 8 Referenser

1. [http://www.posten.se/skickaguiden/main\\_do.jsp](http://www.posten.se/skickaguiden/main_do.jsp)  
[2009-06-18]
2. <http://www.sizeasy.com/>  
[2009-06-18]
3. <http://blog.thejit.org/2009/01/12/a-new-canvas-element/#canvas>  
[2009-06-18]
4. [https://developer.mozilla.org/en/drawing\\_graphics\\_with\\_canvas](https://developer.mozilla.org/en/drawing_graphics_with_canvas)  
[2009-06-18]
5. <http://jqueryui.com/demos/sortable/>  
[2009-06-18]
6. <http://jquery.com/>  
[2009-06-18]
7. <http://docs.jquery.com/Ajax/jQuery.getJSON#urldatacallback>  
[2009-06-18]
8. <http://www.w3schools.com/JS/default.asp>  
[2009-06-18]
9. <http://www.w3schools.com/jsref/default.asp>  
[2009-06-18]
10. Davis, Harold, 2004. Learn How to Program Using Any Web Browser.  
Apress.