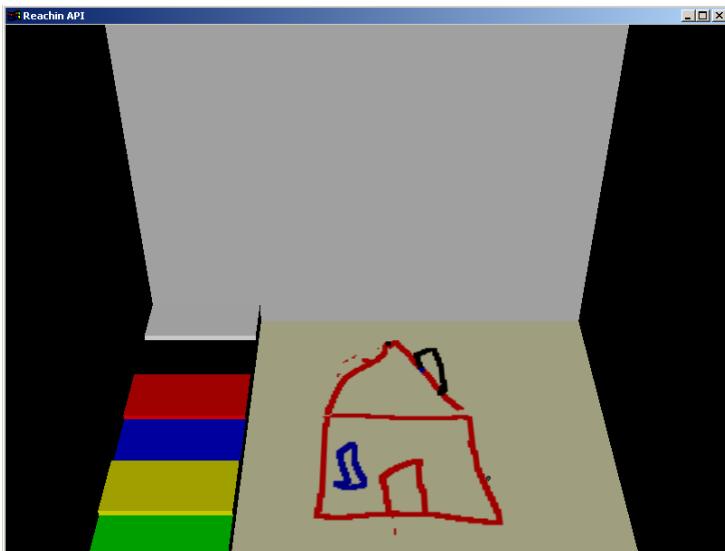




EXAMENSARBETE CERTEC, LTH NUMMER 5:2003

Christin Hansson

Haptiskt ritprogram



Avdelningen för rehabiliteringsteknik
Institutionen för designvetenskaper
Lunds tekniska högskola

Sammanfattning

Blinda personer har ett behov av att uttrycka sig i och att tolka bilder precis som seende personer. Blinda personers behov är dock åsidosatt, ofta antar seende att blinda personer inte kan rita och inte har ett behov av det - ”De ser ju ändå inte vad de ritar för något”.

I det här examensarbetet beskrivs det hur jag utvecklar ett ritprogram med olika färger för blinda och synskadade personer. För att använda programmet utnyttjas the PHANToM, på svenska Fantomen, som är ett haptiskt användargränssnitt som används för att känna virtuella objekt. Fantomen ser ut som en robotarm med en penna som användaren håller i. När användaren rör pennan i olika riktningar ger armen motstånd då man stött på det virtuella föremålet. Fantomen använder sig av enpunktshaptik, så det känns som om man känner på föremålet med en penna eller ett finger.

Man kan programmera Fantomen på olika sätt, jag har valt programmeringsspråket C++ och Reachin API.

För att utveckla ritprogrammet har jag använt mig av designteorin Usability Engineering som är utvecklad av dansken Jakob Nielsen.

Resultatet av examensarbetet är ett ritprogram i vilket man kan rita egna bilder i olika färger och sedan känna på resultatet. Hur det fungerar och hur det har utvecklats diskuteras i denna rapport.

Nyckelord

haptik, rita, blind, synskadad, Fantomen, the PHANToM, ritprogram, färger

Abstract

Blind people have a need for expressing themselves in pictures and for interpreting them just like seeing people. The needs of blind persons are often ignored because seeing persons assume they do not know how to paint and have no need for it – “They can’t see what they paint anyway.”

This master’s thesis describes the development process for a haptic drawing program with colors. The program’s target group consists of blind and visually disabled persons.

The PHANToM is a haptic interface device that makes it possible to feel virtual objects in three dimensions. In appearance, the PHANToM resembles a pen whose nib is attached to a robot arm. The user holds the PHANToM pen in the same way as an ordinary pen. It can be moved in all directions and gives resistance when it touches a virtual object. To use the PHANToM is almost like holding a pen and dragging its nib along an object.

The PHANToM can be programmed in different ways. In this thesis I have used the Reachin API interface and the C++ programming language.

The development process of the program is based on the Usability Engineering Theory of Jacob Nielsen from Denmark.

The result of the thesis is a drawing program in which you can paint your own pictures and then feel what you have created. The thesis describes how it works and how it has been developed.

Keywords

haptics, paint, blind, visually disabled, the PHANToM, drawing program, colors

Förord

Detta examensarbete är det sista steget i min civilingenjörsutbildning i datateknik. Arbetet har utförts på Certec i Lund mellan mars och oktober 2003, med uppehåll under juni och juli då jag istället programmerade mobiltelefoner på Sony Ericsson.

Jag vill tacka alla som hjälpt mig att fullborda arbetet, främst är det Charlotte Magnusson som varit min handledare på Certec, Annika Svensson för hjälp med pilottestet och korrekturläsning, Johanna Isacsson för korrekturläsning, Kia Andersson som lånat ut sin röst till programmet och de blinda personer som ställt upp på användartestet.

Christin Hansson

Lund, oktober 2003

Innehållsförteckning

Innehållsförteckning	6
1 Inledning	9
1.1 Problem.....	9
1.2 Mål	9
1.3 Innehåll	9
2 Bakgrund	11
2.1 Haptik	11
2.2 The PHANToM – Fantomen.....	11
2.3 Reachin API	12
2.4 Förundersökning.....	13
2.5 Synskadade och datoranvändning.....	13
2.6 Synskadade och att rita	14
3 Teori – Usability Engineering	15
3.1 Lär känna användaren.....	15
3.2 Analys av konkurrenter	16
3.3 Användbarhetsmål	16
3.4 Parallell design.....	17
3.5 Medverkande design	17
3.6 Design av hela gränssnittet	17
3.7 Riktlinjer och heuristisk utvärdering	18
3.8 Prototyp	19
3.9 Utvärdering.....	19
3.10 Iterationer	20
4 Metod	21
4.1 Utgångspunkt	21
4.2 Lära känna användarna.....	21
4.3 Vad ska man kunna göra med programmet?.....	22
4.4 Analys av tidigare program	22
4.5 Design av programmet.....	22
4.6 Genomgång av riktlinjer	24
4.7 Prototyp	25
4.8 Iterationer	25
5 Användartest	27
5.1 Utförande.....	27
5.2 Man, 23 år	27
5.3 Kvinna, 29 år.....	31
5.4 Man, 45 år	33
5.5 Resultat av testen	36
6 Framtid och förslag till utökningar	37
6.1 Brister	37
6.2 Förslag till utökningar	37
6.3 Framtid	38

7	Tankar och erfarenheter	39
7.1	Tid	39
7.2	Sätta sig in i blindas situation.....	39
7.3	Kontinuerlig dokumentation.....	40
7.4	Testsituationen.....	40
	Referenser	41
	Tryckta referenser.....	41
	Elektroniska referenser.....	41
	Bilaga A: Sammanfattning av telefonintervju.....	43
	Bilaga B: Testprotokoll	45

1 Inledning

1.1 Problem

Det är lätt för en seende person att använda ett ritprogram på en dator. Om man är synskadad eller blind är det inte riktigt lika enkelt att utnyttja ett sådant program eller att få färger och former beskrivna för sig. Det krävs att personen som beskriver hur något ser ut har stor vana och kunskap om hur man beskriver ett föremål eller en bild för en person som inte kan se. På Certec har det tidigare utvecklats ett haptiskt ritprogram som heter "Rita med fingrarna". Detta har bedömts vara för gammalt och inte längre användbart, därför har jag fått i uppdrag att göra ett nytt.

1.2 Mål

Målet med mitt examensarbete var att skapa ett nytt haptiskt ritprogram som skulle styras med hjälp av "the PHANToM", eller Fantomen, som är ett haptiskt känselgränssnitt. Med programmet ska man kunna urskilja olika färger och det ska vara lätt att känna övergången mellan de olika färgerna. Man ska kunna välja om man vill rita eller om man vill känna på det man har ritat. Det ska även uppfattas som enkelt att använda.

1.3 Innehåll

Först kommer en beskrivning av problemet, därefter hur det är löst och sedan hur jag kom fram till just den lösningen. Sedan följer ett avsnitt som tar upp möjliga förbättringar och utökningar som kan komma att implementeras senare.

Kapitel 2 innehåller bakgrundsfakta och presentation av verktygen som använts under utvecklingen. I kapitel 3 beskrivs den designteori (Usability Engineering) jag använt mig av under utvecklingen. Därefter följer kapitel 4 som beskriver hur teorin har använts i praktiken och kapitel 5 där det redogörs för utförandet och resultatet av användartestet. Kapitel 6 tar upp programmets framtid och förslag till utökningar. Som avslutning finns tankar och erfarenheter om det färdiga examensarbetet i kapitel 7.

2 Bakgrund

2.1 Haptik

Ordet haptik kommer från det grekiska "haptesthai" som betyder "att känna". Betydelsen av haptik i detta examensarbete är att man kan sitta vid en dator och känna på ett objekt som är datorgenererat och alltså inte existerar i den riktiga världen. Man kan göra allt från att känna på en kub eller trycka på en knapp till att känna hur det känns att utföra ett kirurgiskt ingrepp.

2.2 The PHANToM – Fantomen

The PHANToM, även kallad Fantomen, är ett haptiskt känselgränssnitt. Den är utvecklad på Massachusetts Institute of Technology (MIT) i USA och vidareutvecklas av ett företag som heter SensAble Devices Inc.. Namnet PHANToM har dubbla betydelser, dels för att man känner föremål som inte finns, dels är det en förkortning av Personal Haptic iNterface Mechanism.

Fantomen använder sig av enpunktshaptik. Det betyder att man hela tiden bara återger en punkt i taget, dvs att det är som att känna på ett föremål med bara pekfingret, eller om man håller i en penna som man känner med.



Figur 1: Fantomen

2.2.1 Fantomen på Certec

Certec använder Fantomen främst för att göra applikationer för blinda användare. I ett pågående projekt bygger man upp olika trafikmiljöer i datorn så att den blinde användaren ska kunna träna sig på hur det ser ut i verkligheten för att kunna ta sig fram säkrare. Det har även gjorts spel, matematikprogram och program för att kunna känna på en ritning.

2.2.2 Olika användningsområden för Fantomen

Fantomen används inte bara som hjälpmedel för synskadade. Det finns även andra spännande användningsområden som till exempel för att träna läkare i kirurgi och för att utforska komplex 3D-information.

2.3 Reachin API

För att kunna styra Fantomen har jag använt mig av Reachin API. Det kombinerar C++, VRML (se 2.3.1), Open GL och Python för att på ett så enkelt sätt som möjligt kunna skapa haptiska applikationer.

Reachin API är framtaget av ett svenskt företag som heter Reachin. De tillverkar även andra produkter man kan använda tillsammans med Fantomen, men de har jag inte använt.

För att skriva enklare applikationer där man bara vill känna på ett virtuellt föremål räcker det att använda VRML. Vill man göra mer avancerade saker, till exempel flytta på saker eller om man vill att något ska röra på sig, måste man använda C++ eller Python. Jag valde att använda C++ eftersom jag hade lite kunskap om det sedan tidigare. Python hade jag däremot aldrig stött på.

Ett program som baserats på Reachin API är uppbyggt kring två loopar. En grafikloop som sköter det som finns på skärmen och en haptikloop som sköter det som känns. Haptikloopen uppdateras minst 1000 gånger per sekund och grafikloopen uppdateras 20-30 ggr per sekund. Anledningen till att haptikloopen uppdateras mycket mer frekvent är att det annars är stor risk att man ”åker igenom ytan” på föremålet och hamnar ”inuti det”, vilket inte är meningen.

2.3.1 VRML – Virtual Reality Modeling Language

VRML står för Virtual Reality Modeling Language och används för att beskriva objekt i en tredimensionell värld. Det fungerar ungefär som HTML men HTML presenterar text och VRML presenterar istället tredimensionella bilder.

Man kan skriva kod manuellt eller använda sig av en 3D-modellerare, till exempel 3D-studio MAX, i vilken man bygger objekten grafiskt och sedan plockar ut koden för att lägga in den i

sitt program. Jag valde att skriva in koden manuellt eftersom jag ville ha kontroll över det jag gjorde och mina objekt inte är speciellt avancerade.

Reachin API använder sig av en utökad version av VRML97. Man har lagt till möjligheten att ge ytor textur och det finns noder som gör det möjligt att känna det som syns. Detta är information som är onödig om man bara ska titta på VRML-objekten, men om man ska känna på dem är det högst intressant att veta om en yta är skrovlig eller hal.

2.3.2 C++

C++ är ett objektorienterat programmeringsspråk som bygger på programmeringsspråket C. Det skapades av dansken Bjarne Stroustrup i slutet av 1970-talet och är ett standardiserat språk som används av miljoner utvecklare i tusentals projekt över hela världen.

2.4 Förundersökning

Jag har sökt runt på nätet för att se om det finns något liknande program, så att man kan kika och se vad de har gjort för att få inspiration, men jag har inte lyckats hitta något liknande. Det som har funnits tidigare är det redan nämnda ”Rita med fingrarna” som inte fungerar och därför inte går att undersöka. Det som har använts ifrån det är kunskapen att åtta färger är lagom många att skilja på [Rasmuss, 1996].

2.5 Synskadade och datoranvändning

I dagsläget finns det en del hjälpmedel för synskadade som vill använda datorer. Det finns till exempel förstoringsprogram för dem som ser lite grand. För de som ser väldigt dåligt eller inte alls finns det talsyntes, som läser upp texten på skärmen för användaren. Om man hellre vill läsa punktskrift, finns det speciella tangentbord som har en display där punktskrift visas då användaren för musen över text på skärmen.

De flesta synskadade kan använda ett vanligt tangentbord då det finns upphöjda markeringar på tangenterna F och J på det alfabetiska tangentbordet och på 5 på det numeriska. De används även av seende personer som oftast inte tittar på tangentbordet för att få rätt fingerställning.

För synskadade är det enklast att använda det numeriska tangentbordet och piltangenterna, då de är placerade i mindre grupper som gör det lättare att kontrollera dem.

2.6 Synskadade och att rita

Jag har många gånger under tiden jag hållit på med arbetet fått frågan om varför blinda och synskadade personer ska kunna rita. ”De ser ju ändå inte vad de ritar för något” är det vanligaste argumentet. Jag har förstått att många människor inte tror att blinda överhuvudtaget har något behov av att uttrycka sig i bilder. Så förhåller det sig dock inte.

Det finns många blinda och synskadade som ägnar sig åt konst i alla dess former och läser högre konstutbildningar på universitet runt om i världen. För övrigt är det många kända konstnärer som var synskadade och man kan se det i deras verk. Claude Monets synskada syns om man tittar på tavlor han gjort i olika tidsperioder på samma/liknande motiv, ju nyare tavla desto suddigare motiv. Andra kända konstnärer med synskador är Mary Cassatt, Vincent van Gogh, Edgar Degas och Rembrandt [Axel & Levent(red), 2002].

Vad är det då som motiverar en blind person att rita? Detta tar John M. Kennedy upp i en artikel i boken ”Art beyond sight”. [Axel & Levent(red.), 2002] Han poängterar att många blinda personer som barn har blivit hindrade att försöka sig på att rita eftersom föräldrar och lärare sagt åt dem att de inte kan eftersom de är blinda.

2.6.1 Skillnad mellan synliga och taktila bilder

Om man sätter sig ner och börja tänka på vad det är vi egentligen gör när vi ritar bilder, kommer man fram till att det vi främst ritar är konturer. Det man kan känna på olika objekt är också konturerna, så om man ritar det man kan känna är det oftast igenkännbart för både seende och blinda personer. Dock finns det en del saker man måste tänka på. Ytorna ska vara lätta att särskilja och det ska vara lätt att följa linjerna. Det ska inte vara för mycket detaljer i bilderna utan man ska försöka få det så rent som möjligt med bara de nödvändigaste detaljerna som krävs för förståelse. Man ska helst inte försöka återge perspektiv i bilden eftersom det är svårt att uppfatta. Dock finns det forskare som anser att det är möjligt att uppfatta vissa former av perspektiv [Eriksson, 1997].

3 Teori – Usability Engineering

Det finns ingen speciell designteori för hur man utvecklar mjukvara för blinda på ett användarcentrerat sätt. Därför har jag använt mig av metodiken *Usability Engineering* som är en metodik för utveckling av mjukvara med användbarhet som ett centralt perspektiv.

Usability Engineering är inte något man applicerar i slutet av utvecklingscykeln, utan det måste man planera in redan från början för att få med kunskap om användarna i utvecklingen av mjukvaran.

Hela teoriavsnittet bygger på Jacob Nielsens teorier [Nielsen 1993] om inga andra referenser ges.

3.1 Lär känna användaren

Första steget är att lära känna användarna genom att studera hur de fungerar och hur de går till väga för att utföra sina uppgifter idag. Alla användare har egna lösningar och man ska tänka på att även studera de som indirekt kommer i kontakt med systemet, till exempel de som ska installera och administrera systemet.

Man vinner mycket på att ta reda på hur användarna är och vad de tycker om, istället för att sitta och diskutera sig fram till hur de bör använda ett system och vad de borde tycka om det.

3.1.1 Utmärkande drag hos användarna

Det är nödvändigt att veta vilken grupp av användare som ska använda systemet. Ibland är det lätt, till exempel om man ska utveckla ett system till en viss avdelning på ett företag. Ibland är det svårare om systemet ska vara vidare spritt, då får man välja ut ett par karakteristiska användare.

Genom att veta användarnas datorvana, ålder, utbildningsnivå och så vidare, är det lättare att anpassa systemet på ett sätt som gör det lättare för dem att lära sig det och enklare att bestämma komplexiteten för användargränssnittet. Ett exempel är små barn som inte kan läsa, till dem är ett icke textbaserat gränssnitt att föredra. Blinda personer kan inte se och då är ett känselbaserat och/eller ljudbaserat gränssnitt att föredra. Man bör också ta reda på hur lång tid användarna har på sig att lära sig utnyttja programmet. Är det tänkt att det ska göras utan träning, måste gränssnittet vara gjort på ett sådant sätt att det krävs minimalt med träning och användaren ska klara av att använda det direkt.

Miljön användaren sitter i ska också studeras. Är det ett kontorslandskap eller ska systemet användas av personer som sitter i en lugnare miljö? Om det är ett kontorslandskap ska man till exempel inte använda sig av ljud som feedback för det kan verka dels störande för de omkringvarande, dels pinsamt för personen som genererar ljudet.

3.1.2 Uppgiftsanalys

Vad vill användarna kunna göra med systemet? Vad behövs för att man ska kunna göra det? Det är två saker som ska komma fram under uppgiftsanalysen. Man ska även ta reda på hur användaren konkret vill utföra det och gärna även observera dem när de utför uppgifterna, då det är lätt hänt att glömma vissa steg när man förklarar eftersom man själv ser dem som självklara.

Man ska i detta steg även ta hänsyn till undantagsfall som kan inträffa och på vilket sätt användarna tycker att de är enklast att lösa.

I denna fas ska man inte tänka på själva designen av systemet utan bara fokusera på vilka uppgifter det ska klara av att utföra.

3.2 Analys av konkurrenter

Finns det konkurrerande produkter är det en bra idé att analysera dem enligt etablerade användbarhetsriktlinjer och att utföra empiriska tester på dessa produkter. Tester på riktiga produkter är mer realistiska än de som görs på prototyper. Man kan testa hela händelser och se om programmets funktionalitet stöder den sorts uppgifter den nya produkten är tänkt att kunna utföra.

Finns det många olika konkurrerande produkter kan man även göra en jämförande analys på deras olika gränssnitt och därigenom få nya uppslag över hur det kan lösas på ett bättre sätt.

3.3 Användbarhetsmål

Användbarhet är inte bara något man enkelt kan bedöma om den är bra eller dålig. För att kunna utvärdera ett system måste man sätta upp användbarhetsmål. Man måste också prioritera vilka av målen som är viktigast, med analysen av användarna som grund eftersom det inte är helt ovanligt att de olika målen strider mot varandra.

För att kunna avgöra om ett mål är uppnått måste det vara mätbart. Olika exempel på mätbara mål är hur många fel en användare gör per tidsenhet, hur lång tid det tar att utföra en viss uppgift, men även hur användaren uppfattar programmet, till exempel hur nöjd användaren är med resultatet eller hur lätt användaren anser att det är att lära sig hantera programmet.

Det är lämpligt att analysera ekonomiska perspektiv när man sätter upp användbarhetsmål. Om programmet är så lättanvänt att man behöver en dags mindre utbildning än vad man annars räknar med kan vara värt en hel del pengar. På liknande sätt kan förbättringar av användbarheten leda till ökad produktivitet och att man på så sätt sparar pengar för företaget.

3.4 Parallell design

Det är bra att starta designprocessen med flera designers som jobbar med samma problem. Målet är att man ska utforska designrymden innan man bestämmer sig för ett förslag som man senare utvecklar en prototyp av. Därför är det viktigt att de olika designerna inte har insikt i vad de andra gör för att få så många olika förslag som möjligt.

Det är speciellt viktigt att använda parallell design då man utvecklar system där det inte existerar så många riktlinjer för vad som fungerar bäst. För mer traditionella system, där konkurrerande produkter är tillgängliga kan man använda konkurrentanalysen (se avsnitt 3.2) som parallell design. Dock är det fortfarande fördelaktigt att ha flera olika designers som utvecklar olika förslag för att utöka möjligheterna.

3.5 Medverkande design

Medverkande design eller ”participatory design” är en metod som används för att bryta barriärer mellan de som utvecklar mjukvaran och de som ska använda den. Användare ställer ofta frågor som utvecklarna inte tänkt på eller sett som självklara. Om man låter användarna vara med i utvecklingsprocessen kan man hitta många brister i god tid så att de blir lätta att ändra på.

Användarna ska dock inte komma med övergripande förslag om hur systemet ska se ut, då de trots allt inte är designers. Däremot är de väldigt bra på att komma med kommentarer och åsikter om konkreta designförslag.

Om man har ett stort projekt är det viktigt att byta ut de användare som medverkar i designgruppen, eftersom de ju mer kunskap de anskaffar sig om systemet, blir mindre lika medelanvändaren.

3.6 Design av hela gränssnittet

Det viktigaste man ska tänka på när man designar ett gränssnitt är att man ska vara konsekvent. Det gäller även användarmanualen och all dokumentation av systemet. Konsekvensen gör att det är

lättare för användaren att känna sig bekant med systemet och på så sätt tycka att det är bra.

Vid utveckling av stora system gäller det att ha tydliga rekommendationer och riktlinjer för hur man ska uttrycka termer och för hur vanligt förekommande objekt ska se ut.

3.7 Riktlinjer och heuristisk utvärdering

Heuristisk utvärdering är ett sätt att utvärdera ett gränssnitt genom att titta på tumregler som forskare och designers utarbetat. [Alwood, 1998]

Det finns riktlinjer för många olika sorters gränssnitt som har arbetats fram genom mångårig erfarenhet och av forskning. Dessa riktlinjer kan användas som stöd för en heuristisk analys.

3.7.1 Riktlinjer för haptiska användargränssnitt

Calle Sjöström [Sjöström, 2002] har sammanställt riktlinjer som man kan använda då man designar ett haptiskt användargränssnitt för blinda personer. De riktlinjerna, med exempel på vad de innebär är:

- Utarbeta en virtuell objektdesign för sig
 - Undvik objekt med små och spridda ytor
 - Använd rundade hörn
 - Optimera den haptiska applikationen. Tänk på affordance
 - Se till att modellerna är haptiskt korrekta och fungerar utan att man ser
- Framhåll navigation och överblick
 - Sträva efter väldefinierade referenspunkter som är lätta att hitta
- Tillför kontextuell information
 - Presentera den haptiska modellen i sin naturliga kontext
 - Tillhandahåll information om programmets syfte
 - Tillhandahåll information om möjligheter och svårigheter i miljön
- Utnyttja alla tillgängliga modaliteter
 - Kombiner haptik med ljud, brailledisplayer och/eller talsyntes för att hjälpa till att identifiera objekt
 - Sträva efter feedback till användaren för alla tillgängliga sinnen

- Stöd användaren i att lära sig interaktionsmetoden, den specifika miljön och programmet
 - Var konsekvent
 - Ge klar och tydlig feedback

3.8 Prototyp

Idén med en prototyp är att tydliggöra koncept och illustrera idéer som ska kunna utvärderas. En prototyp är ett praktiskt exempel på olika designförslag. En idé är svår att utvärdera då den är väldigt abstrakt och ibland svårförståelig, men en prototyp kan utvärderas på flera olika sätt.

En prototyp behöver inte vara ett datorprogram, utan det går även bra att göra pappersmodeller, så kallade lo-fi prototyper som är minst lika effektiva [Löwgren 1993].

3.9 Utvärdering

Ett gränssnitt behöver utvärderas för att kontrollera om användarna kan använda det och för att ta reda på om de tycker om det [Preece, 2002]. För att göra detta utför man ett användartest där man låter testpersoner göra vissa uppgifter, analyserar resultaten och därefter utvärderar vad som kan ändras för att kunna förbättra gränssnittet så att de fel man har observerat inte uppkommer så ofta.

3.9.1 Planera testet

För att kunna utföra ett test på ett bra sätt krävs planering. Man måste först och främst bestämma vad som är målet med testet. Vad är det man vill uppnå med det? Man bör också beräkna hur lång tid det tar att genomföra det, i vilken lokal man ska vara, hur många testanvändare som behövs och hur man bestämmer att testet är avklarat.

3.9.2 Pilottest

För att veta att man lyckats i planeringen av testet är det en god idé att utföra ett pilottest på en eller två personer. Detta gör man för att kontrollera att testet fungerar. Testpersonerna till pilottestet behöver inte vara de som kommer att använda programmet senare, utan det räcker med en kollega eftersom det inte är programmet i sig man ska testa.

Ofta märker man i pilottestet att tidsplaneringen inte stämmer, vissa uppgifter är svårare än man anat, andra är lättare.

3.9.3 Genomföra testet

Ett användartest har fyra faser:

- Förberedelse – Ställ upp all utrustning som behövs. Testa att tekniken fungerar, skärmläckaren är avstängd och att andra saker som kan störa testet är bortkopplade.
- Introduktion – Berätta för testpersonen om testet. Informera om att testet inte är till för att bedöma personens färdigheter utan att det är programmet/gränssnittet som ska utvärderas. Testet är frivilligt och det är tillåtet att avbryta det när helst testanvändaren vill det.
- Testet – Genomförandet av testet bör ske utan för mycket inblandning av testledaren. Det är viktigt att vänta på testanvändaren och försöka låta testpersonen lösa problemen själv.
- Debriefing – Gå igenom vad som gjorts, fråga om hur testpersonen upplevde testet och programmet. Fråga testanvändaren om hon har några förslag på förbättringar av programmet.

3.10 Iterationer

Det är inte vanligt att man på första försöket lyckas nå alla sina mål. Därför gör man många iterationer i sin utvecklingscykel. Tidigare arbetade man ofta med den s.k. ”vattenfallsmodellen”, som är en linjär modell för utveckling av mjukvara. Nu har man övergått till en modell kallad spiralmodellen istället, där man itererar flera varv och utvärderar kontinuerligt för att veta när man uppnått sina mål [Preece, 2002].

4 Metod

Det första jag gjorde var att fundera ut vad jag egentligen skulle göra. Det hela var ganska vagt formulerat och det gick åt mycket tid för att jag skulle lära mig hur Reachin API fungerar. Det har en bristande dokumentation och det finns inte några speciella handledningar som förklarar hur man ska använda API:t. Man måste istället läsa i dokumentationen och sedan testa.

Då det inte är så många som använder API:t är det också svårt att hitta folk på Internet som har stött på liknande problem och på så sätt få hjälp. Det finns ett diskussionsforum på Reachins hemsida, men det är inte så frekvent använt.

4.1 Utgångspunkt

Jag har utgått från ett befintligt program där man ritar i svartvitt och kan känna det man ritat. Detta har jag utökat med en ”meny” för färgerna och olika höjder som gör att man ska känna de olika färgerna.

Då jag utgick från ett redan skrivet program var det första stora problemet att förstå hur det fungerade. Detta var inte så lätt eftersom jag inte hade läst C++-kod på över ett år och det ju aldrig är lätt att sätta sig in i kod skriven av andra. Detta löstes under tiden, men var en stor stötesten i början.

Jag fick studera en hel del exempel och skriva egna små program innan pusselbitarna sakta med säkert började trilla på plats och jag började förstå hur Reachin API och VRML fungerar.

4.2 Lära känna användarna

Det största gemensamma draget hos användarna till programmet jag skulle utveckla var att de var blinda. Detta är deras i princip enda gemensamma nämnare och det är svårt att veta exakt vem som kommer att använda programmet i framtiden.

För att lära känna användarna gjorde jag en telefonintervju med en blind dam för att höra vad hon tyckte om idén samt hur hon förhöll sig till att använda bilder (se Bilaga A). Resultatet av intervjun var att hon tyckte att hon varit blind så länge att hon inte längre tänkte i bilder och kände inget större behov av att rita. Hon tyckte att jag skulle försöka få tag på en person som varit seende för inte så länge sedan för att få bättre svar. Jag letade efter en sådan, men fick tyvärr inte tag på någon. Det hade varit mycket intressant att prata med en person som varit van vid att kunna använda bilder men som nu är förhindrad att göra det.

Jag har även medverkat vid ett användartest som Charlotte Magnusson på Certec utförde till sitt Trafikmiljöprojekt och på så sätt fått studera och prata med en blind person.

4.3 Vad ska man kunna göra med programmet?

Då jag inte fick tag på någon ”nyblind” användare att fråga fick jag tänka ut funktionaliteten själv. Detta är teoretiskt inte speciellt bra, men då jag fått en ”beställning” av Certec så visste jag vad de ville att programmet skulle göra.

Programmets huvuduppgift är att man ska kunna rita med olika färger som i den icke-visuella delen ska kunna kännas. Det viktigaste är att det ska vara en tydlig avskiljning mellan de olika fälten så att man lätt ska kunna skilja på de olika färgerna. Detta var också det stora målet med programmet.

4.4 Analys av tidigare program

Då det utvecklats ett program ”Rita med fingrarna” på Certec tidigare var det den enda konkurrenten. Detta program var så gammalt att det inte ens gick att provköra på dagens datorer. Jag fick därför nöja mig med att Charlotte Magnusson på Certec sa att det var svårt att känna var de olika färgfälten började och slutade då det saknades tydlig avgränsning.

Det verkar inte existera något kommersiellt haptiskt ritprogram och sökningar på Internet resulterade inte i något.

Andra möjligheter som finns för blinda personer att tillgodogöra sig bilder är taktila bilder, det vill säga bilder man känner på. Dessa utformas enligt vissa riktlinjer som jag studerade och jag läste litteratur om hur det fungerar till exempel ”Att känna bilder” av Yvonne Eriksson [Eriksson, 1997].

4.5 Design av programmet

Att rita med olika färger är ett av huvudmålen med programmet och jag försökte att få det att fungera på många sätt och vis, men det enda jag lyckades ändra på var hur högt det jag ritade kändes. Detta var ju i och för sig bra att veta, då det kom till användning senare i utvecklingsprocessen.

Vid närmare undersökning av hur exempelprogrammet jag utgått från fungerar, visade det sig att det använde sig av en bild man ritade på och som var sammankopplad med den bilden man känner på. Om man istället använde sig av en visuell bild, som är i färg och en haptisk bild som man känner, kunde man ha den

visuella bilden med stöd för färg och den haptiska som stöder känsel. Jag gick alltså från att i exempelprogrammet ha en bild där de visuella och haptiska var sammankopplade till att ha två bilder: en synlig och en man bara känner på, som representerar samma sak på skärmen. Vid uppdateringar måste man rita på båda bilderna så att det både ska synas och kännas.

4.5.1 Parallell design

Vid designen av programmet har jag inte använt mig av den sorts parallella design som beskrivits i teoriavsnittet (se 3.4) av den enkla anledningen att jag gör examensarbetet själv och på så sätt inte kan involvera flera personer i designprocessen. Det hade självklart varit bättre om det funnits olika förslag om hur gränssnittet skulle se ut, speciellt då det inte existerar så många riktlinjer för program med känselgränssnitt och det inte existerar några ritprogram sedan tidigare.

4.5.2 Medverkandedesign

För att kunna göra ett bra program för blinda personer måste blinda utvärdera och säga vad de tycker om olika designförslag. Jag har under utvecklingen inte haft någon blind jag har bollat idéer med, eftersom jag har ansett att det är enklare att få synpunkter på en konkret prototyp. Jag har under utvecklingsarbetet märkt hur svårt det är att förklara exakt vad jag gör utan att visa, därför har jag bedömt att det skulle bli väldigt svårt att få bra input utan att visa en konkret prototyp.

4.5.3 Programmets gränssnitt

För att göra det så enkelt som möjligt valde jag att efterlikna ett kommersiellt ritprogram. Jag har en rad med knappar till vänster som man kan trycka på för att välja färg. Den vita knappen representerar en suddfunktion.

Resten av området fungerar som rityta. Hela programmet är inbyggt i en ”skokartong” för att användare inte ska kunna halka över kanten och tappa orienteringen. Om man trycker på skokartongens väggar hörs ett ljud som bekräftar att man rört vid väggen.

4.5.4 Att känna skillnad på färgerna

För att kunna känna skillnad på färgerna har jag utvecklat två parallella lösningar.

- En som bygger på att varje färg har en höjd och att man med de olika höjdernas hjälp ska kunna skilja dem åt. Knapparna har samma höjd som färgen.
- En som har en textur per färg. Denna version har även höjdskillnaden implementerad för att man ska kunna skilja på

de olika färgfälten. Här har knapparna samma textur som färgen de representerar.

4.5.5 Svårigheter textur

Under en stor del av utvecklingsarbetet ville jag ha en lösning med textur på färgfälten, men det visade sig vara krångligt då jag inte riktigt behärskade API:t till fullo. Därför kom lösningen med höjderna till som ett alternativ till textur, vilket visade sig vara riktigt bra.

Nu var det så bra att jag kom på hur man på ett enkelt sätt får textur på det man ritar så därför finns den lösningen tillgänglig för användartestning.

4.6 Genomgång av riktlinjer

För att kontrollera att jag var på rätt väg analyserade jag mitt gränssnitt med avseende på Calle Sjöströms (se 3.7.1) riktlinjer.

4.6.1 Utarbeta en virtuell objektdesign för sig själv

Man bör se till att man inte har taggiga objekt, utan de ska vara rundade. Det man ritar har tyvärr ganska taggiga sidor då den ritade bilden bygger på att man manipulerar pixel för pixel. Jag lämnar till användartestet för att se om det kommenteras av användarna där. Jag har försökt mig på att rätta till det men inte riktigt hittat någon bra lösning på problemet.

Jag har tänkt på att knapparna ska vara relativt stora och de ger haptisk affordance när man trycker på dem. Modellen fungerar bra även om man inte tittar på den.

4.6.2 Framhåll navigation och överblick

Man befinner sig inuti en ”skokartong” för att underlätta orientering. Ritytan ligger lägre ner än knapparna så att man ska känna skillnad på var knappraden och var ritytan är.

4.6.3 Tillför kontextuell information

Så som programmet ser ut nu finns det inte någon kontextuell information tillagd, eftersom jag inte har koncentrerat mig på den delen av programmet utan istället satsat min tid på själva ritningen.

När testerna har utförts har jag istället berättat hur programmet ser ut och hur det fungerar. I ett färdigt program hade man kunnat spela upp ett intalat meddelande som beskriver programmet och dess funktioner. Man skulle kunna välja om man vill höra meddelandet varje gång man startar programmet.

4.6.4 Utnyttja alla tillgängliga modaliteter

Det finns en version av programmet som har ljudfeedback när man trycker på en knapp. Det finns en idé om att det ska vara ljudfeedback även på själva bilden för att lätt ta reda på vilken färg fältet har.

4.6.5 Stöd användaren i att lära sig interaktionsmetoden och den specifika miljön och programmet

Programmet är enkelt uppbyggt. Det enda man kan göra är att trycka på de sex knapparna i vänstra kanten, samt trycka på Fantomenknappen när man vill rita.

4.7 Prototyp

Efter den heuristiska utvärderingen är prototypen färdig att testas. Då programmet ska vara haptiskt finns det ingen möjlighet att göra en lo-fi-prototyp utan hela programmet måste implementeras på riktigt. Detta gör att det inte kommer att finnas tid till att göra många iterationer, utan jag får nöja mig med att analysera resultaten och ge förslag på vad som kan vidareutvecklas.

4.8 Iterationer

Programmet är inte färdigt än, utan det finns en hel del att lägga till. Jag har gjort den första iterationen i designcykeln och kommer inte att göra fler. Däremot hoppas jag att det kommer någon och tar vid där jag slutade och fortsätter att iterera så det blir ett riktigt bra program till slut.

5 Användartest

5.1 Utförande

Testen har utförts på tre testpersoner. De är alla tre blinda och två av dem har använt Fantomen tidigare. De har fått testa två olika varianterna av programmet (se 4.5.4) enligt testprotokollet (se Bilaga B).

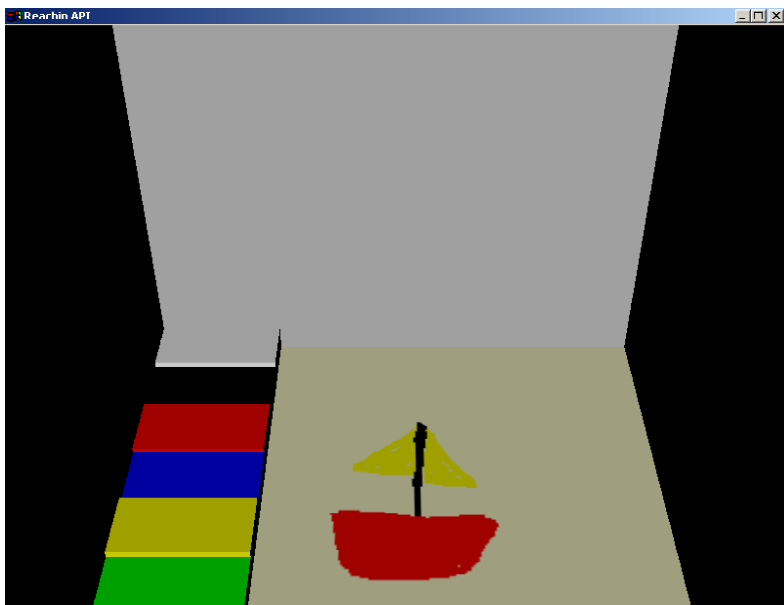
Först får testanvändarna bekanta sig med miljön och känna hur programmet är uppbyggt. Därefter genomförs testen på de två olika varianterna av programmet och avslutas med en diskussion om det allmänna intrycket testpersonen har fått av programmet.

5.2 Man, 23 år

Den första försökspersonen är en man, 23 år gammal som använt Fantomen två gånger tidigare. Han har vana av att använda taktila kartor och han har även använt sig av taktila bilder då han gick i skolan.

5.2.1 Identifiera former

Mannen tyckte det var lätt att identifiera de olika formerna. Det tog 7-10 sekunder för honom att identifiera en form sedan han börjat känna på den. Han tyckte det var lätt; två på en skala mellan ett och fem där fem är svårast.



Figur 2: Båt

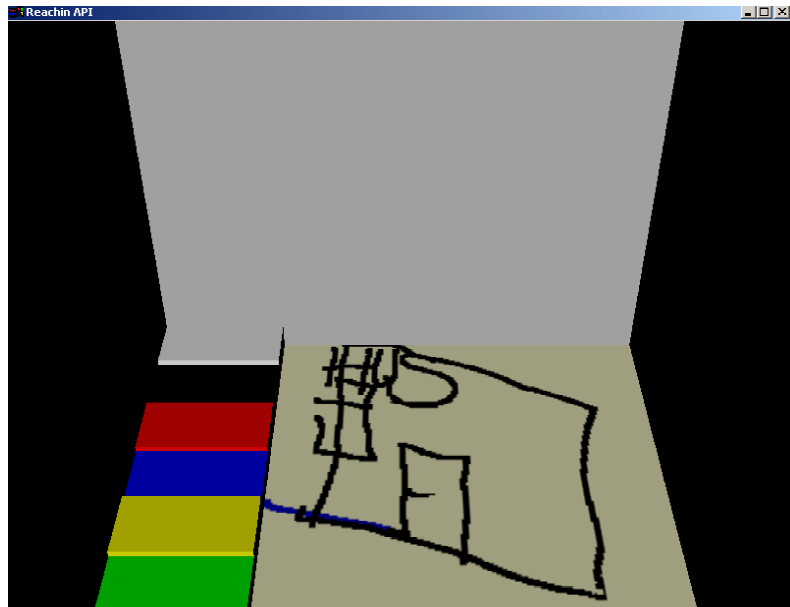
5.2.2 Identifiera en ritad bild utan textur

Mannen tyckte det var svårt att identifiera en bild utan att få reda på något om den. Bilden på en båt (se Figur 2) gissade han var ett träd och därefter en måne med ett träd. Då han fick en ledtråd att det var en farkost sa han direkt att det var en båt. Det tog allt som allt en minut och tjugo sekunder för honom att få fram rätt svar.

Han bedömde det som svårt att veta vad det var, men med lite vägledning var det mycket enklare.

5.2.3 Rita en bild utan textur

Nästa testfall innebar att mannen skulle rita ett hus. Detta gjorde han utan problem och resultatet finns att skåda i Figur 3. Dock tyckte han att det för säkerhets skull var bäst att han även skrev hus på bilden.



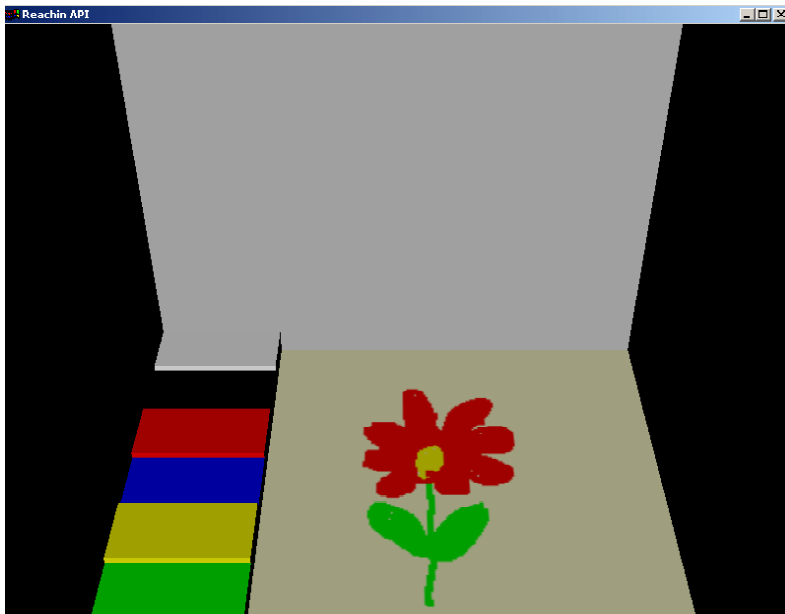
Figur 3: Hus av man, 23 år

5.2.4 Identifiera texturer

Han tyckte det var för liten skillnad mellan de olika texturerna och tyckte att det borde vara någon randig textur med, för att göra det enklare att särskilja de olika texturerna. Trots att han tyckte det var svårt, fyra på en skala upp till fem, lyckades han på en och en halv minut identifiera alla färgerna. Han föreslog att texturerna inte skulle vara så grova så att man lättare skulle kunna glida över dem.

5.2.5 Identifiera en ritad bild med textur

Nästa uppgift var att känna på en blomma (se Figur 4) och gissa vad det var. Han gissade på löv och ett stekt ägg. Han var svårt och fick en ledtråd om att det var en växt efter två minuter. Då gissade han direkt på blomma. Han tyckte det var mycket svårt och gav svårigheten betyget fem på en femgradig skala.



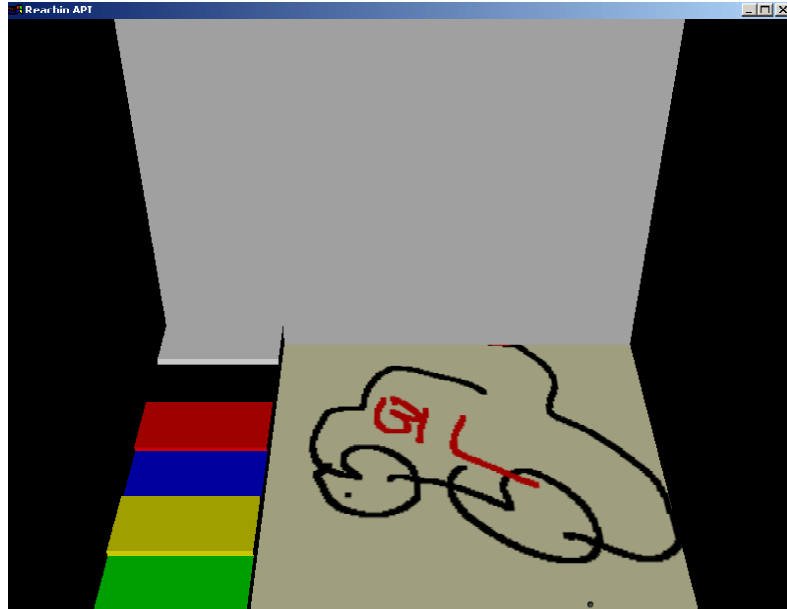
Figur 4: Blomma

5.2.6 Rita en bild med textur

Att rita en bild med textur tyckte han inte var någon större skillnad jämfört med utan textur. Uppgiften var att rita en bil och den finns i Figur 5.

5.2.7 Allmänna kommentarer om programmet

Mannen tyckte det var svårt att sudda. Knapparna man väljer färg med tyckte han inte borde vara tryckkänsliga på sidan. Dessutom borde de ha en kant som gör att man inte kan glida från knapparna ner på ritytan. Han hade även gärna sett en kant mellan knapparna så man inte kunde höra ljudet från de olika knapparna på samma gång. Vidare tyckte han idén med att ritytan lutar var bra, men att den tyvärr gjorde det svårare att rita på den. Knapparna däremot fick gärna fortsätta att luta.



Figur 5: Bil av man, 23 år

5.2.7.1 Förslag på utökningar

Han tyckte det behövdes en hel del fler funktioner för att det skulle gå att använda programmet på ett effektivt sätt. Funktioner han efterlyste var en metod för att kunna rensa hela ritytan och en roller så man kan rita stora fält på en gång. Han hade förslag på en färgidentifieringsmetod som skulle kunna stängas av och på för att känna på bilden och få aktuell färg uppläst. Han hade även önskemål om att man skulle kunna flytta föremål och att man skulle kunna lägga färger i lager för att kunna känna på sidan av det upphöjda. Andra förslag var att ha ett större ritbord, uppbyggt av många små. Det som finns nu tyckte han var lagom stort att överblicka. Han tyckte det hade varit bra om man kunde välja olika paneler, till exempel en med färger, en med ljud, en med filegenskaper och så vidare.

5.2.7.2 Nyttan med programmet

Han tyckte att programmet hade varit roligt att ha och kom på massor med förslag till vad man kan göra med det och utöka det med. Till exempel skulle man kunna rita in en resväg på en befintlig karta, känna på befintliga bilder, lägga till ljudillustrationer på bilder för till exempel barn, lägga till ljud på kartor, spela luffarschack och göra ett konstverk med känselupplevelse.

5.3 Kvinna, 29 år

Den andra försökspersonen är en kvinna som är 29 år gammal och aldrig har använt Fantomen tidigare. Hon har god vana av att använda sig av taktila bilder, även om det inte sker så ofta nuförtiden, det var mer när hon gick i skolan och när hon utbildade sig.

5.3.1 Identifiera former

Kvinnan tyckte det var lätt att identifiera de olika formerna. Det tog cirka 20 sekunder per form, eftersom hon var väldigt noggrann och ville vara helt säker på vad det var. Hon tyckte inte det var speciellt svårt, två på en femgradig skala. Hon tyckte att triangeln som skulle vara liksidig gav intryck av att inte vara det, likaså kommenterade hon att den runda formen inte var helt rund, utan lite rak på ena sidan.

5.3.2 Identifiera en bild utan textur

Kvinnan tyckte det var lätt att identifiera båten jag ritade upp och gav det svårighetsgrad ett. Hon kände på konturerna och sa sedan efter 15-20 sekunder att det var en segelbåt. Hon kommenterade att anledningen till att det gick så fort var att hon tyckte om båtar och att hon gärna ritade dem själv, så hon hade lätt för att känna igen formen då den var en referensform för henne.

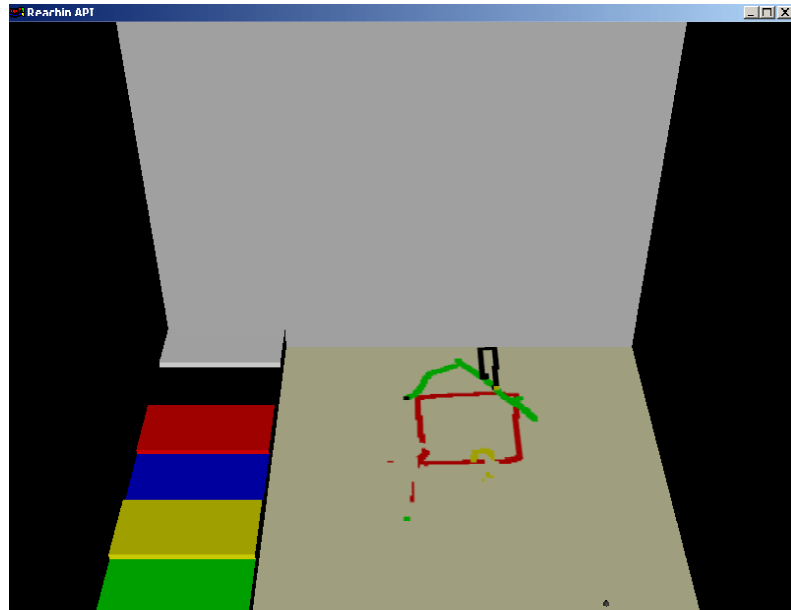
5.3.3 Rita en bild utan textur

Nästa steg var att rita en bild på ett hus. Hon började först rita på ett ställe på ritbordet, men det blev för nära knappraden så hon valde att börja om från början efter en stund. Hon var väldigt noggrann när hon mätte ut husets storlek för att inte linjerna skulle korsa varandra. Dock tyckte hon det var svårt eftersom man inte känner var linjerna är när man håller ner Fantomenknappen då man ritar. Hon ville väldigt gärna att bilden skulle bli fin och kommenterade att svart tak var tråkigt, därför valde hon grönt istället så det blev ett ärgat koppartak. Resultatet finns att beskåda i Figur 6.

5.3.4 Identifiera texturer

Hon tyckte det var väldigt svårt att identifiera de olika texturerna, mellan fyra och fem på den femgradiga skalan. Hon höll på i ungefär 10 minuter med att försöka lista ut vilken som var vilken. Hon gissade att den röda var svart eller gul, den svarta klarade hon av, den gula chansade hon på var blå, den blåa trodde hon var grön och den gröna var blå. Hon kommenterade redan då hon fick känna på texturerna att det skulle bli svårt att känna skillnad på

dem, speciellt mellan blå och grön. Hon kommenterade även att hon inte tyckte det var så roligt att känna på de olika texturerna, det var roligare att känna på former.



Figur 6: Hus av kvinna, 29 år

5.3.5 Identifiera en ritad bild med textur

Jag ritade upp en blomma och kvinnan fick känna på den och sade efter cirka 15 sekunder att det var en blomma. Hon använde sig bara av konturerna för att identifiera vad det var. Hon gav själva identifierandet av formen svårighetsgrad två och färgerna orkade hon inte med eftersom hon tyckte det var svårt redan innan.

5.3.6 Rita en bild med textur

Hon fick uppgiften att rita en bil, hon kommenterade direkt att hon inte tyckte bilar var så roliga men ritade lite ändå. Hon orkade inte göra den riktigt färdig. Bilen finns att beskåda i Figur 7.

5.3.7 Allmänna kommentarer om programmet

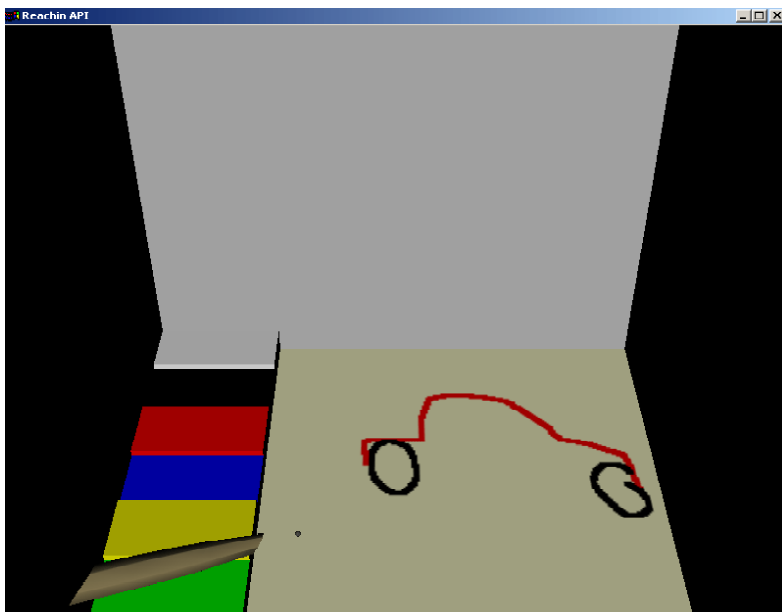
Hon tyckte det var lätt och roligt att använda programmet. Hon tyckte att texturerna och färgerna var överkurs för henne just nu och hon tyckte det var bra att kunna rita och känna på endast konturerna. Hon poängterade att man nog kunde lära sig att använda texturerna och uppskatta dem, men inte efter första försöket.

Hon såg programmet som en stor tillgång till exempel i skolan där man lätt hade kunnat få en överskådlig bild över olika föremål.

5.3.7.1 Förslag på förändringar

Hon tycker att texturerna bör ändras om de ska vara med och hon gav förslag på olika randiga mönster. Höjderna på de olika färgfälten borde vara större för nu var det för liten höjdskillnad tyckte hon. Hon ansåg att taggigheten gjorde det svårt att känna hur man dragit en linje och känna exakt var två linjer skär varandra.

Hon föreslog att det hade varit bra med en funktion för att spara bilder och läsa in dem och känna på dem igen.



Figur 7: Bil av kvinna, 29 år

5.4 Man, 45 år

Den tredje försökspersonen är en man som är 45 år gammal och har använt Fantomen två gånger tidigare. Han har god vana av att använda taktila bilder och är intresserad av konst.

5.4.1 Identifiera former

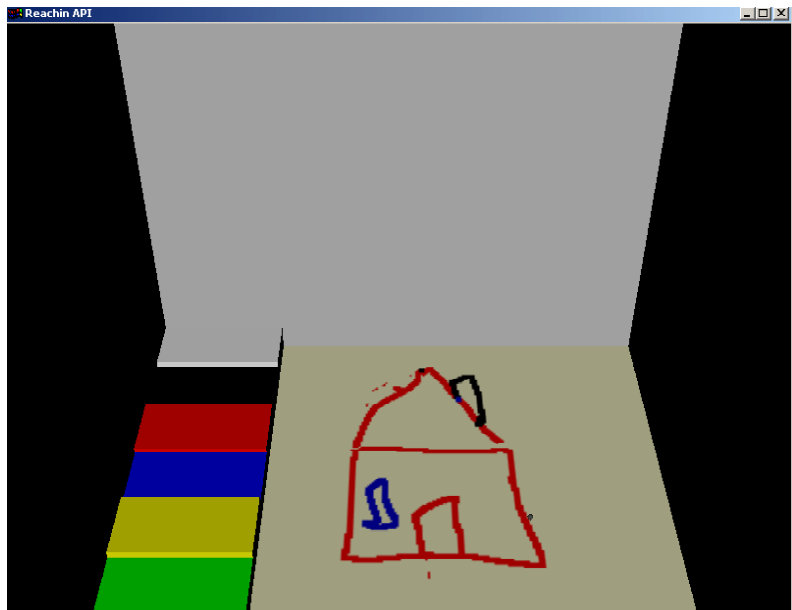
Att identifiera de olika formerna tyckte han inte var så svårt, ett på en femgradig skala. Han hittade de olika formerna på ungefär 20 sekunder per form.

5.4.2 Identifiera en bild utan textur

Båten tog ungefär 1,5 minut att identifiera. Han gissade på att det var en segelbåt direkt och han bedömde det som ganska svårt men gav det en trea på skalan.

5.4.3 Rita en bild utan textur

Han ritade snabbt och lätt ett hus när jag bad om det. Han hade mycket bra känsla för avstånd och använde sig av sudd/vit färg med mycket bra precision. Bilden han ritade finns i Figur 8.



Figur 8: Hus av man, 45 år

5.4.4 Identifiera texturer

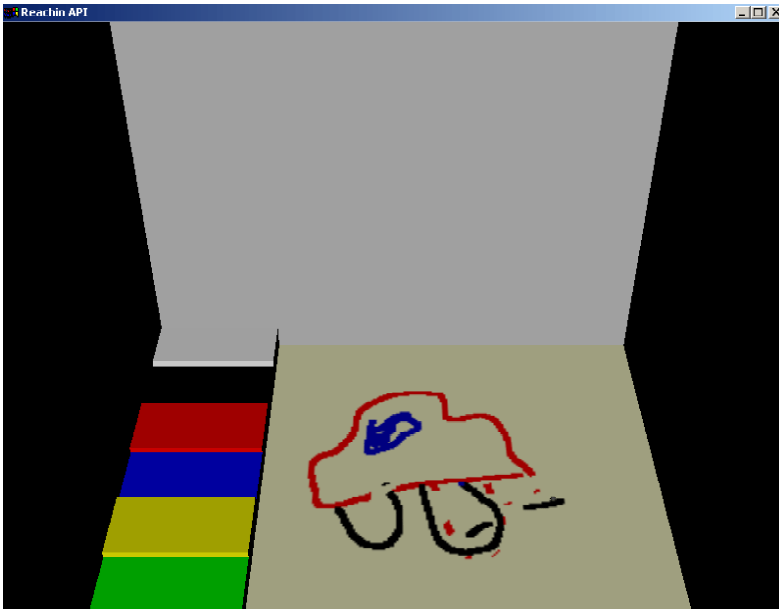
Han tyckte det var svårt att identifiera de olika texturerna. När han kände på dem på de olika knapparna kommenterade han att det var för liten skillnad mellan de olika förutom den svarta. Det var den första färgen han försökte identifiera och det lyckades han med på 15 sekunder. Sedan försökte han lite med de andra men han tyckte det var för svårt och avbröt.

5.4.5 Identifiera en bild med textur

Därefter var det dags att försöka identifiera en blomma som jag ritat upp. Det tog ungefär 2 minuter innan han sa att det kändes som en sol och jag bekräftade att formen stämde. Då gissade han direkt på en blomma. Han tyckte det var svårt, fyra på den femgradiga skalan.

5.4.6 Rita en bild med textur

Att rita en bil gjorde han med samma precision som huset. Det märktes att han började tröttna lite och hans ledarhund tyckte det var tråkigt att ligga och vänta. Han klagade även över krampkänslor i armen. Men en bil blev det ändå. Den finns i Figur 9.



Figur 9: Bil av man, 45 år

5.4.7 Allmänna kommentarer om programmet

Han tyckte programmet skulle fungera bättre om man använde sig av en Fantom med fingertuta istället för den med pennfattning, eftersom han tycker man har en annan typ av känsel i fingertopparna än då man använder sig av pennan.

Han föredrog att använda den första versionen av programmet men trodde att den andra, den med textur, är bättre om det är större skillnad på texturerna så att man lätt kan skilja dem åt.

Han tyckte programmet överlag är lätt att använda och är uppbyggt på ett bra sätt. Han trodde att det skulle vara bra att ha när hans dotter (nu 5 månader) blir äldre och de då vill rita tillsammans.

5.4.7.1 Förslag på förändringar

Han tyckte det skulle vara bra om det fanns ett ljud som berättade när man befinner sig på ritytan och även att man får färgen på det som är ritat på ytan uppläst om man rör på det.

Han tyckte att den taggighet som uppstår när man ritar gör det svårt att känna och att det bör åtgärdas.

Han tyckte inte om att ritytan lutar. Antingen ska den vara lodrät eller vågrät, inte mittemellan som den är nu.

Ett annat önskemål var att man skulle kunna rita tredimensionella objekt istället för bara tvådimensionella.

5.5 Resultat av testen

- Texturerna i versionen med texturer är för lika varandra. För att man ska kunna använda sig av dem ska det vara mycket större skillnad
- Ritytan ska vara horisontell och hela programmet ska vara längre ner så att man inte behöver anstränga armen.
- Det är svårt att säga något om vilken av de två versionerna som är bäst, eftersom skillnaderna mellan texturerna inte är tillräckligt stor.
- Det är lätt att använda programmet.

6 Framtid och förslag till utökningar

Programmet är inte en färdig produkt än, då det inte har funnits tid till att göra de iterationer som krävs för att det ska bli ett lättanvänt program som fungerar som det ska. Det finns vissa brister och saknas en del funktionalitet som jag kommer att ta upp i detta avsnitt som hjälp till vidare utveckling.

6.1 Brister

Det finns en del saker som inte riktigt fungerar som det ska. Vissa saker kan rättas till, andra beror på brister i Reachin API.

6.1.1 Färgbyte

När man byter färg med hjälp av knapparna hänger programmet inte riktigt med. Då man börjar rita med den nya färgen kommer det att bli ett ”färgstänk” av den nya färgen där man sist ritade med den förra färgen. Detta är inte så bra utan gör att man får små prickar av fel färg i sin bild och det bör ändras.

6.1.2 Hål i ytorna

Ibland faller man igenom ritytan eller knapparna trots att det inte ska vara möjligt. Detta sker främst vid knapparna och gör att man kan fastna inuti en knapp och då måste man starta om programmet. Detta är inte så lustigt då allt man ritat går förlorat.

6.2 Förslag till utökningar

Det finns mängder av förslag till vad programmet ska kunna utföra.

- Spara bilder
- Läs in sparade bilder
- Höra vad det är för färg bilden har om man trycker på ritytan
- Större rityta
- Fördefinierade former att rita ut
- Olika paneler för olika funktioner

6.3 Framtid

Det verkar som om det finns massor med idéer om vad man kan använda programmet till och hur man kan utveckla det vidare. Om man tittar på vanliga ritprogram som Paint Shop Pro eller Illustrator finns det en hel del funktionalitet som man skulle kunna applicera på min applikation. Jag tror det enda som krävs för att det ska finnas en säker framtid för programmet är att det finns någon som har tid att vidareutveckla det.

7 Tankar och erfarenheter

Nu har det blivit ett program som fungerar efter en termins arbete. Det är alltid intressant att stämma av vad man lärt sig efter att ha utvecklat något.

Det är alltid svårt att veta precis vad man lärt sig, men jag tycker det finns en del saker som är uppenbara, andra kommer jag att komma på senare och vissa tänker jag inte på alls att jag lärt mig.

7.1 Tid

Det första jag kommit att tänka på är hur svårt det är att göra en tidsplanering och att sedan hålla den. Det är så många faktorer som spelar in och man är alltid för optimistisk även om man lägger in reservtid. Detta examensarbetet har innehållit en hel del moment där det gäller att lära sig nya saker och inlärningstid är något som är svårt att planera tid för.

Jag visste redan från början att Reachin API har dålig dokumentation och att det därför tar lång tid att lära sig det. Men så lång tid trodde jag inte det skulle ta. Det tog inte lång tid innan jag kunde använda API:t till enklare saker, men jag kände inte att jag började behärska det förrän jag var nästan färdig med implementationsdelen.

7.2 Sätta sig in i blindas situation

Jag har under mitt arbete märkt att många har förutfattade meningar och inte tror att blindas har ett behov av bilder och att rita. Jag var inte skeptisk till blindas bildbehov innan jag började med arbetet, men jag hade ingen kunskap om det förutom att jag visste att svällpapp fanns. Jag har lärt mig enormt mycket om hur blindas vardag fungerar och vilken hjälp de kan behöva. Men samtidigt har jag förstått att det går att klara sig väldigt bra trots att man är blind. Det har jag speciellt märkt när jag träffat mina trevliga och ambitiösa testanvändare.

Något jag lärt mig mycket om är också hur mycket vi seende använder vår syn och hur svårt det är att sätta sig in i hur det är att inte se.

7.3 Kontinuerlig dokumentation

Jag är väldigt glad att jag under tiden jag arbetat har fört en ”dagbok” där jag skrivit ner tankar och funderingar om mitt arbete. Detta gör att det är så mycket enklare att dokumentera arbetet efteråt. Dessutom blir det mycket enklare att kunna gå tillbaka och motivera varför man egentligen gjorde sina olika val av lösningar.

7.4 Testsituationen

Att utföra ett test är inte så enkelt. Det krävs ordentligt med förberedelser och rejäl tanke bakom de olika testuppgifterna. Detta trodde jag inte skulle ta så lång tid, men det krävs mycket mer än vad man tror för att utarbeta ett test som svarar på det man vill.

Jag tyckte också att det var svårt att utföra testet på samma sätt de olika gångerna, speciellt de sista. Det är så lätt att man kommenterar något som hänt i ett tidigare test och att man på så sätt gör testförutsättningarna annorlunda. Likaså var det frustrerande i det tredje testet där jag med stor sannolikhet visste att det skulle bli problem med texturerna, man vill ju visa upp ett bra program och så vet man att det är svårt att känna skillnad på texturerna. Jag hade gärna lagt in nya texturer att testa efter första testet, men om jag gjort det hade testen inte varit likadana och inte gått att jämföra. Texturerna hade nog inte varit så svåra om jag haft en blind person som hade testat de olika texturerna innan jag lade in dem i programmet.

Referenser

Tryckta referenser

Alwood, C.M. (1998) *Människa-datorinteraktion*. Lund: Studentlitteratur

Axel, E.& Levent, N.(red.)(2002) *Art beyond sight*. New York: American Doudnation for the blind

Eriksson, Y. (1997) *Att känna bilder*. Solna: Statens institut för handikappfrågor

Löwgren, J. (1993) *Human-computer interaction*. Lund: Studentlitteratur

Nielsen ,J. (1993) *Usability Engineering*. San Fransisco: Morgan Kaufmann

Preece, J. (2002) *Interaction Design*. New York: John Wiley & Sons

Rassmus, K. (1996) *Projektarbeten i rehabiliteringsteknik 1994-1995*

Reachin Technologies AB (2001) *Programmer's guide*

Sjöström, C. (2002) *Non-Visual Haptic Interaction Design*

Skanse, J. (2002) *C++ Direkt*. Lund: Studentlitteratur

Stroustrup, B (2000). *Programmeringsspråket C++*

Elektroniska referenser

Haptik:

http://www.tfe.umu.se/courses/systemteknik/Multimed2/mm2_99/Bok98/Kapitel4/4.2/haptik.html 2003-08-25

Reachin: <http://www.reachin.se>

Bilaga A: Sammanfattning av telefonintervju

Hur gör hon om hon vill titta på en bild eller ett diagram idag?

Hon har oftast någon som berättar för henne hur bilden eller ev. diagrammet ser ut. Hon kan rita små bilder med händerna. Man kan beställa svällpappersbilder från syncentralen, men det är inget man gör till vardags. Hennes man har tex. fått en karta över Malmö i svällpapp.

Hon berättar också att det är väldigt svårt för en seende som inte är van att beskriva en bild att göra det, för vi är inte vana vid att inte se. Så vem som helst kan inte ge en bra beskrivning som gör att hon förstår vad som är på bilden.

Vad finns det för saker som hon skulle vilja rita som hon inte har möjlighet att göra idag?

Svårt att säga. Hon är inte van att se saker och har därför inte något behov av att se dem längre. Hon tror det kan vara en bra idé att tala med någon som inte har varit synskadad så länge som hon och på så sätt fortfarande har kvar en seendes "bildkänsla".

Hon tror även att man kan skapa ett behov av att känna bilder med Fantomen om det finns bra möjligheter att göra det. "Det man inte vet finns saknar man inte"

Slutsats av detta

Hon känner inget större behov av att uttrycka sig i bilder. Men det kan nog finnas användning för ett dylikt program ändå om bara ett behov skapas. Jag bör hitta någon "nyblind" att tala med för andra reflektioner; denna person kanske har ett större behov av att uttrycka sig i bilder.

Bilaga B: Testprotokoll

Användartest

Namn :

Ålder:

Fantomenvana:

Vana av att läsa taktila bilder:

Version 1 (ljud + höjd):

1. Identifiera olika former:

- Tid
- Vad de trodde det var om det är fel
- Svårighetsgrad 1 2 3 4 5

Form	Tid
Cirkel	
Triangel	
Kvadrat	

2. Det kommer att finnas en streckbild uppritad (båt) Försök att identifiera vad den föreställer.

- Tid
- Vad de tror att det är
- Bedöma det svårt att identifiera olika färger
- Svårighetsgrad 1 2 3 4 5

Kommentarer:

3. Rita en bild på ett hus. Försök att använda "rätt" färger.

- Gör en skärmdump av resultatet

Version 2 (ljud + textur):

1. Identifiera olika färger:

(Ritar upp olika blobbar med färgerna, vissa färger finns det två av, andra bara en.)

- Tid
- Vad de trodde det var om det är fel
- Är det några som verkar svåra att blanda ihop?
- Svårighetsgrad 1 2 3 4 5

2. Det kommer att finnas en streckbild uppritad (blomma). Försök att identifiera vad den föreställer.

- Tid
- Vad de tror att det är
- Bedöma det svårt att identifiera olika färger
- Svårighetsgrad 1 2 3 4 5

Kommentarer:

3. Rita en bild på en bil. Försök att använda ”rätt” färger.

- Gör en skärmdump av resultatet

Allmänt:

1. Tror du att du kommer att ha någon nytta programmet?
2. Finns det något du kommer på som man kan använda programmet till?
3. Tycker du det är lätt att använda programmet?

Certecs rapporter

Ett urval av rapporter från CERTEC

Bengtsson, K., Nyström, M

Symbol Support

Examensarbete, Certecrapport 4:2003

<http://www.certec.lth.se/doc/symbolsupport/>

Aili, M., Nilsson, A

Algoritmer för huvudstyrning av dator

Examensarbete, Certecrapport 3:2003

<http://www.certec.lth.se/dok/algoritmerforhuvudstyrning/>

Lindholm, A

Haptisk skattjakt

Examensarbete, Certecrapport 2:2003

<http://www.certec.lth.se/dok/haptiskskattjakt/>

Sjöström, C

Non-Visual Haptic Interaction Design

Doktorsavhandling, Certec 2:2002

<http://www.certec.lth.se/doc/hapticinteraction/>

Mandre, E

Vårdmiljö eller Lärandemiljö?

Om personer med autism inom vuxenpsykiatrin

Doktorsavhandling, Certec 1:2002

<http://www.certec.lth.se/dok/franvardmiljotill/>

Blinda personer har ett behov av att uttrycka sig i och att tolka bilder precis som seende personer. Blinda personers behov är dock åsidosatt, ofta antar seende att blinda personer inte kan rita och inte har ett behov av det - "De ser ju ändå inte vad de ritat för något". För en seende person är det lätt att använda ett ritprogram på en dator. Om man är synskadad eller blind är det inte riktigt lika enkelt att utnyttja ett sådant program eller att få färger och former beskrivna för sig.

Det här examensarbetet beskriver utvecklingen av ett ritprogram med olika färger för blinda och synskadade personer. För att använda programmet utnyttjas the PHANToM, på svenska Fantomen, som är ett haptiskt användargränssnitt som används för att känna virtuella objekt.

Den här rapporten hittar du också på internet:

<http://www.certec.lth.se/dok/haptiskritprogram/>



Avdelningen för
rehabiliteringsteknik,
Inst för designvetenskaper,
Lunds tekniska högskola



Certec, LTH
Box 118
221 00 Lund



Sölvegatan 26
223 62 Lund



046 222 46 95



046 222 44 31



certec@certec.lth.se



<http://www.certec.lth.se>

Certec är en forsknings- och utbildningsenhet inom Institutionen för designvetenskaper vid Lunds tekniska högskola. Främst via Internet gör vi en stark satsning på "Certecmärkt" information.

Vi är ca 25 anställda och har en årsomsättning på omkring 14 miljoner kronor. Basanslag kommer huvudsakligen från Region Skåne och LTH. Projektmedel får vi t ex från KK-stiftelsen och ett flertal andra bidragsgivare.

EXAMENSARBETE CERTEC, LTH NUMMER 5:2003
ISRN CERTEC-ER-03/5-SE

OKTOBER 2003

Christin Hansson

Haptiskt ritprogram