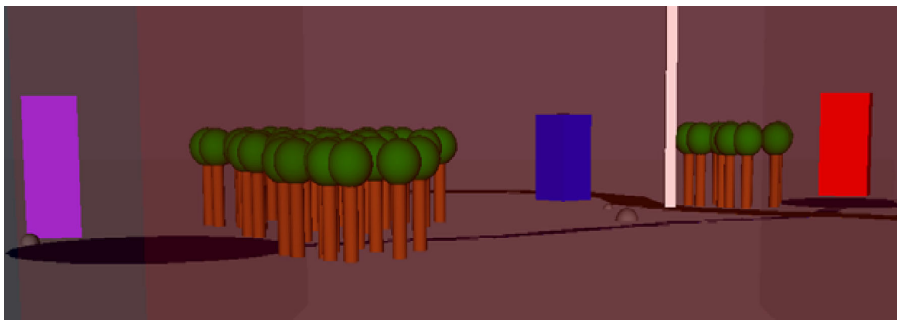
EXAMENSARBETE CERTEC, LTH NUMMER 2:2003

Anna Lindholm

# Haptisk skattjakt



Certec
Avdelningen för rehabiliteringsteknik
Institutionen för designvetenskaper
Lunds tekniska högskola

# Sammanfattning

I detta examensarbete gör jag ett skattjaktsspel vars målgrupp huvudsakligen är blinda och synskadade barn och ungdomar. För att spela spelet används the PHANToM, på svenska Fantomen. Ett av problemen jag ställts inför är att Fantomens räckvidd är begränsad. Därför behöver den virtuella världen flyttas på något sätt utan att den synskadade användaren tappar orienteringen i spelet. För att bestämma hur detta ska ske, och för att se om spelets idéer fungerar, har ett antal användartester utförts.

Fantomen är ett verktyg för att känna virtuella saker i tre dimensioner. Fantomen kan till utseendet liknas vid en penna vars spets sitter på en robotarm. När Fantomen används håller användaren i pennan på samma sätt som en vanlig penna, pennan kan flyttas i alla riktningar och har ett visst motstånd när den stöter i det virtuella föremålet. Att använda Fantomen känns ungefär som att hålla i en penna och dra dess spets längs ett föremål.

Fantomen kan programmeras på olika sätt. Jag har använt Reachin API vilket är ett gränssnitt framtaget av det svenska företaget Reachin för att programmera just Fantomen. Reachin API kombinerar VRML, C++ och Python.

Resultatet av examensarbetet blev ett skattjaktsspel som fungerar, men har vissa brister. Hur spelet fungerar och vad det har för brister diskuteras i denna rapport.

## Nyckelord

haptik, spel, blind, synskadad, Fantomen, the PHANToM

# Abstract

In this master thesis a computer game for blind children and youth is constructed. To play the game the PHANToM is used. One of the main problems is that the range of the PHANToM is finite, and that the virtual world has to be moved. To decide how this will be done and to see if the ideas of the game works, some user tests have been performed.

The PHANToM is a tool that makes it possible to feel virtual things in three dimensions. The PHANToM can to its appearance be resembled to a pen whose nib is on a robot arm. When the PHANToM is used, the user holds the pen in the same way as a normal pen. The pen can be moved in all directions and has a resistance when it touches the virtual object. To use the PHANToM is almost like holding a pen and drag its nib along an object.

The PHANToM can be programmed in some different ways. I have used Reachin API which is an interface developed by the Swedish company Reachin. Reachin API is combined by VRML, C++ and Python.

The result of the master thesis is a computer game where the player has to find a treasure. The game works, but is hard to use in some ways. The game and the way it is hard to use are discussed in this report.

## Keywords

haptics, game, blind, visually handicapped, the PHANToM

# Förord

Detta examensarbete på 20 poäng är det sista steget i min civilingenjörsutbildning. Att få göra mitt examensarbete på Certec har varit min förhoppning sedan jag hösten 1999 läste kursen ”Rehabiliteringsteknik AK” och våren 2000 läste dess fortsättningskurs ”Rehabiliteringsteknik FK”. Detta är två av de roligaste kurserna jag läst under min tid på LTH, förmodligen beror det på att jag har ett stort intresse både för teknik och för människor, vilket kombineras här.

Jag vill tacka Charlotte Magnusson och Kirre Rassmus-Gröhn som varit mina handledare på Certec. Jag vill också tacka Simon Kågström som svarat på massor av programmeringsrelaterade frågor. Ett sista tack går till de som varit mina försökspersoner.

Anna Lindholm, mars 2003

# Innehållsförteckning

**1 Inledning ..... 9**
1.1 Bakgrund ..... 9
1.2 Mål ..... 9
1.3 Innehåll ..... 9
**2 Bakgrundsfakta ..... 11**
2.1 The PHANToM, Fantomen ..... 11
2.1.1 Certec och Fantomen ..... 12
2.1.2 Andra användningsområden för Fantomen 12
2.2 Reachin API ..... 12
2.2.1 VRML – Virtual Reality Modelling Language 13
2.2.2 C++ ..... 13
2.2.3 Python ..... 13
2.2.4 Systemkrav ..... 14
2.3 Vad fanns innan mitt examensarbete? ..... 14
2.4 Synskadade personers vana vid datorer ..... 14
**3 Spelets utveckling ..... 17**
3.1 Problem och lösningar ..... 17
3.1.1 Kartomvandling ..... 17
3.1.2 Hur hitta skatten? ..... 17
3.1.3 Områdesbegränsning ..... 19
3.1.4 Programmeringsspråk ..... 19
3.2 Områdesbegränsning ..... 20
3.3 Användartest ..... 21
3.3.1 Förutsättningar vid användartest ..... 21
3.3.2 Kvinnans test ..... 22
3.3.3 Mannens test ..... 22
3.3.4 Sammanfattning av användartestet ..... 23
3.3.5 Andra iaktagelser under användartestet ..... 23
3.3.6 Resultat av användartestet ..... 24
3.4 Hård- och mjukvarubegränsningar ..... 25
**4 Det färdiga spelet ..... 27**
4.1 Spelstart ..... 27
4.1.1 Utförliga instruktioner ..... 27
4.1.2 Provkarta ..... 28
4.1.3 Starta spelet ..... 28
4.2 Funktioner i spelet ..... 29
4.3 Problem i spelet ..... 30
4.3.1 Ljud ..... 30
4.3.2 Osynliga hål ..... 30
4.3.3 Fantomen slår ..... 31

4.3.4 Fantomen hackar ..... 31
**5 Slutligt användartest ..... 33**
5.1 Försökspersonen ..... 33
5.2 Försökets genomförande ..... 33
5.3 Försöket ..... 33
5.4 Utvärdering av användartestet ..... 35
5.4.1 Instruktioner ..... 35
5.4.2 Termer ..... 35
5.4.3 Stor värld ..... 36
5.4.4 Förflyttning av världen ..... 36
5.4.5 Lång tillvänjningstid ..... 36
5.5 Brister i användartestet ..... 36
**6 Skattjaktsspelets framtid ..... 39**
6.1 Vidareutveckling av spelet ..... 39
6.1.1 Ljud ..... 39
6.1.2 Banor ..... 39
6.1.3 Skatter ..... 39
6.1.4 Verkliga platser ..... 39
6.1.5 Ledtrådar ..... 39
6.1.6 Spara ..... 40
6.1.7 Sakers utseende ..... 40
6.1.8 Tid ..... 40
6.1.9 Poängberäkning ..... 40
6.2 Andra spelidéer ..... 40
6.2.1 Schack ..... 40
6.2.2 Äventyrsspel ..... 41
**7 Spelets tekniska konstruktion ..... 43**
7.1 Världens utseende ..... 43
7.1.1 World.wrl ..... 43
7.1.2 Treasure.wrl ..... 44
7.1.3 Button.wrl ..... 44
7.2 Världens funktionalitet ..... 44
7.2.1 testWorld.cpp ..... 44
7.2.2 World.cpp ..... 44
**8 Tankar runt det färdiga examensarbetet ..... 45**
**Referenser ..... 47**
**Bilaga 1, Programmeringsträd ..... 49**

# 1 Inledning

## 1.1 Bakgrund

Det finns många datorspel för seende personer, men för blinda och synskadade personer finns det bara väldigt få. Detta beror förmodligen främst på att traditionella datorspel bygger på det som visas på datorskärmen, och för den som inte ser är det då omöjligt att spela spelet.

Ett annat problem som blinda personer ofta utsätts för är att det är svårt för dem att hitta i okända miljöer och veta hur de ska bete sig där. En seende person kan lätt titta på en karta för att hitta till ett ställe, motsvarande för en blind person blir då att den behöver känna denna karta. Det finns gott om tvådimensionella kartor som passar för seende, men det finns inte lika många tredimensionella kartor.

## 1.2 Mål

Målet med mitt examensarbete var att kombinera de två fakta jag beskrivit ovan, bristen på datorspel för blinda personer och bristen på tredimensionella kartor. Detta har resulterat i ett skattjaktsspel där spelaren ska försöka hitta en skatt i en ”verklighet”. Till sin hjälp har spelaren en tredimensionell karta. Till skillnad från vanliga datorspel där spelaren ser vad som händer känner spelaren här den värld som spelet utspelas i. För att känna spelet används ”the PHANToM”, eller Fantomen, vilket är ett haptiskt känselgränssnitt. Fantomen beskrivs mer i kapitel 2.

## 1.3 Innehåll

I denna rapport beskrivs problemet för examensarbetet, hur det är löst och vägen dit. Även framtida förbättringar av lösningen diskuteras.

Kapitel 2 innehåller bakgrundsfakta till examensarbetet. I kapitel 3 beskrivs spelets utveckling och i kapitel 4 beskrivs det färdiga spelet. Kapitel 5 innehåller en redogörelse av det slutliga användartestet. I kapitel 6 diskuteras spelets framtid. I kapitel 7 redogörs hur spelet programmerats och som avslutning finns i kapitel 8 tankar runt det färdiga examensarbetet.

# 2 Bakgrundsfakta

## 2.1 The PHANToM, Fantomen

The PHANToM, eller Fantomen, är ett haptiskt känselgränssnitt som är utvecklat på MIT (Massachussets Institute of Technology) och vidareutvecklas nu på ett företag i Cambridge som heter SenseAble Devices Inc[1]. Namnet PHANToM har två olika innebörder. Det är dels en förkortning för Personal HAptic iNTerface Mechanism och dels står det för att den visar saker som inte finns.

Bild 1 Fantomen

Fantomen visar saker i tre dimensioner. Fantomen visar dock inte en bild, istället känner användaren det Fantomen visar. Förutom att bara visa saker med släta ytor kan Fantomen även ge ytor olika struktur, en yta kan kännas olika trög eller glatt och kan ha olika mönster, ytor kan också röra sig t.ex. vågformat. Fantomen kan också visa saker som rör sig. Vad Fantomen visar och hur sakerna den visar ska kännas och uppträda beror på hur Fantomen är programmerad.

På varje Fantom sitter en knapp, till den kan olika händelser kopplas. Knappen kan t.ex. programmeras så att om den inte är intryckt känner användaren alla saker som Fantomen visar, men kan inte komma in i dem, men om den är intryckt går det att komma in i sakerna genom att trycka hårt på någon saks yta.

[1] För mer information, se www.sensable.com.

### 2.1.1 Certec och Fantomen

Fantomen kan användas till många olika saker. På Certec används den främst för att göra applikationer för blinda personer. Ett av Certecs projekt[2] är att bygga upp olika trafiksituationer, t.ex. en gatukorsning med bilar som rör sig och trafikljus, där blinda personer kan träna sig på hur de bör bete sig i trafiken. De kan också lättare få en överblick över hur t.ex. en rondell ser ut och hur den fungerar.

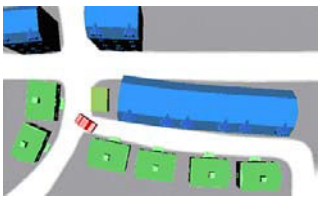


Bild 2 Gatukorsning

Ett annat Certec-projekt är att göra ett ritprogram för blinda och synskadade där de kan rita med Fantomen och sedan känna det de ritat.

### 2.1.2 Andra användningsområden för Fantomen

På andra ställen används Fantomen bland annat för att träna läkare i operationsteknik och titthålskirurgi. Vid detta användningsområde kopplas flera Fantomer ihop.

Fantomen används också för att utforska berggrunder t.ex. när ett djupt hål ska borras.

## 2.2 Reachin API

Det finns två olika programvaror för programmering av Fantomen; Ghost och Reachin API. Jag har valt att använda Reachin API.

Reachin API är framtaget av det svenska företaget Reachin[3]. Reachin har även utvecklat andra produkter med anknytning till Fantomen, jag har dock inte använt någon av dem i mitt arbete.

Reachin API kombinerar VRML (Virtual Reality Modelling Language), C++ och Python. När Fantomen programmeras med Reachin API används vanligen VRML. För riktigt små applikationer, sådana där Fantomen bara ska visa något som inte rör sig och användaren inte kan påverka det Fantomen visar på något sätt, räcker det att använda VRML. Men om det ska hända något med det som Fantomen visar, om något t.ex. ska röra på sig, eller om användaren ska kunna flytta på saker som Fantomen visar måste även C++ eller Python, eller båda två, användas.

---

[2] Mer information om Certecs haptikprojekt finns på http://www.certec.lth.se/haptics/projects.asp.

[3] För mer information se www.reachin.se.

### 2.2.1 VRML – Virtual Reality Modelling Language

VRML är en förkortning av Virtual Reality Modelling Language och används för att beskriva objekt i en tredimensionell värld. Koden kan antingen skrivas manuellt eller genereras av en 3D-modellerare där objekten först byggs grafiskt och sedan exporteras till VRML-kod. VRML kan liknas vid HTML, där HTML används för att presentera text på internet och VRML används för att presentera tredimensionella bilder på internet. Både HTML och VRML kan även användas till andra saker än internetpublicering.

I Reachin API används en modifierad version av VRML97[4], vilket är standarden för 3D-publicering på internet enligt ISO (International Organization for Standardization) och IEC (International Electrotechnical Commission). Skillnaderna mellan VRML97 och den VRML-del som används i Reachin API är bland annat att i Reachin API kan ytor ges olika strukturer och att det finns noder som gör det möjligt att känna det som syns. När VRML endast används för visuella 3D-objekt är information om hur ett objekt känns överflödig, men när VRML används för haptiska 3D-objekt är det intressant att veta om en yta är t.ex. skrovlig eller hal.

### 2.2.2 C++

C++ är ett objektorienterat programmeringsspråk som bygger på C. I Reachin API används en något modifierad version av ANSI C++ (ANSI är det amerikanska standardiseringsorganet American National Standards Institute.).

### 2.2.3 Python

Python[5] är ett objektorienterat programmeringsspråk och även ett scriptspråk och behöver således inte kompileras innan koden körs. När Python inte är integrerat i något annat språk körs det via en tolk där koden antingen kan skrivas in direkt eller läsas från filer.

---

[4] VRML97-specifikationen finns att läsa på http://www.web3d.org/Specifications/VRML97.

[5] Mer information om Python finns på www.python.org.

### 2.2.4 Systemkrav

För att kunna använda Reachin API finns vissa systemkrav specificerade enligt följande:

- Windows 2000/NT
- 500 MHz Intel Pentium II dubbla processorer
- 256 MB RAM
- Grafikaccelerator med stöd för Open GL

## 2.3 Vad fanns innan mitt examensarbete?

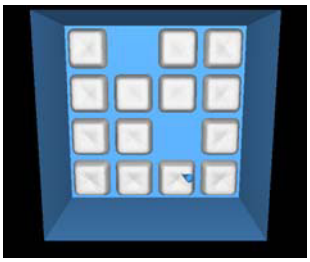


Bild 3 Memory

Innan jag började med mitt examensarbete ville jag veta om något liknande gjorts tidigare. Jag sökte på internet och hittade ett spel där Fantomen används för att spela det[6], det var dock gjort för seende personer. Jag hittade inga spel för blinda personer där Fantomen används förutom de som är gjorda på Certec. Det första Fantomenspel som gjordes på Certec var ett sänka-skeppspel. Därefter har bland annat ett memoryspel som spelas med Fantomen gjorts. I memoryspelet hör spelaren olika ljud när den vänder på korten istället för att se en bild som i traditionella memory.

Det finns även datorspel[7] för blinda som styrs på andra sätt än med Fantomen. De kan då styras med t.ex. ljud eller med ett tangentbord med punktdisplay för brailleskrift.

## 2.4 Synskadade personers vana vid datorer

Det finns flera hjälpmedel som underlättar för synskadade personer som vill använda datorer. Bland annat finns det talsyntes som kan läsa upp den text som visas på monitorn. De som inte vill använda talsyntes kan istället använda speciella tangentbord som har en display där punktskrift visas, när användaren ställer muspekaren över någon text visas den texten på tangentbordets display.

---

[6] Spelet finns på http://techhouse.brown.edu/~dmorris/haptic.battle.pong/.

[7] Datorspel för blinda och synskadade finns bland annat på http://www.tpb.se och http://gamesfortheblind.com.

Då synskadade personer kan använda datorer kan de även lära sig använda tangentbord. De flesta tangentbord har någon typ av markering på bokstäverna f och j samt på 5 på det numeriska tangentbordet, vilket gör det lättare att hitta vilka knappar man vill använda. Lättast är det dock att använda piltangenterna och det numeriska tangentbordet då dessa knappar sitter grupperade i mindre grupper bredvid övriga knappar.

# 3 Spelets utveckling

## 3.1 Problem och lösningar

Det första steget i mitt examensarbete var att fundera igenom vad som var målet med det och var jag skulle lägga tyngdpunkten. Här nedan beskriver jag de stora problemen jag hade i utvecklingsarbetet och tänkbara lösningar på dem, jag skriver också vilken lösning jag valt.

### 3.1.1 Kartomvandling

Från början ville jag utgå från en befintlig tvådimensionell karta och omvandla den till en haptisk karta som Fantomen skulle visa. Jag tänkte då att det skulle gå att läsa in vilken karta som helst och omvandla den så att den gick att känna i Fantomen och sedan skulle spelaren leta efter en skatt där. Om jag gjort så skulle den som spelar spelet kunna lära sig hitta på ett ställe samtidigt som den spelar ett spel och har roligt. Att möjliggöra en överföring från karta till en Fantomenbild är dock ganska tidskrävande och om jag skulle göra det skulle jag inte hinna göra något spel, jag har således utgått från en påhittad värld. Jag undersökte ändå om något liknande gjorts tidigare och hittade några orienteringsspel[8] där kartor automatiskt görs om till tredimensionella bilder. Jag hade kontakt med den ene speltillverkaren, men hans omvandlingsalgoritm var hemlig och inget han ville dela med sig av.

Detta var det största beslutet angående tyngdpunkten i examensarbetet. Jag valde alltså att lägga tyngdpunkten på att utveckla ett spel och inte på att göra om tvådimensionella kartor till tredimensionella världar.

### 3.1.2 Hur hitta skatten?

Nästa sak jag funderade över var hur spelaren ska få information om var skatten finns; ska den söka planlöst, få några ledtrådar eller få känna på någon karta över området? Jag funderade över hur bra de olika lösningarna var för spelaren, hur de egentligen skulle fungera och hur de skulle implementeras. Jag kom fram till några för- och nackdelar med varje sätt:

[8] Ett av spelen finns på http://www.melin.nu/winol/winol_direct.html.

#### *3.1.2.1 Planlös sökning*

Planlös sökning innebär att spelaren inte får någon information alls om var skatten finns utan får söka efter den överallt. Detta är förmodligen ett ganska dåligt sätt, det kommer bara att vara slumpen som avgör hur lång tid det tar att hitta skatten. Fördelen med planlös sökning är att det är väldigt lätt för programmeraren att implementera.

#### *3.1.2.2 Ledtrådar*

Ledtrådar måste ges genom ljud. Det finns flera sätt att göra detta:

- Spelet har ett ljud som låter hela tiden och varierar i styrka beroende på hur nära skatten spelaren är.
- Spelaren får ledtrådar där en röst till exempel säger: ”Skatten ligger under det största trädet”.

Båda dessa söksätt finns även i liknande spel för seende personer. Framförallt finns det flera spel som bygger på att spelaren får ledtrådar. Det finns bland annat textäventyr där spelaren får olika ledtrådar beroende på vad den försöker göra och var den är.

Nackdelen med ett ljud som låter hela tiden är att det kan bli ganska störande och tröttar ut spelaren efter ett tag. Fördelen är att det förmodligen skulle vara roligt i början att försöka lokalisera ljudet.

Fördelen med talade ledtrådar är att det förmodligen skulle vara roligt och det skulle bli lagom lätt att hitta skatten. Nackdelen med ledtrådar är att de inte är så dynamiska. Om skattens position är slumpmässig och olika varje gång spelet spelas måste det finnas många inspelade meddelanden som passar till alla ställen skatten kan finnas på.

#### *3.1.2.3 Karta*

Om en karta ska visas är frågan hur den ska se ut. Den kan antingen vara en nerskalning av världen eller mer likna en tvådimensionell orienteringskarta där olika saker representeras av olika tecken. En fråga när detta alternativ används är om spelaren ska kunna växla mellan karta och verklighet hur ofta den vill och om den ska få sin nuvarande position utritad på kartan, eller bara sin startposition. Att ha möjlighet att växla mellan en karta och en verklighet är en vanlig metod även i spel för seende personer.

Nackdelen med detta alternativ är att spelaren först får leta efter skatten på kartan och sedan leta efter den i verkligheten, den får alltså leta efter skatten två gånger.

Skatten är dock mycket lättare att hitta på kartan än i verkligheten.

#### *3.1.2.4 Mitt val*

Efter mycket funderingar kom jag fram till att den bästa lösningen förmodligen är att spelet innehåller en karta som spelaren kan se hur många gånger den vill. På kartan ska spelarens startposition och skattens position anges, men inte spelarens nuvarande position. Kartan är en nerskalning av den värld som spelet utspelas i, skatten är dock inte nerskalad och är därför lätt att hitta på kartan.

### 3.1.3 Områdesbegränsning

Det problem jag jobbat mest med är att Fantomens räckvidd är begränsad. Detta innebär att antingen kan bara ett litet område användas för spelet, eller så måste världen förflyttas på något sätt så att det går att känna hela världen med Fantomen. Detta har jag jobbat mycket med och gjort användartester på och behandlar därför det i ett eget kapitel, 3.3.

### 3.1.4 Programmeringsspråk

Vilket programmeringsspråk som används har inte någon betydelse för den slutliga användaren, det har dock betydelse för programmeraren och valet av programmeringsspråk har vållat mig en del problem.

Som jag skrivit tidigare har jag använt Reachin API för att programmera Fantomen. Från början tänkte jag skriva spelet i Python, jag tänkte att det skulle vara bra eftersom det inte behöver kompileras och jag tyckte att det verkade lite lättare än C++. Efter ett tag upptäckte jag att det inte alls var särkilt lätt och jag fick ingen kontroll över vad som hände, små ändringar i programmet kunde göra att inget alls fungerade längre utan att jag hittade vad som var fel. Det fanns inte heller någon i min omgivning som kunde Python, och därför fanns det ingen jag kunde får råd från när jag hade problem. Jag ändrade mig då och skrev i C++ istället, jag kunde lite av det innan och det fanns många andra som kunde det som jag kunde fråga om jag behövde hjälp.

## 3.2 Områdesbegränsning

Som tidigare nämnts var Fantomens områdesbegränsning det jag lade ner mest tid på och det förtjänar därför lite extra uppmärksamhet i rapporten.

För att det ska bli meningsfullt att spela ett skattjaktsspel med Fantomen behöver utrymmet som spelet spelas på vara större än Fantomens räckvidd. Om spelet bara använder området som finns inom Fantomens räckvidd kommer de saker som visas i spelet antingen att vara väldigt små, eller om sakerna ska vara stora kommer spelplanen att bli mycket liten. För att få tillräckligt stort spelutrymme måste världen flyttas på något sätt. Jag har funderat på olika sätt och under implemenationen upptäckte jag både saker jag tänkt göra som inte fungerade särskilt bra och saker jag inte tänkt på tidigare som fungerade riktigt bra. Jag utvecklade till att börja med följande sex olika sätt att flytta världen på:

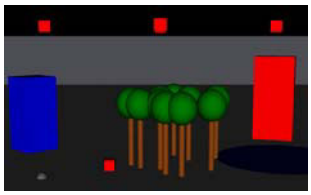


Bild 4 Förflyttningssätt 1, fyra knappar i taket.

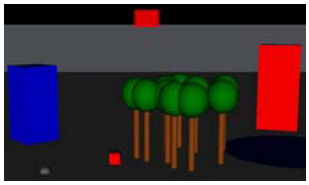


Bild 5 Förflyttningssätt 2 och 3, styrlåda i taket.

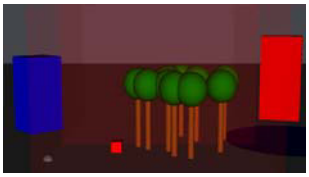


Bild 6 Förflyttningssätt 4, styrväggar.

1. Fyra knappar i taket, en för varje riktning. Knapparna sitter ungefär 1 dm från varandra. Knappen som finns längst till höger flyttar världen åt höger, knappen som finns längst bak flyttar världen bakåt, och analogt med knappen som finns till vänster och längst fram. Knapparna är på samma ställe hela tiden och följer inte med när övriga världen flyttas.
2. Styrlåda i taket. Lådan består av fyra väggar, höger, vänster, fram och bak. Om spelaren går in i lådan och trycker åt höger flyttar sig världen åt höger, övriga väggar i lådan fungerar på samma sätt. Styrlådan finns på samma ställe hela tiden och följer inte med när övriga världen flyttas. Kan även göras så att det finns styrlådor överallt i taket, då sitter de vägg i vägg med varandra. Många styrlådor gör att det är lätttare att hitta dem än vad som är. föregående, men styrlådan följer med i förflyttningen av världen.
3. Som
4. Spelområdet begränsas av fyra väggar. När spelaren trycker på en av väggarna flyttas världen. Väggarna flyttas inte när världen flyttas utan är på samma ställe hela tiden.
5. Världen flyttas med piltangenter och alltså inte med Fantomen. Om piltangnterna hålls inne en längre stund flyttas världen många steg på en gång

och spelaren behöver alltså inte trycka på piltangenterna en gång för varje förflyttning.

6 När knappen på Fantomen trycks in samtidigt som spelaren rör på Fantomens arm flyttas världen.

Det är endast i version 5 som världen flyttas många steg på en gång om spelaren håller in knappen länge. I version 6 flyttas världen så länge Fantomens knapp trycks in. I övriga versioner flyttas världen bara ett steg för varje tryck oavsett hur länge knappen hålls inne.

I version 1, 2 och 3 begränsas världen av ett tak, detta för att det ska vara lättare att hitta de knappar som finns i taket. Version 4, 5 och 6 begränsas ej av något tak eftersom de inte har några knappar högt upp.

Bild 7 Förflyttningssätt 5, piltangenter.

Bild 8 Förflyttningssätt 6, Fantomenknappen.

## 3.3 Användartest

Från början var det tänkt att de olika förflyttningssätten skulle testas på seende personer för att på så sätt upptäcka saker som inte alls fungerar. När de fem första versionerna var färdiga kom emellertid två blinda personer till Certec för att prova Fantomen och då gjordes även ett mindre användartest av skattjaktsspelet med de två. För att det förmodligen skulle bli jobbigt för försökspersonerna att testa för många versioner testade de endast version 4 och 5. Att just dessa två versioner testades berodde på att de var de lösningar som fick bäst resultat i inledande tester och att det var de som fungerade bäst. Förmodligen skulle version 6 också kunna vara bra, men då den inte fungerade vid användartestet kunde de inte testa den.

Det var en kvinna och en man som var mina testpersoner, båda två hade tidigare under dagen provat Fantomen för första gången. Kvinnan fick börja med att prova version 4 och sedan version 5, därefter provade mannen samma två versioner i omvänd ordning. Båda två var i rummet hela tiden, hur mycket de lyssnade medan den andre provade förflyttningen vet jag dock inte och kan därför inte avgöra hur mycket deras beteende, framförallt mannens eftersom han provade sist, påverkade varandra.

### 3.3.1 Förutsättningar vid användartest

De två versioner som användes vid användartestet bestod av en mindre värld som kunde flyttas på de olika sätt som beskrivits ovan. De innehöll således inte någon skatt, ingen karta över området, inga ljud eller andra finesser.

### 3.3.2 Kvinnans test

Kvinnan började med version 4, den där världen flyttas om spelaren trycker på väggarna. Hon undersökte metodiskt vad som fanns inom det område hon kunde känna och började därefter flytta världen genom att trycka på väggarna. Hon hade lätt att orientera sig, framförallt så länge hon var i närheten av den damm som fanns i världen och tog den som utgångspunkt. Hon flyttade sig dock inte särskilt mycket åt sidan och fick därför inte se så stor del av världen, detta kan eventuellt bero på att det fanns ett tomt utrymme mellan det som fanns mitt i världen och det som fanns en bit bort.

Efter att ha bekantat sig lite med världen fick kvinnan istället prova version 5, den som flyttas med tangentbordets piltangenter. Jag tyckte att det såg lättare ut för henne, hon kunde röra Fantomens arm över en betydligt större yta än när världen begränsades av väggar och hon fick därmed känna fler saker än tidigare. Hon sa dock själv att hon tyckte bättre om den första versionen hon testade, hon tyckte att det var lättare att orientera sig i den.

### 3.3.3 Mannens test

Mannen provade versionerna i omvänd ordning mot kvinnan och började alltså med version 5 där världen flyttas med piltangenterna. Mannen hade ganska bråttom och tog sig inte så mycket tid att känna var han verkligen var. Han flög mest omkring överallt och var ofta vilse. När han flyttade världen höll han gärna inne piltangenterna länge så att världen flyttades snabbt och långt på en gång. När han kom i utkanten av Fantomens räckvidd hade han svårt för att avgöra om det var därför det tog emot eller om det var för att det fanns ett föremål som han kände.

Därefter provade mannen version 4 där världen flyttas när spelaren trycker på väggarna. Då det inte går att flytta världen mer än ett steg åt gången gick det inte lika fort och han tog det lite lugnare och kände mer på sakerna. Mannen kände inte alltid när han tryckte på väggarna och var ofta inte medveten om att han råkat trycka på dem och därmed flyttat världen. Han hade lättare för att orientera sig i denna version än i den förra.

### 3.3.4 Sammanfattning av användartestet

Båda användarna tyckte att det var bra med den begränsning som väggarna i version 4 gav och tyckte att den var lättast att orientera sig i. Kvinnan tyckte att versionen med väggarna

var den klart bästa medan mannen var mer tveksam och hade svårt för att bestämma sig vad han tyckte bäst om. Fördelen som han såg med att flytta med väggarna var att han bara behövde använda en hand då. För att använda piltangenterna måste spelaren ha en hand på Fantomen och en på tangentbordet. Mannens förslag var att världen ska kunna flyttas både med piltangenterna och när spelaren trycker på väggarna. En annan variant är att ta en del från varje version så att världen flyttas med piltangenterna men att den begränsas av väggarna som då inte har någon annan funktion än att just avgränsa världen.

Det kan tyckas att två personer är för få för att göra ett användartest. Om resultatet ska vara riktigt bra behövs några till eftersom ännu fler problem hade visats då (Nielsen 1993). Men även om endast två personer deltar upptäcks många saker som behöver förbättras. Saker som inte alls uppmärksammats tidigare under utvecklingen kan bli väldigt tydliga. Om båda försökspersonerna har stora problem med samma sak är det stor sannolikhet att även andra personer kommer att ha problem med de sakerna och då bör de förbättras.

### 3.3.5 Andra iaktagelser under användartestet

Innan användartesterna utfördes hade jag och min handledare, Charlotte Magnusson, provat de olika versionerna och båda tyckte att version 4 och 5 (de som testades) flyttade sig åt motsatt håll mot vad vi förväntade oss när piltangenterna användes för att flytta världen. Detta sade inte försökspersonerna något om och när vi frågade dem om det sa de att det inte var något de funderat över och de tyckte att det sätt som användes var bra.

En annan iakttagelse var att blinda personer inte har samma uppfattning som seende personer om hur saker ser ut. T.ex. är det inte helt självklart hur ett träd ser ut ovanför den del av trädet som går att nå från marken. Dessutom använder seende personer ofta symboler som är förenklingar av saker, t.ex. är en stam med en klotformad krona en symbol för ett träd. För en synskadad person är detta inte lika tydligt som för en seende person då de inte tidigare stött på denna förenkling.

De träd som jag gjort i min testversion bestod av en ganska smal stam och längst upp en klotformad krona, varken kronan eller stammen hade någon speciell struktur utan var bara hala. Båda försökspersonerna kunde gå rakt genom den skogsdunge som fanns i världen utan att

upptäcka att de gick genom skogen och undrade då var skogen tagit vägen. Stammarna bör alltså vara tjockare så att spelaren åtminstone stöter i någon av dem när den går genom en skogsdunge.

### 3.3.6 Resultat av användartestet

Med utgångspunkt från användartestet har jag gjort vissa val av min design, både vad gäller förflyttning och andra saker som jag uppmärksammade under testerna.

#### *3.3.6.1 Förflyttning*

Jag tog fasta på det mannen i mitt användartest föreslog, att världen ska kunna förflyttas både med Fantomen och med tangentbordet, och kombinerade dessa. Den förflyttning som görs med Fantomen kommer att ske med metod 4 som är beskriven i kapitel 3.2, alltså att världen begränsas av fyra väggar och när spelaren trycker på någon av dem flyttas världen.

#### *3.3.6.2 Ljud*

Eftersom mannen flera gånger råkade flytta på världen utan att vara medveten om att han gjorde det lade jag till en signal som ljuder varje gång världen flyttas.

#### *3.3.6.3 Sakers utseende*

Då både mannen och kvinnan rörde sig mer i luften än på marken beslutade jag att de saker som finns i spelet ska kännas bra både nere vid marken och uppe i luften. Tidigare hade jag bara försökt få dem bra nere vid marken.

Då det inte var lätt för spelarna att veta vad de olika sakerna de kände egentligen var bestämde jag också att när programmet startats ska spelaren kunna välja att antingen börja spela direkt eller komma till en testvärld där den kan känna på de olika sakerna och få veta vad de är för något.

## 3.4 Hård- och mjukvarubegränsningar

Mitt utvecklingsarbete har ibland begränsats av att de hård- och mjukvaror jag haft tillgång till inte räckt till riktigt. Den dator jag huvudsakligen använt har inte tillräckligt stor kapacitet för att utföra allt som är möjligt med Fantomen, den hinner inte med om det blir för mycket för den att göra

på en gång. Det har gjort att jag inte kunnat använda Fantomens knapp och därför inte kunde göra version 6 av de förflyttningssätt som jag beskrivit i kapitel 3.2. Även version 3 av förflyttningssätten fungerade dåligt och det berodde förmodligen på att datorn inte hann med allt som krävdes av den för att den versionen skulle fungera. Som jämförelse kan nämnas att Reachin API kräver dubbla processorer medan den dator som huvudsakligen använts har en processor, den andra datorn har dock två processorer.

# 4 Det färdiga spelet

Resultatet av mitt examensarbete blev ett spel som är spelbart, dock med vissa problem. I detta kapitel redovisas hur spelet fungerar och vilka problem som finns i det.

Världen som spelet spelas i är inte så omfattande då det tog relativt mycket tid att göra världen och jag tyckte att det var viktigare att det andra fungerar än att världen ser fin ut. Världens utseende kan dessutom lätt ändras genom att byta den fil där världen finns specificerad.

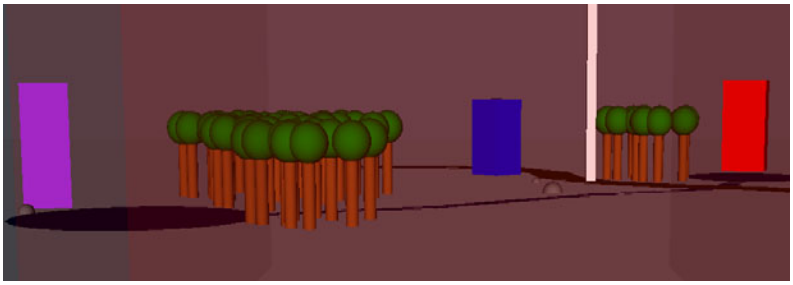


Bild 9 Vy från spelet.

Spelet styrs både med Fantomen och med tangentbordet. Vilka knappar på tangentbordet som används är valda efter hur lätta de är att hitta, de tangenter som används är främst piltangenterna, det numeriska tangentbordet, samt space och enter. Eftersom det är lätt att råka trycka på någon annan knapp än den spelaren tänkt sig ges vid flera knapptryck återkoppling då det berättas vilken knapp som tryckts ner.

## 4.1 Spelstart

När spelet startar får spelaren instruktioner och väljer därefter mellan olika saker som kan hända. De första instruktioner som läses upp är följande:

*”Hej och välkommen till skattjaktsspelet. För att få utförligare instruktioner om hur spelet spelas, tryck 1. För att få en provkarta över de objekt som finns i skattjaktsspelet, tryck 2. För att starta spelet, tryck 3. För att höra dessa instruktioner en gång till, tryck enter.”*

En översikt över de val spelaren har och vad som händer när spelaren väljer olika saker redovisas senare i detta kapitel.

### 4.1.1 Utförliga instruktioner

Om spelaren trycker på 1 och alltså väljer att få fler instruktioner om spelet läses följande instruktion upp:

*”Du har tryckt 1. Detta är ett skattjaktsspel, din uppgift är att hitta skatten som finns utplacerad på marken någonstans i världen. Du känner på världen med Fantomen. För att flytta världen så att du kan känna även de saker som finns utanför Fantomens räckvidd kan du antingen använda tangentbordets piltangenter eller Fantomen. Om du vill använda Fantomen för att flytta världen trycker du på väggarna i världen, när du trycker på en vägg hörs ett pling och samtidigt flyttas världen, väggarna flyttas dock inte.*

*Om du vill veta var du är och hur du ska hitta skatten kan du växla mellan en karta och världen, för att göra det trycker du på space och för att växla tillbaks trycker du på space en gång till. På kartan kan du känna allt som finns i världen i en förminskad form, skatten är dock förstorad. Hela kartan får plats inom Fantomens räckvidd. För att det ska vara lättare att orientera sig i spelet finns det mitt i spelet och mitt på kartan en hög och smal stolpe som kan användas som utgångspunkt.*

*Om du vill höra dessa instruktioner en gång till kan du trycka på 1 på tangentbordet. Vill du till provkartan trycker du 2 och om du vill börja spela trycker du 3. Om du vill höra de instruktioner du fick när spelet startade kan du trycka enter.*

*Lycka till i skattjaktsspelet!”*

### 4.1.2 Provkarta

Om spelaren väljer att se provkartan och trycker 2 läses följande instruktion upp:

*”Du har tryckt 2 och kommit till provkartan. Här kan du känna hur de olika sakerna som finns i spelet känns. Om du vill känna på nästa sak trycker du 0, vill du känna på föregående sak trycker du på komma. 0 och komma finns bredvid varandra längst ner på tangentbordets numeriska del. Det som visas nu är XXXX, det finns X saker att känna på efter denna och Y saker före.”*

Det ställe där det står XXXX säger rösten det som visas. Där det står X och Y talar rösten om antalet saker som finns att se före och efter den sak som visas för tillfället, detta för att underlätta för användaren.

### 4.1.3 Starta spelet

Om användaren vill börja spela spelet trycker den på 3 och får då höra följande:

*”Du har tryckt 3 och därmed startar spelet. Lycka till!”*

## 4.2 Funktioner i spelet

Spelet innehåller en del funktioner och finesser, vissa finns för att spelet ska gå att spela medan andra främst finns för att underlätta utvecklingsarbetet för mig och sedan har jag tyckt att de kunde vara roliga att ha kvar då de ändå inte stör spelets övriga funktioner.

Spelets viktigaste funktioner läses upp enligt kapitel 4.1.1. Spelet innehåller fler funktioner än de som läses upp, de är emellertid irrelevanta för spelaren och om de lästes upp skulle instruktionerna bli långa och spelaren skulle förmodligen glömma bort en stor del av dem och istället för att få hjälp av dem bli förvirrad. I denna rapport är det dock lämpligt att redovisa alla funktioner vilket görs i Tabell 1. Oavsett var i spelet spelaren är för tillfället händer samma sak om en viss knapp trycks ner.

| Knapp | Detta händer |
|---|---|
| Enter | Kortfattade instruktioner läses upp. |
| Space | Växla mellan karta och verklighet. |
| 1 | Detaljerade instruktioner om spelet läses upp. |
| 2 | Provkarta visas. |
| 3 | Spelet startar. |
| 0 | Visa nästa sak i provkartan. |
| , | Visa föregående sak i provkartan. |
| ← | Världen flyttas åt vänster. |
| → | Världen flyttas åt höger. |
| ↑ | Världen flyttas bakåt. |
| ↓ | Världen flyttas framåt. |
| . | Världen roteras medurs. |
| - | Världen roteras moturs. |
| Insert | Världen flyttas neråt. |
| Delete | Världen flyttas uppåt. |
| Home | Världen flyttas åt vänster med stora steg. |
| End | Världen flyttas åt höger med stora steg. |
| Page Up | Världen flyttas bakåt med stora steg. |
| Page Down | Världen flyttas framåt med stora steg. |

Tabell 1 Vad som händer vid olika knapptryck.

De knappar som används är valda efter hur de är placerade på tangentbordet och hur lätta de är att hitta. Det är företrädelsevis de stora knapparna och knapparna på det numeriska tangentbordet som används. T.ex. har 0 och

komma valts att användas för att visa nästa respektive föregående sak i provkartan. 0 och komma ligger bredvid varandra på det numeriska tangentbordet och är därför lätta att hitta.

De funktioner som är tänkta att spelaren ska använda är de översta, de som är ovanför det tjockare strecket, de undre funktionerna har jag främst implementerat för att underlätta utvecklingsarbetet för mig själv. Funktionerna i tabellen ovan fungerar oavsett var i spelet spelaren är, spelaren kan alltså alltid få instruktioner upplästa. Funktionerna är dock tänkta att användas som ett menysystem enligt Tabell 2.

Spelet startar Kortfattadeinstruktioner läses upp
- Repetera kortfattade instruktioner (Enter)
- Detaljerade instruktioner läses upp (1)
  - Detaljerade instruktioner repeteras (1)
- Provkarta visas (2)
  - Visa nästa sak (0)
  - Visa föregående sak(,)
- Spelet startar (3)
  - Flytta världen (←→↑↓)
  - Växla till karta (Space)
    - Växla till världen (Space)

Tabell 2 Menysystem

## 4.3 Problem i spelet

Spelet innehåller fortfarande vissa fel och problem. Vissa fel går att rätta till, andra fel beror troligen på Reachins programvara.

### 4.3.1 Ljud

Ljudet fungerar inte som det ska överallt. De ljud som ska spelas upp när användaren trycker på något på tangentbordet spelas inte upp, däremot spelas ljudet upp när användaren trycker på något med Fantomen. Detta går att rätta till.

### 4.3.2 Osynliga hål

Fantomen faller på en del ställen genom ytor och kommer på fel sida av dem, t.ex. under marken eller i ett hus. Detta problem är störst på de ställen där det finns saker på marken, t.ex. i övergången mellan en stig och vanlig mark, eller i övergången mellan en sjö och vanlig mark. Det är också

vanligt att fantomen ramlar genom förflyttningsväggarna om spelaren är längst nere vid golvet samtidigt som den trycker på väggen och världen flyttas. När Fantomen hamnat på fel sida en yta går det oftast inte att komma genom den igen till rätt sida, spelet får då startas om för att spelaren ska kunna fortsätta spela. Förmodligen är detta ett fel som beror på Reachin API.

### 4.3.3 Fantomen slår

När det som Fantomen visar ändras mycket på en gång kan det ibland hända att det på det ställe som Fantomen är för tillfället hamnar en sak. När detta händer kan Fantomen ibland slås undan med stor kraft och Fantomens arm slår då runt ganska mycket. Ibland när armen slås bort studsar den på en av förflyttningsväggarna och gör då att spelet flyttar sig lite till. Detta ställer till problem både för Fantomen och för användaren. För Fantomens del finns det viss risk att den skadas när den slår hårt åt något håll. För användaren är det största problemet att det är otäckt när Fantomen plötsligt slås ur användarens hand. Det är också problem med att Fantomen kan råka trycka på saker när den slås iväg och då händer det saker i spelet som spelaren inte har kontroll över. Detta problem bör gå att rätta till.

### 4.3.4 Fantomen hackar

Inte alltid, men ibland, hackar Fantomen när spelaren känner på saker. Vad detta beror på vet jag inte. Det kan vara något som är fel i programkoden, det kan bero på att det är för mycket för datorn att beräkna på en gång. Annars kan det bero på att Fantomen inte hinner med allt den ska göra. Problemet blir dock värre om spelet varit igång lång tid. Om det startas om hackar Fantomen oftast mindre en stund för att så småningom hacka mer igen. Detta ställer till vissa problem för användaren, när Fantomen hackar hela tiden blir det svårare att känna precisionssaker.

# 5 Slutligt användartest

För att se hur resultatet av skattjaktsspelet blivit utfördes ett slutligt användartest. I det användartestet deltog en person som testade den slutliga produkten. Då ljudet inte fungerar riktigt som det ska fick det simuleras. Försökspersonen fick höra exakt de meddelanden som datorn ska läsa upp, men det var jag som läste upp dem. Eftersom den enda skillnaden var varifrån ljudet hördes, från min mun istället för från högtalarna, påverkades inte försöket särskilt mycket av detta.

## 5.1 Försökspersonen

Försöket gjordes alltså på en person, en seende kvinna i 50-årsåldern utan vana vid Fantomen.

## 5.2 Försökets genomförande

Försökspersonen fick före genomförandet bekanta sig lite med Fantomen. För att lära sig hur Fantomen fungerar fick hon känna på några saker som inte har något med skattjaktsspelet att göra. Därefter fick hon kortfattade instruktioner om spelet och sedan startades det.

Kvinnan fick inte se monitorn under försöket och hon fick inte titta på tangentbordet när hon tryckte på någon knapp där, men hon hade ingen bindel för ögonen och behövde inte blunda, men gjorde det emellanåt ändå.

## 5.3 Försöket

Eftersom det är tänkt att en person som inte vet något om spelet ska kunna spela det fick kvinnan inte så många instruktioner innan spelet startades. Hon fick veta att det var ett skattjaktsspel och att hennes uppgift var att hitta skatten.

Spelet startade och startinstruktionerna lästes upp. Kvinnan valde direkt att trycka 1 och få mer information om spelet. När hon hört de utförliga instruktionerna ville hon höra dem en gång till, och därefter ytterligare en gång till. Hon tyckte att det var svårt att memorera alla instruktioner. Hon är van vid att kunna läsa igenom instruktionerna och läsa om det som är svårt, här går det inte att få bara en del av

instruktionerna, antingen spelas hela instruktionerna upp, eller inga alls.

Efter tre försök tyckte hon att hon förstått det mesta av det som sades och valde att trycka 2 och alltså se provkartan. Hon kände på sakerna, vissa saker tyckte hon var svåra att hitta och andra hittade hon nästan direkt. Hon tog det lugnt och kände noga på sakerna och försökte memorera dem.

När kvinnan känt på alla saker tryckte hon 3 och startade spelet. När hon skulle titta på kartan blev hon lite förvirrad, hon hade inte uppfattat att det finns en provkarta och en karta och trodde att det var provkartan hon fick känna på när det var kartan. När hon fått klart för sig hur det fungerar var det lättare för henne att känna de saker som fanns på kartan. Hon kände länge på kartan och försökte identifiera allt hon kände, det var inte så lätt eftersom hon inte hade någon riktig uppfattning om hur mycket sakerna skalats ner. Dessutom hackade Fantomen mycket, så de små sakerna var svåra att känna exakt hur de kändes.

Efter ett tag bestämde sig kvinnan för att spela spelet i den riktiga världen. Hon rörde sig inte så mycket och försökte inte flytta på världen. Efter ett tag förstod jag att hon inte riktigt förstått vad som hände när hon tryckte på väggarna och vad det var som flyttade sig. Hon förstod inte heller riktigt hur stor del av världen som fanns innanför väggarna och hur mycket som var utanför dem.

Flera gånger medan kvinnan gick runt i värden ramlade hon genom marken och sidväggarna, och några gånger fastnade hon inuti saker. Vissa gånger lyckades hon komma tillbaks till rätt sida om marken och väggarna, men flera gånger var vi tvungna att starta om spelet för att det skulle gå att fortsätta spela.

Efter lång tid och mycket letande lyckades kvinnan hitta skatten.

Efter att hon funnit skatten fick hon spela spelet en gång och titta samtidigt, det gick, som väntat, mycket lättare och snabbare då. När hon såg sakerna var hon förvånad över hur mycket sakerna förminskades när kartan visades och hur stora de var när hon var i den riktiga värden.

Kvinnan använde inte alls piltangenterna medan hon spelade och när jag frågade efteråt fick jag veta att det berodde på att hon glömt bort att det gick att använda dem.

## 5.4 Utvärdering av användartestet

Då användartestet endast är gjort på en person kan det inte ses som fullt tillförlitligt. Användartestet bekräftade dock några av mina tankar och jag tycker att det trots sin litenhet var värdefullt då jag fick flera nya infallsvinklar på det.

Problemet med att Fantomen hackar gör att användartestet blev något mindre rättvisande än vad det hade varit om Fantomen fungerat på rätt sätt. Även problemet med att Fantomen faller genom väggarna ibland störde försökspersonen. Jag sammanfattar här de viktigaste upptäckterna under användartestet.

### 5.4.1 Instruktioner

Instruktionerna är långa och svåra att komma ihåg, trots det saknas viss information och försökspersonen skulle velat veta mer innan hon spelade. Det är svårt att ge tillräckligt mycket information utan att användaren tappar koncentrationen på att lyssna på informationen. Det bästa sättet för en nybörjare att snabbt förstå hur spelet fungerar vore att någon som redan kan spelet förklarar och att den nye användaren kan ställa frågor om det den inte förstår. Ett annat sätt är att bifoga skrivna instruktioner, i både vanlig text och punktskrift, där läsaren själv kan välja vad den vill läsa många gånger. Problemet med skrivna instruktioner är att icke läskunniga personer inte kan tillgodogöra sig den informationen.

Kvinnan hade också lite synpunkter på den text som lästes upp, efter att ha funderat på det hon sade har jag modifierat texten något och det är den modifierade texten som finns redovisad i kapitel 4.1.

### 5.4.2 Termer

Kvinnan hade svårt för att veta vad alla termer betydde och vad som menades med de olika sakerna. Det hon hade svårast för var skillnaden mellan provkarta och karta, detta fick jag förklara för henne under spelets gång. Vilka termer som används borde tänkas igenom och gärna förklaras bättre. Att de bör förklaras bättre kommer in under problemet med instruktionerna som förklarats ovan.

### 5.4.3 Stor värld

Världen var väldigt stor för en nybörjare. Första gången en person spelar spelet räcker förmodligen en värld som inte är större än Fantomens räckvidd. Om spelaren inte är van vid varken Fantomen eller spelet tar det ganska lång tid för den att orientera sig i den lilla värld som finns innanför förflyttningsväggarna. Om spelet hade flera banor borde den första vara liten och sedan kan de växa allt eftersom spelaren hittar skatterna.

### 5.4.4 Förflyttning av världen

Försökspersonen hade svårt att förstå hur världen förflyttades, hur stor del av världen som finns innanför förflyttningsväggarna och hur långt de flyttas åt gången. Efter ett tag förstod hon bättre hur det gick till, och det blir förmodligen bättre när användaren blir mer van. Detta problem finns till viss del inbakat i problemet med instruktioner då det skulle behöva förklaras bättre innan spelaren börjar spela.

### 5.4.5 Lång tillvänjningstid

Spelet har ganska lång tillvänjningstid. Det är svårt att börja spela spelet direkt, spelaren behöver få många instruktioner först och sedan tar det lång tid innan spelaren fått en bra uppfattning om vad som händer i spelet.

## 5.5 Brister i användartestet

Det finns några brister i användartestet vilka redovisas i detta kapitel.

Användartestet bestod endast av en försöksperson. Om fler försökspersoner deltagit skulle försöket blivit mer tillförlitligt och fler brister uppenbarats. Jag tycker dock att försöket kombinerat med mina egna kunskaper och tankar var givande och gav mig flera nya tankar.

Att Fantomen hackade gjorde användartestet sämre, det påverkade dock inte saker som uppläsning av instruktioner.

Användartestet utfördes bara vid ett tillfälle, det hade varit bra om försökspersonen haft möjlighet att komma tillbaks några gånger med en veckas mellanrum och prova spelet. Detta skulle göra att användaren vande sig vid spelet och kunde då gett fler synpunkter.

Då det inte finns någon som är van vid att använda detta skattjaktsspel fanns det inte möjlighet att göra testet på en

van användare. Detta hade annars varit ett önskemål då det i så fall hade varit möjligt att se hur stora och svåra banorna kan bli innan de blir omöjliga att klara.

# 6 Skattjaktsspelets framtid

Det finns mycket som fortfarande behöver göras innan skattjaktsspelet är riktigt bra. Jag har funderat över vad som kan göras för att få spelet bättre och roligare. Jag har också funderat över andra lösningar av spelet som skulle kunna göras och idéer till andra spel som skulle kunna spelas med Fantomen.

## 6.1 Vidareutveckling av spelet

### 6.1.1 Ljud

Spelet kan bli betydligt bättre och roligare med mer ljudeffekter och på de ställen där det redan finns ljud kan de vara lite roligare.

### 6.1.2 Banor

Spelet skulle bli roligare om det hade fler banor och inte bara en. Banorna skulle kunna bli svårare och svårare.

### 6.1.3 Skatter

Spelet kan innehålla flera skatter per bana och blir då mer som orientering där spelaren tar olika kontroller i en förutbestämd ordning.

### 6.1.4 Verkliga platser

För att få spelet riktigt bra borde möjligheten att läsa in verkliga kartor finnas. Efter att en karta lästs in omvandlas den sedan automatiskt till en tredimensionell värld som kan kännas med Fantomen.

### 6.1.5 Ledtrådar

Istället för att visa en karta över det område som skatten finns i kan muntliga ledtrådar ges om var skatten finns. Detta kräver dock ganska mycket arbete och med muntliga ledtrådar blir det svårare att slumpmässigt välja skattens placering.

### 6.1.6 Spara

Om det finns flera banor i spelet kan det vara intressant för en spelare att kunna få ett lösenord när den vill sluta spela och då inte behöva spela om de första banorna vid nästa speltillfälle.

### 6.1.7 Sakers utseende

De saker som finns i de tredimensionella världarna, t.ex. träd, kan göras bättre. Sakerna kan få ett bättre utseende och bättre känsla i ytorna.

### 6.1.8 Tid

Om spelet går på tid kan spelaren försöka slå sitt eget rekord och spela snabbare för varje gång. Men eftersom skatten är på olika ställen varje gång kommer de olika spelomgångarna inte att kunna jämföras på något rättvist sätt.

### 6.1.9 Poängberäkning

Spelet skulle kunna utvidgas med poängberäkning och tio-i-topp-lista. Poängen kan då beräknas på flera olika sätt, den kan bland annat bero av hur snabbt spelaren hittar skatten och hur ofta spelaren behöver växla mellan verklighet och karta

## 6.2 Andra spelidéer

Medan jag har utvecklat skattjaktsspelet har jag funderat på andra idéer till spel som blinda personer skulle kunna spela med Fantomen.

### 6.2.1 Schack

Ett schackspel för Fantomen skulle kunna fungera ungefär som ett ”vanligt” datorschackspel, skillnaden är att spelaren kan känna pjäserna. Fördelen med ett Fantomenschackspel jämfört med ett manuellt schackspel är att pjäserna inte riskerar att råka flytta sig och därmed byta plats när spelaren känner på dem, dessutom skulle det vara möjligt att känna schackbrädets rutmönster med Fantomen.

Förutom att spela schack mot datorn med Fantomen skulle det också gå bra att spela mot en annan person. Nackdelen är att om spelarna bara har tillgång till en Fantomen skulle det innebära att det inte gick att känna på pjäserna medan den andre spelaren funderar

Även andra brädspel skulle gå bra att spela med Fantomen.

### 6.2.2 Äventyrsspel

Förmodligen skulle det gå att göra ett roligt äventyrsspel i stil med Super Mario Bros även för Fantomen. I Super Mario Bros springer spelaren genom en tvådimensionell värld, spelaren är hela tiden i mitten och världen förflyttas olika fort beroende på om spelaren går eller springer. Under spelet finns det vissa hinder som spelaren ska ta sig förbi. Det skulle gå att göra ett liknande spel som spelas med Fantomen. Beroende på vilka faror som är på väg kan då olika ljud spelas upp, eller maken kännas på olika sätt.

# 7 Spelets tekniska konstruktion

Skattjaktsspelet är utvecklat med Reachin API och främst skrivet i C++ och VRML. I detta kapitel görs en beskrivning av hur spelet är uppbyggt programmeringsmässigt. En trädstruktur över programmet finns sist i bilaga 1.

## 7.1 Världens utseende

Reachin API ger programmeraren en möjlighet att kombinera VRML[9] och C++[10]. Detta har jag använt för att bestämma hur olika saker i världen ser ut. Jag har då specificerat ett grundutseende i VRML, därefter har jag gjort ett gränssnitt till det i C++. Detta gör att det i en annan VRML-fil går att utnyttja det gränssnittet och lägga till flera objekt av samma sort, men med modifierat utseende, utan att behöva göra om hela objektet.

### 7.1.1 World.wrl

Hur världen där skattjaktsspelet äger rum ser ut finns i VRML-filen World.wrl. En World.wrl-fil kan för en liten värld som innehåller mark, ett hus och en skog se ut enligt följande:

```
DEF world Group {
      children [
            Ground {
                  size 2 0.001 2
                  translation 0 -0.0005 0
            }
            House {
                  size 0.05 0.1 0.04
                  translation 0.2 0.05 -0.17
                  color 1 0 0
                  rotation 0 1 0 -0.3
            }
            Wood {
                  translation 0.02 0 -0.02
            }
      ]
}
```

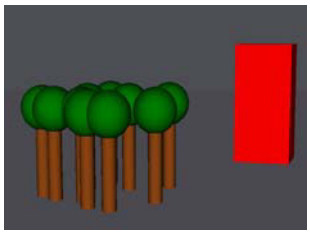


Bild 11 Den värld som beskrivs i koden till vänster.

[9] VRML-filers ändelse är .wrl.

[10] C++-filers ändelse är .cpp och .h.

Denna värld består alltså av mark, ett hus och en skog. Alla dessa har ett C++-interface i vilken en VRML-fil som anger utseendet på sakerna läses in.

### 7.1.2 Treasure.wrl

I Treasure.wrl finns skattens utseende beskrivet. När spelaren hittar skatten och trycker på den spelas ett ljud upp, vilken ljudfil som ska spelas upp anges i denna fil.

### 7.1.3 Button.wrl

Button.wrl innehåller utseendet på de väggar som spelaren trycker på för att flytta världen, när spelaren trycker på väggarna och världen flyttas spelas ett ljud upp. Precis som för Treasure.wrl anges den ljudfil som ska spelas upp här.

## 7.2 Världens funktionalitet

### 7.2.1 testWorld.cpp

Huvudprogrammet heter testWorld.cpp, det som sker i det är att World.cpp anropas.

### 7.2.2 World.cpp

World.cpp är den fil där allt händer. Först byggs världen upp genom att World.wrl, som beskrivits ovan, läses in, därefter läses även Treasure.wrl och Button.wrl in. Var skatten ska placeras bestäms slumpmässigt strax efter att dess VRML-fil lästs in. Därefter beskrivs allt som ska hända när användaren gör något.

Att världens, skattens och knapparnas utseende finns i en egen fil och läses in i World.cpp gör det lätt att byta deras utseende genom att bara byta vilken fil som läses in, eller modifiera de befintliga filerna.

# 8 Tankar runt det färdiga examensarbetet

När examensarbetet nu är färdigt vill jag sammanfatta det och fundera över vad jag lärt mig och vad som kunde ha varit bättre och varför det inte blev så.

Det är svårt att direkt veta allt jag lärt mig, vissa saker är uppenbara, men andra kommer jag förmodligen inte att upptäcka förrän om ett tag. Det som varit min första lärdom är att det inte alltid gått att göra saker så snabbt som jag önskat. Det beror till viss del på sådant som man inte själv kan styra över, t.ex. datorhaveri. Det kan ta många timmar att fixa med hård- eller mjukvara som av någon anledning inte fungerar som det ska.

En annan anledning till att saker inte alltid gått så fort som jag tänkt är att jag nog ofta varit optimistisk i mina tidsberäkningar. Det tar ett tag att sätta sig in i nya problem. Dessutom har jag behövt fundera igenom mina lösningar ganska mycket, när jag funderat över hur lång tid olika saker skulle ta att göra trodde jag att implementationen av programmet skulle ta mest tid, men funderingar om hur programmet ska fungera har tagit mer tid än implementationen.

Efter att ha jobbat ett tag med mitt examensarbete kom jag fram till att det varit mycket lättare om mina förkunskaper varit bättre. Det jag har haft mest problem med är rena implementationssaker som hade varit betydligt mycket lättare att lösa om jag varit bättre på C++ innan jag började jobba. Jag har dock lärt mig en hel del C++ och är nu betydligt bättre på det än jag var tidigare.

Den kanske viktigaste lärdomen jag dragit är att det inte är så lätt att sätta sig in i en annan situation än den man är i. Jag har vid flera tillfällen börjat tänka på någon lösning, men efter en stund har jag kommit på att det går inte, spelet ska rikta sig till blinda personer och det jag tänkte på kräver viss syn. Det gäller att försöka sätta sig själv i den situation som de man vänder sig till är i. I mitt fall har det bland annat inneburit att jag fått blunda när jag testat de saker jag implementerat.

En annan viktig sak, som jag egentligen redan visste men nu fått bekräftat, är att det är viktigt att testa sin lösning på andra. I mitt första test deltog endast två personer, men det gav mig ändå mer än jag trodda att det skulle göra. Jag hade också föreställningen att det behövdes många försökspersoner för att testet skulle vara givande, men ett försök med bara två personer ger mycket, för man får två andra personers syn på det man gör och det kan vara tillräckligt för att kunna ta tillvara på alla synpunkter. Om jag hade haft mer tid på mig skulle jag ha gjort fler tester, men det var inte möjligt med den begränsade tid ett examensarbete sträcker sig över.

Nu när rapporten är i stort sett färdigskriven har jag lärt mig ytterligare en sak. Jag har kommit på många nya idéer genom att skriva om det jag gjort. Jag har bland annat kommit på flera saker som kunde ha lösts på ett bättre sätt och saker som gjort att spelet blivit roligare och bättre att spela. I detta skede finns det dock inte längre något tidsutrymme för att förbättra de sakerna.

Som avslutning vill jag säga att detta varit ett roligt och lärorikt examensarbete och jag är glad att jag fick möjlighet att göra det.

# Referenser

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

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

Nielsen, J (1993). *Usability engineering.*

Löwgren, J (1993). *Human-computer interacion.*

Norman, D. A. (1993). *Things that make us smart.*

Norman, D: A. (1998). *Design of everyday things.*

Sjöström, C (1996). Fantomaten – The PHANToM – för handikappade barn. www.certec.lth.se/dok/fantomaten1/

Sjöström, C (1997). Fantomaten. www.certec.lth.se/dok/fantomaten2/

Reachin: www.reachin.se

Python: www.python.org

Sensable: www.sensable.com

VRML-specifikation: http://www.web3d.org/Specifications/VRML97

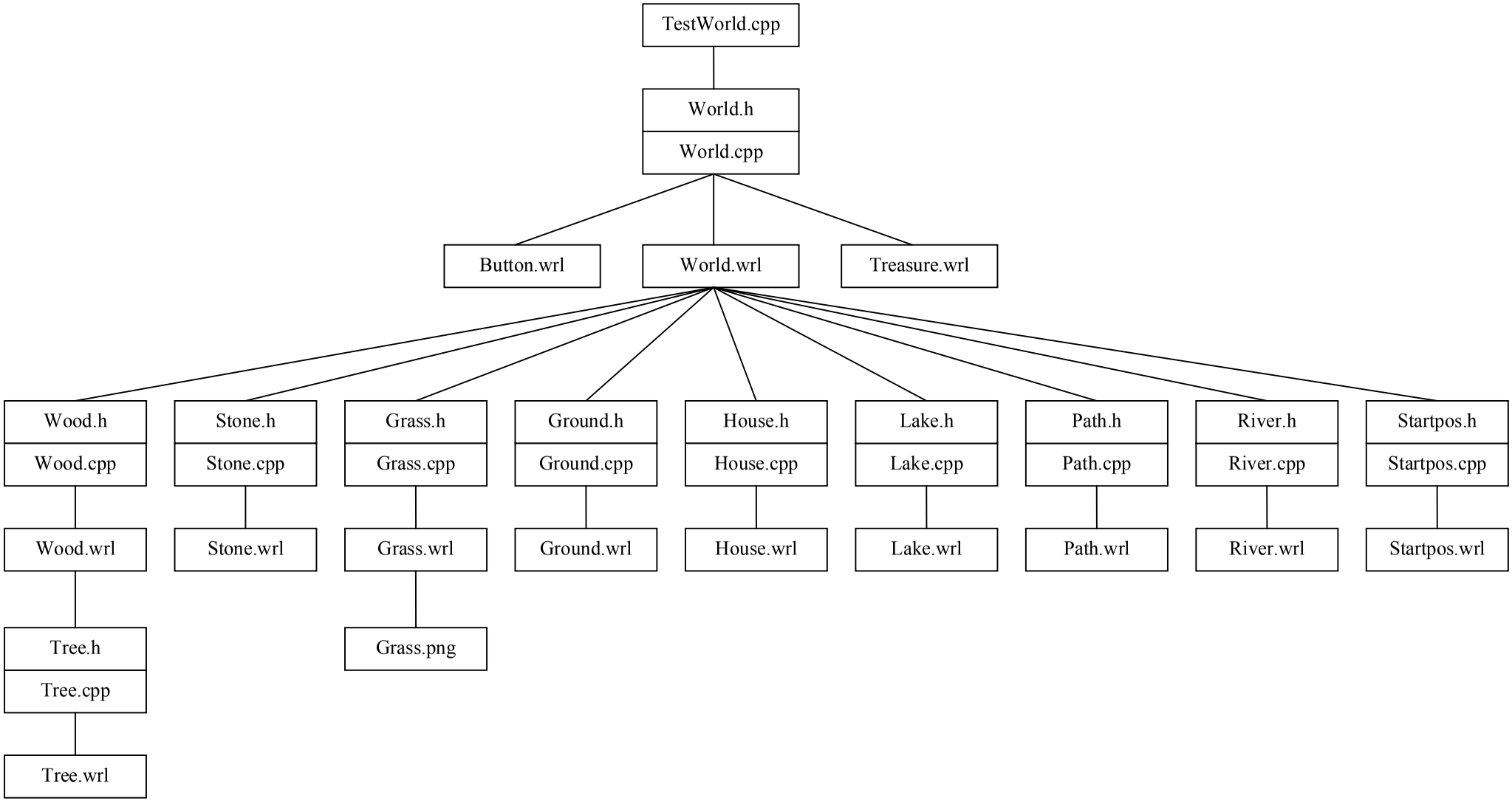
Fantomenspel för seende: http://techhouse.brown.edu/~dmorris/haptic.battle.pong/

Orienteringsspel: http://www.melin.nu/winol/winol_direct.html

Datorspel för blinda och synskadade: http://www.tpb.se

Datorspel för blinda och synskadade: http://gamesfortheblind.com

# Bilaga 1, Programmeringsträd


TestWorld.cpp
World.h
World.cpp
Button.wrl
World.wrl
Treasure.wrl
Wood.h
Wood.cpp
Stone.h
Stone.cpp
Grass.h
Grass.cpp
Ground.h
Ground.cpp
House.h
House.cpp
Lake.h
Lake.cpp
Path.h
Path.cpp
River.h
River.cpp
Startpos.h
Startpos.cpp
Wood.wrl
Stone.wrl
Grass.wrl
Ground.wrl
House.wrl
Lake.wrl
Path.wrl
River.wrl
Startpos.wrl
Tree.h
Tree.cpp
Grass.png
Tree.wrl

# Certecs rapporter

Ett urval av rapporter från CERTEC

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/

Anderberg, M
*Rörelsehinder och Toalettdesign*
Examensarbete, Certecrapport 1:2002
http://www.certec.lth.se/dok/rorelsehinderoch/

Gulin, A
*Certec-Brandtalaren*
Examensarbete, Certecrapport 3:2001
http://www.certec.lth.se/dok/certecbrandtalaren/

Det finns många datorspel för seende personer, men för blinda och synskadade personer finns det bara väldigt få. Detta beror förmodligen främst på att traditionella datorspel bygger på det som visas på datorskärmen, och för den som inte ser är det då omöjligt att spela spelet.
Ett annat problem som blinda personer ofta utsätts för är att det är svårt för dem att hitta i okända miljöer och veta hur de ska bete sig där. En seende person kan lätt titta på en karta för att hitta till ett ställe, motsvarande för en blind person blir då att den behöver känna denna karta. Det finns gott om tvådimensionella kartor som passar för seende, men det finns inte lika många tredimensionella kartor.
Målet med examensarbetet var att kombinera bristen på datorspel för blinda personer och bristen på tredimensionella kartor. Detta har resulterat i ett skattjaktsspel där spelaren ska försöka hitta en skatt i en ”verklighet”. Till sin hjälp har spelaren en tredimensionell karta. Till skillnad från vanliga datorspel där spelaren ser vad som händer känner spelaren här den värld som spelet utspelas i. För att känna spelet används ”the PHANToM”, eller Fantomen, vilket är ett haptiskt känselgränssnitt.

Den här rapporten hittar du också på internet:
http://www.certec.lth.se/dok/haptiskskattjakt/

Certec
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 20 anställda och har en årsomsättning på omkring 10 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 2:2003
ISRN CERTEC-ER-03/2-SE

MARS 2003

Anna Lindholm

# Haptisk skattjakt