Användning av motion sensor-teknik i mobila enheter

Daniel Grahn & Henrik Persson

Master's Thesis

Department of Design Sciences
Lund University
ISRN:LUTMDN/TMAT-5105-SE

EAT 2007

1 Sammanfattning

Dagens mobiltelefoner styrs till största del med hjälp av knappar. Vi vill med vårt examensarbete testa och bygga upp en kunskapsbank angående ett alternativt sätt att styra mobiltelefoner, nämligen med motion sensor-teknik. Mobiltelefonernas grafiska användargränssnitt liknar varandra väldigt mycket och är också väl utarbetade ur ett rent användarvänligt perspektiv. Utifrån dessa traditionella användargränssnitt har vi försökt utveckla gränssnitt styrda med motion sensor-teknik och vi har lagt fokus på hur användarvänligheten påverkas.

Vi började med att använda oss av ett traditionellt grafiskt användargränssnitt som var uppbyggt med en grid-meny och navigerade i menyn genom att skaka på telefonen. Resultatet blev dock inte tillfredställande utan vi gick vidare och tog fram en ny design på vår meny. Istället för en grid-meny placerade vi ikonerna runt om på skärmen och navigerade genom att luta telefonen. För att välja i detta gränssnittet användes en timer som gick igång när något element blev fokuserat. Om elementet var fokuserat tillräckligt länge tolkades detta som ett val. Utifrån den designen tog vi fram ytterligare ett gränssnitt där timern ersattes av en knapp. Knappen förenklade navigationen eftersom användaren fick större kontroll över vad som ska väljas.

När gränssnitten var utvecklade genomförde vi användartester på de två senaste gränssnitten, det första valdes bort då vi ansåg att det var för dåligt och inte så innovativt som vi hade hoppats. Dessa användartester hade vi sedan som grund för att diskutera och dra slutsatser kring vad som blir bättre eller sämre när motion sensor-tekniken används istället för de traditionella navigationssätten.

Vi har gjort vårt examensarbete på TAT AB i Malmö och har använt de produkter och utvecklingsverktyg TAT utvecklat för att skapa de användargränssnitt vi sedan testat. Mobiltelefonen vi använt är en Nokia 5500 sport och är en av få mobiltelefoner som har en inbyggd accelerometer. Telefonen använder tredje upplagan av Nokias Series-60 användargränssnitt samt version 9.1 av operativsystemet Symbian. Med hjälp av TATs Cascades har vi gjort användargränssnitt som sedan lades in som små program på telefonen och på så sätt simulerade vi ett riktigt användargränssnitt på en telefon.

2 Abstract

The cell phones on the market today are usually controlled by the key pad. The goal in our thesis is to test and build a knowledge bank regarding an alternative way for navigation, motion sensor technique. The GUI in modern cell phones is quite similar and well formed from a usability point of view. Derived from these user interfaces we tried to design and develop user interfaces navigated with motion sensors, and trying to get ours as user friendly as possible. We wanten to see if navigation with motion sensor-technique was possible and how it affected the user friendliness.

We started with an existing user interface designed in a common way with a grid menu, and navigated through the menu by shaking the device. The outcome from this design wasn't satisfying, so an alternate design was developed. In the new design the grid menu was removed and the icons were placed along the edge of the screen and the navigation was instead done by tilting the device. A timer was used for choosing in the interface. The timer was trigged when an element was focused and after being focused for some time this was interpreted as choosing in the interface. On the basis of this new design a third interface emerged and in this interface the timer-selection was replaced by a button. The button in the third interface made it easier to navigate to the desired destination since the user had more control of the actual navigation.

Usability testing was done on the later two of the three interfaces. We considered the first interface to so bad that it was removed from the test, and it wasn't so innovative that we had hoped for either. These usability tests worked as a foundation for our discussions and conclusions on whether the interfaces with motion sensor were better than the existing interfaces.

We have done our thesis at TAT AB in Malmö and therefore had TAT's products and utilities at our disposal. The cell phone we used was a Nokia 5500 sport and is one of the few cell phones with an integrated accelerometer. The phone uses the third edition of Series-60 and Symbian version 9.1. We built our interfaces with TAT Cascades and then installed them into the phone to simulate a real user interface.

3 Förord

Vi som genomfört detta examensarbete har utbildat oss till civilingenjörer inom information- och kommunikationsteknik på Lunds Tekniska Högskola. Vi har båda fördjupat oss inom interaktion och männsika-tekniksystem.

Idén för vårt examensarbete, att utveckla rörelsestyrda användargränssnitt, uppenbarade sig under vår sista studieperiod på LTH. På senare tid har intresset för motion sensor-teknik ökat drastiskt, främst inom spel och mobiltelefonmarknaden. Succén med Nintendos senaste spelkonsol Wii, vilken styrs med motion sensor-teknik har banat väg för ett nytt interaktionssätt.

Vi vill med vårt examensarbete forska kring motion sensor-tekniken som en interaktionsteknik och försöka skapa en kunskapsbas kring hur tekniken används på bästa sätt. Utvecklingen kommer främst inriktas på gränssnitt till mobiltelefoner.

Vi genomförde vårt examensarbete på TAT AB. TAT AB är ett framgångsrikt företag inom mobiltelefonibranschen vars affärsidé är att utveckla nyskapande användargränssnitt som förbättrar användarupplevelsen. Vi har under projektets gång fått värdefull hjälp av TAT i form av att få utnyttja deras produkter och deras kunskap för att lyckas nå våra mål med detta examensarbete.

Innehåll

1 Sammanfattning **2**

2 Abstract **3**

3 Förord **4**

4 Inledning **7**
4.1 Traditionellt mobilt gränssnitt . . . 7
4.1.1 Menysystem . . . 7
4.1.2 Input från användaren . . . 7
4.1.3 Fördelar och nackdelar . . . 7
4.2 Motion Sensor-teknik . . . 8
4.2.1 Accelerometer . . . 8
4.2.2 Gyro . . . 9
4.2.3 Fördelar och nackdelar . . . 9
4.3 Tidigare användning av motion sensor-teknik . . . 10

5 Problemställning **12**

6 Målsättning **13**
6.1 Skapa en kunskapsbas . . . 13
6.2 Navigation endast med motion sensorn . . . 13
6.3 Navigation med motion sensorn och en knapp . . . 13

7 Metod **14**
7.1 Utveckling . . . 14
7.2 Testning . . . 14
7.3 Uppföljning . . . 15

8 Utveckling **16**
8.1 Utveckling på TAT AB . . . 16
8.1.1 Företaget . . . 16
8.1.2 Produkter . . . 16
8.2 Utvecklingsverktyg . . . 17
8.2.1 Kastor . . . 17
8.2.2 Cascades . . . 17
8.2.3 MotionLab . . . 18
8.2.4 Symbian OS . . . 18
8.3 Bosch accelerometer på kretskort . . . 18
8.4 Våra motion sensor-gränssnitt . . . 20
8.4.1 GMUI1 . . . 21
8.4.2 GMUI2 . . . 22
8.4.3 GMUI3 . . . 23
8.5 Funktioner i telefonen . . . 24
8.5.1 Sifferinmatning . . . 24
8.5.2 Textinmatning . . . 25
8.6 Problem under utvecklingen . . . 26

9 Användartest **28**

9.1 Målgrupp 28
9.2 Testmetodik 28
9.3 Testfall 28
9.4 Testmiljö/Testutrustning 30
9.5 Intressant data 30
9.6 Sammanställning av avslutande enkät 32
9.7 Resultat av användartest 33
9.7.1 Diagram och tabeller 33
9.7.2 Kommentarer 34

10 Slutsats **36**

11 Diskussion **37**
11.1 Andra tillämpningar av motion sensor-teknik 37
11.1.1 Svara/Lägga på 37
11.1.2 Övervaka användarens rörelsemönster 37
11.1.3 Rörelseigenkänning 37
11.2 Framtida arbete 39
11.2.1 Vidareutveckling av motion sensor-gränssnitt 39
11.2.2 Kalibrering 39
11.2.3 Integrering av accelerometrar 39
11.3 Koncept och idéer 40
11.3.1 Musikspelare 40
11.3.2 Bildspår 41
11.3.3 Fixerade fält 42
11.3.4 Zoom-meny 43
11.3.5 Bildgalleri 1 44
11.3.6 Bildgalleri 2 44
11.3.7 3D galleri 45

A Ordlista **I**

B Avslutande enkät **II**

4 Inledning

4.1 Traditionellt mobilt gränssnitt

De gränssnitt som existerar i dagens mobiltelefoner styrs till största del av knapptryckningar. Det finns en handfull telefoner som styrs med hjälp av en pekskärm, men även de telefoner som styrs med pekskärm har oftast en knapp.

4.1.1 Menysystem

De menysystem som finns i dagens gränssnitt är oftast upplagda som ett rutnät med de olika ikonerna på skärmen. Det är oftast huvudmenyn som är upplagd på detta sätt, övriga menyer kan ha ett annat utseende fast vanligast är ändå en lista med olika alternativ som man kan scrolla igenom. Ibland är dessa listor uppdelade i tabbar för att förenkla navigationen och för att göra det mer översiktligt. Det förekommer även listor i listnoderna, vilket ofta används bland kontakterna då varje kontakt har flera olika telefonnummer.

4.1.2 Input från användaren

Som tidigare nämnts så styrs de flesta gränssnitten med hjälp av knappar eller med en pekskärm. Navigation i ett traditionellt gränssnitt brukar styras med hjälp av styrknappar, det vill säga upp, ner, vänster, höger och en centrumknapp som används för att välja i gränssnittet. Ibland kan styrknapparna vara ersatta med en styrspak. Centrumknappen är då ersatt med att man trycker ner styrspaken som en knapp.

Om fallet är så att en pekskärm används finns inga direkta navigationsnormer. Användaren pekar på det han/hon vill titta närmre på. Denna navigationsteknik är i många avseenden överlägsen eftersom gränssnittet enkelt kan anpassas för olika applikationer eller funktioner.

4.1.3 Fördelar och nackdelar

Fördelen med fysiska knappar är att det är ett väldefinierat sätt att styra ett gränssnitt på, eftersom tidigare mobiltelefonanvändare har erfarenhet av detta navigationssätt. Det är också lätt att sätta en etikett på en knapp som talar om exakt vad knappen gör. Nackdelen med knappar är att dagens mobiltelefoner gärna ska vara så små och behändiga som möjligt, vilket medför att knappsatsen måste ta så liten plats som möjligt då man fortfarande vill ha en så stor skärm som möjligt. Detta leder till att man gör knapparna mindre vilket leder till att de blir svåra att trycka på. Denna nackdel försvinner då man använder en pekskärm som styrdon eftersom knapparna då kan tas bort.

4.2 Motion Sensor-teknik

Motion sensor-tekniken går ut på att accelerationen i två eller tre dimensioner registreras och utifrån denna rörelsedata kan man utföra applikationsspecifika beräkningar. I vårt fall hade vi en accelerometer som mätte accelerationen i tre dimensioner. Med denna accelerometer kunde vi registrera alla rörelser, det vill säga alla krafter som påverkade telefonen.

4.2.1 Accelerometer

En accelerometer är en mätgivare som registrerar ett objekts acceleration i olika riktningar. Detta illustreras i figur 1 på sidan 8. Accelerometern ger upphov till en elektrisk signal som är proportionell mot den påverkande accelerationskraften.

Figur 1: Tre frihetsgrader för en accelerometer

En accelerometers grundläggande beståndsdelar är ett hölje som är fast förankrat i objektet vars acceleration ska mätas. Inuti höljet finns en tyngd som inte är fast förankrat. Tyngdens tröghet ger upphov till dragkrafter vilka omvandlas till elektrika signaler. De två vanligast förekommande typer av accelerometrar reagerar på:

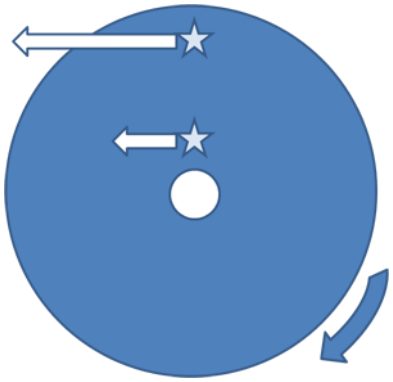
- Lägesförskjutning mellan magneter och spolar.
- Tröghet av en tyngd upphängd mellan piezoelektriska kristaller[1].

Ett vanligt förekommande problem vid användning av en accelerometer är då objektets exakta position i rummet ska bestämmas. Teoretiskt sett kan positionen beräknas genom att integrera objektets acceleration över tiden, men i praktiken fungerar inte detta eftersom felaktig och brusig mätdata kommer resultera i stora felberäkningar av positionen. I ett sådant fall måste objektets position beräknas med en annan teknik för att uppnå pålitliga beräkningar av objektets position. Accelerometern kan alltså i praktiken endast användas för att beräkna en momentan acceleration av objektet.

[1] Piezoelektricitet är en egenskap som vissa kristaller har. En piezoelektrisk kristall har egenskapen att om den deformeras omvandlas det mekaninska arbetet till elektricitet eller tvärtom. Denna egenskap lämpar sig bra för användning i en accelerometer.

4.2.2 Gyro

Ett gyro mäter vinkelhastigheten av ett objekt. Detta görs genom att utnyttja coriolisaccelerationen, vilken är en effekt som uppstår när ett objekt befinner sig i ett roterande system. Effekten kan förklaras med illustrationen i figur 2 på sidan 9. Pilarna visar den hastighet personen måste ha för att behålla rak kurs. Om personen rör sig ut från centrum påverkas personen av en större kraft i rotationsriktningen. Om personen istället rör sig mot centrum påverkas personen av en kraft i rotationens motsatta riktning.



Figur 2: Coriolisaccelerationen

Denna effekt kan användas för att registrera om ett system roteras. Detta görs på så sätt att objektet som motsvarar personen i exemplet byts ut mot en liten massa som med en fast frekvens oscillerar mellan centrum och kanten av systemet. Genom att mäta kraften objektet utsätts för kan vinkelhastigheten av systemet bestämmas.

Figur 3: Tre frihetsgrader för ett gyro

4.2.3 Fördelar och nackdelar

Fördelen med motion sensor-teknik är densamma som med pekskärm, det vill säga att knapparna försvinner. En annan fördel är att man kan använda rörelser

som är relativt intuitiva för att representera vissa funktioner, till exempel så kan man luta telefonen åt olika håll för att scrolla i menyer. Nackdelarna med motion sensor-teknik är att det krävs en del beräkningar för att det ska fungera tillfredställande, detta gäller främst de enklare funktionerna. Vill man ha mer komplexa rörelser eller funktioner kräver det väldigt mycket beräkningar eller att man använder ett artificiellt neuralt nätverk för att känna igen olika rörelser. Denna metod berör vi närmre i avsnittet 11.1.3 på sidan 38. En annan nackdel är att det krävs en hel del finkalibrering och inställningar för att det ska fungera bra. Det är väldigt personligt hur mycket man vill röra till exempel telefonen för att rörelserna ska registreras. Applikationen måste därför ge användaren möjlighet att ställa in känslighet för att optimera användarvänligheten.

4.3 Tidigare användning av motion sensor-teknik

Den mest kända användningen av motion sensor-teknik är Nintendos senaste spelkonsol Wii. Wii var revolutionernade när det kom i och med styrning med motion sensor-teknik, vilket ingen annan spelkonsol använde. Nintendo använder sig av motion sensor-tekniken i sina handkontroller vilket innebär att man kan styra spelen genom att röra kontrollen. Ett exempel är när man spelar tennis så använder man kontrollen som ett riktigt tennisrack och slår alltså i luften med kontrollen precis som man skulle gjort ifall man spelat tennis på riktigt. Kontrollen har en inbyggd accelerometer som registrerar accelerationen i alla tre dimensionerna. Detta har givit tv-spelandet en helt ny dimension då man nu känner sig mer delaktig i spelvärlden eftersom man påverkar spelet när man rör på kontrollen. Något som många missuppfattat när det gäller Wii-kontrollen är att man inte kan få ut kontrollens position med hjälp av accelerometern, detta löser Nintendo istället med hjälp av IR-sensorer. En längre förklaring av principerna bakom accelerometern finns i stycke 4.2.1 på sidan 8

Figur 4: Wii-kontrollen registrerar användarens rörelser. Bild tagen från http://www.nintendowiiremotes.com/

Det Nintendo gjort med sitt Wii är att det skapat ett intresse för användandet av motion sensor-teknik som inte tidigare fanns. Detta har gjort att fler fått upp ögonen för tekniken och börjat fundera på vad den kan användas till. Även

Sony Ericsson har börjat integrera accelerometrar i sina telefoner. Än så länge används inte tekniken till några avancerade funktioner i telefonen, utan den fungerar till mediaspelaren i deras Walkman-version vilken möjliggör spårbyte genom att skaka telefonen. Accelerometern används även till spel som följer med telefonen. De har även en kameramobil med accelerometer där accelerometern används för att känna av ifall telefonen hålls horisontalt eller vertikal och visar innehållet på skärmen antingen i porträtt- eller landskapsvy beroende på detta.

En telefon som kallas MyDevice lanserades under 2003 av företaget MyOrigo, som numera verkar ha bytt namn till F-Origin. Denna telefon hade funktioner som utnyttjade motion sensor-tekniken. Precis som Apples mobiltelefon iPhone har MyDevice en funktion som känner av i vilket läge mobiltelefonen befinner sig i och anpassar skärmläget efter det, antingen porträtt- eller landskapsläge. I MyDevice finns även möjlighet att panorera och zooma i till exempel ett foto, ett stort dokument eller en webbsida. Detta görs genom att luta telefonen åt det håll man vill panorera. Denna teknik kallar de Iris [5].

Det har tidigare utvecklats en teknik för att göra textinmatning till små apparater med begränsat utrymme för knappar. Tekniken fungerar som så att en accelerometer känner av vinkeln apparaten befinner sig i. Utifrån aktuell vinkel och vilken knapp som trycks ner dras slutsats om vilket tecken som ska registreras [8]. Vi har använt oss av en liknande lösning i GMUI2 och GMUI3, där vi använder oss av olika vinklar för att markera/välja olika ikoner i gränssnittet.

I bärbara datorer har det funnits accelerometrar sedan ett bra tag tillbaka. Dessa används för att detektera om datorn faller mot marken. Om så är fallet låses läshuvudet i hårddisken, vilket skyddar datorn från att skadas av fallet. Ett exempel där accelerometern även används till andra saker är i Apples MacBook Pro, där det finns ett öppet API mot accelerometern. Detta har använts för att göra diverse applikationer så som ett rullande kula spel och för att göra så att datorn låter som en lightsaber från star wars när man svingar den fram och tillbaka.

Digitala systemkameror och kompakta digitala videokameror använder sig också av gyro eller accelerometrar för att kompensera för de darrningar man har när man försöker hålla en kamera på fri hand, vilket förbättrar bildkvalitén avsevärt.

Logitech har nyligen lanserat en mus med accelerometer, som heter MX Air. Detta gör att musen kan användas som en typ av fjärrkontroll och kan uppfatta en del handrörelser. Man kan till exempel skruva upp och ner volymen genom att röra den till höger eller vänster.

Figur 5: MX Air. *Bild tagen från Logitechs hemsida* [6]

5 Problemställning

Den huvudsakliga problemformuleringen inför detta examensarbete har varit:

- Är det möjligt att styra ett användargränssnitt endast med hjälp av motion sensor-teknik?
- Om det är möjligt, hur används tekniken på bästa sätt?

Vi har jobbat utifrån denna problemställning genom hela projektet. Andra problemställningar har uppkommit under projektets gång, men vi har aldrig avvikt från den huvudsakliga problemställningen.

6 Målsättning

6.1 Skapa en kunskapsbas

Vårt huvudsakliga mål med examensarbetet är att skapa en kunskapsbas för hur motion sensor-tekniken på bästa sätt används vid användarinteraktion. Utvecklingen kommer främst inriktas på gränssnitt till mobiltelefoner. Användning av tekniken som en interaktionsteknik är relativt nytt, därför behövs referensmaterial för framtida användning av tekniken.

6.2 Navigation endast med motion sensorn

Vi ska testa navigation och användning av en mobiltelefon som endast använder motion sensorn, dvs. den ska styras helt utan knappar. Användning av tekniken på detta sätt ställer höga krav på hur användargränssnittet utformas. Förfaranden som annars är självklara med ett traditionellt gränssnitt blir plötsligt en utmaning vid utveckling av ett gränssnitt som helt styrs med rörelser. Exempel på sådana förfaranden kan vara:

- Vilken rörelse ska motsvara att välja i gränssnittet?
- Vilken rörelse ska motsvara att navigera bakåt i gränssnittet?
- Hur skriver man ett meddelande med rörelser?
- Hur slår man ett nummer med rörelser?

Detta är bara några av de många svårigheter som måste tacklas. Vi testar olika sätt att göra detta på i vårt examensarbete. På detta sätt ges en bättre förståelse för vad som fungerar bra och vad som verkligen inte fungerar.

6.3 Navigation med motion sensorn och en knapp

Vi ska även testa ett gränssnitt där motion sensor-tekniken kombineras med en fysisk knapp. Vi ska testa om en kombination av motion sensor-teknik och en fysisk knapp förändrar användningen av gränssnittet till det positiva eller det negativa. Följande fördelar tror vi införandet av en knapp ger:

- Knappen löser automatiskt problemet med att kunna välja i gränssnittet.
- Navigation med en knapp blir mer självklart. Användaren känner igen sig från traditionella gränssnitt.

7 Metod

För att samla ihop data till vår kunskapsbas har vi valt att utveckla några användargränssnitt som vi sedan testar och utvärderar. Utvecklingen av användargränssnittet har behandlat såväl tekniska som designmässiga delar.

De tekniska delarna har varit:

- Hantering av accelerometerdata (Bosch-kretskort via USB).
- Integrering av accelerometer i TAT Cascades.
- Portning till Symbian Series 60.

De designmässiga delarna har varit:

- Kontinuerlig utveckling av motion-gränssnitt.
- Användartest av gränssnitten.
- Uppföljning

7.1 Utveckling

Utvecklingen inleddes med att vi hade tillgång till en accelerometer från Bosch. Vi implementerade dels kommunikation mellan PC och accelerometern (via USB) samt implementerade algoritmer för att hantera accelerometerdatan.

Nästa steg i utvecklingen var att integrera accelerometern till TAT Cascades för att på så sätt kunna använda accelerometern vid navigation i vårt användargränssnitt. Vi utvecklade en dataservice till TAT Cascades för att åstadkomma detta. TAT Cascades är ett verktyg utvecklat av TAT för att kunna skapa användargränssnitt. Utvecklingsverktyg utvecklade av TAT beskrivs närmre i avsnitt 8.2 på sidan 17.

Utvecklingen av våra motion sensor-gränssnitt genomförde vi parallellt med de andra delarna i projektet. Vi utvecklade olika varianter av gränssnitt för att senare kunna testa om någon metod var överlägsen en annan. Utvecklingen resulterade i tre olika gränssnitt.

7.2 Testning

Under testfasen var målet att fastställa vilka fördelar/nackdelar våra gränssnitt hade och om något var bättre eller sämre än traditionella gränssnitt som idag används i stor utsträckning.

7.3 Uppföljning

Användartesterna dokumenterades med bild- och ljudupptagning. I uppföljningsfasen användes det dokumenterade materialet för att dra slutsatser om användarvänligheten och upplevelsen av de olika gränssnitten.

8 Utveckling

8.1 Utveckling på TAT AB

Vi har genomfört vårt examensarbete på TAT AB (The Astonishing Tribe) i Malmö. TAT är ett företag som är specialiserat på att utveckla innovativa och tilltalande användargränssnitt för portabla enheter. Att kunna genomföra vårt examensarbete på TAT har förbättrat förutsättningar att kunna uppnå våra mål med vårt examensarbete. TAT har hjälpt oss genom att tillhandahålla kontorsplats samt genom att låta oss använda deras produkter för att enkelt kunna utveckla våra gränssnitt. Vi har även kunnat ta del av deras kunskap kring design av användargränssnitt.

8.1.1 Företaget

TAT grundades 2002 av 6 personer som hade ett brinnande intresse för visuellt tilltalande användargränssnitt där teknik och design möts på rätt sätt. TAT är ett svenskt mjukvaru- och designföretag som tillhandahåller produkter och tjänster för att kunna göra skillnad och förbättra användarupplevelsen av gränssnitt på portabla enheter. TAT möjliggör snabbare utveckling av nya och visuellt tilltalande användargränssnitt till deras kunder.

Med TATs egenutvecklade teknologiplattform, Kastor™, kan skräddarsydda användargränssnitt utvecklas oberoende av den underliggande plattformen eller det underliggande operativsystemet.

8.1.2 Produkter

TAT tillhandahåller produkter till deras kunder för att kunna skapa bättre och mer visuellt tilltalande användargränssnitt på kortare tid. TAT Cascades™ är TATs kärnprodukt och är ett komplett ramverk för att utveckla gränssnitt för de flesta portabla enheter. TAT Cascades™ bygger på Kastor™, TATs berömda UI-grafikmotor. För att ytterligare förenkla utvecklingen av tilltalande gränssnitt på kortare tid har TAT utvecklat TAT Motion Lab™, vilket är ett PC-baserat utvecklingsverktyg för avancerade användargränssnitt.

8.2 Utvecklingsverktyg

8.2.1 Kastor

TAT Kastor™ är TATs egenutvecklade grafikmotor. Kastor™ tillhandahåller en plattform för avancerade användargränssnitt för portabla enheter. Kastor™ möjliggör snabba och mjuka animationer, transitioner, wipes, effekter m.m. vilka bidrar till enkel utveckling av visuellt tilltalande gränssnitt.

8.2.2 Cascades

TAT Cascades™ är ett ramverk för att enkelt kunna utveckla användargränssnitt. TAT Cascades™ bygger på grafikmotorn TAT Kastor™.

Traditionella UI-ramverk tillåter inte förändring av användargränssnittet utan att behöva genomföra större förändringar i mjukvaran vilket ofta resulterar i långa utvecklingstider, begränsad kreativitet och en lägre kvalité av den slutliga användarupplevelsen. Med TAT Cascades™ utvecklas visuellt tilltalande användargränssnitt snabbare och ser till så att kreativiteten får stå i centrum.

TAT Cascades™ använder en designprincip som separerar applikationens logik och utseende. I kombination med användandet av ett "markup language" (TML - TAT Markup Language) som specificerar gränssnittet minimeras både mängden och komplexiteten av koden. TAT Cascades™ tillhandahåller UI-kontroller (t.ex. formulär, meny, knapp) och funktionalitet för att kunna skapa ett komplett gränssnitt.

8.2.3 MotionLab

TAT Motion Lab™ är en XML-utvecklingsmiljö för TAT Cascades. Förutom att kunna redigera och skapa användargränsnitt med XML finns det även möjlgheter att grafiskt bearbeta användargränssnitten.

8.2.4 Symbian OS

Mobiltelefonen vi använde i projektet var en Nokia 5500. Telefonen är en av få mobiltelefoner som har en inbyggd accelerometer. Telefonen använder tredje upplagan av Nokias Series-60 användargränssnitt samt version 9.1 av operativsystemet Symbian. För att kunna använda våra användargränssnitt på denna mobiltelefon använde vi oss av en integrering av TAT Cascades till en Symbianmiljö. På så vis kunde vi integrera ett gränssnitt som använde sig av telefonens inbyggda accelerometer.

Symbian OS är ett operativsystem utvecklat av Symbian ltd. Operativsystemets fördel är att det är specifikt designat för att köra på mobiltelefoner vilka oftast har begränsade minnesresurser och kan ibland exekvera oavbrutet i flera månader eller år. Operativsystemet använder endast CPUn vid behov vilket bidrar till längre batteritid. Det finns flera plattformar baserade på Symbian OS vilka förser ett SDK för utveckling mot Symbian OS. De största och mest använda av dessa är UIQ och Series 60.

Mobiltelefoner kan ha individuella utbyggnader av deras SDK vilka oftast finns för nerladdning på tillverkarens hemsida. SDKt innehåller dokumentation-, header- och lib-filer vilka behövs för att bygga mot Symbian OS. I vårt fall med Nokia 5500 fanns en utbyggnad tillgänglig på Nokias hemsida för att komma åt telefonens inbyggda accelerometer.

Utvecklingsverktyget vi använde för att utveckla applikationer i Symbian C++ var Carbide C++, vilket är ett Eclipe-baserad IDE utvecklat av Nokia.

8.3 Bosch accelerometer på kretskort

Utvecklingen av vårt nya interaktionssätt har som tidigare nämnts skett i flera olika steg. Vi valde att börja med att koppla ett kretskort med en accelerometer från Bosch [3] till en PC. För att kunna få ut mätdata från kretskortet och kunna bearbeta mätdatan skrev vi ett program i C. Det var redan lite jobb

gjort på TAT AB angående kretskortet från Bosch, så det fanns en grund vi kunde fortsätta arbeta på. På grund av detta gick själva uthämtningen av data från kretskortet snabbt och smärtfritt, vi behövde bara göra små justeringar för att få det att fungera. Datan från accelerometern skickades som tabell 1 på sidan 19 visar.

1. Byte	Start delimiter	0x20
2. Byte	Length	0x09
3. Byte	Highbyte acceleration X-Axis	0x??
4. Byte	Lowbyte acceleration X-Axis	0x??
5. Byte	Highbyte acceleration Y-Axis	0x??
6. Byte	Lowbyte acceleration Y-Axis	0x??
7. Byte	Highbyte acceleration Z-Axis	0x??
8. Byte	Lowbyte acceleration Z-Axis	0x??
9. Byte	Checksum	0x??
10. Byte	Interrupt active/inactive	0x03

Tabell 1: Presentation av dataflödet från accelerometer

Själva mätdatan låg i byte 3-8, eftersom värdena skickades uppdelade i två bytes fick vi lägga ihop dessa till ett heltal på 16 bitar. Eftersom ett vanligt heltal i C-kod, en Int, är 32 bitar och att de 16-bitars värdena vi hade var utan tecken var vi tvungna att göra om 16-bitars talen till 32-bitars tal. Vi var även tvungna att omvandla från teckenlösa tal till tal med tecken.

När vi väl kunde hämta in data från accelerometern till datorn gick vi vidare med att utveckla algoritmer för att kunna avgöra huruvida vi rörde kretskortet åt vänster, höger, upp eller ner.

När algoritmerna fungerade tillfredsställande började vi jobba på att integrera accelerometern till TATs Cascades där vi styrde ett tidigare utvecklat gränssnitt. Detta gick förhållandevis snabbt men utfallet blev inte alls så bra som vi hade hoppats på. Den dåvariga implementationen av rörelseigenkänningen visade sig ha vissa brister, vilket resulterade i att systemet registrerade felaktiga rörelser. Ytterligare problem som uppstod var det faktum att vi bara kunde registrera vänster, höger, upp eller ner, något som också var önskvärt var att kunna registrera en speciell input som ska tolkas som att man väljer i gränssnittet och en input för att gå bakåt i gränssnittet.

Vi hade två förslag varav det första innebar att man skulle dubbelklicka neråt för att välja något och dubbelklicka uppåt för att gå tillbaka. Detta visade sig vara mer problematiskt att implementera än vi först trodde, våra försök gav inte de resultat vi hoppats på utan lämnade mycket kvar att önska. Det var då vi gick över till vårt andra alternativ vilket gick ut på att man istället för att dubbelklicka skulle göra nedåt- eller uppåtrörelsen snabbare, vilket då innebar att man gjorde ett hårdare slag för att välja eller att gå tillbaka. Även denna lösning visade sig ha en del brister då det var väldigt personligt för hur hårt eller snabbt man gjorde rörelsen och det visade sig att även det blev alldeles för opålitligt för att kunna användas.

Efter dessa något misslyckade försök började vi tänka mer "outside the box" och kom då fram till att vi inte ville göra ett användargränssnitt som såg ut som de som redan finns på marknaden. Detta ledde fram till vår slutgiltiga design på de användargränssnitt som sedan användes i användartesterna. Denna design såg ut som så att vi placerade ikonerna längst kanten på skärmen och delade i och med detta in skärmen i 9 olika zoner, se 7(a) på sidan 23.

En zon för mitten på skärmen vilket var en neutral zon, sedan delade vi in resten av skärmen vänster, höger, upp, ner, vänster upp, höger upp, vänster ner och höger ner. Sedan valdes de olika ikonerna (zonerna) genom att tilta telefonen mot respektive zon. För att kunna navigera vidare i menyerna var man tvungen att hålla en liten pekare över ikonen och efter en kort stund navigerades man vidare. Detta sättet att navigera bidrog till att det lätt inträffade felnavigeringar då man råkade hamna på en ikon man inte hade för avsikt att träffa eller om man flyttade på telefonen.

Detta tog vi fasta på och gjorde en vidareutveckling av den befintliga lösningen. Vidareutvecklingen bestod i att vi la till en knapp för att välja, vilket gjorde att man inte kunde navigera fel utan att själv aktivt göra fel.

8.4 Våra motion sensor-gränssnitt

Vi har gjort tre olika gränssnitt som alla bygger på motion sensor tekniken, gränssnitt 2 och 3 är relativt lika medan gränssnitt 1 skiljer sig mer från dessa två. De olikheter och likheter som finns mellan dessa gränssnitt kommer att behandlas närmre i de följande avsnitten.

Övergripande kan man säga att likheterna mellan alla tre är att man genom rörelser navigerar i menyerna. Skillnaderna dem emellan är vilka rörelser som används för att navigera samt hur menysystemet ser ut och är uppbyggt. Gränssnitt 1 som vi har valt att kalla GMUI 1 har ett mer traditionellt menyutseende, medan vi valt att förnya menyn för gränssnitt 2 och 3 så att det ska passa motion sensor-tekniken bättre. Gränssnitt 2 och 3 har vi valt att kalla GMUI2 respektive GMUI3. Skillnaden mellan GMUI2 och GMUI3 är dels grafiska skillnader samt att i GMUI2 används inga fysiska knappar men i GMUI3 används en fysisk knapp för att välja i gränssnittet.

Med dessa gränssnitt vill vi testa om det är möjligt att skapa en mobiltelefon utan några knappar eller med några få knappar. Ett lyckat resultat kommer innebära att fysiska knappar inte behövs och tar på så sätt inte så stor plats vilket ger mer plats för t.ex. större skärmar på dagens små mobiltelefoner, där knappsatsen idag ofta tar upp större delen av platsen.

8.4.1 GMUI1

Vårt första användargränssnitt är ett försök att kombinera ett traditionellt mobilt användargränssnitt med motion sensor-navigation. Vi ville testa detta för att se om införandet av motion sensor-navigation förenklar eller försvårar handhavandet av användargränssnittet. Vi ville även se vilka funktioner tekniken lämpade sig för. Själva kombinationen var att vi tog en befintlig design på en meny, och försökte mappa om knapparna som används vid navigering till rörelser.

Figur 6: Traditionellt utformad meny i GMUI1

Knappar som huvudsakligen används vid navigation i ett traditionellt användargränssnitt är:

- Styrspak (vänster, höger, upp, ner)
- Bakåt-knapp
- Select-knapp

Vi försökte representera varje knapp med en rörelse. På detta sätt skulle det traditionella gränssnittet kunna användas med motion sensor-teknik. Rörelserna vi valde var:

- Styrspak representerades av skak (vänster, höger, upp, ner)
- Två infallsvinklar för Bakåt-knapp
  - Dubbelskak (upp)
  - Hårt skak (upp)
- Två infallsvinklar för Select-knapp
  - Dubbelskak (ner)

- Hårt skak (ner)

Vi implementerade två olika sätt att registrera rörelserna. De olika implementationerna är:

- Tröskelvärden
- Artificiellt neuralt nätverk

Det enklaste sättet att registrera en rörelse i en viss riktning är att registrera om accelerationen i någon ledd överstiger ett tröskelvärde. Möjlighet att ändra tröskelvärdet kan enkelt ges vilket medför att känsligheten kan skräddarsys av användaren. Problemet med denna rörelseigenkänningen var att för varje rörelse så registreras även en motrörelse, dvs det blir en eftersläng i mätdatan. Algoritmen måste vara utformad på så sätt att det går att särskilja på faktiska rörelser från dess efterslängar.

Att registrera rörelser med ett artificiellt neuralt nätverk (ANN) är en mer avancerad metod. Ett neuralt nätverk är ett system som är inspirerat från vad vi tror oss veta hur ett naturligt neuralt system fungerar. En av de huvudsakliga användningsområdena ett ANN kan användas för är mönsterigenkänning, dvs kunna göra igenkänning och separation av mönster. I vårt fall använde vi ett öppet bibliotek FANN [4] för att integrera ett artificiellt neuralt nätverk i vårt gränssnitt. Artificiella neurala nätverk berörs närmre senare i rapporten på sidan 38 i stycke 11.1.3

Vi tränade vårt ANN med några exempel på rörelser som vi ville att systemet skulle registrera. Det tränade artificiella neurala nätverket behandlade därefter mätdatan från accelerometern kontinuerligt för att kunna registrera om någon fördefinierad rörelse inträffade. Nätverket kände igen rörelser, men tillförlitligheten att den kände igen rätt rörelse var inte så hög som vi hade hoppats på. En möjlig anledning till att det var så stor felmarginal kan bero på att vi tränade nätverket från rådata från accelerometern. En bättre tillförlitlighet tror vi hade kunde uppnås genom att träna nätverket på skillnader mellan mätvärden istället för rådata, så att nätverket även detekterade förhållanden mellan mätvärdena.

8.4.2 GMUI2

När vi designade GMUI2 försökte vi definiera vilka ytterligare möjligheter motion sensor-tekniken ger, som annars inte finns vid användning av ett traditionellt knappstyrt gränssnitt. Några fördelar med tekniken är:

- Analoga värden
- Obegränsat antal möjliga rörelsemönster
- Naturlig mappning på triviala operationer, t.ex. scrollning.

För att kunna använda precisionen en accelerometer ger valde vi att lämna vår första rörelseigenkänning som användes i GMUI1, vilken baserades på att användaren skakade åt olika håll för att navigera. I GMUI2 navigerar användaren genom att luta telefonen åt olika håll, vilket medför att användaren får mer precision i navigeringen. Det visade sig även vara problematiskt med skaknavigationen som användes i GMUI1 eftersom användaren lätt tappade fokus från skärmen. Detta problem försvinner vid användning av en navigation liknande den i GMUI2 då telefonen lutas istället för att skakas.

Vi ville bryta mönstret med ett traditionellt menysystem baserat på en gridmeny eller liknande med gränssnittet till GMUI2. För att på bästa sätt utnyttja precisionen och den naturliga mappningen som accelerometern ger valde vi att skapa layouten på följande vis:

(a) Huvudmeny

(b) Kontaktlista

Figur 7: Skärmdumpar från GMUI2

8.4.3 GMUI3

Under tiden vi utvecklade GMUI2 lade vi snabbt märke till hur lätt det var att felnavigera i gränssnittet. För att försöka minimera antalet felnavigeringar ville vi även testa en kombination mellan motion sensor-tekniken och en fysisk knapp. Den fysiska knappen, vilken har samma funktion som select-knappen på traditionella mobiltelefoner, skulle tvinga användaren att välja när gränssnittet skulle navigera.

För att maximera den grafiska feedbacken vid navigation omarbetade vi en del av grafiken från GMUI2 för att skapa ett mer levande gränssnitt i GMUI3. Detta gjorde vi genom att förstora ikoner i gränssnittet.

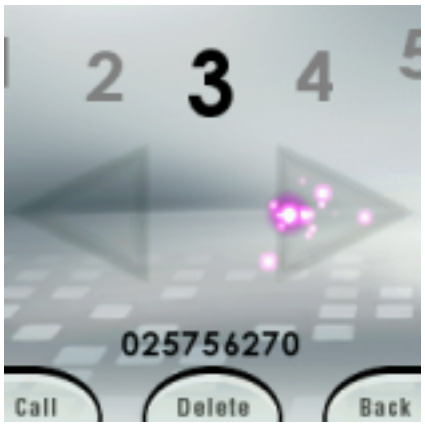
8.5 Funktioner i telefonen

8.5.1 Sifferinmatning

Vi har tagit fram två olika sätt för att mata in siffror. Det första alternativet är en form av bläddringslista, dvs man är tvungen att bläddra sig fram till rätt siffra. Vi anade att detta kunde bli jobbigt i längden och därför arbetade vi fram ytterligare ett alternativ som bygger på att siffrorna är grupperade två och två. Detta medför att man måste gör två val istället för ett men blir ändå lite snabbare och enklare än det tidigare alternativet.

Det första alternativet bygger som tidigare nämnts på ett traditionellt bläddringskoncept. Man kan alltså inte direkt välja vilken siffra man vill mata in utan måste bläddra fram till önskad siffra, det är ändå acceptabelt då antalet siffror bara är 10 stycken. Vi har dock lagt till några extra symboler som används vid speciella tillfällen, dessa är tre till antalet och det gör att totala mängden tecken blir 13 stycken. Även 13 tecken är acceptabelt att bläddra igenom. Eftersom listan är cyklisk, dvs när man kommer till slutet på listan börjar den om från början igen, kommer man som mest att behöva bläddra 6 steg för att nå önskad siffra.

Siffrorna/symbolerna som kan väljas ligger i en s.k. zoomview högst upp på skärmen, se figur 8 på sidan 24, och siffran väljs genom att vinkla telefonen uppåt, vinklar man telefonen nedåt så raderas senast slagna siffran. De slagna siffrorna visas längst ner på skärmen.



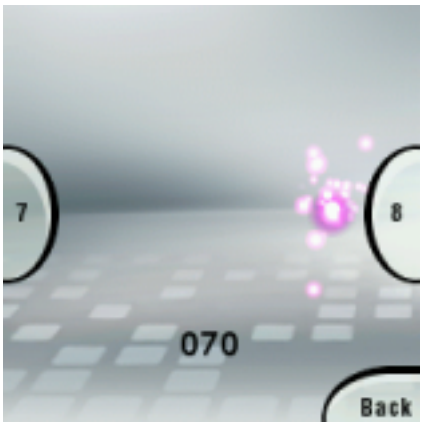
Figur 8: Alternativ 1 till sifferinmatning

Själva inmatningen är implementerad med hjälp av två listor, en överst på skärmen och en underst. Genom att ta ut vilken position man är på i den översta listan får man reda på vilken siffra/symbol som ska matas in. Detta sker via anrop från TAT Cascades till c-kod som behandlar informationen från listan. Den information man får ut från listan är som nämnts tidigare positionen i listan. Denna position avgör vilken siffra det är som ska matas in, position 0 för siffra 0, position 1 för siffra 1 osv. Siffrorna läggs sedan in i en länkad lista som

representerar telefonnumret.

Det andra alternativet som är det lite effektivare alternativet är uppbyggt av grupperade siffror runt om skärmen, se figur 9 på sidan 25. Genom att välja den grupp av siffror som innehåller den siffra man vill slå kommer man vidare till en sida där de två siffrorna är placerade på var sin sida av skärmen, figur 9(b) på sidan 25. Det är alltid så att den högsta siffran är placerad till höger på skärmen vilket gör att användaren snabbt lär sig att gå till höger ifall han/hon vill slå den högre siffran av de två, eller till vänster om det är den lägre.

(a) Val av siffergrupp



(b) Val av siffra

Figur 9: Alternativ 2 till sifferinmatning

Inmatningen är implementerad på ungefär samma sätt som i alternativ ett bortsett från att det inte används någon lista. Istället anropar vi olika funktioner beroende på vilken siffra som valts och dessa funktioner lägger in siffrorna i en länkad lista precis som i alternativ ett.

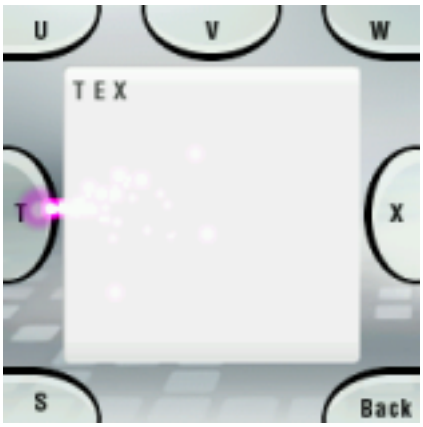
8.5.2 Textinmatning

Textinmatningen har precis som sifferinmatning två olika alternativ, varav det första är uppbyggt på samma sätt som sifferinmatningsalternativ 1, jämför figur 9 på sidan 25 med figur 10 på sidan 26. Alternativ 2 liknar sifferinmatningsalternativ 2 fast rent designmässigt ser de ganska olika ut.

Alternativ 1 är som redan nämnts i stort sett helt identiskt med sifferinmatningsalternativ 2. Skillnaden är att istället för grupper om 2 är bokstäverna indelade i större grupper, vilket medför att sidan där man väljer bokstav har fler val än i sifferinmatningen, se figur 10(b) på sidan 26. Annars fungerar det på samma sätt, vi anropar olika funktioner som lägger till bokstäverna i en länkad lista som sen visas på skärmen som ett meddelande.

Alternativ 2 har samma uppdelning i grupper som alternativ 1, men i alternativ

(a) Val av bokstavsgrupp



(b) Val av bokstav

Figur 10: Alternativ 1 till textinmatning

2 är det uppdelat i 2 olika listor. Första listan är indelad i bokstavsgrupper och är en cirkulär lista. Detta visas i figur 11(a) på sidan 26. När man valt bokstavsgrupp får man fram ytterligare en lista med bokstavsalternativen vilket visas i figur 11(b) på sidan 26. Inget i implementationen skiljer mellan alternativ 1 och alternativ 2. I båda fall anropas metoder i c-kod som lägger till bokstäverna i en länkad lista och bygger på så sätt upp meddelandet.

(a) Val av bokstavsgrupp

(b) Val av bokstav

Figur 11: Alternativ 2 till textinmatning

8.6 Problem under utvecklingen

Vi har under vårt examensarbete stött på en del problem på vägen. Det första problemet vi stötte på var egentligen inget riktigt problem men var ändå något som var tvunget att göras för att vi skulle kunna sätta igång. Detta var att

bekanta oss med de produkter TAT tillhandahöll. Vi var tvungna att skriva c-kod för att kunna integrera våra mätvärden från accelerometern in i Cascades. Vi lade mycket tid på detta men det gick efter lite slit ganska bra.

Projektet rullade därefter på och vi fick lite självförtroende. Vi beslöt då att vi skulle ge oss på att integrera vårt program på en Nokia-telefon i en symbian-miljö. Detta visade sig vara vårt största hinder på vägen då symbian är ett invecklat språk att programmera i. Till detta ska tilläggas att vi varken kunde debugga vårt program i en emulator eller på telefonen. Vår applikation kunde endast exekveras på telefonen eftersom vi programmerade mot accelerometern. Vi hade däremot inte möjlighet att debugga vårt program på telefonen vilket försvårade utvecklingen avsevärt.

Allt som allt får man väl ändå säga att projektet har gått bättre än förväntat, vi trodde att det skulle vara större problem att behandla de mätvärden vi fick från accelerometern. Det var lite problem med detta också men det löstes relativt snabbt och smärtfritt.

9 Användartest

Vi har valt att genomföra ett användartest i ett användbarhetslabb, för att på detta sätt kunna studera de reaktioner och synpunkter som användaren både medvetet och undermedvetet uttrycker. Fördelen med att genomföra användartesterna i ett labb är att vi då har full kontroll på omgivning och hur testet genomförs.

9.1 Målgrupp

Vår målgrupp är folk som tidigare använt ett traditionellt mobilt gränssnitt. Vi testar inte motion-gränsnitten på folk som aldrig använt en mobiltelefon eftersom denna grupp är underrepresenterad i dagens tekniska samhälle.

Vi har testat våra motion-gränssnitt på 8 personer. Vi anser att denna storlek på testgrupp är tillräckligt stor för att identifiera eventuella fördelar och nackdelar med motion-gränssnitten.

9.2 Testmetodik

Vi har följt en plan med fyra aktiviteter [9].

1. Introduktion. Försökspersonen får en kort muntlig genomgång av testet.
2. Testet genomförs. Testet består av en serie olika testfall. Försökspersonen går igenom testfallen under observation och utan någon direkt handledning från testledaren.
3. Summering och uppföljning av testet. Vi intervjuar försökspersonen både muntligt och skriftligt om testet. Frågorna behandlar testets svårighetsgrad, försökspersonens upplevelse, testets utformning m.m.

9.3 Testfall

Nedan följer de användarfall vi använt oss av i testet. Användarfallen är designade så att de ska testa de vyer/sidor vi skapat och som har behandlats i tidigare kapitel.

UC 1 - Ring från telefonbok

- Välj telefonboken.
- Välj person (Emilia).

- Ring upp personen.
- Lägg på.
- Gå tillbaka till huvudmenyn.

I det första användarfallet ville vi testa hur intuitivt det var att scrolla i listor med hjälp av att vinkla telefonen antingen upp eller ner.

UC 2 - Ring ett nummer (alternativ 1)

- Välj telefonboken.
- Slå in ett nummer (0708559183).
- Ring upp.
- Lägg på.
- Gå tillbaka till huvudmenyn.

Detta användarfall testade hur enkelt/svårt det var att slå ett nummer med vårt första alternativ av nummerinmatning.

UC 3 - Ring ett nummer (alternativ 2)

- Gå in i nummerinmatning alt. 2.
- Slå in ett nummer (0708559183).
- Ring upp.
- Lägg på.
- Gå tillbaka till huvudmenyn.

Detta användarfall testade hur enkelt/svårt det var att slå ett nummer med vårt andra alternativ av nummerinmatning.

UC 4 - Skriv meddelande (alternativ 1)

- Gå in i meddelande alt. 1.
- Skriv ett nytt meddelande (Hej, ska vi fika idag?).
- Skicka iväg meddelandet till en person i telefonboken.
- Gå tillbaka till huvudmenyn.

Detta användarfall testade hur enkelt/svårt det var att skriva ett textmeddelande med vårt första alternativ av textinmatning.

UC 5 - Skriv meddelande (alternativ 2)

- Gå in i meddelande alt. 2.
- Skriv ett nytt meddelande (Hej, ska vi fika idag?).
- Skicka iväg meddelandet till en person i telefonboken (Emilia).
- Gå tillbaka till huvudmenyn.

Detta användarfall testade hur enkelt/svårt det var att skriva ett textmeddelande med vårt andra alternativ av textinmatning.

UC 6 - Kolla upp e-mail i telefonboken

- Välj telefonboken.
- Välj en person (Emilia).
- Kolla efter e-mail.
- Gå tillbaka till huvudmenyn.

I det sista användarfallet testade vi hur intuitivt det är att scrolla i sidled för att bläddra i en inre lista.

En sammanställning av samtliga användarfall görs i tabell 2 på sidan 31.

9.4 Testmiljö/Testutrustning

Testen kommer att genomföras i testlaboratoriet på IKDC. Miljön benämns som ett klassiskt testlaboratorium [9]. Laboratoriet består av ett observationsrum och ett testrum som åtskiljs av en enkelriktad spegelvägg. Testledaren kommer att befinna sig i testrummet och i observationsrummet sitter observatörerna och styr inspelningsutrustning.

Testet genomförs på en mobiltelefon. Dels kommer försökspersonens uttryck och sinnestämning dokumenteras samtidigt som hans/hennes kommandon till mobiltelefonen registreras.

9.5 Intressant data

Vi kommer att mäta hur bra varje testperson klarar uppgifterna på flera olika sätt. Följande saker kommer att mätas under varje deluppgift:

Test	Uppgiftsbeskrivning	Uppgiftsdetaljer
1	Use Case 1	Reqs: Nokia 5500 sport med GMUI SCC: När den önskade kontakten är uppringt och användaren har lagt på. MTC: 1.0 min
2	Use Case 2	Reqs: Nokia 5500 sport med GMUI SCC: När det önskade numret är uppringt och användaren har lagt på. MTC: 3.0 min
3	Use Case 3	Reqs: Nokia 5500 sport med GMUI SCC: När det önskade numret är uppringd och användaren har lagt på. MTC: 3.0 min
4	Use Case 4	Reqs: Nokia 5500 sport med GMUI SCC: När hela meddelandet är inmatat och användaren har skickat iväg det. MTC: 7.0 min
5	Use Case 5	Reqs: Nokia 5500 sport med GMUI SCC: När hela meddelandet är inmatat och användaren har skickat iväg det. MTC: 7.0 min
6	Use Case 6	Reqs: Nokia 5500 sport med GMUI SCC: När personens e-mail är hittad. MTC: 2.0 min

Tabell 2: Användarfall

- Tid för att slutföra uppgift.
- Antal slutförda uppgifter.
- Hur användaren uppfattade gränssnittet.

9.6 Sammanställning av avslutande enkät

Försöksperson	UC1	UC2	UC3	UC4	UC5	UC6	Enkät:	1	2	3	4	5
GMUI 2												
Fp 1	2.30	3.15	1.20	4.00	3.30	0.30		3	4	3	3	2
Fp 2	0.30	1.45	0.50	3.30	3.00	0.15		2	5	3	4	4
Fp 3	1.00	2.20	1.10	2.15	5.00	0.30		4	4	4	3	2
Fp 4	2.00	1.30	1.45	3.15	2.00	0.20		4	5	4	3	1
Fp 5	1.00	1.40	1.15	2.15	2.00	0.20		1	4	2	3	3
Fp 6	1.00	2.00	1.40	2.30	1.45	0.20		2	2	1	4	5
Fp 7	0.20	1.50	1.00	1.40	4.00	0.45		4	4	3	3	2
Fp 8	0.15	1.40	1.00	2.00	1.40	0.40		3	4	3	4	2
GMUI 3												
Fp 1	1.10	1.40	1.00	1.50	2.30	0.30						
Fp 2	0.15	1.45	1.00	3.00	4.00	0.15						
Fp 3	0.20	1.20	1.00	3.15	4.00	0.30						
Fp 4	0.30	1.10	1.15	3.00	2.00	0.20						
Fp 5	0.10	1.00	1.30	1.30	1.40	0.15						
Fp 6	0.20	1.00	1.00	1.20	2.20	0.20						
Fp 7	0.20	1.40	1.15	2.45	2.50	0.45						
Fp 8	0.30	2.00	1.00	1.30	3.40	0.40						

9.7 Resultat av användartest

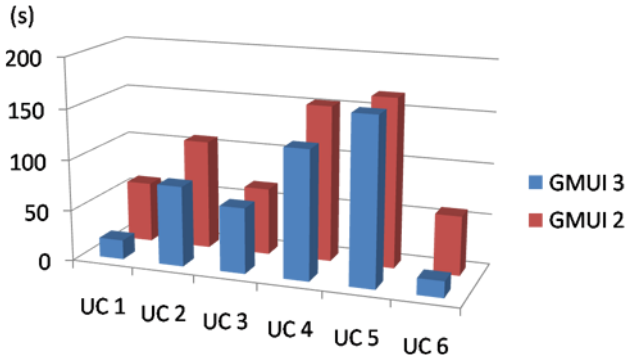
Resultaten från användartesterna visade det vi redan innan testen hade anat, nämligen att försökspersonerna inte skulle kunna utföra uppgifterna på lika kort tid eller snabbare än vad de hade gjort på ett traditionellt utformat gränssnitt. Utifrån tiderna vi mätt kan man konstatera att försökspersonerna hade lättare för gränssnittet med en knapp. Det blev mindre felnavigeringar vilket gjorde att tiderna blev kortare.

Vi lät försökspersonerna börja med olika gränssnitt för att eliminera inlärningseffekten mellan de olika gränssnitten.

En del funktioner visade sig vara ganska intuitiva, som t.ex. scrollning i listor. En del saker visade sig också vara intuitiva men de stöddes inte i vår implementation. Ett exempel på detta var att många ville klicka på det element de skulle välja. Detta är antagligen ett beteende man tar med sig från datoranvändandet med muspekare och ikoner.

Synpunkter från testpersoner tas upp i sammanställningen av enkäten i tabell 9.6 på sidan 32, och enkäten i orginalutförande hittas i appendix B.

9.7.1 Diagram och tabeller



Diagrammet visar på skillnader i tid mellan GMUI 2 och GMUI 3. Tiderna är medeltid för varje användarfall. Det man kan se i diagrammet är att tiderna för GMUI 3 överlag är kortare än tiderna för GMUI 2. Detta tror vi beror på den knapp man har för att göra sina val i GMUI 3. Felnavigeringarna blir färre med knappen och därför blir tiderna också kortare.

Hur svårt/enkelt tyckte du testet var? (1=väldigt enkelt, 5=väldigt svårt)	2,875
Hur svårt/enkelt tycker du GMUI 2 (utan knapp) var att använda? (1=väldigt enkelt, 5=väldigt svårt)	4
Hur svårt/enkelt tycker du GMUI 3 (med knapp) var att använda? (1=väldigt enkelt, 5=väldigt svårt)	2,875
Var det roligare/tråkigare än ett traditionellt gränssnitt? (1=mycket tråkigare, 5=mycket roligare)	3,375
Skulle du vilja ha en liknande funktion i din mobil? (1=absolut inte, 5=absolut)	2,625

9.7.2 Kommentarer

Första gränssnittet (GMUI2, utan knapp) var svårare att navigera och resulterade ofta i feltryck och andra småfel. Mycket smidigare att, som i andra gränssnittet (GMUI3), använda en knapp för val. Smidigt med tecken grupperade i skärmens kant istället för "scrollande" vy.

Skulle gärna vilja ha en liten tröghet i "vändningarna". Lättare att ha bara två alternativ (helt höger eller helt vänster) än att välja många i en lista. Mycket skönare att ha en knapp att bekräfta med än att råka "slinta" in på vissa alternativ och välja dem ofrivilligt.

Man vet ofta inte från början hur man ska navigera, men förstår det fort och lär sig snabbt. På detta korta testet hann man dock inte lära sig det tillräckligt bra för att det ska gå lika fort att använda som ett vanligt navigeringssystem med knappar.

Det var väldigt långsamt överlag. Om man kunnat ställa in hastigheten underlättar det nog lite. För stora "send" och "back"-knappar, de hoppade in lite hipp som happ när man kom för långt ner i gränssnittet. Man håller inte mobilen vågrätt när man använder den!

När man använder GMUI3 (en knapp) borde man inte först behöva välja siffra/bokstav och sen bekräfta; man har ju redan valt bokstaven. Mellanslag borde vara mer lättillgängligt eftersom det

används mest. När man skriver meddelande borde “back” och “delete” inte vara lika stort som det andra då man bara använder dessa vid fel och alltså inte lika ofta (förhoppningsvis). Det tar mycket längre tid att utföra alla moment än en vanlig metod men kan ju bero på att man inte är lika van vid det. Det var svårt att välja rätt namn i telefonlistan, tror det gick lite för fort, borde kanske vara mer som “hack” att det blir mer markerat liksom. Man borde kunna söka på namn/bokstav i telefonlistan, Totalt en rolig metod, men mycket klurigare.

Svårt, måste koncentrera sig. Grafiskt snyggt, vissa funktioner skulle funka men inte att ringa eller skicka meddelande.

Gillar tekniken. Lättare med “select”-knapp. Omotiverad uppdelning av bokstäver. Gillar popup-knappen i GMUI3. Vända telefonen för att avsluta samtal är j*vligt häftigt!

10 Slutsats

Vårt huvudsakliga mål med vårt examensarbete var att skapa en grundläggande kunskapsbas för hur motion sensor-teknik kan användas som en interaktionsteknik. Vi ville testa vilka möjligheter och vilka begränsningar denna tillämpning av tekniken har. Detta ansåg vi vara väldigt intressant att testa eftersom det tidigare endast gjorts mindre öppen forskning inom området.

Tekniken kan användas på många olika sätt. Vårt arbete berör bara några få sätt att använda denna typ av interaktionsteknik. Utvecklingen har även begränsats av tillgång på hårdvara vilket har påverkat vårt arbete på olika sätt.

Testresultaten visar på att tekniken i många avseenden försämrar användandet av en mobiltelefon vid jämförelse med ett traditionellt gränssnitt med knappar. Triviala funktioner som att slå ett telefonnummer eller skriva ett meddelande tar betydligt mycket längre tid och bidrar till en sämre användarupplevelse med motion-interaktion. Redan när vi började utvecklingen anade vi att tekniken inte skulle lämpas för samtliga funktioner i en mobiltelefon, men vi ville se om andra funktioner lämpade sig bättre.

En funktion som lämpade sig betydligt bättre med motion sensor-interaktion var scroll-funktionen. Användningen av scrollning när telefonen lutas är naturlig och användaren får snabbt en förståelse för hur det fungerar.

Användartesterna visade att användning av motion-interaktion bidrog till att göra gränssnittet roligare och mer intressant att använda. Detta bidrar i sin tur till att användarupplevelsen blir bättre.

En annan sak som motion-interaktion bidrar med är möjligheten att använda analoga värden för att styra ett gränssnitt. Användning av analoga värden utnyttjar precisionen i användarens rörelser.

Vi har visat att en mobiltelefon blir svårare att använda om endast motion-interaktion används, men tekniken bidrar till ett nytt och intressant interaktionssätt. Det optimala är att använda motion-interaktion i kombination med ett mer traditionellt knapp-gränssnitt för att på så sätt undvika att gränssnittet blir svåranvänt samtidigt som fördelarna med motion sensor-tekniken kan utnyttjas.

11 Diskussion

11.1 Andra tillämpningar av motion sensor-teknik

Motion sensor-tekniken kan tillämpas på många olika sätt. I vårt arbete har vi tillämpat tekniken som en navigationsteknik, men tekniken kan användas för många andra ändamål och i andra produkter än i en mobiltelefon. I detta avsnittet har vi samlat idéer och tillämpningar som vi har stött på under projektets gång för att försöka ge en övergripande förståelse för vad tekniken kan användas till. Avsnittet kan även bidra med kreativitet kring användning av tekniken.

11.1.1 Svara/Lägga på

Rörelser för att representera att användaren svarar på ett samtal eller lägger på telefonen kan användas för att skapa en naturlig användning av en mobiltelefon. En rörelse för att svara på samtal kan t.ex. vara att dra telefonen mot örat och för att lägga på ett samtal kan telefonen läggas med framsidan nedåt. Liknande rörelser kan definieras för att stänga av alarm, sätta telefon på ljudlös profil eller en liknande funktion. Genom att ge användaren möjlighet att definiera vilka rörelser som motsvarar en specifik funktion kan telefonen skräddarsys för användaren.

11.1.2 Övervaka användarens rörelsemönster

Genom att övervaka användarens rörelsemönster kan telefonen efterhand lära sig och känna igen dessa mönster. Telefonen kan då reagera på förutsedda eller oförutsedda händelser. Om telefonen känner av att användaren springer iväg kan slutsatsen dras att telefonen just stulits och därför stängs telefonen av pga säkerhetsskäl. En annan tillämpning kan vara att telefonen känner av när användaren inte rört sig på ett tag och går därför in i energisparläge för att maximera batteritiden. En mängd andra tillämpningar kan tänkas då telefonen övervakar användarens rörelsemönster.

11.1.3 Rörelseigenkänning

Rörelseigenkänning används för att definiera och känna igen olika rörelser som användaren kan använda för att interagera med ett system. En liknande teknik är teckenigenkänning vilket tidigare har använts i t.ex mobiltelefoner där tecken kan matas in genom att skriva med en penna på en tryckkänslig skärm.

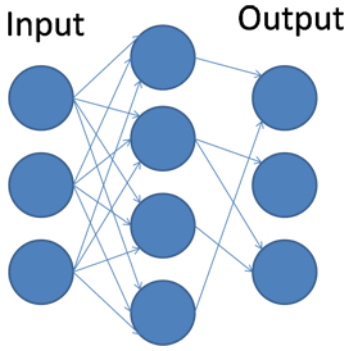
I båda fallen handlar det om en mönsterigenkänning. Detta går att implementera på olika sätt. En enklare typ av igenkänning tar hänsyn till skillnader mellan ett förutbestämt mönster för att dra slutsatsen om det var det mönstret som

användaren matade in. En mer avancerad metod är att använda ett artificiellt neuralt nätverk (ANN) som bestämmer vilket mönster användaren matade in. Ett artificiellt neuralt nätverk försöker återskapa hur vi tror ett naturligt neuralt nätverk fungerar.

Algoritmer som består av neurala nätverk kan lösa problem som ofta är svåra att lösa med konventionella datalogiska metoder. Några exempel på tillämpningar är:

- Mönsterigekänning
- Signalbearbetning
- Reglerteknik

Ett ANN går att beskriva med avancerade matematiska formler, men för att ge en mer övergripande förklaring av vad ett ANN innebär ska vi jämföra med hur vi idag tror den mänskliga hjärnan fungerar.



Figur 12: Ett ANN bestående av neuroner och viktade länkar.

Precis som den mänskliga hjärnan måste ett ANN tränas innan det kan användas. Det behövs därför en inlärningsfas där systemet tränas på den uppgift som ska utföras. Systemet kan sedan användas för att dra slutsatser baserat på tidigare inlärning. Det neurala nätverket kan även tränas kontinuerlig för att förbättra systemets slutsatser.

Ett ANN är uppbyggt av sammankopplade neuroner, precis som i den mänskliga hjärnan. Uppbyggnaden av ett ANN illustreras i figur 12 på sidan 38. I inlärningsfasen av systemet beräknas en vikt till varje länk mellan neuronerna, vilka senare ligger till grund för systemets beslutsfattande.

Vi ska i vår rapport inte gå djupare in på hur ett ANN fungerar, men för den intresserade läsaren finns några referenser till mer ingående förklaring av ett ANN [2][1].

11.2 Framtida arbete

11.2.1 Vidareutveckling av motion sensor-gränssnitt

Vi har i vårt projekt utvecklat och testat utifrån hur vi tyckte ett motion sensor-gränssnitt bör se ut. Det finns en mängd andra möjliga sätt att designa gränssnittet på, därför behövs en vidare utveckling av denna typ av gränssnitt. Att använda accelerometern för att navigera i ett gränsnitt är ett relativt nytt användningsområde, därför har teknikens sanna potential fortfarande inte uppenbarat sig. Därför behövs vidare arbete kring denna typ av navigationsteknik och gränssnitt.

11.2.2 Kalibrering

Något som hade förbättrat våra gränssnitt väsentligt är om vi hade kalibrerat telefonen till ett mer användarvänligt läge än att behöva hålla den i ett horisontellt plan. Vi har valt att inte fokusera något nämnvärt på kalibreringen eftersom det är ett stort ämne för sig, och vi skulle därför behöva lägga väldigt mycket tid på det vilket vi kände att vi inte skulle hinna.

Vi har ändå berört ämnet ytligt och har därför en del funderingar på hur det ska göras och vilka problem som kan uppstå.

Först och främst har vi kommit fram till att man i stort sett kan ha två olika tillvägagångsätt. Det första är att hitta ett läge som anses vara det vanligaste och hårdkalibrera telefonen till detta läge. Ett annat alternativ är att när telefonen varit still i en viss tid kalibrera telefonen för det aktuella läget. Båda alternativen har problem. I det första alternativet är problemet att man fortfarande måste ha telefonen i ett bestämt läge. I det andra alternativet är problemet istället att hitta när telefonen ska kalibreras.

Själva kalibreringen har vi bara funderat lite på, men vår idé är att man skulle använda sig av koordinattransformation och på så sätt vrida ett virtuellt rum med telefonen hela tiden.

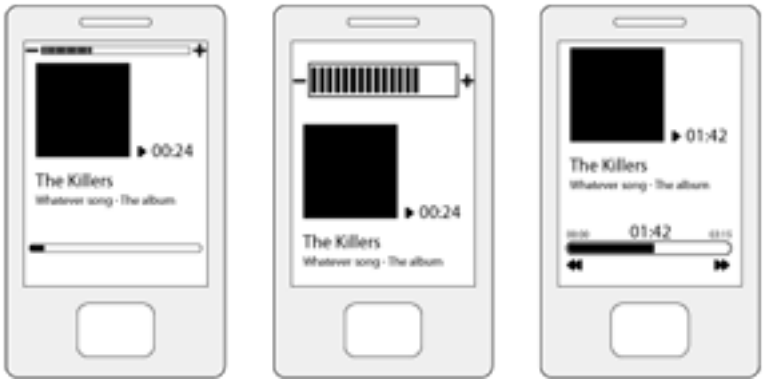
11.2.3 Integrering av accelerometrar

Mycket tyder på att accelerometrar kommer att finnas integrerade i många av framtidens tekniska apparater. Detta på grund av att de accelerometrar som finns idag redan är väldigt små och billiga, och de kommer bli ännu mindre och billigare med åren. Det kommer inte vara någon större belastning för tillverkarna att lägga till en eller flera accelerometrar. Fördelarna med att integrera en accelerometer i sina produkter överstiger de kostnader i både pengar och utrymme som uppkommer vid integrering av en accelerometer [7].

11.3 Koncept och idéer

I detta avsnitt tar vi upp andra typer av koncept och idéer kring användning av motion sensor-teknik. Vårt huvudsakliga mål med vårt examensarbete var att testa användningen av motion sensor-tekniken som en interaktionsteknik men vi har även funderat kring andra tänkbara funktioner som motion sensor-tekniken kan använda till. Detta avsnitt sammanfattar de koncept och idéer vi inte hunnit implementera inom ramarna för exjobbet.

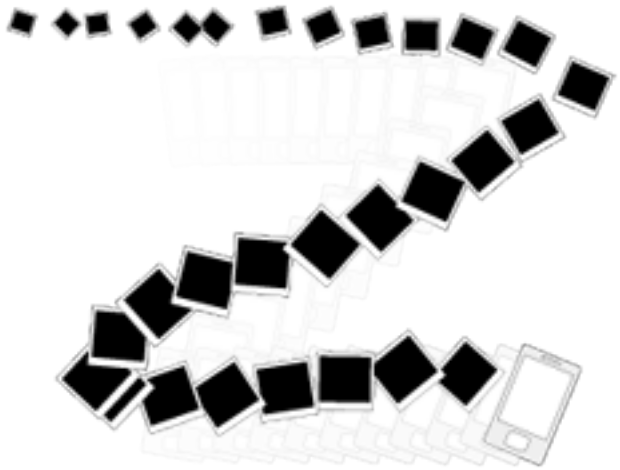
11.3.1 Musikspelare



Figur 13: Musikspelare

Tanken med musikspelaren är att man ska använda accelerometerfunktionen till att höja och sänka volymen samt att spola i låtarna. Genom att luta telefonen antingen uppåt eller nedåt blir volymfältet respektive tidsfältet aktiverat, och man kan då höja, sänka volymen eller spola genom att luta telefonen antingen åt höger eller vänster. Detta illustreras i figur 13 på sidan 40.

11.3.2 Bildspår



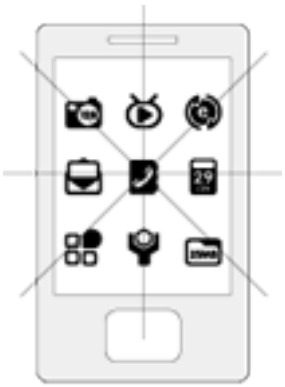
Figur 14: Användaren sprider bilderna i ett virtuellt rum.

Figur 15: Bilderna presenteras så som användaren placerat bilderna i det virtuella rummet.

Tanken är att man genom att röra telefonen i ett mönster lägger ut bilderna som finns i telefonen efter det angivna mönstret i ett virtuellt rum.

Sedan används det angivna mönstret för att visa upp bilderna som man sedan kan bläddra igenom genom att luta telefonen åt höger eller vänster. Detta illustreras i figur 14 på sidan 41 och 15 på sidan 41.

11.3.3 Fixerade fält



Figur 16: Fixerade fält

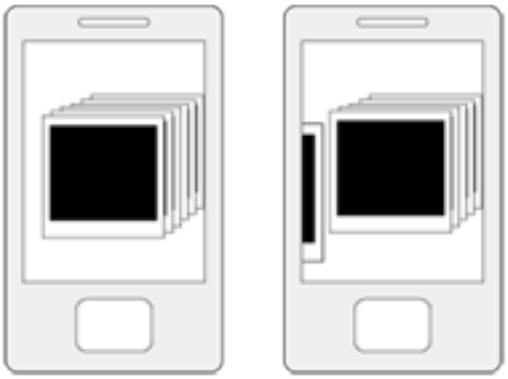
Tanken med de fixerade fälten är att man ska ha kvar accelerometern för att navigera i en meny, till exempel en grid-menu, men att man begränsar rörelsefriheten på markören till att bara följa vissa linjer. Detta gör att det blir lättare att träffa önskad ikon, då man inte behöver vara lika precis i sina rörelser. Detta illustreras i figur 16 på sidan 42.

11.3.4 Zoom-meny

Figur 17: Zoom-meny

Zoom-menyn är tänkt som en vanlig grid-meny fast istället för att visa hela menyn på skärmen visas en del av menyn och genom att luta på telefon rör sig menyn och dolda delar blir synliga. Fördelen med detta är att man kan visa mer på små skärmar eller ha många ikoner i sin meny men ändå bara visa så många att det blir överskådligt. Detta illustreras i figur 17 på sidan 43.

11.3.5 Bildgalleri 1



Figur 18: Bildgalleri 1

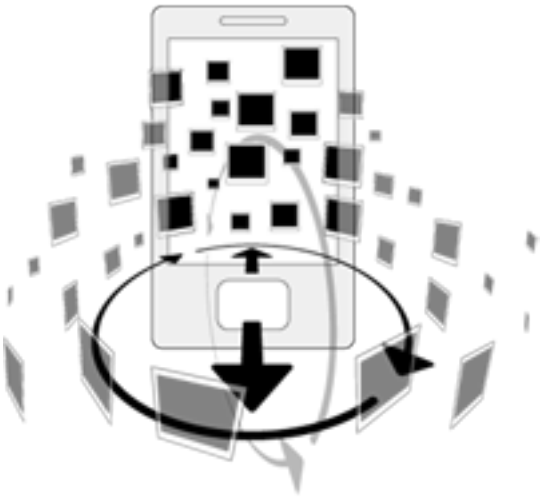
Tanken med detta bildgalleri är att man har, som bilden visar, bilderna överlappande och bläddrar mellan dem genom att skaka eller luta telefonen åt vänster eller höger. Detta illustreras i figur 18 på sidan 44.

11.3.6 Bildgalleri 2

Figur 19: Bildgalleri 2

Bildgalleri 2 har bilderna i en hög och genom att skaka på telefonen läggs bilderna ut på skärmen och man kan då bläddra bland dem genom att luta telefonen. Detta illustreras i figur 19 på sidan 44.

11.3.7 3D galleri



Figur 20: Virtuellt 3D-galleri

Referenser

[1] http://en.wikipedia.org/wiki/Artificial_neural_network,
(2007-10-24)

[2] http://sv.wikipedia.org/wiki/Neurala_n%C3%A4tverk,
(2007-10-24)

[3] http://www.bosch-sensortec.com/content/language1/downloads/SMB360_V1_1105.pdf,
(207-10-24)

[4] http://leenissen.dk/fann/,
(2007-10-24)

[5] http://www.f-origin.com/,
(2007-10-24)

[6] http://www.logitech.com/index.cfm/mice_pointers/mice/devices/3443&cl=se,sv,
(2007-10-24)

[7] *Inertial MEMS sensors for consumer applications*,
Richard Dixon and Jérémie Bouchaud (2006/05),
http://www.memsinvestorjournal.com/2006/05/inertial_mems_s.html,
(2007-10-30)

[8] *TiltType: AccelerometerSupported Text Entry for Very Small Devices*,
Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello and
Roy Want (2002)
http://portolano.cs.washington.edu/projects/tilttype/uist2002.pdf,
(2007-10-24)

[9] *Handbook of Usability testing*,
Jeffrey Rubin (1994)

A Ordlista

GMUI - Graphical Motion User Interface

ANN - Artificiellt Neuralt Nätverk

FANN - Fast Artificial Neural Network

MTC - Maximum Time to Complete

SCC - Successful Completion Criteria

Reqs - Requirements. Krav för att genomföra uppgift.

B Avslutande enkät

Hur svårt/enkelt tyckte du testet var? (1=väldigt enkelt, 5=väldigt svårt)	
Hur svårt/enkelt tycker du GMUI 2 var att använda? (1=väldigt enkelt, 5=väldigt svårt)	
Hur svårt/enkelt tycker du GMUI 3 var att använda? (1=väldigt enkelt, 5=väldigt svårt)	
Var det roligare/tråkigare än ett traditionellt gränssnitt? (1=mycket tråkigare, 5=mycket roligare)	
Skulle du vilja ha en liknande funktion i din mobil? (1=absolut inte, 5=absolut)	

Andra synpunkter eller kommentarer på gränssnitten (GMUI2 & GMUI3):

Andra synpunkter eller kommentarer på testet: