

## Grupp 4.01

Ellen Orraryd ellen.orraryd@gmail.com  
Jöns Danelius jons@mdweb.se  
Oliver Vallée olivervallee@hotmail.com

# GESTUS.

Gestus är ett system för rörelse- eller gestigenkänning. Systemet bygger på värden hämtade från mobiltelefonens accelerometer. Tanken är att man med systemet ska kunna utföra handlingar som normalt skulle kräva minst ett par knapptryck.

Rörelsebaserade gränssnitt har de fördelarna gentemot traditionella knapp-baserade gränssnitt att de inte kräver att användaren fokuserar på skärmen, de kan även vara snabba och mer intuitiva eller naturliga för användaren.

Ett rörelsebaserat gränssnitt behöver innefatta möjligheter till personalisering eftersom gester är något personligt som varierar mellan olika individer och kulturella kontexter. Användaren bör även ges möjlighet till att öva på att utföra gesterna genom att se hur en rörelse ”ska” utföras och även se hur nära han/hon gör rörelsen. Hur tredimensionella rörelser visualiseras i telefonens gränssnitt så att användaren förstår hur de ska utföras är en annan viktig aspekt av rörelsebaserade gränssnitt som vi dock inte haft tid att utveckla i den här prototypen.

## Konceptet och dess historia.

Konceptet utgick ursprungligen från tanken att tillskriva digitala data fysiska attribut och egenskaper så som tyngd, tröghet och volym. Konceptet byggde i detta stadium till största del på hur man kunde hantera och behandla data som genomgått omskrivningen till de fysiska attributen. Tankar som att kunna hålla ut, skopa upp samt kasta ut data verbaliserades. Detta koncept skulle innebära att vi använde oss av accelerometern för att registrera vilka rörelser telefonen utsattes för. Från denna punkt gick vi vidare till att fundera kring ett öppnare system. Istället för ett system där vi definierade rörelserna och dess innebörd började vi snegla på möjligheterna att skapa ett system som var öppet både vad det gällde val av gester samt dess innebörder. Vi ville göra ett system där man kunde spara sina egna rörelser och knyta dem till ett flertal valbara händelser så som att öppna/starta olika menyer och applikationer.

Konceptet i dag är:

Med hjälp av rörelsemönster/gester kunna aktivera funktioner i telefonen. Man kan skapa sina egna rörelser samt tillgå de förinställda som levereras med en naturlig och intuitiv anknytning till dess respektive funktion. Applikationen kan även användas som ett säkerhetssystem, då genom att man kan spara en komplex rörelse som man använder för att låsa/låsa upp telefonen eller funktioner.

I prototypapplikationen har vi ställt in tre gester som är kopplade till olika funktioner. Den första gesten härmar hur man vrider om en nyckel i ett lås och används för att aktivera knapplåset. Den andra gesten utgår ifrån metaforen att kasta iväg information: när man gör en kaströrelse med telefonen aktiveras bluetooth. Idén till den tredje gesten kom ifrån hur man får ut det sista av ketchupen ur en ketchupflaska genom att banka på den. Gesten symboliserar att överföra information från telefonen till andra enheter, i prototypen startar den synkronisering med en dator.

Användaren får feedback visuellt och taktilt. När applikationen känner igen en gest visas en splashscreen där det står vilken funktion som aktiverats. Samtidigt vibrerar telefonen, vibrationssignalen har olika längd beroende på vilken gest som utförts.

## Målgrupp och kontext.

### Målgrupp:

Målgruppen för denna applikationen är i slutändan mycket bred genom att den kan delas in i ett par olika grupper som kan tänkas dra nytta av den.

- Avancerade användare som använder många olika funktioner. De mer avancerade funktionerna i en telefon ligger ofta ett par steg ner i menyhierarkin, därigenom kan det vara både tidskrävande och frustrerande att navigera sig fram till dem. Genom detta system utökas antalet snabbkommandon förbi de normala antalet som knappsatsen erbjuder och skyndar därmed på arbetsprocessen.
- Mindre tekniskt bevandrade användare. Hos mindre tekniskt erfarna personer kan det uppstå problem med att navigera i menyer och appar. Den abstrakta miljön i operativ- och filsystemen kan, trots försöken (av tillverkarna) att efterlikna en konkret miljö, ställa till navigeringsproblem och svårigheter att över- och genomskåda systemen. Genom att erbjuda dem möjligheter att knyta de fåtal funktioner dessa användare oftast nyttjar till rörelser som, för dem, känns logiska hoppas vi skapa ett system som är lättare att ta till sig för denna grupp.
- Fysiskt begränsade personer. Mobilen är ett fantastiskt redskap, möjligheterna och funktionerna växer i ett rasande tempo i takt med den tekniska utvecklingen. Tyngdpunkten hos de flesta produkterna och applikationerna ligger dock ofta på att nå de så kallade "early adapters" samt teknikintresserade. Med denna applikation kan man med samma telefoner locka användare som har svårt att interagera med dagens mobiler med små tangenter och skärmar.

### Kontext:

Eftersom man kan knyta applikationen till valfria funktioner samt att den har en så bred målgrupp så blir även kontexten inom vilken den kommer att jobba ganska bred. Då systemet är ett alternativt sätt att navigera det redan befintliga funktionerna i telefonen kommer inte applikationen att aktivera eller skapa några nya kontexter utan snarare istället i viss utsträckning förändra den kontext den verkar och agerar i.

## Teknisk beskrivning.

Vår prototyp består av två applikationer: Gestus och Gestus\_config som båda läser accelerometerdata från telefonen. Detta sker med hjälp av ett Pythonscript som sedan läses av javaapplikationerna. När Gestus körs startas en tråd som läser accelerometerdata och kontrollerar ifall värdena motsvarar någon av de sparade gesterna med hjälp av if-satser. När en rörelse identifierats skickas man till en splashscreen som illustrerar den funktion man aktiverat.

När man startar upp applikationen så kommer man direkt till huvudmenyn 'main canvas', därifrån kan man välja mellan fem olika alternativ:

### [Mainmenu]

Practice gesture	----->	[submenu]		
Record gesture	-->	splashscreen	Keylock	----> video form ---> gest igenkänning
Manage gesture	-->	splashscreen	Bluetooth	----> video form ---> gest igenkänning
Exit application	-->	splashscreen	Synchronization	----> video form ---> gest igenkänning

Beroende på vart du har din markör i menyn så ändras texten från italic till bold, även en vit halvtransparent png-bild följer med i bakgrunden för att förstärka effekten vart man befinner sig i menyn.

Beroende på vilket alternativ man har valt i huvudmenyn så skickas man antingen vidare till en undermeny eller till en ny splashscreen. Det finns fem olika splashscreens, en för varje meny alternativ samt två forms, dessa två forms finns under Practice gesture, i den första formen så visas en filmsnutt beroende på vilken gest man valde i submenyn. Filmen kommer att visas ett oändligt antal gånger ända tills man tycker att det är nog och klickar sig vidare, när man klickar för att komma vidare till den sista formen (gestigenkänningen) så startas även en tråd som avlyssnar accelerometerdatan och går sedan igenom ett antal if-satser som avgör om du har gjort rätt gest som du har valt.

Det utan tvekan största problemet var att identifiera gester, framför allt om detta skulle ske i realtid, kontinuerligt. Medan vi programmerade letade vi parallellt efter metoder, de vi fann var:

Neurala-nätverk ([http://sv.wikipedia.org/wiki/Neurala\\_n%C3%A4tverk](http://sv.wikipedia.org/wiki/Neurala_n%C3%A4tverk)) en teknik som fått sitt namn från dess funktion att försöka efterlinka det sätt en hjärna arbetar. Dessa nätverk jobbar med en mängd noder i ett eller flera plan vilka sedan knyts samman med banor mellan dessa, banorna och noderna viktas sedan för att komma fram till ett resultat. Dessa systemen jobbar i två faser, inlärningsfasen samt användningsfasen, dock finns det inget som hindrar att fas ett fortgår under fas två även om detta inte är brukligt. Exempel på tillämpningar är: mönsterigenkänning, signalbehandling, prognoser och självorganisering, som vi snabbt blev varse var detta ingen acceptabel lösning, dels på grund av att komplexiteten gör att det krävs långt mer än två veckor att bemästra och dels på grund av telefonens, i detta sammanhanget, begränsade prestanda.

Hidden Markov Models eller dold Markovmodell på svenska ([http://en.wikipedia.org/wiki/Hidden\\_Markov\\_Models](http://en.wikipedia.org/wiki/Hidden_Markov_Models)) var metod två vi tittade på. Denna modell bygger på att beräkna sannolikheten för en variabel utifrån en variabel även den baserad på en sannolikhet osv. Denna metod används idag bla. för att räkna på taligenkänning, tidsseriedata och mönster i molekylära sekvenser som till exempel DNA och proteiner. Även detta alternativ visade sig ganska snabbt vara mer lämpat för en matematisk doktorand än för detta tvåveckorsprojekt.

Efter detta bakslag fick vi nöja oss med den ovan nämnda metoden som använder sig av tröskelvärden.

### Vidareutveckling:

Under arbetets gång har vi utvecklat en hel del bitar som inte i dagens läge finns med i systemet men som kan tänkas komma till nytta under en fortsatt utveckling av projektet. För närvarande är dessa, en metod som sparar ner accelerometervärdena i en textfil (vilken vi använt för att bestämma tröskelvärdena manuellt), en readfile-metod som hämtar upp värdena från en textfil och returnerar en array av objekt ur den skapade klassen position. Till denna klass finns det även ett flertal metoder för kunna göra olika jämförelser mellan två eller flera objekt. Vi hade tänkt använda dessa metoder för att jämföra de "grafer" i x, y och z-led som genereras jämfört med sparade grafer för rörelser, tre för varje rörelse. Vi hade tänkt jämföra punkt för punkt och stega oss igenom graferna, genom att använda de olika compare-metoderna hoppades vi kunna hitta en lösning som skulle kunna ge en tillfredsställande igenkänning, mycket tid har gått åt att titta på dessa alternativ. I dessa metoder används också tre variabler som anger toleransen på värdena, dvs hur långt ifrån punkterna får vara varandra för att bli godkända. Ett alternativ kring dessa variabler var att dessa skulle sparas ned i textfilen för rörelsen tillsammans med grafvärdena, varje rörelse skulle då kunna ha sin egen tolerans beroende på hur lätt den var att efterlinka. Det största problemet med denna metod var att kontinuerligt och i realtid jämföra utförd rörelse mot textfiler samt att justera graferna så att de "ligger i fas". Om en tilltänkt vidareutveckling innebär en fördjupning, då troligtvis med viss hjälp av matematiker, i neurala nätverk eller dolda Markovmodeller å ena sidan eller en lösning med grafer å andra sida vet vi i skrivande stund inte.

## Gränssnittet och diskussioner där kring.

Utgångspunkten för gränssnittet var att hitta ett formspråk som signalerar rörelse och luftighet. Valet för på en blå bakgrund som ger associationer till en vattenyta, en vattenytan är i ständig rörelse, rörelserna är naturliga och logiska, egenskaper som ville införa i vår applikation. För att visa våra sparade rörelser valde vi att spela in korta videoklipp där respektive rörelse illustreras. Tanken är dock att i den färdiga produkten så ska detta ersättas av en datorgenererad bild av en telefon som rör sig i ett 3D-grid baserat på banan som är rörelsen. Denna förändring är nödvändig för att kunna illustrera de rörelsemönster användaren själv skapat. Andra alternativ för detta var att visa värdena för accelerometern som grafer eller kurvor men detta alternativ skulle vara för svårt för användaren att förstå sig på och tolka. Eftersom systemet bygger på ett tangent- och skärmlöst navigerande och användande så finns det inte så mycket mer att tillägga om denna biten.



## Projektgruppen.

Gruppen består av Ellen, Jöns och Oliver. Arbetsfördelningen som snarare växte fram än klubbades har sett ut som följer. Ellen har haft huvudansvaret för att spara ner värden från accelerometern samt att kalibrera tröskelvärden för rörelserna, hon har även skött uppspelningen av filmklippen. Oliver har fokuserat på skapandet av klassen position och dess flera comparemetoder samt skapandet av en filereader som returnerar en array av positionobjekt. Jöns har ansvarat för den grafiska biten i projektet samt flödet mellan dess olika element. Utöver detta har vi alla tre deltagit i utformandet av navigeringsflödet samt interaktionen i systemet. Vi har även alla tre deltagit aktivt i de många långa diskussioner kring de olika matematiska beräkningsalternativen och finkammat Internet efter metoder och lösningar på vår uppgift.