

# Performance Based Musicmaking



## Short

Via mijn thesis, "Performance based Musicmaking" onderzoek ik hoe men de expressiviteit van een laptopmuzikant kan vergroten door zijn lichamelijke beweging als input te gebruiken voor zijn muziekinstrument. Dit project resulteert in twee prototypes die elk hun speciale karakteristieken hebben en een platform bieden voor verdere experimenten.

## Performance Based Musicmaking

Student: Wout Standaert

Promotoren: Liesbeth Huybrechts, Lieven Menschaert

Afstudeerrichting: Communicatie & MultimediaDesign



Jaar: 2009

Online: <http://www.blobkat.com>

# Abstract

Muziekperformance met laptops heeft altijd zekere begrenzings gekend. Op het podium zijn ze eerder een noodzakelijk kwaad dan een visueel expressief iets. Ze werken meestal hoofdzakelijk met een visuele interface, waardoor de muzikant gekluisterd zit aan zijn scherm. Ook is de laptop niet mobiel en dit dwingt de muzikant bijna om achter zijn laptop te blijven staan. Er zijn ongelofelijk veel mogelijkheden om muziek te maken met een laptop, maar het gebrek aan een vrijere manier om een laptop te besturen is momenteel een groot nadeel.

Uiteraard hebben al meerdere mensen die problematiek proberen aan te raken via nieuwsoortige interfaces, maar daar gaan we later verder op in.

Muziek wordt gecomponeerd om een ervaring te creëren, of die nu cognitief, emotioneel of sensorisch is. De muzikant creëert dus een muzikale ervaring (en live

de hierbij horende performance) en de toeschouwer neemt hopelijk zo veel mogelijk van die ervaring op. Als de muzikant echter niet los komt van zijn laptop en controllers, laat dit een sterk visueel expressief aspect weg, waardoor de toeschouwer mogelijk minder vervoerd wordt door de live performance in het algemeen.

Tegelijk zijn mensen ook veel op zoek naar nieuwe muzikant interfaces. Het homemade - eigen gemaakte - instrument biedt een manier tot personaliseren van de performance door de muzikant. Met open-sourcetechnologieën - technologieën waarvan de broncode vrij beschikbaar is - zoals Arduino en technieken als circuitbending maakt een muzikant van een instrument iets heel persoonlijks waarmee hij zichzelf uitdrukt.

Op vlak van bewegingsdetectie zijn er enorm stappen gemaakt voor de consument via de introductie van de Nintendo Wii. Mensen sluiten thuis Wii Remotes aan op hun computer om via beweging hun

computer te besturen. Deze Wii Remotes zijn zeer goedkoop, maar bevat heel nuttige hardware die gemakkelijk te gebruiken is voor eigen doeleinden.

Met deze thesis wil ik daarom op zoek gaan naar een geadapteerd muzikant instrument voor muziekperformance. Ik stel de onderzoeksvraag :

“Hoe kunnen we een digitaal muzikant instrument maken met behulp van een laptop dat lichamelijke beweging als input gebruikt om zo een lage leercurve te hebben voor de muzikant en een sterk expressief aspect naar het publiek toe.”

Mijn doel is om deze drie puntjes; nl.

- de mogelijkheid van geadapteerde muzikant interfaces
- de beperkingen van de laptop
- en de gemakkelijke beschikbaarheid van de wiimotes

te combineren in een visueel expressief instrument dat “gemakkelijk” te gebruiken is.

Ik ga pogen om de ervaring die de muzikant wil creëren te vertalen naar bewegingen die muziek en mogelijk ook beeld aansturen.

Mijn veldwerk neemt de vorm aan van case-studies en analyses. Via een analyse van de huidige stand van zaken over het onderwerp ga ik kijken waar ruimte is voor verbetering en hoe mijn prototypes die verbetering kunnen betekenen. Door te ontwerpen ga ik mijn conclusies kunnen maken, dit heet “research through design”.

Om mijn prototypes te maken pas ik de techniek van het iteratief ontwerpen toe: eens ik een vorm van prototype heb ontwikkeld ga ik het beoordelen en laten testen door muzikanten en niet-muzikanten waardoor ik mijn conclusies kan trekken voor het volgende prototype.

# Contents

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Abstract                                                           | 4  |
| Race In                                                            | 9  |
| Part 1: Literatuurstudie                                           | 10 |
| Human Computer Interaction                                         | 13 |
| Ubiquitous computing                                               | 14 |
| Affordances                                                        | 15 |
| Tactiele en Haptische feedback, materialiteit (Tangibility)        | 15 |
| Interaction Design                                                 | 19 |
| Pleasurability                                                     | 21 |
| Computers as Theatre                                               | 23 |
| Do It Yourself                                                     | 24 |
| Circuit bending & customization                                    | 26 |
| Open Source                                                        | 27 |
| Gesturale interfaces                                               | 29 |
| Part 2: Technologisch overzicht                                    | 34 |
| Technologieën                                                      | 35 |
| Part 3: Kwalitatief onderzoek                                      | 42 |
| Methodologie                                                       | 43 |
| Muzikale interfaces voor computers nu - analyse                    | 44 |
| Case studies: Toepassingen van gesturale interfaces binnen muziek. | 51 |
| Wii remotes als input van menselijke beweging                      | 54 |
| Ervaringen met gesturale interfaces tijdens dit masterjaar         | 59 |
| Part 4: Prototypes                                                 | 64 |
| PRE-PRODUCTION: Wii Remote data-analyse                            | 65 |
| PROTOTYPE 0: Trigger events by acceleration                        | 69 |
| PROTOTYPE 1: Convert movement into sound                           | 70 |
| PROTOTYPE 2: Trigger events by performing own gestures             | 74 |
| Race Out                                                           | 79 |
| Media                                                              | 80 |
| Bibliografie                                                       | 83 |





Battles

## Race In

Deze thesis sluit zeer nauw aan bij mijn persoonlijke interesses en opvattingen over muziek en is het logische gevolg van de projecten die ik in dit masterjaar heb ondernomen. Los van het feit of het nu touchscreens, multi-touch tafels, wiimotes, of sensorbanken waren, de rode draad van mijn mastertraject was de zoektocht naar en het experimenteren met aparte, originele interfaces, in combinatie met geluiden. Zo was bijvoorbeeld onze "artgame" Glasgow Mega Snake sterk verbonden met het snerpande geluid dat de "snake" maakte.

Telkens bij het presenteren van één van de projecten merkte ik hoe veel effect de wisselwerking tussen geluid en beeld in interactieve projecten kan hebben op de algemene sfeer van een project. Net als bij film kan een dreigende soundscape direct een setting creëren die dreigend aanvoelt.

Interessanter wordt het voor mij echter als het geluid niet de installatie ondersteunt, maar de twee samenwerken om een zo groot mogelijke totaalervaring neer te zetten. En zelfs bij live muziek, waar de sound centraal staat, is er sprake van een totaalervaring. Muziek fuseert hier met sfeer, visuele aantrekkingskracht en wordt zo iets magisch waar mensen over de hele wereld geld aan spenderen.

Deze thesis gaat uit van de hypothese dat bewegingsvrijheid van een muzikant de fusie tussen muziek en andere sensorische ervaringen (m.a.w. de totaalervaring) kan ten goede komen.





# Part 1: Literatuur- studie

## Inleiding

Als C-m'd'er bekijk ik mijn onderzoek vanuit het standpunt van een designonderzoeker. Meer specifiek in dit geval het veld van **Human Computer Interaction (HCI)**. HCI is de studie van de interactie tussen mensen en computers. Daarom verken ik enkele inzichten van vooraanstaande personen in het vakgebied.

Ik maak een **interface die werkt met menselijke beweging**. Een interface is dat wat tussen een mens en een computer in zit zodat deze met elkaar kunnen communiceren. Als deze bewegingen van de mens de traditionele muis en keyboard overstijgen noemt men dit soort interfaces **"Gesturale Interfaces"**. Ik bespreek enkele gebruiken en theorieën die er tot nu toe rond ontstaan zijn.

Ik gebruik met mijn prototypes een bestaand product en verander de functie ervan om uiteindelijk een nieuw product te maken. Daarom bespreek ik ook enkele punten van de **"Do-It-Yourself" (DIY)** beweging en hacking. De kenmerken van deze bewegingen houden in dat ze producten die niet zijn gemaakt om te veranderen toch aan hun noden gaan proberen aan te passen. Dit is ook wat ik ga proberen met de **Wii Remotes**.

Ik heb besloten om mijn software gratis te verspreiden, met inzicht van de broncode (**"open source"**). Open source houdt in dat elk ontwerp of elke programmacode vrij beschikbaar is en vrij gebruikt mag worden in nieuwe ontwerpen. Het concept van open source is niet alleen toepasbaar op

software, maar heeft ook vele toepassingen in cultuur.

Het is bij wetenschappelijk onderzoek ook belangrijk om de gevonden inzichten vrij beschikbaar te maken voor derden, zodat deze inzichten verder gebruikt kunnen worden in andere onderzoeken. Ook worden mijn prototypes gemaakt met open source software en heeft "open source" raakvlakken met "DIY". Daarom biedt mijn literatuurstudie ook enkele inzichten op open source.

Tot slot bespreek ik enkele methodologieën die mij helpen om inzichten te verwerven tijdens het schrijven van deze thesis en verschillende programma's en technologieën die mij helpen bij de uitvoering van de prototypes.





Xerox Alto

# Human Computer Interaction

**Human Computer Interaction (HCI)** is een veld dat even oud is als de computer, maar niet altijd die benaming had. Al sinds de eerste computers worden er manieren uitgedacht om op een (gemakkelijke) manier mee te interageren.

In het begin was er een een-op-een relatie tussen mens en computer door de knoppen en wijzers aan de voorkant van enorme machines zoals de ENIAC (Walker, 1990); mensen gaven input en kregen direct output van de computer. Met ponskaarten en automatische lijstverwerking werd deze directe interactie vervangen door een computeroperator. Tijdsdelen en “glass teletypes” (“glazen elektromechanische typmachines”, beeldschermen) introduceerden opnieuw die directe communicatie en hebben geleid tot de command-line en menu-interfaces (zoals bijvoorbeeld DOS).

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>SET ULTRASND=240,3,3,5,5
Z:\>SET ULTRADIR=C:\ULTRASND
Z:\>dir
Directory of Z:\.
COMMAND COM          20 01-10-2002 12:34
AUTDEXEC BAT         84 01-10-2002 12:34
KEYB COM             20 01-10-2002 12:34
IMGHOUNT COM         20 01-10-2002 12:34
BOOT COM            20 01-10-2002 12:34
INTRO COM            20 01-10-2002 12:34
RESCAN COM           20 01-10-2002 12:34
LOADFIX COM          20 01-10-2002 12:34
MEM COM              20 01-10-2002 12:34
MOUNT COM            20 01-10-2002 12:34
MIXER COM             20 01-10-2002 12:34
CONVIO COM           20 01-10-2002 12:34
DOS                  304 Bytes.
                      0 Bytes free.
Z:\>
```

Brenda Laurel zegt in haar boek *Computers as Theatre* dat deze simpele manier van converseren (persoon doet iets - computer reageert) ertoe geleid heeft dat interaction designers een model van interactie hebben ontworpen dat mens en computer als twee aparte partijen beschouwt, wiens conversatie door het scherm wordt gemedieerd (Laurel, 1991). Er is, zegt ze, nog een extra dimensie aan een conversatie, namelijk het begrip van de **“common ground”**. Dit houdt in dat beide partijen een groot deel informatie en kennis delen, waarop ze hun acties baseren. Deze samenwerking zorgt ervoor dat de eigenschappen, doelen en gedrag van beide onverbrekkelijk met elkaar verstrengeld zijn.

Deze “common ground” ondersteunt het idee dat een interface een gedeelde context is waarin beide partijen “agents” zijn. Brenda Laurel gebruikt het begrip “agent” zoals Aristoteles het gebruikt in de *Poetica*, namelijk “diegene die de actie begint”.

Interaction designers begonnen dan metaforen uit de echte wereld te gebruiken in hun design, om voor bepaalde programma’s hun functie te vergelijken met een functie die de gebruiker reeds kende en zo een “common ground” te scheppen. Zo is er het begrip van de prullenmand, dat nu in verschillende besturingssystemen wordt gebruikt als metafoor om iets te verwijderen.

Alan Kay vond de **GUI** (Graphical User Interface, grafische gebruikersinterface) uit in 1973 in dienst van het Xerox Palo Alto Research Center (PARC). Deze interface

gebruikte de WIMP (Window, Icon, Menu, Pointing device) stijl van interactie (Booth, 2008).

De metafoor van een “bureaublad” is uitgevonden door Tim Mott terwijl hij voor PARC werkte. Deze doorbraak is van groot belang geweest voor het succes van de eerste Apple personal computers, de Lisa en uiteindelijk de (meer betaalbare) Macintosh. Deze metafoor wordt nog steeds door de



Windows en Mac OS X besturingssystemen gebruikt.

Over de jaren heen zijn er verschillende paradigma's (modellen) ontwikkeld voor het ontwerpen voor graphical user interfaces. Verder in deze thesis worden enkele visies hierop besproken.

## Ubiquitous computing

Gesturale interfaces krijgen de laatste jaren steeds meer aandacht als middel om te streven naar “**ubiquitous computing**” (**ubicomp**). Ubiquitous computing houdt in dat vele objecten in het dagelijkse leven computers zullen bevatten die zo geïntegreerd zijn met de objecten en de

omgeving rondom ons dat ze onmerkbaar zijn. Gesturing zou hierbij dan een oplossing kunnen zijn voor interfacing; een computer die onmerkbaar mensen analyseert in een huis kan reageren op gebaren van die mensen. Wuiven naar de gordijnen kan bijvoorbeeld een teken zijn om ze te openen of te sluiten. De locatie van personen kan bijgehouden worden door een computer en elke kamer kan aan hun persoonlijke voorkeuren worden aangepast.

Een ander voorbeeld van deze analyses die computers kunnen doen is “gait analysis”. “Gait” is de postuur van iemand en hoe hij loopt. Momenteel lopen er verschillende onderzoeken om via mensen hun postuur hen te identificeren. Toepassingsgebieden zijn bijvoorbeeld de beveiliging van luchthavens.

Rob van Kranenburg schrijft in zijn “The internet of things” over ubiquitous computing:

“Electricity was the actual metaphor that the EU 1st project officer, Jakub Wechjert, used. He spoke of a vision of the future is one in which our everyday world of objects and places become ‘infused’ and ‘augmented’ with information processing. Computing, information processing, and computers disappear into the background, and take on the role more similar to that of electricity today - an invisible, pervasive medium distributed on or real world. In contrast, what will appear to people are new artifacts and augmented places that support and enhance activities in natural, simple and intuitive ways.”

## Affordances

Voordat we kunnen interageren met een gesturaal systeem moeten we weten dat er een is en hoe we kunnen beginnen met interageren. Hier komt het begrip “**affordance**” in het spel. Een “affordance” is volgens Don Norman een of meerdere eigenschappen van een object die een indicatie geven over hoe er met dat object moet omgegaan worden (Norman, 1988). Een knop heeft een affordance om in te drukken, door de manier waarop het kan bewegen. Uitzicht en textuur zijn de belangrijkste indicaties van affordances.

Affordances zijn origineel door psycholoog James Gibson vastgelegd als alle mogelijkheden van acties die met objecten mogelijk zijn voor een bepaald individu (Gibson, 1977). Een ladder van twee meter hoog heeft bijvoorbeeld niet de affordance om beklommen te worden als het individu in kwestie een baby is.

Het verschil met wat Norman zegt is dat volgens Gibson alle mogelijke acties een affordance hebben, terwijl Norman het heeft over de meest mogelijke acties. Als een man een kamer binnenkomt waarin een bal op een stoel ligt, zijn het gooien van de stoel en het zitten op de bal eveneens affordances volgens Gibson. Norman zegt hier dat het meer waarschijnlijk is dat de persoon de bal gooit en in de stoel gaat zitten.

Hij beschrijft dus affordances als de meest waarschijnlijke actie van een object, en relateert dit aan de bruikbaarheid (usability) en het algemene ontwerp ervan.

## Tactiele en Haptische feedback, materialiteit (Tangibility)

Een belangrijk punt bij het ontwerpen van interfaces die bestuurd worden met ons lichaam is om op een bepaalde manier naast de gewone visuele feedback van het scherm ook nog **tactiele** of **haptische feedback** te geven.

**Tactiel** wil zeggen voelbaar. Het verwijst naar feedback die we kunnen voelen met de toppen van onze vingers. Een voorbeeld is de verhoogde streepjes op de F en J toets van het toetsenbord. Onze vingers gebruiken dit als referentiepunt voor positie en daarom slaagt men erin om blind te typen. Een ander voorbeeld is hoe verschillende makers van aanraakschermen zoals in de iPhone beginnen te experimenteren met elektriciteitsvelden die delen van een aanraakscherm anders doen aanvoelen, waardoor de vereiste van een aanraakscherm om het constant te bekijken tijdens het gebruiken wegvalt.

**Haptisch** wil zeggen dat de interface bepaalde frictie, kracht of beweging uitoefent op de gebruiker. Haptische feedback is in games bijvoorbeeld bekend als “Force Feedback”. Haptische feedback is wat ons een indicatie geeft over hoe bepaalde interface-elementen zich gedragen. De weerstand die een draaiknop geeft, geeft ons een goed inzicht over hoe ver we die knop al gedraaid hebben.

Tactiele interfaces zijn dus een subcategorie



van Haptische interfaces.

Dan is er nog het begrip “**tangible**”. Tangible is vrij vertaald “**materieel**” of “**tastbaar**”. Als een interface-element een fysiek volume heeft dat we kunnen vastnemen (in tegenstelling tot bijvoorbeeld een knop op een aanraakscherm) is het tangible. De ReacTable (later besproken bij de analyse van muzikale interfaces) is een voorbeeld van hoe een multi-touch scherm tangible elementen kan gebruiken.

Tangible user interfaces nemen een stuk data of functionaliteit en belichamen dit dan in een fysiek object dat manueel kan gemanipuleerd worden. De “**Poken**” is een goed voorbeeld van hoe een **object een stuk data kan voorstellen**, in dit geval contactgegevens. Door met dit object fysiek om te gaan (een poken fysiek tegen andere pokens duwen) wordt een actie uitgevoerd, namelijk het uitwisselen van contactgegevens.

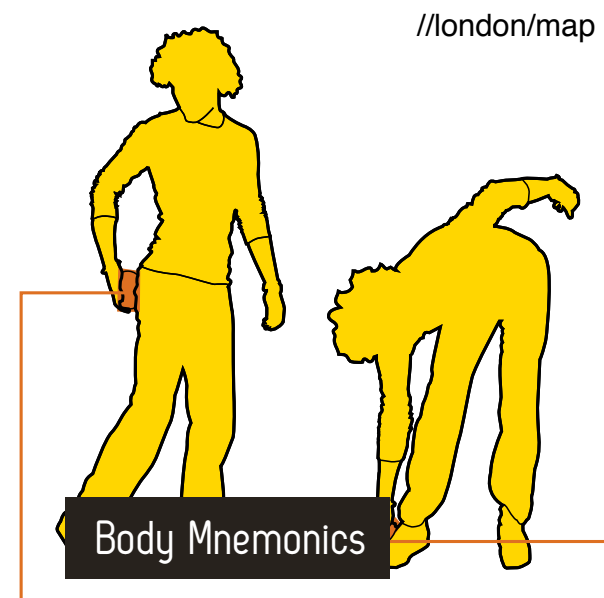
Picture by Alexander Klink



Deze drie soorten user interfaces, **tactiel**, **tangible**, en **haptisch**, zijn belangrijk voor gesturale interfaces. Free form gesturale interfaces hebben bijvoorbeeld **weinig feedback**. Het lichaam is compleet vrij in het bewegen zonder restricties, terwijl die

er soms misschien wel zouden moeten zijn.

In zijn thesis “Body Mnemonics” gebruikt Jussi Ängeslevä het lichaam zelf als begrenzing en feedback. Gestures worden uitgevoerd met een eindpunt op het lichaam, en hebben afhankelijk van de plaats op het lichaam hun functie.



Ängeslevä stelt het project voor in de vorm van een gesturale interface die de accelerometers in GSM's gebruikt om gestures te herkennen. Contactpersonen in de gsm kunnen aan bepaalde lichaamsdelen gekoppeld worden. Als de gebruiker bijvoorbeeld zijn partner wil bellen, moet hij de telefoon naar zijn hart toe brengen en het tegen zijn borst kloppen. De borst van de gebruiker vormt een eindpunt en geeft feedback over de gesture. Ook is het belangrijk als **mnemonisch hulpmiddel**, het helpt om de gesture te onthouden (Ängeslevä, 2004).

Een ander experiment met een free form gestural interface is het VR/TX Vibrotactiele Stimulator Project van Joseph Rovin en Vincent Hayward. Zij gebruiken een

trillende ring rond een van de vingers van de gebruiker om een aanwijzing te geven



over zijn positie in de interface.

Ze schrijven in hun verhandeling over de interface het volgende:

“In het algemeen, zou men de volgende observaties kunnen maken over de meeste free form controller systemen:

- Afhankelijkheid van visuele feedback
- Afhankelijkheid van proprioceptie, het vermogen van ons lichaam om de positie van het lichaam tegenover andere lichaamsdelen waar te nemen.
- Afhankelijkheid van egolocatie, het vermogen van ons lichaam om te voelen of waar te nemen waar we ons bevinden.

De visuele feedback kan hier versterkt worden door het gebruik van computerschermen die allerlei indicaties geven. Maar er zijn nadelen aan visuele feedback verbonden, namelijk:

- Visuele feedback is meestal enkel nodig door beginnende performers, ervaren performers

vertrouwen zelden op hun zicht.

- Computerschermen zijn dikwijls onaangepast en onpraktisch voor performance.
- Er zijn veel belangrijkere visuele elementen waarop een performer moet letten tijdens de performance: interacties met het publiek, interacties met andere performers, het lezen van muziek en dramatische aspecten
- Mechanische kanalen van feedback geven strakkere en preciezere respons dan visuele feedback, zowel psychologisch als fysiologisch.

**Proprioceptie** en **egolocatie** zijn beiden dus vitale componenten in het brede scala aan feedback dat nodig is voor free form gesturale interfaces. Toch komen ze allebei tekort als er op moet vertrouwd worden als enige indicatie van oriëntatie. Redenen om een zware afhankelijkheid van proprioceptie en egolocatie te vermijden zijn:

- Het is moeilijk om consistente controle en herhaalbare resultaten te bekomen.
- Vereist lange training en speciale vaardigheden.
- Nooit accuraat in absolute coördinaten.”

Deze nood aan feedback zal later verder worden besproken bij mijn prototypes.

# Interaction Design

Human Computer Interaction beslaat niet alleen het veld van grafische gebruikersinterfaces, maar behelst elke manier waarop met een computer kan geïnterageerd worden. Bill Moggridge heeft als eerste de term “**Interaction Design**” gebruikt, en hij definieert het zo:

“Een nauwe definitie van Interaction Design is: “het design van de subjectieve en kwalitatieve aspecten van alles dat beide digitaal en interactief is, door designs te maken die nuttig, wenselijk en toegankelijk zijn.” De designer werkt in de artificiële context van bits, pixels, inputapparatuur, conceptuele modellen van de gebruikers, en het organiseren van metaforen.

...

Een bredere definitie van Interaction Design is “het design van alles dat zowel digitaal als interactief is” ”

Moggridge, stichter van IDEO, geeft ons in zijn boek Designing Interactions een brede blik op de inzichten van verschillende designers. Hijzelf en Duane Bray zien drie categorieën van Interaction Design, namelijk:

## Schermbaseerde interfaces

**Grafische elementen gebaseerd op pixels.** De ontwerper manipuleert pixels om software-interacties uit te drukken. Dit komt ook terug in ontwerpen voor het internet, aangezien websites zich vooral manifesteren als schermgebaseerde interfaces.

## Interactieve producten

In de tweede categorie is het fysieke object **geïntegreerd in de elektronische hardware en software**. Als er een scherm gebruikt wordt, moet de designer letten op de relatie tussen fysieke besturingselementen en de algemene vorm van de interface op het scherm. Als er geen scherm is, vertrouwt het design op feedback door de omgeving via licht, geluid of beweging.

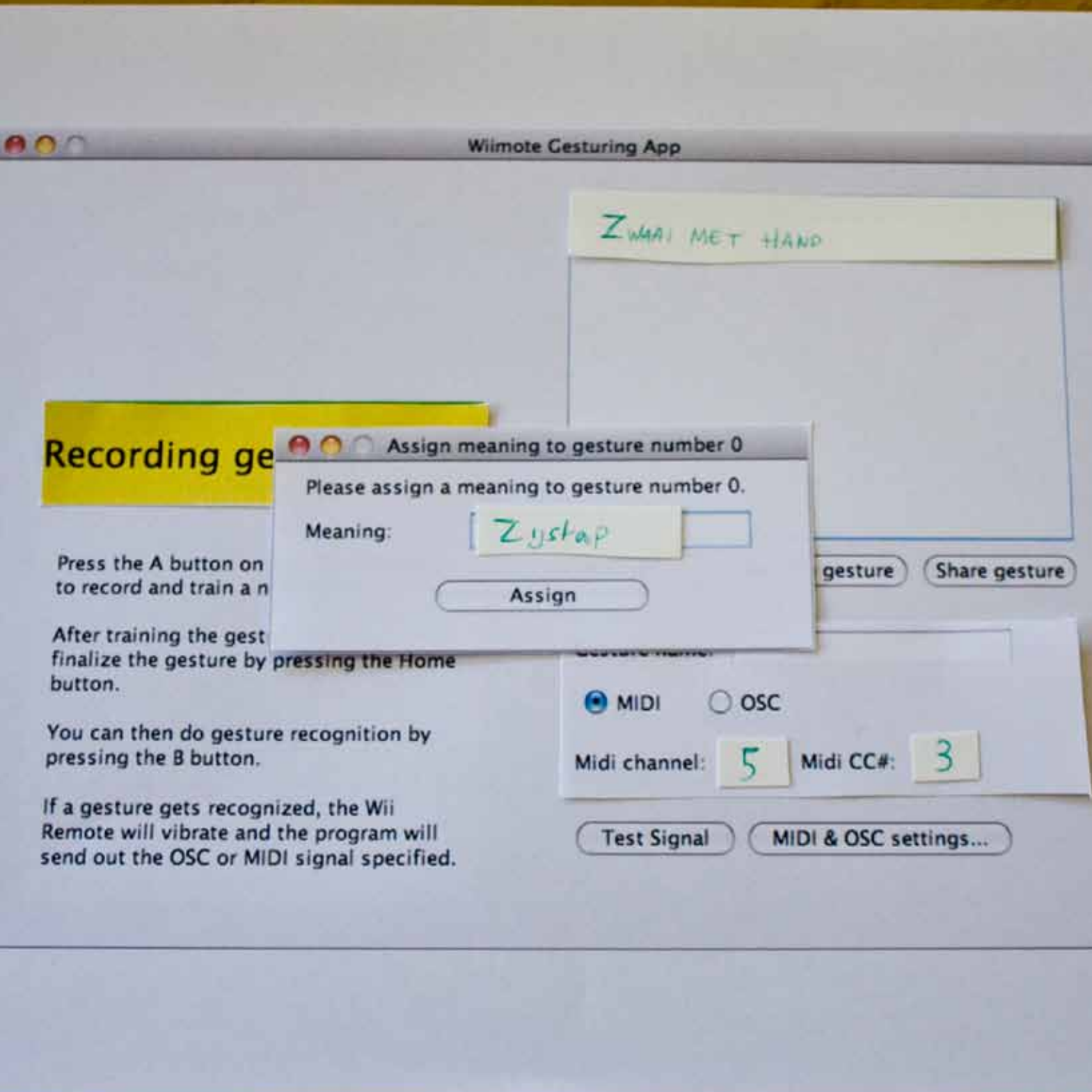
## Diensten

De derde categorie behelst het design van diensten (services), waar de **interactiviteit gebeurt tussen een bedrijf en een bredere relatie met de klant**. Hierin worden tijdgebaseerde interacties samengevoegd met verschillende kanalen – ruimtes, producten, internet, enzovoort. Dit vervaagt de grens tussen Interaction Design en bedrijfspsychologie (Moggridge, 2006).

Hij geeft ons enkele richtlijnen om voor deze drie categorieën prototypes te ontwerpen. Ik bespreek hiervan de schermgebaseerde interfaces en de interactieve producten aangezien ik hiermee te maken ga krijgen in mijn designproces. Mijn gebruik van de Wii Remotes maakt mijn prototypes tot **interactieve producten**, terwijl de laptop dient als **schermgebaseerde interface**.

## Schermbaseerde interfaces

Moggridge begint met de schermgebaseerde interfaces. Hij stelt dat de eerste grafische ontwerpen op computers aanvankelijk gelimiteerd waren door hun lage resolutie,



vooral voor de weergave van lettertypes.

Toen het gebruik van het internet explosief toenam, kwamen vele ontwerpers uit de printwereld en waren ze gewend voor 300 beeldpunten per inch (dpi) te werken in plaats van voor de 72 van het scherm. Daarom waren vele eerste websites van bedrijven gewoon digitale exemplaren van hun brochures in print, waardoor de kwaliteit van de afbeeldingen en de tekst achteruit ging. Al gauw namen bedrijven zoals Razorfish het voortouw om te specialiseren in ontwerpen die onder andere meer focusten op experience dan slechts print designs omgezet voor het web. Met het gebruik van Flash, Java en geanimeerde .GIF bestanden behoorde ook animatie tot de mogelijkheden.

Aanvankelijk verschilden software ontwerp sterk van website ontwerp maar naarmate de functionaliteit en de mogelijkheden van de verschillende webtechnologieën toenamen begonnen ze steeds meer op elkaar te gelijken.

## Vroege verkenningen

Moggridge benadrukt het belang van **vroeg prototypes** maken, maar niet noodzakelijk in het uiteindelijke medium van het project. Hij stelt dat designers vaak veel te vroeg beginnen te werken met hun uiteindelijke middelen en zich te vroeg concentreren op alle visuele details van een interface in plaats van te experimenteren met usability (Moggridge, 2006).

## Paper Prototyping

Hij haalt hier de techniek van het “**papieren-prototype**” (paper prototyping) aan. Papier kan effectief ingezet worden als prototypemateriaal. Het is toegankelijk en veelzijdig. Interaction Design concepten kunnen snel geschetst worden met papier om elementaire usability te testen. Zo wordt het mogelijk om snel design concepten te organiseren, uit te werken, en visualiseren.

De structuur en het gedrag van een interface kan ook gedemonstreerd worden via papieren prototypes, door de verschillende componenten van de interactiviteit te scheiden. Elementen die gescheiden gaan zijn op het scherm kunnen voorgesteld worden door aparte stukjes papier, bijvoorbeeld voor een pop-up of een pull-down menu.

Zo kunnen mogelijke functionaliteiten worden gedemonstreerd met de hand, zonder tijd te verspillen aan het programmeren van animaties.

Moggridge zegt ook dat dit soort prototypes ook **eerlijkere reacties** uitlokken van testpersonen. Omdat de prototypes er niet perfect uit zien en dus weinig emotionele waarde hebben voor de ontwerper geven mensen er eerlijkere kritiek op.

Het is ook belangrijk voor de ontwerper dat hij de hele workflow van de interface die hij ontwerpt in het hoofd houdt. Hiertoe kunnen storyboards gemaakt worden die een lineair verhaal schetsen van een handeling.

Om de ontwerpideeën te communiceren

kunnen prototype tools worden gebruikt zoals Macromedia (nu Adobe) Director of zelfs PowerPoint, hoewel de focus de laatste jaren verschuift van Macromedia Director naar Adobe Flash.

Zodra deze ideeën zijn vastgelegd kunnen de “echte” prototype tools ingezet worden. De ontwerper kan beginnen coderen in HTML of applicaties maken in Flash die steeds meer het eindresultaat benaderen.

## Interactive products

De tweede categorie die Duane Bray en Moggridge beschrijven is in de context van interactieve, tastbare producten.

Hierbij wordt een fysiek object geïntegreerd met elektronische hardware en software. Mensen interageren dan met “verborgen” computers, ze hebben eerder de indruk dat ze met een responsief of “intelligent” object aan het werken zijn. Dit is iets waar ontwerpers met ubiquitous computing naar streven.

Bij het prototypen van deze producten kunnen simpele materialen zoals plasticen doosjes of stukken schuimrubber gebruikt worden in combinatie met eenvoudige elektronische circuits die de functionaliteit van het uiteindelijke product moeten simuleren.

Er bestaan verschillende elektronische prototypetools. Een van de tools die snel aan populariteit wint is de Arduino, een microcontroller die zowel op zichzelf kan werken als in verbinding met een computer via USB voor meer complexe functionaliteiten. Via vrij verkrijgbare

sensoren en outputmodules kan er een enorme hoeveelheid aan functionaliteiten snel gesimuleerd worden.

De derde categorie, die van de diensten, is in het bereik van deze thesis niet van toepassing.

In de bespreking van mijn methodologieën heb ik het verder over prototypes.

## Pleasurability

Don Norman heeft een nog ruimere blik op interaction design; voor hem is interaction design het ontwerp van **elke manier waarop met alledaagse objecten wordt omgegaan**, niet alleen met elektronische interfaces. Zijn inzichten zijn toch sterk gelinkt aan en toepasbaar op HCI.

Hij bekijkt in het ontwerpen **3 niveaus van emotionele en rationele verwerking door de mens** (Norman, 2004). Deze drie niveaus zijn direct het gevolg van ons evolutionair traject. Eerst is er het **viscerale** niveau. Dit niveau heeft vaste, snelle routines om te volgen. Het brein analyseert de wereld rondom de mens en kan snel oordelen of er bijvoorbeeld gevaar is. Dit deel van verwerking zorgt ervoor dat we overleven op het meest basale niveau. Het zorgt er bijvoorbeeld voor dat we (soms irrationele) angst hebben van hoogtes, omdat hoogtes evolutionair als gevaarlijk zijn bestempeld.

Het niveau daarboven is het **gedragsniveau** (behavioral). Dit niveau zijn de aangeleerde reflexen en vaardigheden, zoals autorijden. Dit niveau is onbewust, waardoor we tijdens



het autorijden over andere dingen kunnen nadenken. Goede performers gebruiken dit niveau om bepaalde activiteiten onbewust te kunnen doen. Dit zorgt er bijvoorbeeld soms voor dat een pianist of gitarist zijn plaats op de bladmuziek verliest en het bewust niet meer kan vinden. Zijn vingers deden immers al het werk.

In het derde niveau, het **reflectieve** niveau, kan het menselijk brein nadenken over zijn eigen werking. Dit is de bewuste gedachte, en dit kan zo sterk zijn dat bepaalde sterke viscerale impulsen kunnen overwonnen worden, zoals het bezoek aan een achtbaan, dat angst veroorzaakt, maar we gaan er toch in voor de spanning, en om er over te kunnen opscheppen achteraf. Hier gaat de viscerale angst rechtstreeks in conflict met het reflectieve plezier.

Alles wat we doen heeft daarenboven een cognitief en affectief effect, **cognitief** om er betekenis aan toe te schrijven, **affectief** voor waarde. Dit affectieve is niet te ontwijken, het gebeurt sowieso. En het heeft consequenties voor ons gedrag.

Het positief affectieve brengt ons in een staat van kalmte en deze staat stimuleert onze creativiteit. Daarom beginnen brainstormsessies ook op een luchtige basis, als alle participanten op hun gemak zijn, zijn ze creatiever. Het negatief affectieve (zoals een gevaarsituatie) zorgt voor momenten van sterke concentratie en sterke focus. Dit kan in projecten gebruikt worden in de vorm van net iets te korte deadlines, waardoor de angst ervoor zorgt dat het werk klaar geraakt.

Dit moet in acht genomen worden bij het

ontwerp van interfaces en objecten. Als een vliegtuig vliegt zoals gepland, mag de cockpit rustig ogen, de piloten voelen zich op hun gemak. Als er iets fout gaat begint er een rood indringend licht te flikkeren en een alarmsignaal te snerpen. De piloot verhoogt zijn focus in deze situatie en kan snel handelen. Zijn creativiteit is nu echter verminderd, waardoor hij minder goed zelf oplossingen zal kunnen bedenken. Daarom worden noodprocedures eindeloos herhaald en gerepeteerd zodat de piloten quasi automatisch kunnen handelen vanuit hun gedragsniveau.

Nielsen, Norman's partner in de Nielsen – Norman onderzoeksgroep, focust doorgaans enkel op functionaliteit en usability. Hij zegt bijvoorbeeld voor websites dat ze grafisch niet persé goed ontworpen moeten zijn, omdat grafiek alleen maar als doel heeft om mensen te lokken, en die mensen toch al voor de website hebben gekozen. Dit is hetzelfde pad dat Don Norman lang heeft bewandeld voor hij “Emotional Design” schreef. **Maar functionaliteit en usability speelt slechts in op één van de drie niveau's, namelijk het gedragsniveau.**

Norman stelt dat **aantrekkelijke dingen beter werken**, al is het maar schijn dat ze beter werken (Norman, 2004). Hij haalt het voorbeeld van een recensie van de Mini Cooper aan: De recensent poneert de stelling dat de Mini Cooper zo plezierig is om mee te rijden, dat je over de fouten heen kijkt. Zo is er ook het voorbeeld van de ingenieur die in een hotel de snelheid van de lift moest verhogen omdat de gasten klaagden dat ze te lang voor de gesloten deuren moesten staan. Uiteindelijk

installeerde de ingenieur spiegels op de deuren van de lift, waardoor mensen naar zichzelf konden kijken tijdens het wachten. De klachten hielden op, de ervaring van het wachten werd een pleasurable experience.

# Computers as Theatre: theatertheorie als model voor een ubicomp toekomst

Brenda Laurel past met “Computers as Theatre” de dramatische structuren van het theater toe op HCI. Gebruikers zijn actoren op een podium samen met de interface-actoren, waar alles dat de gebruiker niet moet zien in de coulissen zit. **“All that matters is that which is illuminated”** (Laurel, 1991).

Ze geeft vier oorzaken waarom de dingen worden gemaakt waarom ze zijn, en past dit toe op het theater en HCI (Zie Tabel).

| Oorzaak   | Algemeen                                                                            | Theater                                                                                                                                        | HCI                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Formeel   | De vorm van het object, het overzicht, de visie van het uiteindelijke object        | De complete plot, alle actie dat de theatermaker probeert voor te stellen                                                                      | Een voorstelling van acties tussen agents, menselijk, computergestuurd, of beide.                                                                                                     |
| Materieel | Waar het object van is gemaakt                                                      | De dingen waaruit een theaterstuk bestaat. De geluiden en bewegingen van de acteurs als ze zich over het podium bewegen                        | Datgene dat zich ontplooit voor de zintuigen van de persoon, zoals beelden, geluid, muziek, tekst en zelfs tactiele en kinesthetische effecten                                        |
| Efficiënt | De manier waarop het gemaakt wordt. Dit behelst zowel de maker en de gereedschappen | De vaardigheden, gereedschappen en technieken van de toneelschrijver, acteurs en andere artiesten die bijdragen tot het afgemaakte theaterstuk | De vaardigheden en gereedschappen van de makers van de interactie. Dit is meestal een groep mensen die samenwerken vanuit verschillende disciplines                                   |
| Einddoel  | Het uiteindelijke nut, wat het moet doen eens het voltooid is                       | De aangename opwindning en expressie van bepaalde emoties in het publiek (catharsis)                                                           | Wat het moet doen eens de activiteit is voltooid, niet alleen qua functionaliteit, maar ook ervaringsgewijs; wat een persoon denkt en voelt is even belangrijk als de functionaliteit |

Tabel samengesteld uit gegevens uit Computers as Theatre

Zij ziet dus zoals Don Norman een **nood aan pleurability en experience in human computer interaction**.

Het vergelijken van HCI met theater zorgt ervoor dat we niet meer gaan denken in termen van “een interface is een scherm op een computer en we klikken op knoppen en iconen” maar meer dat een interface verschillende actoren, verschillende agents kan hebben die op ons inspelen en vice versa, wij die als actor inspelen op de actoren van de interface, of de interface die inspeelt op ons. De interface kan dan een ruimte rondom ons zijn, een ruimte waarin we ons vrij kunnen bewegen, zoals het theater.

De link naar free-form gesturale interfaces (zie hoofdstuk “Gesturale interfaces”) en ubiquitous computing is dus snel gelegd. Zodra computers elke beweging die we maken kunnen volgen met verschillende sensoren en hierop dan kunnen inspelen wordt de interface onzichtbaar, en gaan we intuïtief en onbewust met de computers rondom ons interageren.

## Do It Yourself

In DIY: The Militant Embrace of Technology beschrijft Marcin Ramocki hoe we vervreemden van onze producten. Karl Marx had het al over de “alienation of labor”, de vervreemding van ons werk. De arbeiders werken aan de lopende band in een kleine rader van het geheel en worden ontkoppeld van hun eindproduct. Ze worden de vruchten van hun werk onzegd, ze krijgen er geen emotionele voldoening van. Dit

scenario is nog steeds geldig voor vandaag de dag, zegt Ramocki, al ligt de nadruk nu op onze witteboordenmaatschappij, door de consumptie van artikelen geproduceerd in ontwikkelingslanden en de economie van “**intellectual property**”, immateriele dingen die verhandeld worden, zoals computerprogramma’s en kennis (Ramocki, 2008). De nieuwe vervreemding is volgens hem de vervreemding van consumptie en grondstof, wat logisch is, aangezien consumptie de witteboordenarbeid is volgens Ramocki.

De gebruikers hebben geen inzicht meer in de motivatie achter een productie en het bestaan van handelswaren. Ze weten niet hoe het gemaakt is, waar het gemaakt is en wat de eigenlijke waarde is van de waren binnen hun eigen referentiekader.

Hackers zijn echter in staat om deze “intellectual properties” te **vervormen**. McKenzie Wark heeft zo het Hacker Manifesto gepubliceerd, een aanpassing van het Communistisch Manifest, dat het begrip van intellectual property hierin verwerkt. In het post-kapitalistische tijdperk van de Verenigde Staten en West-Europa zijn de economieën verschoven naar auteursrechten, patenten, en handelsmerken.

Hackers zijn hier in staat om deze informatie **creatief te vervormen tot iets nieuws**. Zoals McKenzie Wark in zijn Hacker Manifesto zegt:

“Whatever code we hack, be it programming language, poetic language, math or music, curves or colourings, we create the possibility

of new things entering the world. Not always great things, or even good things, but new things. In art, in science, in philosophy and culture, in any production of knowledge where data can be gathered, where information can be extracted from it, and where in that information new possibilities for the world are produced, there are hackers hacking the new out of the old. While hackers create these new worlds, we do not possess them. That which we create is mortgaged to others, and to the interests of others, to states and corporations who control the means for making worlds we alone discover. We do not own what we produce – it owns us.”

Het **nut** van hacken is dus simpel gezien het **creëren van iets nieuws**. Wark ziet hier het begrip “hacker” als iets heel ruims, elk individu dat betrokken is in het uitvinden van nieuwe concepten is een hacker, inclusief hedendaagse kunstenaars, wetenschappers, programmeurs. Ramocki stelt hier dat de hackers dus in direct conflict zijn met de “vectoralisten”, de macro-managers van de digitale economie, de mensen die de evolutiepaden van producten vastleggen voor de markten van de toekomst.

Hackers doordrenken zich met de kennis en informatie over de productie, zowel op hardware als softwarevlak. Zo overwinnen ze de vervreemding van arbeid en de vervreemding van grondstof. Gekoppeld hieraan is hun politiekactivistische inslag door hun conflict met die vectoralistische agenda.

Onder de werken van hackers valt onder meer het concept van circuit bending. Dit geeft het apparaat dikwijls een **nieuw doel**

(educatief speelgoed zoals de speak ’n spell wordt omgebouwd tot muziekinstrument) en **nieuwe interne werking**. Het geeft ook aan de hacker een **uniek product**, dat hij zelf heeft aangepast tot vervulling van zijn noden en wensen. Ook hackerkunst en het veranderen van structuren van een game (zoals grote DIY mods die bouwen op de interne werking van een spel maar het spel zelf helemaal veranderen qua inhoud, of door middel van tools die geleverd worden bij een spel om zelf mee te bouwen) komen veel voor.

Het uitzoeken hoe producten werken en er zelf mee aan de slag gaan lijkt dus de officiële taak te worden van de kunstgemeenschap. Het werk van duizenden creatieve hackers brengt veel meer nieuwe dingen teweeg dan het werk van één researchteam in een bedrijf.

Caroline Nevejan schrijft in het door Liesbeth Huybrechts geredacteerd boek “CrossOver” over het **vervagen van de grens tussen professional en amateur door deze deling van kennis**:

“Een nieuwe dynamiek die het internet heeft veroorzaakt, is de verandering van de wijze waarop de amateur en de expert zich tot elkaar verhouden. Kennis ontstaat waar mensen delen wat ze weten. Doordat dit nu op wereldschaal kan gebeuren, is de snelheid waarmee kennis – ook gespecialiseerde kennis – wordt geproduceerd heel groot geworden.”

Deze kennisdeling zorgt dus voor een grote informatiestroom naar niet alleen leken, maar ook producent toe. De opgedane kennis bij hacking is kennis die

niet openbaar zou mogen zijn, maar het (met enige moeite) toch geworden is. Om mensen de moeite te besparen, kunnen we als producent zelf het initiatief nemen om kennis te delen. Dit is waar het begrip “Open Source” van toepassing is.

## Circuit bending & customization

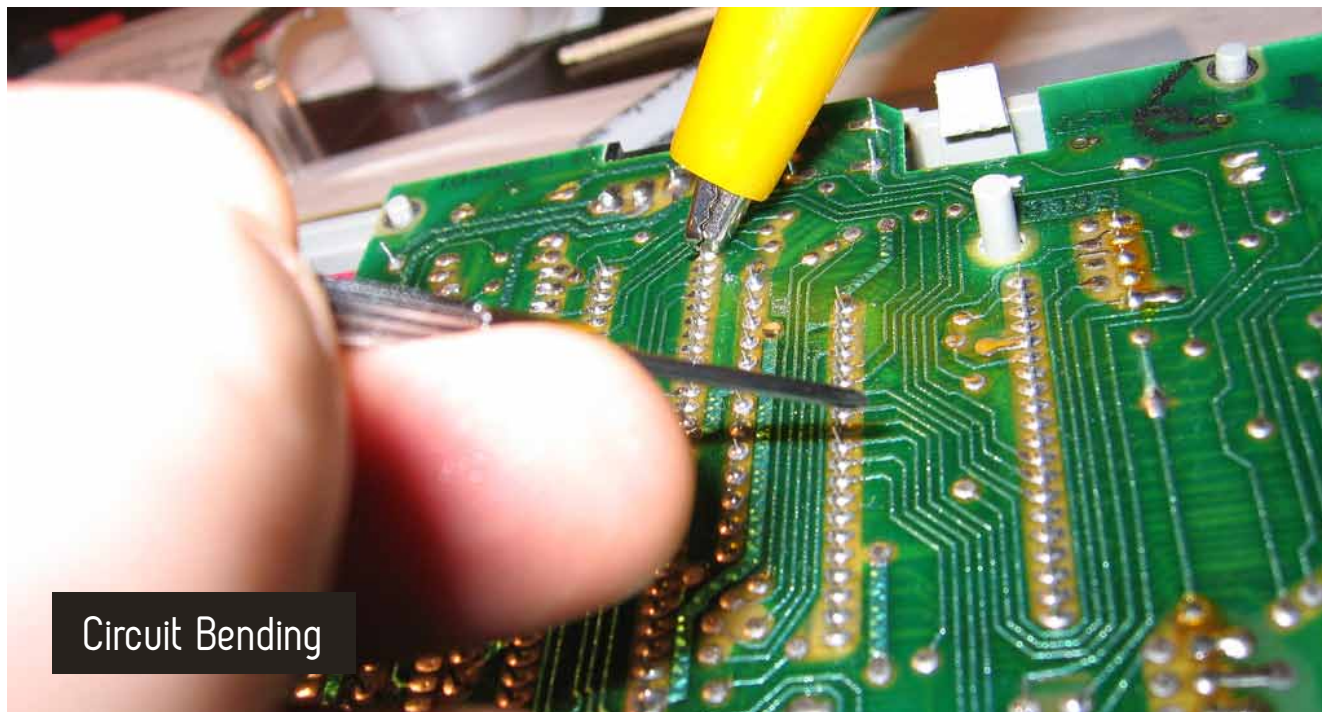
**Circuit bending** is (vermoedelijk) in de jaren 60 gecreëerd door Reed Ghazala. Het is een creatieve manier om bestaande elektronische apparaten kort te sluiten waardoor hun output anders wordt dan vooropgesteld door de fabrikant. Ghazala heeft de techniek toevallig ontdekt via een speelgoedje dat was blijven liggen in zijn bureau waarvan de behuizing open was geraakt. De interne contacten van het speelgoed raakten een stuk metaal aan waardoor het een geluid begon te produceren dat leek op synthesizers van die tijd.

Het kans-element speelt een grote rol in circuit bending. Het houdt in dat men een stuk speelgoed of andere elektronische circuits die geluid produceren gaat **openmaken** en **elektrische contacten leggen die normaal niet gelegd zouden worden** (Ghazala, 2005).

Door zelf te experimenteren kan een muzikant zijn **eigen unieke instrumenten** construeren.

Ook het circuitbenden van apparaten die video uitsturen is mogelijk. Onder meer Gijs Gieskes maakt vele apparaten die gebaseerd zijn op een samenwerking tussen beeld en geluid. Gieskes' Acid Machine is bijvoorbeeld een step sequencer met een draaiende plaat die afhankelijk van de frequentie van het licht dat erop schijnt patronen toont, die dan op de beat gesynchroniseerd zijn.

Hier kunnen we gemakkelijk de relatie maken naar de theorieën van Don Norman.



Don Norman stelt dat wanneer we een product kopen, we zelden tevreden zijn met alle functionaliteit ervan. Voor hem zijn er vijf manieren om hier mee om te gaan (Norman, 2004):

1. **Leg je erbij neer.**
2. **Pas het product aan.** Producten zijn echter dikwijls moeilijk te customizen. Software geeft dikwijls veel verschillende mogelijkheden tot personalisatie, maar deze mogelijkheden zijn nog vastgelegd en de dingen die je écht wil veranderen, kunnen niet veranderd worden.
3. **Aangepaste massaproductie.** Sommige items kunnen custom besteld worden bij de fabrikant. Klanten kunnen zo de producten aanpassen aan hun smaak, maar de keuze is nog steeds beperkt en is dus ver van personalisatie.
4. **Eigen ontwerp van producten.** Het ontwerp van technologisch simpele producten is goed mogelijk. Maar onze noden veranderen en de nood voor technologie maakt dat het ontwerp van de objecten die we echt nodig hebben technologisch gezien zeer moeilijk kan zijn.
5. **Verander gekochte producten.** Hier komt het concept van Circuit bending en hacking terug.

## Open Source

Overige mogelijkheden om producten te veranderen bestaan in het gebruik van

“**Open Source**”. Open source is een methode van werken die in productie en ontwikkeling **vrije toegang** geeft **tot de bronmaterialen** (de source) **van het eindproduct**. Het wordt zowel bekeken als filosofie en pragmatische methodologie.

Open Source bestaat niet alleen op het vlak van software, vele gebieden van studie en sociale en politieke meningen zijn beïnvloed door het concept van Open Source. Linus Torvalds, één van de grondleggers van Open Source en de uitvinder van Linux, een Open Source besturingssysteem, zou ooit gezegd hebben dat de toekomst is om alles te open sourcen.

Het belooft betere kwaliteit, betrouwbaarheid, flexibiliteit, lagere kost en het einde van het toe-eigenen van bepaalde productietechnieken door bedrijven. Door de constante mogelijkheid tot controle van buitenaf worden fouten of dingen die beter kunnen sneller gevonden en dit verhoogt de kwaliteit en betrouwbaarheid. Doordat de bronmaterialen vrij kunnen gebruikt worden leidt dit ook tot vooruitgang door mensen die er verder mee experimenteren.

Open Source software mag **niet verward worden met Vrije Software** (Free Software) waar Richard Stallman de voortrekker van is. Vrije Software is meer filosofisch, en pleit ervoor het gebruik van software zo vrij mogelijk te maken. Het bekritiseert Open Source door te zeggen dat politieke en sociale aspecten genegeerd worden. In de praktijk is de meeste Open Source software ook vrije software en vice versa.





Nintendo Wii

## Gesturale interfaces

Onder de term “**Gesturale interfaces**” verstaan we elke interface die **focust op het maken van gebaren om met een computer te communiceren**. Hieronder vallen onder andere **touchscreens** en “**free form**” gesturing zoals de Wiimote, waar de gesture niet beperkt blijft tot een scherm maar eender wat kan zijn in de ruimte. Deze thesis gaat via de vrijheid van een gesturaal systeem pogen om een expressievere performance toe te laten van de muzikant.

Een “**gesture**” (een gebaar) is zo volgens Dan Saffer in zijn boek “Designing Gestural Interfaces” elke fysieke beweging die opgevangen kan worden en op gereageerd kan worden door een digitaal systeem, zonder het gebruik van een traditionele “pointing device” zoals een muis of stylus (Saffer, 2008). Voorbeelden van een gesture zijn zo bijvoorbeeld het vegen met je vinger op een iPhone op naar de volgende pagina te gaan, of het tennisgebaar van Wii Sports.

Traditionele interfaces beperken mensen tot gebruik van enkel hun vingers en hun ogen en oren. Gesturale interfaces laten een groter gebruik van ons lichaam toe. Hierdoor zijn er ook meer mogelijke acties die als interactie kunnen dienen.

Het eerste apparaat dat aanraking op een precieze manier gebruikte was de Elograph en werd uitgevonden in 1971. Dit was een soort tekentablet, waar men grafieken op uittekende om ze te digitaliseren. Van hetzelfde bedrijf, Elographics, zou in 1977 de eerste touchscreen,

Accutouch, verschijnen. Sindsdien is dit apparaat helemaal ingeburgerd geraakt als invoermogelijkheid bij computer op publieke plaatsen, zoals bankautomaten en nieuwskiosken.

In 1979 heeft Richard Bolt het prototype gemaakt voor “Put-that-there”, een interface met spraak- en bewegingsherkenning. De personen die met de computer in interactie traden konden zowel via stem als via wijzen met hun vinger boten plaatsen op een kaart (Bolt, 1980). Put-that-there is daarom een van de eerste “free-form” gesturale interfaces.

In 1985 werd door Thomas Zimmerman een patent ingediend voor zijn “instrumented glove”, een handschoen die getrackt kon worden in driedimensionele ruimte door ultrasoon geluid. Ook kon de kromming van de vingers gedetecteerd worden door buigsensoren. Dit patent vormde de basis voor de VPL Dataglove, en uiteindelijk ook voor de door Mattel geproduceerde Powerglove.



Powerglove

De Powerglove was een invoerapparaat voor het Nintendo Entertainment Systeem (NES).

Het had echter goedkopere hardware aan boord dan de vrij succesvolle Dataglove, en mede hierdoor was het weinig betrouwbaar. Mattel heeft de Powerglove dan ook maar twee jaar geproduceerd. Hoewel het project een flop was, heeft de Powerglove toch de basis gevormd voor het latere gebruik van free-form gestures binnen games en meer specifiek, de Nintendo Wii.

Dan Saffer stelt dat elke gesturale interface minstens drie algemene delen heeft: een **sensor**, een **comparator** (vergelijker) en een **actuator** (uitvoerder). Deze drie delen kunnen één fysiek geheel vormen, of, vooral, verschillende componenten van eender welk gesturaal systeem, zoals een bewegingsdetector (sensor), computer (comparator) en een motor (actuator).

De sensor is meestal een elektrische of elektronische component die veranderingen in de omgeving moet detecteren. Deze veranderingen kunnen vanalles zijn, afhankelijk van welke sensor gebruikt wordt. De meest voorkomende types sensoren voor interactieve gestures zijn:

## Druk

Om te detecteren of iets ingedrukt is of op wordt gestaan. Vaak mechanisch van aard.

## Licht

Om de aanwezigheid van lichtbronnen te detecteren. Dit wordt meestal in omgevingen gebruikt, vooral bij lichtsystemen.

## Nabijheid

Om de aanwezigheid van een object in

een ruimte te detecteren. Dit kan op verschillende manieren gebeuren, van infrarood sensoren tot bewegings- en geluidssensoren.

## Akoestisch

Om de aanwezigheid van geluid te detecteren. Dit wordt meestal met kleine microfoons gedaan.

## Tilt

Om hoek en helling te meten. Tiltsensoren genereren een artificiële horizon en meten dan het verschil tot die horizon.

## Beweging

Om beweging en snelheid te detecteren. Sommige sensoren gebruiken microgolf of ultrasone pulsen die werken door de pulsen tegen het object te laten terugkaatsen. Men kan ook accelerometers gebruiken, maar die meten enkel ten opzichte van zichzelf, niet ten opzichte van de omgeving.

## Oriëntatie

Om positie en richting te detecteren. Deze worden het meest gebruikt in navigatiesystemen, maar positie binnen omgevingen wordt steeds belangrijker, en zou gemeten kunnen worden via camera's, verschillende nabijheidssensoren via driehoeksmeting, of GPS.

Het soort sensoren dat gekozen wordt bepaalt geheel de verschillende interacties die mogelijk zijn. Als het systeem bepaalde gestures niet kan detecteren, kunnen ze net zo goed niet aan het gebeuren zijn.

De comparator vergelijkt de data die binnenkomt van de sensoren met data die eerder is binnengekomen, of met het uiteindelijke doel van wat de data moet zijn, en velt dan een oordeel. Voor vele gesturale interfaces is de comparator een microprocessor die software draait die beslist wat er met de binnenkomende data gedaan wordt.

Deze beslissingen worden doorgegeven aan de actuator in de vorm van een commando. Actuators kunnen analoog of mechanisch zijn, zoals het aansteken van een licht; of ze kunnen digitaal zijn, zoals het kantelen van een iPhone de schermoriëntatie verandert. Bij mechanische systemen is de actuator dikwijls een kleine elektrische motor die een fysiek object aandrijft, zoals bij een automatische schuifdeur. Voor digitale systemen is het de software die de actuator is aangezien de software elementen op het scherm gaat veranderen.

Saffer noemt ook de voornaamste voordelen en nadelen van gesturale interfaces op:

Enkele redenen om niet voor een gesturale interface te kiezen zijn:

## Zware data input

Sommige gebruikers zijn snel vertrouwd met touchscreen keyboards, maar toch is een fysiek keyboard voor de meeste mensen beslist sneller bij het ingeven van tekst of nummers.

## Afhankelijk van het visuele

Vele gesturale interfaces gebruiken enkel visuele feedback om aan te geven dat

een actie gebeurd is (zoals een knop die ingedrukt werd). Ook werken de meeste touchscreens enkel met een scherm met geen haptische indicaties of feedback. Als de gebruikers een visuele handicap hebben kan een gesturale interface dus ongepast zijn.

## Afhankelijk van het fysieke.

Evenzo kan een gesturale interface fysiek veeleisender zijn dan een keyboard of scherm. Hoe meer fysiek een gesture wordt (zoals een slag), hoe meer kans dat sommige mensen de gesture niet kunnen uitvoeren door leeftijd, handicap, of gewoonwegomstandigheden; touchscreens gebruiken met handschoenen aan in de winter is bijvoorbeeld geen sinecure. Het omgekeerde is ook waar, hoe fijner en preciezer de beweging, hoe minder kans er is dat mensen ze kunnen uitvoeren.

## Ongepast in de context

De omgeving kan tegen een gesturale interface werken in vele situaties, bijvoorbeeld door privacyredenen of gewoonweg om de gebruikers niet te generen. Ontwerpers moeten daarom de uiteindelijke omgeving van de interface in het hoofd houden en bepalen of een gesture zal werken en wat voor gesture zal werken binnen die omgeving.

Er zijn echter ook vele redenen om wel een gesturale interface te gebruiken.

## Meer natuurlijke interacties

Mensen zijn fysieke wezens en we houden ervan om direct te interageren met

objecten. Interactieve gestures laten toe om natuurlijk te interageren met digitale objecten op een fysieke manier, zoals we doen met fysieke objecten.

### Minder omslachtige of zichtbare hardware

Bij vele gesturale systemen zijn de gebruikelijke hardware zoals muis en keyboard niet nodig. Een touchscreen of andere sensoren laten gebruikers toe om acties uit te voeren zonder deze hardware. Dit maakt gesturale interfaces geschikt voor plaatsing in ruimtes waar een traditionele computerconfiguratie onpraktisch zou zijn, zoals in winkels, museums, luchthavens en andere publieke plaatsen.

### Meer flexibiliteit

In tegenstelling tot vaste, fysieke knoppen, kan een touchscreen net zoals alle digitale schermen zich veranderen naar keuze, afhankelijk van welke functie hij moet invullen. En bij free-form gesturale interfaces zijn de mogelijkheden eindeloos.

### Meer nuance

Keyboards, muizen, trackballs, stylussen en andere inputapparaten, hoewel zeer geschikt voor vele situaties, zijn niet in staat om zoveel subtiliteit over te brengen als het menselijk lichaam. Een opgeheven wenkbrauw, een berispende vinger of gekruiste armen kunnen enorm veel informatie overbrengen als aanvulling op het besturen van een tool. Gesturale systemen hebben nog niet het hele palet aan menselijke emoties weten te detecteren

maar ze zijn goed op weg.

### Meer plezier

Een spel kan gemaakt worden waarbij de gebruiker op een knop moet drukken en het spelkarakter zwaait met een tennisracket. Maar het is plezieriger om te doen alsof je de tennisracket zelf rondzwaait en je beweging vertaald wordt naar het scherm.

Niet alle gestures zijn even goed. Sommige leveren bijwerkingen op. Eén van deze bijwerkingen is de “gorilla arm”. Dit was een bijwerking die verticaal georiënteerde touchscreens de das heeft omgedaan, ondanks een veelbelovende start in de vroege jaren ‘80.

De ontwerpers van zulke interfaces hadden niet opgemerkt dat mensen niet gemaakt zijn om hun armen voor zich uit te houden en kleine bewegingen te maken. Na enkele acties begon de arm pijnlijk, verkramp, en groter aan te voelen. De gebruiker leek op een gorilla terwijl hij de touchscreen bestuurd en voelde zich er een achteraf.

Gorilla-arm is geen probleem voor gebruik op korte termijn aangezien de interactie lang aan een stuk moet duren voordat de eerste symptomen opkomen.

De gorilla-arm is nu exemplarisch in het ontwijken van zulke vergissingen tijdens het ontwerp van gesturale interfaces.

Gesturale interfaces zijn ook sterk afhankelijk van de precisie van de sensoren. Storing (noise) kan het analyseren van een videobeeld erg moeilijk maken. Om gestures te detecteren met visuele sensoren zoals

camera's moeten solide computer-vision methodes worden toegepast, bijvoorbeeld in het detecteren van gezichtsexpressies.

### Interactieve gestures communiceren

Voordat gestures uitgevoerd kunnen worden moeten gebruikers eerst weten wat de gestures zijn en wat ze doen. Vooral in het vooruitzicht van gestures delen met andere gebruikers in mijn tweede prototype bekijk ik verschillende manieren die Saffer aanhaalt om gestures te documenteren en te communiceren.

### Geschreven instructies

Geschreven instructies zijn een voor de hand liggende keuze en zijn vooral doeltreffend bij simpele acties - “hier aanraken om te beginnen”, “klap in je handen om het licht aan te doen” - acties zijn hier normale veel voorkomende handelingen of ondubbelzinnig genoeg om uitgelegd te worden met tekst.

### Illustraties

Wanneer tekst te omslachtig is of verkeerd zou geïnterpreteerd kunnen worden kan een simpele illustratie dikwijls voldoende zijn om te communiceren hoe het systeem moet gebruikt worden. Illustraties kunnen ook klein zijn, maar niet te klein aangezien ze dikwijls beweging of details moeten overbrengen.

### Demonstratie

Een demonstratie is een bewegend beeld dat de gesture toont dat gedaan moet worden. Dit kan via animatie of via een

video met een acteur die het voordoet. Dit vereist wel dat de interface een soort scherm heeft.

### Gebruik van symbolen

Er zijn verschillende ideeën geopperd, vooral binnen de ubiquitous computing (later meer uitleg) community, om symbolen te maken die de aanwezigheid van een interactief systeem in een ruimte aangeven als dit systeem onzichtbaar is. Zoals andere methodes kunnen de symbolen gebruikt worden om aanwezigheid te communiceren en om gebruikers te vertellen hoe ze het systeem moeten gebruiken.

### Belang van metaforen

Soms is het simpel om een bepaalde gesture met iets te vergelijken. Concepten zoals “wuiven” en “in een pot roeren” zijn zo wijdverspreid dat het dikwijls simpeler is om dit soort instructies te geven dan wel elk element van de beweging nauwkeurig uit te leggen. Er moet dan wel opgepast worden dat de betekenis van de gesture niet in de war geraakt met deze metafoor. Als je “roer in de pot” beweging niets te maken heeft met iets te mixen (bijvoorbeeld beelden of geluid) kunnen gebruikers erdoor verward worden en gaan ze het zeker moeilijker hebben om de beweging te onthouden.



# Part 2: Technologisch Overzicht

## Technologieën

### Ableton Live

**Ableton Live** is in het leven geroepen als een programma dat gemakkelijk in enorm veel situaties gebruikt kan worden, en niet in het minst bij het **improviseren** en **hercombineren** van bestaande muziek On Stage. Het biedt zowel een **horizontale view** zoals de traditionele digital audio workstations (DAW) zoals Cubase, ProTools of Logic, als een **verticale view** die geheel eigen is aan het programma.

In die verticale view zitten audio of MIDI samples die gesynchroniseerd zijn met de maat. Het activeren van deze “clips” kan via muis, MIDI of toetsaanslagen. Als er in alle clips loops zitten is het mogelijk voor de muzikant om gemakkelijk live te improviseren met verschillende drum- en muziekpatronen.

Het is ook mogelijk om hele rijen tegelijk te activeren en zo een nieuw deel in het lied te laten beginnen.

Alle knoppen, sliders en draaiknoppen kunnen gekoppeld worden aan midi commando's. Via een programma als Pure Data kunnen we ook OSC signalen omzetten naar MIDI om ze zo door Ableton Live geïnterpreteerd te laten worden.

Live werkt **heel stabiel** en is gemaakt om **zo weinig mogelijk te crashen** of vast te hangen, het moet immers in een performance betrouwbaar zijn.





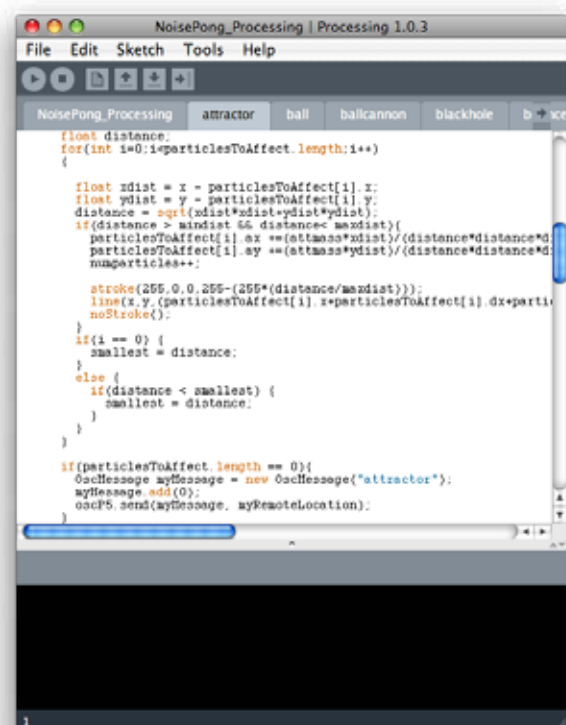
## Processing

**Processing** is door Ben Fry en Casey Reas gestart als programmeertaal die het voor artiesten gemakkelijker moet maken om **visueel te programmeren**.

Het is gebaseerd op Java en leent veel van dezelfde syntax, maar met enkele grote verschillen. Het uitvoeren gebeurt met een play knop, waardoor men direct het resultaat van zijn code kan zien. Ook het importeren van verschillende java libraries is niet meer nodig zodat programmeren heel wat minder complex wordt.

Daarbovenop heeft Processing een set functies aan boord die het visueel programmeren gemakkelijker maken. In Processing kan men bijvoorbeeld om een vierkant te tekenen op het scherm gewoon de functie `rect(x,y,breedte,hoogte);` gebruiken. Dit soort van visuele inslag maakt het direct veel simpeler voor mensen die niet vertrouwd zijn met programmeren.

De gemakkelijke samenwerking van Processing met OSC maakt dit in team met OSCulator, Pure Data of Ableton Live het ideale platform voor visualisatie.



## Pure Data

**Pure Data (Pd)** laat toe om **eigen audiovisuele programma's** te schrijven in een **modulaire omgeving**. Het is vrij uitbreidbaar met eigengescreven objecten, en biedt momenteel connectiviteit voor een breed spectrum aan technologieën. Zo werkt het goed samen met Arduino, OSC, MIDI en video (via de GEM API).

Pure data werd geschreven door Miller S. Puckette, als open source variant van Max. Puckette bedacht Max (toen het nog Patcher heette) in dienst van IRCAM midden jaren '80 en IRCAM heeft dan ook altijd het copyright behouden.

Pure Data is een complete re-design van het bestaande Max en is door Puckette in 1996 uitgebracht. Het grootste verschil was dat Pd uit zichzelf geluid kon maken, terwijl Max toen nog enkel voor Midi processing geschikt was. Uiteindelijk heeft Max dezelfde mogelijkheden gekregen door een grote upgrade, MSP (MSP staat ofwel voor Max Signal Processing of voor Miller S. Puckette), onder toezicht van Dave Zicarelli, wiens bedrijf Cycling '74 sinds 1999 de commerciële versie van Max onderhoudt.

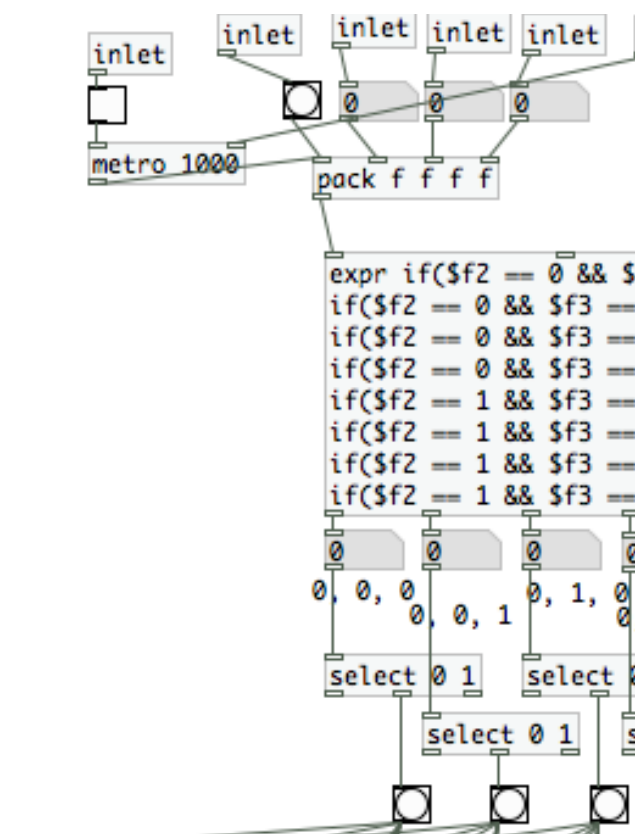
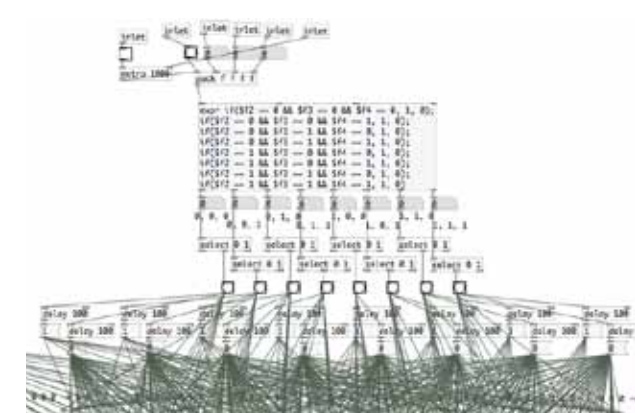
Pd bevat een aantal identieke objecten, en ook de workflow is zeer gelijkend. Sinds Max 5 werd uitgebracht in 2008 heeft Pure Data echter een merkbare achterstand op vlak van zijn user interface, maar het wint aan populariteit omdat het gratis is.

De snelle workflow op audio- en datavlak van Pd maakt het een **uitstekende tool voor rapid-prototyping**.

## OSCulator

**OSCulator** is een relatief goedkoop programma voor Mac OS X dat toelaat om enkele **randapparaten simpel op de Mac aan te sluiten** en de binnenkomende data door te sturen via een aantal protocols.

Het ondersteunt o.a. de Wii Remote, Wii Balance Board, Space Navigator en





Guitar Hero controllers voor Wii. Om snel met de Wii Remote te kunnen verbinden zit er in OSCulator een technologie die “perfect pairing” wordt genoemd. Het maakt dat OSCulator een enorm stabiele connectie onderhoudt met de Wii Remote, in tegenstelling tot programma’s als darwiinremote. Ook ondersteunt het tot 6 Wii Remotes tegelijk.

OSCulator kan zijn data schalen en herformatteren naar OSC en Midi, maar ook naar data voor Kyma sound design workstations. Dit maakt het enorm veelzijdig voor allerlei toepassingen.

OSCulator wordt verkocht aan een minimumprijs van \$ 19, adviesprijs \$ 39, en biedt voor mijn prototypes het meest veelzijdige platform mogelijk om met Wii Remotes aan de slag te gaan.

## MIDI

**MIDI** is ontworpen als een standaard om **verschillende muzikale apparaten met elkaar te verbinden**. Het is voorgesteld door Dave Smith in 1981 en werd in 1983 als specificatie werkelijkheid. In datzelfde jaar kwam de eerste synthesizer met MIDI uit.

Zesentwintig jaar later is MIDI nog steeds de **standaard** om muzikale apparaten te verbinden. Doordat vlak na de introductie van MIDI ook interfaces verschenen voor computers zoals de Apple Macintosh, Commodore en PC’s onder DOS heeft de standaard zich diep genesteld in muzieksoftware. Vrijwel elk muzikaal compositieprogramma ondersteunt MIDI.

Verschillende soorten data kunnen verzonden worden via MIDI. Eerst en vooral is er de informatie over de noten die gespeeld moeten worden. MIDI kan 128 verschillende noten (0-127) beschrijven, en geeft elke noot ook een speelsterkte (velocity) waarde mee, ook van 0 tot 127.

Naast deze informatie kan MIDI ook controleberichten sturen. Er zijn 120 verschillende kanalen voor deze controleberichten, die elk ook een waarde van 0 tot 127 kunnen aannemen. Deze berichten kunnen gebruikt worden op vele manieren, van het besturen van settings in een synthesizer tot het bepalen van volume van een mixapparaat.

De snelheid van MIDI is 31,25 kbit/s, wat meestal voldoende is maar wat wel tot hoorbare vertraging leidt als de volle capaciteit van de MIDI verbinding gebruikt wordt.

Om preciezere data en meer data te kunnen verzenden zijn er verschillende opvolgers van MIDI voorgesteld, waaronder OSC de belangrijkste is.

MIDI is momenteel echter nog steeds zeer belangrijk omdat nog maar weinig programma’s OSC ondersteunen. De populairste Digital Audio Workstations (DAW’s) hebben het nog steeds niet geïntegreerd. Eén van de meest populaire programma’s voor gebruik op het podium, Ableton Live, ondersteunt ook alleen maar MIDI en krijgt pas OSC ondersteuning in een opkomende add-on dat de functionaliteit van Cycling ’74 Max integreert.

## OSC

**Open Sound Control (OSC)** is een protocol dat ontworpen is als de opvolger van Midi. De grootste voordelen tegenover MIDI zijn snelheid en hoeveelheid van data, connectiviteit via internet, en het overzichtelijkere binden van gegevens aan betekenis door elk data-element met een symbolische structuur mee te sturen in plaats van 8-bit adressen.

Elk OSC bericht kan via eender welke netwerkverbinding worden doorgestuurd. Ook via Wifi. OSC berichten worden zo snel mogelijk doorgestuurd naar de ontvanger en kunnen eventueel na aankomst een vertraging mee gespecificeerd krijgen om pas op een later ogenblik in actie te treden.

De soort data en variabelen die door OSC kunnen doorgestuurd worden zijn divers en gaan van 8-bit getallen tot audio data of afbeeldingen. De meeste toepassingen van OSC ondersteunen echter alleen variabelen met lettertekens en 32-bit drijvendekommagetallen.

Een OSC bericht ziet er bijvoorbeeld zo uit:

“/wii/1/accel/pty/2 0.513”

De lettertekens aan het begin van het bericht vertellen ons dat dit bericht over een wiimote gaat, deze wiimote nummer 1 is, dat het over de accelerometer van de wiimote gaat, en meer bepaald over de Pitch Roll en Yaw (PRY) parameters. De 2 op het einde van de lettertekens vertelt ons dat het over de derde van de drie (te tellen vanaf 0) gaat, Yaw.



Achter deze identificerende tekst bevindt zich een nummer dat de waarde van de parameter voorstelt, in dit geval 0,513.

Het aantal hardwareproducten die OSC ondersteunen is nog klein, maar voor computer naar computercommunicatie is het goed uitgewerkt. Het bekendste hardwareproduct dat OSC gebruikt (naast MIDI) is de Jazzmutant Lemur.

Het aantal softwareproducten groeit gestaag, maar de grote spelers op de markt blijven momenteel nog uit. De bekendste programma's die gebruik kunnen maken van OSC zijn Native Instruments Reaktor en Traktor DJ Studio.

Voor elke grote programmeertaal bestaan er OSC implementaties.

## Bluetooth

**Bluetooth** is een draadloos protocol ontworpen voor mobiele draadloze netwerken. Het wordt veelvuldig gebruikt in mobiele telefoons, draadloze headsets, muizen en game controllers. Onder meer de Nintendo Wii Remote en de Playstation 3 SIXAXIS controller werken via Bluetooth.

Afhankelijk van de versie kan Bluetooth snelheden van 1 tot 3 Mbit/s bereiken. De meeste apparaten zijn van klasse twee, wat betekent dat ze tot op tien meter van elkaar kunnen gebruikt worden.

Er is een aparte Bluetooth chip in een computernodig om te kunnen communiceren met apparaten. Bij het verbinden maken identificeren Bluetooth apparaten zich met naam, klasse, functionaliteiten, en

technische informatie.

Bluetooth is een betrouwbaar protocol geworden, dat een stabiele connectie levert, zeker voor de relatief lage hoeveelheid informatie dat een wiimote doorstuurt.



# Part 3: Kwalitatief onderzoek

## Inleiding

Als kwalitatief onderzoek geef ik eerst een korte inleiding tot mijn methodologieën en daarna maak ik een **analyse** van de **muzikale interfaces** die er nu voor computers gemaakt worden, hun verschillende eigenschappen en zal ik bepaalde **nadelen of voordelen** die aan deze eigenschappen verbonden zijn bespreken. Zo kan ik mij een goed beeld vormen van tekortkomingen die opgelost zouden kunnen worden met een gesturale interface.

Vervolgens bekijk ik een aantal **artiesten** die momenteel **gebruik maken van gesturale interfaces** in de context met muziek, welke technieken ze gebruiken, hoe ze de verschillende gebaren omzetten in muziek, en welke de eigenschappen zijn van hun verschillende interfaces teneinde inspiratie op te doen voor mijn eigen interfaces.

Daarna volgt een **uitgebreide analyse van een Wii Remote** voor het gebruik als inputinterface voor menselijke beweging. Ik leg ook uit waarom ik voor Wii Remotes heb gekozen als prototypemateriaal en bespreek kort Johnny Chung Lee die veelvuldig de Wii Remote aanpast en omvormt zodat het andere doelen dient.

Tot slot bespreek ik enkele **projecten** die ik tijdens **dit masterjaar** heb gedaan met **gesturale interfaces** en hoe deze belangrijk zijn geweest voor mijn prototypes in deze thesis.

## Methodologie

### Research through design & iterative design

Voor mijn prototypes en onderzoek gebruik ik een methode die “**research through design**” heet.

Peter Lunenfeld beschrijft in het door Brenda Laurel geredacteerd boek Design Research hoe er drie belangrijke modi van design research zijn: research **into** design, research **through** design en research **for** design (Lunenfeld, Design Research, 2003). Research into design omvat de traditionele historische en esthetische studies over kunst en design. **Research through design is projectgebaseerd en omvat het researchen van materialen en ontwikkeling.** Research for design heeft tot doel om objecten en systemen te maken die de resultaten van het onderzoek weer te geven en de waarde van het onderzoek te bewijzen.

Anne Burdick stelt dat de projecten die men doet tijdens research through design dienen als experimenten om ideeën te toetsen, hypotheses te testen en nieuwe vragen te stellen (Burdick, Design Research, 2003). Kritische reflectie is nodig, om vragen te stellen en om conclusies te kunnen trekken. De waarde van dit soort formele onderzoeken is niet enkel het palet aan mogelijkheden te verbreden waar de ontwerper toegang tot heeft, maar ook om de disciplines waaruit design bestaat te analyseren en een begrip te ontwikkelen van het werk dat ontwerpers doen.

## Rapid & Paper Prototyping

De meest simpele vorm van prototype is het “**paper prototyping**” (papieren prototypes). Sommigen zouden kunnen argumenteren dat omdat er geen digitale component aan verbonden is dat ze niet echte prototypes zijn, maar ze zijn wel zeer geschikt om het **algemene concept** en de “**flow**” van een product te **testen**, zowel met het eigen team als met externen.

Indien goed toegepast, kunnen papieren prototypes **grote usability fouten ontdekken voor ze te diep geïmplementeerd zijn**. Maar dat is niet hun enige functie. Ze kunnen vroeg in het ontwerpproces een indruk geven van hoe het uiteindelijk product gaat werken, of het prettig zal zijn om er mee te werken, of het duidelijk genoeg is.

**Rapid prototyping** is het gebruiken van **simpele elektronische tools** om werkende modellen te creëren van het uiteindelijke artefact. Tools om snel te prototypen zijn er te over; voor schermgebaseerde interfaces kan dit via Flash of Director, voor interactieve producten met microcontrollers zoals Arduino (en alle uitbreidingen ervan) of nog simpeler, men kan speelgoed zoals Lego Mindstorms gebruiken en het omvormen tot zijn noden.

Als men vroeg in het ontwikkelingsproces prototypes maakt, kan men grote ontwerp- en redeneerfouten al vroeg in het proces filteren. Het concept van iteratief design garandeert bovendien dat elk opeenvolgend prototype een zekere vooruitgang heeft geboekt tegenover het vorige.

## Muzikale interfaces voor computers nu – analyse

Het is evident dat muzikanten niet enkel in staat willen zijn om hun laptop met keyboard en muis te besturen. Daarom worden er steeds meer nieuwe interfaces bedacht om te communiceren met allerlei apparatuur, zowel **softwarematig** als **hardwarematig**. Een overzicht van enkele initiatieven van de voorbije jaren, en wat ze te bieden hebben op vlak van Human Computer Interaction.

### Traditionele hardware muziekinterfaces

De meeste muziekinterfaces voor computers zijn met **dezelfde elementen** opgebouwd als die waarmee **traditionele instrumenten** en **elektrische instrumenten** werken. Een interface om piano te spelen op een computer gaat een klavier hebben. Een interface om een geluidsmixer te besturen gaat faders en draaiknoppen hebben. Variaties op interfaces zijn meestal combinaties van deze traditionele elementen.

De elementen zijn ook terug te vinden op analoge synthesizers en zijn waarschijnlijk daar van overgenomen toen de eerste computerinterfaces verschenen.

### Draaiknoppen

De meeste MIDI draaiknoppen zijn eindeloos draaiend, wat wil zeggen dat ze geen begin



of einde hebben. Dit is gedaan om vlot te kunnen wisselen van functie, wanneer de knop een volumeparameter bedient, en er daarna wordt gewisseld naar een andere functie zoals een effectenparameter, kan de waarde die de knop vertegenwoordigt natuurlijk niet dezelfde blijven. Met andere woorden, als de volumeknop origineel op 50 stond, mag deze waarde niet automatisch over worden gezet op de effectenparameter. Als de draaiknop eindeloos kan blijven draaien, kan deze op eender welke positie eender welke waarde vertegenwoordigen. De gebruiker weet gewoon dat één klik naar rechts een vermeerdering van 1 in de huidige parameter vertegenwoordigt.

### Faders

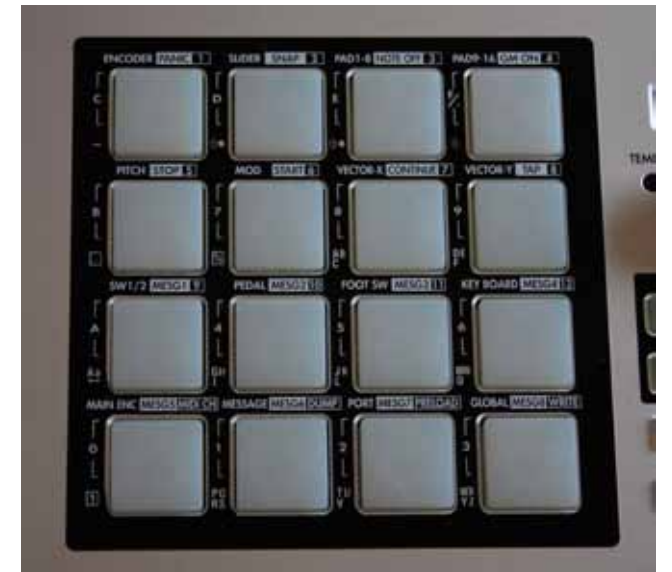


Faders worden vooral gebruikt om het volume te regelen van bepaalde sporen. Ze kunnen in feite dezelfde functies verrichten als draaiknoppen, maar worden door hun lineaire aard vooral voor het mixen van volume gebruikt.

Faders kunnen moeilijker van parameter wisselen, omdat ze sowieso exact aangeven op welke positie ze staan. Daarom bestaan er ook gemotoriseerde

faders (motorfaders), die snel automatisch naar een andere waarde kunnen springen.

### Pads



Pads (of drumpads) zijn grote knoppen van rubber, gevoelig voor aanslagsterkte. Ze zijn gepopulariseerd door de MPC midi controller van Akai.

De pads zijn beter geschikt voor het invoeren van ritmes dan een klavier door de analogie met percussie.

De haptische feedback die een performer krijgt van de pads is afhankelijk van de soort van pads. Sommige pads zijn vrij hard, anderen geven vering terug, een beetje zoals een drumvel. De positie van de drumpads kan je wel gewoon worden en het instrument kan dus blindelings bespeeld worden net zoals je een gitaar of piano blindelings zou bespelen.

### Drukknoppen

Drukknoppen zijn veelzijdig en kunnen in vele situaties toegepast worden. Ze gaan van harde plastieke knoppen tot zachte rubberen, met dat verschil met pads dat





ze niet gevoelig zijn voor aanslagsterkte. Ze kunnen dus enkel een digitaal signaal “aan” of “uit” doorsturen.

### Pitch & Modulation wheel



Deze elementen zijn standaard aanwezig op elke controller met een klavier. Het zijn allebei cilinders die loodrecht in de controller staan. Het pitchwheel is met veren gecentreerd in het midden, en het bewegen van dit wiel naar boven of naar beneden buigt in de meeste instrumenten de toon om naar boven of beneden.

Wat het modulationwheel doet hangt van instrument tot instrument af, soms bedient het een low-pass filter (waarbij de hoge tonen worden weggefilterd), soms bedient het een vibrato (waarbij de toonhoogte van de noot zachtjes afwisselend wordt afgebogen naar boven en onder, waardoor de noot ‘trilt’).

### Klavier



Een klavier zit in muziekinterfaces om gemakkelijk muzieknoten te kunnen spelen met detectie van sterkte van aanslag.

Deze toetsen kunnen in grootte verschillen en hoeven niet persé op dezelfde schaal te zijn als de toetsen van een piano. Ook de haptische feedback van een piano is maar soms aanwezig. De toetsen kunnen zijn zoals bij een synthesizer, en amper weerstand bieden, of ze kunnen “weighted” keys zijn, toetsen met een zekere weerstand en gewicht, en hamermechanieken, om zoveel mogelijk het gevoel van een piano na te bootsen.

Deze klavieren worden niet enkel gebruikt door personen die goed piano kunnen spelen. Iedereen met een beetje gevoel voor ritme in hun vingers (of de mogelijkheid om het achteraf goed te corrigeren) kan dit als tool gebruiken om bepaalde melodieën in te spelen.

### XY controller



Op sommige MIDI controllers zit een tweedimensionale controller in de vorm van een touchpad zoals op een laptop of een kleine joystick. Dit geeft de mogelijkheid tot manipulatie van twee parameters tegelijk.

Deze elementen zijn de meest bekende interface-elementen op een MIDI of computercontroller. Ze hebben elk hun eigen typische gebruiken. Over het algemeen heb je veel controle over je parameters die je wil controleren, en krijg je ook relatief veel tactiele en haptische feedback. Ruimte voor lichaamsexpressie is er echter weinig, hoewel sommige performers veel moeite doen om zo expressief mogelijk aan een draaiknop te draaien. De vaste aard van huidige software- en MIDI controllers vereist van de performer dat hij achter zijn bureau met technologie blijft staan en zich lichamelijk weinig expressief kan uiten.

### Jazzmutant Lemur



De Lemur Input Device van het Franse Jazzmutant is een interface voor verschillende audiovisuele toepassingen gebruik makend van **OSC** of **MIDI**. Het is een **multi-touch** scherm waar zich controleobjecten bevinden die door de gebruiker zelf gespecificeerd kunnen worden in opstelling en grootte. Het wordt **bestuurd met de vingers**.

In hun multi-touch opstelling zijn ze **pioniers** geweest op hardwarevlak. In oktober 2003 ontwikkelden ze hun eerste prototype en dit is dan via IRCAM een jaar later, in oktober 2004 gepresenteerd. In 2005 werd de Lemur beschikbaar voor aankoop. Hierdoor predateren ze op de markt andere populaire multi-touch toepassingen zoals de iPhone (januari 2007) en is het ongeveer tegelijk bekend geworden met de Reactable (in 2005 aan het publiek getoond).

De Jazzmutant Lemur is **erg duur** (momenteel rond de 2000 euro) en daarom **niet toegankelijk** voor vele performers.

Eén van de nadelen van een multi-touch oppervlak met controleobjecten als faders en draaiknoppen is het **gebrek**

**aan tactiele feedback** en ruimer gezien zelfs een gebrek aan materiële feedback. Performers moeten ten alle tijden hun multi-touch oppervlak zien terwijl ze het bespelen.

Hun sterkte ligt erin dat ze in functionaliteit van het multi-touch scherm niet beperkt worden. Een vrijwel oneindig aantal opstellingen en schermgebaseerde interfaces zijn mogelijk. Door ondersteuning van zowel OSC als MIDI kan vrijwel elke applicatie voor multimediale projecten bestuurd worden, niet alleen voor audio maar ook voor VJ applicaties zoals Modul8, VDMX en Resolume.

Toch is de Lemur vooral bedoeld voor live performance. De andere multi-touch interface die Jazzmutant uitbrengt is de Dexter, waar alle elementen van de interface vastliggen en geoptimaliseerd zijn voor samenwerking met de Logic Pro, Cubase, Nuendo en Sonar DAW's (Digital Audio Workstations).

## Reactable



Reactable Systems is vanuit ruwweg dezelfde premisse opgestart als de voorgaande interfaces; om de manier

waarop we interageren met computers te herdefiniëren. Met dit doel hebben ze de Reactable gebouwd, een muziekinstrument gebaseerd op een computer, met een **sterk visuele representatie** van het geluid dat het produceert.

De interactie met het instrument bestaat uit het **plaatsen van blokjes en slepen met vingers** op een ronde projectietafel. Onder de blokjes bevinden zich speciale tekens die herkend worden door een camera onder de tafel. Hierdoor kan de computer een beeld terug projecteren met objecten waarvan de positie overeenkomt met de blokjes. Ook herkent de tafel vingers op het tafelvlak, waardoor het ook een multi-touch interface is.

Een groot verschil met de Jazzmutant Lemur is dat hier **zowel vingers als fysieke objecten** kunnen gebruikt worden als inputmethode. Dit geeft tactiele feedback aan de performer en laat toe om te interageren met het instrument zonder te hoeven vertrouwen op visuele feedback.

Een ander groot verschil is dat de Reactable rond is, en **uitnodigt om te gebruiken in groep**. Hierdoor kan de interface publiek worden gemaakt tijdens optredens, in tegenstelling tot de Lemur die meestal zoals een traditionele interface enkel door de performer wordt bekeken.

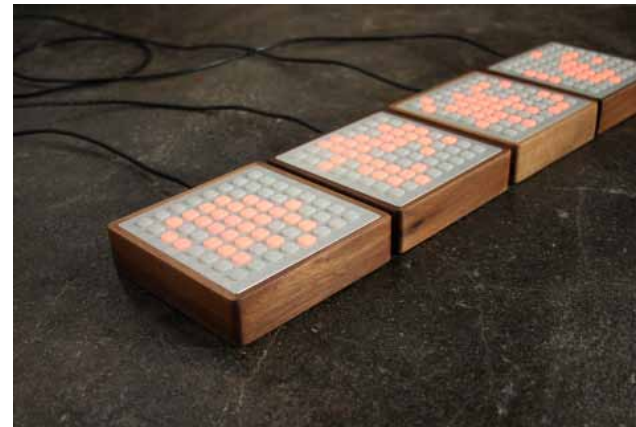
Onder andere Björk heeft een Reactable mee op tour genomen en tijdens het spelen op de Reactable werd elk aspect van dat spelen gefilmd en getoond op de grote LED schermen. De Reactable leent zich dus goed als een **visueel performant instrument**.

Reactable Systems heeft hun detectiesoftware voor de Reactable vrij downloadbaar gemaakt onder de naam ReactIVision. Hierdoor kan iedereen met de juiste middelen zelf een tafel bouwen voor gebruik met fysieke blokken en multi-touch. De software van de synthesizer die op Reactable draait is betalend.

ReactIVision stuurt OSC signalen uit waardoor iedereen met kennis van een programmeertaal die OSC ondersteunt het kan gebruiken.

Een nadeel van de Reactable en de ReactIVision techniek is dat er **wel tactiele feedback** is, **maar weinig haptische feedback**. Een blok kan eindeloos rondgedraaid worden zonder haptische notie van frictie of einde.

## Monome



De Monome is een initiatief van Brian Crabtree en Kelli Cain om een **open source interface** te maken met als doel de relatie tussen mensen en interfaces te herdefiniëren. Door een systeem te maken dat beide flexibel, eindeloos aanpasbaar en toch simpel is stellen ze gebruikers in staat om **meer te experimenteren**. Ze zien flexibiliteit niet als eigenschap, maar als

basis voor hun gehele project. Ook levensduur en ecologie staat centraal door de keuze voor kwalitatieve milieuvriendelijke materialen.

De minimalistische interface heeft een 8 x 8, 8 x 16 of 16 x 16 rooster van knoppen, in totaal goed voor 64, 128 en 256 knoppen. Achter elke knop zit een LED die de knop in kwestie kan oplichten.

De software die geleverd wordt bij een Monome is begonnen met een viertal zelfgeschreven programma's en is nu door de open source aard van de interface **sterk uitgebreid met vele community-made tools**. Ook stelt het toestel gebruikers in staat om zelf te programmeren en communiceren via OSC.

De roosteropstelling van knoppen maakt het ook een handige tool in samenwerking met de verticale interface van Ableton Live, die kan gebruikt worden voor live improvisatie met samples of MIDI. De activatie van samples of MIDI lijkt sterk op de manier waarop de APC-40 controller van Akai dit doet.

Doordat de Monome open source is, kan iedereen met kennis van zaken er zelf een maken. DIY projecten zijn zo bijvoorbeeld de Arduinome en de Bonome, elk veel goedkoper dan een "echte" Monome.

De interface **vertrouwt op visuele feedback**, zeker bij de grotere modellen, waar je enkel via tactiele feedback al snel verdwaald kan geraken op de knoppen.



## Conclusie

In dit overzicht van muzikale interfaces hebben we kort gezien wat de meest voorkomende vormen interfaces zijn en enkele voorbeelden van nieuwe interfaces die gesturaal, tactiel, tangible, en open source zijn. De manier waarop deze interfaces werken en waarop ze verspreid worden, is een inspiratiebron geweest voor mijn prototypes. De veelzijdigheid van mijn prototype 2 is rechtstreeks geïnspireerd door de oneindige mogelijkheden van de Jazzmutant Lemur en Momome, terwijl ik er het visueel performance-aspect aan probeer te geven, gelijkwaardig aan dat van de Percussa Audiocubes.



## Case studies: Toepassingen van gesturale interfaces binnen muziek.

### Case: Eboman



Een initiatief uit Nederland dat qua opzet sterk op deze thesis lijkt is **Eboman**, een performer die sinds 2001 bezig is met het controleren van zijn muziek en VJ-sets via een **SensorSuit**. Dit pak bevat verschillende buigsensoren die o.a. de kromming van zijn rug en de strekking van de vingers meten.

Hij gebruikt zijn SensorSuit op een expressieve manier om bepaalde **samples te triggeren**. De achtergrond wordt verzorgd door een dj en bevat meestal een minimalistische beat terwijl Eboman zelf ritmisch **video en audio activeert met zijn vingers**. Het buigen van zijn rug heeft een vervormend effect op de video en audio.

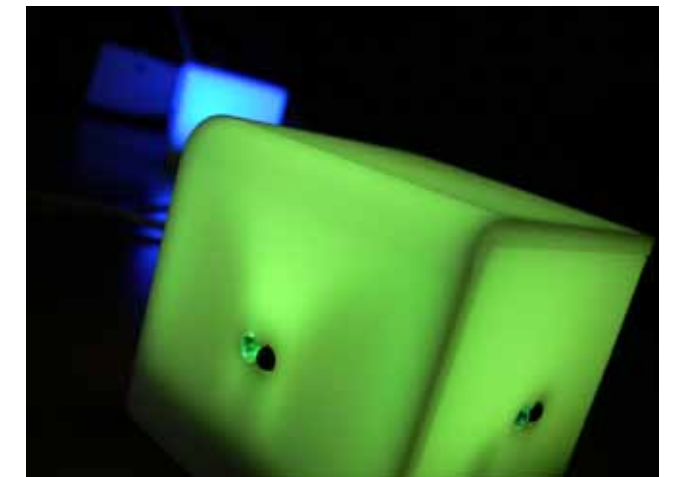
Eboman doet verschillende performances

hiermee, telkens goed ontvangen bij het publiek. Hij heeft het instrument origineel ontworpen om **meer vrijheid te hebben op het podium** en kan er twee laptops tegelijk mee aansturen.

De interface is een **free-form gesturale interface**. De parameters die uitgelezen worden van de verschillende buigsensoren worden soms lineair gemapt op effecten van de muziek, andere buigsensoren zijn meer “triggers” in dat ze een digitaal 1 of 0 signaal sturen naar Eboman’s zelfgemaakte software.

De interface kan dus vertrouwen op **auditieve feedback** en **feedback van het eigen lichaam** via **proprioceptie**.

### Case: Percussa Audiocubes



Percussa is een Belgisch initiatief van Bert Schiettecatte, een elektronisch muzikant die o.a. aan het CCRMA in Stanford een Master of Arts heeft behaald. Zijn droom was om de toekomst van elektronische muziekinstrumenten mee te bepalen en heeft met de Audiocubes zeker een invloed gehad op de manier waarop computers muzikaal kunnen bestuurd worden.



De Audiocubes zijn **kubussen van een mat plastic** waarin **LED's** en enkele **sensoren** in zitten. Ze kunnen **draadloos met elkaar communiceren**, en afstand meten van de handen van de gebruiker. Dit laat toe ze op een theremin-achtige manier te bespelen.

De positie van Audiocubes tegenover elkaar en de afstand die gemeten wordt door de sensoren bespeelt verschillende parameters in applicaties die bestuurd kunnen worden via MIDI.

De Audiocubes kunnen ook zelf geluid produceren, via een ¼ inch jack uitgang.

De cubes zijn **tangible**, materieel, dus kunnen ze in feite op dezelfde manier worden bespeeld als een reactable, **minus de visuele feedback** van de projectie. Het is ook in weze een gesturaal systeem door de afstandssensoren, dezelfde technologie die Godfried Willem Raes gebruikt in zijn onzichtbaar instrument.

### Case: PLOrk & SLOrk

Aan Stanford University is er een afdeling die zich enkel bezig houdt met het produceren van nieuwe instrumenten met computers, het Center for Computer Research in Music and Acoustics (CCRMA).

Eén van de spin-offs van het CCRMA is het **Stanford Laptop Orchestra (SLOrk)** in navolging van het **Princeton Laptop Orchestra (PLOrk)**, beide onder leiding van Ge Wang.

De premisse van het PLOrk is om **de laptop zelf te gebruiken als expressief**

**muziekinstrument**. Elke laptopmuzikant werkt op een macbook die aangesloten is aan een **omnidirectionele speaker** zodat het geluid dat hij produceert naar alle kanten resoneert, net als bij een akoestisch instrument.

Ze pogen elk element van de standaard macbook te gebruiken. Het keyboard wordt gebruikt om noten te spelen, terwijl het trackpad fungeert als “strijkstok” voor de noten.

Wat ze als extra expressief element gebruiken is de **Sudden Motion Sensor (SMS)** die in elke macbook zit. Deze sensor is in weze een **accelerometer** zoals in de wiimote, en zorgt ervoor dat de harde schijf van de macbook zijn schrijfnaalden “parkeert” zodra het een val detecteert zodat de data op de harde schijf niet beschadigd wordt. Dit maakt dat de macbook als gesturale interface kan gebruikt worden als geheel.

Aangezien de sensor de helling van de macbook ruwweg kan bepalen in dezelfde manier waarop de Wii Remote dit doet, kan die helling gebruikt worden als extra factor om het geluid aan te passen op variërende wijze, bijvoorbeeld het moduleren van een low-pass filter.

Hun muziek programmeren ze op voorhand of live met Chuck en Max/MSP. Chuck is het product van de doctoraatsthesis van Ge Wang en is in principe laptoporkesten op het lijf geschreven.

Ondertussen zijn er naast de twee originele laptop orchestras verschillende nieuwe

opgestart, waaronder het Oslo Laptop Orchestra (OLO) die een dure MacBook inruilen voor netbooks op Windows.

### Case: Godfried Willem Raes' onzichtbaar muziekinstrument

Godfried Willem Raes is een Belgisch componist, performer, en instrumentbouwer. Hij is de stichter van de **Logos Foundation**, een stichting voor de promotie van “nieuwe muziek”. Hij heeft een doctoraat behaald aan de rijksuniversiteit Gent met zijn verhandeling **“Een Onzichtbaar Muziekinstrument”** in 1993.

In dit doctoraat geeft Raes een zeer uitgebreid technologisch overzicht van manieren om een gesturaal muziekinstrument te maken rond die tijd, **met als doel een “onzichtbaar” muziekinstrument te maken dat via menselijke beweging kan bespeeld worden**. Enkele besproken technologieën zijn sonar, radar, spierspanningstranducers en beeldanalyse.

In de jaren na zijn doctoraat heeft Raes nog vele andere methoden besproken, waaronder accelerometers. Raes heeft deze gebruikt in een gesturale controller voor robotische instrumenten, “Godfried’s Ei”.

Hoe de binnenkomende bewegingsdata gemapt wordt tot muziek verschilt van performancetotperformance. De technische capaciteiten van zijn instrumenten laten toe dat ze voor bewegingsherkenning worden gebruikt.

# Wii remotes als input van menselijke beweging

Korte geschiedenis – waarom Wii Remotes?



De **Wii Remote**, of “**wiimote**” (vanaf hier worden deze begrippen door elkaar gebruikt) is in 2006 op de markt verschenen als de revolutionaire nieuwe controller van de Nintendo Wii. Al gauw bleek dat Nintendo grafische kracht had opgeofferd voor meer onderzoek naar nieuwe methodes van interactie en samen met slimme marketing zorgde dit ervoor dat hun doelpubliek veel groter werd dan traditionele gamers. Momenteel hebben ze **meer dan 50 miljoen Wii-consoles verkocht**.

Aangezien er minstens één wiimote komt geleverd bij elke Wii, betekent dit dat er momenteel meer dan 50 miljoen wiimotes in omloop zijn.

Al gauw had de mod/hackercommunity door dat de wiimotes een handig apparaat zou

zijn voor verschillende vormen van input. Het feit dat een wiimote zijn connectiviteit met de Wii onderhoudt via het standaard Bluetooth-protocol gaf modders ook directe toegang tot de data uit de wiimote. Deze Bluetooth technologie zit tegenwoordig in vele laptops en desktops ingewerkt

Via forums op het internet hebben deze hackers hun kennis over de wiimote verspreid en de wiimote gedetailleerd ontleed. **Bijna elk deel van de wiimote is nu bruikbaar voor eigen projecten, wat de wiimote quasi open-source maakt.**

De wiimote is ontworpen met het oog op een groot doelpubliek. Dit blijkt uit de **minimalistische stijl zonder decoratie** die het kan doen neigen naar een bepaald doelpubliek toe. Het ontwerp van de wiimote nodigt uit tot beweging en is **esthetisch pleasurable door zijn strakke minimalisme**.

Origineel kwam de wiimote met een beschermingsbandje dat soms brak onder de grote kracht die erop werd uitgeoefend als de wiimote uit de handen van de gebruikers slipte. Dit was rechtstreeks de aanleiding voor de website wiihaveaproblem.com, waar filmpjes en foto's werden gepost van gebroken tv's als gevolg van een wiimote-inslag. Sindsdien levert Nintendo stevigere bandjes.

Ook, om andere ongelukken te vermijden of te verzachten, worden wiimotes geleverd met een rubberen hoes. Deze hoes is dikker rond de voorkant van de wiimote, om de interne camera te beschermen en al wat

die wiimote zou kunnen raken in zijn vlucht.

Wanneer we de wiimote vasthouden zonder hoes, merken we weinig van zijn gewicht. Het is **non-intrusive**, we merken weinig van de aanwezigheid van de wiimote en het **wordt een natuurlijke extensie van de arm**, die locatiegegevens kan doorspelen aan de computer. Dit heeft het voordeel dat de **gebruiker niet wordt afgeleid door het gewicht of de manier waarop de wiimote moet vastgehouden worden, maar ook het nadeel dat tactiele feedback ontbreekt**, je hebt ook geen gevoel van kracht, want je slaat met je arm in het ijle. Wanneer de rubberen hoes errond zit, wordt het zwaartepunt meer naar voor gelegd, en krijg je meer middelpuntvliedende kracht tijdens het bewegen met de wiimote. Dit zorgt ervoor dat het gevoel van kracht en snelheid groter wordt, cfr. Het gebruik van een tennisracket.

Omdat we hier met een **free-form gesturale interface** werken vertrouwt het systeem op **visuele feedback** van de TV. Acties zijn direct merkbaar door de snelle respons van de games. Ook is er in zekere zin **haptische feedback mogelijk door de trilmotor in de wiimote**. Deze vorm van feedback is in de meeste controllers voor gameconsoles beschikbaar en kan de controller doen trillen in de hand bij bepaalde acties. Als er met de wiimote aanwijzer bijvoorbeeld over een knop wordt gegaan, kan er een korte trilling gegeven worden, zodat de visuele knop ook een haptische dimensie krijgt.

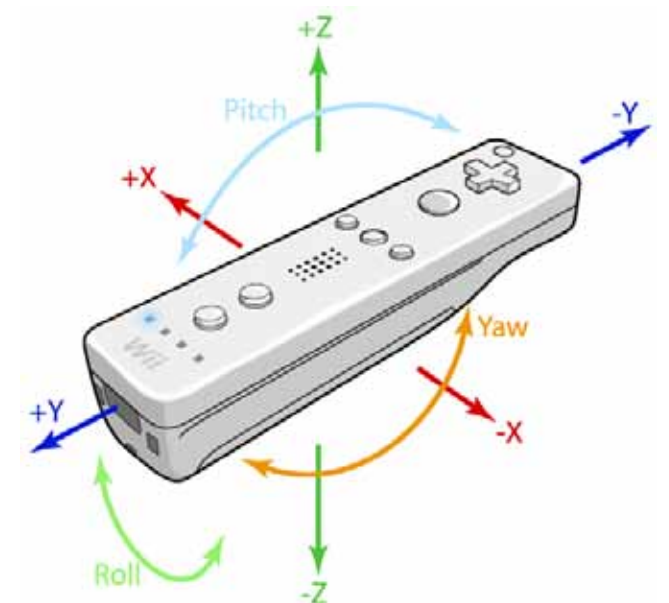
Een wiimote is te koop voor ongeveer 40

euro in verschillende winkels, wat goedkoop is voor de technologie die erin zit. Dit is nog een grotere motivatie om er zelf mee aan de slag te gaan.

## Werking van een Wii Remote

De wiimote wordt van stroom voorzien door twee AA batterijen, dus men heeft geen Wii nodig om hen op te laden, in tegenstelling tot bijvoorbeeld de controllers van een Playstation 3.

Elke wiimote heeft 7 functieknoppen plus een 8-dimensionale D-pad. In het toestel zelf zit een 3-dimensionale accelerometer, een infraroodcamera die op 100hz tot 4 IR LEDs kan tracken, een vibratie-unit, 4 status LEDs en een luidspreker. De accelerometer meet positieve en negatieve kracht in x,y,z richtingen, inclusief de zwaartekracht. **De zwaartekracht zorgt er ook voor dat het ruwweg mogelijk is om te schatten in welke hoek een wiimote zich bevindt tegenover de grond.**



Deze afbeelding laat zien hoe de verschillende krachten gemeten worden. Dit

is echter geen goede basis om betrouwbaar Pitch, Roll en Yaw te berekenen. De “Yaw” kan bijvoorbeeld niet berekend worden als de wiimote plat in het x-y vlak ligt, omdat Yaw net die assen nodig heeft om berekend te worden, en ze hier evenwijdig liggen aan de inwerkende kracht, -Z. Meer hierover bij Prototype: Wiimote-analyse.

## Wii MotionPlus



In juni 2008 kondigde Nintendo een uitbreiding voor de Wii Remote aan die de detectie van bewegingen en rotatie van de hand preciezer moest maken; de “Wii MotionPlus”. In juni 2009 is dit product op de markt gebracht als klein blokje dat onderaan in de Wii Remote geprikt moet worden. Het apparaat bevat in tegenstelling tot de normale accelerometers van de standaard Wii Remote een **gyroscop**.

Dit stelde het team van Nintendo voor vele ontwerpproblemen. Een gyroscop is immers **niet ongevoelig voor verschillende omgevingsparameters**. Door temperatuurverschillen en luchtvochtigheid kan de kalibratie van de gyroscop veranderen en zo een onprecies resultaat opleveren. Daarom moet de MotionPlus elke keer bij het opstarten gekalibreerd worden. Ook werkt het samen met de reeds aanwezige accelerometers om een zo goed mogelijke meting van de menselijke beweging te bekomen.

Hoewel het systeem dus zekere problemen heeft gekend tijdens de ontwikkeling zijn de voordelen ervan groot. Toen de Wii origineel uitkwam stond iedereen verstoemd van een systeem dat je bewegingen nabootste. De originaliteit van de Wii overwon zijn tekortkomingen, namelijk dat de accelerometers onprecies waren en moesten afgaan op eerder ingespeelde en vastgelegde bewegingen die de ontwikkelaars van spellen gemaakt hadden. Neem het spel Zelda als voorbeeld. De speler kan met zijn Wii Remote een slagbeweging maken waardoor het hoofdkarakter, Zelda, met zijn zwaard slaat. De gedane beweging lijkt echter niet op de beweging die het hoofdkarakter doet. Dat komt omdat de verwerking van de gesture van de speler zo verloopt:

**Speler doet gebaar → Wordt herkend als “zwaardslag” → spel geeft een commando aan het karakter om een zwaardslag te doen → karakter doet voorgeprogrammeerde beweging.**

De camera aan de voorkant van de Wii

Remote zorgt voor een veel preciezer respons, maar vereist dat de Wii Remote in de richting van het scherm wordt gehouden. Dit wordt dan ook gebruikt in de context van first-person shooters.

Met de gyroscop in de MotionPlus echter, is **perfect de rotatie van de Wii Remote in drie dimensies te meten**, en dit kan leiden tot een veel hechtere connectie tussen spel en realiteit. De verwerking van deze bewegingen loopt zo:

**Speler doet gebaar → rotatie van Wii Remote wordt letterlijk vertaald naar de rotatie van het zwaard van het karakter → karakter zwaait exact op dezelfde manier met zijn zwaard als de speler heeft gedaan met zijn Wii Remote.**

Deze directe connectie geeft aan de speler veel meer het gevoel dat hij de controle heeft over het spel. Ook kan hij zo precies kiezen hoe hij zijn zwaardslag wil doen. Het spel moet dan gewoon berekenen waar dat zwaard de vijanden raakt en met welke snelheid, en hierop de geleden schade van de vijanden berekenen.

De speler leeft zich vervolgens veel meer in in de actie van het spel. Als het spelkarakter tientallen vijanden van zich af moet slaan, zal de speler in plaats van de hele tijd het gebaar voor “zwaardslag” te doen zich met hand en tand moeten verdedigen.

## Case: Johnny Chung Lee

Johnny Chung Lee is zeer waarschijnlijk de bekendste Wii Remote gebruiker. Zijn wiimote projecten hebben op YouTube al

gezamenlijk 11 miljoen views opgeleverd.

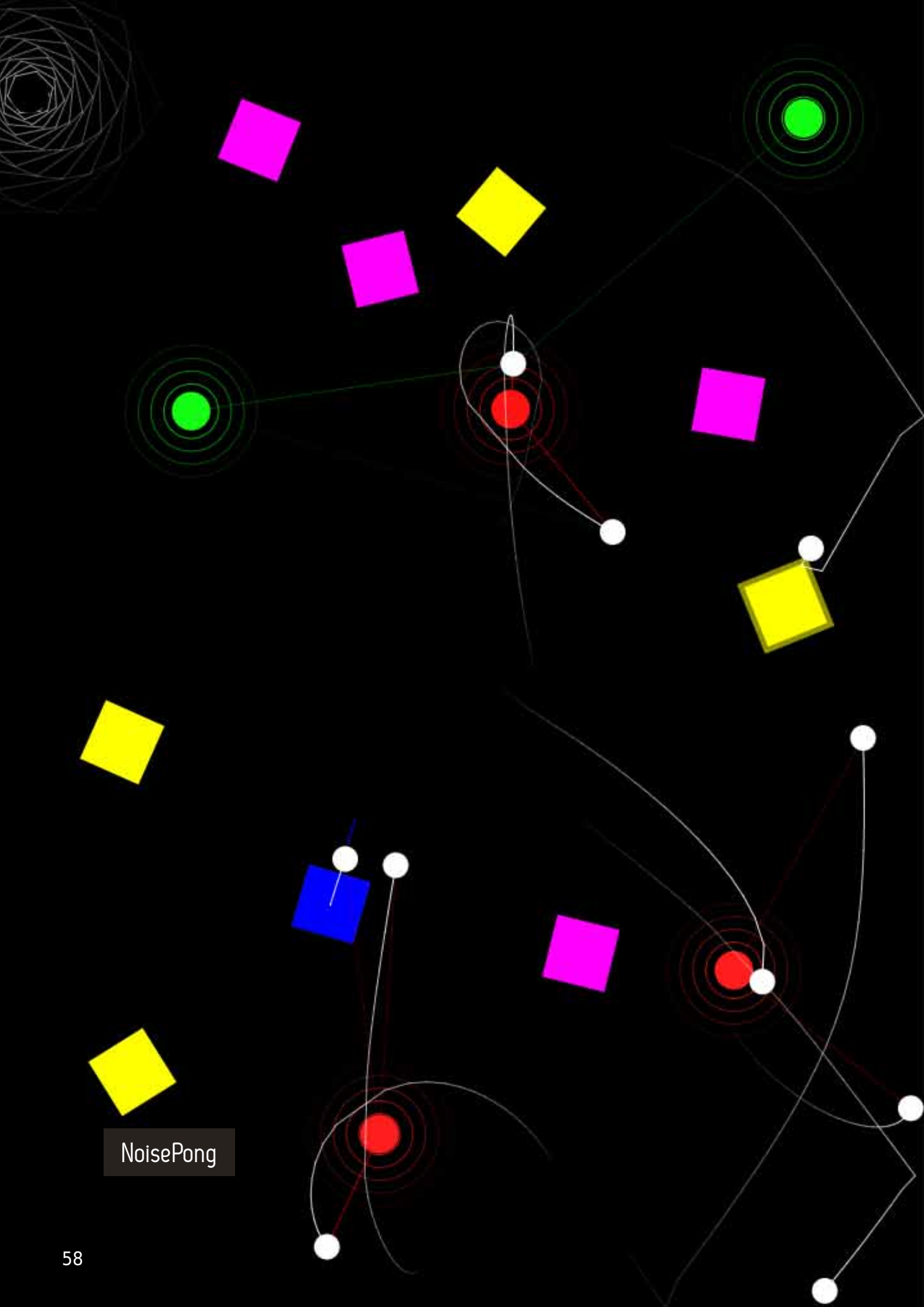
Hij werkt **steeds met de ingebouwde infraroodcamera** van de wiimote. Deze camera trackt automatisch tot **4 infrarood LED's** en geeft de posities door aan OSCulator of andere programma's. De IR LED's geven een **precieze plaatsbepaling**, maar worden bijvoorbeeld niet gebruikt bij Wii Tennis, daar doen alleen de accelerometers hun werk. De IR LED's zijn uit het zicht bij een tennisslag en daar moet de wiimote vertrouwen op zijn accelerometers.

Hij heeft drie projecten die bekend zijn bij het grote publiek, en dat zijn een wiimote-fingertracker (verwant aan rond wat Gilles Vermeulen vorig jaar met zijn thesis heeft gedaan), een multi-touch whiteboard en het bekendste project, gezichtstracking van een gamer waardoor hij zijn zicht op een speelveld kan veranderen door zichzelf fysiek te verplaatsen.

De projecten zijn zo populair wegens hun **grote toepasbaarheid**. Lee toont dat met één enkele Wii Remote en enkele IR LED's de inleving in een spel sterk verbeterd kan worden. Het is nog wachten op de eerste games die dit soort dieptezicht introduceren.

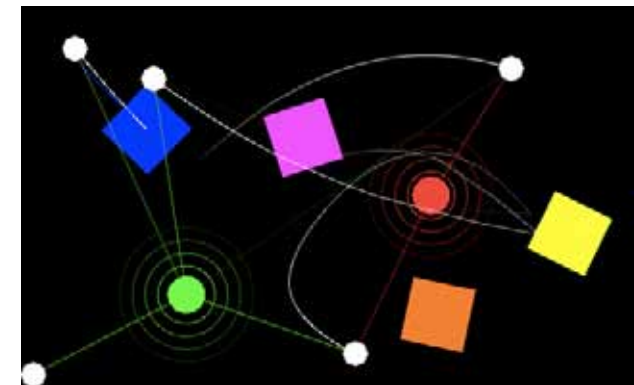
Johnny Chung Lee werkt momenteel als researcher bij Microsoft mee aan Project Natal.





## Ervaringen met gesturale interfaces tijdens dit masterjaar

### Noisepong



Noisepong was een project dat Jim Bollansée en ik maakten voor onze keuzemodule "Play".

De bedoeling van deze module was een **game of applicatie te maken met een fysieke interface**. Sommige studenten deden projecten met webcam vision, sommige projecten waren multi-touch gebaseerd, en Jim en ik gebruikten het **ReactIVision platform** om een **tastbare tafelinterface** te maken.

We wilden in analogie met de originele ReactTable een **muziekinstrument** maken maar ook een **spelelement introduceren**. We kozen ervoor om een muzikale interface gebaseerd op fysica te maken.

ReactIVision (zie de eerdere bespreking van ReactTable) gebruikt speciaal ontworpen tekens die de webcam kan volgen en zo via een projector die onder de tafel zit, dingen projecteren om op de positie van

deze tekens in te spelen. De tekens worden meestal aan de onderkant van blokken geplakt om er handig mee te kunnen omgaan.

In ons geval stond één blokje voor een kanon dat witte ballen lanceerde. Deze ballen werden dan beïnvloed door andere blokken en dat maakte de muziek.

De gele blokken kaatsten ballen af, en lieten daarbij een heldere noot horen. De draaiing van de blokken koos de toonhoogte van de noot die gespeeld werd. Dit gebeurde in een vaste toonaard, zodat er geen valse noten te horen waren.

De paarse blokken werkten met dezelfde manier van noten kiezen, maar hadden geen invloed op de baan van de ballen. Ze dienden enkel om muziek te produceren.

De rode en groene ronde bollen waren aantrekkings- en afstootvelden. Hoe dichtere een bal bij zo'n veld kwam, hoe hoger het volume van het strijkinstrument dat door dat veld vertegenwoordigd werd.

De spiraal was een zwart gat. Dit element had een verhoogde aantrekkingskracht om snel bollen te kunnen vernietigen. Een oranje blok met dezelfde functie bestond ook, maar dan zonder aantrekkingskracht. Dit was vooral handig om op het einde van een bepaald ballenpad te zetten en zo een oneindig voortdurend motief te maken (aangezien alle ballen wiskundig gezien hetzelfde pad volgen van lancering tot vernietiging als er geen blokken verplaatst worden).

Hoewel dit project bedoeld was voor een

tastbare tafelinterface, is het ons toch **niet succesvol gelukt om te doen werken met een ReactIVision tafel**. De lichtomstandigheden en camera waren niet optimaal, en na lang proberen hebben we dan toch besloten om het programma te presenteren via een simulatie van een ReactIVision tafel.

Dit project was echter wel onze **eerste ervaring met ontwerpen voor een gesturale interface**. We moesten ontwerpen voor **meerdere gebruikers** die tegelijk handelingen konden uitvoeren en ermee rekening houden dat alles nog **gebruiksvriendelijk** en **intuïtief** moest zijn. Tevens is dit het eerste volledig object-georiënteerd programma dat ik aan mee heb gewerkt, wat belangrijk is in toepassingen voor meerdere gebruikers om een eindeloos schalend systeem te kunnen maken.

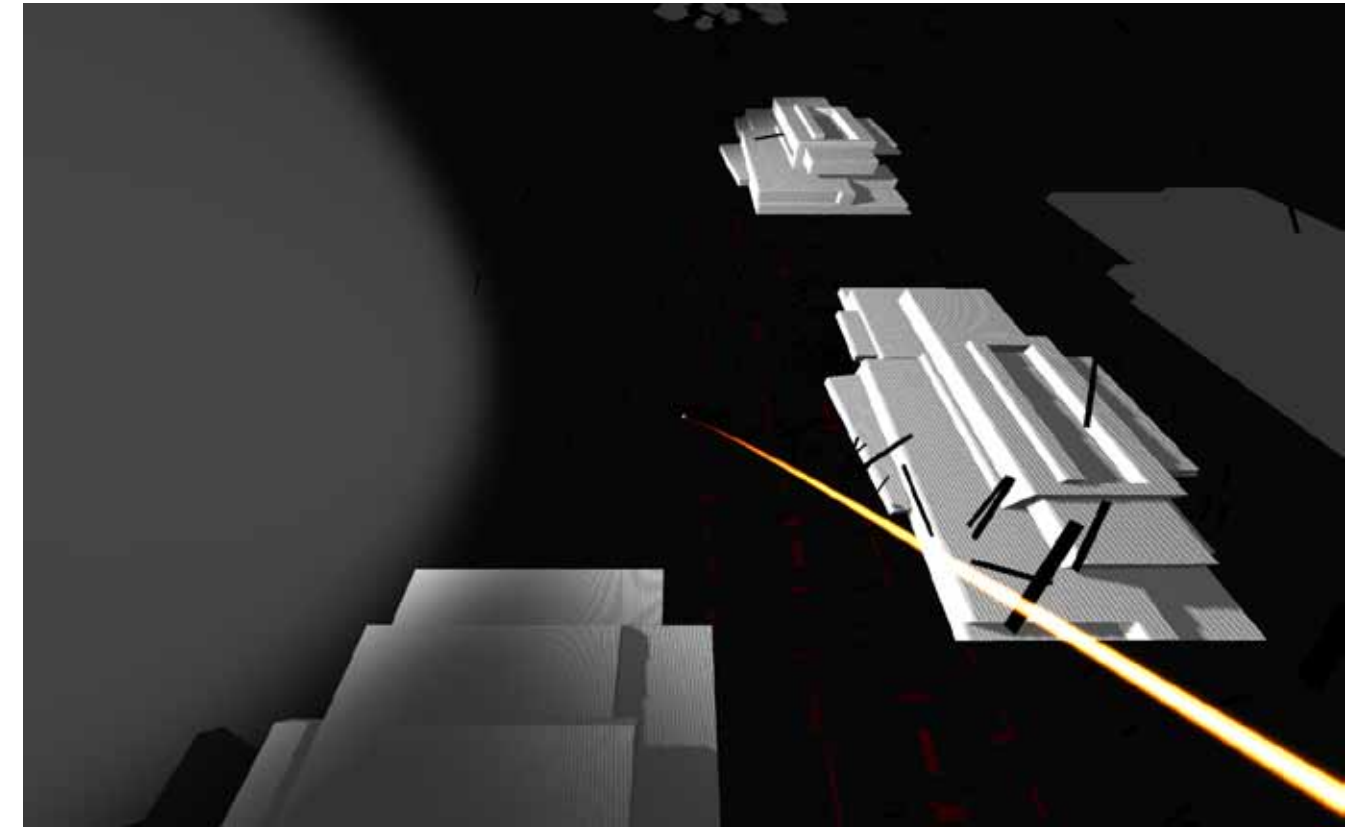


Ook de keuze van muziekparameters moest goed over nagedacht worden. Uiteindelijk leek me de nabijheid van de dichtste bal bij de velden een goede parameter om te gebruiken in het volume van de geluiden, zodat je de ballen zou kunnen horen langs zoeven.

Op **technisch vlak** tenslotte was dit project ook belangrijk omdat ik voor de **eerste keer een audioprogramma bestuurd vanuit een ander programma**. Onze Processing applicatie zond OSC signalen uit naar een Pure Data applicatie, dat deze vertaalde naar MIDI parameters die gebonden waren aan verschillende bedieningselementen van Ableton Live. Als je hierbij nog eens het ReactIVision platform rekent, draaiden we **4 programma's tegelijk**.

### Glasgow Mega Snake

"Glasgow mega snake" was een project dat we hebben gemaakt voor een workshop in



Vlaams cultuurhuis "De Brakke Grond" in Amsterdam. Samen met Danny Leen, Tom Luyten en Jim Bollansée ben ik naar daar afgezakt voor een **workshop in fysieke input in games**, gebruik makende van de game-engine Unity.

Na deze vier dagen workshop hadden we een spel gemaakt dat niet zozeer beoordeeld moest worden op zijn doel of verhaal, maar **eerder iets dat een visuele stijl nastreefde en intuïtief kon bestuurd worden**.

Genoemd naar een lied van Mogwai, bestond het spel eruit om een "slang", een bal die een heldere streep achterliet, te besturen met een Wii Remote door een eindeloze level. De level bestond uit een grijze geul met op de bodem een rood-zwarte stylistische "stad".

De bal kon bestuurd worden door de Wii

Remote naar voor of achter te hellen, en naar links en rechts te draaien. Als de gebruiker een schok omhoog gaf aan zijn Wii Remote, sprong de bal. De bijhorende soundscape benadrukte de snelheid van de bal, de bochten die hij nam, en de hoogte van de sprongen.

Vele elementen moesten we niet meer programmeren omdat ze al in Unity zelf zaten, zoals partikelgenerators, zwaartekracht en botsingdetectie.

De Wii Remote werd aangesloten via **OSCulator**. Onze applicatie zond op zijn beurt dan ook nog eens OSC signalen uit, die weer via Pure Data geformatteerd werden tot hapklare MIDI voor Ableton Live.

Dit was mijn eerste project waar ik een Wii Remote heb gebruikt om interactie met een applicatie te doen.

De beweging moest **zo intuïtief mogelijk**



**zijn**, en daarom hebben we vooruit-achteruit-links-rechts letterlijk vertaald naar het scherm. Als de Wii Remote niet te snel bewogen werd en de hele tijd met de bovenkant naar boven werd gehouden, konden deze parameters immers vrij goed uitgelezen worden.

Voor de herkenning van de sprong was geen gecompliceerd detectiesysteem nodig, integendeel. De gebruiker kende maar één snelle beweging, en dat was de sprong. Dus konden we bij een verhoogde acceleratiewaarde er zeker van zijn dat de gebruiker een sprong probeerde te doen. Dit haalde meteen de verschillen tussen hoe mensen hun beweging deden weg. Door niet te vertellen aan de mensen dat ze hun Wii Remote ook naar onder, links en rechts konden tikken, voerden ze deze bewegingen ook niet uit. Bij mijn weten heeft geen enkele persoon die het spel testte ooit ongewild de bal doen springen.

### I think I broke myself

I think I broke myself was een **ervaringsruimte** die we maakten in opdracht van de keuzemodule “Experiency”. Met een **multidisciplinair team** bestaande uit vier TV-Film studenten, één Productdesign student en twee Communicatie & Multimediadesign studenten maakten we **een kamer die een immersieve ervaring bood voor de personen die er binnengingen**.

Na enkele moeizame brainstormsessies gingen we er gezamenlijk mee akkoord om een spiegel te maken die de gespiegelde zou opbreken in allemaal stukken als die

dichter bij de spiegel kwam. Hoe dichter bij de spiegel, hoe meer gebroken. We besproken verschillende methodes om de digitale reflectie te maken, inclusief vrij complexe oplossingen met halftransparante spiegels om de camera zo direct mogelijk tegenover de persoon in de ruimte te zetten. Uiteindelijk werden al deze systemen te complex bevonden en kozen we ervoor om de **camera** en de **afstandssensor** vlak naast de spiegel te zetten. De spiegel was niet echt een spiegel, maar kalkpapier waar langs achter op geprojecteerd werd.

We hadden net een workshop over het programma **Isadora** achter de rug, en deze tool leek ons het best geschikt om het binnenkomende beeld in stukken te hakken, nadat enkele eerdere experimenten met Processing niet tevredenstellend bleken door een langzame snelheid van beelden.

De ruimte was in twee helften verdeeld door een zwarte doek met een uitsparing voor de spiegel, zodat de computers, sensor, camera en projector niet of moeilijk zichtbaar waren.

Dit project gebruikte **sensoren om het menselijk lichaam te detecteren** en is dus eigenlijk een **vorm van gesturale interface**. Dit was zeer belangrijk voor het immersief element, aangezien het opbreken van je eigen spiegelbeeld ogenschijnlijk **vanzelf** ging. Wat begon als een gewone webcam-reflectie werd gauw iets griezelig bij het weten dat de computer je in de gaten houdt.

Het is dus ook een soort van **ubiquitous computing**; de computer speelt in op onze

acties zonder dat we het merken, zonder dat we het weten zelfs. Dit soort naadloze opstellingen zorgt voor een onverwachte en indrukwekkende ervaring.

Met dit project heb ik veel geleerd over de **natuurlijkheid** die een gesturale interface voortbrengt. Op geen enkel moment hadden de personen het gevoel dat ze met

een computer aan het interageren waren, ook al waren de beelden en het geluid computergestuurd.

De gemeten data was ook vrij simpel; één afstandsparemeter. Dat deze enkele parameter de immersiviteit zo ten goede kon komen was indrukwekkend.





# Part 4: Prototypes

## Inleiding

Mijn prototypes moeten beschouwd worden als **iteraties**. Elke keer dat een prototype af is, **maak ik een evaluatie en trek ik er conclusies uit voor het volgende prototype**.

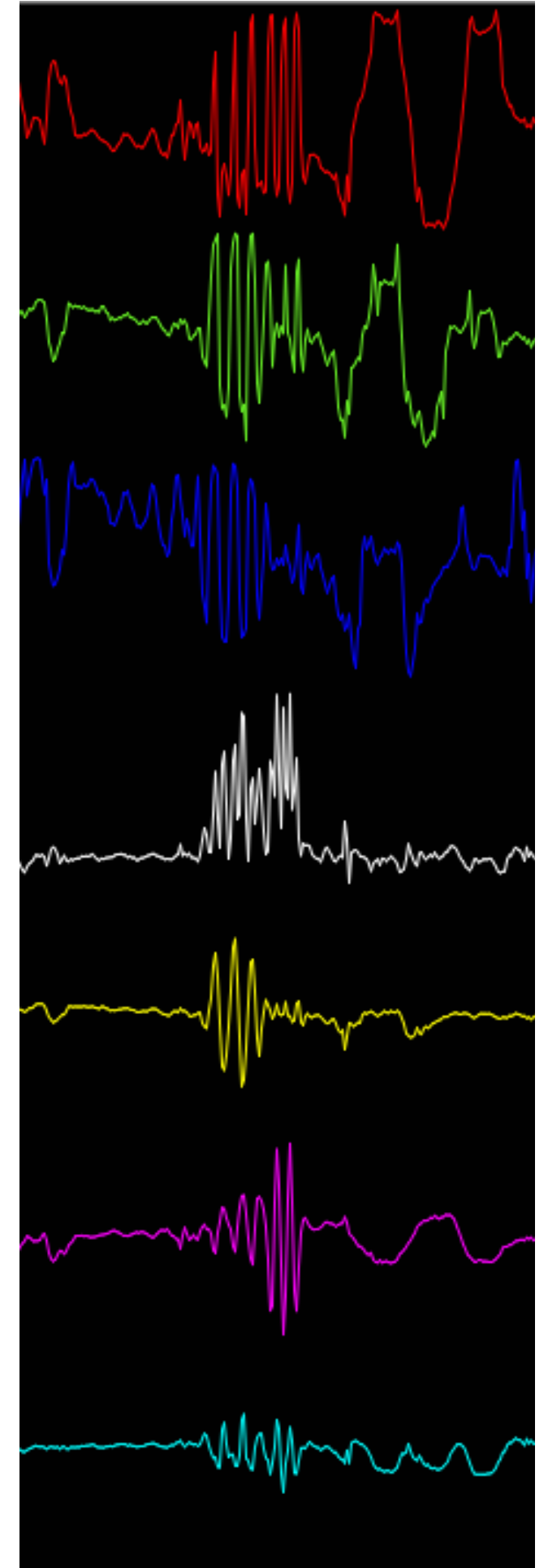
Elk prototype is een **free-form gesturale interface met Wii Remotes om computermuziek mee te besturen** en dat is de enige restrictie die ik me opleg. Het gebruik van de Wii Remotes verschilt bij elk prototype om **oplossingen te zoeken** voor problemen die in de prototypes daarvoor voorkwamen.

## PRE-PRODUCTION: Wii Remote data-analyse

### Concept

Voor ik aan werkende prototypes wou beginnen te werken besloot ik om eerst de **data te bestuderen** dat de wiimotes produceerden met hun accelerometers. Om enig idee te krijgen over hoe de wiimotes zich gedroegen in verschillende posities en bewegingen wou ik de data **lineair mappen in de tijd**.

Dit zou me inzicht geven in hoe de accelerometers zich gedragen bij langzame en snelle bewegingen, en zo kon ik de betrouwbaarheid van de afgeleide data controleren.

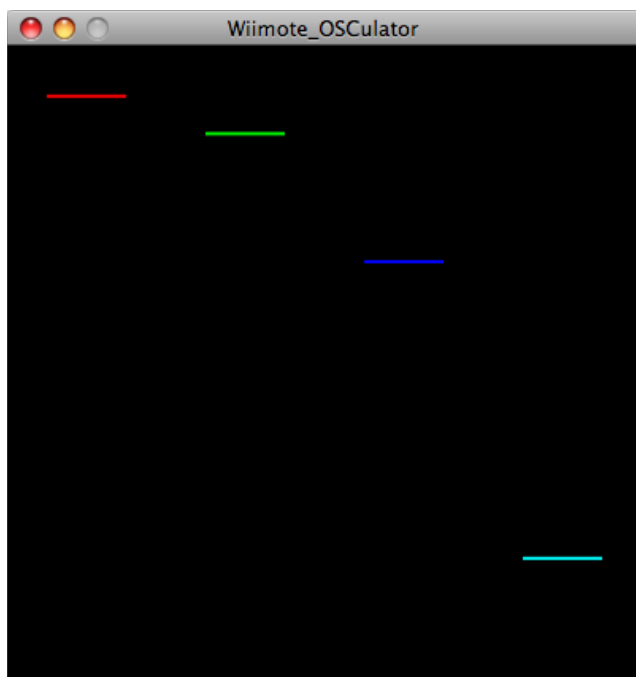


## Technologie / uitvoering

De wiimotes konden aangesloten worden via OSCulator. Dit programma zendt MIDI of OSC signalen uit naar andere applicaties. Het kan de **ruwe accelerometerdata (x,y,z) opvragen** en leidt ook zelf enkele parameters hieruit af, zoals Pitch ("stampen"), Roll ("rollen"), Yaw ("gieren") en een getal van 0 tot 1 dat algemene acceleratie aangeeft als som van de drie assen.

Ik koos ervoor om Processing te gebruiken als programmeertaal omdat het functies aan boord heeft die snel elementen op het scherm kunnen tekenen zoals een vierkant. Ook heeft Processing een library die het gebruik van OSC zeer gemakkelijk maakt.

Mijn eerste iteratie van mijn analyseapplicatie was een programma met een statische resolutie dat louter

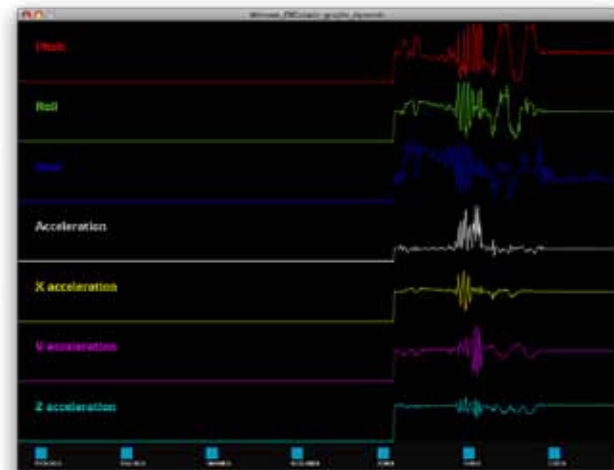


de **connectie tussen OSCulator en Processing testte**. Dit programma gaf alleen de actuele weergave van parameters

en niet wat ze in het verleden hadden gedaan.

Na de connectie tussen OSCulator en Processing succesvol te hebben opgezet, maakte ik een programma met dynamische resolutie (de resolutie werd aan het begin van het programma vastgelegd in variabelen en alle inhoud van het programma werd volgens deze resolutie geschaald) dat lineair alle parameters weergaf om data met elkaar te vergelijken.

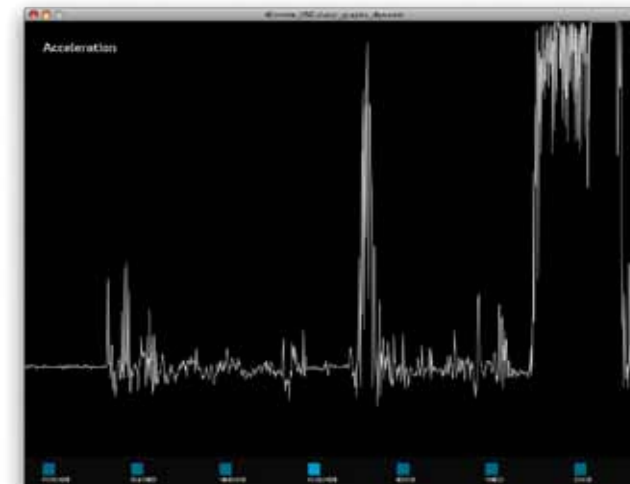
Dit visueel met elkaar vergelijken was soms moeilijk, aangezien de verschillende grafieken soms ver uit elkaar lagen. Daarom voegde ik in mijn laatste versie enkele simpele knoppen toe waarmee tijdens het lopen van het programma bepaalde sporen konden worden aan of uitgezet.



Applicatie met knoppen toegevoegd



Applicatie met enkel de Pitch, X, Y en Z parameters geactiveerd



Precieze studie van de acceleratieparameter door alle andere sporen uit te zetten.

## Conclusie

De 7 waardes die OSCulator produceert over de accelerometers werden uitgemaakt en konden vergeleken worden. Zo is het bijvoorbeeld duidelijk dat de drie **Pitch, Yaw, en Roll parameters zeer onbetrouwbaar zijn** om zomaar voor waar aan te nemen. Deze drie parameters worden door OSCulator berekend op basis van de x,y,z parameters en de (zwaarte)kracht die op deze sensoren wordt uitgevoerd. Rotatie in een vlak evenwijdig aan de grond (yaw) heeft zo geen meetbare uitlezing. Als de wiimote plat op een tafel ligt, kan OSCulator

geen goed oordeel vellen in welke richting de wiimote al dan niet aan het bewegen is in de yaw richting, en deze waarde begint dan ook willekeurig in het rond te springen.

Ook, aangezien deze drie parameters direct van de x,y,z waarden worden berekend, zijn ze gevoelig voor snelle bewegingen. De wiimote snel naar boven bewegen terwijl hij naar boven gericht is, doet de Pitch parameter eerst pieken en dan bijna naar nul zakken omdat hij eerst denkt dat hij helemaal naar boven is gericht door veel kracht in de richting van de zwaartekracht, en dan zakt hij naar nul door de plotselinge stopkracht die erop wordt uitgevoerd in de andere richting.

De Pitch, Roll en Yaw parameters zijn dus enkel betrouwbaar in langzame bewegingen, en alleen in bepaalde vlakken.

Wat ook zeer duidelijk bleek is dat de accelerometers **weinig hapklare, direct bruikbare data** bieden. De data is complex en plaatsbepaling of **1-op-1 3D oriëntatie is er niet uit af te leiden**. Dit is dan ook een veel gehoorde kritiek op de wiimote in zijn huidige staat. De accelerometers zijn niet precies genoeg en sommige games zijn een teleurstelling doordat de protagonist niet dezelfde beweging als de speler uitvoert met zijn zwaard/tennisracket/pistool. Daarom heeft Nintendo in juni van dit jaar de Wii MotionPlus uitgebracht. Deze bevat gyroscopen om een correcte 1-op-1 beweging te kunnen vormen en moet aan de achterkant van de wiimote ingepluggd worden. Tot op heden is het modden ervan enkel mogelijk via een Arduino microcontroller.

In de spellen die voor de Wii uitkomen wordt dan ook zelden dit soort absolute positieberekeningen gedaan. Bij bijvoorbeeld “MarioKart Wii”, waar de Yaw parameter betrouwbaar kan uitgelezen worden omdat de speler gedwongen wordt om de Wii remote altijd in hetzelfde vlak te houden.

Andere spellen maken dan weer gebruik van het detecteren van ruwe bewegingen, zoals het spel “The Blob”, waar de speler even met de wiimote moet schudden om het spelkarakter te laten springen.

Een grote categorie spellen maakt dan ook gebruik van **gesture recognition** om complexe bewegingen te detecteren, bijvoorbeeld bij Wii Sports Tennis. Het bedrijf AiLive, dat een zeer goede gesture recognition tool biedt voor ontwikkelaars van spellen voor de wii, heeft samen met Nintendo de wiimote ontwikkeld. Hun tool, LiveMove, drijft de besturing van talloze Wii spellen aan.

Voor precieze positionering en het wijzen naar het televisiescherm gebruikt de Wiimote een infraroodcamera die in de neus van het toestel verscholen zit. De Sensor Bar die op of onder de televisie geïnstalleerd moet worden bevat twee infrarood LED's die door de camera opgevangen worden. Dit soort positionering wordt gebruikt tijdens het besturen van de aanwijscursor

in de Wii menu's of het besturen van een schietspel.

## Waarom enkel accelerometers en geen IR gesturing?

Ik heb echter gauw ervoor gekozen om voor de rest van mijn prototypes de accelerometers te gebruiken en wel voor twee redenen. Vooreerst zit ik inherent verbonden aan een **performatieve context**, en op podia zijn er verschillende **zeer heldere lichtspots** die op de performer gericht staan. Dit zou de **metingen van de infraroodcamera kunnen verstoren**, aangezien deze lichten door hun helderheid ook worden opgepikt door de camera.

Een tweede reden is dat ik voor de detectie van complexe bewegingen kon vertrouwen op de open source Java library “WiiGee”. Deze library kan ook gestures detecteren die met de infraroodcamera worden gemaakt, maar dan enkel in de context van wijzen naar een scherm, met andere woorden als je bepaalde gebaren met één wiimote in een bepaalde richting doet.

De infraroodcamera kan ook in **omgekeerde zin** worden gebruikt. LED's (maximum vier) worden bevestigd aan de armen en benen van de performer en twee wiimotes traceren langs de voorkant en zijkant, desnoods vanuit nog meer richtingen voor betrouwbaarheid, de bewegingen die deze

LED's uitvoeren. Dit brengt echter het concept van **3D traceren** van elke LED mee, en dat overstijgt de mogelijkheden van een gesture library als WiiGee, en het overstijgt zeker mijn capaciteiten als programmeur. Bij het Expertisecentrum voor Digitale Media (EDM) in Hasselt gebruiken ze wel deze vorm van gesturing.

Een ander bezwaar tegen deze opstelling is dat ze beperkend werkt. De performer zal zich steeds binnen het gezichtsveld van de wiimotes moeten bewegen terwijl hij toch dicht genoeg bij elke wiimote moet staan zodat de wiimotes de LED's kunnen detecteren.

Dit gemaakte analyseprogramma heeft er dus toe geleid dat ik de keuze maakte voor accelerometer gesture recognition voor mijn tweede prototype, maar het gaf ook de inspiratie voor mijn eerste.

## PROTOTYPE 0: Trigger events by acceleration

### Concept

Mijn prototype 0 beschouw ik niet echt als prototype omdat er **amper ontwikkelingstijd** bij aan te pas is gekomen. Toch heb ik er enkele lessen uit kunnen trekken.

Het concept was dat met vier Wii Remotes **elke remote een eigen geluid** zou kunnen maken als de remote ruw werd bewogen. Zo konden de Wii Remotes ongeveer als drumstokken worden gebruikt.

**Elke Remote stuurde een noot van een sampler aan**, met samples die ik uit het lied “Eye of the tiger” had geknipt. Als ik twee wiimotes in mijn broekzakken stak, reageerden zij op schokken tijdens het wandelen / springen. Heen en weer trappelen zorgde dan voor het gitaarpatroon op de achtergrond van Eye of the Tiger.

De twee andere wiimotes hield ik in mijn handen, en die stuwde ik met een boksbeweging vooruit (in analogie met de film Rocky III), wat de korte akkoorden van het lied produceerde.

### Technologie/uitvoering

OSCulator, het programma dat ik had aangekocht om Wii Remotes met mijn laptop te verbinden, liet toe om **MIDI berichten direct uit te sturen naar Ableton Live**. De noten werden gemaakt via de **algemene acceleratieparameter** van elke wiimote.

### Conclusie

Het **voordeel** van dit prototype is dat **snelle acties van de performer direct verbonden worden aan geluiden**, wat hetzelfde directe effect geeft als een drumstel bespelen of een akkoord aanslaan op piano. Het geeft een zekere **“instant gratification”** (onmiddellijke bevrediging).

Eerste tests wijzen uit dat het een zekere **fun factor** heeft; gebruikers lopen wild in het rond terwijl ze constant boksen met hun vuisten om de wiimotes op hun armen over de drempel te stuwten.

Geluiden die geactiveerd worden zijn echter





beperkt. Per ledemaat kan men één geluid toekennen, ook omdat de Pitch parameter te onbetrouwbaar is om bijvoorbeeld één geluid af te spelen als je vooruit bokst, en een ander als er omhoog wordt gestoken. Dit aantal geluiden is dus eigenlijk zelfs te weinig voor een succesvolle renditie van “Eye of the Tiger”.

Dit kunnen we gedeeltelijk verhelpen. Met behulp van de A en B knop aan de wiimotes die in de handen van de performer worden gehouden, kunnen we zo tot vier verschillende geluiden komen per hand. Het totaal aantal geluiden wordt dan tien, wat al een goed aantal is om bijvoorbeeld een drummachine mee te besturen. Een nadeel van deze methode is dat beweging niet meer spontaan gebeurt en er over moet nagedacht worden.

Het **nadeel** van dit prototype is dus vooral het **onvermogen om complexe bewegingen in complex gedrag om te zetten**.

## PROTOTYPE 1: Convert movement into sound

### Concept

De inspiratie voor dit prototype kwam er wanneer ik de data visualiseerde die van de wiimotes afkomstig was. Ik bekeek een reeks **golven die geproduceerd werden door de accelerometers** en kwam op het idee om deze **golven letterlijk te vertalen naar geluid**, dus de golven die getekend werden direct door te geven

naar de luidsprekers om om te zetten naar hoorbare trillingen.

De trillingen die een mens met zijn ledematen kan maken zijn natuurlijk te langzaam om een hoorbaar geluid te produceren. Daarom heb ik een **“wavetable”** gemaakt die zichzelf altijd update. Een wavetable is een kort stukje data dat enorm veel herhaald wordt door een synthesizer. Het is als het ware een tabel die kort stukje geluidsgolf beschrijft. De bekendste wavetable is één fase van een sinusgolf. Dit soort golf genereert een rond, zacht geluid in zijn puurste vorm. Enkele andere veel gebruikte golven zijn zaagtand- en pulsgolven. Deze golven beschrijven zeer letterlijk het pad dat de luidspreker volgt bij het uitstoten van deze geluiden.

**Verschil in toonhoogte** wordt gemaakt door de **frequentie** waarop deze fase wordt herhaald te **variëren**. **Verschil in volume** wordt hier bepaald door de **snelheid** waarmee de muzikant zich beweegt; grotere snelheid betekent grotere uitschieters van de accelerometers, wat zich in audio vertaalt als **amplitude**, volume, van het signaal.

De frequentie wordt bepaald door een wiimote met de linkerhand hoger te richten voor hogere tonen, en lager voor lagere tonen. Via de trekkerknop op deze wiimote kan men noten spelen. Men kan er ook voor kiezen om de toonhoogte af te ronden op bepaalde waarden zodat ze binnen ons westers tonaal systeem passen van 12 chromatische tonen, of om ze af te ronden binnen een bepaalde toonaard.

### Case: de Theremin



Een instrument dat kan beschreven worden als de eerste primitieve vorm van een gesturale interface is de **Theremin**. Dit instrument werd uitgevonden door de Rus Léon Theremin (Lev Sergeivich Termen) in 1919 en was het resultaat in een onderzoek in **nabijheidssensoren**. De technologie manifesteerde zichzelf in **een van de eerste elektronische muziekinstrumenten**, de Theremin.

De Theremin is een toestel met **twee antennes**, één verticale en één horizontale. Het instrument wordt bespeeld via **nabijheid van de handen**; de rechterhand bepaalt via de afstand tot de verticale antenne de toonhoogte van een sinusgolf. Hoe dichterbij, hoe hoger de toon. De linkerhand regelt het volume van de toon. Hoe verder, hoe luider.

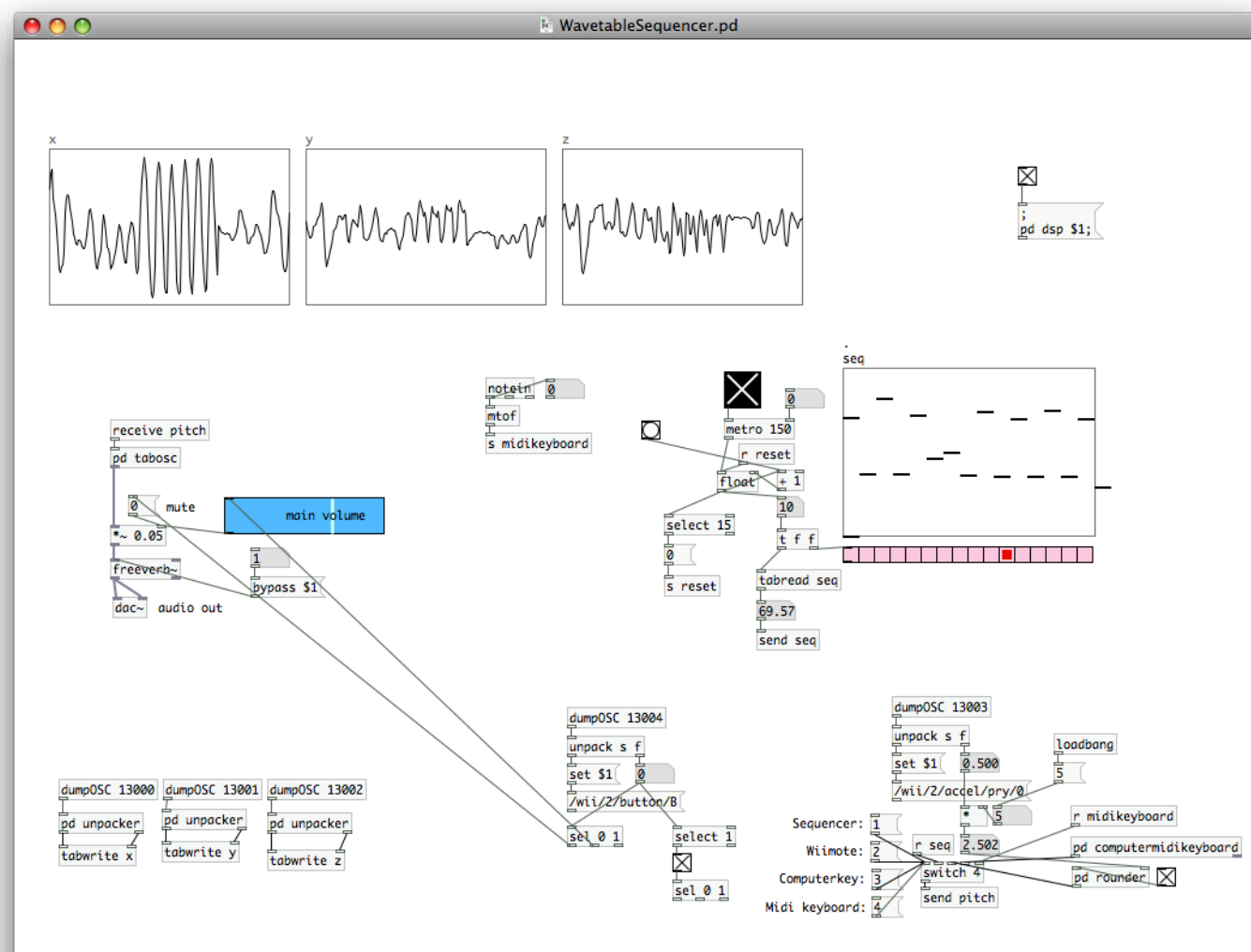
Omdat een toon moeilijk afgebakend kan worden (een hand kan immers niet plotseling van positie veranderen) klinkt het instrument erg vloeiend (portamento).

Het instrument won aan populariteit door het gebruik als **geluidseffect** in **sciencefictionproducties** van de jaren 50 en 60. Na een stille periode herwon het heel wat populariteit door een documentaire over het instrument in 1994.

Ik schets hier een kort overzicht van de aard van de Theremin om de **analogie tussen de Theremin en het instrument dat ik met dit prototype gemaakt heb**. Beiden worden **bestuurd door enkel beweging** en al hun expressie komt direct van letterlijke vertaling van deze bewegingen naar geluid. Allebei zijn ze een **synthesizer** die **op zichzelf als instrument kan gebruikt worden** zoals een gitaar ook een instrument op zich is. Allebei hebben ze een zeer eigen **herkenbaar geluid** dat toch rechtstreeks door het lichaam van de performer bepaald wordt.

De Theremin is ook **niet gebonden aan een bepaalde toonaard**. Er kunnen in principe alle mogelijke tonen mee gemaakt worden, of ze nu binnen het westers tonaal systeem van 12 halve tonen passen of niet. Dit is ook mogelijk met mijn prototype, al is de resolutie waarmee de richting van de wiimote wordt uitgelezen wel beperkt. Eventueel kan de applicatie afgerond worden op bepaalde waarden zodat het enkel tonen die in ons systeem voorkomen genereert.

Voor dit prototype heb ik dan ook een **gelijkaardig gebruik op het oog als de Theremin**. Met dat distinctieve **verschil** dat dit prototype volledige **vrijheid van beweging** garandeert op het podium.



## Technologie/uitvoering

De uitwerking gebeurde in **Pure Data**, wat ideaal bleek om wavetable synthese in uit te voeren. Eerdere experimenten in Processing draaiden op niets uit wegens bugs in de Java Virtual Machine implementatie voor Mac OS X, waar Processing op gebaseerd is.

Dit prototype **combineert de snelheid van reactie van prototype 0 en de eigen identiteit van beweging van de performer van prototype 2**. Het is te gebruiken als synthesizer in een muziekgroep, zoals een synthesizer normaal zou gebruikt worden.

Ik heb verschillende parameters als golf gebruikt; de Pitch, Roll en Yaw parameters, maar ook de algemene acceleratieparameter en de x,y,z ruwe acceleratieparameters. Uiteindelijk gaf een **combinatie van de x,y,z parameters** het beste resultaat met de meeste controle over het geluid. De golven worden bij elkaar opgeteld om elkaar te vervormen. De algemene acceleratieparameter was geen goede kandidaat voor dit prototype omdat het mogelijk is om constant heel veel te bewegen, waardoor de parameter constant bijna 1 benadert en uiteindelijk de hele wavetable met 1 gevuld is, waardoor er geen trilling meer ontstaat door de speakers. Snelle bewegingen werden dus uiteindelijk terug omgezet in stilte.

Doordat dit prototype in Pure Data gemaakt is, een **open source** programma, is het **bruikbaar en aanpasbaar door iedereen** met toegang tot een computer met bluetooth en wiimotes. De communicatie met de wiimotes verloopt via OSCulator, maar kan vervangen worden door een van de open source alternatieven zoals DarwiinRemote.

Pure Data scoort echter **niet goed op vlak van usability** als de gebruiker het programma of de manier van programmeren niet kent. Daarom is **enige training in Pure Data** aangeraden bij het gebruiken van dit prototype.

## Conclusie

Het instrument is een **vrij vermoeiend instrument** om te bespelen. Aangezien de wavetables worden gevuld met bewegingsdata, moet de performer **blijven bewegen om geluid te kunnen maken**. Ook is de actie niet gelijk verdeeld over de beide armen, aangezien de **rechterhand over het algemeen meer beweegt** dan de linkerhand. Dit is echter niet anders bij andere instrumenten zoals gitaar, waar de verdeling tussen linker- en rechterhand ook sterk kan verschillen.

Het is vrij goed mogelijk om bepaalde tonen te vormen met de rechterarm. Hevige bewegingen lokken een scherpere toon uit dan ronde, zachte bewegingen. Volume is rechtstreeks verbonden aan snelheid van beweging.

Het instrument heeft echter wel een sterke "release" tijd, dit is de tijd waar het

instrument nog kan gehoord worden nadat er een noot is gespeeld (hierbij wordt een noot dus afgebakend door bewegen - niet bewegen). Dit komt doordat de data van de vorige, misschien hevige bewegingen nog even in de wavetable blijven. De oplossing kan zijn om de wavetable te verkleinen in aantal samples, maar zo verliest de performer dan weer de mogelijkheid om complexere geluidsgolven te maken.

Daarom heb ik een extra "drempel" toegevoegd om een noot te doen spelen: de knop op de rug van de wiimote van de linkerhand moet ingedrukt zijn. Zo kan de muzikant zelf eender wanneer hij wil de noot afbreken.

Beide mogelijkheden zijn uiteraard aan te passen en te verwijderen of toe te voegen door verdere gebruikers van de software.

De voordelen van dit prototype zijn:

- Het achterliggende concept van beweging die rechtstreeks wordt vertaald naar geluid.
- Vrijheid van beweging op het podium, de performer mag eender welke beweging uitvoeren die hij wil.
- Te gebruiken als instrument in de traditionele zin.

De nadelen zijn:

- Slechts één reactie op beweging door computer: een synthesizergeluid.
- Complexe beweging lokt niet zozeer een complexer geluid uit, eerder een

“ander” geluid.

- Vermoeidheid van ledematen na langdurig gebruik. (Is met traditionele instrumenten ook zo).

Als reactie op deze eerste twee nadelen heb ik de basis voor mijn volgende prototype gevormd.

## PROTOTYPE 2: Trigger events by performing own gestures

### Concept

Na deze twee prototypes kon ik nog geen werkelijke bewegingen herkennen, dus daarom wou ik nog een derde prototype fabriceren dat zou werken op het principe van **gesture recognition**.



Gesture recognition werkt via het **herkennen van eerder ingespeelde bewegingen**. De muzikant kan een beweging maken die later herkend wordt zodat er een signaal wordt gestuurd (via MIDI of OSC) dat de muzikant zelf vastlegt.

Zo kan hij de beweging aanpassen aan zijn eigen stijl en genre.

Elke beweging **stuurt een MIDI of OSC signaal** uit waar de gebruiker in de eindapplicatie mee kan doen wat hij wil. Zo kan een bepaald dansgebaar in zijn muziekprogramma een muziknoot spelen, terwijl een ander gebaar kan betekenen dat de volgende strofe van het lied geactiveerd moet worden.

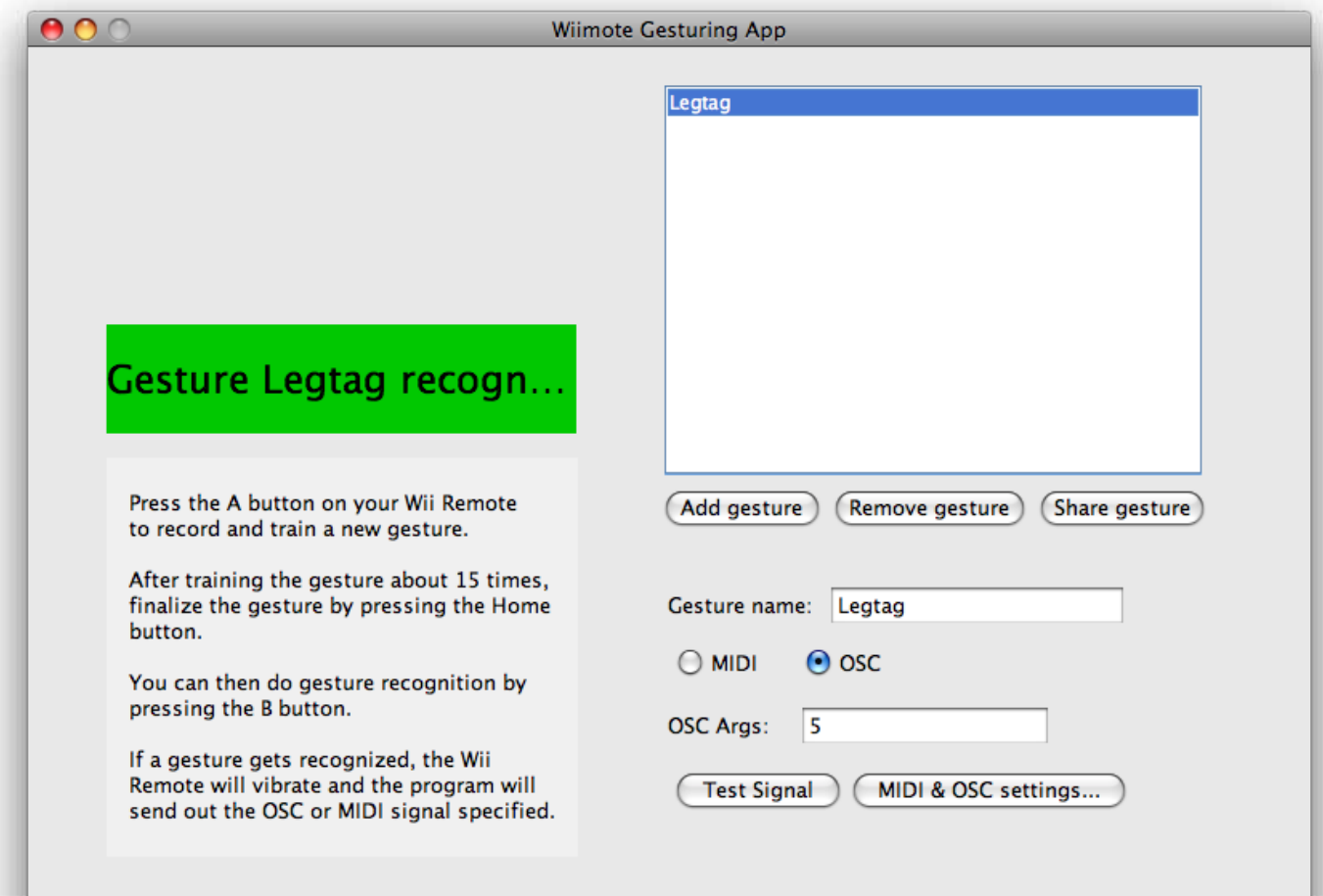
Dit geeft de **volledige vrijheid van beweging** aan de performer, en zo kan hij zijn **volledige eigen persoonlijkheid in stoppen**.

### Case: OSCulator

**OSCulator**, het programma dat ik meestal gebruik om mijn Wii Remotes aan mijn laptop te verbinden, heeft **gelijkaardige output**. De verschillende parameters van een Wii Remote worden via MIDI of OSC uitgestuurd en kunnen bijvoorbeeld volume regelen of bepaalde noten spelen. Via OSC is ook het gebruik in andere programma's mogelijk, zoals zelfgeschreven games.

OSCulator **herkent echter geen specifieke complexe bewegingen**. Het geeft enkel de output van de Wii Remote door aan andere applicaties, zoals draaiing, acceleratie, en ingedrukte knoppen.

OSCulator wordt door verschillende artiesten gebruikt op het podium op deze simpele manier. Mijn prototype zou een **sterke uitbreiding van de mogelijke betekenisvolle bewegingen** zijn.



### Technologie/uitvoering

Er bestaat een commerciële vorm van gesture recognition voor wii games: LiveMove van AiLive. Ook bestaan er verschillende open-source libraries voor C, C# en java, waar een open-source developer zelf mee aan de slag kan. De “**WiiGee**” library is de enige library voor Java, waarin ik een dus een applicatie maak.

Wiigee is gemaakt aan de Universiteit van Oldenburg door Thomas Schlömer, Benjamin Poppinga, Niels Henze en Susanne Boll en is vergezeld door een verklarende paper.

Data die binnenkomt van de accelerometers wordt eerst door twee filters gehaald. De eerste filter dient ertoe om het effect van

de zwaartekracht op de accelerometers te verminderen, terwijl de tweede het aantal vectoren vermindert en afrondt. Dan wordt er een quantizer op toegepast, die het aantal mogelijke bewegingen in een 3D ruimte afrondt tot 14 mogelijkheden. Wiigee zet dit getal op 14, omdat 18 geen merkbare verbetering van herkenning geeft, maar het proces wel vertraagt, en 10 is te weinig om betrouwbare herkenning op uit te voeren.

Na deze quantizer volgt er een wiskundig model dat veel gebruikt wordt binnen de signal processing en recognition; het Hidden Markov Model (HMM). Dit is een model met verschillende “staten” en een kansenmodel dat beschrijft hoeveel kans er is dat hij van een bepaalde staat naar een andere overschakelt.



De uitkomst van het HMM wordt aan een Bayes-classificatiesysteem gekoppeld om de juiste gesture te herkennen.

Met deze library is een open source demo applicatie gemaakt die downloadbaar is van de Wiigee website. Met behulp van deze demo-applicatie ben ik beginnen programmeren aan mijn eigen applicatie.



De user interface wordt gemaakt met behulp van de Java Swing classes. Om deze user interface te ontwerpen heb ik gebruik gemaakt van de “**paper prototyping**” methode. Het liet me toe om de belangrijkste usabilityproblemen op voorhand te kunnen omzeilen. Ik testte het papieren prototype op verschillende gebruikers, waaronder een iemand die nog maar weinig met een computer had gewerkt, waardoor bepaalde “common ground” problemen kunnen voorkomen.

Als de wiimote is aangesloten toont hij via de LED's op de voorkant welke nummer hij heeft toegewezen gekregen. De wiimote trilt ook even om te laten weten dat hij aangesloten is. Deze trilling treedt ook op bij herkenning van een gesture.

Via een lijst wordt bijgehouden welke gestures gedetecteerd kunnen worden

en wat hun actie is als ze gedetecteerd worden. Het aanleren en detecteren van gestures gebruikt met knoppen op één van de wiimotes zelf.

Het signaal dat de gedetecteerde gestures uitzenden kan uitgezonden worden als testsignaal, bijvoorbeeld voor in gebruik met “MIDI-learn”, waarbij parameters automatisch gemapt worden aan binnenkomende MIDI signalen.

### Sharing & DIY aspect

Gestures kunnen in tekstuele vorm opgeslagen worden. Deze tekstbestanden kunnen doorgestuurd worden naar andere gebruikers die dan die gesture kunnen gebruiken in hun eigen performance.

Bij dit delen van gestures komt wel het probleem van het communiceren van de juiste uitvoering van de gestures. Een gebruiker zou bij het verspreiden van zijn gesture een youtube filmpje online kunnen zetten dat de gesture demonstreert en documenteert.

Dit delen van gestures kan interessante gevolgen hebben. Zo zou het competitief kunnen worden om bepaalde technisch moeilijke gestures te kunnen uitvoeren. Het onderstreept ook nog eens te meer het open karakter van deze prototypes en deze thesis in het algemeen.

### Conclusie / Usertests

Eerste tests met de wiigee demo applicatie wezen uit dat de gesture recognition goed werkt voor gebaren die ongeveer 15 keer aan het systeem aangeleerd zijn. het

gebruik van meerdere wiimotes zou deze herkenning nog moeten verbeteren.

Als het leerproces echter niet goed wordt doorlopen of de verschillende trainingen onnauwkeurig verlopen gaat het systeem snel de verschillende gestures met elkaar verwarren. Daarom raad ik elke gebruiker aan de verschillende gestures uitvoerig te testen voordat die het in een live situatie gebruikt.

Een parameter die ook belangrijk is voor de herkenning van verschillende gestures is tijdsduur. Dit is voor sommige muziekgenres perfect aangezien de verschillende bewegingen op de maat van de muziek kunnen ingespeeld worden. Zo kan elke gesture die de muzikant op een podium gebruikt bijna exact even lang zijn als de gesture die aangeleerd is aan het systeem.

Door de papieren prototypes en common ground heb ik vele usabilityproblemen kunnen vermijden. Toch is het programma niet voor iedereen geschikt, vereiste is dat men iets kent van MIDI of OSC en hun toepassingen vooraleer er iets met de applicatie kan ondernomen worden. Vele computermuzikanten hebben deze technieken echter goed onder de knie door hun vele voorkomen in andere applicaties.

Sommige van de problemen met free form gesturale interfaces ontweek ik door de gebruiker zelf hun gebaren te laten kiezen. Dit kan echter wel op een verkeerd gebruik ervan uitlopen, zoals gestures die fysiek uitputtend zijn bij lang gebruik. Maar dit is in weze niet anders bij traditionele

instrumenten, waar het overoefenen tot vermoeidheid en fysiek letsel kan leiden. Dit kan bij mijn prototype vooral voorkomen bij het inoefenen van de gestures. De limiet van 15 “trainingen” per gesture zou dit gedeeltelijk moeten voorkomen samen met de relatief korte duur van een standaard muzikoptreden.

Nog een punt dat ik vermijd is de ongeschiktheid van gesturale interfaces voor fysiek gehandicapten, aangezien de interface vrijwillig wordt gebruikt door performers die fysiek in staat moeten zijn om op een podium te staan.

Dit prototype biedt ook haptische feedback aan de gebruiker, maar misschien weinig betekenisvolle, aangezien de enige feedback mogelijk een trilling van de wiimotes is. Het variëren van deze trilling zou in de toekomst hiervoor een oplossing kunnen zijn.



Battles

## Race Out

Ik ben deze thesis begonnen met een **vaag beeld op gesturale interfaces** en hoe men ervoor moet ontwerpen. Na een lang proces heb ik een goed beeld gekregen op wat er momenteel allemaal mogelijk is, gedaan wordt, en wat er in de toekomst mogelijk gaat zijn met gesturale interfaces.

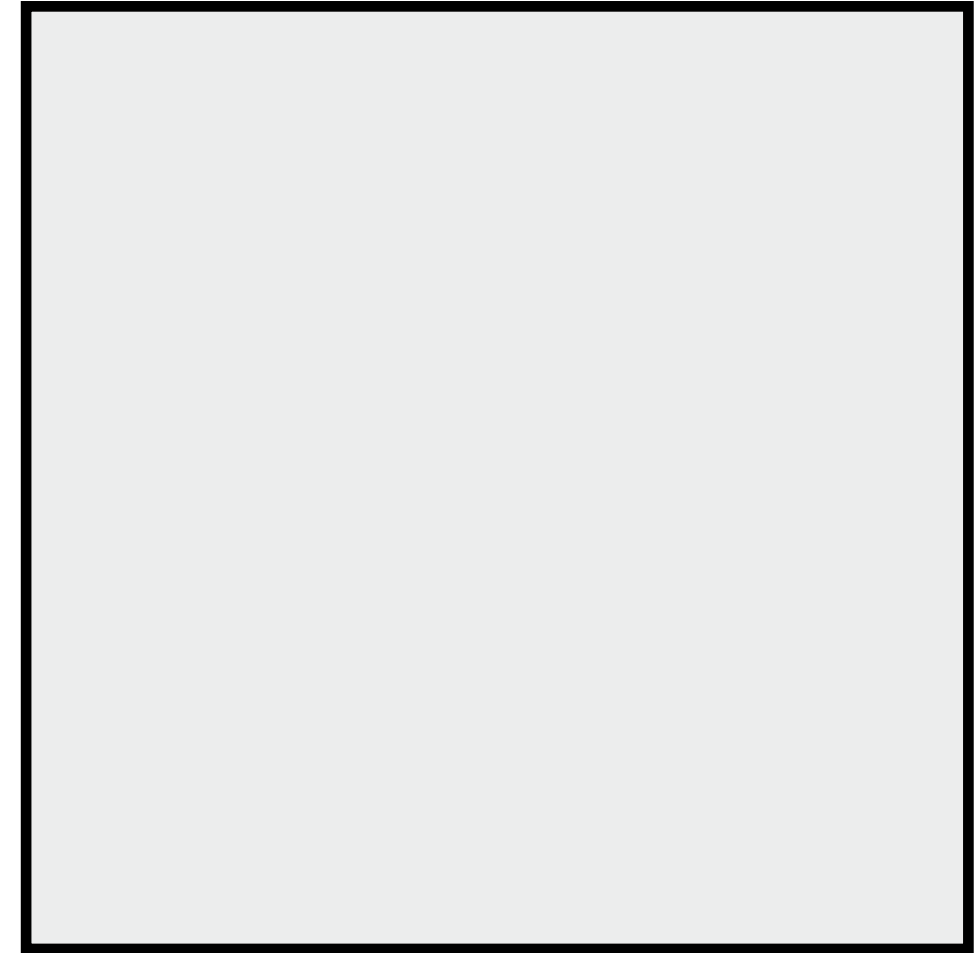
Ik voel me nu ook **sterker aangetrokken door de open source beweging** en de idealen waarvoor ze staan. Waar het voor de meeste mensen gelijk staat aan “gratis” geloof ik nu echt dat het open source model en transparantie op het vlak van softwareontwikkeling zowiezo **positief** is. Deze ervaring heeft gemaakt dat ik in de toekomst zoveel mogelijk mijn software en andere projecten toegankelijk ga proberen te maken voor de massa.

Of ik zelf veel de tools zal gebruiken die ik tijdens deze thesis ontworpen heb weet ik nog niet zeker. Wat ik wél weet is dat ik de verdere ontwikkeling van het gesture recognition prototype zal ondersteunen en **hopelijk springen er enkele open source developers mee op de boot.**

Wat ik vooral wou bereiken was een interface die laptopperformers vrijer maakte op het podium, en ik heb in elk geval enkele interessante en leerrijke pogingen ondernomen.

## Media

De schijf op de volgende pagina bevat de broncode voor mijn prototypes, plus geluidsopnames en extra foto's van gebruikerstesten.





# Bibliografie

## Literatuur

Angesleva, J. (2004). Body Mnemonics - interface design proposal for portable devices. University of Lapland, Department of Art and Design

Chu, L. (1996). Haptic Feedback in Computer Music Performance. Center for Music Technology (CCRMA), Northwestern University.

Ghazala, R. (2005). Circuit Bending. Wiley Publishing, Inc.

Huybrechts L. (2008). CrossOver. Lannoo/BAM.

Jorda, S. (2003). Interactive music systems for everyone: exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments. Music Technology Group, Universitat Pompeu Fabra.

Kranenburg, R. v. (2008). The Internet of Things. Amsterdam: The Institute of Network Cultures.

Laurel, B. (1991). Computers As Theatre. Addison-Wesley Publishing.

Laurel, B. (2003). Design Research. MIT Press.

Moen, J. (2006). Kinaesthetic Movement Interaction: Designing for the pleasure of motion. Kungliga Tekniska Högskolan, Stockholm.

Moggridge, B. (2006). Designing Interactions. MIT Press.

Norman, D. (2004). Emotional Design. Basic Books.

Preece, J. , Rogers, Y. , Sharp, H. (2007). Interaction Design. Wiley Publishing, Inc.

Raes, G.W. (1993). Een Onzichtbaar Muziekinstrument. Rijksuniversiteit Gent.

Rovan J. , Hayward V. (2000). Typology of Tactile Sounds and their Synthesis in Gesture-Driven Computer Music Performance.

Saffer, D. (2008). Designing Gestural Interfaces. O'Reilly.

Schlömer T. , Poppinga B. , Henze N. , Boll S. (2008). Gesture Recognition with a Wii controller. University of Oldenburg & OFFIS Institute for Information Technology.

## Web

BOLT, R, "Put-That-There": Voice and Gesture at the Graphics Interface (2009, [http://www.media.mit.edu/speech/papers/1980/bolt\\_SIGGRAPH80\\_put-that-there.pdf](http://www.media.mit.edu/speech/papers/1980/bolt_SIGGRAPH80_put-that-there.pdf))

BOOTH, C, Alan Kay and the Graphical User Interface (2009, <http://www.lottiebooth.com/pdf/essay.pdf>)

Center for Computer Research in Music and Acoustics (CCRMA) (2008, <http://ccrma.stanford.edu/>)

Eboman SmadSwiki (2009, <http://www.eboman.info/wiki/nl/index.php?title=Eboman>)

Glove-based input interfaces (2009, <http://www.geocities.com/mellott124/glove1.htm>)

Het EI van Godfried (2009, [http://www.logosfoundation.org/instrum\\_gwr/godfriedsei-ned.html](http://www.logosfoundation.org/instrum_gwr/godfriedsei-ned.html))

Hidden Markov Models (2009, <http://www.cs.brown.edu/research/ai/dynamics/tutorial/Documents/HiddenMarkovModels.html>)

Iwata Asks - Wii MotionPlus (2009, [http://us.wii.com/iwata\\_asks/wiimotionplus/](http://us.wii.com/iwata_asks/wiimotionplus/))

Johnny Chung Lee – Projects – Wii (2008, <http://johnnylee.net/projects/wii/>)

NIELSEN, J, Top-10 Application-Design Mistakes (2008, <http://www.useit.com/alertbox/application-mistakes.html>)

RAMOCKI, M, DIY: The Militant Embrace Of Technology (2009, <http://www.ramocki.net/ramocki-diy.pdf>)

SLOrk, Stanford Laptop Orchestra (2008, <http://slork.stanford.edu/>)

STEIM (2008, <http://www.steim.org/steim/>)

WARK, M, A Hacker Manifesto (2009, [http://subsol.c3.hu/subsol\\_2/contributors0/warktext.html](http://subsol.c3.hu/subsol_2/contributors0/warktext.html))

WiiGee (2009, <http://www.wiigee.org/index.html>)

Wikipedia

Alle inhoud van Wikipedia werd voor het gebruik in deze thesis gecontroleerd via de originele bronnen van de Wikipedia artikels.

Wikipedia, Affordance (2009, <http://en.wikipedia.org/wiki/Affordance>)

Wikipedia, Agency (2009, [http://en.wikipedia.org/wiki/Agency\\_%28philosophy%29](http://en.wikipedia.org/wiki/Agency_%28philosophy%29))

Wikipedia, Alan Kay (2009, [http://en.wikipedia.org/wiki/Alan\\_Kay](http://en.wikipedia.org/wiki/Alan_Kay))

Wikipedia, Douglas Engelbart (2009, [http://en.wikipedia.org/wiki/Douglas\\_Engelbart](http://en.wikipedia.org/wiki/Douglas_Engelbart))

Wikipedia, Free Software (2009, [http://en.wikipedia.org/wiki/Free\\_software](http://en.wikipedia.org/wiki/Free_software))

Wikipedia, Gesture Recognition (2009, [http://en.wikipedia.org/wiki/Gesture\\_recognition](http://en.wikipedia.org/wiki/Gesture_recognition))

Wikipedia, Haptic technology (2009, [http://en.wikipedia.org/wiki/Haptic\\_technology](http://en.wikipedia.org/wiki/Haptic_technology))

Wikipedia, Ivan Sutherland (2009, [http://en.wikipedia.org/wiki/Ivan\\_Sutherland](http://en.wikipedia.org/wiki/Ivan_Sutherland))

Wikipedia, Lemur Input Device (2009, [http://en.wikipedia.org/wiki/Lemur\\_Input\\_Device](http://en.wikipedia.org/wiki/Lemur_Input_Device))

Wikipedia, Mac OS (2009, [http://en.wikipedia.org/wiki/Mac\\_OS](http://en.wikipedia.org/wiki/Mac_OS))

Wikipedia, MIDI controller (2009, [http://en.wikipedia.org/wiki/MIDI\\_controller](http://en.wikipedia.org/wiki/MIDI_controller))

Wikipedia, Musical Instrument Digital Interface (2009, [http://en.wikipedia.org/wiki/Musical\\_Instrument\\_Digital\\_Interface](http://en.wikipedia.org/wiki/Musical_Instrument_Digital_Interface))

Wikipedia, Open Source (2009, [http://en.wikipedia.org/wiki/Open\\_source](http://en.wikipedia.org/wiki/Open_source))

Wikipedia, Reactable (2008, <http://en.wikipedia.org/wiki/Reactable>)

Wikipedia, Theremin (2008, <http://en.wikipedia.org/wiki/Theremin>)

Wikipedia, Wii (2008, <http://en.wikipedia.org/wiki/Wii>)

Wikipedia, Xerox Alto (2008, [http://en.wikipedia.org/wiki/Xerox\\_Alto](http://en.wikipedia.org/wiki/Xerox_Alto))

# Dank aan

Mijn thesispromotoren, Liesbeth Huybrechts en Lieven Menschaert voor de bijstand in het voltooien van deze thesis.

Mijn ouders, Martin Standaert en Hildegard Willems voor het gebruiken van hun infrastructuur terwijl die infrastructuur soms op bepaalde vuilnisbelten leek.

Mijn vriendin, Goele Swenters, voor de oneindige steun en vertrouwen in het welslagen van deze thesis.

Mijn broer en zus, Roel en Nina Standaert, allebei muzikanten, voor het testen van bepaalde prototypes.

Tom Luyten, Danny Leen en Jim Bollansée voor de geslaagde projecten die we dit jaar voltooid hebben.

Alle vrienden en familie die niet té gek keken toen ik hen probeerde uit te leggen wat het ook al weer was dat ik deed met mijn thesis.

Alle enthousiaste proefpersonen en medestudenten.



