Gem

The helping hand

Report
Pieter Overgoor s128891
Meis Suker s126469

Table of Contents

Introduction..............................3
→ Abstract
→ Project Describtion

First Steps.................................4
→ Visiting Client
→ Forming a Team

1st Prototype............................5
→ Concept
→ Technology
→ Overview

Ideation.....................................6
→ Design Directions
→ Ideas

Chosen Direction.....................7
→ User: Nienke
→ 1st User Meeting

2nd Prototype..........................8
→ Concept
→ Technology
→ Overview
→ Form

Midterm Demo........................10
→ 2nd User Test
→ Demo Day

New Technology....................11
→ Options
→ 3rd User Test

3rd Prototype..........................12
→ Concept
→ Technology
→ Overview
→ Form

Final User Test.......................15
→ Feedback Nienke

Presentation...........................16
→ Poster
→ Movie

Final Demo Day......................17
→ Feedback

Future Concept......................18
→ Concept
→ Technology
→ Overview
→ Form

Reflection...............................19
→ Pieter
→ Meis
→ Teamwork

Appendices.............................20

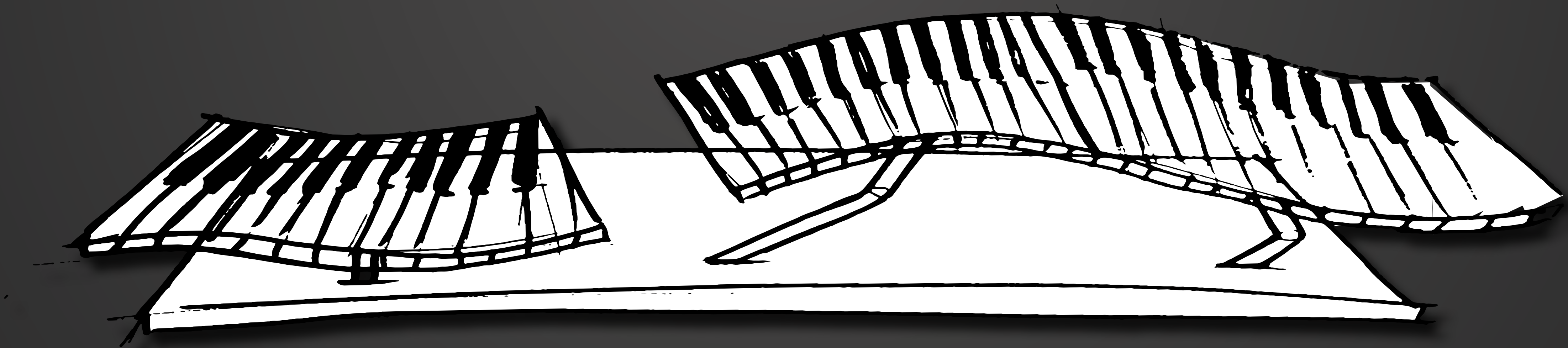
Introduction

Abstract

Gem is the outcome of a the project Adap-tive/able music by Meis Suker and Pieter Overgoor. It is a keyboard extension for Nienke, our user who misses part of her left arm. With Gem Nienke can play piano while using her right hand as well as her left arm. By using a gesture sensor, this extension ads a whole new dimension to playing piano. For the first time Nienke found her interaction with the piano not frustrating, but actually stimulating and practical.

Project Describtion

This project focuses on the development of new musical instruments for mentally and/or motorically impaired people. The challenge in this project is to combine the notion of new ways of controlling music with the extra challenge of designing for specific groups, which in turn may lead to a very innovative controller.



First Steps

Visiting Client

At the start of this semester we went to Amsterdam to meet our client; STEIM. There we were introduced to the nature of STEIM's activities as an introduction to the sorts of things we could be doing in the upcoming semester.

Interesting points were that they showed us that they could work with one explicit user, but also that there was a division between working with mentally handicapped as well as physically handicapped users. It came to our understanding that our main goal would be to create a new interaction between users with special needs and music(al instruments).

Forming a Team

After the meeting at STEIM we talked about our intentions for the upcoming semester. For the project we both had a very strong preference towards working with physical disabilities rather than mental disabilities.

This was mainly based on the idea that being able to communicate with our user group with as little restrictions as possible would help us make the project more concrete at an earlier stage. Also we both wanted this project to be different from the ones we had already had. We wanted to get beyond the conceptual phase rapidly and make something that really mattered to someone. In finding these similarities in our goals we teamed up.

1st Prototype
LDR sound

1. Concept

The first practical step for the project was to build an electronic circuit that generates sound. Both of us had no experience with electronics before, so it took some time to complete this easy seemingly task. The plan was to copy and rebuild an existing electric circuit and code but this did not work. While it was not our own code and circuit we could not find the error.

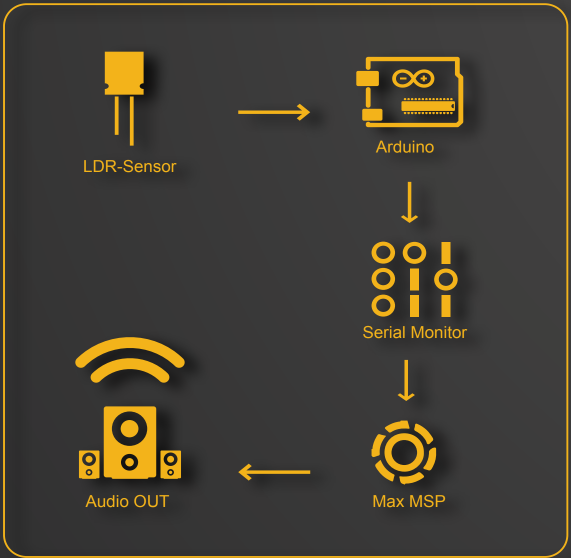
After that it was decided that it was better to build something easy from scratch. A Light Dependent Resistor (LDR) became the base to make an interaction whereby movement generated changes in sound.

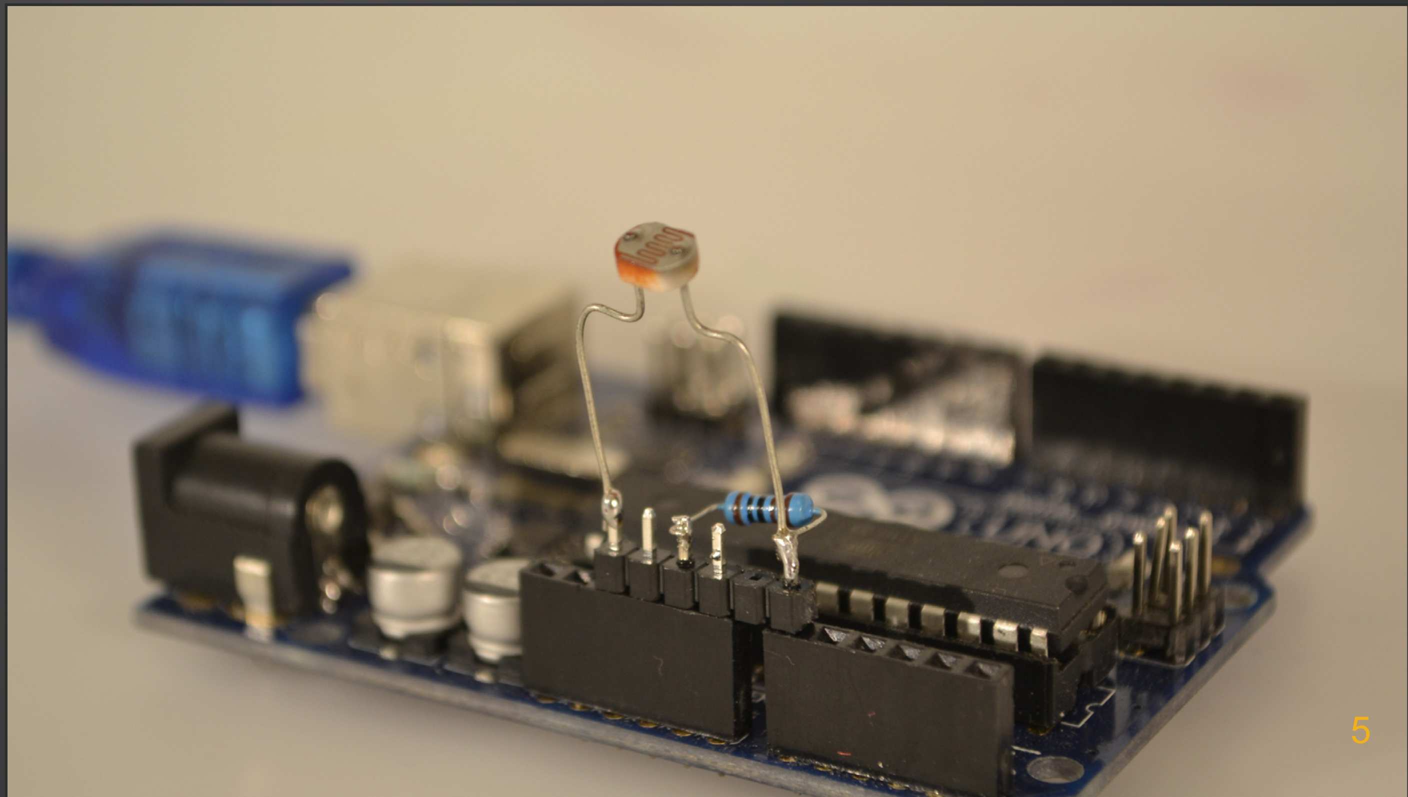
2. Technology

The values of the light intensity were mapped to usable values in the Arduino code and send to the serial monitor. A Max MSP patch was made with help from Rhys Duindam to read out these values and map them to the change of frequency of a single tone. This resulted in a change of sound when a hand is approaching the sensor. A resistor in parallel was used to stabilize the values of the sensor. Even though the sound was a bit annoying, it worked!

The Arduino code and can be found in Appendix A.

3. Overview





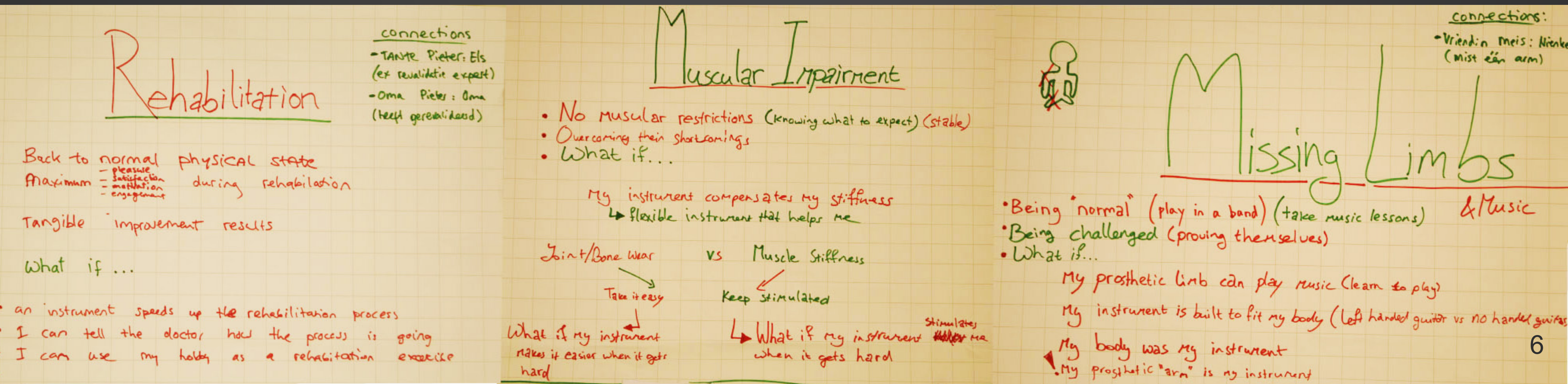
Design Directions

Ideation

Within physical limitations we found three main areas as we thought to be interesting to explore within the project. For each area we asked ourselves the same questions: what exactly is the limitation of this target group and what kind of instrument would be of good use to them? Second off we looked at how to get into contact with these groups, for practical reasons.

We thought of people who were going through a rehabilitation process. What if the instrument would lighten up this process and make it more pleasurable for the patient as well as others involved? Here we were particularly drawn to improving the experience of rehabilitation on a social level as well as on a physical level, mainly by tackling motivation through design.

We also looked into muscular diseases, could we somehow slow down the process these people were going through or otherwise make it easier on them? The third option we explored was designing for people who did not have all of their limbs. Could we somehow compensate for their handicap within this project? Here we had very straight-forward options right away; making the body an instrument, having prosthetics as instruments or adapting the instrument to the body.



Chosen Direction

User: Nienke

We found the options we saw in the field of missing limbs to be the most interesting to explore further, also because we already personally knew the perfect person to work with and for.

Nienke Dalstra was born with only one arm and has always wanted to learn to play the piano. When we gave her a call explaining the project and our offer to design a system that would enable her to learn to play the piano, she was more than enthusiastic. Another big plus for this option was that this seemed to be a relatively underexplored area in general.

1st User Meeting

Soon after contacting Nienke we met up with her and made the first steps. We had already established that she wanted to learn to play the piano. At the start of our ideation sessions we were contemplating working with prosthetics our making the body into an instrument somehow.

We left these ideas behind soon after talking to Nienke, though, as she argued that she was comfortable with her body the way it was and did not want us to change make changes to her. We found that she had a very valid point in that and realized that the innovation should not be sought in our user, but in the instrument itself.

2nd Prototype
'Gem' with buttons

When playing piano one usually uses one hand to play single notes and the other hand to play multiple notes at once; chords. Since chords are usually played with the left hand and Nienke only has a right hand, it was a natural step to look for compensation for the left hand.

We first thought of having extended buttons, one button being one chord. We started off with four basic chords and this was the central idea of the second prototype. We had two main goals fort his prototype: the first one was making it user test ready for Nienke, the second one was making it feedback ready for the midterm.

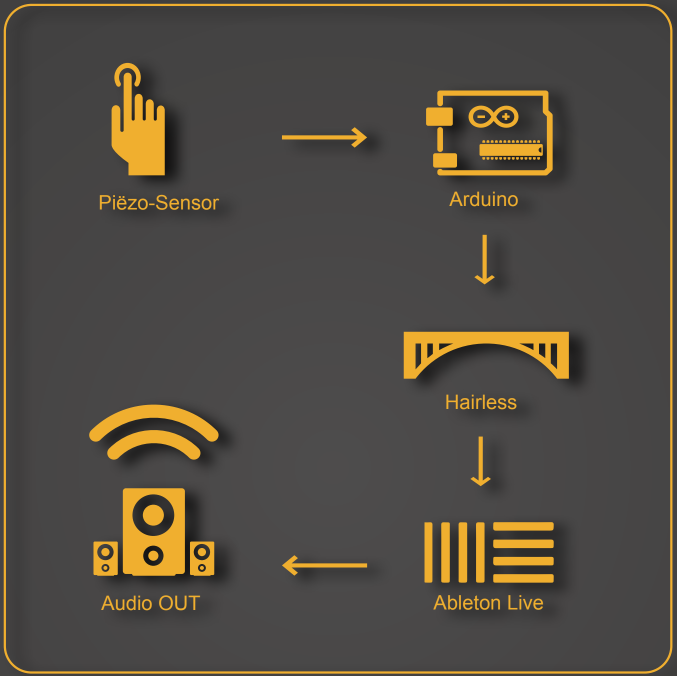
2. Technology

To make the buttons, Piëzo elements were used as pressure sensors. Piëzos are electronic components that can either make sound from a electric pulse or they can recognize (sound) waves and translate them to electric signals; the second function was used. The Pïezo elements were chosen above regular pressure sensors because a bigger surface needed to be covered instead of the small radius of the sensor.

In the Arduino code the values were compared to a threshold value and when higher a MIDI-message was send. Windows computers need an internal MIDI-bridge to send MIDI-messages so a MIDI-BRIDGE called Hairless was used to read the serial monitor and send these messages to Ableton. Here the MIDI-message is recognized and a piano sound is added to give the MIDI-message a piano sound.

The Arduino code can be found in Appendix B

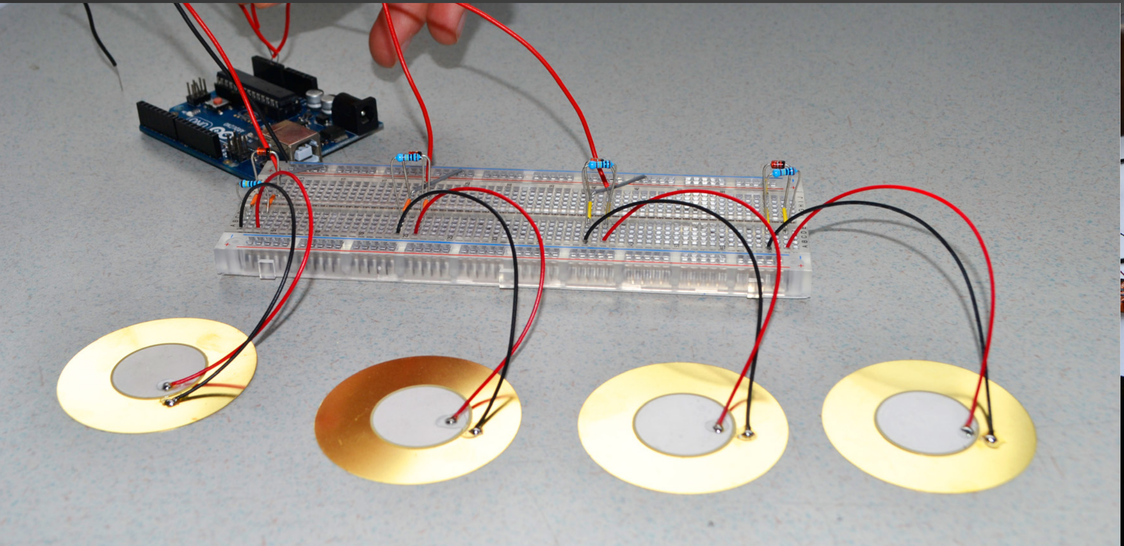
3. Overview



4. Form

Although we had one specific user to design for, we also kept into consideration that any outcome might be helpful for others in similar situations as well. In deciding on form and function we wanted to display this prominently. So for the first prototype we let the extension with the four buttons slide above the keyboard over wooden sticks. It would slide above the keyboard, because we thought that the missing arm length could compensated that way. Having it slide from right to left or the other way around was to display the possibility of making it adjustable under any circumstance to meet the preference of the user.

Furthermore in terms of the form we went for a look (in form and in colors) that was inspired by actual piano keys. Instead of choosing random forms, we thought staying close to the concept's core would be best to do. The actual interaction between the user and the extension would be in touching the buttons, so we also decided to use soft touchpads. The entire prototype, except for the touchpads, was made out of wood. It's skeleton was a product of hole-drilling, sawing and painting.



Midterm Demo

2nd User Test

First of all, after meeting with Nienke she was clearly enthusiastic about coming closer to actually playing piano. However, simply making the touchpads soft was not nearly good enough. The contrast between hitting the big buttons left was just too big with the gentle pressing of her right-hand fingers on the keyboard keys.

Also the range we chose was way too big. We realized then that the range we called “small” was based on the length and strength of our own arms. Nienke’s range is completely different and even though the extension was placed so that it would be closer to her arm, the movement from the first to the fourth button was not comfortable for her.

Demo Day

For the midterm Demo Day we mainly wanted to receive feedback on how to proceed given the feedback of our second user test and second prototype. We laid out cards with the options for proceeding with pushing buttons, swiping over buttons or a screen or using a gesture sensor.

Even though when we asked Nienke the same question she was very hesitant about the gesture sensor, a lot of people from the midterm Demo Day were in favor of this idea. We were curious about it as well, so we decided to try it.

New Technology

Options

In order to recognize gestures a new technology was needed. Several options could be used such as Kinect or Leap Motion. Kinect is better for longer ranges than our instrument. Leap motion works optimal in our range and it is pre-programmed to detect hands and arms. The only uncertainty was if it would detect Nienke's left arm. If this was the case, Leap Motion would be the best option. The sensor was bought to test this with Nienke.

3rd User Test

We went on the do the user test with Nienke to see if and how the Leap Motion device would react to her. Luckily, the two cameras read her left arm as a hand with all the fingers stuck together. So from this position of reading we could conclude that we could already start mapping different gestures with Nienke’s left arm.

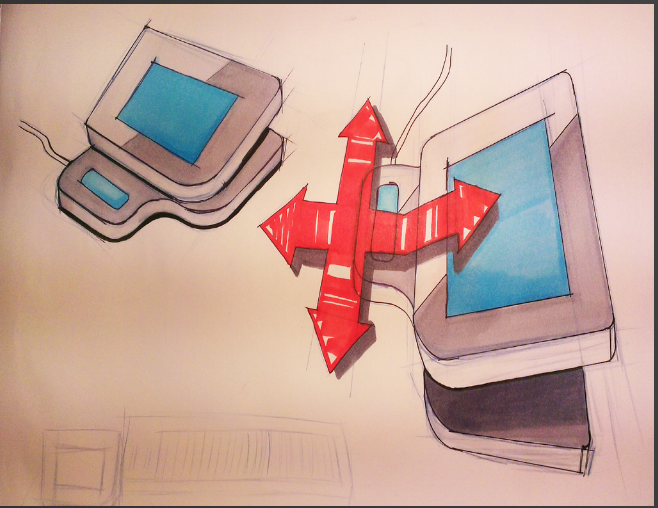
Looking back at her hesitance about motion sensors during the second user test, where she addressed that she thought it would feel too unnatural, this was a very crucial user test. After trying it she was very curious and eager to explore the possibilities through Leap Motion.

3rd Prototype
'Gem' with gestures

1. Concept

The concept we ended up with is a system that connects three components to one laptop; the keyboard, a visualization screen and the Leap Motion device. The user can look at the screen and see what kind of gesture to make in order to create a certain chord.

The gestures in this case are made with the user's left arm by moving up, down, left or right. A circle on the screen explains to the user at what position they are at and where they should move to hit the right chord, without ever touching a single thing physically. The field in which the user makes the gesture is quite small and therefore easy to be made comfortable and manageable.



2. Technology

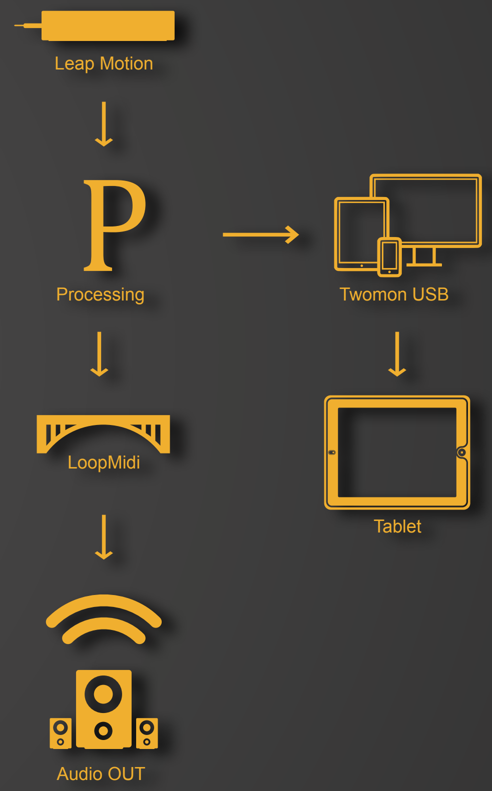
For the final prototype an app called Geco was bought that recognizes hand gestures and maps them to a MIDI-message to Ableton. This worked but it was too restricted for our purpose. After that it was decided to write our own code, which would give us more freedom and the possibility to make a visual for feedback. This visual would tell Nienke where her left arm is in the field of the Leap and which motion is required to play a specific chord.

The code was written in Processing. It was chosen to make a screen in the form of a table of three by three fields, this way a blank centre field could be used as resting point where no chord would be played. Four of the fields represent a chord of Nienke's favourite song called 'Ride' from Lana del Rey. The values from the Leap Motion were imported into processing with the Z-axis left out to make it 2D. The X-and Y values were used to make a moving ellipse through the fields. If a chord field was entered by the ellipse, the corresponding MIDI-message would be send to a MIDI-BRIDGE called LoopMidi which forwards it to Ableton.

For the visual, the same code with the fields was used. A background in the code was added to represent. The first version only displayed the chords, while the second gives a richer feeling in the interaction. Here a piano is displayed and the corresponding notes of the chord played. This looks good and it is also a learning opportunity for Nienke to see which notes she is actually playing. The visual is send through USB via TwoMon, an app that allows mirroring a screen. A tablet was bought to display this visual so the electronics of the final prototype consists of the Leap Motion sensor and the tablet.

The Processing code can be found in Appendix C

3. Overview



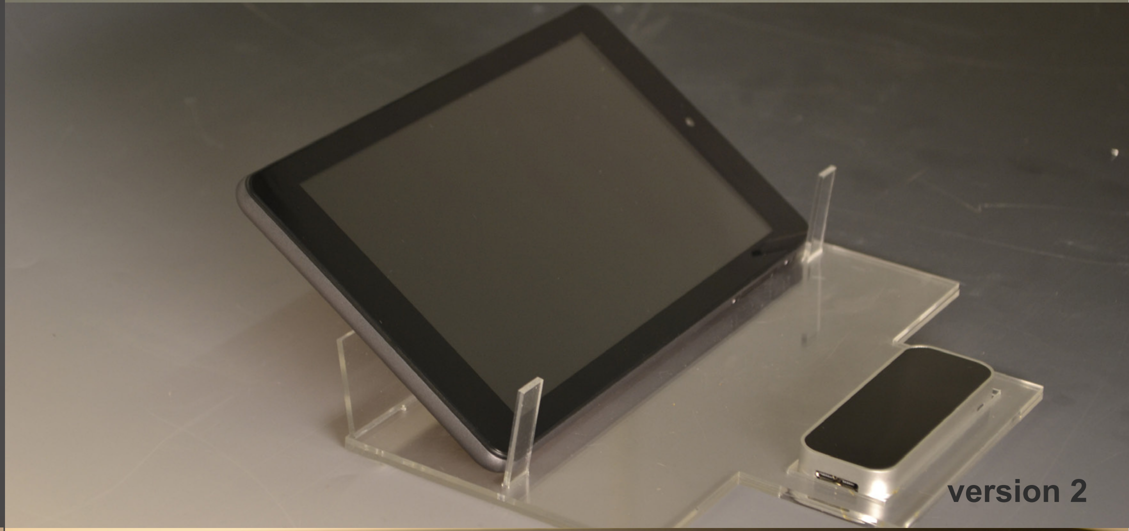
4. Form

For the form we wanted a product with elegance, dynamic features and a blend between modern design and classical elements. We found elegance in the gesture sensor, giving the interaction with Gem a magical and intriguing experience. By integrating round and asymmetric forms based on a piano wing, Gem captures the synthesis between classic elements (the shape) and modern design (the screen and the sensor). When we built the final prototype we used a combination of different iterations. We searched for materials that were smooth and complimentary to the shape of Gem and its setting.

Gem is smoothly black to match and empower the piano wing principle. After experimenting with a handmade wooden model and a laser cut model out of perplex we decided to combine them. We used layers of wood laser cut in the shapes needed and glued them all on top of each other, softening the edges and making the wood paint-ready. The size of Gem was an important feature to be considered. The screen size had to be big enough to display all the features needed. So Gem's size is chosen more or less around the size of the screen (a 17" tablet).

The screen is held up by a diagonally placed wooden stick much like a piano wing has. This was done for the comfort of the user while looking at the screen so they would not endure problems with lightning or having to bend forward too much.

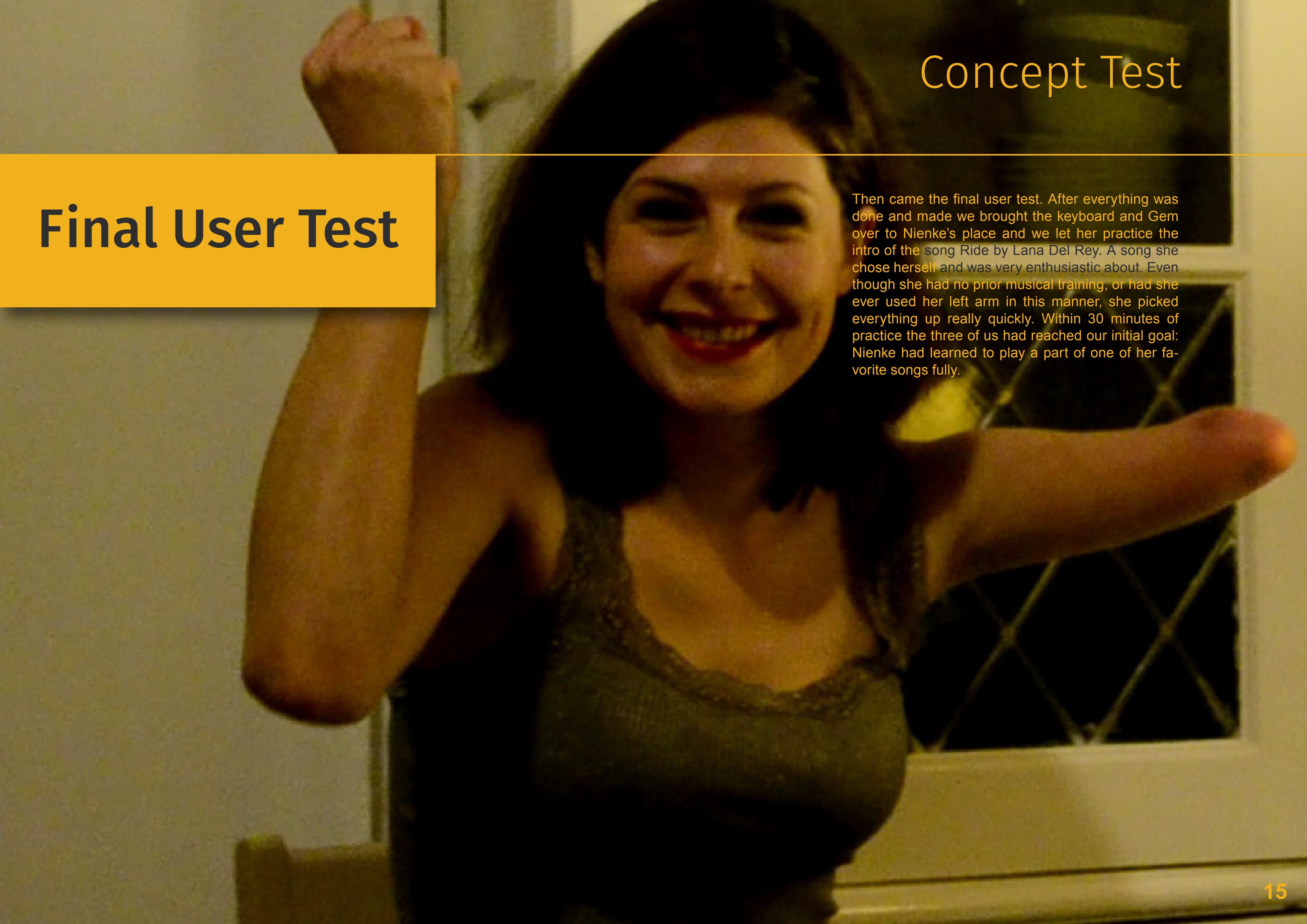
version 1



version 2

version 4

version 3



Final User Test

Then came the final user test. After everything was done and made we brought the keyboard and Gem over to Nienke's place and we let her practice the intro of the song Ride by Lana Del Rey. A song she chose herself and was very enthusiastic about. Even though she had no prior musical training, or had she ever used her left arm in this manner, she picked everything up really quickly. Within 30 minutes of practice the three of us had reached our initial goal: Nienke had learned to play a part of one of her favorite songs fully.

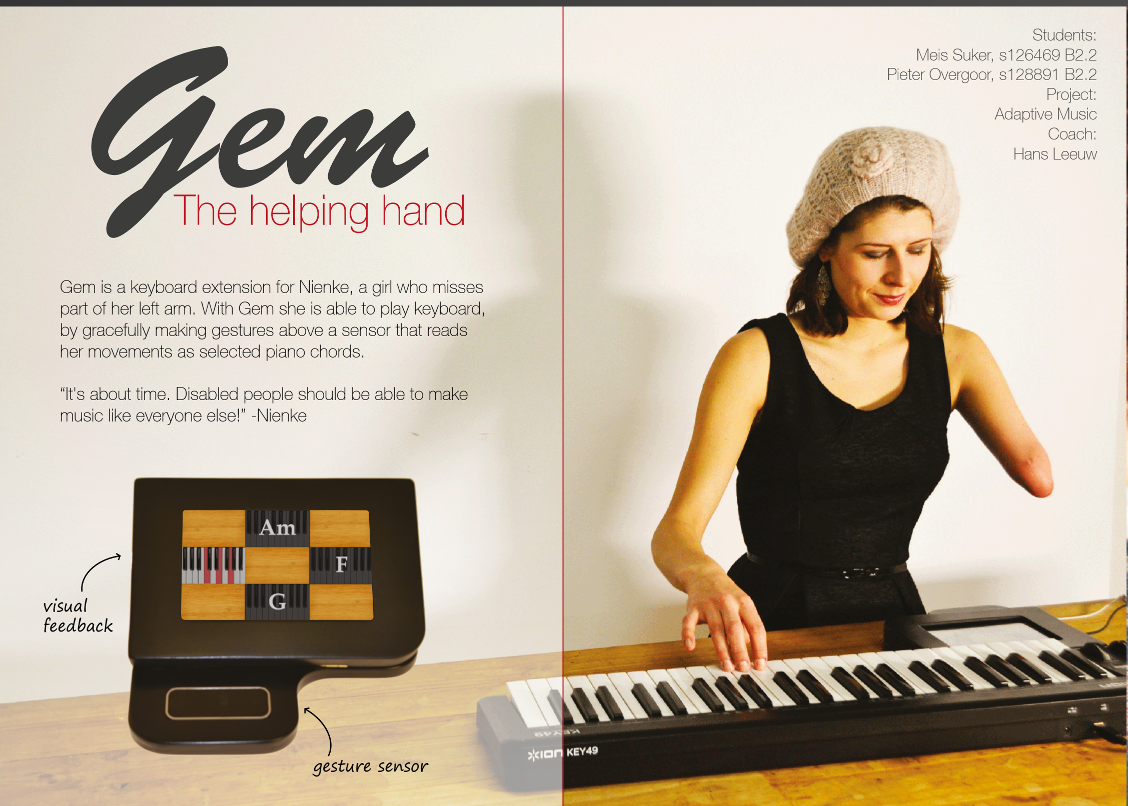
Presentation

Poster

After the final prototype was finished it was time to prepare for the Final Demo Days. The poster was made using the same form definitions as the final prototype; simplicity, elegance and technology versus classical The concept needed to be simple and understandable after scanning the poster. The elegance of the interaction is displayed in the title font and the composition, in which Nienke plays an important role. The contrast is brought by the use of the wooden table in contrast to the technology and the two different layers of scenario and explanation of the concept.

Movie

For the final demo day we decided to make a short movie showing our process with Nienke. We had filmed and photographed every meeting we had with her and ended up with material from 5 different meetings. In showing the process we mainly focused on showing Nienke's struggle to play piano at the start and her ability to actually learn a song she wanted to learn towards the end by using Gem. We also wanted the video to function as an introduction to the project for people outside of the project.



Final Demo Day

During the final demo day we had a setting with a poster, the video, a keyboard and the device itself. The general idea was of course to present the result, but also see the reaction of people outside of our user group.

The main thing we learned there was that people were very enthusiastic and that we had a clear presentation. Most of the constructive feedback we received was about the future development. We also had extra leaflets that we showed when people asked us about this with a future interaction possibility, displaying the record function mention earlier in this report. All in all we ended up reaching our goal and the reactions we hoped for on these two Demo Days.

Future Concept

1. Concept

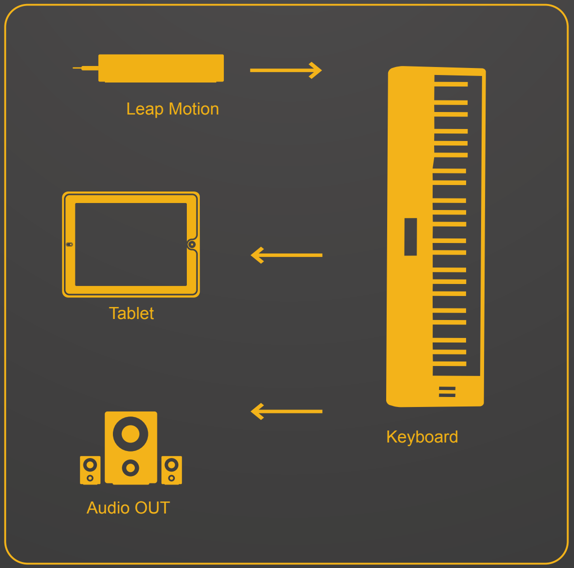
Now all the three components (midi keyboard, leap motion and tablet) find their source in the laptop. In the future it would be great to only have interaction between Gem (the screen and the gesture sensor) and the keyboard. However, leap motion can only be mapped through programs on the computer. So an alternative motion sensor might be a solution. It would be best if in the end we would not need a laptop anymore and Gem could be an independent product.

To expand the freedom of the musician and stimulate their learning curve, we suggest that chords can be recorded from keyboard to Gem and self-chosen gestures can be matched to them.

2. Technology

In terms of technology for future development we would be facing the challenge of integrating all the external programs we now use and all the connections to the laptop into a compact component of Gem itself. As mentioned before, we could either be looking at different sensors, but also we could see how far we can get with an internal motherboard (such as an Arduino as we used for our first prototype). Also this can be further developed to have more than just piano sounds come out. It would be just as easy to customize any other sound into Gem if the user would want that.

3. Overview



4. Form

When it comes to future form we would like to stick to the same goals we set out for the final prototype. The physical metaphors for the piano wing and the colours really seem to work. However, we might go on to exploring the possibilities with 3D-printing. This would spare a lot of time and possible mistakes are easily corrected, At this point the final prototype only looked adjustable, but in fact it was not foldable or adjustable in height. Very important in the further development of Gem is making it adjustable and usable for more people in the same situation as Nienke.

Reflection

Teamwork

The main reason we decided to work together was because of sharing the same mindset and goals for this semester and project. We wanted to reach a certain level of depth that we felt we couldn't reach on our own. We actually succeeded in doing so, since we got to work through multiple iterations properly in different fields. Both form as well as technology were explored thoroughly. On our own we could not have divided the workload as we did (Meis mainly focusing on the form and Piet mainly focusing on the technology) and so we could have never learned as much or come as far as we have now.

Also in doing so we learned from each other and about ourselves. While Piet is very good with punctuality, planning and motivation Meis is good with communication and documentation. From there on we started developing further in the weaker points we still wanted to get better at. We did everything in deliberation which was helpful in stimulating each other, but also in still learning and developing second-hand from each other's work. This happened because we had to seriously think about each other's parts and make decisions and form opinions about each other's part during small and big brainstorm sessions.

Pieter

For me this project was a learning experience in the fields of technology, planning and teamwork. I never dived deeply into technology so I wanted to explore it and to see to what extend I can grow. Electronics appeared not interesting for me, while programming appealed to me a lot so I will continue developing this skill. Also in planning I developed; while we both had a lot of external activities we needed to have a tight schedule in order to work efficient and goal orientated. This rewarded us with time and a relaxed attitude towards the end as we knew we were on the right track. In teamwork I learned to be always honest, even when you are friends that work together. Having high standards and goals while working together with somebody else can give friction, which leads to stress and inefficiency when not shared.

Meis

At the start of this project all I could hope for was to reach deeper than I had ever done in any of the three previous projects I had done. This time I wanted to learn more about myself and give form to my future development within Industrial Design. At the start I really wanted to explore the competency area Integrating Technology. However, after diving loosely into it, Pieter and I decided to split the project into two pieces. Pieter dove into the technology and I into the form. It was then that I realized that I wanted to develop further in physically exploring Form and Sense rather than Integrating Technology. Through watching what Pieter did I could learn about IT second-hand, though. Also within this project eventually a great contrast between me and Pieter was in punctuality and dealing with stress. During this project I learned to push myself more and be better in dealing with insecurity.

Appendix A
Arduino code

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
   sensorValue = map(sensorValue, 0, 600, 0, 100);

  // print out the value you read:
  Serial.println(sensorValue);
  delay(5);        // delay in between reads for stability
}
```

Appendix B
Arduino code

```
//include MIDI Library
#include <MIDI.h>

//analog pins used on Arduino
int sensor1 = A1;
int sensor2 = A2;
int sensor3 = A3;
int sensor4 = A4;
int treshold = 100;

MIDI_CREATE_DEFAULT_INSTANCE(); //create a MIDI
Instance


void setup()
{
   Serial.begin(115200); //set Baudrate

}

void loop()
{
 //create integers for the sensorvalues
   int sensor1val = analogRead(sensor1);
   int sensor2val = analogRead(sensor2);
   int sensor3val = analogRead(sensor3);
   int sensor4val = analogRead(sensor4);


   if (sensor1val > treshold)          // If sensor 1 value is
higher then treshold then
   {
                         //Cmaj chord
    MIDI.sendNoteOn((0x30),100,1);    //C2//(hex dec pitch,
velocity, channel)
    MIDI.sendNoteOn((0x34),100,1);    //E2//(hex dec pitch,
velocity, channel)
    MIDI.sendNoteOn((0x37),100,1);    //G2//(hex dec pitch,
velocity, channel)
      delay(450);                  // Wait for a second
     MIDI.sendNoteOff((0x30),0,1);     // Stop the note
```

```
    MIDI.sendNoteOff((0x34),0,1);
    MIDI.sendNoteOff((0x37),0,1);
  }
  if (sensor2val > treshold)          // If sensor 2 value is
higher then treshold then
  {                         //F chord
    MIDI.sendNoteOn((0x35),100,1);    //F2v
    MIDI.sendNoteOn((0x39),100,1);    //A2
    MIDI.sendNoteOn((0x3C),100,1);    //C3
     delay(450);                   // Wait for a second
    MIDI.sendNoteOff((0x35),0,1);     // Stop the note
    MIDI.sendNoteOff((0x39),0,1);
    MIDI.sendNoteOff((0x3C),0,1);
  }

     if (sensor3val > treshold)       // If sensor 3 value is
higher then treshold then
  {                          //Em chord
    MIDI.sendNoteOn((0x34),100,1);    //E2
    MIDI.sendNoteOn((0x37),100,1);    //G2
    MIDI.sendNoteOn((0x3B),100,1);    //B2
     delay(450);                   // Wait for a second
    MIDI.sendNoteOff((0x34),0,1);     // Stop the note
    MIDI.sendNoteOff((0x37),0,1);
    MIDI.sendNoteOff((0x3B),0,1);


  }
  // last sensor chord
     if (sensor4val > treshold)       // If sensor 4 value is
higher then treshold then
  {                          //Gmaj chord
    MIDI.sendNoteOn((0x37),100,1);    //G2
    MIDI.sendNoteOn((0x3B),100,1);    //B2
    MIDI.sendNoteOn((0x3E),100,1);    //D3
     delay(450);                   // Wait for a second
    MIDI.sendNoteOff((0x37),0,1);     // Stop the note
    MIDI.sendNoteOff((0x3B),0,1);
    MIDI.sendNoteOff((0x3E),0,1);
  }

}
```

Appendix C 1
Processing code

```
//import libraries
import com.onformative.leap.*;
import com.leapmotion.leap.*;
import com.leapmotion.leap.Controller.PolicyFlag;
import themidibus.*;

//Start the library
LeapMotionP5 leap;
MidiBus myBus;

//Image as BG
PImage bg;
PImage bgC;
PImage bgF;
PImage bgG;
PImage bgAm;

//Timer ints
int savedCTime;
int savedFTime;
int savedAmTime;
int savedGTime;
int totalTime = 2000;

//borderlines
int Z1 = 200;
int Z2 = 400;
int X1 = 341;
int X2 = 682;

//ellipse size
int ES = 200;

//Chord booleans
boolean C1 = true;  //starts checking if you are in area
boolean C2 = false;  //plays note for time given at totalTime
int
boolean F1 = true;
boolean F2 = false;
boolean Am1 = true;
boolean Am2 = false;
boolean G1 = true;
boolean G2 = false;

void setup() {
  size(1024, 600, P3D);          //size screen and P3D Render
  bg = loadImage("unpressed.png"); //Load default BG im-
age
  bgC = loadImage("C_color_red.png"); //Load C BG image
  bgF = loadImage("F_color_red.png"); //Load F BG image
  bgAm = loadImage("Am_color_red.png"); //Load Am BG
image
  bgG = loadImage("G_color_red.png"); //Load G BG image
  fill(0);                    //initial fill colour ellipse
  stroke(0);                  //initial stroke colour ellipse
  leap = new LeapMotionP5(this); //setup the leapmotion
  leap.getController().setPolicyFlags( PolicyFlag.POLICY_
BACKGROUND_FRAMES );
  MidiBus.list();              //setup midi
  myBus = new MidiBus(this, "loopMIDI Port", "loopMIDI
Port", "loopMIDI Port");
}

void draw() {

  //load default BG
  background(bg);

  //load C BG
  if (C1 == false) {
    background(bgC);
  }

    //load F BG
  if (F1 == false) {
    background(bgF);
  }

    //load Am1 BG
  if (Am1 == false) {
    background(bgAm);
  }
```

Appendix C 2

```
//load G BG
if (G1 == false) {
  background(bgG);
}

//start tracking hand with leapmotion using HZ and HX for
the positions
for (Hand hand : leap.getHandList ()) {

  PVector handPosition = leap.getPosition(hand);
  float HZ = handPosition.z;
  float HX = handPosition.x;
  ellipse (HX, HZ, ES, ES);
  fill(180, 200);

  //---------------------------------------------------------------//

  //starts looking if you are on the C chord field
  if (C1 == true) {
    if (HZ>Z1 && HZ<Z2 && HX<X1) {
      C2 = true;
      C1 = false;
      savedCTime = millis();          // set savedTime to millis
       int passedCTime = millis() - savedCTime; //calculate
passed  time
      if (passedCTime > totalTime) {
        C1 = true;
      }
    }
  }

  //plays the C chord note once and turns itself off
  if (C2 == true) {
    myBus.sendNoteOn(1, 48, 100);
    myBus.sendNoteOn(1, 52, 100);
    myBus.sendNoteOn(1, 55, 100);
    C2 = false;
  }

  //---------------------------------------------------------------//
```

```
    //starts looking if you are on the F chord field
  if (F1 == true) {
    if (HZ>Z1 && HZ<Z2 && HX>X2) {
      F2 = true;
      F1 = false;
      savedFTime = millis();          // set savedTime to millis
       int passedFTime = millis() - savedFTime; //calculate
passed  time
      if (passedFTime > totalTime) {
        F1 = true;
      }
    }
  }

  //plays the F chord note once and turns itself off
  if (F2 == true) {
    myBus.sendNoteOn(1, 53, 100);
    myBus.sendNoteOn(1, 57, 100);
    myBus.sendNoteOn(1, 60, 100);
    F2 = false;
  }

  //---------------------------------------------------------------//

  //starts looking if you are on the Am chord field
  if (Am1 == true) {
    if (HZ<Z1 && HX>X1 && HX<X2) {
      Am2 = true;
      Am1 = false;
      savedAmTime = millis();          // set savedTime to millis
     int passedAmTime = millis() - savedAmTime; //calculate
passed  time
      if (passedAmTime > totalTime) {
        Am1 = true;
      }
    }
  }

  //plays the Am chord note once and turns itself off
  if (Am2 == true) {
    myBus.sendNoteOn(1, 57, 100);
```

Appendix C 3

```
    myBus.sendNoteOn(1, 60, 100);
    myBus.sendNoteOn(1, 64, 100);
    Am2 = false;
  }

  //----------------------------------------------------------------//

    //starts looking if you are on the G chord field
  if (G1 == true) {
    if (HZ>Z2 && HX>X1 && HX<X2) {
      G2 = true;
      G1 = false;
      savedGTime = millis();          // set savedTime to millis
       int passedGTime = millis() - savedGTime; //calculate
passed  time
      if (passedGTime > totalTime) {
        G1 = true;
      }
    }
  }

  //plays the G chord note once and turns itself off
  if (G2 == true) {
    myBus.sendNoteOn(1, 55, 100);
    myBus.sendNoteOn(1, 59, 100);
    myBus.sendNoteOn(1, 62, 100);
    G2 = false;
  }

  //----------------------------------------------------------------//

 //checks if your not in a chord field and therefor sets chord-
field tracking(C1,F1,Am1 & G1) on again
     else if (HZ<Z1 && HX<X1 || HZ>Z2 && HX<X1 || HZ>Z1
&& HZ<Z2 && HX>X1 && HX<X2 || HZ<Z1 && HX>X2 ||
HZ>Z2 && HX>X2) {
      C1 = true;
      F1 = true;
      Am1 = true;
      G1 = true;
      //C Chord fff
      myBus.sendNoteOff(1, 48, 100);
      myBus.sendNoteOff(1, 52, 100);
      myBus.sendNoteOff(1, 55, 100);
      //F Chord off
      myBus.sendNoteOff(1, 53, 100);
      myBus.sendNoteOff(1, 57, 100);
      myBus.sendNoteOff(1, 60, 100);
      //Am Chord off
      myBus.sendNoteOff(1, 57, 100);
      myBus.sendNoteOff(1, 60, 100);
      myBus.sendNoteOff(1, 64, 100);
      //G Chord off
      myBus.sendNoteOff(1, 55, 100);
      myBus.sendNoteOff(1, 59, 100);
      myBus.sendNoteOff(1, 62, 100);

    }
  }
}
```