



MISSION

Project Report by:
Joca van der Horst
Bart Leeuwenburg
Linde van Beers

Coach:

MISSION

Project Report by:
Joca van der Horst
Bart Leeuwenburg
Linde van Beers

Coach:



This report shows the process of creating Massine. It is part of the group music improvisation system (GMIS) project, which has the goal to explore the possibilities of system design using the inspiration from music inspiration. Massine is a tool for dancers to influence the musical instruments around them by controlling sound effect filters and special movements that trigger an effect on another instrument. This is demonstrated with movements that change the scale on a snare instrument called WOodd.

Next to the process, the report shows the specifics of the latest iteration of the prototype and concept from different perspectives. For example technology, form & and senses and users. An important point of improvement for the prototype was the mapping of the sensor, in order to give a nice sounding feedback to expressive movements.

CONTENTS

| | |
|------------------------------------|----|
| Abstract | 4 |
| Contents | 5 |
| Introduction | 6 |
| Process | 8 |
| 3d quartile | 8 |
| Mid-Term Demo Day | 12 |
| Working together with another team | 13 |
| Movement recognition | 14 |
| Testing | 20 |
| Design | 26 |
| Functions | 26 |
| Form and Materials | 27 |
| Technology | 28 |
| Business | 35 |
| Conclusion | 38 |
| Reflections | 40 |
| Joca van der Horst | 40 |
| Linde van Beers | 44 |
| Bart Leeuwenburg | 47 |
| Appendices | 52 |
| Appendix A | 52 |
| Appendix B | 62 |
| Appendix C | 64 |
| Appendix D | 66 |
| References | 74 |



At the department of Industrial Design there is a focus on design for systems. In the group music improvisation system (GMIS) project we explore the possibilities in this area by looking to a field with a long tradition in interaction and connection between people: music improvisation.

We chose this project because of our connection with music, we play instruments ourselves. Another reason to choose this project has an explorative goal, having the freedom to define our own context and way of working. The context we had in mind was a small improvisation session, or a concert in an intimate setting with a small audience.

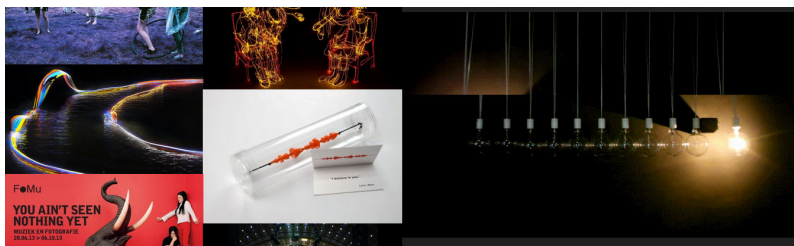
In this report we show the process of creating Massine, a product to make improvisation more accessible by giving people the ability to influence music with dance movements. We focus on what we have learned, which challenges arose during the creation of the prototypes and which key decisions made Massine the concept it is now.



PROCESS

3rd quartile

During the first meeting, the coach suggested to work together with two groups for a few weeks to start up the project. Everyone liked the idea of working together and it was used to accelerate our idea generation. With both groups, a mood-board -> foto was made to get all the thoughts organized. After this, the groups did some jam sessions on the recommendations of Bart Hengeveld. Instantly the groups noticed that the jam sessions where very helpful. It helped to get to know each other, but also to get into the interaction of music. After the session, there were a lot of fresh ideas during the brainstorm sessions.



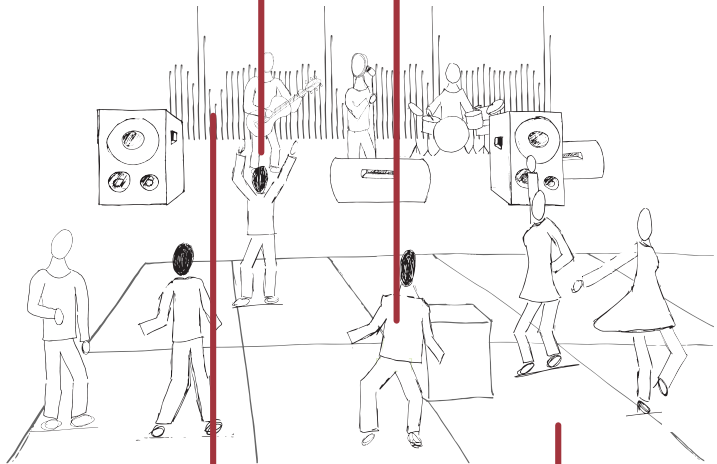
In the second week, after exploration and brainstorming, the group came up with 4 different concepts. As Bart Hengeveld also told his students; prototypes should be make as quick as possible. This way, all the possibilities could be explored. Following this approach of ideazation by making, the group came with the following 4 prototypes.

Conductor's wand

Every conductor has a baton. With a new kind of baton, which measures the acceleration and displacement at the tip, sound effects can be created and through this the music can be influenced directly.

Dancing with objects

When dancing around an object, this will influence sound frequencies. The closer to an object a person is, the more interesting frequencies will emerge. This might invite people to interact with objects around them in a surprising way.



Heartbeat sensor

An orchestra copies the emotions and feelings that the conductor wants them to display. This concept visualizes the conductor's energy as light effects on a big screen, by measuring his heartbeat. This way, the audience might feel more connected to the conductor.

Floorstages

There are several mini-stages in the crowd. The amount of people on and location of such a stage has influence on the way the music sounds. A combination of the stages gives different sound effects. This might create a playful interaction with musicians and the audience.

The conductor baton.

The baton-group managed to make a quick prototype as soon as the second jam session in week 2. Linde played the violin with the accelerometer taped to her hand. Every time there was a peek in the data, a drum sample was played. This way, every time she went from back to forth with the bow, a click was played. The result wasn't a stunningly interesting musical performance, but it was a first step towards measuring place in space, and link this to music.



Throughout the following weeks, the groups worked towards the cross-coach session. This was a sort of mini Demo Day, where students could show their concepts and get feedback from other coaches. The conductor-group experimented with the accelerometer, and figured out how it exactly worked by analyzing the data. The group also tried to link the data to different effects, instead of playing sound samples.

Fortunately, the group got it working just before the cross-coach session. People could hold a foam stick with the accelerometer in it, and influence the effect on the music by moving it up and down. This way a nice experience was created, an insight of what could be possible with the concept. This prototype was also very useful to get feedback on. Getting valuable feedback was most important at this point, because there was decided that 2 of the 4 concepts had to be chosen before the Mid-Term Demo Day.

After a lot of discussion and voting, the group decided that the conductor baton and the dancing with objects where the two concepts with the biggest potential. This way, there were two concepts that were quite different (one static and one dynamic), but at the same time both working with dance. Everyone spoke their heart out what concept they liked most, and so the Magicwand group emerged: consisting of Joca van der Horst, Bart Leeuwenburg and Linde van Beers.



Mid-Term Demo Day

On the Mid-Term Demo Day, the feedback on the first iteration of the concept focused on two aspects: the form and the interaction. The coaches recommended to do some extra research on the form. The Mid-Term prototype was designed with the shape of a conductor's baton in mind. The group got the feedback to have a look at projects from the past years that were designed like drumsticks or had a bigger form factor. Another recommendation was to experiment with the center of gravity of the object. By putting this in an unexpected place it is possible to create a new kind of interaction with it.

The feedback on the interaction was that the prototype was too complicated using a wide set of gestures to change sound filters and fix values. This could be simplified. A suggestion was to research recognition of movement patterns to activate certain functions and also do tests with creating sound instead of influencing the sound of others.

After the Demo Day this feedback was used to set up a series of user tests to improve the concept and make decisions for the creation of the second iteration.



Working together with another team

Since the project was about systems interaction, There had to be found a team to link our product to. After looking into several of the other team's activities, WOodd appeared to be the best project to work together with, as their project goal connected very well with that of Massine: making improvisation more accessible to inexperienced musicians.

Their product existed of a guitar-like instrument with 8 strings that was always tuned perfectly, but could change between scales. They were looking for a product to control their changes in scales, and our product needed a product to influence.



The collaboration is as follows: a dancer using Massine can make certain trigger-movements that send a signal to WOodd. WOodd will then change to a scale, depending on the movement that has been made. The system that emerged gives an opportunity for accessible improvisation to two parties: a budding musician and a dancer, who usually would not take part in influencing the music at all.

Movement recognition

For the connection with the WOodd project it was required to recognize emotion in movement, to make Massine give a command for a matching scale on WOodd. The first approach was to determine the emotion of the movement by measuring the average acceleration and displacement of every movement. The hypothesis was that a movement that symbolizes happiness would have a big acceleration and displacement, while a movement that shows sadness would have a low displacement and acceleration. Looking at this data the system could send a command for a major or a minor scale.

This caused some problems though. Calculating an average depends on multiple movements on a long timespan. From a user perspective there is not a directly visible relation between a certain movement and the change of scale on the other instrument. Another issue was the possibility that Massine would change the commands continuously and at a high pace. The WOodd system would break if it needs to change the scale every other second. From the perspective of playing music this situation negatively influences the sound as well.

Trigger movements

To solve these problems a new method to control WOODD was developed, select movements that trigger a scale. By doing this there is a clearer relation for the user between a movement and the effects of it on the other instrument. By choosing the right parameters in the sensor data it is also possible to avoid the situation where the scale is continuously changed by falsely recognized movements.

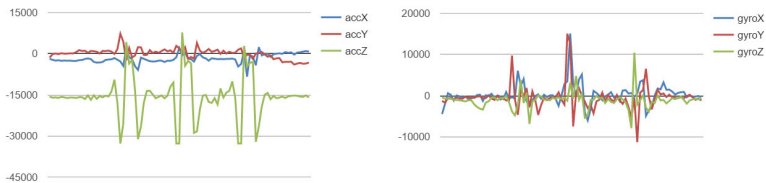
An important factor is the balance between the 'uniqueness' of a movement, avoiding false positives, and the intuitiveness of making this movement. A series of user tests were done by the group to find out what kind of movements are typical for music with a strong happy, sad or other emotion. A set of these movements was selected and the group did research on which movement could be easily recognized using a model based on sensor data. Out of these set of movements four were selected. In the Arduino source code in Appendix A it is possible to see the code for recognizing the movements. In the following sections the process of choosing the movements and ways to recognize them out of sensor data is shown.

Jump

The jump is a typical movement that people do when dancing on happy music. This movement is chosen to trigger a major scale on WOODD, because this scale is commonly associated with happy music.



When modeling a jump to raw accelerometer and gyroscope data, measurements show a distinct pattern in the Z-axis of the accelerometer. A jump is characterized by spikes in the value that are extremely low and high within a short timespan. This pattern is easily recognizable in the graph that shows the sensor values of 3 subsequent jumps.

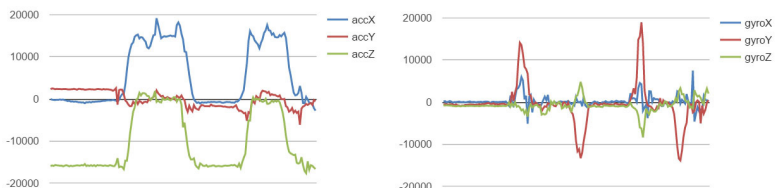


Vertical

A harmonic minor scale on a snare instrument results in music that sounds oriental. Holding the Massine vertically, resembles gestures common in the eastern culture to greet people and to dance.

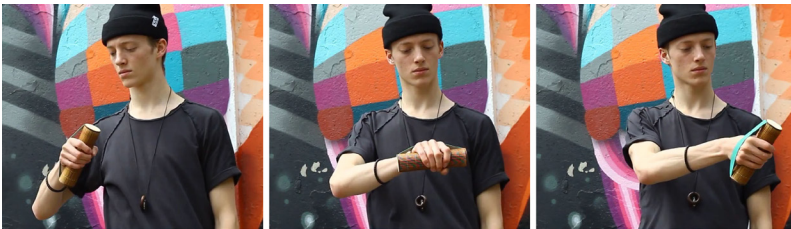


In the graph the peaks of the Y-axis of the gyroscope shows the moments that Massine is turned sideways and back. The peaks of the X- and Z- axis of the accelerometer show when the orientation of the system. Instead of recognizing this movement on raw data, it was decided to make use of the AHRS algorithm that combines the gyroscope and accelerometer to calculate the orientation. This was the most reliable way to recognize that Massine was hold sideways at 90 degrees.

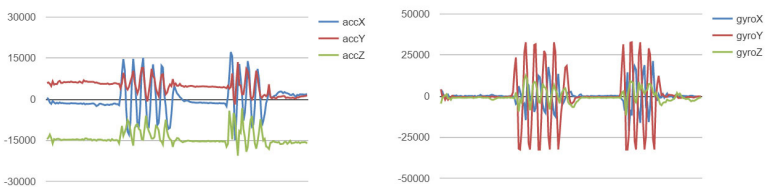


Wave

By repeatedly moving the left side of the Massine up and down (and thus moving the right side down and up), a person makes a movement that resembles a wave. This movement also has a funky aspect, which connects with the blues scale on WOodd.

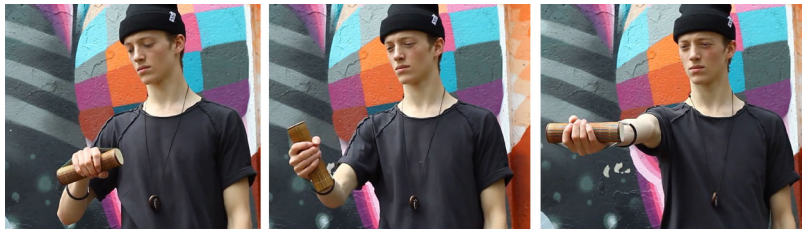


By repeating this movement a user creates a pattern in the sensor data that is unique for this gesture. Every axis shows a similar pattern of peaks. In the model to recognize this movement the Y-axis of the gyroscope and the X-axis of the accelerometer are chosen, because the pattern for this does not overlap with other movements.

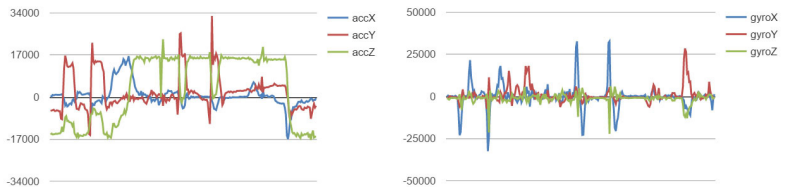


Turning Massine upside down and opening hand palm

The open hand palm is a gesture associated with fragility and looking for a trust. The sound of a minor scale is in line with this because this scale is usually connected with emotions like sadness and doubt.



Using one or two variables to detect this gesture wasn't enough. Tests showed that the system confused this movement with a jump. To avoid this the following detection method was used: The Z-axis of the accelerometer needed to be positive, indicating that the Massine was hold upside down, which is needed to open a hand palm. Opening up the hand palm caused a peak in the X-axis of the gyroscope and the Y-axis of the accelerometer, as can be seen in the graph.



Testing

At various stages during the design process, the product was tested both formally and informally.

First prototype

Testing started when the first working prototype was finished, just before the Mid-Term Demo Day. At this stage the prototype existed of a stick with the technology embedded at the top. During a cross-coaching session several people used it. From the feedback on and observations of this informal test it was evident that the mapping was not intuitive enough yet, as moving cancelled out the sound instead of adding to it, which resulted in that people actually felt discouraged to move.

Besides, the technology existed of only an accelerometer at this point, and acceleration wasn't something people were naturally in control of, like they are of speed and displacement, with which they tried experimenting. It also turned out that people moved very rhythmic, while slow tilting gave a more desirable effect to the music.

Altogether, the user experience was unsatisfying, as people couldn't interact with the product intuitively.

In order to decide what formgiving of the product and properties of the product would trigger a desirable movement, another test had to be done. This test was conducted with a series of shapes with different qualities: big and small, heavy and light, loud and silent, etc. The test-subjects were asked to hold all the objects, and choose the 4 they would most like to use to influence music. Afterwards, they were asked to act out some pieces of music with these objects. Afterwards, the way people moved to music with different emotions, and how the object they used affected this was analysed.



The most interesting results were found in the dancing-part of the test. It was clear that whenever people were moving with conductor-like sticks, the movement was very rhythmic, sudden, and not very diverse. When using

more organic shapes, however, the movement became more fluent and diverse. This was especially clear with the cushion-shape. People mentioned especially that they liked that it was somehow attached to their hand, which gave them the idea that they could move more freely and naturally. A negative quality of this shape was that people were mostly confused at its shape (although it also caused a certain attraction, which was also evident at the Final Demo Day, where it was presented as well), and that it wasn't very comfortable to hold. Objects that scored higher on this were the cylinder-shaped ones. Some mentioned in particular that they liked the grip of the stick wrapped in wire, because it had a clear structure. Furthermore, all heavy objects got instant negative feedback; people saw dance as something 'light and quick' and therefore felt the object should be light as well.



From the feedback the final design was constructed: a wooden cylinder with a laser-cut pattern to suggest a grip, with an elastic band to attach it to your hand. It combines qualities that people liked in the several objects and was aesthetically pleasing as a bonus. The diameter of the cylinder was determined through a test with a few paper mock-ups (including a representation of the elastic band).

The test-subject held all of them and chose the one they felt most comfortable holding. Both males and females, with different hand sizes, took part in the test. The great majority preferred a diameter of 4,5 cm.

When the prototype was finished, it was evident that it invited people to touch and hold it, as everyone who walked by our table asked to hold it. There was a positive response to how it felt and looked.



Professional dancer

The final user test took place while making the video for the final Demo Day. Here a professional dancer used the prototype and evaluated it.

The feedback was mostly positive: The dancer liked the fact that he could give something back to the musician by influencing their instrument. Also, he was positive about the design of the product and about the way it had to be held.

As for dancing with it, he did notice that his dance was influenced a lot by the product, but he didn't necessarily see this as a bad thing: He discovered new movements and broadened his perspective in this way. About the trigger moves he did have a remark: he felt that they might be too easily initiated without meaning to. This would have to be tested more elaborately, as the dancer wasn't using a working prototype, and therefore it could not be evaluated whether the signals were triggered unconsciously. The dancer did feel that the movements were easily incorporable in his routine.

The conclusion is that if the project would continue, the movements of the dancers have to be tested more extensively to see if they might accidentally trigger changes in tuning.





Functions

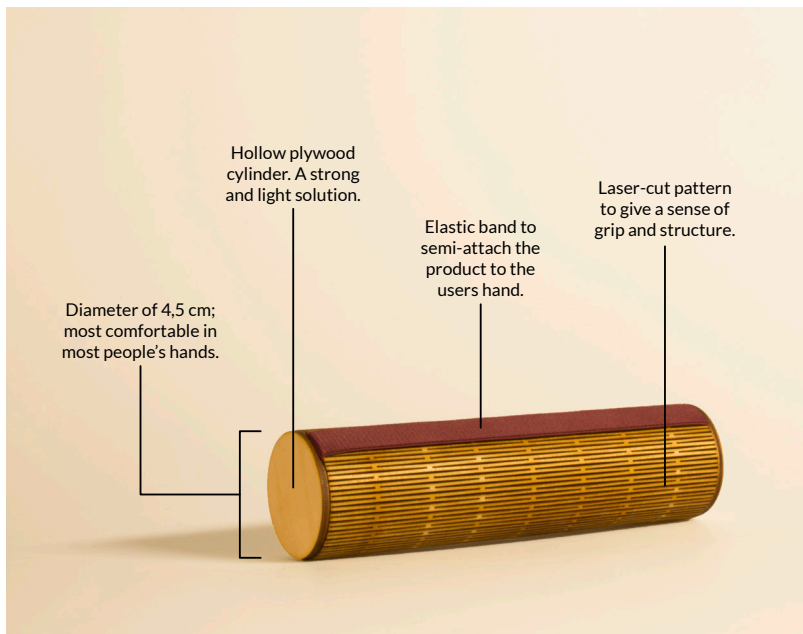
Massine can be used in the two following ways. The first user case is to make an easily accessible instrument that can influence filters over music. The data of two axes is converted to midi signals. Any Digital Audio Workstation (for example Fruity Loops or Ableton Live) can link this midi data to different filters. This way, the user can move Massine through space, controlling the impact of the effects.

The second user case is expressed in dancing. Normally, the music only influences the dancer, using Massine it's also the other way around. Massine can recognise 4 different emotions in dance. When you trigger such an emotion, Massine gives you subtle vibration feedback, and sends the emotion to the OOSCI network. This is where the Group Music Improvisation System kicks in.

We worked together with WOodd, a string instrument that can change it's tuning between 4 different scales. Because of this automatic tuning, you can play whatever you want without playing out of tune. Massine can recognise happiness (major scale), sadness (minor scale), mysteriousness (Harmonic minor scale) and swing/funk (blues scale). After it recognises an emotion, this emotion is send to WOodd, and the scale of WOodd changes. This way, the dancer and the instrument have a very strong connection during the performance.

Form and materials

The goal of the shape was to invite users to interact with the object with fluent motions. The product should invite people to dance with it, and should be comfortable to hold while doing so. Through testing it was determined that to realize this, the shape should be light, compact and in a way attached to your hand. For comfort a kind of grip was desirable. These are qualities that are combined in the final prototype.



The pattern was decided on after trying several different ones (Appendix B). Using formulas Joca calculated the amount of repetitions necessary to make the wood flexible enough to bend it in a full circle with a 4.5 cm radius. Unfortunately, not all the patterns were flexible enough after all. The current pattern was chosen because it was both flexible enough, and still strong. Other patterns became so brittle when they had the right flexibility that they could not be used.

Technology

Dancing requires a freedom of movement. That's why the focus is on making a prototype that is wireless and compact, to still provide this freedom to move when someone is using the product. From a hardware perspective this means the use of a smaller version of the popular Arduino microcontroller and bluetooth for wireless communication. A diagram of the circuit is included in Appendix C.

In terms of software the group tries to integrate Massine into existing systems for creating music. For this reason MIDI is used as the protocol to communicate with our digital audio software. During the design process FL Studio was used, but it is also possible to use equivalent Digital Audio Workstations like Ableton Live for example. The connection with other projects in the Group Music Improvisation System project is built on the OOCIS platform. This platform is especially made for designers to easily make systems that communicate with each other. A descriptive model of the Massine-system is included in Appendix D.

Components

Arduino Mini Pro 328:

To control the system and its components an Arduino microcontroller was selected. The choice for an Arduino Mini Pro with an ATmega328 was made for two reasons. It has a low energy use and really compact dimensions of 18x33mm. These aspects make this component perfect for prototyping a wireless and portable system.

Sparkfun LSM9DS0 IMU:

This chip provides a full motion-sensing system. It has a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer. By using this component it is possible to measure more precisely the movements of the user than by just using an accelerometer as was done in the first iteration of the prototype.

Sparkfun Bluetooth Mate Silver:

Bluetooth is the protocol of choice to wirelessly transfer the information from the product to a pc, because this protocol matched best the requirements for for project: it is easy to implement, relatively power efficient and has a range of about 20 meters. This bluetooth 2.0 modem is especially designed to be used with the Arduino Mini Pro.

Linear Resonant Actuator (LRA) Vibration Motor:

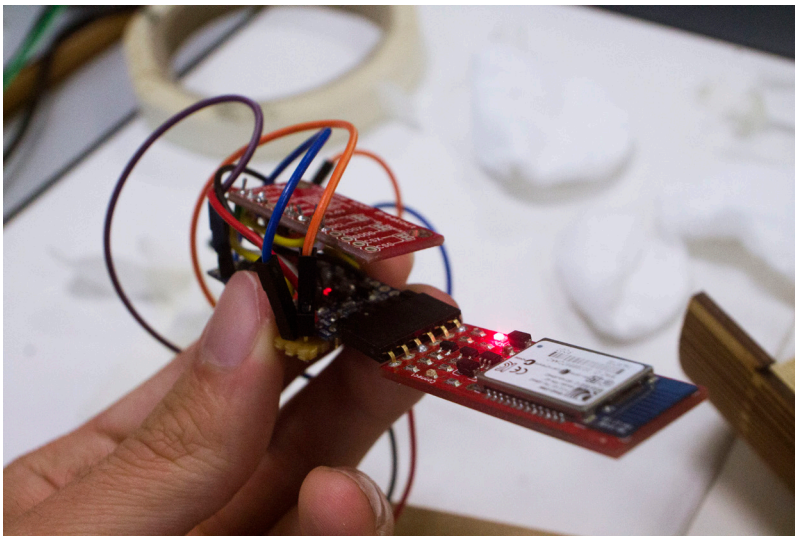
To provide the user a second source of feedback on his or her movements besides sound, haptic feedback was implemented. A LRA vibration motor is integrated in the system to provide this feedback.

Two AAA Alkaline Batteries:

The whole system works on 3 Volts. In an earlier iteration the battery powering the system was an RS CR2450 coin cell. At 600mah the capacity of this type of battery is sufficient, but after tests the group found out that it can't deliver enough current to shortly activate a LRA vibration motor. After some user tests for the best form, the result was that prototype could be bigger than was expected beforehand. This leaves more space for batteries. For this reason the powersource changed to two 1,5 Volts AAA batteries connected in series. They offer double the capacity of the RS coin cell and can provide enough power for the vibration motor.

PC:

The PC acts as a bridge between our product and the rest of the system. It receives the data, converts it to the right format and sends it to other software and projects.



Software

Arduino sketch:

The code on the Arduino has two functions. The first one is to send the data of the LSM9DS0 to the pc. The second one is to recognize certain patterns in movements. When the system recognizes a pattern the vibration motor is shortly activated and a command is sent to the pc via bluetooth, which is seen as a serial port. The source code is included in Appendix A.

Processing sketch:

This program on the pc receives the sensor data and the commands triggered by patterns of movements. The software sends the command to an OOCSI channel, so other projects in GMIS can use it to make their instruments react on it.

The software turns the sensor data for orientation, which is a number of -180 to 180 for each axis, into a scale that is suitable for a MIDI-signal: 0 to 127. Then it structures the information as a typical MIDI-signal. This is sent to a virtual serial port that connects Processing to a Serial-to-MIDI converter running on the same PC. The sourcecode is available in appendix E.

Virtual Serial Port Emulator (VSPE):

This piece of software makes it possible to create virtual serial ports on a PC. It is used to connect the processing sketch to the Serial-to-MIDI converter by creating a pair of serial ports called COM9 and COM10. The processing data goes in port COM9 and goes out COM10 into the serial-to-MIDI converter.

V2D Serial-to-MIDI converter & LoopBE virtual MIDI port:

The converter makes an MIDI signal out of the serial data. To send this data to a music production program as FL studio, a virtual MIDI port is needed. During our project, LoopBE was used.

FL Studio 11:

MIDI signals itself are only commands and not a sound. To make audio effects out of this signal, FL Studio was used as a Digital Audio Workstation. FL Studio makes the connection between the signals and the control of certain sound effects.

Use of energy

| Component | Energy use (mAh at 3V) |
|-----------------------|------------------------|
| Arduino Mini Pro | 4.74 ¹ |
| Bluetooth Mate Silver | 45 ² |
| LSM9DS0 IMU | 6.45 ³ |
| Vibration Motor | 0.77 ⁴ |
| Total | 56.96 |

When developing a portable and wireless product the energy use is an important factor. It is required to balance the size and weight of the battery with the desired user time on a battery charge.

The average energy use of Massine is about 57 mAh at 3V when in use. Two AAA batteries provide a capacity of 1200 mAh at 3V. Capacity divided by the energy use gives the

1 - Home Automation Community. (2015) <http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/>

2 - Roving Networks. <http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf>

3 - ST. (2013) <https://cdn.sparkfun.com/assets/f/6/1/f/0/LSM9DS0.pdf>

4 - Based on 100 vibrations in an hour of 0.4 seconds each. When activated this component uses 70 mAh. - Precision Microdrives Ltd. (2015) <https://catalog.precisionmicrodrives.com/order-parts/datasheet/308-100-8mm-vibration-motor-3mm-type>

time that the system could run on a set of batteries. This optimal user time is influenced negatively by temperature and differences in battery quality for example. This is covered by using a correction factor of 0.75.⁵

$(1200 \text{ mAh} / 56.96 \text{ mAh}) * 0.75 = 15.8 \text{ hours.}$

About 16 hours of battery life is sufficient for demonstrations on a Demo Day. There is however room for improvement as the next section shows.

Improvements for a final product

From a hardware perspective the group foresees that the microcontroller, motion-sensing chip and bluetooth modem will be combined in one custom made system on a chip. By doing this, the system could be made smaller and more importantly: the battery life will be better.

One change that directly doubles the battery life is using another bluetooth modem. A relatively old modem with bluetooth version 2.0 was implemented in the prototype. The new 4.0 version of bluetooth is more power efficient; a modem with this version will use 16 mAh⁶ instead of 45 mAh. Another hardware aspect that can be improved is the use of a rechargeable lithium-polymer battery instead of disposable AAA batteries. This is more environmentally friendly and a lithium-polymer battery offers more capacity than a alkaline battery of the same size.

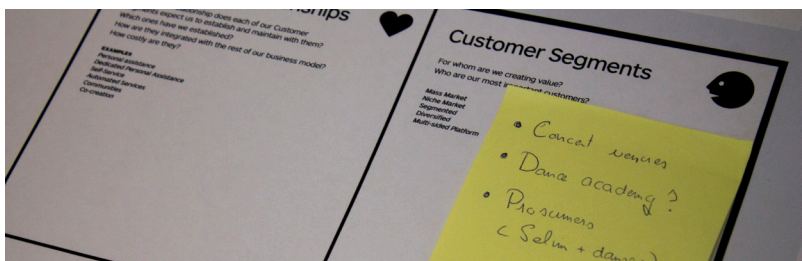
5 - Jenesis International. (2015) <http://jenesisproducts.com/how-is-battery-life-calculated/>

6 - Blue Creation. https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/BC118_DS.pdf

On the software side the prototype is complex at the moment. It is time consuming to start the system and the way to do it is not intuitive. In a final product the processing sketch that receives the sensor data, the serial-to-MIDI converter and the virtual MIDI-port need to be combined in one piece of software, eliminating the need for workarounds like virtual serial ports to connect different programs. This will make the process of connecting the device to a pc faster and more user friendly.

Business

To get a better idea of the connection between the competency Designing Business Processes and the project, the business model canvas⁷ was used. The canvas is a tool that helps in designing a business model by showing the different aspects in a template that one can fill in. Based on the work with this tool the group decided on the target groups for the product and made a first attempt in creating a business model.



7 - The Business Model Canvas in 2 minutes:. (2015). Retrieved June 9, 2015, from http://businessmodelgeneration.com/canvas/bmc?_ga=1.130332175.1832063013.1433859580

Target groups

Massine is made with two user segments in mind. Professional users like concert venues and festivals that can hand multiple Massines to the audience to stimulate interaction with the performing artists. The typical context for this user case is an event. The second group is formed by prosumers like dancers and people that produce music. They use Massine for experimentation and creating new experiences with dance and music.

Business model

The business model of the open source microcontroller Arduino⁸ served as an inspiration for designing a business model for Massine. The value proposition of the product is that it makes group music improvisation accessible for a big audience, offering them fun while using it. There is space to learn and make the system more complex, which gives surprise as a third value proposition.

The main channel for selling the product will be a special website, where customers can order the product. Besides acting as a sales channel the website fulfills another important role serving as a place to build a community. By open sourcing the software people can help in development, exploring new features and share knowledge. This community can also serve as a source of (starting)capital for development of Massine.

8 - Thompson, C. (2008). Build it. Share it. Profit. Can open source hardware work?. Wired, 10, 08.

This makes the user not only a customer, but also a key partner. Other partners include the manufacturer, the suppliers of components and developers of music software where Massine can plug in to.

Possible revenue streams are profits from selling the product and crowdfunding from the user community to gain capital. A continuous stream of revenue is customizing our product for professional users. For example developing a custom experience using Massine for a concert tour of an artist.

From the cost perspective important factors are the costs for components and manufacturing the hardware. Another resource with costs is the hosting for the website and the costs of employees to develop software, control the community and arrange the sales.

At this stage it is still complicated to turn this idea in a viable business process though. The number of potential users is not clear, nor are the height of the possible revenue and the costs for creating and maintaining the system. Further research on this topic is required.

CONCLUSION

Based on the feedback we got on the final Demo Day and our own experiences during the design process we discovered several points of improvement. In terms of interaction the mapping of the sensor data needs to be more intuitive. The movements that trigger interaction with the WOODD project are easy to integrate in expressive dance, but for influencing the filters it is still not fully possible to translate expressive movements into expressive sound. We made a lot of progress in comparison to the Mid-Term Demo Day though.

Looking back at our user tests a positive note is that we did a big number of them and integrated them in our process for decisions on movement and form. On the other hand we can improve our documentation of the user tests and make them more formal, for example by formulating fit criteria beforehand. If we do this we will get more information out of these tests and the results will form a stronger proof that our concept works. It would have added a lot of value to a user test if we had involved more experts, dancers and musicians for example.

For the form of our product we could have improved the internal construction. The components were fixed on a piece of wood and empty space was filled with cotton wool. The sides were not glued, because they needed to be opened to replace the batteries. The result was a prototype that could not withstand movements that were extremely expressive.

The construction was easy to repair though, but it is better to avoid this problem at all. By thinking more about the internal construction we could have improved the durability of the prototype.

During the Final Demo Day we got positive feedback on the concept. It was simple and the connection with the other project made it fit well in the goal of this project to create a system for group music improvisation. The look and feel of our project and the presentation of it reflected this combination of simplicity and refinement. The teamwork was really good. Everyone was open for each others thoughts and ideas. Tasks were finished in time and there was a good working atmosphere. When we were stuck in the process, the coach meetings helped us in getting back on track and setting the right priorities. Working together with other fellow students helped us in getting knowledge about music, accelerated idea generation and resulted in an interesting collaboration between our project and WOodd using the OOCSI platform to connect these projects.

We are really happy with the results. This project gave us the opportunity to develop in a wide range of competencies, most notably user focus and perspective, form and senses and integrating technology. The point of improvements are aspects that we keep in mind in our future work. Looking back at the total project we can be proud of what we have done.

REFLECTIONS

Joca van der Horst

What I liked about the GMIS project was the use of music improvisation as a context for designing an instrument and a system. Another aspect I appreciated was the explorative nature of this project. The project I did in my B1.1 was more restricted and structured. I think that such a project is good for a first semester, but the freedom offered in GMIS gave me the opportunity to develop in a wide range of competencies, most notably the following ones.

Ideas & concepts

For example working together with another first years group at the start of the project. We tried methods like brainstorming, making moodboards and ideazation by making. While doing this we shared knowledge. Selim taught me for example the basics of MIDI signals and Fruity Loops studio, which was of great use for developing the different prototypes. This collaboration between groups helped a lot in starting up the project, because we could iterate fast and test multiple concepts before selecting. It was a great way to develop the competency Ideas & Concepts.

Teamwork & communication

After the Mid-term Demo Day this good working atmosphere continued. The teamwork with Linde and Bart was really good, because our personalities match but we all

have our different expertises. This made our product Massine a project with the common vision to make the joy of music improvisation accessible for everyone, while everyone had still space to put something personal in it. Bart did that for example for the sounds, using his expertise in music and improvisation. Linde has a great artistic talent which is clearly visible in the logo and posters she designed. My hobby of telling stories reflected in the presentation on the Demo Day that was set up to show the story of our process.

Part of the project was to create a connection between our project with another one using the OOC SI platform. This required a lot of contact with the first-years group that created WO ODD, a snare instrument that can change its scale by moving our product Massine. We both had to make changes in our projects to create a logical interaction that would not break the prototypes. A scale that would continuously change without pausing is for example fatal for the sound of an instrument. This process went well and I learned about how powerful it can be to work together. Both groups had their own projects, but we worked together on the product movie for example. We got help from the ID community to compose music for the movie and arrange a professional dancer for the movie. This was fun to do and thanks to great teamwork the result was something I am really proud of.

In terms of communicating ideas I made a step forward from past semester, but there is still a lot to learn in this aspect. The product video for example looked good, but could have been more clear in showing the exact connection between the two projects.

Integrating technology

The concept we had was challenging from a technological perspective. For my past project the idea was to create a wireless product that was part of a central system, but the prototype wasn't. This semester we succeeded in that and it helped a lot in testing the prototype and communicating how our concept could work in a final iteration.

During the process to reach this result I learned about implementing wireless protocols in a prototype and selecting the right batteries to create a product that is actually usable. Next to hardware I developed a lot in software, using processing as the core of the technological side of the project. I learned how to connect my code to our product, existing systems like MIDI for the Digital Audio Workstation and platforms like OOCsI.

Another important thing I learned was how to work with multiple people on the same code. I made the code to make the Arduino communicate with the PC, while Bart made the code to recognize movements for example. To make this work out well it is required to document well, for example which variables are used. In the future I'd like to develop my skills to interpret sensor data. The prototype was working, but it was difficult to convert expressive movements into an interesting sound. A better knowledge of how to use the sensor data helps in writing code to create a more intuitive interaction.

Form & Senses

Besides technology I focused a lot on Form and Senses. I followed the assignment Basic Formgiving Skills (BFS) last semester and I tried to integrate the lessons learned in the process. As a SDL-activity I followed a laser cutting workshop. For the final prototype I sketched a shape based on the outcomes of the user test and used the lattice hinge technique to create the basic shape using a laser cutter. Using the woodworking skills I developed at BFS the prototype was assembled. The result was visually pleasing and on the Final Demo Day lots of people liked the simple, yet interesting look of it.

Form and Senses is however not limited to just the outside and I think there is room for improvement on this part. We implemented a vibration motor in the prototype that simply buzzed when a function was activated. The time and strength of this vibration highly influences the interaction though. This is an example of parts of the interaction that could be more polished.

User focus and perspective

User tests were an integral part of our process. We used them to define the best shape for our product, by making form studies that testers used to dance on music. By analyzing their movements we researched what kinds of movements are typical for music with emotions like happiness and sadness. I learned how to set up a user test in a short time and integrate the results in the design process, but I think that by changing some aspects we could have get more information out of these tests. Our user tests were mostly informal and didn't include things like fit criteria. Our results could have been more convincing if we had done that.

In future projects I'd like to add more depth to the concept, for example by more extensive research at the start of the project and by doing user tests using a more structured and quantified method. Besides that I am happy with the results of the project. On areas as form & senses, technology and teamwork I developed a lot in comparison to my past project. What I found out by doing this project is that projects like GMIS that are a bit more abstract and explorative are a good fit for the sort of designer I want to be.

Linde van Beers

I chose this project because I wanted to do something with more 'space' than my previous project. I was in Tools for Teaching last semester and found that it was too defined for my liking. Therefore I wanted to do a project that was more free. This project definitely lived up to my expectations in that sense. I liked that it left a lot of room for experiment and think I learned a lot, and a lot of different things from last semester, because of that. My main developments have been in Integrating Technology, Ideas and Concepts, Form and Senses, User Focus and Perspective and communication.

Integrating Technology

The first half of this semester (Q3), my main occupation was programming. I already knew quite a lot about processing because of the Creative Programming assignment I did last year, but didn't have a lot of experience with Arduino yet. Because we started experimenting with technology almost immediately in this project, I had a lot of opportunity to make up for that. I worked with an accelerometer for the first time. Using code from Creative Programming,

sources I found on the internet and a library for music projects (beads), I was able to make a working prototype that created a beat when the sensor was moved with a certain acceleration. I feel like I have much more extensive understanding of programming now, since I took my knowledge from last semester and built on it further by adding elements one after the other. I combined both things I had already done and new things, which meant I had to at least understand the code to prevent errors. I might not have written everything personally, but as Arduino is an open-source platform I don't think this should be the goal. I think it's much more important to be able to combine existing code to form new code for new purposes.

Ideas and Concepts

The way we tackled the ideation process this semester was completely different from last. Back then, it existed mainly of brainstorming, which, this time, was just a small part of the equation. It was what we started of with, but soon we moved on to different techniques. One of them was a jam session: With about half the people working on this project we brought all our instruments to make music and find inspiration. Not only was it a lot of fun, but it was also from where our idea originated. I also had my first experience with making mood boards. This was useful because we were with a group of 8 people, all with their own ideas about how the project should be interpreted, and making a mood board was a good way to visualize these interpretations, and gave an opportunity to discuss them. The most valuable thing we did to go from idea to concept, however, was, for me, experimenting. I really loved that in this project we could just start building something and see what would come out of it. It was an extreme contrast with what we did before, namely only thinking to develop the concept

further. Getting started on prototyping right away gave me lots of energy and motivation, and new ideas were generated automatically, often even accidentally.

Form and senses + User focus and perspective

For me, these competencies were very closely linked together this semester. When I first started working on the appearance of our project, I kind of blocked. I didn't know how to get started at all. I tried making some drawings and looking at instruments for inspiration, but it didn't get me very far. When at last I had some kind of idea of what I wanted, it wasn't actually based on anything and it was still very vague. At this point our coach helped me out. She pointed out that I was diving in too deep at once. She advised us to start testing with some extremes first, and see what qualities people found important in the kind of product we were trying to make. So this is where UFP came into the mix. We crafted some very quick objects, you can't even call them prototypes, and made sure they were all different in a way. Some were small, others big, some heavy, some light, some were just weird. We then asked people to pick their favourites and move with them. This gave us the opportunity to see what users found important in this product, and what shapes triggered the most interesting movements. From there we could easily decide what shapes to use as a basis, and create something aesthetically pleasing with the qualities of each. We continued testing what uncertainties we had, like the diameter, and consequently came to the final design.

Communication

I was responsible for most of the graphic design during this project, and I think I've grown a lot from it. When comparing the poster I made for mid-term demo day and

the presentation we had on final demo day, the difference is enormous. I think the main cause for this is that at final demo day I had a basis to work from. The photo's I took while filming the movie, depicted a clear colour scheme and a certain atmosphere that I could easily work from. Besides, there was the finished prototype, with the iconic pattern that I could use as inspiration. This time, the graphic design really made sense with all the rest of the presentation, including the product itself, thus creating a very clear and appealing exhibit.

In conclusion, I'm quite happy with my development during this semester. I was able to reach a lot of my goals within the project. It has been very good for me to work in such a different project from the last one, as I've now found a design process that suit me better. I look forward to putting my new skills to use in another project next year.

Bart Leeuwenburg

During the project I was able to develop every competency, but there are a few that I want to highlight. Because of the dynamic forming of groups, I got to know a new working method. As to the competencies, I especially got the chance to develop IT, UFP and FS. I also want to reflect on the teamwork of the Baton group, which was outstanding to me. I will reflect on how I can improve myself and my working method as much as possible.

SCRUM

Quite early in the project, we were learned the SCRUM working method. This meant that we started having meetings every morning to see if anyone needed help with something. This was a very nice way of working, because this way we could use the manpower we had very efficient. We could take on big problems together, and then everyone could continue with their own part. This method of working is definitely something I'm going to use again, because it helped me to work efficient in a team.

Integrating Technology

From the beginning to our final prototype, IT was an essential competency. I learned about different sensors (proximity meter, accelerometer, 9 degrees of freedom sensors, a heartbeat sensor) and I saw a lot of diverse implementations in our own and other projects. I got the chance to develop my Arduino as well as my Processing coding and understanding. As I thought beforehand, the assignment Creative Programming was perfect to implement in my project. In the project, I was able to write a code that recognizes certain movements in dance. Also, I was able to understand complicated codes that we found on the internet.

I'm sure I have developed a strong awareness to build further on in the future.

User Focus and Perspective

We wanted to make a prototype that was attainable and intuitive for everyone. This required a lot of user testing. To some extent, we choose for doing many user tests with a lot of people, instead of to document a lot. This, we could have done differently. However, this way we managed to

do a lot of form exploration on our product as well as movement analyses of dancing with objects.

I learned how to set up a good user test, so we can efficiently get out of it what we want to know. What I would do better next time, is to make clearer fit criteria. This time our method worked, but next time I would document convincing data for our choices.

Form and Senses

Our process was also all about form and senses. Especially in the last quartile we developed a lot of different forms.

After we user tested the forms, we thought about the pattern and the material. After we choose for wood, we printed different patterns to see which people thought was best. We also tested the color of the elastic band, the strength of the band, and thought about if we should paint the cap of the prototype.

Working together with people that knew a lot about form and senses gave me a lot of new insights about the beauty of a product. I learned how to develop an opinion on things, partly in Milan, and was able to implement it in my project to some extent, for example in painting our product or finishing the wood.

I think beauty and art is very much based on feelings. For things like a presentation I first had no feeling at all, however, at our final demo day I learned to draw imaginary lines to line up our product and posters. This made it look a lot more tight, and felt finished.

Teamwork

The teamwork in the Baton group was excellent, and I wanted to know why it worked so well. My hypothesis was that it came because of the diverse characters in the group. We complemented each other perfectly, as later was confirmed by an online personality test. Because we were with just the 3 of us, everyone had the chance to say their opinion. We were also persons that were very open for critique, which made it easy to adjust each other's ideas. I learned to be patient, to respect other people's ideas and to get the best out of everyone's input. I will try to use this way of working in the future.

Something that could have gone better, was the cooperation with Ward a few weeks. Together we worked on the heartbeat sensor at the beginning of the semester. The progress we made was slow, and this was mainly because we both weren't very structured. This made me think about my way of working, and I tried to improve it the rest of the semester. Now I work with to do lists, which work very well. I learned to make not too big goals, so you can cross of things of a list. This works very motivating to me. Also, I'm more conscious about time I waste, and I'm making progress in putting myself back to work when I notice this.



Appendix A

```
// This Arduino sketch is made for Massine, a result of the GMIS
project.
// The code recognize movements is based on a LSM9DS0 that is
oriented upside down.
// Information on how to connect the components can be found in the
appendix in the project report.
```

```
// This code is partially based on the example sketch of the Adafruit
Simple AHRS library
// DiCola, T. (2014) "ahrs_lsm9ds0". https://github.com/adafruit/Adafruit\_AHRS
```

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM9DS0.h>
#include <Adafruit_Simple_AHRS.h>
```

```
int emotion = 3; // emotion 3, the minor scale, is the one the system
starts to send.
```

```
// Major - jump
float accZ = 0; // new input
int i = 5; // teller van herhalingen
int t = 0; //teller
```

```
// Harmonic - hold sideways
int x = 90; // om en nabij de 90, waarde verticaal
float inputX; // input in graden
float tijdVerticaal = 1.5; //1.5 seconden verticaal, dan wordt emotie
afgegeven
```

```

int j;

// Blues - make a wave movement
float accX = 0; // new input
float gyroY = 0; //second new input
int k = 0; // teller van accX waarde
int l = 0; // teller van gyroY waarde
int m = 0; //teller

// Minor - Turn Massine and open hand palm
float roll; // new input
float oldRoll; // previous input
float d; // difference in every value
float p; //een gradenteller

//vibration
int vibration = 11;
int temps = 400; // How long the vibration will continue
//int brightness = 0; // how powerful the motor will vibrate
//int fadeAmount = 5; // how many points to fade the vibration power
by

// Create LSM9DS0 board instance.
Adafruit_LSM9DS0 lsm(1000); // Use I2C, ID #1000

// Create simple AHRS algorithm using the LSM9DS0 instance's acce-
lerometer and magnetometer.
Adafruit_Simple_AHRS ahrs(&lsm.getAccel(), &lsm.getMag());

// Function to configure the sensors on the LSM9DS0 board.
// You don't need to change anything here, but have the option to select
different
// range and gain values.

```

```

void configureLSM9DS0(void)
{
  // 1.) Set the accelerometer range
  lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_2G);
  //lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_4G);
  //lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_6G);
  //lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_8G);
  //lsm.setupAccel(lsm.LSM9DS0_ACCEL_RANGE_16G);

  // 2.) Set the magnetometer sensitivity
  lsm.setupMag(lsm.LSM9DS0_MAG_GAIN_2GAUSS);
  //lsm.setupMag(lsm.LSM9DS0_MAG_GAIN_4GAUSS);
  //lsm.setupMag(lsm.LSM9DS0_MAG_GAIN_8GAUSS);
  //lsm.setupMag(lsm.LSM9DS0_MAG_GAIN_12GAUSS);

  // 3.) Setup the gyroscope
  lsm.setupGyro(lsm.LSM9DS0_GYRO_SCALE_245DPS);
  //lsm.setupGyro(lsm.LSM9DS0_GYRO_SCALE_500DPS);
  //lsm.setupGyro(lsm.LSM9DS0_GYRO_SCALE_2000DPS);
}

void setup(void)
{
  pinMode(vibration, OUTPUT); // vibration motor
  //digitalWrite(11, HIGH); // turn the vibration motor on (HIGH is the
  //voltage level)
  //delay(400); // wait for a certain time
  //digitalWrite(11, LOW); // turn the vibration motor off by making
  //the voltage LOW

  Serial.begin(115200);

  // Initialise the LSM9DS0 board.
  if(!lsm.begin())
  {
    // There was a problem detecting the LSM9DS0 ... check your con-
    //nections
    Serial.print(F("Ooops, no LSM9DS0 detected ... Check your wiring or
    I2C ADDR!"));
    while(1);
  }
}

```

```

// Setup the sensor gain and integration time.
configureLSM9DS0();
}

void(* resetFunc) (void) = 0; //declare reset function @ address 0

void loop(void)
{

sensors_vec_t orientation;
lsm.read();

// Use the simple AHRS function to get the current orientation.
if (ahrs.getOrientation(&orientation))
{

//_____
// Code to detect movement that triggers a blues scale on WOodd
(command is emotion 0)
// The left and right side of Massine are moved up and down for 6
times, resembling a wave.

float gyroY = (int)lsm.gyroData.y; // get the desired sensor data
float accX = (int)lsm.accelData.x; // get the desired sensor data

m = m+1; // start a counter

// The wave movement has a really distinct pattern in sensor values
for accelero X-axis and gyro Y-axis.
// We count how many times both meet the desired values typical for
this movement within a certain
// timeframe. If both counters reach a value we activate emotion 0,
unless the emotion is already 0.

//accX

if((accX>=9000) && (k==0)){

```

```
k = k+1;  
m=0;  
}
```

```
if((accX<=-9000) && (k==1)){  
k = k+1;  
m=0;  
}
```

```
if((accX>=9000) && (k==2)){  
k = k+1;  
m=0;  
}
```

```
if((accX<=-9000) && (k==3)){  
k = k+1;  
m=0;  
}
```

```
if((accX>=9000) && (k==4)){  
k = k+1;  
m=0;  
}
```

```
if((accX<=-9000) && (k==5)){  
k = k+1;  
m=0;  
}
```

```
if((accX>=9000) && (k==6)){  
k = k+1;  
m=0;  
}
```

```
if((accX<=-9000) && (k==7)){  
k = k+1;  
m=0;  
}
```

```
// _____
```



```
//gyro

if((gyroY>=20000) && (l==0)){
  l = l+1;
  m=0;
}

if((gyroY<=-20000) && (l==1)){
  l = l+1;
  m=0;
}

if((gyroY>=20000) && (l==2)){
  l = l+1;
  m=0;
}

if((gyroY<=-20000) && (l==3)){
  l = l+1;
  m=0;
}

if((gyroY>=20000) && (l==4)){
  l = l+1;
  m=0;
}

if((gyroY<=-20000) && (l==5)){
  l = l+1;
  m=0;
}

if((gyroY>=20000) && (l==6)){
  l = l+1;
  m=0;
}

if((gyroY<=-20000) && (l==7)){
  l = l+1;
  m=0;
}
```

```

if ((k==8) && (l==8) && (emotion != 0)) {
  emotion = 0;
  m=0;

  digitalWrite(11, HIGH); // turn the vibration motor on (HIGH is the
  voltage level)
  delay(temps);          // wait for a 20ms
  digitalWrite(11, LOW); // turn the vibration motor off by making the
  voltage LOW

  k=0;
  l=0;

}

if (m==10) { // when the timer is 10, the sequence is reset
  k=0;
  l=0;
  m=0;
}

//_____
// Code to detect movement that triggers a major scale on WOODD
(command is emotion 1)
// The person holding Massine jumps

float accZ = (int)lsm.accelData.z; // get the desired sensor data
t = t+1; // start counter

// A jump is recognizable from extreme negative and positive spikes
for acceleration in the Z-axis.
// This code divides the spikes in 4 values that the system should
measure. If they are all measured
// within a second, emotion 1 is activated.

if((accZ<=-25000) && (i==0) && (emotion != 1)){
  i = i+1;
  t=0;
}

```

```

if((accZ>=2000) && (i==1)){
i = i+1;
t=0;
}

if((accZ<=-25000) && (i==2)){
i = i+1;
t=0;
}

if ((i==3) && (emotion != 1)){
emotion = 1;
i=0;

digitalWrite(11, HIGH); // turn the vibration motor on (HIGH is the
voltage level)
delay(temps); // wait for a 30ms
digitalWrite(11, LOW); // turn the vibration motor off by making the
voltage LOW

}

if (t==10) { // reset the counter is not all values are measured within
the timeframe
i=0;
t=0;
}

// _____
// Code to detect movement that triggers a harmonic scale on
WOODD (command is emotion 2)
// Massine is hold sideways at 90 degrees for 1.5 seconds

float inputX = abs(orientation.pitch); // // get the desired sensor
data. Input in degrees

if ((x-inputX< 10) && (emotion != 2)) {
j = j+1; //if the value is between the desired values, start a
counter
} else {
j = 0; // if not, reset the counter
}

```

```

}
if ((j>=(tijdVerticaal*10)) && (emotion != 2)) {
  emotion = 2;          //activate emotion 2 if the counter reaches a
                        //certain value and the emotion is not 2 already
  j = 0;              // reset the counter

  digitalWrite(11, HIGH); // turn the vibration motor on (HIGH is the
                          //voltage level)
  delay(temps);        // wait for a 30ms
  digitalWrite(11, LOW); // turn the vibration motor off by making the
                        //voltage LOW
}

//_____
// Code to detect movement that triggers a minor scale on WOodd
// (command is emotion 3)
// Massine is hold upside down and the hand palm is opened

float gyroX = (int)lsm.gyroData.x;
float accY = (int)lsm.accelData.y;

if (accZ>=0) { // Check for this emotion is the Z-axis of the accelero
              //is positive

if ((gyroX >= -3000) && (abs(accY) >= 15000) && (emotion != 3)) {
//If the x-axis of the gyro is bigger than -3000 + accelero Y bigger than
15000 + emotion is not 3
  emotion = 3;
  p = 0;

  digitalWrite(11, HIGH); // turn the vibration motor on (HIGH is the
                          //voltage level)
  delay(temps);        // wait for a 30ms
  digitalWrite(11, LOW); // turn the vibration motor off by making the
                          //voltage LOW

}}

// And now we send some values to processing over bluetooth //
/* 'orientation' should have valid .roll and .pitch fields */
Serial.print(F("Orientation: "));

```

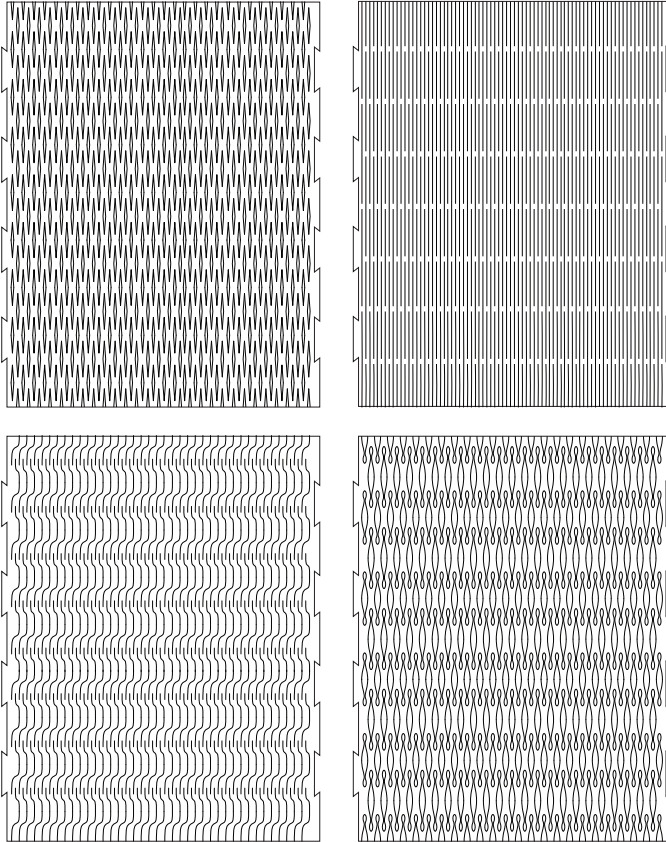
```
Serial.print(orientation.roll);
Serial.print(F(""));
Serial.print(orientation.pitch);
Serial.print(F(""));
Serial.print(orientation.heading);
Serial.println(F("")); Serial.print(F("Acc: "));
Serial.print((int)lsm.accelData.x);
Serial.print(F(""));
Serial.print((int)lsm.accelData.y);
Serial.print(F(""));
Serial.print((int)lsm.accelData.z);
Serial.println(F("")); Serial.print(F("Gyro: "));
Serial.print((int)lsm.gyroData.x);
Serial.print(F(""));
Serial.print((int)lsm.gyroData.y);
Serial.print(F(""));
Serial.print((int)lsm.gyroData.z);
Serial.println(F("")); Serial.print(F("Emotion: "));
Serial.print(emotion);
Serial.println(F(""));

}

delay(100);

}c
```

Appendix B



Patterns that we tried for our final prototype.

Calculations

To create the prototype the lattice hinges are used. By laser cutting parallel and overlapping cuts it is possible to bend a flat sheet of an material. The flexibility depends on the thickness of the material and the number of torsional links, the area of the material that is between two cut-out parts.

Different patterns⁹ for the lattice hinge technique were tested. The straight lattice pattern was chosen for its combination of flexibility and aesthetic qualities. To make sure the tube created by bending the wood would withstand all forces that are working on it, the number of required torsional links was calculated. The formula¹⁰ for this is based on the work of Patrick Fenner.

Nomenclature

| | |
|-------------|---|
| k_{laser} | = Laser Kerf (m) = 0.001 |
| n | = Number of columns of torsional links = 82 |
| t | = Material thickness (m) = 0.003 |
| Θ | = Total bend angle of the piece ($\Theta = \theta \times n$) |
| θ | = Angle of twist per link (radians) ($90^\circ = \pi/2$ radians) (we have 83 links) |

$$n \geq \frac{\Theta}{\frac{\pi}{4} - \cos^{-1} \left(\frac{k_{laser} + t}{2\sqrt{\frac{t^2}{2}}} \right)}$$

When filling in the formula with our data we get this:

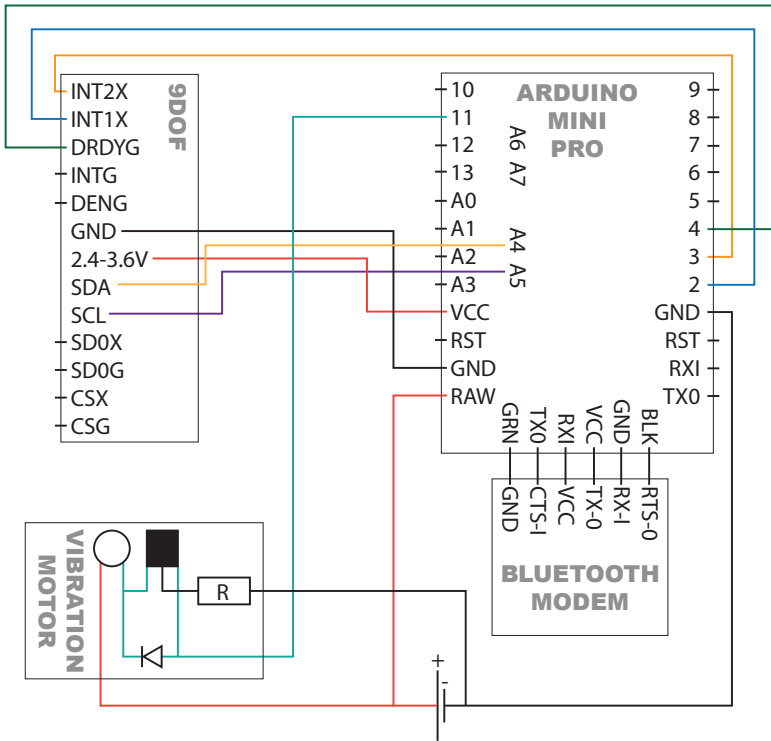
$$82 \geq \frac{0.075 \times 82}{\frac{\pi}{4} - \cos^{-1} \left(\frac{0.001 + 0.003}{2\sqrt{\frac{0.003^2}{2}}} \right)}$$

The result is 13.9, which is less than the number of torsional columns: 82. This means that the number of torsional links is sufficient to achieve the amount of flexibility needed to create a tube.

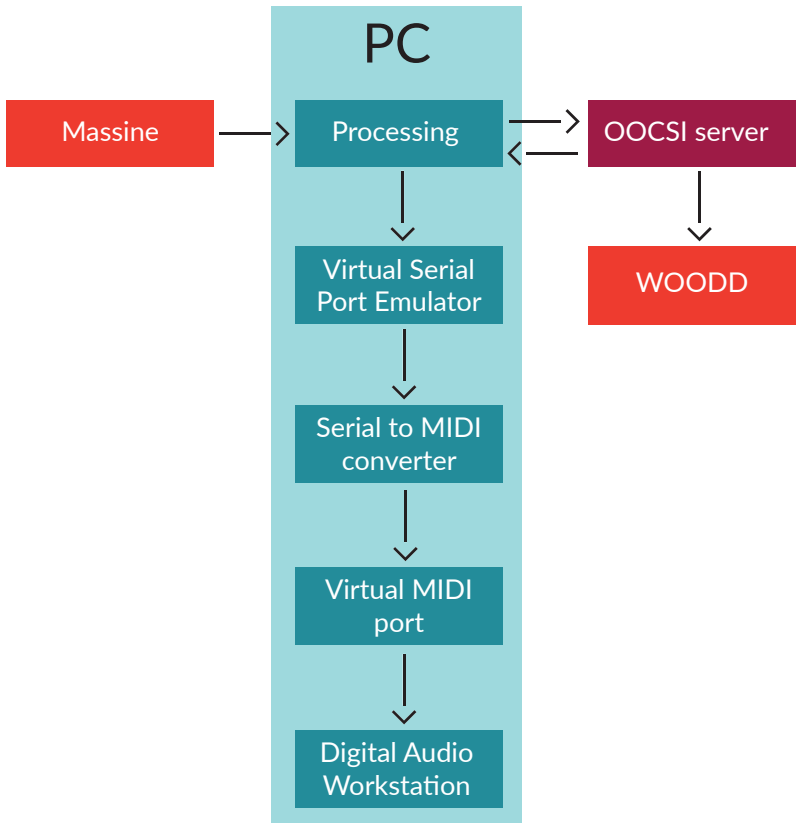
9 - Porterfield, A. (2015). Curved laser bent wood. Retrieved May 15, 2015, from <http://www.instructables.com/id/Curved-laser-bent-wood/?ALLSTEPS>

10 - Fenner, P. (2012). Lattice Hinge Design - Minimum Bend Radius. Retrieved May 15, 2015, from <http://www.deferredprocrastination.co.uk/blog/2012/minimum-bend-radius/#more-2743>

Appendix C



The electric circuit of the prototype.



Descriptive model of the system.

Appendix E

// MIXSTICK, made by Linde van Beers, Bart Leeuwenburg & Joca v/d Horst - June 2015

// This program reads data sent by the 9DOF sensor LSM9DS0 and turns it into a MIDI signal

// The MIDI signal is suitable to use in digital music producing software like Fruityloops Studio

// Besides that, it also sends and receives data over OOC SI.

// PREPARATIONS

// Before using this program you need VSPE, this program creates a pair of virtual serial ports. Call these ports "COM9" and "COM10"

// Download: <http://www.eterlogic.com/Products.VSPE.html>

// To convert the serial MIDI signal made by this program, use a serial to MIDI converter like

// Download: http://www.spikenzielabs.com/SpikenzieLabs/Serial_MIDI.html

// SOURCES

// Townsend, K. - Bunnyrotate.pde - <https://learn.adafruit.com/ahrs-for-adafruits-9-dof-10-dof-breakout/visualizing-data>

// Funk, M. - Examples: Channelsender, Channelreceiver - <https://github.com/iddi/oocsi-processing>

// Tomczak, S. - Examples of sending MIDI data - <http://little-scale.blogspot.nl/2011/08/simple-examples-of-sending-midi-data.html>

// We start with importing some libraries

```
import processing.serial.*;
```

```
import java.awt.datatransfer.*;
```

```
import java.awt.Toolkit;
```

```
import processing.opengl.*;
```

```
import g4p_controls.*;
```

```
import nl.tue.id.oocsi.*;
```

```
OOC SI oocsi;
```

```

// Time to set up some variables
float roll;
float pitch;
float yaw;

float oldyaw;
float heading;
float value2;

float accX;
float accY;
float accZ;

float gyroX;
float gyroY;
float gyroZ;

int MIDIcontrol1;
int MIDIcontrol2;

float value1;

int value1tosend;
int value2tosend;
int value3tosend;
float emotion;

// Serial port state.
Serial port;
String buffer = "";
final String serialConfigFile = "serialconfig.txt";
boolean printSerial = true;
String incoming;

Serial portout;
boolean firstContact = false;

// UI controls.
GPanel configPanel;

```

```
GDropList serialList;  
GLabel serialLabel;  
GCheckbox printSerialCheckbox;
```

```
void setup()
```

```
{  
  size(400, 400, OPENGLE);  
  background(120);  
  frameRate(10);
```

```
  // connect to OOCSE server with "mixstick" to be my channel others can  
  send data to
```

```
  oocsi = new OOCSE(this, "Massine", "diserver.id.tue.nl");  
  // subscribe to channel "gmis2015"  
  oocsi.subscribe("gmis2015");
```

```
  // Serial port setup.
```

```
  // Grab list of serial ports and choose one that was persisted earlier or  
  default to the first port.
```

```
  int selectedPort = 0;  
  String[] availablePorts = Serial.list();  
  if (availablePorts == null) {  
    println("ERROR: No serial ports available!");  
    exit();  
  }
```

```
  String[] serialConfig = loadStrings(serialConfigFile);
```

```
  if (serialConfig != null && serialConfig.length > 0) {
```

```
    String savedPort = serialConfig[0];
```

```
    // Check if saved port is in available ports.
```

```
    for (int i = 0; i < availablePorts.length; ++i) {
```

```
      if (availablePorts[i].equals(savedPort)) {  
        selectedPort = i;
```

```
      }
```

```
    }
```

```
  }
```

```
  // change the port to the one associated with the free virtual port on  
  your computer. Call it COM9 in Virtual Serial Port Emulator
```

```
  portout = new Serial(this, "COM9", 36000);
```

```

// Build serial config UI.
configPanel = new GPanel(this, 10, 10, width-20, 90, "Configuration
(click to hide/show)");
serialLabel = new GLabel(this, 0, 20, 80, 25, "Serial port:");
configPanel.addControl(serialLabel);
serialList = new GDropList(this, 90, 20, 200, 200, 6);
serialList.setItems(availablePorts, selectedPort);
configPanel.addControl(serialList);
printSerialCheckbox = new GCheckbox(this, 5, 50, 200, 20, "Print
sensor data X-axis");
printSerialCheckbox.setSelected(printSerial);
configPanel.addControl(printSerialCheckbox);
// Set serial port.
setSerialPort(serialList.getSelectedText());

}

```

```

// Here we use the sensor data and OOCSI data to create a MIDI signal
and send some data back too OOCSI
void serialEvent(Serial p)
{

```

```

// Let's split the string into separate values of movements measured
by the sensor
String incoming = p.readString();

if ((incoming.length() > 8))
{
String[] list = split(incoming, " ");
if ((list.length > 0) && (list[0].equals("Orientation:")))
{
roll = float(list[1]);
pitch = float(list[2]);
yaw = float(list[3]);

buffer = incoming;
}
if ((list.length > 0) && (list[0].equals("Acc:")))
{

```

```

accX = float(list[1]);
accY = float(list[2]);
accZ = float(list[3]);

buffer = incoming;
}
if ( (list.length > 0) && (list[0].equals("Gyro:")) )
{
gyroX = float(list[1]);
gyroY = float(list[2]);
gyroZ = float(list[3]);

buffer = incoming;
}
if ( (list.length > 0) && (list[0].equals("Emotion:")) )
{
emotion = float(list[1]);

buffer = incoming;
}
}

```

```

// convert the sensorvalues to a scale of 0-127 for the pitch
float value1 = abs(pitch) * 1.42;
float value2 = (180 - abs(roll)) * 0.71;

```

```

// Print the values for debugging purposes
if (printSerial) {
println(value1);
}

```

```

// let's structurize it like a nice MIDI-signal
portout.write(0XB0); // MIDI control change; channel 1
portout.write(1); // MIDI controller #1
portout.write(int(value1)); // MIDI controller value from pitch

```

```

// let's structurize it like a nice MIDI-signal
portout.write(0XB1); // MIDI control change; channel 2
portout.write(1); // MIDI controller #1

```

```

portout.write(int(value2)); // MIDI controller value from pitch

}

// This was a first step to code a function that other instruments in
// GMIS could control the sort of filter Massine controls.
// We found out that OOCSI unfortunately does not support sending
// and receiving messages in the same processing sketch.

// void handleOOCSIEvent(OOCSIEvent e) {
// assign the new MIDI channel from the OOCSI event
// int MIDIcontrol = e.getInt("balk1", 1);
// }

void draw()
{
// We convert the variable "value" to a nice int for OOCSI
int value1tosend = int(value1);

int value2tosend = int(value2);

int value3tosend = int(emotion);

// send to OOCSI ...
oocsi
// on channel "onsChannel"...
.channel("onsChannel")
// data labeled "pitch"...
.data("pitch", value1tosend)
// data labeled "roll"
.data("roll", value2tosend)
// data labeled "toonPos"
.data("toonPos", value3tosend)
// send finally
.send();
}

// Here we deal with some Serial Port issues. Set serial port to desired
// value in this case.

```

```

void setSerialPort(String portName) {
// Close the port if it's currently open.
if (port != null) {
port.stop();
    }

try {
// Open port.
port = new Serial(this, portName, 115200);
port.bufferUntil('\n');
// Persist port in configuration.
saveStrings(serialConfigFile, new String[] { portName });
    }

catch (RuntimeException ex) {
// Swallow error if port can't be opened, keep port closed.
port = null;
    }
}

// UI event handlers
void handlePanelEvents(GPanel panel, GEvent event) {
// Panel events, do nothing.
}

void handleDropListEvents(GDropList list, GEvent event) {
// Drop list events, check if new serial port is selected.
if (list == serialList) {
setSerialPort(serialList.getSelectedText());
}
}

void handleToggleControlEvents(GToggleControl checkbox, GEvent
event) {
// Checkbox toggle events, check if print events is toggled.
if (checkbox == printSerialCheckbox) {
printSerial = printSerialCheckbox.isSelected();
}
}

// That's it folks!

```


REFERENCES

Home Automation Community. (2015) <http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/>

Blue Creation. https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/BC118_DS.pdf

DiCola, T. (2014) "ahrs_lsm9ds0". https://github.com/adafruit/Adafruit_AHRS

Fenner, P. (2012). Lattice Hinge Design - Minimum Bend Radius. Retrieved May 15, 2015, from <http://www.deferredprocrastination.co.uk/blog/2012/minimum-bend-radius/#more-2743>

Funk, M. - Examples: Channelsender, Channelreceiver - <https://github.com/iddi/oocsi-processing>

Jenesis International. (2015) <http://jenesisproducts.com/how-is-battery-life-calculated/>

Porterfield, A. (2015). Curved laser bent wood. Retrieved May 15, 2015, from <http://www.instructables.com/id/Curved-laser-bent-wood/?ALLSTEPS>

Precision Microdrives Ltd. (2015) <https://catalog.precisionmicrodrives.com/order-parts/datasheet/308-100-8mm-vibration-motor-3mm-type>
Roving Networks. <http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf>
ST. (2013) <https://cdn.sparkfun.com/assets/f/6/1/f/0/LSM9DS0.pdf>

The Business Model Canvas in 2 minutes:. (2015). Retrieved June 9, 2015, from http://businessmodelgeneration.com/canvas/bmc?_ga=1.130332175.1832063013.1433859580

Thompson, C. (2008). Build it. Share it. Profit. Can open source hardware work?. *Wired*, 10, 08.

Tomczak, S. - Examples of sending MIDI data - <http://little-scale.blogspot.nl/2011/08/simple-examples-of-sending-midi-data.html>

Townsend, K. - Bunnyrotate.pde - <https://learn.adafruit.com/ahrs-for-adafruits-9-dof-10-dof-breakout/visualizing-data>

