

University of the Arts Helsinki

Sibelius Academy

Music Technology

Master thesis

**Resonata**

***– Interactive Sonification as a Creative Tool for  
Sound Art –***

Student:	Janne Storm
Studies:	Master of music

Helsinki, 21.04.2020

<b>Title</b>	<b>Number of pages</b>
Resonata – Interactive Sonification as a Creative Tool for Sound Art –	68
<b>Author</b>	<b>Semester</b>
Janne Storm	Spring 2020
<b>Degree programme</b>	
Music Technology	
<b>Abstract</b>	
<p>In this research I study data sonification and how to use it in interactive sound art. In the thesis I look into selected sonification applications and methods for utilising instrument- like interactivity in sonification. As part of this master's project a new software application Resonata is presented, and its features are explained in this text.</p> <p>Resonata is an interactive audio-visual installation built with Max graphical programming environment. It explores the artistic possibilities of sonification and presents methods for interacting with audio software and data processing via the computer game controller Gametrak. In the design of Resonata I've used my knowledge and experience of music technology, sound control devices and audio processing. I have applied techniques used in musical performance, such as feedback systems between the player and the instrument, to data analysis in form of interactive sonification, in order to create an artistic experience.</p>	
<b>Keywords</b>	
Installaatiot, mediataide, interaktiivinen taide, äänitaide, instrumentit, elektroninen musiikki	

### Acknowledgements:

I want to thank Marianne Decoster-Taivalkoski, Kalev Tiits and Andrew Bentley for their help and guidance with this project, Alejandro Montes -de Oca, Alejandro Olarte, Josue Moreno Prieto and Sami Klemola for teaching me the tools for making the software, and Hugh Sheehan for his help and support with the *Gametrak* -controller. Big thanks goes to Sibelius Academy, it's music technology department and everyone involved with it. I also want to thank my family, my wonderful spouse Paula Praktig for her help and support with the text, and my dear friends Tero Holopainen, Mirjami Holopainen and Ruu for their support. A special thanks to Essi Kaila for enabling me financially to focus on my studies at the Sibelius-Academy.

# Table of Contents

<b>1 INTRODUCTION .....</b>	<b>1</b>
1.1 MY BACKGROUND.....	2
1.2 RESONATA.....	4
<b>2 CONCEPTS.....</b>	<b>6</b>
2.1 SONIFICATION .....	6
2.1.1 <i>Introduction</i> .....	6
2.1.2 <i>Background</i> .....	6
2.1.3 <i>Sonification methods</i> .....	8
2.1.4 <i>Why sonify?</i> .....	9
2.1.5 <i>Is sonification Art?</i> .....	11
2.2 INSTRUMENT AND INTERACTION.....	11
2.3 GAME.....	14
2.3.1 <i>Introduction</i> .....	14
2.3.2 <i>Are computer games art?</i> .....	14
2.3.3 <i>The role of the controller</i> .....	15
2.3.4 <i>Games that have inspired this project</i> .....	16
2.3.4.1 <i>Everything</i> .....	16
2.3.4.2 <i>Hellblade: Senua's Sacrifice</i> .....	17
<b>3 MOTIVES AND GOALS.....</b>	<b>18</b>
3.1 ABOUT THE CHOICE OF PROGRAMMING PLATFORM.....	19
3.2 ABOUT MY DECISIONS IN DESIGNING THE SONIFICATIONS.....	19
3.3 ABOUT CHOOSING THE DATA SETS.....	20
3.4 ABOUT THE GRAPHICAL INTERFACE.....	23
3.5 CONNECTEDNESS.....	24
3.6 CONTROL.....	25
3.7 GESTURALITY .....	26
3.8 VISUALIZATION.....	27
3.9 INCLUSIVITY.....	28
<b>4 IMPLEMENTATION .....</b>	<b>29</b>
4.1 ACQUIRING THE DATA SETS .....	30
4.1.1 <i>An ultrasonic recording</i> .....	31
4.1.2 <i>A neural electromyograph readings of a myopathy patient</i> .....	32
4.1.3 <i>An electroencephalography readings of a sleeping subject</i> .....	32
4.1.4 <i>An electrocardiography signal from an intensive care unit -patient</i> .....	33
4.1.5 <i>An audification of an tectonic event</i> .....	33
4.1.6 <i>Air temperature measurements from Finnish weather station</i> .....	34
4.1.7 <i>Planetary coordinates of some bodies of the solar system</i> .....	35
4.2 IMPLEMENTING THE SONIFICATIONS.....	37
4.2.1 <i>Audification</i> .....	37
4.2.2 <i>Tonal processing</i> .....	38
4.2.3 <i>Applying an oscillator bank</i> .....	38
4.2.4 <i>Envelopes</i> .....	39
4.2.5 <i>Transient detection</i> .....	40
4.2.6 <i>Sonification of the Ephemerides</i> .....	40
4.3 THE CONTROL SYSTEM.....	40



4.3.1	<i>Gametrak</i> .....	40
4.3.2	<i>Routing</i> .....	44
4.3.3	<i>Parameter mapping</i> .....	48
4.4	GRAPHICAL INTERFACE .....	52
4.4.1	<i>Oscilloscope</i> .....	53
4.5	TRANSITIONS.....	54
4.6	MACHINE LEARNING.....	55
<b>5</b>	<b>THE INSTALLATION</b> .....	<b>56</b>
5.0.1	<i>Introduction</i> .....	56
5.0.2	<i>Setup</i> .....	57
5.0.3	<i>Reflections</i> .....	58
<b>6</b>	<b>CONCLUSION</b> .....	<b>60</b>
6.0.1	<i>What next?</i> .....	61
<b>7</b>	<b>BIBLIOGRAPHY</b> .....	<b>63</b>

## List of Figures

Illustration 1: Platonic solids photo: J.Kepler, Harmonices mundi, public domain. distributed under a CC-BY 2.0 license.....	7
Illustration 2: Spectrogram of the piezo microphone's frequency response, with an sine wave sweep up to 88kHz.....	29
Illustration 3: Visual presentation of EDF file in EDF browser.....	31
Illustration 4: Example of use of FMI API. The command includes API key, the requested measurement (t2m = air temperature), observatory id (101004 = Kumpula weather station), and time frame (11. - 16.2018). Download -message tells jit.uldI -object to write the data into defined file.....	32
Illustration 5: Example of the code returned by the API request.....	33
Illustration 6: Javascript for converting the timestamp in max.....	33
Illustration 7: Example of the code to parse the ephemeris files in Python. The export is restricted into 15000 samples at the time, since larger files slowed Max down too much.....	34
Illustration 8: Interpolation error modulating the waveform.....	34
Illustration 9: The orbits of the moons of Uranus distorted by the interpolation error. Screenshot from an oscilloscope, at playback speed of 104857600 % (or +20 octaves).....	35
Illustration 10: Gametrak PCB. REV2 can be soldered fairly easily to work with pc, REV1 is incompatible.....	39
Illustration 11: Gametrak -controller.....	40
Illustration 12: Diagram of Arduino micro's connections. Ground, power, and used input pins highlighted.....	41
Illustration 13: the Max patch that receives signals from Playstation controller and Gametrak.....	41
Illustration 14: Patch that interpolates between the stored snapshots of the control parameters.....	42
Illustration 15: The signal flow of the patch.....	43
Illustration 16: The "sonificationUnit" -sub-patch, presented here with mono -audiosignals for graphical simplicity, although the patch used in the Resonata is in stereo.....	44
Illustration 17: 3d parameter mapping -space. Each corner represents one of the 27 spaces where a snapshot is stored. the red lines represent the strings of the controller. ....	46
Illustration 18: Nodes -object returned an interpolated weight of each node as a list. ....	47
Illustration 19: patch used for trilinear interpolation.....	47
Illustration 20: Code for converting the Unix time ino Gregorian time.....	51
Illustration 21: Graphical interface of the Resonata, showing the oscilloscope, data's waveform with the playhead (vertical line), and the calendar date at the playhead.....	51
Illustration 22: Routing of the oscilloscope.....	52
Illustration 23: Signal chain of the machine-learning algorithm.....	53
Illustration 24: The Showroom.....	56

## 1 Introduction

This paper is the written part of a master's thesis. The other part of the thesis is an art project. This paper discusses the process of creating the project, its outcome, and also the questions and ideas that have evoked in the process.

Besides this written part, I would like to have the installation, as documented in the video <https://youtu.be/aXu0JvtnALo>, reviewed.

When I was in high school I read the “Dirk Gently's Holistic Detective Agency”, a sci-fi detective novel by Douglas Adams. In the book an alien spaceship has been orbiting the Earth for eons, collecting every bit of information about the planet, from weather systems to the wing movement of birds and shapes of the coastlines. All this data is collected to the ship's computer as a software program. In the end of the book the protagonists enter the spaceship and run the program. When the program is executed, the ship's speakers start playing Bach.

Sonification is a method to interpret data through sound. Sonification is usually not considered music or sound art.<sup>1</sup> Its function is to communicate information. My goal with the project is to develop techniques for applying interactivity and adding artistic expression to sonification. Therefore I would call my work sound art instead of sonification, since its function is to communicate an experience rather than information.

The artistic part of the thesis is called *Resonata*. It's an interactive audiovisual installation. *Resonata* was born out of the desire to study sonification, interactive music and instrument-like interfaces. These are combined in *Resonata* into a work that is at the intersection of a sound installation, an instrument, and a game. The installation is made for an audience of one to five

---

<sup>1</sup>Hermann & Hunt & Neuhoff 2011, 1

people, and it can be used by one user at a time. I will describe *Resonata* in more detail later in this text.

As *Resonata* is an interactive work, the artistic input is shared between myself and the audience. The audience's artistic input is to play the installation. My goal is to design the interface so that it is intuitive enough for anyone to internalize quickly. My artistic input is to select the data sets for sonification and to design the sonifications, visualizations, user interface, and the mappings for user control. Creating the user experience is a process that combines technical engineering and artistic creation.

I study the questions that have risen from this process through artistic-based research. Do the patterns and resonances that we can find in data sets collected from our surroundings have some inherent artistic value? How to apply artistic expression to sonification by adding interactive features to it? If we interact with the data as an active agent like an instrumentalist, what can be gained from the experience? And how to make these experiences accessible to the audience? Results of this research are presented partly in this text, and partly through the experience of the artistic work.

In the title of this text I use the term “Sound Art” as it's defined in Tate gallery's online glossary: as “Art which uses sound both as its medium (what it is made out of) and as its subject (what it is about)”<sup>2</sup>

## 1.1 My Background

One of my goals has been to explore areas that I haven't explored before. I have no prior experience in data processing (beyond digital audio processing), data analysis, or coding. My background is mainly in sound design and live music. I started playing guitar with local rock bands in school. From there I started slowly expanding to live electronics but found out that the experience of the instant and complicated feedback you can get from acoustic/analog instrument is hard to produce with electronic/digital devices. I started using

---

<sup>2</sup> Tate.org 2020

laptops in my live setup in 2008. Since then I have been trying to come up with ways to control the computer like an musical instrument, beyond mouse and computer keyboard. I have found the controllers designed for computer games, like joysticks, to offer surprisingly intuitive multi-modal interfaces for music performance.

In several groups that I have played with I have assumed the role of processing other people's playing with live electronics. These experiences have helped me in designing the interactivity between the data and user control, where they both share the control of the output.

I have had a little bit contradictory relationship between composing and performing music. To me compositional process is quite conscious, cognitive process. Playing live on the other hand is almost the opposite, where I try to free myself from the constrains of cognition, and go with my intuition instead. Data analysis is something that usually goes into the first, cognitive category. In this project I have tried to apply the power of intuition, like one found in live music, to data analysis in the form of interactive sonification.

I have been a casual gamer since I was a child, and I have always been interested in the development of game design and it's possibilities. Applications and interfaces for digital music systems is where my interest in gaming interfaces and game design meet with my love for music performance and generative composition (a music composition method where music is generated, at least partly, by system or algorithm). But where music applications usually require some amount of dedication to explore the depth of their possibilities, many game interfaces are designed to be easily internalized by a casual consumer, with a lot of attention given to the interface to be easily understood without being obtrusive.<sup>3</sup>

Since 2016 I have been doing music workshops for children of ages 8 to 12. In my work I have noticed that sometimes experimental approaches in music

---

<sup>3</sup>IronEqual, 2020

making can get the children more engaged in the process. Unusual sound sources like found instruments and novel controllers encourage to experiment, while traditional instrument can feel intimidating to approach, if the user has no prior experience with the instrument. These experiences have influenced my decisions when I have been designing *Resonata*.

## 1.2 *Resonata*

What follows is a description of the experience *Resonata* offers to the audience.

By the door leading to the exhibition space there's a note that says:

This is an instrument that plays information

The screen shows what is the signal you are hearing, and how much the signal is sped up/slowed down.

By holding the tethers and slowly moving your hands to different postures, you can explore the signals.

When the program finds a connection with another signal, you can transport yourself into that signal by pressing the pedal on the ground for 5 seconds.

In the room there's a continuous droning hum from the speakers. On the wall opposite the entrance an image is projected. It shows an oscilloscopic visualization of the drone, and is pulsing in sync with the sound. Below the oscilloscope there's a image of a waveform and a timeline. They tell the visitor that the waveform, the same one that we're listening, is made of ten years' worth of air temperature measurements that are sped up hundred million percent. In the middle of the room, on the floor, there's a controller about the size of a cigar box. Attached to the controller there's two strings connected to bracelets, and a foot pedal. When the strings are picked up, the sound in the speakers comes to life. It changes according to users movements. When taken to one direction, the fundamental tones in the drone are emphasized. When moved in opposite direction, the drone transforms into rhythmic beat. All this is generated from the data of the air temperature measurements in the wave file. If makes fast enough gesture of opening of closing their arms, the pitch of the sound changes, one octave at the time, up or down.

Suddenly the lights in the room get brighter, signalling that the program has found a match. The frequency spectrum of the sound the user is making has a strong correlation with the sound previous user made with another data set, that time with the planetary coordinates of the solar system. The program now suggests for the user to press the foot pedal. When pressed, the program stores the spectral information of the user's current sound as a reference for the future users to be compared with. When the pedal is released, The sound starts to increase in pitch, and the oscilloscope seems to be zooming out. The sound transforms into sonification of the planetary bodies, and the oscilloscope shows the planets orbiting the sun. The lights in the room are dim again. The exploration of the sounds can continue...

This was a user-scenario of *Resonata*, an interactive installation that lets the user explore selected data sets by the means of sonification. *Resonata* plays data sonifications as an ongoing loop. The user can control the sound of the sonifications using the tether controller, with physical gestures. As the user changes their posture or moves their hands, a new elements in the sound are revealed/explored. If the program finds matching patterns between the explored data set and some of the other data sets, it allows the user to “travel” into a new data set and continue exploring there.

*Resonata* is mainly done with Max, a graphical programming environment that I started studying for this project in 2017 in sibelius academy under the tutoring by faculty's teachers Josue Moreno Prieto, Kalev Tiits, Alejandro Montes -de Oca and Sami Klemola. Some of my decisions in programming *Resonata* have been based purely on my limitations as a programmer.

This paper is divided into six sections. You are now reading section one. In the second section I describe concepts that are related to my project, such as sonification, controller as an instrument, and interactivity in sound art. On the third I will discuss the decisions I have made in this project and the motives behind them. In the fourth part I go through the implementation of my ideas into a Max patch and working installation. The fifth part is about the execution

of the installation. On the sixth part I discuss the conclusions of the project and ideas how to extend the ideas born from the project. My studying method has been mostly “learning by doing”, and that reflects a little bit into how this text is written. Since some of my decisions are based on iterative trial-and-error process, The “motives” and “implementation” are sometimes pretty much the same thing.



## 2 Concepts

### 2.1 Sonification

#### 2.1.1 Introduction

“Hence it is no longer a surprise that a man, the ape of his Creator, should finally have discovered the art of singing polyphonically [per concertum], which was unknown to the ancients, namely in order that he might play the everlastingness of all created time in some short part of an hour by means of artistic concord of many voices”<sup>4</sup>

How would a weather sound like, if ten years were compressed into thirty seconds? Does it sing in unison with our orbit around the sun? What kind of pulse will our nerves make when time is slowed down?

The digital technology has given us access to an exponentially growing amount of data, a lot of which is publicly available. In my study I have wanted to explore possibilities of making music out of these data sets, such as movements of astronomical bodies, meteorological data and medical measurements, and try to sonify them so that the result would be aesthetically pleasing, but could also communicate something about the underlying dynamics in the data.

#### 2.1.2 Background

Exploring data sonically is an old idea. It can be traced at least as far back as *Musica Universalis*, a philosophical concept attributed to Pythagoras (born ~569 BCE), that sees (hears) the proportions in the movements of celestial bodies as musical harmonies. This idea played a unifying role in the development of the arts and sciences<sup>5</sup> and was taught as a part of *quadrivium* beside arithmetic, geometry and astronomy. In *Republic* Plato writes: "As the eyes, said I, seem formed for studying astronomy, so do the ears seem formed for harmonious motions: and these seem to be twin sciences to one another, as also the Pythagoreans say"<sup>6</sup>. In *Timaeus* Plato suggests that the

---

<sup>4</sup> Kepler 1619

<sup>5</sup> Worrall 2018

<sup>6</sup> Plato, 380BC.

*Platonic solids*, five polygonal shapes, form the ancient elements and therefore all matter.



**Illustration 1: Platonic solids** photo: J.Kepler, *Harmonices mundi*, public domain. distributed under a CC-BY 2.0 license.

In 1619 Johannes Kepler expanded these ideas in his *Harmonices mundi*, and proposed that the dynamics of solar system and orbits of the known planets are based on the ratios of the platonic solids. Because of technical limitations of the era, sonification stayed mainly theoretical until the 20th century, even though the idea of a sonar (SOund Navigation And Ranging) can be found already in Leonardo da Vinci's manuscripts<sup>7</sup>.

Nevertheless one can find also some old practical applications, like wind chimes. They can be seen as an early form of artistic sonification, since they communicate about the wind intensity through sound.

Geiger counter, invented in 1908, is one of the oldest practical applications of sonification, as is Optophone, invented in 1913 by Dr. Edmund Fournier d'Albe. Optophone was a device designed for visually impaired, that used photosensors to detect black ink on a paper and produced a series of notes depending on how much ink the device detected<sup>8</sup>. Artistic use of audification, the most direct method of sonification, has been discussed as early as 1923, in text written by painter, photographer and theorist László Moholy-Nagy, in discussion with Piet Mondrian about composing music by etching lines and curves directly onto disc. Similar ideas were brought up by invention of "talkies", or movies with sound. In 1929 composer Arseny Avraamov and inventor Evgeny Sholpo in Russia<sup>9</sup>, and in 1932 painter and avant-garde

<sup>7</sup> Effenberg, et.al. 2005, 29

<sup>8</sup> Jameson 1966, 1

<sup>9</sup> Smirnov 2011

animation film maker Oskar Fischinger in Germany<sup>10</sup> experimented with painting ornaments directly onto film as a soundtrack.

Technological limitations have been a big reason why sonification as a field has evolved so slowly. Especially with interactive sonification, where processing has to be done in real-time. Only through digital revolution have we got the processing power and means of collecting the data to truly explore the possibilities of sonification.

In 1992 the International Conference of Auditory display (ICAD) was founded by Gregory Kramer as a “forum for presenting research on the use of sound to display data, monitor systems, and provide enhanced user interfaces for computers and virtual reality systems”. In the first conference, sonification pioneer and composer Carla Scaletti provided definition for sonification, as “a mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purpose of interpreting, understanding, or communicating relations in the domain under study”.<sup>11</sup> Today sonifications are used in applications from topics such as chaos theory, bio-medicine, interfaces for visually disabled, data mining, seismology and mobile devices<sup>12</sup>, though it still seems to be a side-note in the field of data representation.

### 2.1.3 Sonification methods

Different methods of sonification can be defined by techniques employed, or by the function of the sonification, although the boundaries are indistinct and often overlapping. New methods for sonification are being developed constantly and definitions keep updating. I will here discuss two of them that are related to my work.

---

<sup>10</sup> Dombois, Eckel 2011. 305

<sup>11</sup> Barrass, Vickers 2011, 147

<sup>12</sup> Hermann, Hunt, Neuhoff 2011, 1.

Carla Scaletti classified sonification mappings by level of directness: level 0) audification, level 1) parameter mapping, and level 2) a mapping from one parameter to one or more other parameters.<sup>13</sup>

### **Audification**

Audification is the most direct method of sonification, where data points of periodic data set are translated directly into sound. This approach usually requires normalization of data values and sometimes frequency- or time-shifting to get the data to audible frequency. This method has been used for seismic data, for example, to identify seismic events. One could also say that every digital-to-analog -converter is an audification module, since it transforms the series of data points into an audio signal<sup>14</sup>. This method works especially well with time-series data, since unlike visualizations, sound inherently includes the element of time. It requires large data sets, since at least 20 data points per second are required for audification to be audible.

### **Parameter mapping sonification**

“Parameter mapping represents changes in some data dimension with changes in an acoustic dimension to produce a sonification”.<sup>15</sup> This can be simple binary data to trigger on/off message or an alarm, qualitative mappings such as pitch change representing a change in data values, or, since sound is inherently multi-modal<sup>16</sup>, a combination of several mappings.

#### *2.1.4 Why sonify?*

Our hearing has an amazingly fast pattern recognition and noise reduction capabilities. We have evolved to react instantly to hostile sounds in order to survive. We can separate a single melody out of a large orchestral

---

<sup>13</sup> Barrass, Vickers 2011, 147

<sup>14</sup> Dombois, Eckel 2011, 302

<sup>15</sup> Grond, Berger 2011, 363

<sup>16</sup> Walker, Nees 2011, 16

arrangement and a minute change in the timbre of a player, or we can follow a conversation in a noisy restaurant all with such an ease that even our best computer algorithms still can't match us.<sup>17</sup> Because of these capabilities, sonification can be a powerful tool for understanding the underlying dynamics and functions in the data.

In some cases hearing also surpasses visual observation. Comparing differences in two or more visual representations requires a lot of concentration, and can be exhausting over extended periods of time, while doing the same task by ear can be much easier and more precise task (like listening two simultaneous signals for differences).<sup>18</sup> Unlike visualizations, the sound is always linked to time. In sound, there is no “snapshot”, only the relationships and the dynamics between the snapshots.

Our hearing works on a logarithmic scale<sup>19</sup>. When played on a keyboard, keys A1, A2, A3 and so on sound evenly spaced, although the frequencies are exponential (55 Hz, 110 Hz, 220 Hz and so on). The same applies on sound levels. If we increase sound level in 6db increments, the increase in volume sounds linear even though the sound level is doubled on each increment.

We also perceive time logarithmically. When we get older, the time seems to flow more quickly<sup>20</sup>. When kindergarteners are tasked to place a number to line labelled with endpoints 0 to 10, they'll place 3 at about halfway.<sup>21</sup> Many visual representations (timelines, graphs, etc.) are linear and therefore sonifications may be more suitable to express exponential functions.

Our hearing has a strong connection to our emotions and experiences. The auditory cortex of the brain has been shown to have influential position within emotion-processing brain networks with “small-world” properties.<sup>22</sup> Sonification can help making the data more engaging by enhancing the emotional

---

<sup>17</sup> Hermann, Hunt, Neuhoff 2011, 3

<sup>18</sup> Worrall 2009 2-4

<sup>19</sup> Pigeon 2020

<sup>20</sup> Bruss, Reuschendorf 2010, 1

<sup>21</sup> Varshney, Sun 2013

<sup>22</sup> Koelsch, Skouras, Lohmann 2018

connection with the data at an intuitive level that can't be achieved by just looking at the numbers. The emotional qualities of sound have been used extensively in product design. Micro-transactions in mobile applications are designed to trigger sound that has positive emotional impact, successful alarm clock makes a stimulating sound, and so on.

### 2.1.5 *Is sonification Art?*

It is debated whether sonification should be called music. Carla Scaletti sees them as having different goals. For her, the goal of a scientific sonification is to present the data as accurately and as clearly as possible. "It's almost like you don't care that it was conveyed by sound. You're trying to hear that underlying structure; whereas for music, you do want people to be aware of the sound."<sup>23</sup> She uses data in her compositions, but she doesn't call them sonifications, calling them data-driven music, or just music instead. The distinction is however very blurry, and sometimes it's just a point of view of an observer that defines whether sonification is art or science. John Maeda (founder of the MIT Aesthetics and Computation Group) suggests that bridging the gap between art and science can lead to "a greater understanding and richness of human experience"<sup>24</sup>.

## 2.2 Instrument and interaction

"What makes experience genuinely satisfying is a state of consciousness called flow-- a state of concentration so focused that it amounts to absolute absorption in an activity. Everyone experiences flow from time to time and will recognize its characteristics: people typically feel strong, alert, in effortless control, unselfconscious, and at the peak of their abilities. Both a sense of time and emotional problems seem to disappear, and there is an exhilarating feeling of transcendence".<sup>25</sup>

As a performing musician, I have always loved the experience of playing an instrument, or experiencing live music. The instant and complex feedback-

---

<sup>23</sup> Scaletti 2017

<sup>24</sup> Beilharz 2005, 4

<sup>25</sup> Csikszentmihalyi 1990, 1

system that the player creates with the instrument is hard to achieve in electronic music with mouse and keyboard.

In march 2019 Composer and musician Hugh Sheehan, who was studying in Sibelius Academy at the time, introduced me to *Gametrak* controller (which is described later in more detail). It's a computer game controller that uses a pair of strings attached to players hands, with which one could then control the game. This controller can be converted into highly expressive musical interface. I started experimenting with applications for using it to control sonifications in 2019.

Many musicians claim that the experience of flow is the reason they play in the first place. The feedback system between the player and the instrument can be so complex and intuitive, that it evokes a meditative experience where the feedback system created is starting to self-oscillate, and music “just happens” effortlessly. One could argue that feedback systems and their synchronicity are essential to flow. Acoustic instruments require a continuous physical input, which engages the user to the feedback-system.<sup>26</sup> The fingers on the keyboard, the bow pressing against the string, these give the player a constant haptic sensation where the player is connected to the sound source. This constant physical contact can make the process to become autonomous, where the operation of the interface is done subconsciously, while mind free to concentrate on the task.<sup>27</sup>

In data analysis, Flow is usually not the priority. It's often important to get the exact values in the data, and for this purpose the best method might still be by looking at the numbers, lists or graphs. Most computer interfaces have evolved to accommodate this method. Computer interfaces have mainly evolved from the text-based command protocol<sup>28</sup>. Recently however there has been an emergence of alternative input methods such as speech control,

---

<sup>26</sup> Hunt, Hermann 2004, 3

<sup>27</sup> Hunt, Hermann 2011, 278

<sup>28</sup> Hunt, Hermann 2011, 278

touch displays, and VR-controllers, but many of them are still crude compared to intricacies of acoustic instruments.

The challenge in developing electronic musical interfaces is to create the continuous, multi-modal feedback-loops found in traditional instruments. The design of the input-to-output -mappings and the feel of the control is essential for the user to consider it to be an instrument. <sup>29</sup>

The development of Digital music interfaces bridge the gap between musical instruments and computer interfaces, and development of Virtual Reality -controllers give even more possibilities to gestural interfaces. Without physical tactile contact, however, these interfaces can be difficult to control.<sup>30</sup> The processing power it takes to make these feedback-systems to interact real-time is one reason why until the mid 1990's sonification methods were quite non-interactive, where user passively listened to the data after it was processed.<sup>31</sup>

Although musical instruments share many features with interactive sonification devices, they are usually not considered as such<sup>32</sup>. Instrument's function is to communicate expression, whereas sonification's function is to communicate information. Our understanding of instruments can however provide us with tools and practices for interacting with data and designing interactive sonification.

We understand our environment through interaction<sup>33</sup>. When playing music, we give meaning to the sound through interaction. This interaction can create a strong emotional connection with it's target. Creating an emotional connection with a subject of study enhances learning. We remember emotional

---

<sup>29</sup> Hunt, Hermann 2004, 3

<sup>30</sup> Hunt, Hermann 2004, 6

<sup>31</sup> Beilharz, K. 2005 , 2

<sup>32</sup> Hunt, Hermann 2011, 278

<sup>33</sup> Hunt, Hermann 2004, 1



experiences better than neutral ones.<sup>34</sup> Maybe by playing the data through a flow-like interaction we could gain some new insights of its functions.

## 2.3 Game

### 2.3.1 Introduction

Play: pelata, leikkiä, soittaa, toimia, näytellä, soida, leikitellä, peluuttaa.<sup>35</sup>

“Before he could recognize pyramids, cones and spirals in shells and crystals, was it not essential that man should first “play with” regular forms in the air and on the sand?”<sup>36</sup>

I wanted to target the project for an audience or a casual user. The design of it should ideally be very accessible. For this I have drawn inspiration from the design of video games. I've took a minimalist approach to the interface, limiting human interface to the two strings of the *Gametrak* controller and a foot pedal. I have left most details of the data out of the graphical interface. I've experimented with ways to map the parameters so that the user experience would be intuitive and easily internalized by the user, with minimal orientation. Besides the feedback that users get from the *Gametrak* controller, I also wanted to create a feedback system between them and other users. The program thus incorporates machine-learning algorithm that compares the sounds that different users create to find matching patterns in the sonifications.

### 2.3.2 Are computer games art?

On one of the earliest books on the subject, *The Art of Computer Game Design*, Chris Crawford states that computer games should be considered as a form of art. Crawford goes on to parallel games with cinema, which by the 1920's was also criticised as being just cheap entertainment for its commercialism and emphasis on technological advances.<sup>37</sup> An American art

---

<sup>34</sup> Canli,, et. al. 2000, 1

<sup>35</sup> Google translate

<sup>36</sup> Focillon, 1989, 163

<sup>37</sup> Falcão, et. al. 2010.

critic of the time, Gilbert Seldes however defended cinema as lively art. He saw cinema as democratic art for the average citizen. Through the 20<sup>th</sup> century the cinema grew to become the dominant form of narrative for consumers. In recent years however, computer games have started to overthrow cinema in terms of popularity. On The Guinness Book of Records the most profitable products of entertainment are no longer movies but computer games.<sup>38</sup>

Although games can be seen as natural evolution from cinema, my opinion is that the aspect of interactivity has more in common with performing music than watching a movie. The behaviour of the player, sometimes even compulsive repetition and fine-tuning of the performance to the point of virtuosity has much more in common with practicing musical instruments than consumption of cinema. Gamers also can experience flow while playing, just like musicians.

### 2.3.3 *The role of the controller*

The other aspect that bridges the gap between musical performance and computer games are the controllers. Some games, such as *Guitar Hero* by Harmonix utilize controllers that are representations of actual musical instruments. The drum controller for *Guitar Hero* is, in fact, technically identical to electronic drum kits used in music production. Some other games like *Rocksmith* by Ubisoft go even further, as it is played with actual electric guitar. One could say that these examples, where *Guitar Hero* controller functions as an instrument, or where an instrument functions as *Rocksmith* controller, are in the exact intersection of game controllers and musical instruments.

With the rise of mobile applications, many music production apps like *Figure* by Propellerhead or *iKaossilator* by Korg have recently been developed with game-like ease of use in mind. Many of them require hardly any orientation, and one can start creating songs instantly when the app is loaded. *Dreams*, a creative game developed for Playstation 4 includes a music production suite very similar to common DAW's (digital audio workstation) such as Logic or

---

<sup>38</sup> Guinness World Records 2013

Garageband, but is completely controlled with a Playstation controller and designed for instant gratification. However, *Dreams* is still foremost presented as a game, instead of a music production tool.

The role of the controller is often overlooked in games. Even though the game controller is the key element of human-computer interaction, its role in human experience has been relatively unstudied<sup>39</sup>. Basic game console controllers have stayed almost unchanged for twenty years while audiovisual capabilities and other aspects of gaming have become dramatically more complex. This stability has however enabled the developers to fine-tune the control interface and gesturality in the games, as the same controller has to cover a wide range of different game genres and actions. My view is that the designers of electronic instruments could learn a lot from game controller design.

#### 2.3.4 Games that have inspired this project.

Game narratives and mechanics have always been limited by the technology they rely on. In the early days of computer games, technology and resources greatly limited the experiences games could provide and the actions the player could perform. Nowadays advances in digital graphics and big production budgets allow for spectacles comparable to Hollywood movies, and the limits for artistic expression start to be more creative than technical.

Two games have particularly inspired me in designing *Resonata*. These are *Everything* by David O'reilly and *Hellblade: Senua's Sacrifice* by Ninja Theory.

##### 2.3.4.1 Everything

*Everything* is a simulation that combines philosophy, art and gameplay. The player can assume the role of an “object” or “being” in the game world. This object can be an animal, a plant, or a landmass such as an island, for example. The player can interact with other objects in the world, explore their surroundings, and “ascend” or “descend” into another objects or creatures, like other animals, planets, galaxies or microbes. While exploring the world, the

---

<sup>39</sup> Brown, et. al. 2010, 1

player also finds recorded lectures by philosopher Alan Watts. The game has no other goal than to listen to the lectures, explore the world, and connect with other beings in it. It inspired me to think of the possibilities of game mechanics and interactivity in artistic presentation of subjects such as large differences in scale, and connectedness of things.

#### 2.3.4.2 Hellblade: Senua's Sacrifice

*Hellblade: Senua's Sacrifice* is an action/adventure game set in a fantasy setting and including elements of puzzle solving and psychological horror. The game was developed with the help of neuroscientists, mental health specialists and people suffering from mental health problems to accurately depict the effects of psychosis. The game's sound design has a central role in this, as the game is constantly guided and commented by protagonist's inner voices. Many of the game's puzzles revolve around apophenia (the tendency so see patterns in unconnected things), as their goal is to find matching patterns that others couldn't see in the game environment. This game inspired me to think of the application of apophenia in artistic experience.

### 3 Motives and Goals

In an interview with Sean Carroll Steven Strogatz, a mathematician and professor at Cornell University, speaks of “mathematic impressionism”. By that he means that he likes to get inspired by nature, and then in mathematics do something that is much simpler, but somehow captures the essence of the natural phenomenon. Like an impressionist painter, who doesn't try to create a realistic presentation, but instead, with dots or broad strokes captures the heart of his subject.<sup>40</sup> To me, that resonates well with what I try to do with artistic research. With *Resonata* my goal has been mainly artistic. It is to give the user an intuitive experience. My goal is not to make the user understand the exact values and details in the data, but to give a feeling of understanding via appreciation, like the one we get when we appreciate music. One does not have to understand music theory, or know how the music is made to get an intuitive understanding of what the music is communicating.

In their *General Resonance Theory* Jonathan Schooler and Tam Hunt from The University of California claim that, at the core of it, everything is resonances of vibration<sup>41</sup>. The big bang, the planets of our solar system, sounds and sights, our bodies and thoughts are all just bundles of resonances interacting with each other. The theory has been criticised for being unfalsifiable, and therefore unscientific<sup>42</sup>. I'm fond of embracing the artistic liberty of being unscientific. In artistic research the concepts of verifiability and falsifiability aren't as simple as in many other fields of science. I find *General Resonance Theory* artistically inspiring, as that is essentially what sonification does to the analysed data. It puts it in the domain of vibration/resonance.

Pattern, synchronization, rhythm, resonance, feedback and harmony are different ways to describe ultimately the same thing. A steady pulse, when at audible frequency, produces a pitch. An audible pattern, or harmony, when pitched below the audible frequency, produces a beat. There is no harmony

---

<sup>40</sup> Sean Carroll's mindscape 2019

<sup>41</sup> Schooler, Hunt 2019

<sup>42</sup> Tan 2018

without synchronization, or no resonance without a pattern or feedback. One of my goals has been to emphasise this by using the same patterns in the data for producing both the rhythms and the harmonies in the piece.

### **3.1 About the choice of programming platform**

I decided to program *Resonata* with Max, which is a graphical programming application. Max is a software platform designed for audio production, and it is frequently used with sound artists and contemporary composers. With Max, it's possible to program your own audio effects and tools. Using Max doesn't require coding skills. Max also has a very active user community that helpfully provides solutions for a programmer in need. Max provided me with enough tools and possibilities to build my work while being simple enough for me to learn in the time it took to design *Resonata*.

### **3.2 About my decisions in designing the sonifications**

In *Resonata*, I've based all sonifications primarily on audification, because it keeps the original data mostly unchanged (as explained in chapter 2). I wanted the default sound to be a direct translation of the data. In Max this allowed me to easily synchronize all the data sets to a common timeline.

For tonal processing I have used *vb.stretch* Max external (a software extension that adds features to Max) by Volker Böhm. It uses a sound stretching algorithm based on Fast Fourier transform (FFT). I chose it because of the musicality of the *vb.stretch* object rather than scientific accuracy of the sonification. It reduces the noise in the data, emphasizes tonal attributes and produces rather musical sounding re-synthesis. It also allows for versatile real-time control of it's parameters, which is crucial for the interactivity and intuitiveness of the final experience.

Parameter mapping sonification can be seen as second layer sonification since the sound is not a direct result of the data, but of the mapping made by

the creator (as discussed in chapter 2). In parameter mappings I didn't want to use any "compositional black magic", like mapping data into predetermined musical scales or cues, since I feel it would distort the data the wrong way.

I've used parameter mapping to map sonic attributes of the audified signal to oscillators, and also to other sonic attributes, such as volume envelopes and transient detection. I've done this to bring rhythmic characteristics into the piece. The actions of the user are also mapped to the parameters of the sonification. All the pitches and rhythms however are derived from the original data/the audification.

When designing the mappings, I have tried to balance between user control and data integrity. For *Resonata* to be treated like an instrument, the sound has to be controlled fluently, but this to some degree distorts the data representation. Choosing which attributes are user-controlled, and to which extent, was one of the main challenges of the parameter mapping.

### **3.3 About choosing the data sets**

I wanted to create an experience where the user can explore vast time scales. Sonification allows for a huge scaling of time. For example a pulse of 60bpm can be played back almost 20,000 times faster and still be audible.

When selecting the data sets for the project, I tried to cover a wide range of time scales and sample rates, ranging from 192kHz to 0,000012 Hz, lengths ranging from eight seconds to hundreds of years. When the user "travels" from one data set to another, the playback speed smoothly transposes to the octave determined by the sampling frequency. This creates a feeling of "ascending" or "descending" to a new level.

In *Resonata*, the user also has the ability to change the pitch of the sonification by adjusting the playback rate one octave at a time in the range of

-10 - +35 octaves. For example, by going up fifteen octaves, we can experience the duration of a month in just slightly over a minute. Or seven octaves down, and create a song out of one second of data.

I've chosen the data sets based on personal interest, availability, ease in converting to wave files. I also wanted to cover a wide range of sample rates. Original idea was to have more data sets in the patch, but that would've required exponentially more processing in the machine learning algorithm, and that in turn would've slowed the computer too much. So I decided to limit the number of data sets to seven, for now.

I ended up using following data sets:

- An ultrasonic recording of a hard disk (sample rate 192kHz, duration 8 seconds)
- Electromyography of neuropathy patient (400Hz, 30 seconds)
- An audification of an tectonic event (147Hz, 4 hours)
- An Electroencephalography readings of a sleeping subject (100Hz, 8 hours)
- An Electrocardiography of intensive care unit -patient (125Hz, a week)
- Air temperature (0,00017Hz, 10 years)
- Planetary coordinates of major bodies of the solar system (0,00001157Hz, 350 years)

#### 3.3.1 Ultrasonic recording of a hard disk

I've always been fascinated by the idea that there's a whole world of sound just above our hearing limits. When my neighbour's dog, while helping me to write, suddenly raises her ears, I can't help but to think: what am I missing?

In most data sets that I've chosen, the main information is in large scale, i.e. below audible frequency. I wanted to include something from above our



hearing limits as well. For this I started to experiment with DIY -methods for ultrasonic recordings. I found out that the contact mic that I've made from a piezo disc could record signals up to the 92kHz that my interface (with 192kHz sample-rate) was capable of. I attached the contact mic to the hard disc drive of a computer, and recorded the hard drive as it loaded the Max patch of the project. In all fairness, this may not be thematically the most interesting data set in the list, and I will probably replace this with some new recording in the spring when all the insects wake up.

### 3.3.2 *Electromyography of Neuropathy patient*

Besides vast planetary systems and tiny ultrasonic snippets, I wanted to have something human. Physionet.org is a database under National Institutes of Health (NIH) that has a large collection of multimodal medical measurements. It includes electromyographs (measurement of electronic signals in muscles) of human neural activity. When slowed down, electromyograms of neuropathy patient have steady polyrhythmic beat, while ones from healthy subjects tend to sound more random. So I chose the neuropathy data set mostly for musical reasons.

### 3.3.3 *Electroencephalography reading of a sleeping subject*

Physionet.org database also includes Sleep-EDF sub database, full-night recordings of sleeping subjects. I found these sonically interesting, because unlike the rather static nature of some other data sets, these included changes in tone over time and sudden events that caught my interest. I found it also inspiring to imagine that I am listening to somebody's dream (no matter how imaginary that connection might be).

#### 3.3.4 *Electrocardiography of an intensive care unit patient*

The third data set from Physionet.org is an electrocardiography (measurement of electrical activity of a heart) of a hospital patient. I chose this one, because as for measurements from humans, it's fairly long (one week) for such a high sample rate (125Hz). It also has a steady drone that modulates slowly over time. This, and other more periodic events in the data made it interesting to me.

#### 3.3.5 *Audification of a tectonic event*

The “Drumbeat earthquake”, as its nickname implies, sounds surprisingly musical. It is a seismogram reading of Mt. St Helens volcanic eruption in 2004. Its regular earthquakes sound like a beat when sped up enough. This data set brought percussive elements to the piece. In terms of scale it also seemed to fit well between medical recordings and larger planetary data.

#### 3.3.6 *Air Temperature*

I included the air temperature data set because it's closely linked to earth's rotation and orbit, and because the data is available for long time periods with relatively high resolution. The steady pulse of daily and yearly cycles give the data interesting tonal characteristics.

#### 3.3.7 *The Solar System*

The largest (in duration) data set that I acquired is 350 years of planetary coordinates of the solar system, recorded once a day. I chose this, since I feel that the rotation and orbit of our planet is the one pulse that affects equally to everyone on the planet, like a universal sync clock. Data is also strongly tonal since the orbits are basically sine waves. This gives nice balance to the noisiness of the other signals.

### 3.4 About the Graphical Interface

I have reflected the ideology of simplicity in the design of the graphical interface as well, as I've left out most of the descriptive text and numerical values of the data. Passage of time is however expressed numerically as a calendar date, because I felt that it helps to understand the scale of the data sets. When travelling to a new data set, there's also text with a short description of the data on the secondary screen. Besides these, most of the visual feedback come in the form of oscilloscope of the audible signal. The lighting of the exhibition space is also controlled from the patch, so the colour of the ambient light depends on the data set, and the lights go brighter when a connection is found in the data sets.

### 3.5 Connectedness

By selecting data sets from different subjects, like measurements from human bodies, climate events and astrological movements, and enabling the user to “travel” from one data set to another, I wanted to give the user an experience of connectedness in which can find harmonies between seemingly unrelated things. That being said, although some aspects of the data sets, such as the orbit of our planet and the temperature changes in earth are clearly connected, I don't claim that there would necessarily be any real causality between all the data sets.

In his paper “*Apophenia as a Method—Or, Everything Is Either A Metaphor Or An Analogue Computer*” Dan Lockton, an Assistant Professor and Chair of Design Studies at Carnegie Mellon University discusses the use of apophenia as a creative tool.<sup>43</sup> Apophenia means the human tendency to perceive patterns and connections in unrelated things. Although we all have the natural tendency to look for meaning in things, strong apophenia can be seen as a symptom of schizophrenia. Dan Lockton however suggests that apophenia can be used as an inspirational device for generative creativity. Apophenia is

---

<sup>43</sup> Lockton 2018, 1

also described as openness to experience and a tendency to be imaginative and curious. Studies have shown that genetic risk scores for schizophrenia predict creativity, and the siblings of schizophrenia patients work in creative field more than average.<sup>44</sup> Both the creative outside-the-box-thinkers and the pathological schizophrenics seem to have the tendency to see patterns, like faces, in random visual data.

I am fascinated by this area between creativity and psychosis. To me it resembles the area between science and art. In *Resonata* I wanted to explore these areas by allowing the user to move from unaltered, naturalistic audification of the subject to more and more processed interpretation that emphasises the perceived patterns and connectedness in the data. So the farther from the “truth” the user moves, the more the game rewards him with the perceived connectedness.

### **3.6 Control**

When I started to design the interface, my first idea was that there would be a screen with menus and sliders, where the user could dial in the preferred values and execute the sonification. The more I experimented with it, the more I felt that the interaction should be in real-time to truly engage the user. When experimenting with real-time approaches I realized that in order to make the interaction as seamless as I wanted it to be, I would have to reduce the amount of control into something where every parameter could be accessed simultaneously.

*Playstation 4* controller has eight continuous parameters, and 15 boolean controls that can be accessed somewhat simultaneously. I managed to get some fairly good results in mapping the *Playstation 4* controller into *Resonata*. With it it's possible to access all the user parameters without taking my eyes away from the screen. Changing parameter was no longer distracting from the listening process, like it was when dealing with dials and values on the screen.

---

<sup>44</sup> Power, et. al. 2015

User still needed some orientation to internalize the functions of all the 15 buttons. Using the *Playstation* controller wasn't very immersive for this application.

When I started to experiment with *Gamettrak* controller I realized that its physicality can bring a whole new level of immersion into the project, since the user has to utilize their whole body. It's also very engaging. By just holding the strings of *Gamettrak*, the user is already actively using it. The user is also constantly controlling all the parameters, unlike with the *Playstation* controller where one has to move the fingers to different places in order to access all the buttons. Even though the *Playstation* controller might be easier to learn than, say, a violin, it's hard to beat the simplicity of the *Gamettrak*. If you just can move your arms, you already have access to all the controls it has.

Using *Gamettrak* meant, however, that I had to get rid of all the controls that I had mapped to the buttons. Some of these, like selecting the data sets, I automated completely. Others, like controlling the playback speed, I mapped into specific physical gestures like opening or closing arms quickly. I also added a foot pedal which controls saving and moving between the data sets. Designing the interface was balancing between depth of control and ease of use. I had to leave out some parameters from the mapping scheme to make the system more approachable.

While designing the interface, I have found many features of the *Gamettrak* that could be highly usable in other musical and performative applications beyond *Resonata*, for example dance or eurythmic practices. I wish to study this further in the future.

### **3.7 Gesturality**

Gestures can be defined as body movements that involve meaning. Musical gestures can be categorized for example as sound-producing, communicative, sound-facilitating and sound-accompaying<sup>45</sup>. With *Gamettrak* controller the

---

<sup>45</sup> Jensenius, et. al. 2010, 2

sound-facilitating gestures can also become communicative, since it allows big, visible movements. This can't be said of the aforementioned *Playstation* controller, which otherwise share much of the same functionality.

When I designed the control scheme for *Resonata*, the main idea was that the physicality of the control gesture should correlate to the sound. The effect I have tried to create is that when the user opens their arms, the sonic content gets richer, and even more so when the arms are raised. Opening the arms “opens the sound”. When the hands are brought close to chest, the sonification gets simpler. The changes in the sound happen instantly. When the strings of the *Gametrak* are held at chest height, *Resonata* plays the unaltered audification. When the user moves their hands to any direction from there, more processing is applied to the sonification. When the right hand is raised up, more harmonic content is introduced. When the left hand is raised, more of the oscillators are mixed into the signal. Moving hands forward and backward introduces high and low pitched material, respectively. Moving hands to left and right affect the attack and release envelopes of the oscillators, the frequency of their triggers, and the window-size of the FFT processing (as explained in the chapter four). All these changes are combinations of several parallel parameter changes, and these examples are a simplification of the process.

The gesture of opening or closing arms quickly raises or lowers the playback rate of the program, respectively. These gestures are in fact the only ones that the program recognises as distinct gestures. The meaning of the other gestures of the user is found and defined by the way they transform the sound.

The nature of the interaction is highly bi-directional, since the data itself also strongly dictates the aural output. This means that even slight changes in gesture can cause big changes in sound. This also means that the gestures used in one data set don't necessarily give the same aural results when used with another data set. I feel that this unpredictability adds to the exploratory charm of the installation.

By removing all the visible parameters and values from the interface, I'm making the user to interact directly with the sound and the visualization. This makes the interaction more intuitive, and brings it closer to those found in traditional instruments (as discussed in chapter 2.2).

### **3.8 Visualization**

Besides the audio output, I wanted to give the user a visual feedback of the data as well. I have based the visualization on a digital oscilloscope (a device for displaying signals on a two-dimensional plot as a function of time), since it retains the waveform of the original signal and thus has a strong connection with the sound. The visualization has several oscilloscopes superimposed on top of one another, and the audio is divided between them. The oscillators of the patch for example have each their own dedicated oscilloscope. This way their waveform is displayed without it being modulated by other signals and the visualization of individual elements is made more visible, while still retaining strong correlation with the audio. In case of the planetary coordinates, the use of several oscillators allowed for displaying the model of the solar system with them. The colours in the oscilloscope are modulated by the *Gamettrak* controller, and do not represent aural properties of the data. The colours are merely for visual effect.

### **3.9 Inclusivity**

The installation is made to be used by the audience of different ages and backgrounds. Because of this I wanted to keep the simplicity in mind. Even though the subject is abstract, and there are constantly dozens of parameters to be controlled, I don't want to overwhelm the user. In game design the designers can create an evolving learning curve, introducing new features and mechanics as the player progresses. *Resonata*, however is designed to be an installation piece, where many users will only use it for a few minutes. That's why I wanted all the features to be instantly on the user's reach. The *Gamettrak* controller is very suitable for this, because just picking up the strings

puts the user in control of the patch. From there on, all one needs to know is how to move. In early iterations of the patch I had several menus and parameters for the user to adjust, but I felt that that activity disconnected the user from the experience, as they had to be oriented with each action and function. So instead I've either incorporated those actions into the gesturality with *Gametrak*, or automated them completely. In the final iteration, the only button to interact with is the foot pedal that allows for switching the data sets.

I thought about few options for how the user experience would play out. One idea was to have several data sets playing together and interacting with each other, but that would've required almost a complete rewrite of the patch. I also thought about treating different data sets as “game levels”, and the goal would've been to go through them in some predefined order. I ended up with something more generative, however. In *Resonata* the order in which one can move from one data set to another is determined by an algorithm that looks for similarities in the data sets. Player's sound is constantly compared to “sonic snapshots” that other players have taken from other data sets. The strongest match will be the data set that the user can transition into. These sonic snapshots will also play in the exhibition space when nobody's controlling the patch. By this I wanted to create some interaction between the users.

Is *Resonata* a game? It lacks many aspects of game design, such as pre-programmed rewards that would keep the player engaged, goals for the player to fulfil, or any ways of completing the “game”. However my favourite activity in many large sandbox-style games is to just wander around aimlessly without any goals or progress and just immerse myself into the game. I would therefore call *Resonata* a sonic sandbox instead of a game, data-driven sound art or instrument instead of sonification.



## 4 Implementation

In this chapter I discuss the process of building the Max -patch. I don't explain every operation in the program, only the ones relevant to this thesis.

I started working on *Resonata* in the spring of 2018. First by learning Max programming, and studying the subject of sonification. I have relied heavily on “Learning by doing”, and some early programming choices seem to me now quite crude and/or redundant. The first step was acquiring the data sets and converting them into wave files. Methods for this varied depending of the format of original data. The next step was to create the sonifications, and their controls, and to modify the *Gamettrak* controller to function with the Max patch. The final step was designing the user interface and user experience, including graphics, interaction, and applying the machine-learning algorithm.

### 4.1 Acquiring the data sets

For *Resonata* I acquired seven data sets that are publicly available on the internet. I selected the data sets based on personal interest, availability and suitability in converting to wave files (in order to audify them/make them audible). With each data set I have also recorded the timestamp (a digital marking of the time when the data has been measured) of the data set and the sampling frequency in order to synchronize the data sets together, as I wanted to be able to present them on a common timeline.

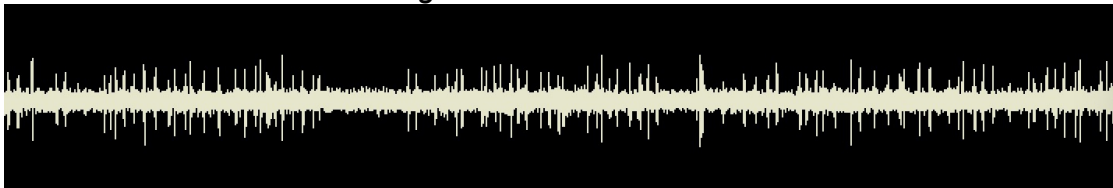
In order to write the data values into audio buffers, I used Max to convert the files. The conversion methods varied depending on the original file type of the data. When I had converted the data into text files, I scaled the data values to -1. – +1. scale. Then I programmed Max to read through each of them and to write them into an audio buffer, one after another. Max is not the most optimal tool for this kind of data processing. For Max to be able to read through all the data values, conversion of large text files needed thousands of iterations, and this slowed the program down on each iteration. For this reason I had to

convert large files in small parts, rebooting the software each time. This meant that the conversion of some of the data sets took several days.

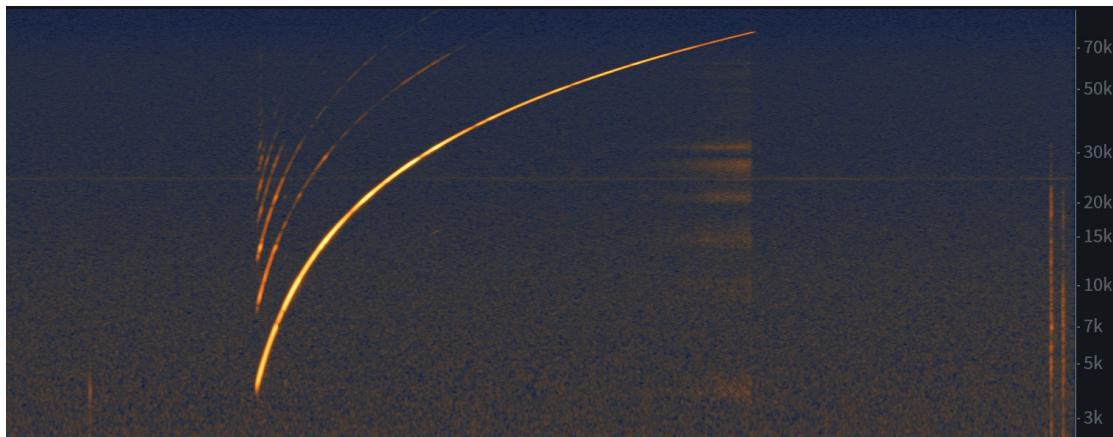
The data sets I used in the final work are:

- An ultrasonic recording of a hard disk drive
- A neural electromyograph readings of a neuropathy patient
- An Electroencephalography readings of a sleeping subject
- An electrocardiography signal from an intensive care unit -patient
- An audification of an tectonic event
- Air temperature measurements from Finnish weather station
- Planetary coordinates of some bodies of the solar system

#### 4.1.1 An ultrasonic recording

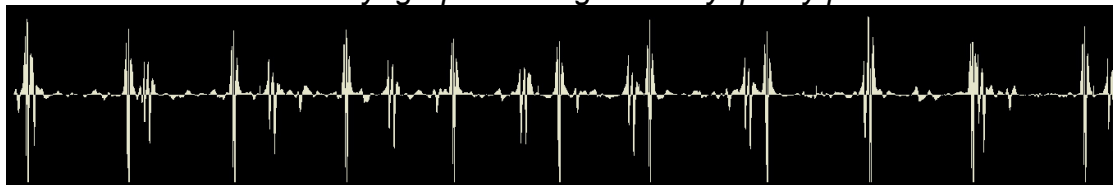


This is the only dataset that I have made myself. I made a contact microphone from piezo disk that could record signals up to 92kHz. I experimented with various sources for signals. For example I attached the piezo to the leaves of my houseplants, but neither my recording nor gardening skills were sufficient for any communication. I ended up attaching the disk to the hard disk drive of my old desktop computer. When running software, the physical movement in the disk generated very high pitched signals. For the recording I loaded a copy of the patch in the computer and recorded the hard drive when it loaded the program. I cut out the signal below 20kHz with a low pass filter, although some data remained near the cut off frequency.



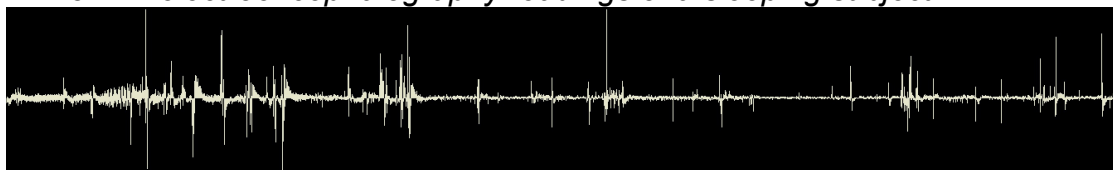
*Illustration 2: Spectrogram of the piezo microphone's frequency response, with an sine wave sweep up to 88kHz*

#### 4.1.2 A neural electromyograph readings of a myopathy patient<sup>46</sup>



Electromyography is a technique for detecting and recording electronic activity in muscles. This data set, as with two other electrographs is obtained from physionet.org website. It is originally recorded at 50kHz, but downsampled to 4KHz by the creator of the data set, since there is not much information above that frequency. The length of the recording was approximately 30 seconds. I chose this data set because of the interesting and quite musical polyrhythm the data has. The data set was provided as a time-value pair text file, and was easy to convert into audio buffer.

#### 4.1.3 An electroencephalography readings of a sleeping subject.<sup>47</sup>



<sup>46</sup> Goldberger AL, et. al. 2003.

<sup>47</sup> Kemp, et. al. 2000

Physionet.org has a database called Sleep-EDF. It contains multimodal measurements of sleeping subjects, recorded at 100Hz, with durations of approximately eight hours. The data is packed into a EDF (European Data Format) files, that contain measurements of electrical activity of the brain (electroencephalography), corneo-retinal standing potential that exists between the front and the back of the human eye (electrooculography), and electromyography of submental chin muscles. For *Resonata* I sonified the electroencephalography data, based on personal aesthetic preference. EDF files can be opened with a free application called EDF-browser, which enables exporting to text file as time-value pairs, which I then wrote to an audio buffer.

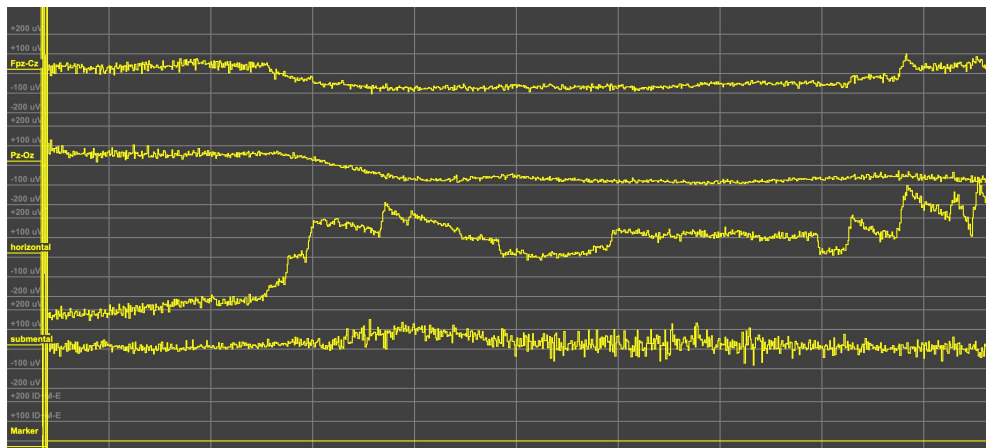


Illustration 3: Visual presentation of EDF file in EDF browser

#### 4.1.4 An electrocardiography signal from an intensive care unit -patient.<sup>48</sup>

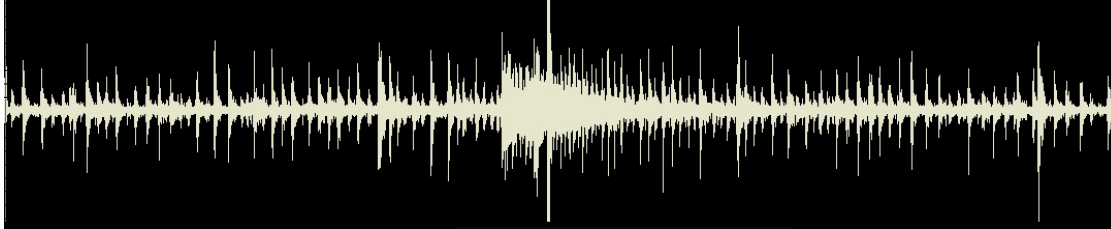


This data set, as the previous one, was obtained from physionet.org website as an EDF file. It's a part of MIMIC database, developed by MIT lab for computational physiology, and includes data recorded from intensive care unit patients. Data includes periodic measurements recorded from bedside monitors and clinical data from patient's medical records. For *Resonata* I used

<sup>48</sup> Johnson, et. al. 2016

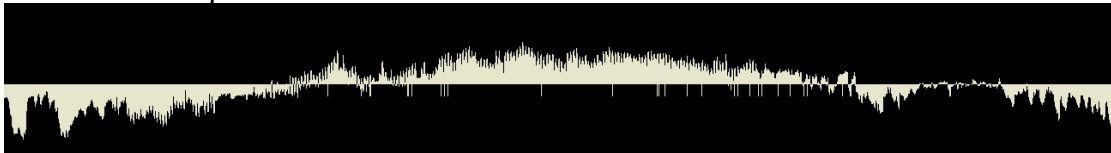
recording of electrical activity of the heart (electrocardiography) recorded at 125Hz. The duration of the recording is about 7 days.

#### 4.1.5 An audification of an tectonic event.<sup>49</sup>



This data set is a seismogram recording of Mt. St Helens during the volcanic eruption in 2004, audified by Debi Kilb. Recorded at 147Hz, with duration of four hours. It's called a “drumbeat earthquake”<sup>50</sup>, which also hints why it was chosen for *Resonata*. As this was already converted into wave file, the implementation was straightforward.

#### 4.1.6 Air temperature measurements from Finnish weather station.<sup>51</sup>



This data set was obtained from Finnish Meteorological Institute's (FMI) open data portal, that provides a large collection of real-time measurements, time series and forecasts. FMI provides an application programming interface (API) that can be used to access their data once user has requested a personal API key. With the API, it was possible to download data directly into Max programming environment, and convert it into a wave file. For *Resonata* I used air temperature measurements recorded from Kumpula weather station at 0,0017Hz (once in ten minutes) from 1.1.2009 to 1.1.2019.

---

<sup>49</sup> Kilb, et. al. 2012

<sup>50</sup> Kilb, et. al. 2012

<sup>51</sup> Finnish meteorological institute 2019

```
download
http://data.fmi.fi/fmi-apikey/JANNENAPIKEY/wfs?request=getFeature&storedquery_id=fmi::observations::weather::timevaluepair&parameters=t2m&fmsid=101004&starttime=2018-01-11T00:00:00Z&endtime=2018-01-16T00:00:00Z
jit.uddl @percent 1 @report 1 @defaultdir Sierra/Users/janne/Music/opinnayte
```

*Illustration 4: Example of use of FMI API. The command includes API key, the requested measurement (t2m = air temperature), observatory id (101004 = Kumpula weather station), and time frame (11. - 16.2018). Download -message tells jit.uddl -object to write the data into defined file.*

The request returns an Xml file with time-value pairs, where time is expressed in calendar date format. In order to write the data into audio file, I needed to convert the time protocol of the file into Unix time format.

```
<wml2:MeasurementTimeseries gml:id="obs-obs-1-1-t2m">
  <wml2:point>
    <wml2:MeasurementTVP>
      <wml2:time>2019-12-17T22:40:00Z</wml2:time>
      <wml2:value>4.9</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
  <wml2:point>
    <wml2:MeasurementTVP>
      <wml2:time>2019-12-17T22:50:00Z</wml2:time>
      <wml2:value>5.0</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
```

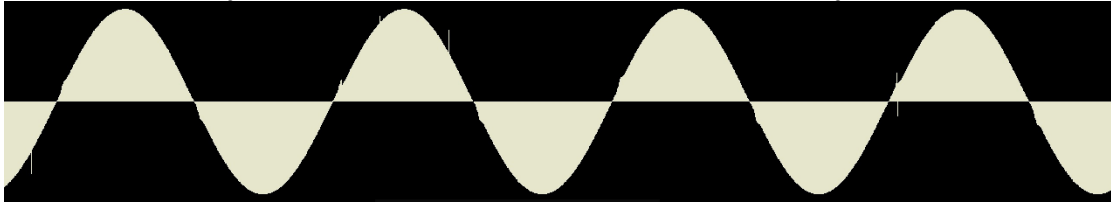
*Illustration 5: Example of the code returned by the API request*

```
outlets = 1;

function toTimestamp(year,month,day,hour,minute,second){
  var datum = new Date(Date.UTC(year,month-1,day,hour,minute,second));
  outlet(0, datum.getTime());
}
```

*Illustration 6: Javascript for converting the timestamp in max*

## 4.1.7 Planetary coordinates of some bodies of the solar system.



Nasa's Jet Propulsion Laboratory (JPL) offers on their website a large collection of ephemeris data, which tells the trajectory of astronomical objects. JPL Ephemerides are generally created to support spacecraft missions to the planets<sup>52</sup>. For *Resonata* I obtained the coordinates of the planets of the solar system, and their major satellites. The sample rate of the data is 0,000012Hz (once a day) and the duration is about 300 years (from 1.1.2000). The data is available in x-y-z coordinates, as three time-value pairs. The files are in BSP file format (file format used by Nasa for their ephemerides files). “Jplephem” python (a common programming language) package created by Brandon Rhodes<sup>53</sup> can load these files and export the data values into text file. For *Resonata* I used x and y coordinates of the objects converted into two audio files, although the y coordinate was only used for visualization purposes (to draw the orbits into oscilloscope). I also recorded relative amplitude of each data set, in order to represent them visually on scale.

```
import numpy as np
from jplephem.spk import SPK
n = 9
z = 'pluto'
np.set_printoptions(threshold=np.inf)
kernel = SPK.open('de430.bsp')
filename = "%s%s.txt" % (z, n)
text_file = open(filename, 'w')

startdate = 2440587.5
amount = 15000
a = np.arange(0, amount)
b = amount * n
time = a + b + startdate
index = (b + startdate - 2429630.5) * 86400
position = kernel[0,9].compute(time)

text_file.write("Amount " + str(amount) + "\nTimestamp " + str(index) + "\n " +
np.array2string(position))

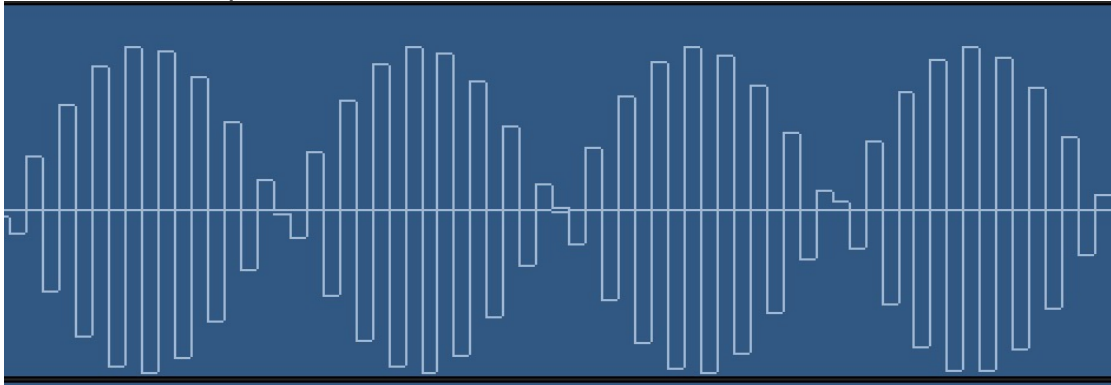
#print(kernel)
print(index)
print('done')
```

**Illustration 7: Example of the code to parse the ephemeris files in Python. The export is restricted into 15000 samples at the time, since larger files slowed Max down too much.**

<sup>52</sup> Giorgini, et. al. 2018

<sup>53</sup> Rhodes 2020

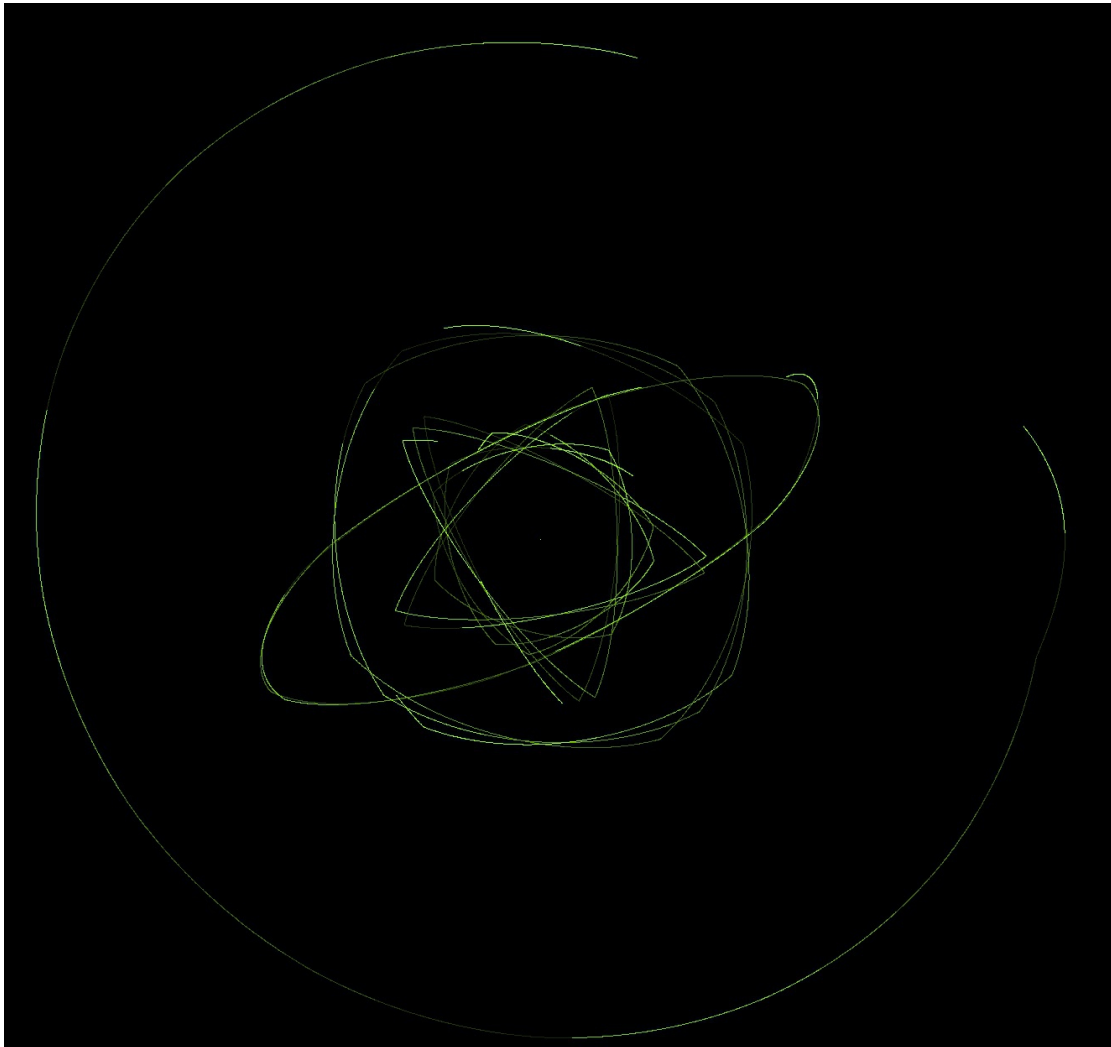
While playing back the audified ephemeris files and looking at them through oscilloscope, they presented an interesting feature. Some of the satellites in the solar system have relatively short orbit times. Major moons of Uranus, for example, had orbit times of only 1.4, 2.5, 4.1, and 8.1 Earth days. Because the measurements in the ephemeris files are taken once a day, this creates a substantial interpolation error, that can be seen in the wave files of the moons.



*Illustration 8: Interpolation error modulating the waveform*

Although this error distorts the information in the data, I found it artistically, or “poetically” pleasing, since the error synchronizes the orbits of the moons to earth's rotation. When viewed through oscilloscope, the signals of the moons of Uranus form, instead of circular orbits, interesting geometric shapes.





*Illustration 9: The orbits of the moons of Uranus distorted by the interpolation error. Screenshot from an oscilloscope, at playback speed of 104857600 % (or +20 octaves)*

## 4.2 Implementing the sonifications

I will here discuss the methods I have used to sonify the data sets. The routings and interaction between different parts of the patch are discussed in more detail later in this section.

### 4.2.1 Audification

All the sonifications in *Resonata* are primarily based on audification (as defined in chapter 2.1.2). All the data sets I use are first written into audio files, which are then processed and modulated with other sonification methods and user interaction. All the audifications are synchronized to the same timeline. The direct audification can be modulated by changing it's amplitude or pitch in

octaves. I've restricted the pitch shifting to octaves so that it would retain more relation to the original data, or "real-time". The shorter data sets are looped in order to make them playable on a wider range. Although this distorts the original shape of the data, it gives more expressive freedom to the user.

#### 4.2.2 Tonal processing

In order to make the audification audible on extreme playback rates, and to apply noise reduction and pitch detection to the data, I have processed it with fast Fourier transformation (FFT) -algorithm. For this I used a Max external object "*vb.stretch*", made by Volker Böhm. *Vb.stretch* is intended for "extreme time stretching"<sup>54</sup>, and this made it suitable for sonifying files that are played back at very slow speeds. It works by taking a sample of defined window size from the buffer, and resynthesizing the most prominent frequencies in it. This object also allows for expressive modulations to the signal, and has musically pleasing sound (at least for my taste). The FFT analysis is first applied to the original file, which is then resynthesized, and played back at the rate according to the playback speed of the patch, but without the playback speed affecting the pitch of the resynthesized sound. This way one can listen the tonal content of the data even if the original signal is below the audible frequency.

The patch utilizes four instances of the object, each performing the FFT at different window sizes for allowing wider range of tonal/temporal definition. The output signal can be then mixed from these four signals. Spectral threshold parameter can be applied to the analysis for noise reduction, and to emphasize tonal characteristics of the signal. The resynthesized signal can also be pitch-shifted in octaves by the user. *Vb.stretch* -object has also a "frame to frame interpolation" -parameter, that can be used to smooth out the transitions in the signal.

#### 4.2.3 Applying an oscillator bank

The resynthesised signal from the *vb.stretch* object is fed into a frequency peak detection object, a Max abstraction called *zsa.freqpeak~* by Emmanuel

---

<sup>54</sup> Böhm 2018

Jourdan and Mikhail Malt. This is based on “Spectral modelling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition”<sup>55</sup>. This object makes another FFT analysis to the signal, and outputs a list of frequency peaks with their amplitudes and frequencies. This list is then used to drive a bank of seven oscillators. The list of frequencies define the pitches of the oscillators. The list is also shifted down several octaves into sub-audible frequency to be used as triggers for the same oscillators for setting their pulses. For example, if a signal of 100Hz is fed into an oscillator, The oscillator would be triggered once every 5,12 seconds (100Hz minus nine octaves), with audible frequency of 100Hz. The audible frequency, and the pulse of the trigger can be modulated in octaves by the user. The attack and release envelopes of the oscillators are also controlled by the user. The oscillators have four waveforms, sine, triangle, rectangular and saw, which are determined by the noisiness of the `vb.stretch`'s output signal. The noisiness is detected with “`analyzer~`” -object, FFT -based perceptual analysis external by Miller Puckette and Tristan Jehan.

For purely artistic effect, a very slight modulation is applied between the left and right channels of the oscillator signal, in order to make the stereo image wider and the visualization in the oscilloscope more two-dimensional. The oscillator bank is based on the Max patch “Sev” by Katsuhiko Chiba.

#### 4.2.4 Envelopes

In signal processing envelopes are functions that modulate the signal over time. The patch has a volume envelope that is fed by the dry audification signal. This envelope modulates amplitude of the output audio signal of the patch. The user controls how much of this envelope is applied to the signal.

---

<sup>55</sup> Serra, Smith 1990

#### 4.2.5 *Transient detection*

The patch includes a transient detection algorithm that listens to the dry audification signal. This triggers an oscillator that emphasizes percussive elements in the audification.

#### 4.2.6 *Sonification of the Ephemerides*

Because the orbits of the celestial satellites are basically sine waves, the `vb.stretch FFT -external` doesn't really do anything to it. The data I've used is also two-dimensional (x – y coordinates of the satellites), mostly for visual purposes, and so the sonification is treated a bit differently. The x-coordinate of the ephemeris is used for audification. Each satellite also trigger an oscillator when completing it's orbit, with frequency relative to their orbit frequency but octave-shifted to audible level. The attack-release envelopes of these oscillators are controlled by the user.

### 4.3 **The control system**

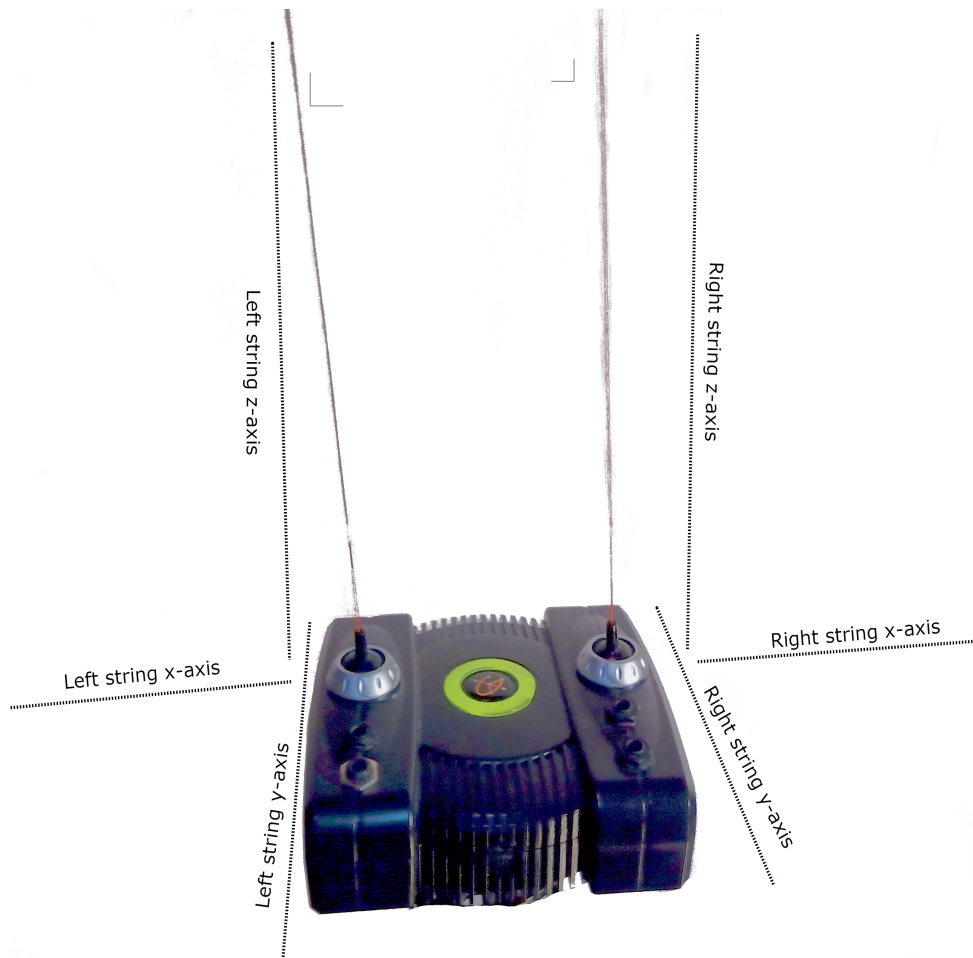
I've wanted to experiment with ways to create a connection between the sonified data and the user. The choices I have made in designing the patch have in many cases been based on immersion and ease of interaction.

#### 4.3.1 *Gametrak*

Haptic feedback is important in musical controllers. I have done some experiments using Ipads and Nintendo Wiimote controllers to control computer music, but the absence of haptic feedback requires a lot of concentration for precise control.

In 2019 Hugh Sheehan held a workshop in Sibelius Academy where he introduced me to a *Gametrak* game controller by In2Games. *Gametrak* is a controller originally made for golf games. It features two strings that the player holds in his/her hands. By moving her hands the player can control the game (to swing a golf club, for example). The device is now discontinued, but it has

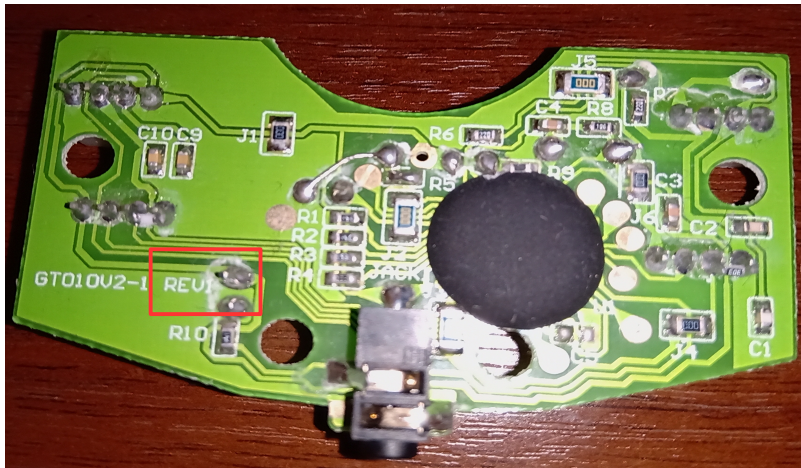
gained some reputation among experimental musicians. Even though the device is somewhat crude compared to today's VR technology, it's still more precise than most midi controllers. Holding physical strings also adds more haptic feel to the interface. Also the sheer physicality of the control adds a haptic feedback of it's own. The *Gametrak* also allows you to control six different continuous parameters simultaneously (both strings have potentiometers for X, Y and Z axis), which is more than with most midi controllers. This kind of controller that uses gesture for interaction has a surprisingly easy learning curve and intuitive interaction.



**Illustration 10: Gametrak -controller**

The strings

The *Gametrak* controller has different platform-specific versions. The ones that work directly with computer are getting exceedingly rare, but I managed to get one that is originally made for Xbox game console. For Xbox there exists two revisions of the controller PCB (printed circuit board), unfortunately the one that I got wasn't compatible with my mac.



**Illustration 11: Gametrak PCB. REV2 can be soldered fairly easily to work with pc, REV1 is incompatible**

It is possible however to replace the original PCB with Arduino electronics platform. Arduino boards are available in different versions. For the project I chose Arduino Micro -model, because it's small enough to fit into the case of *Gametrak*. *Gametrak* has three potentiometers for each string (X, Y, Z), so I needed to get six signals into Arduino. Arduino micro has 12 analog inputs. I connected The signals from *Gametrak* into analog inputs a0 – a5, And ground and 5v -cables from *Gametrak* into respective inputs in Arduino.

For foot pedal I added one digital input to arduino and connected it to a female jack that a attached to the base of the *Gametrak* -device.



**Illustration 12: Diagram of Arduino micro's connections. Ground, power, and used input pins highlighted**

To get the signals from Arduino to Max, I used “Arduino2Max” -Arduino package written by Daniel Jolliffe.<sup>56</sup> It enables Max to see Arduino signals as serial port data. I also used part of his Max patch to help route the signals into my patch. For prototyping I also added the possibility to switch between Playstation 4 -controller and *Gametrak*.

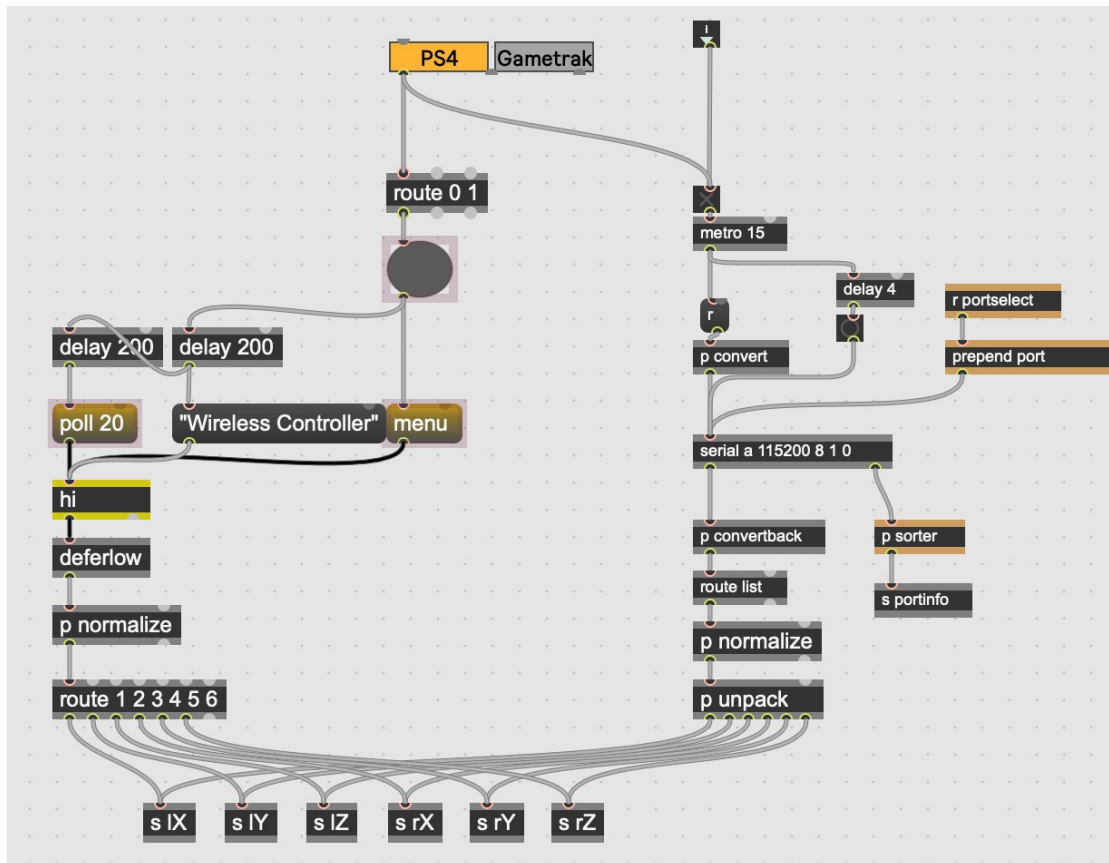
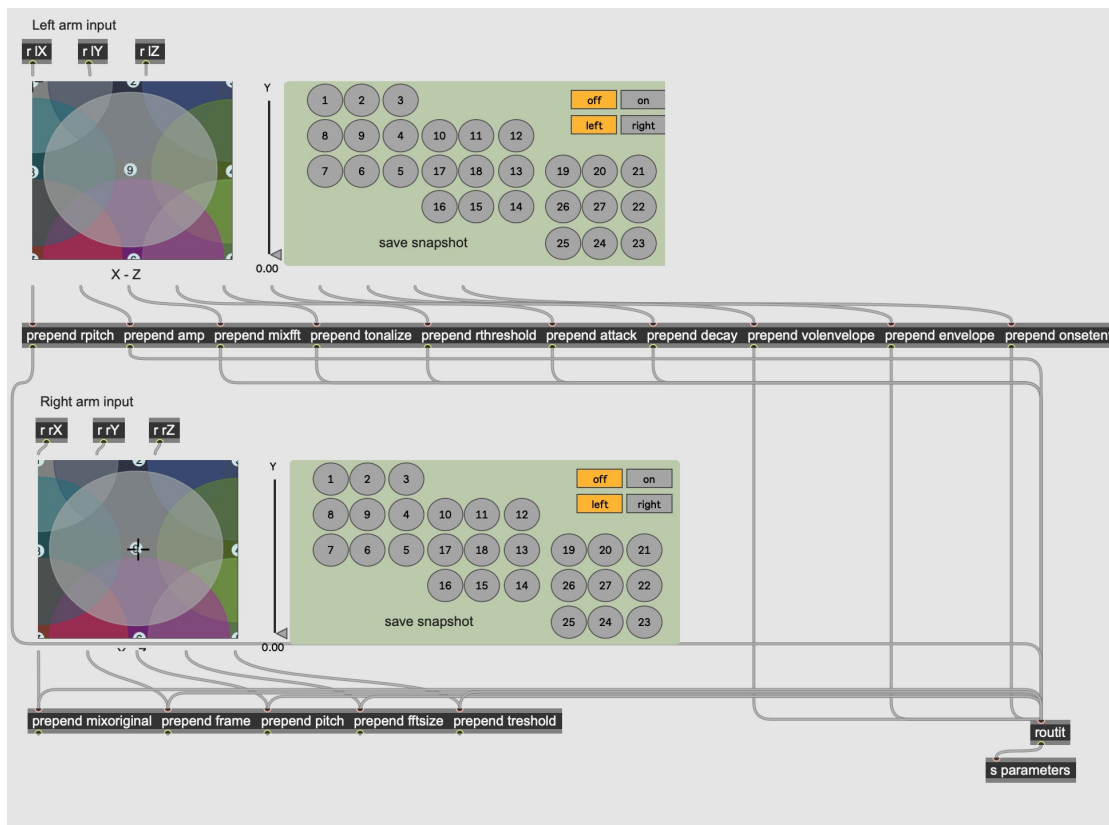


Illustration 13: the Max patch that receives signals from Playstation controller and Gametrak

<sup>56</sup> Jolliffe 2020





*Illustration 14: Patch that interpolates between the stored snapshots of the control parameters*

So this enabled me to have six simultaneous control signals to control the sonifications. However, I had fifteen parameters that I wanted to control. In order to make the controls more flexible, and to be able to calibrate the mappings fluently, I programmed a patch that can take 27 snapshots of the values of all the controlled parameters at the given time, and store them into two three-dimensional  $3 \times 3 \times 3$  table that correlates to the position of the controllers x, y, and z signals. When the user moves the strings of the *Gametrak* along the x-y-z axis, the patch does trilinear interpolation to “morph” between the snapshots that have been stored in corresponding “spaces” of the  $3 \times 3 \times 3$  table.

## 4.3.2 Routing

Signal flow of the patch is presented below.

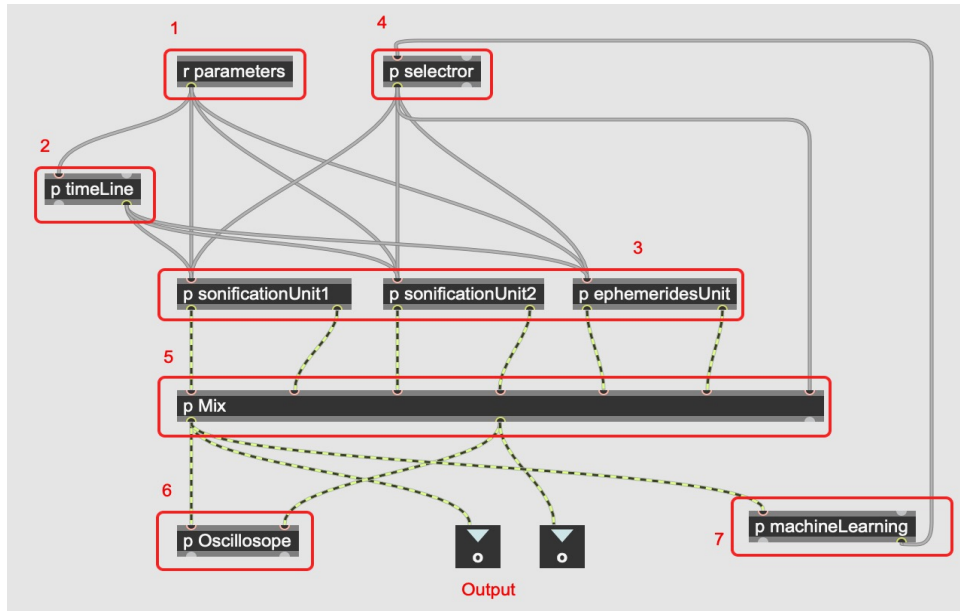
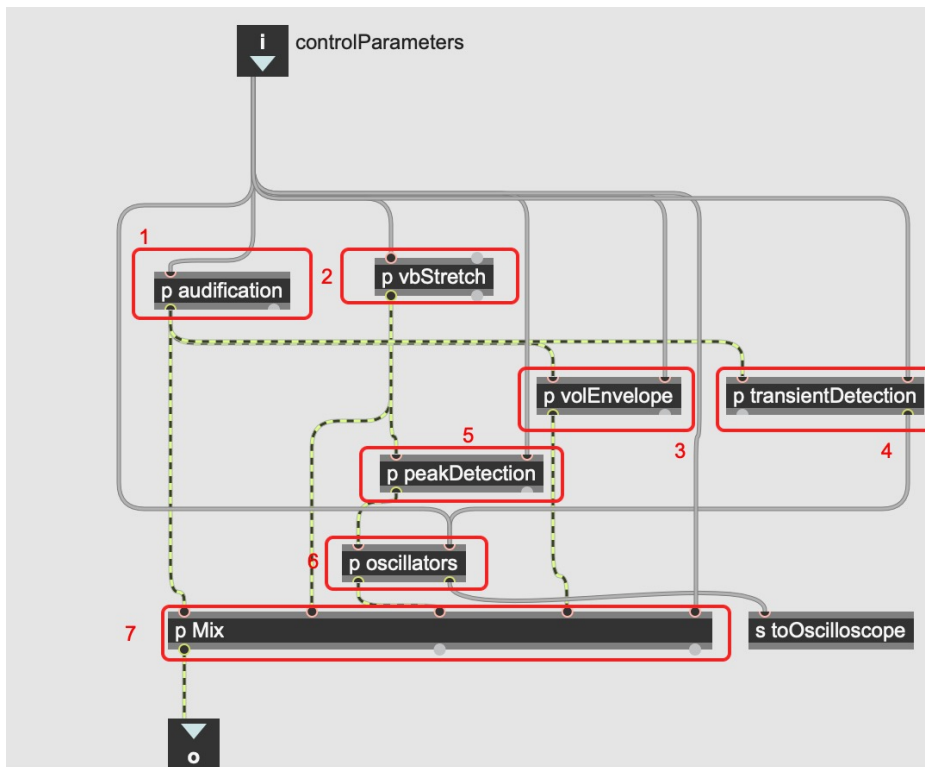


Illustration 15: The signal flow of the patch

All the user-controlled parameters arrive to the patch from “parameters” -receive object (1), from where the time-related parameters (playback rate, playhead position and start- and endpoints of the loop) are routed to the “timeline” -subpatch (2) that keeps the sonifications synchronized to a common timeline. The sonifications are processed in three separate units (3), in order to smoothly crossfade between the data sets. Two of them are identical, and enable to transition from one data set to another, while the third holds the Ephemis data set, which are processed slightly differently. Managing of these is controlled by “selector” -subpatch (4). “Mix” (5) -subpatch, also controlled by selector -subpatch crossfades between the the sonifications. The mixed output is then sent to “oscilloscope” -subpatch (6), for visual presentation of the sound. “machineLearning” -subpatch (7) analyzes the frequency spectrum of the sound, and compares it to the spectra of other data sets stored by other users and tells the selector -subpatch, which data set has the strongest similarity. The selector then selects the next data set accordingly.



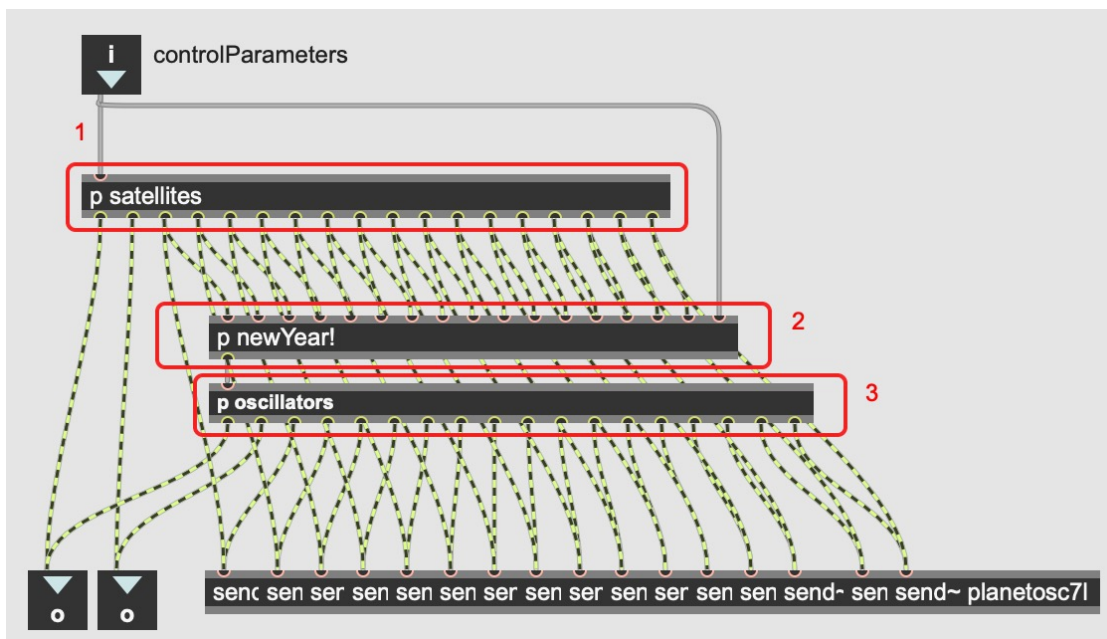
*Illustration 16: The "sonificationUnit" -sub-patch, presented here with mono -audiosignals for graphical simplicity, although the patch used in the Resonata is in stereo*

### The sonification unit -subpatch

Both of the "sonificationUnit" -subpatches are identical. All the user-controlled parameter values come in from "controlParameters" -input. The "audification" -sub-patch (1) holds the data set in a buffer, and plays it back according to user-controlled playback-speed. It's output signal is routed to "volEnvelope" -sub-patch (3) to sidechain the volume envelope, and to "transientDetection" -sub-patch (4) to detect sudden changes in the signal amplitude. "volEnvelope" sends the generated control signal to "Mix" -sub-patch (7) to attenuate the signal accordingly, and "transientDetection" sends generated trigger messages to "oscillators" -sub-patch (6). "vbStretch" -sub-patch (2) holds the same data set in it's buffer than the "audification" -sub-patch. It sends the synthesized output signal to the "Mix -sub-patch, and also to "peakDetection" -sub-patch (5), which analyzes the frequency peaks of the signal. Frequencies and the amplitude peaks are then sent as stream of fvalues to "oscillators" -sub-patch. Based on these lists, it generates the sound of the oscillators and sends the signal to the "Mix" -sub-patch. It also sends it

to oscilloscope (explained later). The “Mix” -sub-patch receives audio signals from “audification”, “vbStretch” and “oscillators” -sub-patches and control signal from the “volEnvelope” -sub-patch, mixes them according to user interaction, and sends the mixed signal to the output of the patch.

### Ephemerides unit -sub-patch



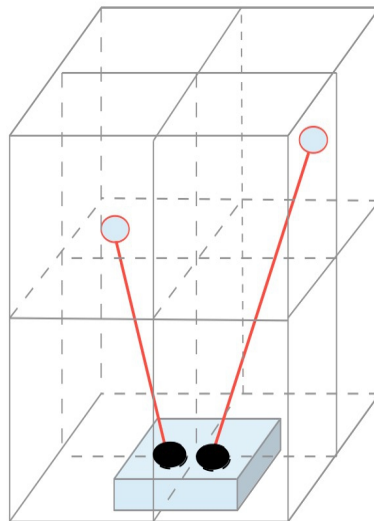
“ephemeridesUnit” -sub-patch works a bit differently. Instead of loading one audification at a time, the “satellites” -sub-patch (1) loads and processes eight audified files, either eight planets of the solar system, or maximum of eight moons of a planetary system at once. This sub-patch also holds the data of diameters of the orbits of the satellites, which enables to scale the orbits correctly in oscilloscope and in the sound output. Each audified signal from the satellites is sent separately to oscilloscope, in order to draw each individual orbit on screen. “newYear” -sub-patch (2), detects the zero crossings in the signal, and triggers an oscillator when a satellite completes a full orbit. It also detects whenever two or more satellites align, triggering corresponding oscillators. “oscillators” -sub-patch (3) holds eight oscillators (one for each satellite). They are triggered as mentioned above, with frequencies corresponding to the frequency of the orbit. The attack and release envelopes

of the oscillators are controlled by the user. The output of the oscillators, like the output of the audified signal, is sent to the main output of the sub-patch, and each oscillator is also sent separately to the oscilloscope.

### 4.3.3 Parameter mapping

The success of the parameter mapping defines, whether the user will regard the device to be an instrument.<sup>57</sup>

To map all the parameters (listed at the end of this chapter) that affect the sonification into two strings and a pedal proved to be tricky task. If I was to simply map the six control signals of the *Gametrak* directly into all the parameters, that would've caused conflicts between them, and wouldn't allowed much control over the combinations of the parameters regarding to gestures of the user. Because of this, and because I wanted to be able to easily update and calibrate the mappings, I programmed a patch that can store the values of all user-controlled parameters and store them in different

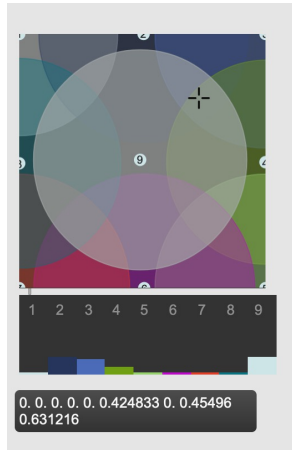


**Illustration 17: 3d parameter mapping -space.**  
*Each corner represents one of the 27 spaces where a snapshot is stored. the red lines represent the strings of the controller.*

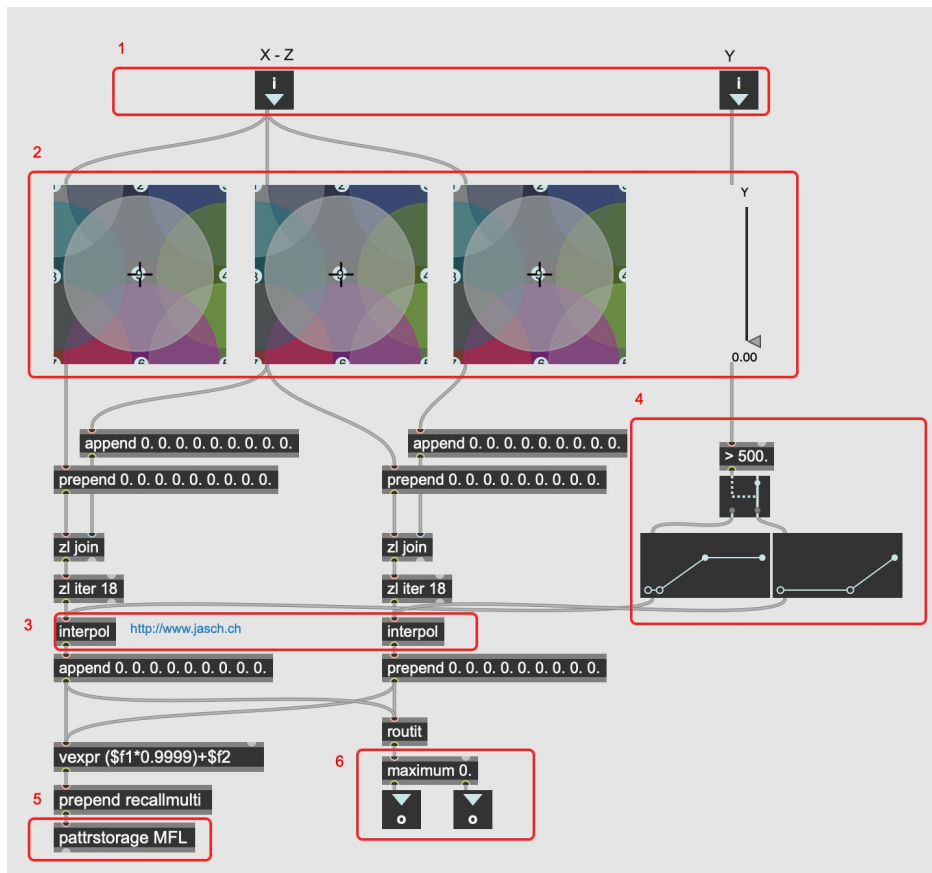
areas of controller occupied 3d-space.

<sup>57</sup>Hunt, Hermann 2004, 3

Max has an object called “nodes” which can interpolate between sets of stored parameter values on a 2d-plane.



**Illustration 18:** Nodes-object returned an interpolated weight of each node as a list



**Illustration 19:** patch used for trilinear interpolation

Input of the controller is received from the patch inputs (1). X and Z axis of the string from one input, and Y axis from another. X and Z control signal interpolate between the values stored three nodes -objects (2). Control signal Y is scaled (4) and sent to the “interpol” -object by jasch (3), that interpolates between the results of the three nodes -objects.

By using three nodes -objects, and interpolating between them it was possible to create a 3d mapping space, where the parameter values can be stored. These snapshots don't take user's speed of movement into account. It's also less suitable to parameters of which constant control isn't preferred.

Playback rate of the sonification is one of the parameters that where not suited for constant interpolation. Since all time-shifting in the patch is done in octaves, it changes the sound drastically, and I wanted it's control to be momentary event. Playback rate is changed in increments of +-1 octave. To shift the octave up, user has to bring his hands from his sides to front of him, in relatively rapid motion. To shift down, he/she has to reverse this motion. Movement of the strings also slightly increases the master volume of the patch, and activates a feedback loop between the oscillators and the frequency peak detection that triggers them (5 & 6, p.27). This kind of movement-based interaction can encourage users to engage with the sonifications more.

Besides the *Gametrak* -controller, the user can also control the patch with a control pedal. This is a sustain pedal used in keyboards, and send simple on/off messages. In the patch it is used to store the frequency spectrum of the sound for evaluation, and to select a different data set for sonification. In order to change the data set, user has to press the pedal for five seconds. During this time, the frequency peaks of the sound are stored in list form into machine-learning algorithm. When the pedal is released, the data set is changed.

Calibration of the mappings is something that could be adjusted forever, and it depends on the user's size and style of playing. The final calibrations were the sum of many compromises to make them suitable for many datasets and different users.

List of user-controlled parameters:

Name	Controller	Description
Buffer length	Gesture	Defines the start- and endpoints of the sonified section of the data set.
Playback rate	Gesture	Defines the playback rate of the sonification, changed in octaves.
Audification volume	Right	The relative amplitude of the direct audified signal.
"vb.stretch" -volume	Right	The relative amplitude of the FFT -synthesis from vb.stretch -object.
Oscillator volume	Left	The relative amplitude of the oscillator bank.
"vb.stretch" window size	Right	Buffer size of the FFT synthesis. Emphasizes either tonal or temporal characteristics of the data. The patch has four window sizes, 256, 1024, 8192, and 16384 samples. This parameter crossfades between these four.
frame-to-frame interpolation	Right	Smoothing of the transients in the signal.
"vb.stretch" pitch	Right	The pitch of the FFT synthesis, changed in octaves.
"vb.stretch" spectral threshold	Right	Spectral threshold applies noise reduction to the FFT -synthesis
Oscillator attack time	Left	The attack time for all the oscillators, from 0 to 5000 ms
Oscillator release time	Left	The release time of the oscillators, from 0 to 7500 ms
Oscillator pitch	Left	The pitch of the oscillators, changed in octaves
Oscillator pulse frequency	Left	The pulse frequency of the oscillators, changed in octaves
Frequency peaks	Left	The number of frequency peaks used to trigger oscillators
Frequency peak threshold	Left	The noise reduction for oscillator-triggering frequency peaks
Volume envelope	Left	The amount of volume envelope attenuation applied to the total output.
Volume envelope time	Left	The attack- and release time of the volume envelope
Onset envelope	Left	The volume of the oscillator triggered by the onset detection
Change the data set	Pedal	Change the sonified data set
Store sound	Pedal	Store the frequency spectrum of the sound into machine learning algorithm



## 4.4 Graphical interface

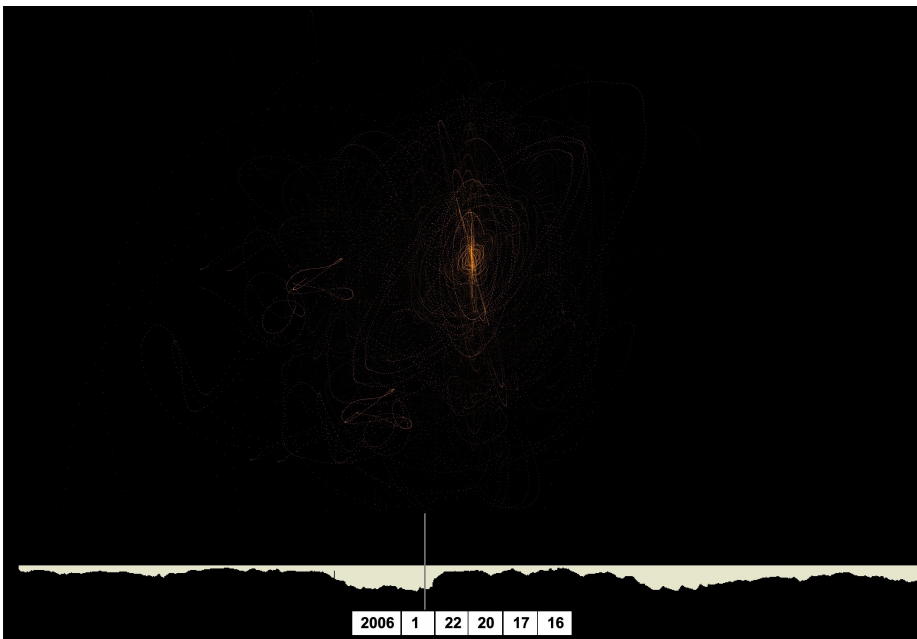
```

1 outlets = 1;
2 function timestamp(val) {
3   var d = new Date (val * 1000);
4   outlet(0, d.getUTCFullYear(), d.getUTCMonth() + 1, d.getUTCDate(), d.getUTCHours(), d.getUTCMinutes(), d.getUTCSeconds());
5 }

```

*Illustration 20: Code for converting the Unix time into Gregorian time*

I removed all numerical values from the interface except time. I did want to include time, since I felt that it would help the user to relate to the data and its scale. With Max it's easy to display the elapsed time of the wavefile in milliseconds or samples, but since some of the data sets were many years in length, using seconds would've been counterintuitive. I decided to display time as a Gregorian time-format. For this to work I needed a system that would take into account the leap years, the length of months, and so on. I did this by using Unix -time format that gives the time in seconds, starting from 1.1.1970. Unix time can be converted to Gregorian time with following javascript -code:



*Illustration 21: Graphical interface of the Resonata, showing the oscilloscope, data's waveform with the playhead (vertical line), and the calendar date at the playhead*

Graphical interface also shows the waveform of the data set. This zooms in and out according to playback speed of the data

### 4.4.1 Oscilloscope

The design of the oscilloscope is based on the Max patch “Sev” by Katsuhiko Chiba<sup>58</sup>. To be able to draw all the planets individually to oscilloscope, and for artistic effect, the oscilloscope is actually composed of eight different oscilloscopes superimposed on top on one another. One of them visualizes the total audio output, while others are dedicated to different oscillators in the oscillator-bank. This makes the visualization more more geometric and less noisy. Even though the result is slightly glossed-over version of the signal, I liked the visual effect it created.

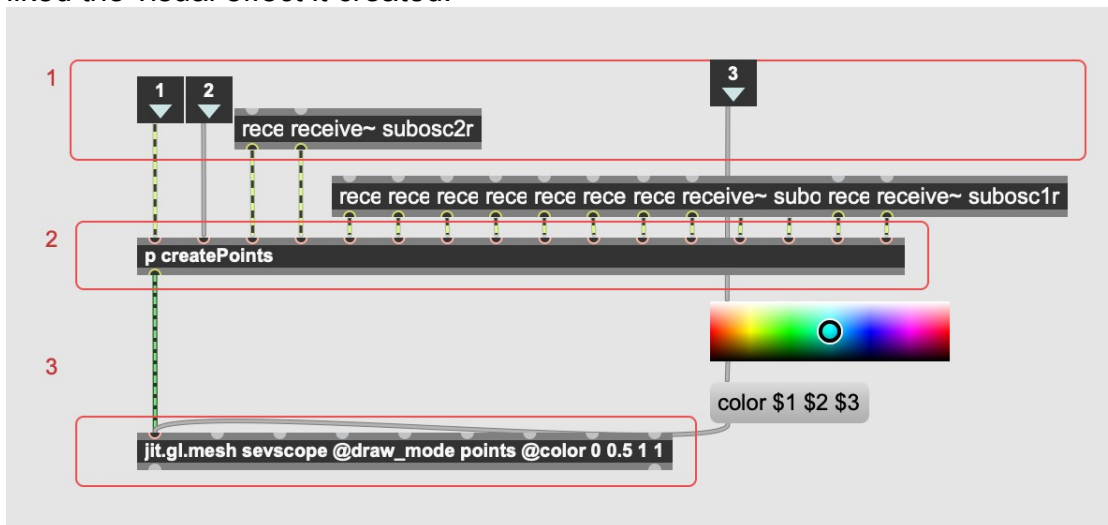


Illustration 22: Routing of the oscilloscope

Inputs 1 and 2 receive audio output from the user. Receive~ -objects receive audio from individual oscillators in the oscillator bank. Input 3 receives pre-programmed colour-code for the oscillator. The colour is based on the current sonification, user control, and noisiness of the signal(1). “Create points” -sub-patch converts the audiosignals into matrices(2). These matrices are sent into jit.gl.mesh -object, that renders the oscilloscope onto screen(3).

#### 4.5 Transitions

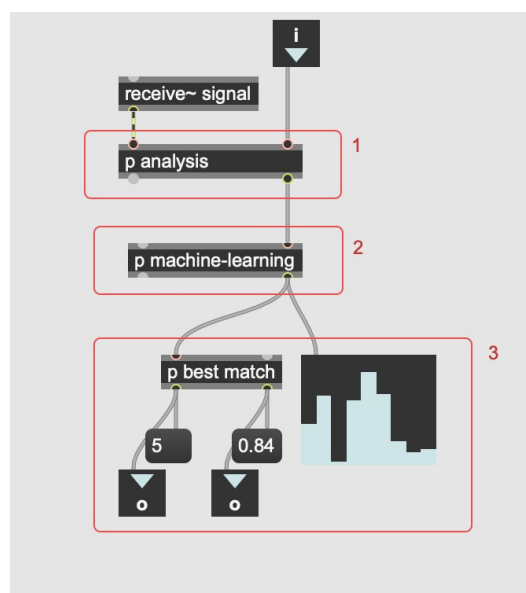
I wanted the user experience of shifting into a new data set to be gratifying. It's also the only part of the project, where I've used sound not derived from the data sets. In order to shift into a new data set, the user has to hold the foot

<sup>58</sup> Chiba, K., 2011

pedal down for two seconds. When the pedal is pressed, it triggers a sound that's not unlike a futuristic motor start-up. The purpose of this sound is to encourage the user to hold the pedal down long enough for the transition to happen. If successful, the sound will crossfade into a sonification of the new data set, playback rate smoothly transitions into the new speed, and the oscilloscope “zooms” in, if the new data set has higher sample rate, or zooms out, if the new sample rate is lower. The zoom effect does not reflect the relative sizes of the data sets or their sources, but it gives an immersive feeling of “diving in to the data”, when projected on a big screen.

#### 4.6 Machine learning

For the machine learning part I have used “ml.lib” machine-learning library by Jamie Bullock and Niccolò Granieri, and I've adapted a patch by Ali Momeni, which in turn is based on work by Makamoto Ono, Buntarou Shizuki and Jiro Tanaka<sup>59</sup>.



*Illustration 23: Signal chain of the machine-learning algorithm.*

Analysis -sub patch (1) takes the audio output of the user, and analyses its frequency content. Every 100 milliseconds it outputs it as a list of 40 peaks and minimums. These lists are compared to the database in machine-learning -sub patch (2). For each of the six sonifications, database has 100 lists that it's

<sup>59</sup> Ono, M, Shizuki, B & Tanaka, J 2013

comparing the input list to. It outputs the relative likenesses of all the candidates as a list, which goes to “best match” -sub-patch (3). That outputs the index of the sonification with strongest likeness with the sound, and it's value (with 0. having no correlation and 1. being identical). When this value reaches a threshold, it enables the user to switch into this sonification by pressing the pedal.

Pedal has to be pressed for two seconds before the patch transitions into a new data set. When the pedal is pressed, the input of the user starts to overwrite the existing machine learning database for maximum of 20 lists at a time. This way the database gradually evolves and rewrites it self over time. It will be interesting to see, how this will affect the general sound over longer periods of time.

## 5 The Installation

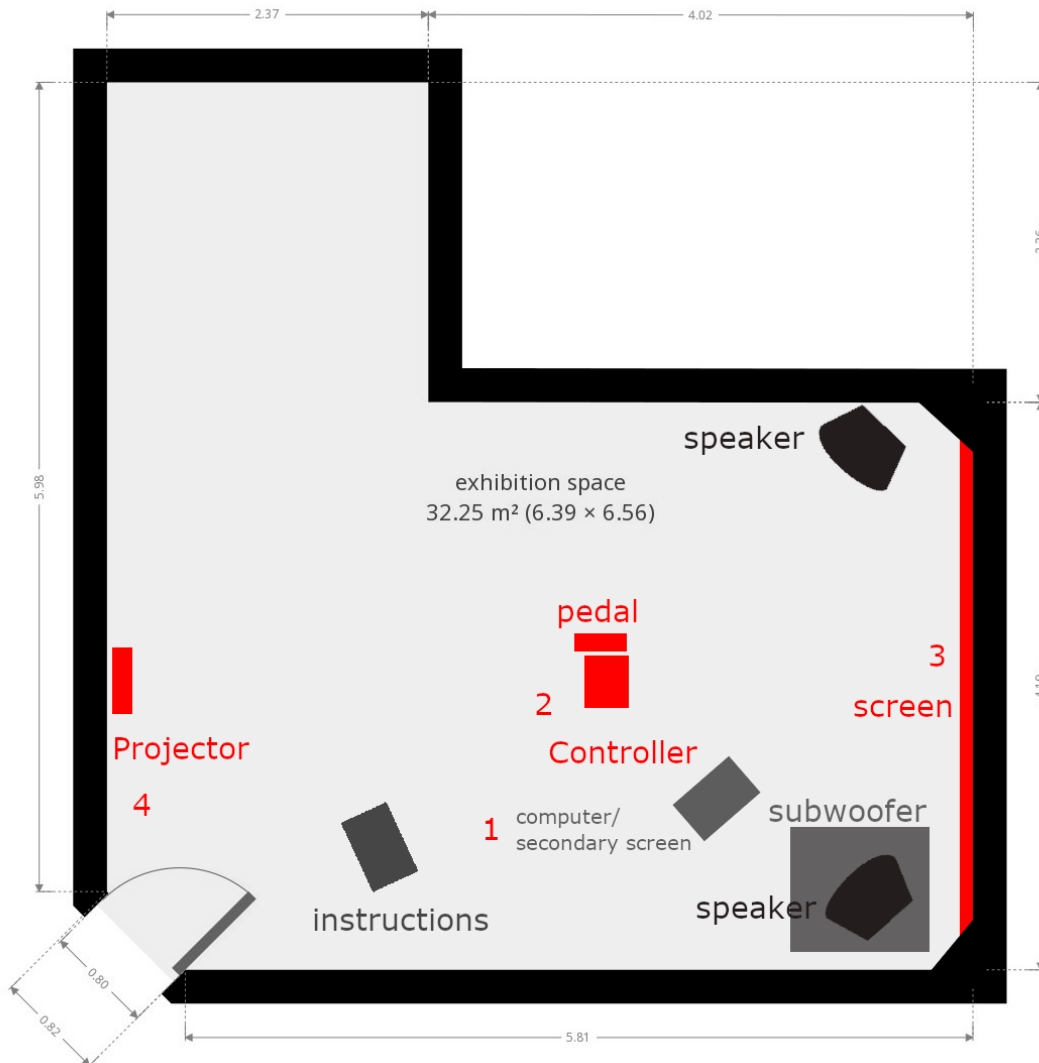
### 5.0.1 Introduction

I started building the installation alongside this research in 2018. Designing the installation and the research are intertwined in a process, where they both are feeding one another and can't be separated from one another. Since the work has mostly required just a laptop and an internet connection, I have been able to do much of the work at home. I have also been working in my recording studio in Vallila, Helsinki, since it has the required audio and video reproduction facilities.

*Resonata* was to be presented as an installation from 5<sup>th</sup> to 11<sup>th</sup> of April 2020 in Vallila, Helsinki. To present the installation I used my own studio space there, since it has a room of appropriate size for the installation, and it's an isolated, sound-proofed space. There's also the possibility to project the visualization on the wall and control the lighting of the room from the Max patch. It was to be held as an open exhibition for one week, with free entrance. I was to be present to tell about the project, help the users to control the piece and get feedback from them. This exhibition was cancelled due the Covid-19 pandemic. Instead I presented my work in online video. (<https://youtu.be/aXu0JvtnALo>)

### 5.0.2 Setup

The installation is set in a 6,4 x 4,2 m room. By the entrance there is a note stand, with short instructions for the user of the piece. On the floor, in the middle of the room is the *Gametrak* -controller and the foot pedal (a sustain pedal for electric piano was used) (2). The computer (1) that's running the software is located beside the controllers, but so that it won't block the view to the screen (3). The computer is on another note stand. The computer's screen provides additional information about the data sets. The projector (4) projects the graphical interface to the opposing wall. The computer's sound output is connected to a mixer that in turn is connected to the speakers and a



**Illustration 24: The Showroom**

subwoofer. The room is lit with Philips Hue -lightbulbs. These are remotely controlled RGB -lights that can be controlled with Max software.

For the video setup I put two cameras in the room, and also took a video screen capture of the oscilloscope screen with audio. The setup reduced the resolution of the screen capture so much that I made a second screen recording with just the laptop and combined these takes in to final video. Because of this, the timeline in the final video jumps occasionally.

For the video I also modified the patch so that it only looks for correlation when the pedal is pressed. This way I was able to present the features in more controlled manner in a short speechless video.

The video is accessible in the address: <https://youtu.be/aXu0JvtnALo>

### 5.0.3 *Reflections*

I intended to gather user-feedback from the installation in the form of open-ended survey and personal interviews for evaluating the success of the user experience. I also planned to use this feedback to further calibrate the mappings of the user control interface.

I couldn't acquire the intended user-feedback because of the ongoing Covid-19 -pandemic, so I use my own experience to evaluate the success of the onstallation.

Even though I have been playing with these sonifications for almost two years, the session where the video was taken was actually the first time I used the program in its final stage. The mapping of the controller was done the previous day, so it was new experience even for me.

To me the controls feel engaging, and fine-tuning the sound feels intuitive. I find myself coming up with fictional narratives for the sounds that I'm hearing. Rather than doing big movements, I like to move very slowly and stay still from time to time. I find the visualization of the patch immersive. At times it was hard to tell, whether I'm trying to control the sonification, or the visualization.

At times I found myself wanting to stop controlling the sonification and just listen to what the data is doing. I ended up programming a bypass switch for the controller just to be able to stop having to engage myself constantly with the production of the sound. When the user is constantly changing the sonification it becomes unclear what sounds come from the data sets and what is caused by user control. Also, the sheer physicality of the control method means that it can be hard to concentrate on the sound for long periods of time if you have to keep your body in a straining posture.

The machine-learning algorithm that compares the datasets still needs work. As it is, it still feels somewhat random. The possibility to listen to several

datasets simultaneously should be added, so that the comparison of data sets could be done aurally, by listening to several simultaneous datasets.

When evaluating musical instrument, probably the most important question is: What does it sound like? I do find some of the sounds *Resonata* creates musically interesting. Each data set also has it's own distinctive mood to it, although the oscillator bank starts to sound a bit monotonous after a while. I think the oscillators could be made more complex in the future, for example by adding intermodulation with other datasets. For example, FM (frequency modulation) synthesis could be applied.



## 6 Conclusion

The results of this project demonstrate some possibilities of applying artistic expression to sonification through interactivity.

Artistic use of sonification can give an emotive weight to the numbers it is representing. If explained through an analogy: When we see in a magazine a picture of a beautiful, brightly-coloured nebula, or an “artist's impression” of a distant planet, we understand that these are visualizations, not actual representations of those things. But still those visualizations can inspire readers, invite them to study the subject and to understand it. I think that artistic use of sonification can have the same kind of value.

Interaction is vital in artistic performance, and can also be a powerful tool in learning. Using it with sonification can help us in studying the data we are sonifying.<sup>60</sup> Interactivity can help to engage users and immerse them with the subject. Whether this interaction is seen as an artistic expression depends on the user's intention. We use musical instruments to express emotions and experiences. If used for that, I feel that *Resonata* can be called a successful musical instrument for artistic expression.

However, I found out that because of the physicality of the controller, the constant interaction in *Resonata* can also become straining for the user (as discussed in chapter 5). The solution was to include the possibility for the user to disengage themselves from the controls, when needed. This way the user can have the best of both worlds: intuitive exploration through interactivity, and the option to step back and just to listen to what the data is telling.

How accessible *Resonata* is to the audience, remains to be seen. I intent to study this with the audience once the pandemic quarantines are lifted.

Artistic experience can be used to communicate subject matters that science and statistics have difficulties to grasp, such as complex emotions and human

---

<sup>60</sup> Hunt, A., Hermann, T. 2004, 2

experience. The interaction between the study of sonification and musical expression can utilize the power of both worlds for greater human experience.

### 6.0.1 What next?

The next step for me will be to rebuild the Max patch, share it with the community of Max users, and start to study possibilities of other control methods. Support for multiple channels should be added in order to be able to study intermodulation and resonances between different data sets.

I also want to study possibilities for converting the user interface into VR environment. There is an application programming interface for Max to communicate with the software development platform Unity. With Unity it would be possible to create a powerful, immersive Virtual Reality experience based on the sonification concepts of my work.

I want to develop the gesturality of the patch further. At the moment the mapping is based on my experimentation and preferences, but I would like to get more user feedback to fine-tune the connection between the sound and the gesture.

Through this project I have also come across with more general new live performance applications in Max that I would like to build and share.

I would like to try *Resonata* as a part of musical ensemble.

This project has greatly expanded my musical toolkit and programming skills. If I would start the project now, the program would end up very different, much simpler and more versatile.

## 7 Bibliography

- Barrass, S., Vickers, P. 2011. "Sonification Design and Aesthetics" *The Sonification Handbook*, Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 147. Logos Verlag, Berlin, Germany
- Beilharz, K. 2005. "Gesture-Controlled Interaction with Aesthetic Information Sonification", *Key Centre of Design Computing & Cognition*, University of Sydney Wilkinson Building G04
- Brown, M., Kehoe, A., Kirakowski, J., Pitt, I., (2010). Beyond the Gamepad: HCI and Game Controller Design and Evaluation. 10.1007/978-1-84882-963-3\_12.
- Bruss, F., T., Reuschendorf, L., 2010 "On the Perception of Time", *Gerontology* June 2010
- Böhm, V. [homepage] <https://vboehm.net/downloads/> accessed 01.04.2020
- Canli, T., Gabrieli, J. D. E., Zhao, Z., Brewer, J., Cahill, L. 2000. "Event-Related Activation in the Human Amygdala Associates with Later Memory for Individual Emotional Experience" *The Journal of Neuroscience* : The Official Journal of the Society for Neuroscience · Vol. 20 RC99
- Chiba, K., 2011, "Sev" (Max patch) <http://audiooo.com/max-patches> accessed: 12.04.2020
- Csikszentmihalyi, M. 1990 "FLOW: The Psychology of Optimal Experience", *Harper and Row*.
- Dombois, F., Eckel, G. 2011. "Audification" *The Sonification Handbook*, Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 302 & 305. Logos Verlag, Berlin, Germany
- Effenberg, A.O., Melzer, J., Weber, A., & Zinke, A. 2005. "Motion Lab Sonify: A Framework for the Sonification of Human Motion Data". *The Ninth International Conference on Information Visualization (IV '05)*, London, UK, pp. 17–23
- Falcão, L., Neves, A., Ramalho, G., Campos, F., Oliveira, B., 2010 *Game as Art: A Matter of Design, Videojogos*, ISBN: 978-989-20-2190-4
- Finnish meteorological institute 2019, [WWW -database] <https://www.ilmatieteenlaitos.fi/avoin-data> , accessed 20.08.2019
- Focillon, H. 1989, "The Life of Forms in Art", *Zone books* (Original work published 1934)

Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. 2003. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*. 101(23):e215-e220.

Giorgini, JD and JPL Solar System Dynamics Group, NASA/JPL Horizons On-Line Ephemeris System, < <http://ssd.jpl.nasa.gov/?horizons> >, accessed 01.04.2020.

Grond, F., Berger, J. 2011 “Parameter Mapping Sonification” *The Sonification Handbook*, Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 363. Logos Verlag, Berlin, Germany

Guinness World Records 2013 “Confirmed: Grand Theft Auto 5 breaks 6 sales world records” [www-document] <https://www.guinnessworldrecords.com/news/2013/10/confirmed-grand-theft-auto-breaks-six-sales-world-records-51900/> accessed 21.04.2020

Hermann, T., Hunt, A., Neuhoff, J. G. 2011 “Introduction” *The Sonification Handbook*, Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 1. Logos Verlag, Berlin, Germany

Hunt, A., Hermann, T. 2004 “THE IMPORTANCE OF INTERACTION IN SONIFICATION”, *Proceedings of ICAD 04-Tenth Meeting of the International Conference on Auditory Display*, Sydney, Australia, July 6-9

Hunt, A., Hermann, T. 2011 “Interactive Sonification” *The Sonification Handbook*, Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 278. Logos Verlag, Berlin, Germany

IronEqual. 2020. “THE ONLY ADVICE YOU WILL NEED TO MAKE A GREAT GAME UI/UX”. [WWW-document] <https://medium.com/ironequal/the-only-advice-you-will-need-to-make-a-great-game-ui-ux-74a0db8de642> accessed 12.04.2020.

Jameson, M. 1966. "The Optophone: Its Beginning and Development" *Bulletin of Prosthetics Research, spring 1966*

Jenseniuss, A., Godøy, R. I., Wanderley, M., Leman, M. 2010 “Musical gestures: concepts and methods in research” *Musical gestures: Sound, movement, and meaning*, R. I. Godøy & M. Leman (Eds.), (pp. 12–35). New York: Routledge.

Johnson, AEW., Pollard, T., J., Shen, L., Lehman, L., H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L., A., Mark, R., G. 2016. “MIMIC-III, a freely accessible critical care database”. *Scientific Data*. DOI: 10.1038/sdata.2016.35. Available from: <http://www.nature.com/articles/sdata201635>

Jolliffe, D., Arduino2Max (arduino code) <https://www.danieljolliffe.ca/writing+code/writing+code.htm> accessed 12.04.2020

- Kemp, B., Zwinderman, AH., Tuk, B., Kamphuisen, HAC., Oberyé, J.J.L. 2000, “Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG”. *IEEE-BME* 47(9):1185-1194 2000.
- Kepler, J. 1619. “Harmonies of the world”. *Bhoomi Digital Apps* (2016) [Apple iBook] accessed 16.04.2020
- Kilb, D., Z. Peng, D. Simpson, A. Michael and M. Fisher 2012, “Listen, watch, learn: SeisSound video products”, *Seismol. Res. Lett.*, 83(2), 281-286, doi: 10.1785/gssrl.83.2.281.
- Koelsch, S., Skouras, S., Lohmann, G., 2018 “The auditory cortex hosts network nodes influential for emotion processing: An fMRI study on music-evoked fear and joy. *PLoS ONE* 13(1): e0190057”. [WWW-document] <https://doi.org/10.1371/journal.pone.0190057> accessed 31.04.2020
- Lockton, D. 2018. “Apophenia As Method—Or, Everything Is Either A Metaphor Or An Analogue Computer”. Conference paper: *Disruptive Improvisation: Making Use of Non-Deterministic Art Practices in HCI*. Workshop at CHI '18, April 21–26 2018. At: Montreal
- Ono, M, Shizuki, B & Tanaka, J 2013, “Touch & Activate: Adding interactivity to existing objects using active acoustic sensing.” in *UIST 2013 - Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. pp. 31-40, 26th Annual ACM Symposium on User Interface Software and Technology, UIST 2013, St. Andrews, 13/10/8. <https://doi.org/10.1145/2501988.2501989> accessed 01.04.2020
- Pigeon, S. 2020 “The non-linearities of the Human Ear”, [WWW-document] [https://www.audiocheck.net/soundtests\\_nonlinear.php](https://www.audiocheck.net/soundtests_nonlinear.php) accessed 31.04.2020
- Plato, 380BC. “The Republic”. *Public domain* (2008) [Apple iBook]
- Power, R. & Steinberg, S. & Bjornsdottir, G. & Rietveld, C. & Abdellaoui, A. & Nivard, M. & Johannesson, M. & Galesloot, T. & Hottenga, J. & Willemsen, G. & Cesarini, D. & Benjamin, D. & Magnusson, P. & Ullén, F. & Tiemeier, H. & Hofman, A. & Rooij, F. & Walters, G. & Sigurdsson, E. & Stefansson, K. 2015 “*Polygenic risk scores for schizophrenia and bipolar disorder predict creativity*” *Nature Neuroscience*. 18. 10.1038/nn.4040.
- Rhodes, B., 2020. *jplephem* (python -programming library) <https://pypi.org/project/jplephem/> accessed 12.04.2020
- Sean Carrol's Mindscape*. podcast, episode 41 accessed 20.04.2020 (spotify) <https://open.spotify.com/episode/6KWT1OSY7esKpxiEMMIMLI?si=z8ZBNsOMTJCUj4O7d-OlgA>
- Scaletti, C. 2017 “Why Sonification Is a Joke” [WWW -page] <https://carlascaletti.com/sounds/data-sonification/> accessed 31.04.2020

Schooler, J., Hunt, T. “The easy part of the Hard Problem: A resonance theory of consciousness”  
[WWW-document]  
[https://labs.psych.ucsb.edu/schooler/jonathan/sites/labs.psych.ucsb.edu.schooler.jonathan/files/pubs/the\\_easy\\_part\\_of\\_the\\_hard\\_problem-resonating\\_structures\\_life\\_and\\_consciousness\\_v18.pdf](https://labs.psych.ucsb.edu/schooler/jonathan/sites/labs.psych.ucsb.edu.schooler.jonathan/files/pubs/the_easy_part_of_the_hard_problem-resonating_structures_life_and_consciousness_v18.pdf) accessed 31.04.2020

Serra, J.O., X. Smith III, 1990. “Spectral modelling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition”  
Computer Music Journal vol.14(4)pp.12-24):

Smirnov, A. 2011. “Graphical Sound” [www-document]  
[https://asmir.info/graphical\\_sound.htm](https://asmir.info/graphical_sound.htm) accessed 15.04.2020

Tan, J, 2018. “Wait...Maybe Hippies Weren’t Right”, blog, *Scientific American*  
<https://blogs.scientificamerican.com/observations/wait-maybe-the-hippies-werent-right/> accessed 20.04.2020

Tate.org 2020. “Sound Art” [web page] <https://www.tate.org.uk/art/art-terms/s/sound-art> accessed 26.04.2020

Varshney, L., R., Sun, J., Z., 2013 “Why Do We Perceive Time Logarithmically”,  
[WWW-document] <https://rss.onlinelibrary.wiley.com/doi/full/10.1111/j.1740-9713.2013.00636.x> accessed 31.04.2020

Walker, B.N., Nees, M.A. 2011“Theory of Sonification” *The Sonification Handbook*,  
Thomas Hermann, Andy Hunt, John G Neuhoff (Edt.), 16. Logos Verlag, Berlin,  
Germany

Worrall, D., 2009 “SONIFICATION AND INFORMATION CONCEPTS,  
INSTRUMENTS AND TECHNIQUES”, *worrall.avatar.com* author's homepage,  
<http://worrall.avatar.com.au/papers/phd/> accessed 19.04.2020

Worrall, D. 2018. “SONIFICATION: A PREHISTORY”. *Michigan Technological University* June 10-15: 2