# The Reactional Technology: An Artistic Practice-Informed Approach to Developing a Generative Music Engine

*Jesper Nordin*

## Abstract

This article describes the process of creating a software for composing and improvising music, now called Reactional Music, and its evolution from a personal tool into a musical engine for interactive experiences, primarily in the field of video games. By tracking the development of the technology over time, with examples of its use in different projects, the article seeks to highlight the bidirectional way in which my artistic practice and the software development have continuously informed each other. The Reactional technology is a generative music engine that enables intuitive control over complex musical structures by reducing the dimensionality of input data. The original use case was to manage many multiple musical parameters with just a few input streams, allowing for fluid generation and improvisation. This article provides perspectives on the creative aspects of composing music and developing software in parallel through an iterative design process. The research method is a retrospective analysis of the system across distinct developmental states, an approach informed by autoethnographic analysis of the continuous feedback loop between artistic practice and technological development. The main contribution of this article is to observe the complex, intertwined process of the technology's creation, thereby providing crucial insight into the relationship between software development and the artistic projects that utilised the technology.

**Keywords:** Composition, Contemporary Music, Game audio, Digital musical instrument, Human-computer interaction, Interfaces for musical expression, Generative music, Procedural audio

## 1 Personal background

As a professional composer of contemporary music, my early musical experiences continue to shape both my artistic output and my compositional methodology. Having begun my musical journey relatively late – by forming a punk rock band at fifteen – I did not receive formal training in music theory or notation until my twenties, driven by a desire to compose. Without a classical foundation before engaging with contemporary music, I have consistently turned to technology as a means of realising my artistic intentions. Over


ISSN: 2002-021X. DOI: 10.58698/stm-sjm.v107.33742 

time, this has led to the development of several interactive audio tools designed to support and enhance diverse compositional processes, often created for specific tasks to explore musical ideas.

The most generic tool I developed is a technology now called Reactional Music. Different iterations of the same core technology can be found in the previous and current iOS apps Gestrument, Gestrument LE, and Gestrument Pro, as well as in the full product suite for game music called Reactional Music. In this article, I will refer to Reactional as a technology rather than software, as the term more accurately captures its development – including various iterations involving distinct hardware configurations and input methods. The article will not delve into the current state of the Reactional Music technology but instead seeks to document and analyse the process of developing the Reactional technology, and the intricate interaction between software development and compositional projects that has led to expanded artistic possibilities from its first iteration until its current form.

From the very first experiments, the technology was never conceived in isolation but as an evolving tool shaped by my compositional ideas and explorations. Each version of the system was created to facilitate specific artistic goals, and subsequently, each iteration influenced my artistic work. This reciprocity has resulted in a body of artistic projects that both reflected and shaped the technological development. By systematically saving and documenting every iteration of the technology – alongside sketches, recordings, and reflections from each project – I established an archival foundation for analysing this continuous feedback loop. This archive consists of around fifty iterations of the original Max/MSP patch (Cycling74, 2024), spanning from the first tests in 2007 to the last active version in 2013. In addition to these full patches, there are multiple versions of several sub-patches iterated during the same timeframe. From 2012 onwards, when the iOS apps were released, the archive consists of the user manuals and development notes from the different Gestrument apps. My compositional practice always involved continuously saving compositions, resulting in multiple versions of all scores, audio sketches, and numerous documents with notes that supported the analysis of the autoethnographic material.

While searching for strategies in my compositional processes that could help me combine my interest in making intuitive decisions while still letting me transcend my own limitations, I tried different ways of composing with the help of technology. During my studies, I used a combination of audio-based techniques – where I recorded musicians and then used those recordings as building blocks and as a base for improvisation – and different computer-aided techniques where the basic material was MIDI and/or lists of pitch and rhythm material that I then shuffled, merged, or randomised in different ways. Without going into the details of these compositional strategies and techniques, I still want to mention them, since the driving force behind the use of those strategies was similar to what later led me to develop the Reactional technology. On a fundamental level, it was based on finding ways of allowing me to become an improviser and a musician without having mastered a traditional instrument. During my studies, I described myself as more of a musical director (as in a movie director or theatre director) than a classical composer, and by that I meant that I preferred to be presented with material – from musicians I worked with, from algorithmic processes, or from my own improvisations – and then take that material as a base for my compositions. Another very important aspect has been that I wanted to surprise myself. One early example of this was when I mapped samples of instruments in a random order on a keyboard and then improvised. This meant that the keys I pressed did not correspond to the pitches I heard, and I could thereby bypass any conscious choice of chords or melodies.

I gradually found that I spent more and more time on my audio sketches and that they almost became pieces in themselves, before I started to transcribe the sonic sketches I had produced. Therefore, I decided to try to find, or build, a tool that would let me keep the aspects of the compositional process that I liked – improvising, listening, and using material from sources that I already had a strong connection to – while still letting me delve deeper into the possibilities of computer-aided composition. I also wanted to find an approach that could lead me to work with the actual score at an earlier stage of my process. All these aims led me to the development of the Reactional technology, and they have been important driving forces behind its continued development.

This article starts by describing the original Max patch in some detail, as well as covering the later iterations of the technology leading up to the fundamental new concept that is the current patented technology as it exists today in the app Gestrument Pro and the Reactional Engine. The three platforms that the Reactional Engine is ported to now – Unity, Unreal, and WebAssembly – are all used to produce video games, interactive applications, and network-based digital experiences.

At the core of the Reactional technology lies a reduction of dimensionality – controlling many musical parameters with just a few data inputs, thereby reducing complexity for the user while also adding possibilities for the composer/audio designer. An example of this is found in a video where I perform a Klezmer piece together with clarinet soloist Martin Fröst (see Nordin and Fröst, 2020: https://youtu.be/aEK-wscIcJg), where every note is performed in real-time (as opposed to triggering predefined melodic melodies/fragments), but where the rules of the software are set up so that every note is within the confines of the song that is played.

# 2 Introduction

## 2.1 Current research

This article contributes to the growing field of Artistic Research (practice-led research), which asserts that artistic practice is a valid mode of inquiry capable of generating unique, tacit knowledge. Following the precedents set by Swedish dissertations that use creation as a research engine – such as Frisk (2008), who employs improvisation to rethink human-computer interaction; Nilsson (2011), who bridges the gap between engineering, musicology, and artistic practice; and Petersson (2025), who studies "musicking" via the specific act of patching modular systems – this article aims to use artistic means, methods, and results to generate both theoretical insights and practical knowledge regarding interactive music systems, and more specifically, to provide examples of how the technological development and my artistic process have influenced each other.

Digital Musical Instruments (DMIs) can often be made to reduce dimensionality since they are software-based and therefore very flexible. I would argue that this aspect of DMIs is still not fully explored and could yield much more interesting musical tools and results in the future. As shown more than twenty years ago by Jordà (2004), DMIs are often situated in a paradigm based on acoustic instruments where the "score" and the "orchestra" are two separate things. In contrast, he encourages an approach where the form and the sound are controlled by the same input, something that he believes would open a path to new types of musical expression. Jordà's 2004 observation was insightful, and the field has since seen significant advancements allowing for tighter integration, mainly regarding the combination of gesture and sound. Recent work in embodied musical interaction, such as Jensenius's (2013) exploration of sounding bodies and Caramiaux's (2014) focus on the ecology of musical interaction, highlights ongoing research into how movement and technology can deeply intertwine in musical performance. More recent studies continue this trajectory, investigating musical "intra-actions" between musicians and DMIs (Tahiroğlu et al., 2020). While these developments are pushing towards the unified control Jordà envisioned, the full realisation of this concept in mainstream musical expression is still ongoing due to legacy practices, technical challenges, and the diverse artistic goals of developers. Therefore, while progress is evident in niche areas and research, the complete fusion of form and sound through a single input remains more of an evolving frontier than a widespread norm in musical practice.

This article also builds on the research by Magnusson (2010), where he describes DMIs through the three perspectives of affordances, constraints, and mappings, claiming that mapping the constraints of a DMI should be considered a compositional process. That aspect is important in this article through its in-depth case studies of how a personal compositional tool developed into a generic musical system driven by real-time input data.

Some aspects of the origins and development of the Reactional technology have previously been described by Bacot and Féron (2016), Lähdeoja (2019), Nordin (2010, 2020, 2024), and Asplund (2022), but the current article is the first fully chronological description, which includes examples from some of the musical compositions created with this technology to highlight and exemplify the technology at its different stages. To put this in context, it should be stated that the Reactional technology has been used in many compositions since its first iteration in 2007, both by myself and by other composers. Apart from being used by contemporary composers, it has also been used extensively by those who have bought and/or downloaded the different versions of the iOS app: Gestrument (2012), Gestrument LE (2013), and Gestrument Pro (2018). These apps have been downloaded by more than 150,000 users, and many users have posted their tracks on platforms like SoundCloud, YouTube, and other online channels.

The first iteration of the Reactional technology was as a Max patch controlled by a Wacom tablet (a drawing tablet with a pencil, which allows access to data from pressure, tilt, and buttons) that was used as a personal composition tool, mainly as an improvisational device to generate material for scored compositions. After becoming a commercially available app in 2012, the technology was then patented in 2018 and has since been developed into a Software Development Kit (SDK) coded in C and ported to the video game engines Unity (Unity Technologies) and Unreal Engine (Epic Games), as well as to WebAssembly.

The first commercial application of the technology was an iOS app called Gestrument. The name was meant to be descriptive since it was a gesture-instrument, and the implications of gestures throughout the development process have been many and important. On the other hand, gestures have never been the starting point – there has never been an analysis during the process of what kind of gestures would produce which kind of music. In part, that has been due to a lack of interest in gestures in themselves on my part, but also because the wide-ranging possibilities in the technology of setting up different musical scenarios and rules have meant that one specific gesture would produce very different results depending on the present settings and mappings. Therefore, this article will not go into any length in describing the relationship between the Reactional technology and the different input methods and gestures used to drive it.

The process of developing this technology has been as much a compositional endeavour as a technical one, where constraints and affordances shaped both the tool itself and the artistic expressions it enabled. The analysis and descriptions will, for instance, draw on the research on interactive music systems by Rowe (1993) as well as the attempts at classification of DMIs made by Malloch, Birnbaum, Sinyor, and Wanderley (2006). Even though this article is more focused on the interrelations between technical development and my artistic practice during the development of this specific technology, it also touches on future possibilities for using this and similar technologies in video games and, to a lesser extent, in Smart Musical Instruments (SMIs) as defined by Turchet, McPherson, and Fischione (2016).

### *2.2 Aims and methods*

This article documents the development of the Reactional technology through archival analysis of software development from 2007 onwards. Furthermore, it provides an analysis of the development of my own compositional techniques across a series of artistic projects in the same time frame and how they informed, and were informed by, the technological development. By comparing the different iterations during the development, the article seeks to show how limitations in earlier versions led to specific innovations and how artistic needs prompted new functionalities. Each artistic project discussed in this article functions as a case study, offering concrete examples of how the technology has been applied and adapted. These case studies not only illustrate the practical use cases but also highlight the conceptual shifts throughout the development process.

While the systems use gestures as control data to drive the music in real time, the focus in this study is not specifically on gesture–music mappings. Rather, it is the compositional and design processes that are in focus, contributing to the broader field of digital musical instrument research. The purpose of this study is to analyse the reciprocal development between the Reactional Music System and my compositional practice from

2007 to 2018, thereby unpacking the creative potential inherent in their interlocking processes. The research questions are:

- How did my compositional processes during 2007–2018 inform the development of Reactional Music?
- How did the technological innovation during the development of Reactional Music contribute to my own compositional practice?

The analysis in this article is based on the archival documentation of the software development since 2007, and by comparing five distinct technical iterations, it seeks to show how limitations in earlier versions led to specific innovations and how artistic needs prompted new functionalities. Each artistic project discussed here functions as a case study, offering concrete examples of this reciprocal process.

The five artistic projects detailed below were selected as illustrative case studies. They provide the clearest, documented examples of the system's reciprocal development, demonstrating instances where (a) the technology informed novel musical outcomes, or (b) compositional requirements that prompted the creation of new features within Reactional Music.

The compositions that I will use in this article are the following:

1. *Surface scintillantes* (2008). Commissioned by Ensemble XXI in Dijon. The first piece composed with the Reactional technology.
2. *Pendants* (2009). Commissioned by Ensemble L'Itinéraire in Paris. The first example where I composed music I never would have written without the Reactional technology.
3. *Vicinities* (2011). Commissioned by the Swedish Radio Symphony Orchestra. In this piece, requests from the soloist for more rhythmic music led me to develop new features in the Reactional technology.
4. *Ärr* (2014). Commissioned by the Royal Stockholm Philharmonic. This piece illustrates how the Reactional technology worked well for sketching, but where I needed to abandon it for the final composition due to constraints. The same piece was also used in the Dream project that is covered in this article.
5. *Emerging from Currents and Waves* (2018). Commissioned by the Swedish Radio Symphony Orchestra and Orchestre Philharmonique de Radio France. This piece utilised the soloist Martin Fröst and the conductor Esa-Pekka Salonen as soloists performing on the Reactional technology live during the performance, which added new possibilities and complexities.

The original idea behind this technology was to provide a tool for my own compositional needs, and since I have composed practically every work since 2007, in full or in parts, using the Reactional technology, it must be considered a goal accomplished. This article seeks to describe the reciprocal relationship between technological development and the artistic ideas in the compositions in which it has been used.

# 3 The original composition tool

*3.1 Reactional Music pre-history*

Computer-aided composition (CAC) is a vast area of research that has been developed and documented over several decades. Even though I was not aware of it at the time of my studies, it is very clear in retrospect that I was researching software and strategies that would later form the framework of my new compositional method. The tools listed below were used and explored mainly during my studies at the Royal College of Music (1995–2001), at IRCAM (2002–2003), and at Stanford University (2004).

1. ***Composer Tools (CT)* by Pär Lindgren.** CT was programmed in HyperCard and gave the user access to a wide collection of contemporary compositional techniques. It had a depth and breadth that I personally did not delve into fully, but I used it to explore different aspects of musical ideas, or as I expressed it at the time, to let the computer improvise for me and present me with material that I could then use for composing. CT was developed during the 1980s and 1990s, and there is practically no online literature on the software, even though I still have a copy of the user manual in my library. The software is mentioned in a Studio report from the Royal College of Music by Brunson and others (2002).
2. ***OpenMusic (OM)* from IRCAM.** OM was developed as a continuation of the composition software PatchWork and is widely documented (e.g., Assayag et al., 1999; IRCAM, 2024). Personally, I never started using OM in my actual compositions. I tried finding ways and strategies that would work, but they never fit what I wanted to do, and I kept working with CT instead. That being said, the experience of learning the software at IRCAM was informative for my future development of Reactional.
3. **Real Time Composition Library (RTC-lib) by Karlheinz Essl.** The RTC library has been under development since 1992 and is documented in Essl (2024). This was my first test of using real-time tools to generate material, and I used some of these tools in a few pieces. I never felt that I could use it to compose the music I wanted to, and the result became more like etudes and technical tests where only a very limited number of those were used in actual composition.
4. ***Boids* algorithm by Craig Reynolds.** The most direct starting point for the Reactional technology must be said to be the idea that I wanted to use the Boids algorithm to generate musical material. This algorithm is used for simulating the movements of moving flocks or animal herds and was developed by Craig Reynolds (1987). I had identified that my music often moved like clouds or flocks, and one of the audio tools I used the most was an audio object I developed in Max/MSP that I called "Heterophony". This object is basically a randomised delay line that takes an audio input and plays it back a predefined number of times with more or less randomised length of the audio snippet and delay of the playback starting point. When I wanted to translate that into musical data that I could use in a score, I thought that the Boids algorithm was a good

candidate. The Boids algorithm exists as a Max/MSP object, and I tried several different ways of controlling it and interpreting it without getting any material that sounded like what I was looking for. This was the final trigger for trying to develop something myself, where I could improvise over a given scale and rhythmic material to produce a material that I wanted to use in my compositions.

### *3.2 The original Max patch*

After using different tools and strategies I started to have a clear idea of what kind of technology I wanted to use when composing. I knew that many composers used a Wacom tablet for manipulating sound, and that felt like a very intuitive and direct way of creating a personal instrument. I had also defined what kind of music I wanted to generate, in the form of heterophonic movements. When I had envisioned what I needed in a compositional software tool, I tried to find out if anyone had made such a tool already. I found software that had some of the aspects of what I was considering, but I could not find anything that did what I wanted, so I tried developing it myself. In 2007, the first version of the technology was developed as a Max patch controlled by a Wacom tablet. It mapped pitch and rhythm to the x- and y-axes of the Wacom tablet and thereby created a collection of playable instruments with some level of parameter control. The full Max patch typically consisted of eight to twelve such instruments being played at once. In two videos from the music centre GRAME in Lyon, percussion professor Jean Geoffroy and I briefly demonstrate the Max patch in its state at that time (see Nordin and Geoffroy, 2011: https://vimeo.com/24004279, https://vimeo.com/24002171). This turned out to be a pivotal moment, since it was when testing my personal tool with a new input method controlled by another musician that I started to realise the potential of the technology I had developed.

The design of this Max patch reflects the rather intuitive and unstructured development process, where new features were added over time and then inserted into the design without any predefined plan or organisation. The basic principle stayed the same over the five years of development (2007–2012), but several new features were added during this period.

Figures 1 and 2 on the next page illustrate similarities and differences between what I then called an “instrument” (the parameter setup for a specific part) in two versions of the patch from 2007 and 2012. Many fundamental things are the same, some are different, and a few parameters and possibilities have been added. The full Max patch was made up of several instances of these instruments, as well as global parameters like scales or presets.

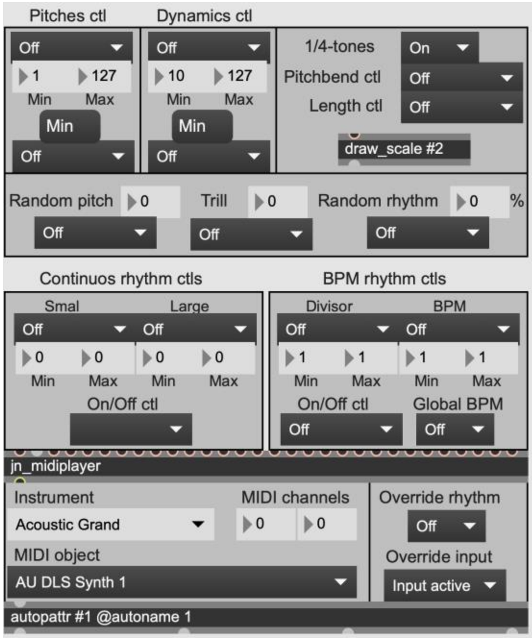


Figure 1: First version of a basic instrument from 2007

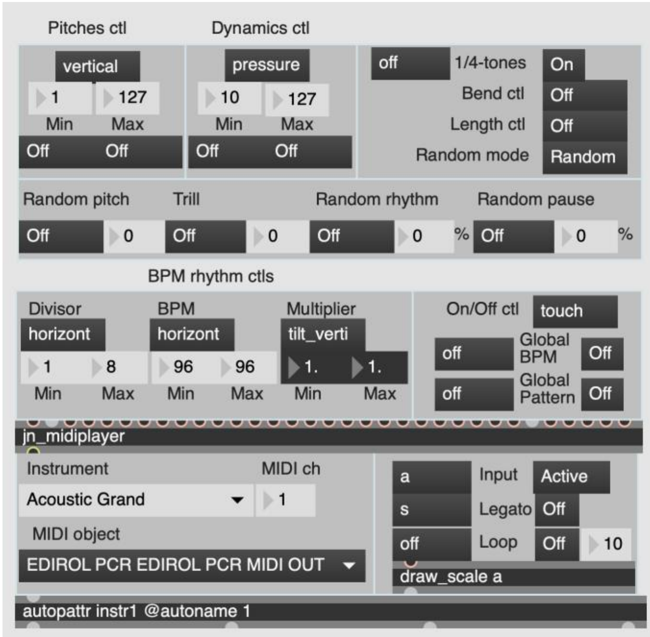


Figure 2: A later version of a basic instrument from 2012

Here is a detailed description of the instrument shown in Figure 2:

1. Starting at the top left, there is a section with the generic mapping of pitches (regardless of which scale is used; the scale is defined in a global part of the patch) called Pitches ctl. It defines which input to use (vertical meaning the vertical axis of the Wacom tablet – all similar black boxes consist of a list of input choices, ranging from the tilt and pressure of the Wacom pen to Faders and Knobs on different external MIDI controllers), what the minimum and maximum pitch is for that specific instrument, and if those minimum and maximum values should be changeable, and if so by which parameter (Off meaning that in this setting they are not changeable by any input control).
2. To the right of the Pitches ctl, there is a section with the same settings as those described above, but here dedicated to dynamics instead of pitches – Dynamics ctl.
3. To the right of the Dynamics ctl, there are some miscellaneous controls, including dynamic control over quarter tones, how to control pitch bend, control of note lengths, and which type of randomisation to use.
4. Below that first row of the parameters sections, there is a full row of randomisation parameters and a trill option. They all work in a similar way, where a control input is defined to control a number box. These number boxes are all changeable in real time with the control input but have different ranges: Random pitch (0–24), Trill (0–12), and the Random Rhythm and Random Pause (0–100%). This means that the random pitch can be from 0 (no randomisation) to any random value between 0 and 24 scale steps over the given value; the trill can be from 0 (no trill) to trills of 1–12 scale steps; the randomisation of rhythm can be from 0% (no randomisation) to 100% (the onset can then happen at any place between the given value and the next

given value); the randomisation of pause (which in practical effect means that a given note is ignored) is between 0% (no pause) and 100% (only pause).

5. Below the parameters section, there is the rhythmic control with three different parameters: Divisor, BPM (Beats Per Minute), and Multiplier. The way this patch worked with rhythm was to have a defined BPM (individual for each instrument) that was then divided into smaller units and/or multiplied by a factor. In Figure 2, the BPM never changes (it goes from 96 to 96) and is never multiplied (going from 1 to 1), but it is divided into units from 1 to 8. These units are then mapped to the horizontal axis of the Wacom tablet so that on the far left corresponds to quarter notes (or whatever unit is used as base for the BPM), and then it will be divided by integers between 2 and 7 to finally land on a division of 8 at the far right of the tablet (which would mean 32nd notes if we started with a quarter note).
6. To the right of the basic rhythmic control, there are some other parameters also affecting rhythm in different ways: On/Off ctl, Global BPM, and Global Pattern. The On/off ctl defines how to turn this specific instrument on or off; most commonly, that was done with the touch of the Wacom pen to the tablet. The Global BPM overrides the local BPM control with a global tempo, which can be static or dynamic – it can, for instance, be triggered by an attack detector on an audio signal. The Global BPM is dynamic and can be turned on and off by a control input while playing. Finally, the Global pattern is a way of using rhythmic patterns instead of just subdivisions; there can be up to eight rhythmic patterns predefined, and each instrument can use one of them. This is not dynamic and can therefore not be changed while playing.
7. Below the dark, narrow section called jn_midiplayer (which is the actual patcher with the musical logic inside), there are controls defining which GM (General MIDI) instrument, MIDI channel, and MIDI port to use. None of these aspects was dynamic in nature.
8. The final section to the bottom right has four parameters that are not really connected to each other. At the top, there is a control for whether the instrument should be active or not; this was used to be able to add and subtract instruments while playing. The second control turned out to be a fundamental part of the technology and is named Legato (possibly not a fully musically correct name). The "legato" mode makes the instrument ignore any onset that has the same pitch as the onset before, and thereby, it is possible to play long notes even though the horizontal axis actually is a rhythmical grid with fixed note values. The third control was for recording loops, but in a rather different way. The loop playback starts as soon as the Wacom pencil stops moving (while still touching the Wacom tablet), and the loop that is played back consists of the last events that were played before stopping, both regarding rhythm and pitch. The exact number of events in the loop was specified per instrument. Finally, in this section, there was a Draw scale box that could be opened and where each instrument could use its own scale. This was developed from using the idea of drawing the scales freehand but was later expanded to accommodate other ways of defining scales as well.

Apart from these parameters in the individual instruments, there are several global controls, where the section that handles scales is the most fully developed. At first, the scales used were octave-based and used standard temperate tuning, but early in the process, support was added for scales that use quarter tones, as well as for scales that have a different scale base than twelve. The scales were defined by setting the highest and lowest pitch, the active pitches on two keyboards (one with quarter tones, one without), and the scale base, or modulo, of the scale – as seen in Figure 4 below. The scale shown in Figure 4 is C-minor with both the third and the seventh degrees being a quarter tone high, starting on MIDI note 24 and ending on 108.

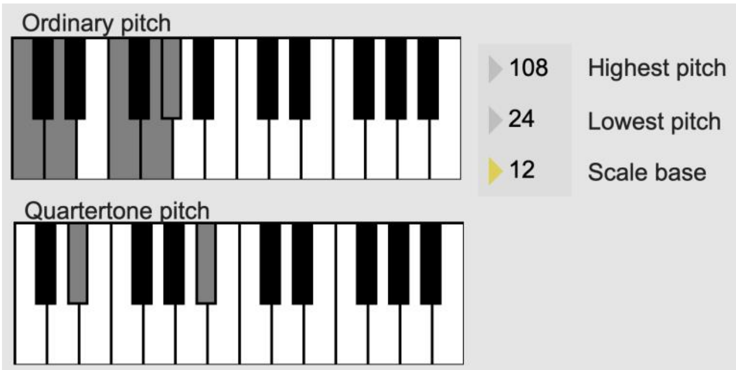


**Figure 4: Scale definition**

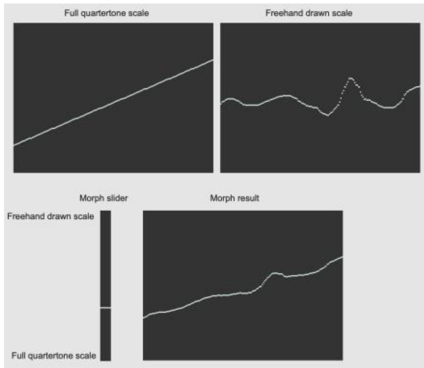


**Figure 5: Morphing between different types of scales**

At an early stage, the possibility to use freehand drawn scales was added, as well as the possibility to morph between a more traditional scale that goes from low to high pitches, and a freehand drawn scale that can go in any direction. This is shown in Figure 5, where a morph slider can gradually move between two extreme positions that are defined in advance, in this case, a full quarter tone scale to the left and a freehand drawn scale to the right.

The next logical step, after working with freehand drawn scales and morphings, was to start importing MIDI files and use them as scales (that is to say, not using the pitches in the MIDI file and sorting them to get the actual scale it is in, but rather using a MIDI file instead of a scale, so that the first pitch in the melody was mapped to the bottom of the Wacom tablet and the final pitch of the melody was mapped to the top) and morph between them and/or other types of scales. This was implemented in 2010 and became a defining feature of both the continued compositional process with the Max patch and an important part of the iOS apps that would follow. Especially the app ScaleGen, which did not use the core Reactional functionality but rather expanded on the ideas of pitch material transformations and compositional techniques, became an intuitive multitouch tool for all genres. The ScaleGen app is not covered in this article since it is not really part of the core functionality, but some of the aspects in that app were later re-incorporated into later versions of the technology to some extent.

Other important additions were, for instance, the added possibility to use other types of external support, mainly support for other types of external support, mainly in the form of the Kinect motion sensor. These global parameters were added in the full overview of the patch (see Figure 6 on the next page). This also showcases how the patch looked when opened – primarily an overview of the instruments and their settings. To the top left, all other settings were added, some on the top level, and some one level down in different sub-patches.

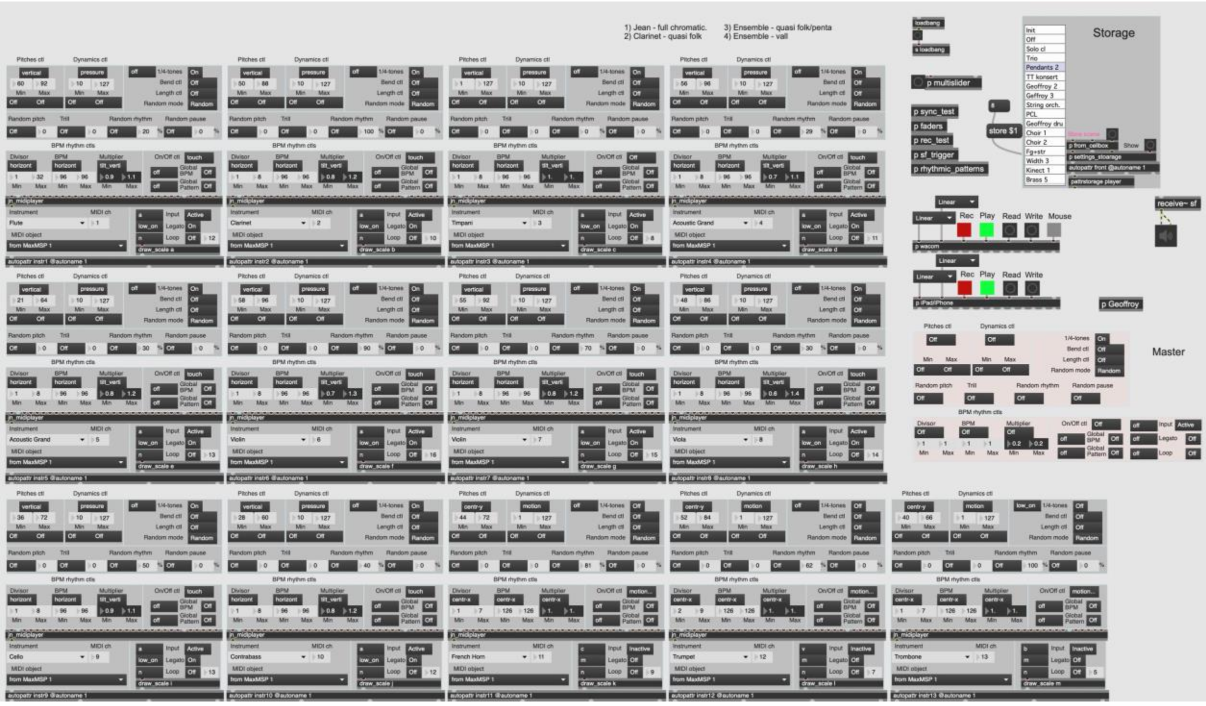


**Figure 6: Patch overview**

*3.3 The first compositions*

This first Max patch was only used by myself, since there was no documentation for the patch and no distribution. I used it to improvise within a predefined material and record everything I did, to later be able to browse through the material and choose the parts I liked the best. A lot of the material for the first pieces I composed with this tool was actually recorded while I was developing the first version of the software. When the Max patch was done in its first iteration, I realised that I had collected so much material during the development process that I did not need to use the actual patch for a long time. The first piece I composed with this tool is the chamber ensemble piece *Surfaces scintillants* (2007). Figure 7 on the next page shows an example of a MIDI file, unedited apart from the time signatures, from the compositional process of *Surfaces scintillants*, and Figure 8 shows the same section in the final score. With this process, I wanted to enable intuitive improvisation in real time on the full ensemble while having global control over parameters like harmonic and rhythmic content. This early version of the *Max* patch was therefore made to easily allow for instruments to be turned on and off while improvising, as well as having ways of changing the global harmonic content of the music. In this way, the role of the composer could be described as a conductor-improviser, who had control over the entire ensemble while still controlling the actual notes and phrases that were produced. The creation of the final score from the unedited MIDI files involved a more traditional compositional process where the music was re-transcribed into a more idiomatic form for the instruments as well as phrased more organically. This transcription process can be described as part informed proof-reading – to make sure the music can be played on the instruments – and part composing. I was in no way trying to be truthful to the generated material, so anything that was deemed not to add to the character of the piece was edited or deleted. The role of the technology at this stage could be described as a material generator, and as such, there was no demand for a coherent graphical representation or a musically consistent structure in the output; that was left to the transcription process afterwards.

Figure 7: MIDI file sketch from *Surfaces scintillants*

Figure 8: The same excerpt from the finalised score © Edition Peters

*Pendants* (2009) is another chamber ensemble piece in which I used the same approach as outlined above. In an article about the composition process of the piece, I wrote a first public walk-through of this technology and the concepts behind it (Nordin, 2010). In *Pendants*, I had my first experience of composing music with this technology that resulted in music that I would never have thought that I would like. While developing the technology and during the first pieces I wrote with it, I mainly used it to generate a type of material that I already had an idea of how it would sound. But with the new mappings I could make with the tilt and pressure of the pen, I added a rather unpredictable control over playing techniques in the software sampler. Therefore, I ended up with a long section of staccato scratch notes, that did not sound at all like anything I had ever composed before, and I realised that the element of surprise in the compositional process could be taken much further than what I had done before. Since then, I have often designed musical situations where there are some parameters that go beyond the range that would feel natural for me by default.

One interesting recollection regarding these first compositions was that I felt that my music had changed at its core and sounded completely different – I remember being afraid that people might not recognise my personal style anymore, since I found the result so profoundly different from my earlier compositions. Therefore, I was very surprised when I asked several friends and colleagues about this and got the reply that they thought it was very much in line with my earlier work, and that they would not have been able to say that I had changed my compositional process. This discrepancy between my own experience of the music and that of others was a source of major comfort to me and made me think that the working process one uses as a composer actually is less important than one could imagine – there is always a personal filtering process that determines the actual aesthetic choices, no matter which composition process is used.

The technology was also used in orchestral compositions, for instance, in *Vicinities* (2011), which is shown in Figures 9 and 10, and *Frames in Transit* (2012). In the later piece, the Max patch was also used on stage as a live instrument for the first time.


**Figure 9: An unedited MIDI file sketch from *Vicinities***

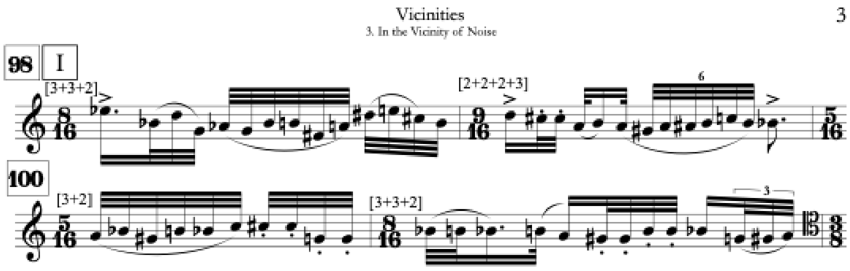


**Figure 10: The same Excerpt from the finalised score © Edition Peters**

*Vicinities* is a piece I have used often as an example when explaining my compositional strategy, since the working process was very clear in a way that is unusual for me. In *Vicinities,* I had some very clear musical ideas for the basic material when it came to tonality and the role of the soloist. With those ideas in mind, I then spent two full weeks only improvising while recording everything. Since that gave me a lot of material, I was careful to take notes on what I found was the most musically interesting parts, so I could navigate the compositional process that was to follow. After these weeks of improvising, I started to make choices, organising the different parts in a coherent form, using the improvisations as building blocks. It was first after I had a form that I believed in that I started to use the raw MIDI material I had and turn that into a score through my transcription process. The composing of this piece took around nine months, and almost the entire score was based on the improvisations I did during those first two weeks.

The examples above from the piece *Vicinities* are chosen because they highlight how the technological development was driven by compositional ideas. The first iteration of the Max patch that was used in *Surfaces scintillants* was not focused on rhythmic, pulse-based music, and therefore, the unedited MIDI data was often lacking in rhythmical precision. When adding more rhythmic possibilities, like rhythmic patterns and more dynamic control over the rhythmic content, the unedited MIDI data became more coherent and usable. The material was still treated in a similar way though, using the unedited MIDI data as a raw material to be freely changed and refined to fit both the musical expression and the idiomatic qualities of the instrument.

This excerpt from the bassoon cadenza of the third movement of *Vicinities* shows both the more rhythmically precise result in the unedited MIDI data as well as illustrate the difference between the unedited MIDI data and the final composition, which is similar to the difference between the two examples from *Surfaces scintillants.*

# 4 The first iOS app - Gestrument

### *4.1 The Gestrument app*

In 2011, I started the development of the first iOS app, Gestrument, together with composer and software developer Jonathan Liljedahl. The first idea was to take all aspects of the original *Max* patch and make an iOS app, but early in the process, we decided to limit the scope of the app to make for a coherent and usable application. One of the first aspects that was limited was the work with scales. Some aspects of the original *Max* patch were still present, like for instance the scale morphing slider and the possibility for quarter tones, but much of the functionality was not migrated to the app. This was remedied in 2014 when a second app called ScaleGen was released by the same team of developers.

This article will not go into the details regarding the structure of the Gestrument app, since the promotion videos and the user manual do this in a thorough and coherent way. What is important for this article is rather to highlight the role of this app as a developmental step. It is clearly a step forward from the ideas in the original Max patch, but still not as fully formulated as in the next iteration, Gestrument Pro, and the patents granted at that stage of the development.

The input and expertise of Jonatan Liljedahl was a fundamental and instrumental part of the Gestrument app. Liljedahl had recently started to make iOS music apps under the name Kymatica and had released a few successful apps already, and released many more, even more successful apps in the upcoming years. His combination of design skills, deep musical knowledge and experience of software development in his own apps as well as joint projects like Supercollider (see Hall, 2016), made him uniquely positioned to develop the app.

Many design choices which limited the functionality were made to make the application understandable and relatively intuitive for the users. The app ended up having eight instruments and four sliders that could be used for one task each: Rhythm randomness, Pitch fluctuation, Pulse density, and Scale morph (see Figure 11). In the edit area (see Figure 12), the rhythmic material that could be used was limited to a predefined selection of note values, and the scales were based on an octave (modulo was always twelve half-tone steps) with the possibility of quarter tones. The settings for range and randomisation were static, so once it was set up for an instrument, there was no dynamic control.


Figure 11: The playing area of Gestrument from 2012


Figure 12: The edit area of Gestrument from 2012

Already during the first year after its release, many parameters and possibilities were added – for instance, a Rhythm pattern possibility, transposition buttons in the playing area, and functionality to make it work with other apps and hardware (MIDI Clock and similar).

After the release of the original Gestrument app, more people started using the technology, and many different examples of how it could be used in versatile musical genres and surroundings emerged in online videos like the ones from Gestrument (2012), Gilligan (2014) or Olivier (2014). Since the first Gestrument app, there has been a constant stream of suggestions for new or revised features from the users. This way of receiving information and suggestions from users is, of course, the preferred method for developing commercial software, and the development team collected all these suggestions and went through them when updates were planned. Many of these changes regarded practical things like connectivity to other apps or different design choices. But the main driving force of the fundamental musical functionality was driven by a combination of ongoing discussion between me and Liljedahl and the actual artistic projects that the app was used in.

### *4.2 Some examples of compositions made with the Gestrument app*

The orchestral piece *Ärr* (2014) is an example of a piece that continues the rhythmical work that was done in the original Max patch. The title of the piece means “scar” in Swedish, and it is inspired by the music of the metal band Meshuggah. More specifically, it is inspired by a song called “Bleed” from the album *Obzen* from 2009, which has a high level of rhythmic complexity combined with a basic riff that melodically is just a small glissando up and down. I decided to use the same basic concept and mapped string instruments to the eight instrument slots in the app, each with a different set of note values and a different range of the glissando. Sometimes the instruments used the same rhythmic pattern, sometimes different variations. With this basic setting, I started improvising on the material and became fascinated by the possibilities of this rather basic setup (see Gestrument, 2014: https://youtu.be/nng6rMkIYO0).

The composition of the piece then started with these improvisations that were used more or less in their original form at the end of the piece. In other sections of the piece, I used the improvisations I did more as placeholders, to indicate the length and direction of a section, but then I composed the music in a much more structured and systematic way. The similarities between the free improvisations and the more systematic versions of these sections are noteworthy, since the global sounding results are similar, but on the detailed level, they are different. This is seen in the two examples in Figure 13, which shows a raw MIDI file from an improvisation, and in Figure 14, which is taken from the score where I had found a way of systematically approaching the music. Basically, what happened was that the material I had improvised was too detailed and complex for me to use through my normal previous transcription process, and it became more difficult to re-transcribe the improvisations than to compose the section in a more systematic way from the ground up, which was what I ended up doing. Therefore, this example, which in many ways is one of the clearest examples of how I imagined the music through the technological possibilities and limitations of the app, is at the same time a clear example of the limitations of the technology as I used it at the time.


Figure 13: An unedited MIDI file sketch from ***Ärr***


**Figure 14: A similar section from the finalised score © Edition Peters**

Another piece composed with this version of the technology is *Sculpting the Air* (2015), for the French ensemble TM+, commissioned by IRCAM. The working process of creating the piece was tracked and documented by the researchers Baptiste Bacot and François-Xavier Féron, who identify the composition as related to current artistic explorations of the role of gesture in musical composition and performance, noting that

> Nordin's concept in Sculpting the Air is part of a musical trend that concentrates on the conductor's gestures, although the number of pieces – apart from soundpainting improvisations – is quite small. The role of the conductor in Nordin's work appears nonetheless unique in several ways: not only he has to conduct the ensemble in the traditional manner; he also has to control the electronics gesturally, as well as play suspended bells. Another difference from de Mey's and Schubert's work is that in Sculpting the Air, the conductor is not directly equipped himself, but is surrounded with the sensors. Moreover, Nordin did not invent a whole new gestural language for the conductor to learn and use; rather, he expanded the conductor's traditional gestures into the realm of electronic interaction. As he explains:
>
> The movements of the conductor have a lot of exformation that, of course, differ greatly, depending on who is watching. But the relatively small amount of measurable information in the movements lends itself perfectly to a first experiment with delving into the exformation of musical aspects. This is done for instance by taking the ordinary gestures from the conductor and placing them in a new and expanded context where the result will be different (WD 1) (Bacot and Féron, 2016, p. 5)

*Sculpting the Air* was one of the last pieces I composed with the original Gestrument app, and by then, the limitations of this version of the technology had started to become obvious. The fact that many things were hard-coded into the app and could not be changed – everything from the performative sliders in the playing area to the rhythmic subdivisions – created frustrating limitations when specific ideas were pursued. Other limitations, like the number of instruments – and even more so, the number of playable cursors – made it less than ideal when I was starting to plan to use the technology in larger settings, such as orchestral works.

The lessons learned from my artistic projects, coupled with input from the growing user base, gradually made it clear to us that we needed to make a fundamental overhaul of the entire codebase. The Gestrument app was a hard-coded version of a concept that we believed could be remade in a much more modular and generic way. Therefore, updates were stopped for this app and all development work was instead focused on the next step. At that point, we started visualising how a musical generative engine could be conceived to function as a versatile musical instrument as well as a procedural music system to be driven by other types of data.

## 5 Making it generic – Gestrument Pro and Patents

### *5.1 The App Gestrument Pro*

Much in the same way as with the Gestrument app, this article will not discuss all details of Gestrument Pro, but will instead contextualise and describe the general outline of the concept and ideas behind it. For anyone who wants a deeper understanding, there is an overview video and a user manual, as well as many other online tutorials from the company Gestrument AB (renamed Reactional Music Group AB in 2022) and from users and reviewers.

The development was still done fully by Liljedahl, while the concepts and design choices were made collaboratively by us. We had many lengthy discussions and many development boards on ideas, concepts and use cases.

Going back to the beginning of the work on Gestrument Pro, one of the fundamental design choices was to make this technology as modular and generic as possible at its core. Figure 16 (see below under 5.2) shows the basic structure of the new approach, where a Pitch Generator and a Rhythm Generator are combined into an Event Producer. These generators could then be exchanged and, even though the first version of Gestrument Pro only shipped with two pitch generators and two rhythm generators (or Pulse Generators as they are called in the app), the plan has always been to expand on that and make more generators of different kinds.

Without covering Gestrument Pro in too much detail, there are still some basic observations to make that show the many changes that were made. Gestrument Pro was a completely new app, made without reusing anything from the codebase of the first Gestrument app. This was an early decision since the first version of the app had many hard-coded design choices that were now redesigned to become generic and modular.

Some of the most obvious differences are that Gestrument Pro is a multi-touch instrument with up to eight touch points (cursors) that can control up to sixteen different instruments in any configuration, as can be seen in Figure 15. There are eight sliders that can be mapped to most parameters in the instrument settings, and there are eight scale slots to more easily change the tonality of the produced pitches. The sliders can also be used as "via" sliders, where, for instance, the amount of randomisation or the range of an instrument is mapped to a slider, making the sliders powerful tools for both real-time improvisation and for compositional work, where they can be used to control formal aspects by changing the global aspects of the music.

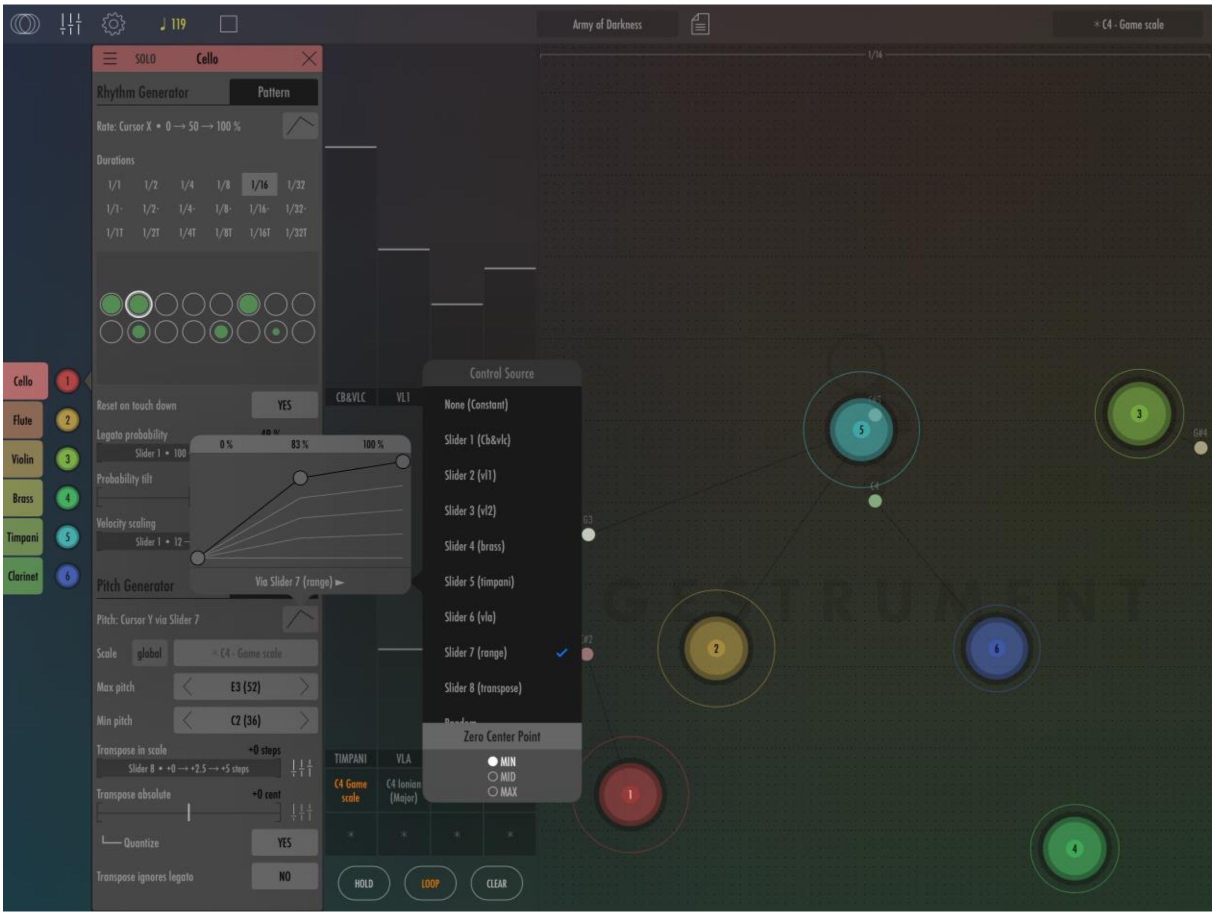


**Figure 15: The iOS app Gestrument Pro from 2018**

*5.2 Patents*

In 2018, the new iOS app Gestrument Pro was released, and before that, two patent applications were filed – patents that have since been granted in several countries. The patents are called *Instrument and Method for Real-Time Music Generation* and *Real-Time Music Generation Engine for Interactive Systems*. In the granted patents, one of the defining aspects is that the musical rule sets that define the pitches and rhythms in this technology are composable. Therefore, it is possible to find new expressions and situations, wherein the technology can be understood as something between a playable instrument and a composition. This concept and a first attempt at a classification of this situation was studied in an earlier article (Nordin, 2020). Another fundamental aspect of the technology, as described in the patents, lies in its focus on real-time driven input data. No matter if the technology is used as a DMI by an active musician, as the run-time musical engine in a video game, or as part of an interactive exposition, it is always driven by the input data.

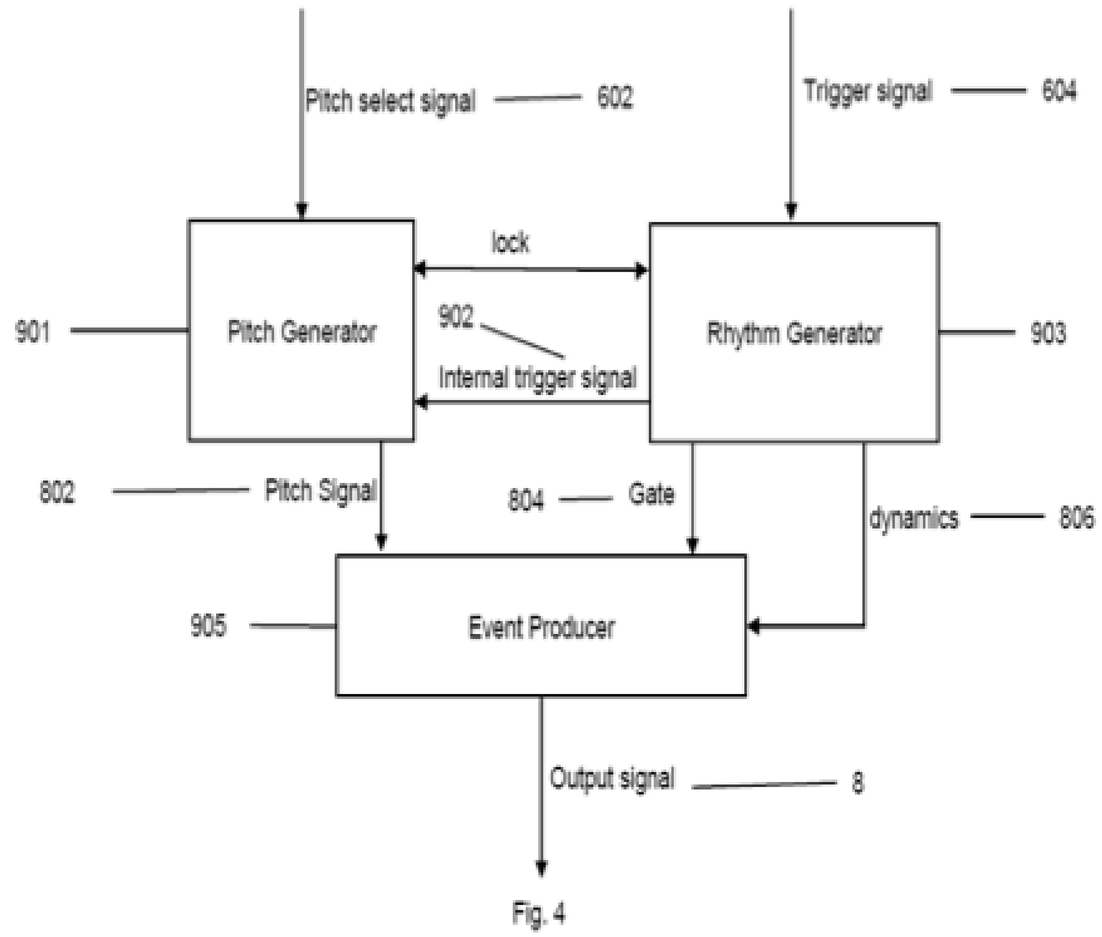


**Figure 16: Image from granted patent**

*5.3 Compositions*

The release of the app coincided with the premiere of a large-scale work, in which Gestrument Pro had been used both as a compositional tool and as a live performance tool for the soloist and the conductor, letting them drive virtual instruments with the movements of their hands. The piece is called *Emerging from Currents and Waves* (2018 a, b & c), and was composed for clarinet soloist Martin Fröst, conductor Esa-Pekka Salonen and the Swedish Radio Symphony Orchestra. One of the fundamental ideas with this project was that the soloist and the conductor would both be controlling virtual orchestras with motion sensors through the Gestrument Pro app. The virtual orchestras always matched the live orchestra – a live interaction designed already in the compositional process. Therefore, the same settings were used in the app when the soloists controlled the virtual orchestras as when I composed the piece. In that way, the notated gestures of the soloist and the conductor, as seen in Figure 17, would always produce results synchronised with the orchestra's performance of the fully notated score.

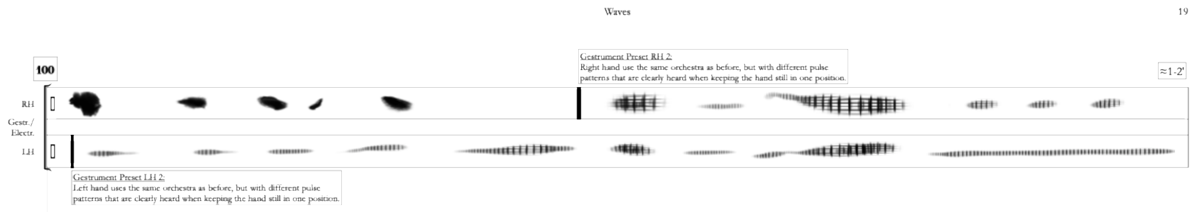


**Figure 17: Excerpt from *Emerging from Currents* and *Waves* © Edition Peters**

In this video, there are several examples of how the virtual orchestras, performed with the motion sensors and the Gestrument Pro app, are in the same style and character as the orchestral score (see Berwaldhallen, 2018: https://youtu.be/JIKPZVaVqbc).

The app Gestrument Pro was always considered as a kind of foundation that was to be as versatile and as generic as possible, since the plans for expanding the technology into new areas had already begun while the app was being developed. Therefore, the next step that was taken – directly after the release of the app – was to start porting the app to a C-library, to enable the technology to function in practically any use case on any platform.

# 6 Making it scalable – Reactional Engine

### *6.1 The Reactional Engine*

In 2019, the development of the Reactional Engine started with the aim of porting the core functionality with as few dependencies as possible, since the use cases envisioned included many different platforms. In parallel with the extraction of the codebase from the app and stripping it of any iOS-specific code and functionality, the different use cases were examined, and some major drawbacks were remedied by adding more pitch and rhythm generators that were custom-made to cover use cases where the composer/developer needed to have more precise control over the specific output. Therefore, the development process entailed adding new functionality while at the same time porting the existing functionality to the new codebase.

The base for the first Reactional Engine was called GE_Core and was visualised as in Figure 18 on the next page and described in the documentation as

> a high-performant, real-time generative music engine. GE_Core can be used as tool to compose and create music in various forms and styles with or without user interaction. GE_Core consist of a dynamic set of components which the composer/programmer can arrange and put together in various configurations. (Gestrument, 2019)

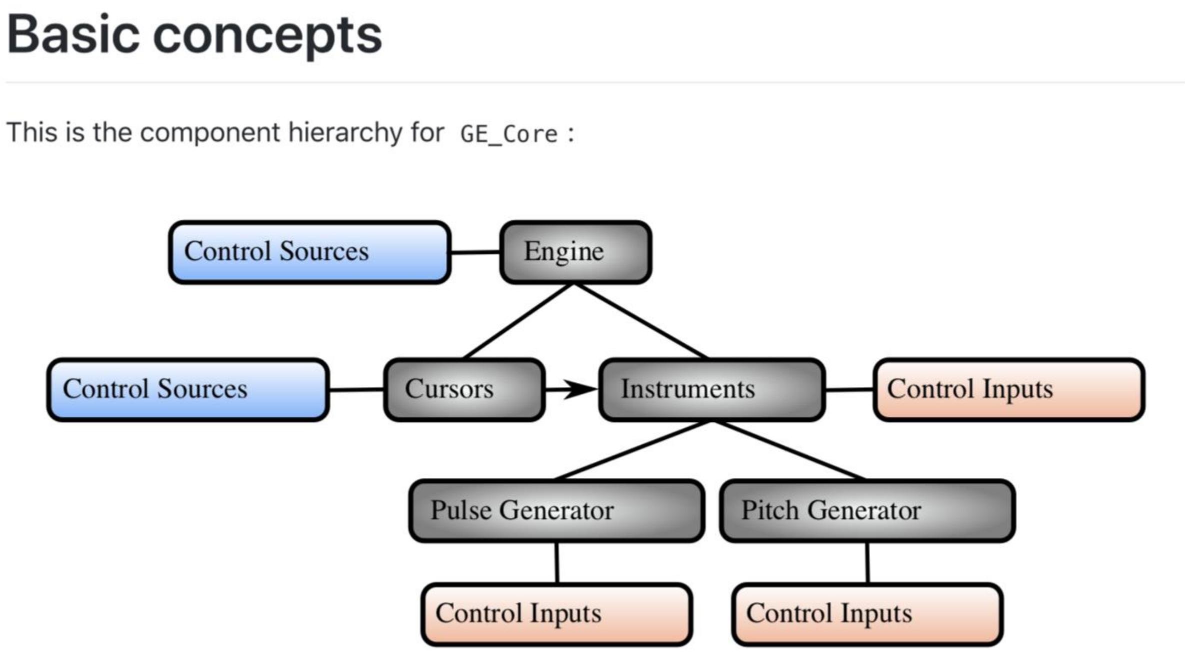


**Figure 18: Overview from the GE_Core documentation**

A fundamental design choice was to make Open Sound Control (OSC) the preferred method for communicating with GE_Core, in order to make it as versatile and easily adoptable as possible. It was possible to access the GE_Core directly as well, but the OSC API was the most fully developed. During 2022, the GE_Core – which was more or less a direct port of the Gestrument technology – was gradually replaced with a new way of achieving similar results while making it more proficient and more streamlined towards the current use case in video games and interactive applications.

Unity and Unreal Engine are foundational platforms for creating interactive systems, employing robust rule-based frameworks. Both engines feature integrated physics engines (PhysX in Unity, Chaos Physics in Unreal) that enable dynamic simulations. These physics engines allow for the creation of systems where object interactions are governed by defined physical rules, a core element of rule-based system implementation (Lengyel, 2012). Furthermore, these engines provide powerful scripting capabilities, through C# in Unity and C++ in Unreal, along with visual scripting systems like Blueprints in Unreal, which facilitate the construction of complex logical relationships. These scripting tools are critical for implementing state machines and behaviour trees, which are essential rule-based systems for controlling entity behaviours within virtual environments (Gregory, 2014). This combination of physics simulation and flexible scripting empowers developers to build highly responsive and interactive experiences where system behaviours are directly tied to defined rule sets.

### *6.2 The Dream Project*

An example of a project made possible by the Reactional Engine was the online performance *Dream*, funded by the UK government through their initiative Audiences of the Future. This project was made possible by a large-scale collaboration which included several UK Universities, and which, therefore, was thoroughly documented and researched, and a more in-depth analysis of the role of Reactional in the project can be found elsewhere (Nordin, 2024). Therefore, in the present article, the *Dream* project will only be briefly described as an example.

*Dream* was described as a "live, online performance set in a virtual midsummer forest. Theatre, music and ground-breaking technology combine in an extraordinary exploration into the future of live performance" (RSC, 2021), and it was performed live by five actors in motion capture suits in a studio at Portsmouth University. The Reactional Engine played a fundamental role and was used inside an Unreal session that also included all the visual aspects of the project. The music in the project consisted of orchestral recordings of music by Maurice Ravel, Esa-Pekka Salonen and me. These recordings were used together with real-time generated music where Reactional was driven by the movements of the actors. Sometimes this real-time generated music was synchronised with the pre-recorded material, and sometimes it was the only musical material heard. In this way, the project exemplified both situations where pre-recorded music was made interactive using the technology, as well as situations where the actors were in control over the full musical result. The actors could control whether there was any music at all, whether the music was loud or soft, slow or fast, and in which range the instruments should play.

Each actor had one or both hands tracked, and through that data, they controlled a single GE_Core cursor each. The y-axis (pitch) of the cursor was mapped to the vertical position of the hand, the x-axis (rhythm) and the z-axis (velocity) were mapped to proximity between hand and hip or to the speed of the movement. There was also a "pinch-to-silence" function that tracked the fingers for turning the cursor on and off. Each cursor then controlled an instrument or a group of instruments within the predefined constraints that were all based on the musical parameters of the three recorded pieces (mainly chord progressions and rhythmical patterns). In this way, the musical experience was controlled and coherent while still being fully in sync with the actions of the actors (see WhatsOnStage, 2021: https://youtu.be/whgeiuQRd4k?si=0oHk70KkaPJaIJQL for technical rehearsal footage and a discussion about the different technologies at play in the project).

The *Dream* project showcased the possibility of real-time music generated by non-musicians, and even if this project was a streamed performance in which the audience was not participating in creating the music, there is no technical hindrance to doing similar things and inviting the audience to participate in creating their own experience.

In my article about this project (Nordin, 2024) – some of the differences are highlighted between traditional composing of a score and the composition process in interactive situations. For instance, one fundamental aspect is that the music is not composed as a single definite version of the score; rather, creating the music for *Dream* entailed composing in different ranges of musical possibilities. Another very important aspect of a project like this is that the defining of input methods and mappings becomes a fundamental part of the compositional process, much in the same way that I have always done when setting up my own situations for generating compositional material. The difference is that it is part of the actual performance in the *Dream* project and not just part of the material-generating process.

## 7 Discussion

This article has documented the reciprocal process at the heart of the Reactional Music technology, tracing its evolution across five iterations and demonstrating its impact through concurrent compositional examples. The core findings confirm the thesis of Artistic Research: that artistic practice acts as a rigorous investigative mode, one where creative necessity actively drives technological development, and, in turn, the affordances of the resulting software – viewed as a DMI – redefine the artistic agenda. The development is thus positioned within a lineage of practice-led inquiry, such as Nilsson's (2011) approach to instrument design as a field of possibilities, but is also framed by the critical theory of interactive systems, which examines how instrument design and compositional form become inextricably linked (Rowe, 1993; Jordà, 2004). This retrospective analysis addresses the foundational challenges identified by Rowe (1993) concerning the complexities of machine listening and interaction and aligns with Jordà's (2004) discussion of how a "good musical instrument" may be defined, and the balance between specific and generic features. What emerges is a complex negotiation between the highly specific needs of a composition – driving features like rhythmic complexity in *Vicinities* – and the constant effort to generalise those features for a broader, more robust platform.

From the initial relatively simple x/y surface for generating music based on lists of note values and pitch ranges, the work on different compositions gradually added more complex possibilities as shown in the different examples in this article, such as the rhythmic approaches in pieces like *Vicinities* or *Ärr,* or the addition of real-time virtual orchestras performed by the musicians in *Emerging from Currents and Waves*. The balance between adding specific features that are valid only for a very specific idea or composition, and features that are very generic, is something that any software developer struggles with and something that is covered by Jordà (2004).

But it is also true that the technology has informed the compositions as well, since the Reactional technology can be viewed as an instrument in many of these use cases and thereby the affordances of the specific version will always influence the output. A very basic example of this would be how many individual MIDI channels, or instruments, it was possible to control at any given time in the different iterations of the Reactional technology. From the flexible Max patch where more instruments could be added at will, to the hard-coded limitations of the different iOS apps (eight instruments in Gestrument and Gestrument LE, sixteen instruments in Gestrument Pro), and then back to a flexible setup in the Reactional Engine. This is not always clear in the pieces composed with the different iterations, since I used different strategies to mitigate the limitations. For instance, I improvised an instrumental section at a time when composing *Vicinities* (2011) and *Ärr* (2014), and most of the earlier pieces were for smaller ensembles. When I did not have enough instruments in my original improvisation, I usually composed the score so that several instruments used the same basic material, either playing in unison or with slight variations.

The influence of the technology on my artistic output is valid both on a smaller scale – what is possible to achieve with this specific version – but also on a larger scale by driving my artistic practice in new directions. I did not have a specific interest in non-linear music or real-time visualisations before starting to use this technology. When I started to use a motion sensor that had visual feedback, ideas for using real-time visualisations in my music emerged, and when the Reactional technology started to be used in video games I started to think about how it would be possible to compose non-linear music in a way that left the compositional control to me but still invited the listener and/or the musician to participate in an inclusive way. Therefore, I would say that the possibilities inherent in the technology opened those doors for me, and they are now an integral and expanding aspect of my artistic practice.

Most of the initial development process was carried out without a proper understanding of the state of the art, as covered by, for instance, Rowe (1993), Jordà (2004) and Magnusson (2010). Their discussions regarding DMIs in general and specifically the prior research into the area between what constitutes an instrument and what constitutes a composition when dealing with a DMI would probably have made the development process a different journey, both from a technical and from an artistic viewpoint. For instance, I had not considered the possibility of thinking about an instrument as being partly a composition in itself when I started to work on the Reactional technology. If I had, I believe the first approach would have been very different. I am not sure that this lack of insight at the time is to be viewed as something negative, since it allowed for a very exploratory and open-minded approach to developing this technology. But I believe that the lessons learnt from other DMIs and interactive music systems have been crucial to the current state of the technology and its future direction. For instance, the attempt to map the intersection between composition and instrument using the model of *The Musical Interactivity Area* (Nordin, 2020) is heavily influenced by prior research by, for instance, Malloch, Birnbaum, Sinyor, and Wanderley (2006).

The entire work on the Reactional Music system was also done without any prior knowledge about other generative systems that utilise high-level parameters to control abstracted algorithmic processes, as seen in Lewis' *Voyager* (2000), Wiggins' (2004) work on evolutionary music systems, Eigenfeldt's (2011) investigations into generative music structures, and Pasquier and other's (2012) explorations in computational creativity. These systems emphasise the use of abstracted parameters to guide the generation of musical material, reflecting a shift from direct manipulation of sound to higher-level compositional control. Much could have been learned by studying systems like the ones mentioned above, but one clear result of not having had prior knowledge of them was that I took a different approach. The development of the Reactional technology was less focused on the generative possibilities in computer music and much more concerned with manipulating musical frameworks – scales, rhythmic patterns, ranges, or composed melodies – rather than employing Markov chains or randomisers of different sorts.

During the entire process of the development of Reactional, commercial applications of musical generative approaches have become increasingly prevalent. Commercially available systems like Ableton Live with Max for Live, hardware modular synthesisers, and advanced software synthesisers with sophisticated sequencing and modulation capabilities provide powerful tools for generative music creation. This might have influenced the development to some extent, but not in ways that are easily traceable.

A central perspective, which has not been possible to fully address in this article, is the role of gesture. The actual gestures used when performing or composing with the technology have immense importance for the artistic output. Since the Reactional technology must be considered an instrument, it becomes obvious that it is something that one can learn, practise, and be more or less attuned to. The examples used in this article range from an ensemble piece with shorter phrases – *Surfaces scintillates* – to a more eruptive solo bassoon part in *Vicinities* and a slow rhythmic exploration in the case of *Ärr*. To some extent, one can then deduce that the affordances of the technology allow for wide-ranging musical results (which is also strengthened by listening to the music posted online by users of the iOS apps), but when looking closely at the examples in this article it is also possible to find similarities. Since the movements used to generate this material are done by me, they are shaped by my embodied musical skills and performative abilities. An absence of larger leaps in the music is one very clear trait that can be traced back both to the practical input methods of the Wacom tablet and the iPad, as well as to my personal taste. Another clear similarity between the pieces is that they have a very organic development, usually also quite slow. This is not something built into the technology as such, but rather a result of my personal taste.

Returning to the question of gestures, Bacot and Féron (2016) summarise my own personal approach to the use of gestures in their description of the working process of *Sculpting the Air*:

> The main concept of this piece is to use the conductor's gestures to control different environments: the ensemble—obviously—but also a set of small suspended bells, and electronics, thanks to motion sensors. Imagining and developing a precise and specific notation for gestures was not the composer's ultimate intention. Rather, he would explore simple gestures that would produce different types of sounds, allowing the conductor nevertheless to keep a certain degree of freedom. (Bacot and Féron, 2016, p. 4)

This article has followed the development process of the Reactional technology with the aim of showing how this technology has evolved from being a personal tool to becoming a more generic platform used in projects by artists in many fields, as well as laying the foundation for a future exploration of new artistic possibilities in the realms of network and internet-based experiences, such as video games. It has not focused at all on how the technological landscape has evolved during the same time, but of course, there are many things that would not be possible without the progress in hardware and software development that has been made during this time. When the original Max patch was made, there were no iPads, and the difference in hardware specifications on iOS devices had a huge impact on the difference between the original Gestrument app and the later Gestrument Pro app.

The idea that the musical aspects in immersive experiences will become interactive to a much larger extent than today seems to be relatively widespread both in the industry and in the research community. The potential of a generative and more fully interactive approach to music in video games has been previously discussed by Collins (2008), Sweet (2015) and Phillips (2014), and the same reasoning can be applied to any audio experiences of an interactive nature:

> Although horizontal resequencing and vertical remixing offer a variety of ways for composers to create interactive music for video games, these forms have limited flexibility in terms of dynamic variability and adaptability from moment to moment during gameplay. Horizontal resequencing and vertical remixing also rely primarily on the use of prerendered audio files, which suffer from problems related to looping and branching (Sweet 2015, p. 145).
>
> This is one of the most difficult problems facing game composers. A perfected audio recording is highly desirable, and yet when we render we also eliminate all of those small musical components that can be so useful for the purposes of interactivity (Phillips 2014, p. 202).

These authors discuss the difficulties involved when making interactive experiences based on recorded music, and without having had a thought about these aspects when I started the development – or even knowing much about video game music – the Reactional technology has become a partial response to these issues. But the original aim and history of the technology is more firmly rooted in the field of contemporary music and computer-assisted composition (CAC).

## 8 Concluding reflections

A concluding reflection on the development described in the article is that I could never have envisioned the direction that this technology has taken or the traction it has received. Possibly more importantly, I could never have envisioned how fundamental the technology has become to my artistic output, and how the two strands of software development and composition have been interlinked and connected for me during these years.

Being so close to the technology and both its history and its potential, it is difficult for me to see conclusions clearly, but I would argue that in some ways Reactional Music can be seen as a contestant for becoming a standard for interactive and real-time generated music within many fields, both current and future. It can be video games, online concerts, metaverse experiences or the strand of SMI and Internet of Musical Things (IoMT) as discussed by Turchet (2018) – where the possibility of generating or manipulating musical events within predefined constraints, synchronised to fixed or adaptive audio and/or visuals, seems like a probable route for the near future.

The most obvious use case here and now is probably video games, where previous research has identified the drawbacks of pre-rendered audio files and the problems with current tools for real-time generated music, as discussed, for instance, by Phillips (2014) and Sweet (2015). As a comparison, the visual aspects of video games have moved away from hard-coded visuals towards rule-based systems like physics engines and shaders, and the most logical step is that music is heading in that direction as well.

This article has sought to explore a direct link between technological development and specific artistic goals, such as has characterised the entire development of Reactional Music up to the present day. I believe this is an aspect worth further study, with the potential to also suggest working methods in the development of other future technologies.

## References


Assayag, Gérard Rueda, Camilo; Laurson, Mikael, Agon, Carlos, and Delerue, Olivier, 1999. Computer-assisted composition at IRCAM: From PatchWork to OpenMusic. *Computer Music Journal*, 23(3), pp. 59–72. DOI: https://doi.org/10.1162/014892699559896.

Asplund, Jonas, 2022. Cyborg learners: Becoming-with in the ecology of digital music composition. *Finnish Journal of Music Education*, 25(1), pp. 10–30.

Bacot, Baptiste, and Féron, François-Xavier, 2016. The creative process of *Sculpting the Air* by Nordin: Conceiving and performing a concerto for conductor with live electronics. *Contemporary Music Review*, 35(4–5), pp. 450–474.

Berwaldhallen, 2018. *Jesper Nordin – Emerging from Currents and Waves (excerpt).* URL: https://youtu.be/JIKPZVaVqbc?si=lF0TNWT2IqNsLD68.

Brunson, William; Sandred, Örjan, and Hedelin, Fredrik, 2002. Studio Report – Kungliga Musikhögskolan in Stockholm. URL: https://quod.lib.umich.edu/cgi/p/pod/dod-idx/studio-report-kungliga-musikhogskolan-in-stockholm.pdf?c=icmc;idno=bbp2372.2002.082.

Caramiaux, Baptiste, 2014. *Towards an ecology of musical interaction.* PhD thesis, Université Pierre et Marie Curie.

Collins, Karen, 2008. *Game sound: An introduction to the history, theory and practice of video game music and sound design.* Cambridge, MA: MIT press.

Cycling74, 2024. *What is Max.* URL: https://cycling74.com/products/max.

Eigenfeldt, Arne. 2011. Real-time Composition as Performance Ecosystem. *Organised Sound*, 16(2), pp. 145–153. DOI: https://doi.org/10.1017/S1355771811000094.

Essl, Karlheinz, 2024. *RTC-Lib.* URL: https://www.essl.at/works/rtc.html.

Frisk, Henrik, 2008. *Improvisation, computers and interaction: Rethinking human-computer interaction through music.* PhD thesis. Malmö: Malmö Academy of Music, Lund University.

Gestrument, 2012. *Gestrument manual.* URL: https://www.dropbox.com/sh/rwhzx3k10xaed3e/AAB_WsyL4n5ce1z3ENRdgkR8a?dl=0.

Gestrument, 2012–2014. *Gestrument instruction videos.* URL: https://www.youtube.com/playlist?list=PLcAA_MWVU_s6UvUSC8ah1-ogzPUK-irB6.

Gestrument, 2014. *Gestrument performance of Jesper Nordin's* Ärr. URL: https://www.youtube.com/watch?v=nng6rMkIYO0.

Gestrument, 2018. *Gestrument pro manual.* URL: https://www.dropbox.com/sh/hmyuv8drbhrgnde/AACmiFShQNhmRSpRIz3uR6voa?dl=0

Gestrument, 2019. *GE core documentation.* Internal company document.

Gilligan, Jay, 2014. *Gestrument workshop.* URL: https://youtu.be/ArmIxzBRMnE?si=RqKw-jMa3qrmQZq3.

Gregory, Jason, 2014. *Game engine architecture.* 2nd ed. Boca Raton: CRC press.

Hall, Tom, 2016. Pitchcircle3D: A case study in live notation for interactive music performance. In: Richard Hoadley, Chris Nash, and Dominique Fober, eds. *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR2016.* Cambridge: Anglia Ruskin University, pp. 57–64. http://tenor2016.tenor-conference.org/articles/08_Hall_tenor2016.pdf.

Ircam, 2024. *OpenMusic.* URL: https://forum.ircam.fr/projects/detail/openmusic/.

Jensenius, Alexander Refsum, 2013. *Sounding bodies: Movement, technology and music.* PhD thesis, University of Oslo.

Jordà, Sergi, 2004. Instruments and players: Some thoughts on digital lutherie. *Journal of New Music Research,* 33, pp. 321–341.

Lähdeoja, Otto, 2019. Elektronisesti laajennetut soittimet: Kokeiluista konserttilavalle. *Musiikki,* 49(2–3): Musiikki ja teknologia III, pp. 49–70.

Lengyel, E., 2012. *Mathematics for 3D Game Programming and Computer Graphics.* 3rd ed. Boston, MA: Course Technology / Cengage Learning.

Lewis, George E., 2000. Too many notes: Complexity, difference and repetition in improvised music. *Leonardo Music Journal,* 10, pp. 33–39.

Magnusson, Thor, 2010. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal,* 34(4), pp. 62–73.

Malloch, Joseph, Birnbaum, Daniel, Sinyor, Elliot, and Wanderley, Marcelo M., 2005. Towards a Dimension Space for Musical Devices. Towards a new conceptual framework for digital musical instruments. In: *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME 05), Vancouver, Canada.* Singapore: National University of Singapore, pp. 192–195.

Nilsson, Per Anders, 2011. *A field of possibilities: Designing and playing digital musical instruments.* PhD thesis, Göteborgs universitet. Göteborg: ArtMonitor.

Nordin, Jesper, 2007. *Surfaces scintillants.* Edition Peters, EP 12825. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_surfaces_scintillantes_ep12825_score.

Nordin, Jesper, 2009. *Pendants.* Edition Peters, EP 12719R. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_pendant_i_ep12719a_score-issuu_versi; https://issuu.com/scoresondemand/docs/jesper_nordin_pendant_ii_ep12719b_score_issuu_vers; https://issuu.com/scoresondemand/docs/jesper_nordin_pendant_iii_ep12719c_score_issuu_ver.

Nordin, Jesper, 2010. En rapport från ett år på Ircam. *Nutida Musik, Nr 1+2 2009,* pp. 6–13.

Nordin, Jesper, 2011. *Vicinities.* Edition Peters. URL: https://www.wisemusicclassical.com/work/68356/Vicinities--Jesper-Nordin/.

Nordin, Jesper, 2013a. *Frames in Transit.* Edition Peters. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_frames_in_transit_ep12739_score_issu.

Nordin, Jesper, 2013b. *Ärr.* Edition Peters. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_rr_ep14035_score_issuu_version_-_fo.

Nordin, Jesper, 2015. *Sculpting the Air.* Edition Peters, EP 14106. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_sculpturing_the_air_ep14106_score_is.

Nordin, Jesper, 2016. *Visual Exformation.* Edition Peters, EP 14161. URL: https://www.wisemusicclassical.com/work/67437/Visual-Exformation--Jesper-Nordin/.

Nordin, Jesper, 2018. *Emerging from Currents and Waves.* Edition Peters, EP 14307. URL: https://issuu.com/scoresondemand/docs/jesper_nordin_emerging_from_currents_and_waves_20.

Nordin, Jesper, 2020. The musical interactivity area – An attempt to map the intersection between composition and instrument through the use of the Gestrument engine. In: Sergey Balandin, Luca Turchet, and Tatiana Tuytina, eds. *Proceedings of the 27th Conference of Open Innovations Association FRUCT, 7–9 September 2020, Trento, vol. 2,* pp. 349–355. https://fruct.org/publications/volume-27/acm27/.

Nordin, Jesper, 2024. Dreaming of new musical interaction – the interactive music in the project *Dream. International Journal of Performance Arts and Digital Media*, 20(1), pp. 5–28. DOI: https://doi.org/10.1080/14794713.2024.2328007.

Nordin, Jesper, Liljedahl, Jonatan, Kjellberg, Jonas, and Risberg, Pär Gunnars, 2018. *Instrument and Method for real-time music generation* (Aug 2018). Patent No. PCT/SE2019/050909. URL: https://patents.google.com/patent/EP3857539A4/en.

Nordin, Jesper, and Fröst, Martin, 2020. *Jesper Nordin & Martin Fröst – Klezmer solo.* URL: https://www.youtube.com/watch?v=aEK-wscIcJg.

Nordin, Jesper, and Geoffroy, Jean, 2011. *Max Patch Demonstration at GRAME, Lyon.* URL: https://vimeo.com/24004279, https://vimeo.com/24002171.

Nordin, Jesper, 2024. *Technical Rehearsal Footage for Dream.* URL: https://www.youtube.com/watch?v=whgeiuQRd4k.

Olivier, Nicolas, 2014. *Gestrument video.* URL: https://youtu.be/FdHLdA3WI_c?si=G4nrLJk0OLM_sBKg

Pasquier, Philippe, Roy, Pierre, and Hudon, Sylvain. (2012). Computational creativity for interactive music generation. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1), pp. 1–13.

Petersson, Mattias, 2025. *The act of patching: Musicking with modular systems.* PhD thesis, Luleå University of Technology. URL: https://ltu.diva-portal.org/smash/record.jsf?pid=diva2%3A1952531&dswid=7972.

Phillips, Winifred, 2014. *A composer's guide to game music.* Cambridge, MA: MIT press.

Reynolds, Craig W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4), pp. 25–34. DOI: https://doi.org/10.1145/37402.37406.

Rowe, Robert, 1993. *Interactive music systems: Machine listening and composing.* Cambridge, MA: MIT Press.

Royal Shakespeare Company, 2021. *Press release regarding the project Dream.* URL: https://www.rsc.org.uk/press/releases/live-performance-and-gaming-technology-come-together-to-explore-the-future-for-audiences-and-live-theatre.

Sweet, Michael, 2015. *Writing interactive music for video games: A composer's guide.* Upper Saddle River, NJ: Addison-Wesley.

Tahiroğlu, Çağrı, Donin, Nicolas, and Rasamimanana, Nirina, 2020. More-than-human musical interactions: Agency and distributed creativity in human-computer improvisation. *AI and Society*, 35(3), pp. 651–663.

Turchet, Luca, 2018. Smart musical instruments: Vision, design principles, and future directions. *IEEE Access* 7, 8944–8963. DOI: https://doi.org/10.1109/ACCESS.2018.2876891.

Turchet, Luca, McPherson, Andrew, and Fischione, Carlo, 2016. Smart instruments: Towards an ecosystem of interoperable devices connecting performers and audiences. In: Rolf Großmann, and Georg Hajdu, eds. *Proceedings of Sound and Music Computing Conference, 2016*. Hamburg: Zenodo, pp. 498–505.

WhatsOnStage, 2021. *RSC's virtual experience Dream, Exclusive behind-the-scenes video.* URL: https://youtu.be/whgeiuQRd4k?si=0oHk70KkaPJaIJQL.

Wiggins, G. A., 2004. Towards a more creative evolutionary computation: Musical applications. In: Gary B. Fogel, and David W. Corne, eds. *Evolutionary Computation in Bioinformatics*. Amsterdam: Morgan Kaufmann, pp. 235–258.

## Biography

Jesper Nordin is a leading Swedish composer and researcher who has garnered considerable international acclaim in recent years. His music, with its powerful emotional impact and traces of traditional Swedish folk music, rock music and improvised music, is broadcast and performed throughout the world by conductors like Esa-Pekka Salonen, Daniel Harding and Kent Nagano and musicians like Martin Fröst, Pekka Kuusisto and Malin Broman. His orchestral pieces have been played and/or commissioned by such orchestras as the San Francisco Symphony, Orchestre Philharmonique de Radio France, BBC Scottish Symphony Orchestra, Sinfonieorchester Basel, Royal Stockholm Philharmonic and the Swedish and Finnish radio orchestras. He is also included in the repertoires of several of the world's foremost contemporary music ensembles, amongst them Ensemble Intercontemporain, ensemble recherche, L'Itineraire, Quatuor Diotima, Talea and The San Francisco Contemporary Music Players.

His iOS apps Gestrument and ScaleGen are used by musicians and amateurs in many different genres and form the base of the company Reactional Music. He is currently a PhD student in Music Performance at the Piteå School of Music, Luleå University of Technology.

## Acknowledgments

I would like to thank Stefan Östersjö, whose guidance and participation in the development of the article has been very important. I also want to thank the entire development team at Reactional Music for feedback, access to documents and information.