Mobile-Based Interactive Music for Public Spaces

ROOZBEH GHARIB

A thesis submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Arts

Graduate Program in Music
York University
Toronto, Ontario

October 2015

© Roozbeh Gharib, 2015

Abstract

With the emergence of modern mobile devices equipped with various types of built-in sensors, interactive art has become easily accessible to everyone, musicians and non-musicians alike. These efficient computers are able to analyze human activity, location, gesture, etc., and based on this information dynamically change, or create an artwork in realtime. This thesis presents an interactive mobile system that solely uses the standard embedded sensors available in current typical smart devices such as phones, and tablets to create an audio-only augmented reality for a singled out public space in order to explore the potential for social-musical interaction, without the need for any significant external infrastructure.

Acknowledgment

My sincere thanks are due to the following people for their cerebral and ethereal help in writing this thesis: my supervisor, Professor Michael Coghlan, who with his skilled insight always reminded me that I have wings to lift me up from the darkness; Dr. Mark Chambers, my second supervisor, for his patience and editing skills, who along with Dr. Darcey Callison inspired me to write the conclusion chapter after I heard their discerning questions and comments; the faculty and staff of York University's Music Department for maintaining a well suited environment; my dear family and friends, who didn't untie me from the mast when I was overwhelmed by the rouge waves; and lastly but not least, the one who stood beside me amongst the mist when I was troubling myself finding the path, next to all the others who disappeared in the shadows of my fading memories.

Excuse my wandering.
How can one be orderly with this?
It's like counting leaves in a garden,

along with song notes of partridges,
and crows. Sometimes organization
and computation become absurd.

(Jalāl al-Dīn Rūmī translated by Colman Barks)

Table of Contents

List of Figures

Chapter 1: Introduction

Interactive art is an emerging genre that requires participation, interaction, or some kind of conversation between an artwork and the audience. By offering an opportunity to become actively involved in the creative process of artistic performance, it engages and empowers amateur 'performers' who have no prior knowledge or particular expertise, to influence the image or sound output from a computer.

In order to change the traditionally passive position of the audience, a composer must give up the rigorous control of the form or conventional presentation, and allow participants using their mind and body to create their own experience. In a study by Peery and Peery (1986), the authors argue that interaction with music may increase the understanding and appreciation of classical music in preschool children. It suggests, that by creating relationships rather than finished works, the interactive artist empowers the audience to 'accept' and thereby pay more attention to the result of their own creative actions.

Interactive systems are usually located at a fixed position, thus the 'installation' artist can create a multilayered work that engages participants on many cognitive, physical, and emotional levels. In closed installation sound and light can be carefully controlled to evoke an intimate experience that resembles a 'reflective' world without distractions of everyday life. In another study by David Rokeby, the author argues that as soon as interactive art become necessary and non-problematic for the future generation, and interaction in art will be necessary and obvious: "Once the hype dies away, interaction in art can return to its natural role as a tool for exploring and critiquing relationship itself, an important role during a time when the nature of all our relationships… personal, economic, political, and with the media are in constant flux" (Rokeby 1995).

Although most people avoid behaviours that will draw the attention of strangers in public,[1] interactive artists can give permission to players to do activities that go beyond normal expected behaviour suitable for public spaces.[2] According to another study (Her and Hamlyn 2010), when the audience encounters computer-based interactive media arts in a public space, various characteristics such as playfulness, dominance transfer, mind-orientedness, and accessible challenge draw them to play and experiment with the artwork. Furthermore, the authors argue that the general public is starting to recognize and become familiar with this new genre as more and more computer-based interactive media art installations are seen in public spaces, therefore, they recommend that at this point the main issue is not only 'engaging' the audience in the creative process, but how 'meaningful' is the experience. On that same notion, Muller and Edmond (2006) suggest that, "we must begin to question how interactivity as a medium produces meaning".

In the first chapter of this thesis, I attempt to look into interactive music and its position in the broader context of electronic and computer music. In the second chapter, notation and documentation of electroacoustic music is discussed to consider how to present an interactive computer music compositions in text format, and how to document and preserve the electroacoustic music repertoire in a broader context. Also I

---

1. "Usually, people do not feel free to dance in front of an audience, and they know not to touch artwork in a museum" (Winkler 2000).

2. "The permission may be liberating or intimidating, depending on social and personal factors, and the intention of the artist" (Winkler 2000).

look at *Pure Data*'s (aka Pd) potential for composing musical scores, which is the central programming language that I have used for writing the music pieces that accompany this text. In the third chapter the methodology and practical considerations that I have made during the research were described by constructing an interactive music platform in which a number of users can interact and respond to each other with their mobile devices over a wifi network. In the final chapter, my own compositional framework for this thesis is described which is a location-based composition for York University's Keele Campus in Toronto, Canada.

1.1 - Brief history of electronic and early computer music

According to Futurist theorists, modernizing the form and content of the arts was an inevitable consequence of the changes brought about by the Industrial Revolution. Following the evolutionary spirit of the movement, Balilla Pratella in the "Technical Manifesto of Futurist Music", published in March 1911, suggested that composers should "master all expressive technical and dynamic elements of instrumentation and regard the orchestra as a sonorous universe in a state of constant mobility, integrated by an effective fusion of all its constituent parts" (Pratella 1911). In 1916, Edgar Varèse, a rebellious young composer against the traditional outlook of the Paris Conservatoire, was quoted in the New York Telegraph as saying that, "Our musical alphabet must be enriched ... We also need new instruments very badly ... In my own works I have always felt the need for new mediums of expression" (Varèse 1916).

Varèse's interest in electronic instruments is well documented in the coda of his piece *Ecuatorial* (1934), in which two ondes martenots take over the orchestra with an ascending continuous glissando, leaving the audience with the fluid world of electronic sounds, perhaps pointing toward the music of the future and the unknowable future of humankind:[3]

> His (Brian Kane's) observation is that this radical rupturing of the musical texture offers us a "sonic image of the tribal's survival" My (Cristopher Lyndon-Gee's) own view is that, perhaps, in a mystical way, through his depiction of the lost spiritual world of a time deep in the past, Varèse was seeking to show a path to the even more unknowable future of humankind. (Lyndon-Gee 2007)

However, Varèse soon pointed out that the futurists have made a serious mistake:

> The Futurists (Marinetti and his noise artists) have made a serious mistake. … Instruments, after all, must only be a temporary means of expression. Musicians should take up this question in deep earnest with the help of machinery specialists ... What I am looking for are new technical means which can lend themselves to every expression of thought. (Varèse 1916)

By anticipating the liberation of performers from instruments' technical barriers, Varèse argues that technical developments demand new forms of composition and was confident that composers' ideas may not be fulfilled perfectly if they are being forced to use the existing instruments of their time:

> The growth of musical art in any age is determined by the technological progress which parallels it. Neither the composer nor the performer can transcend the limits of the instruments of his time. On the other hand technical developments stimulate the creation of certain forms of composition and performance. Although it is true that musicians may have ideas which hurdle these technical barriers, yet, being forced to use existing instruments, their intentions remain unrealized until

3. For more details see (Kane 2015), and discography: (Varèse 1996).

> scientific progress come to rescue… If we admit that the creative imagination of the composer may form musical ideas which, under the specific conditions of a given epoch, cannot be translated into sound, we acknowledge a great dependence of the artist upon the technical position of his era, for music attains reality only through the process of sound. (Varèse in Manning 2013, 8-9)

Throughout the history of music, technological landmarks such as the use of written notation,[4] the development of polyphony, organ building improvements, equal temperament tuning, etc. consistently influenced the evolutionary path of music-making (Manaris and Brown 2014, 9).

Although concepts such as, (1) Sound recording, (2) Sound synthesis, and (3) Automated music, are building blocks of computer music today, they were applied to music production long before the first public performance of real computer music in August 1951 by Geoff Hill and Trevor Pearcey at the Australian Computer Conference:

1. With the advent of the phonograph in 1877, sounds could in fact have their own existence separated from their source. Pierre Schaeffer realized this phenomenon and coined the term "*objet sonore"* (lit. sound object) which refers to the physical-material source, not the effect of the sound. Schaeffer argues that by listening to a recording we are listening to the effect of a sound, since the action of making the sound was in the past. While surrounded by the turn-tables, the mixer, and the potentiometer, he is quoted by saying that, "I operate through intermediaries. I no longer manipulate sound objects myself. I listen to their effect through the microphone" (Kane 2014, 15). In Schaffer's view, by removing the sound from the laws of causality the 'sound object' can

---

4. See chapter 2 of this thesis, "Notation of electroacoustic and computer music" for full detail.

be listened to and studied with an ‘acousmatic' intention, in other words perceived and appreciated for its own sound-value.[5]

2. Another landmark in the history of music was an electrically based sound-generation system registered by Thaddeus Cahill in 1897, and presented to the public in 1906. Cahill saw his invention (known as the Dynamophone or Telharmonium) as an instrument that could produce the notes and chords of a musical composition with any timbre. The desire to create additions to the conventional orchestral range motivated several other engineers to design more electronic devices such as the Theremin (1924), the Spharophon (1927), the Dynaphone (1927-8), the Ondes Martenot (1928), and the Trautonium (1930). Except the Theremin, none of these instruments were widely used, up until the first popular electric keyboard invented by Laurens Hammond in 1935. Although the Dynamophone made use of a similar method of tone generation to the Hammond organ, the Hammond was much smaller and commercially more successful.[6]

3. Automatic instruments have existed for a long time, developed historically parallel to the gramophone, the *player piano* is arguably the most sophisticated. It brought music on demand to many homes with far superior sonic quality than gramophone. Player pianos changed the role of the audience by bringing musical performances into homes, affecting concert attendance and the social status of musical performance skills. The player piano roles could capture pitch, duration, and force (velocity) for each note, and more than any electronic recording technologies can be thought of as the parent of MIDI (Musical Instrument Digital Interface) sequencing. Soon

5. “Originally, “acousmatic” was the name given to the disciples of Phythagoras who, for five years, had to listen to the lessons from behind a curtain, without seeing the master and in absolute silence” (Bittencourt 2005).

6. Hammond organ’s method of tone generation is by utilizing the rotation of suitably contoured discs within a magnetic field.

composers realized that with the help of editable piano roles, they could produce music beyond the humanly performable, and they could involve themselves in all the steps from conception to final sounding (Manaris and Brown 2014, 10-11).

In mid-1950s, Lejaren Hiller and Leonard Isaacson did their first experiments with computer-based music composition on the ILLIAC computer at the University of Illinois by employing both a rule-based system utilizing strict counterpoint along with a probabilistic method based on Markov Chains (also employed by Iannis Xenakis around the same time). The result was a series of four pieces for string quartet published in 1957, which was the first computer-aided composition in the history of music.

A few years later during 1956 and 1957, Max Mathews developed the first real digital synthesis program (Music-I) in the Bell Laboratories. Although according to Mathews only one "terrible" piece of music was composed using the language,[7] the principles involved became the bases for all music programming languages that were to follow. Music-I was quickly replaced by Music-II and later on Mathews and others developed Music-III, -IV, -V, and derivatives of these (aka Music-N family).[8] Since then there are mainly two primary manners of producing computer music: computer-generated sound, and computer-controlled synthesizer-generated sound.

With Robert Moog's introduction of modular voltage controlled synthesizer in 1964-1965, the required hardware resources for electronic music came to be accessible to the "average composer". Compared to other electronic instruments of the time, the technical breakthrough of Moog's synthesizer was the use of transistors instead of

7. Max Mathews' remarks made at International Computer Music Association Conference at the University of Illinois (Kent 1994).

8. See discography: (Pierce, et al. 1983).

vacuum tubes, which dramatically reduced the size of the instrument and increased the stability of voltage control. Moog synthesizers, and several other competitors that quickly appeared in the market (e.g. the Arp, and Buchla synthesizers), were quickly adopted by the world of pop music.[9]

Parallel to Robert Moog's remarkable analog synthesizer, Max Mathews published a highly stimulating *Science* article, titled: "The Digital Computer as a Musical Instrument" (1963). In his article Mathews argues that, "A computer can be programmed to play "instrumental" music, to aid the composer, or to compose unaided." He further claims that, "almost any sound can be produced by treating the numbers generated by a computer as samples of the sound pressure wave". In another influential article, Mathews (1970) states that there are mainly two fundamental criteria for computer sound synthesis: first, the necessity of a very fast program, and secondly, the need for a simple, powerful language.

Digital technologies first made their way into synthesizer design as memory banks, and later in the sound synthesis engine of the first truly digital synthesizer, Yamaha DX7. The DX7 introduced a new type of synthesis called "Frequency Modulation" developed by John Chowning at Stanford University in 1970s. The release of DX7 in 1983 was coincided with the introduction of MIDI, developed by Dave Smith, which is another significant point in the history of electronic music (Manaris and Brown 2014, 15).

It is worth mentioning that the timeline of computer music, as much as it is filled with technological developments, is packed with innovative experiments based on the

9. One of the early popular recordings using the Moog synthesizers was Wendy Carlos's "Switched-on Bach" which was a notable achievement at the time. See discography: (Carlos, et al. 1999).

theoretical rules of music theory and mathematics. Among the theoretical and structural perspectives used in the field of computer music, ideas such as Serial music techniques, the application of music grammars (notably the Generative Theory of Tonal Music by Fred Lerdahl and Ray Jackendoff), concept of structural levels of Schenker analysis, sonification of fractals and chaos equations, stochastic processes (e.g. Markov chain theory), connectionist pattern recognition techniques based on works in neuropsychology, and artificial intelligence (AI) are widely used in the repertoire of intellectual technologies applicable to the computer music.[10] According to Manaris and Brown (2014), the most comprehensive of automated computer music programs so far is David Cope's Experiments in Music Intelligence, also known as *Emmy*, which can compose infinite number of pieces in the style of composers such as J. S. Bach,[11] Chopin, etc., based on a feature analysis on a database of a particular composer's works (Cope 2004).

1.2 - New digital musical instruments and programming tools

With only a few exceptions that were modelled on other instruments, electronic musical synthesizers have always been designed to be played with a piano-like keyboard. With the advent of software synthesizers and the availability of faster and more affordable personal computers, along with the emergence of the MIDI protocol, and more recently with the introduction of OSC (Open Sound Control),[12] musicians can

---

10. See (Rowe 1993, 98-106), and (Manaris and Brown 2014, 12) for more details.

11. See discography: (Cope 1994).

12. Open Sound Control (OSC) is a protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology (see opensoundcontrol.org).

use the computer as a musical instrument without any difficulty. Because software synthesizers are more flexible and can be easily programmed, lately musicians and researchers have begun to realize their own unique digital musical instruments. Although research centres such as STEIM (STudio for Electronic Instrumental Music), and the Hyperinstrument group at MIT's Media Lab proceed their advanced research into new instrument design, plenty of the latest research developments that can be seen in the annual New Interfaces for Musical Expression (NIME) conference, are made by individual researchers and musicians. Although there were a few attempts by the industry to model known types of instruments including guitar, woodwinds, and percussion to build new controller extensions, but so far most of the development in the DMI design were done by individual musicians and researchers who employed programming tools such as *Max/MSP*, *SuperCollider*, *Pure Data*, etc. to build their own instruments, and even use 3D printing, not only to replicate existing forms, but to design cutting-edge unimagined artifacts.

There are various ways to utilize a computer to make music. The easiest way is to use ready-made production softwares know as Digital Audio Workstations (DAW) such as Pro Tools, Logic Pro, Ableton Live, Audacity, etc. to record and produce a piece of music. But on the other side, some musicians prefer versatile and more powerful programming environments such as CSound, Max/MSP, SuperCollider, Extempore, Pure Data (Pd), etc. to realize their music. In their book, *Making Music with Computers*, Manaris and Brown anticipate development of more modern digital musical instruments:

> The ability of computers to follow arbitrary musical (or other) processes makes it possible to design and implement new musical instruments,

> running on regular computing platforms, such as a laptop or a smartphone" (Manaris and Brown 2014, 19).

Before mid-1970s a realtime interactive computer system that could play music was not a practical reality. Although gestural control of music synthesis was allowed by dozens of analog electronic instruments, the first digital systems to allow real-time gestural input were the *hybrid* music system of the 1970s, which combined a digital computer with an analog sound synthesizer. By late 1970s, commercial hybrid systems such as GROOVE (hybrid) system at the Bell Telephone Laboratories allowed musician to use higher levels of musical control. For example, the CONDUCT program for the GROOVE system allowed overall control of amplitude, tempo, and instrument balance, in the same manner that a conductor could globally control an orchestra (Roads 1995, 614, 973). Laurie Spiegel, an early electronic-music composer, then spent two years from 1974, making changes to the GROOVE system and created the VAMPIRE (Video and Music Program for Interactive generated Realtime Exploration/Experimentation) for simultaneous real-time algorithmically generated sound and image (Collins and Rincón 2007, 132).

The desire to use the repeatability and precision of digital synthesis, along with acceptable processing power for real-time performance, launched further technological progress in DMI design. Notably, Di Giugno's 4A, 4B, 4C, and 4X synthesizers constructed at IRCAM (Institut de Recherche et Coordination Acoustique/Musique), allowed for the real-time update of synthesis parameters without disrupting the audio output. By the mid-1980s, more computer languages such as MIDI LOGO and MIDI LISP were developed at the IRCAM that could interact with synthesizers. After some time in 1986, Miller Puckette developed a graphical language for controlling the 4X synthesizer through MIDI which ran on Macintosh. This language initiated the particular revolution of the *Patcher* environment which allowed programmers to build a program without typing text, instead graphically connecting functional boxes. Later, when David Zicarelli had been to IRCAM to talk about his early algorithmic music application *M*, Puckette talked with him about putting together and commercializing the *Patcher* and *M*. Named after Max Mathews, *Opcode Max* was a mixture of *M* and *Patcher* that was released as a separate commercial product in 1989. It was immediately received as a revolutionary product for generating, managing, and manipulating MIDI data and continued to develop through intervening years (Dean 2009, 54-57).

Soon the *Max* program became an iconic toolkit suitable for interactive music that has been used by hundreds of composers around the world to create and control patch editors, responsive instruments, interactive composition, and control of audio devices (Roads 1995, 688-689).[13] One of the early implementations of *Max* include ‘unit generators’ for audio-signal-processing which was first introduced in Music-III. Programmed by Max Mathews and his colleague Joan Miller in 1960, unit generators (UG) according to Roads, are “signal processing modules like oscillators, filters, and amplifiers, which can be interconnected to form synthesis *instruments* or *patches* that generate sound signals” (ibid., 89). The output of a UG can be connected (or patched) to the input of virtually any other unit allowing infinite number of synthesis methods.

Around 1994, Miller Puckette left IRCAM and joined University of California, San Diego (UCSD) to develop another programming environment, *Pure Data* (Pd), which

---

13. Curtis Roads explains Max by the cliché, “One picture is worth a thousand words”, meaning that a simple patch might require a lengthy textual description if it is assumed that the reader is not familiar with general principles of Max (Roads 1995, 689).

was introduced in 1997. It was released as an open source project, functioned and looked almost the same as Max, Pd can equally handle *control*, *audio*, and *video* without any specialized hardware and has a very strong user/developer community. Right after the release of Pd in 1997, Zicarelli developed a new version of Max with Digital Signal Processor (DSP) objects based on Pd, known as Max/MSP (Zicarelli 2002). *Max/MSP* (Max Signal Processing) is in fact a commercial counterpart to Pd with a robust user community of its own.

It is crucial not to forget that all the technologic breakthroughs in computer music became accessible only with the increasing power and affordability of personal computers in the late 1990s. Since then, many of these technical threads such as computer-based composition, recording, publishing, and multimedia were integrated into various softwares. Consequently, more digital music systems were developed that provide rich and expressive tools for musicians. Arguably, this process continued until computing power reached a threshold; personal computers are now powerful enough to manage most audio and video processes in real time, and most recently mobile devices such as smartphones and tablet computers increasingly become the site for established computer practices.

1.3 - Interactive music

There are mainly two genres of electroacoustic music: acousmatic and live electronics. Nowadays, these are primarily computer-based and both make use of electronic technology to generate, explore, and manipulate sound material, while the central medium of transmission is by loudspeaker. According to Miranda and Wanderley

(2006), acousmatic music is intended for listening exclusively through loudspeakers and exists only in recorded form, whereas in live electronic music the technology is used to generate, transform, or trigger sounds in the act of performance.

Since the early 20th century, there were attempts to eliminate the human performer from music. Accordingly in the early days, some composers created pieces for the mechanical piano in order to minimize the intervention of performer's personality. This trend coincided with the emergence of acousmatic electroacoustic music, which could simply remove the need for the human performer, and consequently paved the way for investigation of new modes of expression and artistic exploration of the technology for recording and playing back music which started to appear around the 1920s (ibid., 220). Soon among the emerging new possibilities, composers were encouraged to think of disparate elements such as space, as an effective compositional parameter (Trochimczyk 2001). New practices for live performance and composition with electronics and computing equipment gave birth to the practices of sampling, mixing, and remixing music live, popularized by DJ culture (Reighley 2000), and the live coding of laptop musicians (Collins 2003).

With the arrival of computer-mediated musical performances and developments of computer music research of the late-1970s, interactivity has become a major consideration and many composers have applied various technologies, from artificial intelligence to haptic interfaces to make music interactively. Based on research into artificial intelligence, the music community developed systems that could compose musical material by utilizing algorithmic approaches which respond to some form of in-performance control. The output of such systems often consists of either a predefined collection of musical material triggered on the basis of distinct conditions, or algorithmically generated material like suitable chord progressions or systems that can make compositional decisions during the performance.

The term interactive has various definitions in different fields, however, according to *Collins English Dictionary,* they all outline an action that involves reciprocal influence. In human conversation the interaction is an extremely dynamic process of the exchange of ideas, while in the field of physics the term leads us to understand that, "an exchange of energy takes place." Nonetheless, an interactive music may be referred to a system which behaviour changes in response to musical material. Still, interactive music may not necessarily display any form of musical understanding: "A sensible compromise is to consider that a music system is interactive if it is programmed to interpret the actions of a performer in order to control musical parameters" (Miranda and Wanderley 2006, 224).

Based on methods of response and types of input interpretation, in his book *Interactive Music Systems,* Robert Rowe proposes an organized framework which interactive systems can be discussed and evaluated, mainly to recognize their similarities and identify the relations between new systems and their predecessors. Rowe's classification framework consists of three dimensions, described by using some points along the continuum of possibilities for each dimension: "any particular system may show some combination of the attributes outlined here; however, these metrics do seem to be useful in identifying characteristics that can often distinguish and draw relations between interactive programs" (Rowe 1993, 6-8).

Rowe's first dimension distinguishes *score-driven* from those that are *performance-driven*. Performance-driven programs do not anticipate the realization of any particular score (by mainly depending on the performance itself), while score-driven programs employ the temporal flow of the music and make use of traditional metric categories of beat, meter, and tempo by utilizing predefined event collections or stored music fragments to match against the music arriving at the input.

The second dimension differentiates between response methods as being *transformative*, *generative*, or *sequenced*. Transformative methods take existing musical material as complete musical input and apply transformations to produce variants which may or may not be recognizably related to the original. On the other hand *generative* methods use stored fundamental material such as stored scales or duration sets to produce complete musical output, for instance, taking pitch structures from stored fundamental material by using random distribution techniques. Lastly, *sequenced* techniques make use of prerecorded musical fragments which being used to produce a response to some real-time input, meanwhile some aspects such as tempo of playback, dynamic shape, etc. may also be varied in real-time.

Rowe's third dimension draws distinction between *instrument* and *player* paradigms. In instrument paradigms gestures from a human player are analyzed by the computer, while systems that try to construct an artificial player with a personality and behaviour of its own follows the player paradigm.

Similar to Rowe, Todd Winkler defines interactive computer music as, "a music composition or improvisation where software interprets a live performance to affect music generated or modified by computer" (Winkler 1998, 4). In his book, *Composing*

Interactive Music: Techniques and Ideas Using Max, Winkler argues that interactive music usually involves a performer playing an instrument while a computer generates music that in some way is shaped by the performance. Bearing in mind that music has always been an interactive art in which musicians respond to each other as they play (even when the performer is not playing any musical instrument, i.e. in the interaction between a conductor and an orchestra), although traditional musical relationships seems to be valuable starting points for any interactive music composition, Winkler also proposes that, “interactive techniques may suggest a new musical genre, one where the computer's capabilities are used to create new musical relationships that may exist only between humans and computers in a digital world” (ibid., 5).

An important characteristic of interactive music is that the traditional roles of composer and performer often get blurred. For example, the audience may take on the role of composer and performer, and by changing a variety of parameters, hold the overall control of the process through a computer terminal:

> Computer users may be asked to select and process music, allowing non-musicians the feeling of actively participating in creating music. Computer users may be asked to become performers, playing music from a computer keyboard or other computer device. They may also be asked to become composers, selecting, ordering, and generating computer music form on-screen controls. (Winkler 1998, 9)

Meanwhile, Winkler admits that it is the subtle nuances of artistry performance that brings music to life. He argues that researchers have discovered some rules that may transmit musicality to computer music, but it is still impossible to come close to the delicacies created by the human performer, since the qualities of a good player are mostly coming from an emotional experience which comes from a lifetime of making music. Nonetheless, according to Winkler, specific elements of live performance can be used to bring the human musical sense to a machine, while at the same time there is no doubt that the computer offers new possibilities for musicians to expand their abilities beyond the physical limitations of their instruments (ibid. 8).

1.4 - Brief history of ‘interactive’ music composition

In a live performance, musicians not only interact with each other and their instruments, they also interact with their audience in real-time. But with the advent of recording and other technical innovations, interaction between the audience members and the performers by any means has been eradicated. Although multitrack recording has removed the interaction between musicians in the recording of the music, in the field of computer music, although it did not begin as an interactive art form, limitations of early systems and the experimental nature of the early computer music programs created unpredictable results that lead to meaningful interactions between composers and computers (Winkler 1998, 10).[14]

Early analog electronic experiments of composers such as John Cage in the sixties,[15] arguably triggered by the desire for improvisation and indeterminacy, naturally led to the development of first interactive electronic systems which later showed promising modern ways of musical thought. By 1967, an early interactive composer, Gordon Mumma described his piece *hornpipe*, as “an interactive live-electronic work for

---

14. This technique is still being used by live coding artists to add some unpredictable flavour to their music.

15. See discography: (Cage 2014).

solo hornist, cybersonic console, and a performance space".[16] Mumma's "cyber-sonic console" contained microphones for 'listening' to analyze the acoustical resonance of the space, beside the sounds made by the horn, the acoustics of the room altered the hornist's performance and created an interactive loop that was further processed by electronics (Cope 1977).

On the same path, the most significant step that opened the way for analog interactive techniques was the introduction of voltage-controlled synthesizers. A voltage control (VC) is an electrical signal that can be used to automate almost anything that can be modified on an analog synthesizer module. For example, in several of his compositions, Morton Subotnick used his voice to control various parameters of synthesized sounds on a Buchla analog synthesizers, even in his early tape compositions.[17] Together with Donald Buchla, a long time collaborator and electrical engineer, Subotnick composed an evening-length multimedia opera, *Ascent Into Air* (1983) in which two cellists, who are part of a small ensemble of musicians, control the interactive computer processing of live instruments and computer-generated music. In his later collaboration with composer/programmer Marc Coniglio, musicians were controlling not only the musical processes, but the playback and display of multiple video images. Also the software development for *Hungers* (1986), which eventually became *Interactor*, is a large computer program for interactive composition (Winkler 1998).

---

16. See discography: (Brown, et al. 2010).

17. See discography: (Subotnick 1999).

As soon as microcomputers became available in the mid-seventies, many composers launched their early computer experiments. In his work, *Figure in a Clearing*, David Behrman programs the computer to create harmonies and timbres that depend on the order and choice of the notes improvised by a cellist.[18] Similarly, in his book *New Sounds*, John Schaefers explains: “I used the computer as an interface between some circuity I had built that made electronic music, and a pitch-sensing device that listens for pitches made by acoustic instruments” (Schaefer 1987).

Following his early experiments with the Moog analog system, and by using the automated control of timbre and rhythm, composer Joel Chadabe, along with Roger Meyers, developed another interactive music software to empower one of the first portable digital systems of their time. In his article “Interactive Composing: An Overview” (1983), Chadabe describes his piece, *Rhythms* (1981) as follows, which incidentally according to Rowe’s categories of interactive systems is an example of generative-interactive music.[19]

> In *Rhythms*, the computer automatically generates melodies and rhythmic patterns, articulated in sounds reminiscent of Indonesian, Caribbean, and African percussion instruments. I perform by pressing keys at the terminal keyboard, thereby transposing chords, changing pitch relationships within chords, triggering melodic variations, altering rhythmic patterns, overlapping voices, and introducing random notes. But although I trigger each set of changes to begin, I cannot foresee the details of each change. I must react to what I hear in deciding what to do next. It is distinctive characteristic of interactive composing that a performer, in deciding each successive performance action, reacts to information automatically generated by the system. (Chadabe 1983)

---

18. See discography: (Behrman 1978).

19. See discography: (Chadabe 1981).

Perhaps the potential for improvisation alone has attracted several composers to use interactive systems to broaden the scope for autonomous activity. Started as a hybrid digital-analog system that consisted of a simple computer, sending voltages to an analog synthesizer, George Lewis, a well known improvisor, jazz trombonist, and software developer began building his interactive system in 1979 by using a computer with only 1 kilobyte RAM to control a Moog synthesizer. Lewis's program continues to evolve into a more elaborate system with results that can be regarded as compositions in their own right. According to Lewis, his software enables him to use his improvisational skills to create a true dialogue with the computer (Lewis 1994). Lewis's program has its own behaviour as the initial intent was to build a separate, recognizable personality that participates in the musical discourse along with the human players, which sometimes is influenced by their performance. It can also be said that according to Rowe's categories of interactive systems, Lewis's program follows the player paradigm, and belongs to generative-performance driven systems.

> The generative nature of the algorithm ensures that the program has its own harmonic and rhythmic style, since these are part of the program, not adopted from the input. Further, the stylistic elements recognized by the listening section are made available to the generation routines in a way that elicits responsiveness but not subordination from the artificial performer. (Rowe 1993, 80)

By the early 'eighties, with the availability of small, inexpensive, and sophisticated personal computers, a great number of musicians, composers, and programmers started to explore interactive computer music on their own without the support of large institutions. Inspired by the new potential of personal computers, and advancement of MIDI as a universal standard, Chadabe started a company known as Intelligent Music to provide an outlet for interactive composition softwares. Later David Zicarelli joined the company and the first software package of the company, *M and Jam Factory*, was released. Zicarelli states that his motivation for writing *Jam Factory* was his interest in creating a program that, "would listen to MIDI input and 'improvise' immediately at some level of proficiency", while allowing him (the performer) to improve the program's abilities (Zicarelli 1987).

By 1990, numerous highly programmable interactive 'MIDI systems' were developed that were successful in concert situations as well as recording studios, systems such as Tod Machover's *Hyperinstrument*, developed at MIT, or Robert Rowe's *Cypher*. Although other graphical interfaces such as Rowe's *Cypher*, Daniel Oppenheim's *Dmix*, and Karla Scaletti's *Kyma* allow composers to realize musical ideas quickly, despite a vast number of choices, the most widely used program of this nature is *Max* (Winkler 1998).

Beginning in 1986, *Max* was developed at IRCAM by Miller Puckette. As mentioned earlier, *Max* was developed primarily as a control software for the 4X synthesizer in order to overcome the difficulty of programming the 4X to perform demanding pieces such as *Jupiter* by Philippe Manoury, and *Aloni* by Thierry Lancino.[20] Yet the job shed light on other possibilities as Puckette recalls:

> Although the original idea was to build an oscillator bank, by 1985 most of the composers who tried to use the 4X were interested in "signal processing", using the term to mean transforming the sound of a live instrument in some way. This change of focus was the product both of

---

20. See discography: (Graugaard, et al. 1996), and (Lancino, et al. 1995).

> opportunity and of necessity: opportunity since “signal processing” is capable of a richer sonic result than pure synthesis, and since it is easier to create musical connections between a live player and electronics if the electronics are acting on the live sound itself. Necessity, since it was clear that after eight years of refining the digital oscillator, we lacked the software to specify interesting realtime timbral control at the level of detail needed. Signal processing, by contrast, can often yield interesting results from only a small number of control parameters. (Puckette 1991)

Soon after the development of 4X’s control surface, in response to composers’ demands for interactive real-time signal processing, the IRCAM developed the IRCAM Signal Processing Workstation (ISPW) to replace the 4X system. Although ISPW lifespan was short,[21] the primary software used for signal processing and synthesis on ISPW, the FTS (Faster than Sound), were later used to develop a system that runs on multiple hardware and software platforms. Finally, Miller Puckette’s latest programming software environment, *Pure Data* (Pd), provided the main features of *Max* and FTS together by taking advantage of faster processing speeds while adding more externals such as Graphics Environment for Multimedia (GEM) which delivers graphics functionality and video manipulation to the program:

> Pd is able to integrate audio synthesis and signal processing with video processing and 3-D graphics in a single real-time environment. The graphics program, GEM (Graphic Environment for Multimedia), was written by Mark Danks to operate within Pd environment. This holds great promise for composers and visual artists to explore an interactive and unified audiovisual medium. (Winkler 1998, 19)

21. ISPW depends on NeXT systems. A few years after the ISPW was completed, NeXT stopped making computers.

1.5 - Mobile phone as a Digital Music Instruments (DMI)

One of the breakthroughs of the past decade was the advent and popularization of smartphones. With the constant increase of computational and networking power, as well as the fast growing marketplace, mobile phones recently became a central device among DMI designers, especially those who mainly wish to grant non-musicians access to the creative process. According to Bowen (2013), the flexibility of configurations on smartphones permits composition and performance for the masses, but the lack of standardization has been prohibitive in defining the instrument in a cultural sense. Bowen argues that the answer to questions such as, “What does it look like to play a mobile phone? How should it be held? How does one become proficient at mobile phone performance? Can mobile phones be played as a group?” largely depend on how the software is configured, but still admits that, “there is a corollary to public perception as well” (ibid, 1).

The powerful suit of onboard sensors in smartphones allow a physical experience with music making in ways that other consumer electronics cannot, but the arbitrary relationship between the mechanism for producing sound, and the human body in digital mobile instrument design can somehow become a curse for establishing mobile phones as an acceptable DMI. Since music-making by physical gesture proved to be something that both musicians, dancers, choreographers, and the audience enjoy doing and watching, many within the field of electronic music have shifted focus to how to make computer instruments physically engaging. Many composers now view the computer as a viable real-time performance system which affects the audience’s perception of form, surprise, and mood by making use of the physical flexibility allowed through various configurations. Somehow, the lack of standardized performance practices creates several problems, as well as several opportunities for composers:

> On the one hand, the composer is free to design not only the instruments, but also the actions by which they are performed and the resulting sounds, and is therefore responsible for determining the interactions between conductor (if there is one), performers, and score. (Dahl and Wang 2010)

The plasticity of configurations of such instruments have a great impact on the composer, and the resulting composition. Arguably, such an arbitrary relationship of sound to creator, instead of freeing composers from instruments, made them become more involved in the design and implementation of instruments (Brown, Bischoff, and Perkis 1996).

According to Paul Lanksy (1990), “instrument design and construction now become a form of musical composition. The vision of the instrument-builder can be idiosyncratic, and even compositional. Playing someone else’s instruments becomes a form of playing someone else’s composition”. Later he concludes that:

> Musical systems now become ways to listen, perform and compose through the mind of another. Or perhaps of many others. In some ways an instrument builder becomes a subclass of composer. In other ways composer becomes a subclass of instrument builder. Whatever the formalization, however, it is clear that the number of ways in which the nodes are now capable of interacting has increased greatly. (Lanksy 1990)

According to Bowen (2013), ‘mobile phone music’ belongs to several categories of music, and since the year 2000, the mobile phone—and not some other device—

became an intrinsic element in a musical work which is in the middle of several overlapping circles (Fig. 1.1).

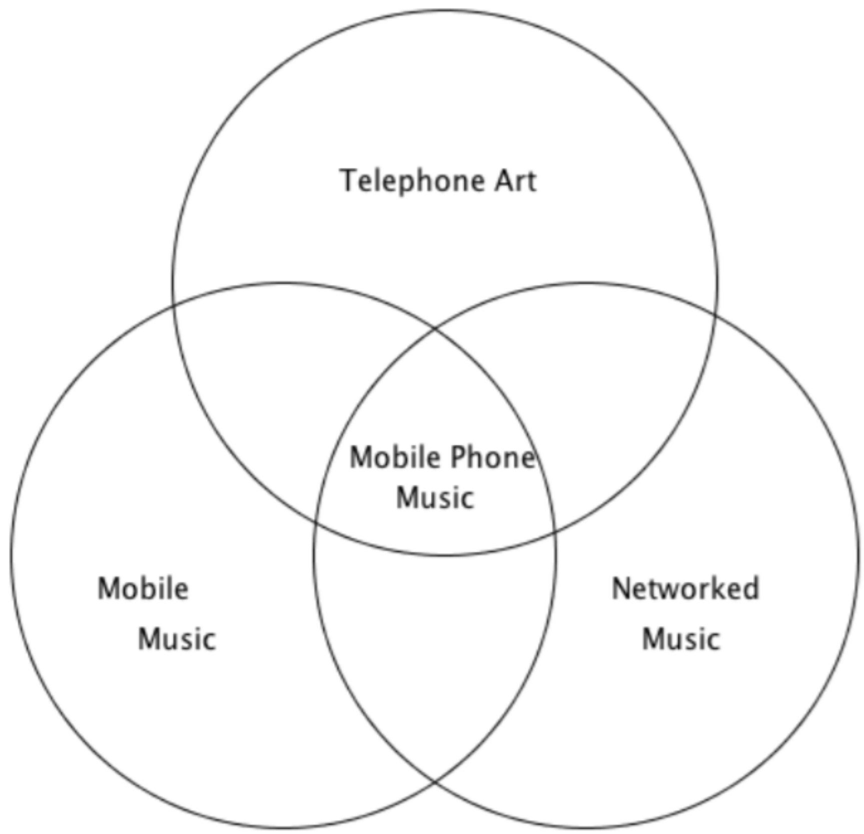


Fig. 1.1 Mobile phone music as part of three concurrent traditions (Bowen 2013, 80).

"Notation is a primitive guide to music, The unimaginative are slaves to it, others see behind it."[1]

Chapter 2: Notation of electroacoustic and computer music

There are moments in the history of music when particular innovations had far-reaching consequences. From the time Pythagoras translated musical intervals into mathematical equations,[2] arguably it was the modern staff notation that, along with the introduction of movable-type printing, had the greatest impact on evolution of music-making and its consequent academic progress in the past century (Manaris and Brown 2014). The first printed music notation appeared in 1473, and by 1600 music notation evolved into the form known today as common music notation. Although the practice of music printing and common music notation (CMN) did not face a significant transformation for a period of about 360 years (Roads 1995, 713), the *sound* of the twentieth century was no longer able to sustain itself in the form of traditional notation, since the musical imagination went far beyond the percepts of CMN, and unconventional notation became a necessity for composers of music concrète and electronic music. Soon the number of scores that used unconventional notation increased dramatically as most of the composers of the twentieth century attempted to devise a personal representation of their works. John Cage's book *Notations* (1969), depicts a collection of music manuscripts mostly written in the 1960's by two hundred and sixty-nine different composers, in which approximately one-third of the scores do not use conventional staved notation.

---

1. Norman Dello Joio (in Cage 1969, 235).

2. According to legends Pythagoras came with this idea as he was passing a blacksmith's shop, although the tale is empirically dubious, there is no doubt that the early acoustical experiments with monochord paved the way for future scholars such as Fārābi and Avicenna in the tenth century. See (al-Farabi's 2009), and (Avicenna 1958) for more details.

Until 1976, the Copyright Act only protected musical compositions that had been reduced to readable form. So each composer had to develop his or her own notation to reach a possible option for publication. Although this is no longer the case and the updated Copyright Act also protects any musical composition that have been recorded on tape or disk (Goldstein 2013, 237), there are many other motivations for notating a piece of music, even when the composer works individually and the performance consist of playing back the recorded audio — which means that there is no need for involvement of the performer's active interpretation, and the audio recording, while being a record of itself, arguably is the finished work, like a painting.

In the following chapter, some of the motivations behind the visual representation of electroacoustic music were examined by looking at some of the issues regarding the graphical representation of music to highlight the problems related to representation of electroacoustic and computer music.

2.1 - History and challenges of common music notation

Visual representation of sound was a historical necessity to associate pitch with the vocalized syllabus. It started with a simple ideogram and gradually became closer to what we call musical notation through the progressive use of the coordinates from left to right and bottom to top, respectively to indicate movement in time, and changes in pitch (Boulez 1960, 84-89). The new 'proportional' system replaced the old neumatic notation, with its ability to generalize a coherent formal system by which durations could be indicated. It could fully represent everything in the old system with its 'particularizing' features while replacing the mass of shifting and ambiguous symbols of neumatic notation with a body of more restricted, more ‘abstract’ symbols by means of a proper reductive process. Pierre Boulez argues that the logical evolution of any language must take ideas that are more ‘general’ and ‘abstract’ at every stage and replace those of the foregoing period:

> Thus the logical evolution of music appears as a series of ‘reductions’, the different basic systems forming a decreasing succession in which each one is slotted into the one that precedes it. (Boulez 1960, 84)

From the seventeenth century onward, notation and performance instructions became more detailed and specific, while the increasing definitions of new symbols caused more restrictions from the side of the composer towards the performer. In the romantic era (1800-1850) scores were still fairly loose and the performer could *interpret* the composer’s message, but by the twentieth century precision in coding of sonic phenomena became so important that to control the musical continuum, composers implemented some sort of comment to every note. Yet, the most complex examples of rigid notation do not necessarily result in more precise performances, mainly because the musical notation systems such as CMN are ambiguous in nature and inherently subjective (Fig. 2.1).

> Notation’s ambiguities are its saving grace. Fundamentally, notation is a serviceable device for coping with imponderables. Precision is never of the essence in creative work. Subliminal man (the real creative boss) gets along famously with material of such low definition that any self-respecting computer would have to reject it as un-programmable. (Roberto Gerhard, in Cage 1969, 240)

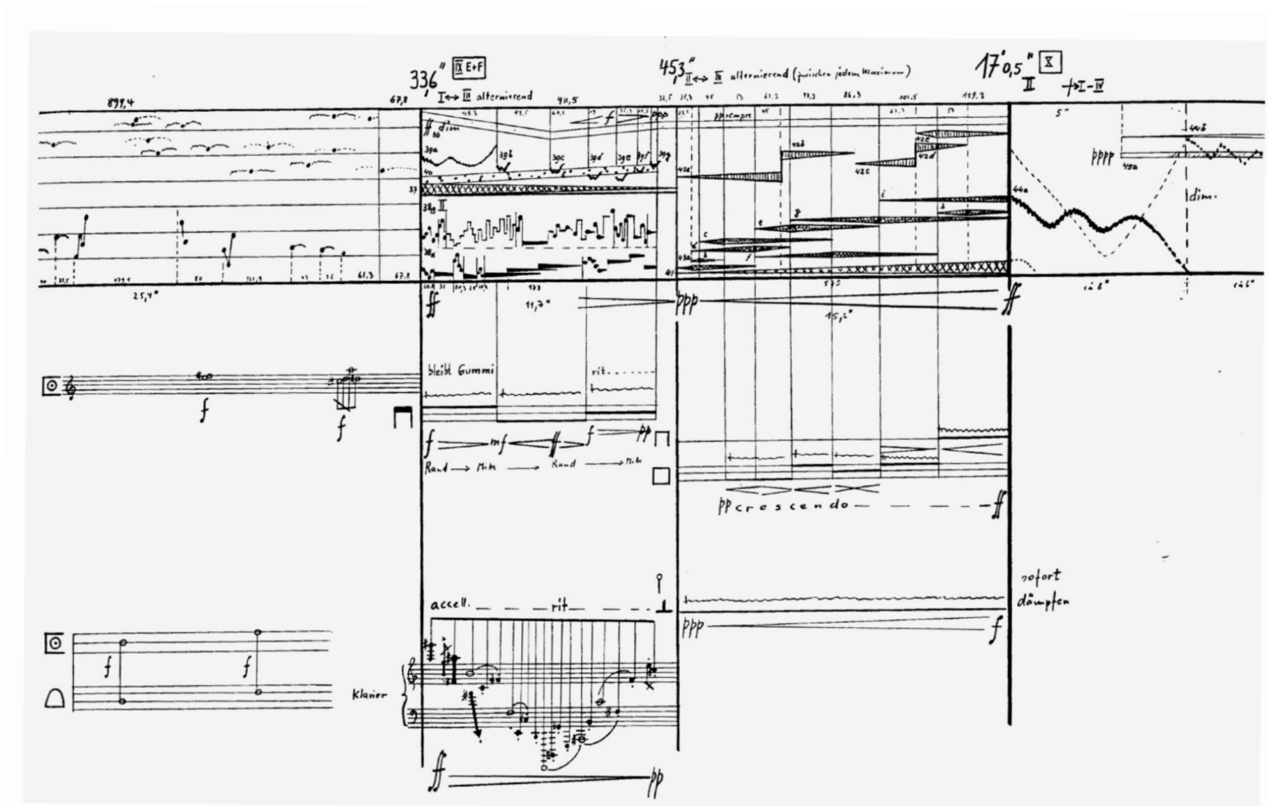


Fig. 2.1 Example of an early unconventional music notation. Score excerpt from Karlheinz Stockhausen's *Kontakte* (1968).[3]

Consequently, many composers bypassed the plain CMN for different reasons, for instance, pitch in CMN is limited to merely equal tempered pitches, and time durations to fractional durations of a single geometric series (1/4, 1/8, 1/16, etc.).[4] Boulez goes beyond possible time-related flaws entirely by saying that "any exclusively use of a notation entirely dependent on paper surface seems to show an ignorance of the true notion of musical time" (Boulez 1960, 86).

---

3. See discography: (Stockhausen 1992).

4. Other problems of CMN are difficulties in timbre and spatial representation, lack of multi-level representation of musical form, and mutating multi-event sound complexes possible with computer music (Roads 1995, 726).

2.2 - Graphical representation of music

By the second half of the twentieth century experiments with computer-generated sound proved that individual sound parameters of a sonic event, such as pitch, duration, spectral content, amplitude envelope, and spatial position can be defined with ultra fine resolution. But as Giancarlo Sica points out, such extended possibilities of definition, control, and processing of a single event (in languages such as Music V) causes a major difficulty in representing the global composing process: “There is always the problem of global versus local representation: that is, if we want to examine the single event, we lose the global view” (Sica 1997, 398-399).

The readability problem of redundant alphanumerical data (known in Music V and its variants) is not a problem restricted to notation of computer music, and it is obvious that any new musical notation should be easily readable, both in local and global ranges. One of the approaches that offer the ultimate notational flexibility over the readability problem is using ‘graphic notation’ where composers can literally draw sonic events (Fig. 2.2). For example, in the most highly developed graphic synthesis system, Xenakis’s UPIC, waveforms and event envelopes can be drawn by the composer and later be organized at a higher level (Roads 1995: 330-332). Although UPIC is good at representing the music on multiple time scales, it fails on providing a complete solution to the problem of synthesis and signal processing (Sica 1997, 399).

Despite using graphical notation himself, Boulez insists that, “any new graphical and coherent notation should include our present symbols, the neumatic symbols, and the ideogram...” and puts forward three reasons for considering exclusively graphic notation as totally regressive (Boulez 1960, 86):

1. It does not use proportional symbols,
2. It appeals to less delicate brain structures (thus leading to rougher approximations),
3. It takes no account of any *overall* definition of musical time. (ibid.)

Boulez also argues that, "until such a system is invented, any graphical notation will only constitute a regression, at best a literal, graphic transcription of a situation that can be translated into symbols. It will be no more than a sheet of figured values, as it were a reproduction on paper squared in millimeters." (ibid., 85)

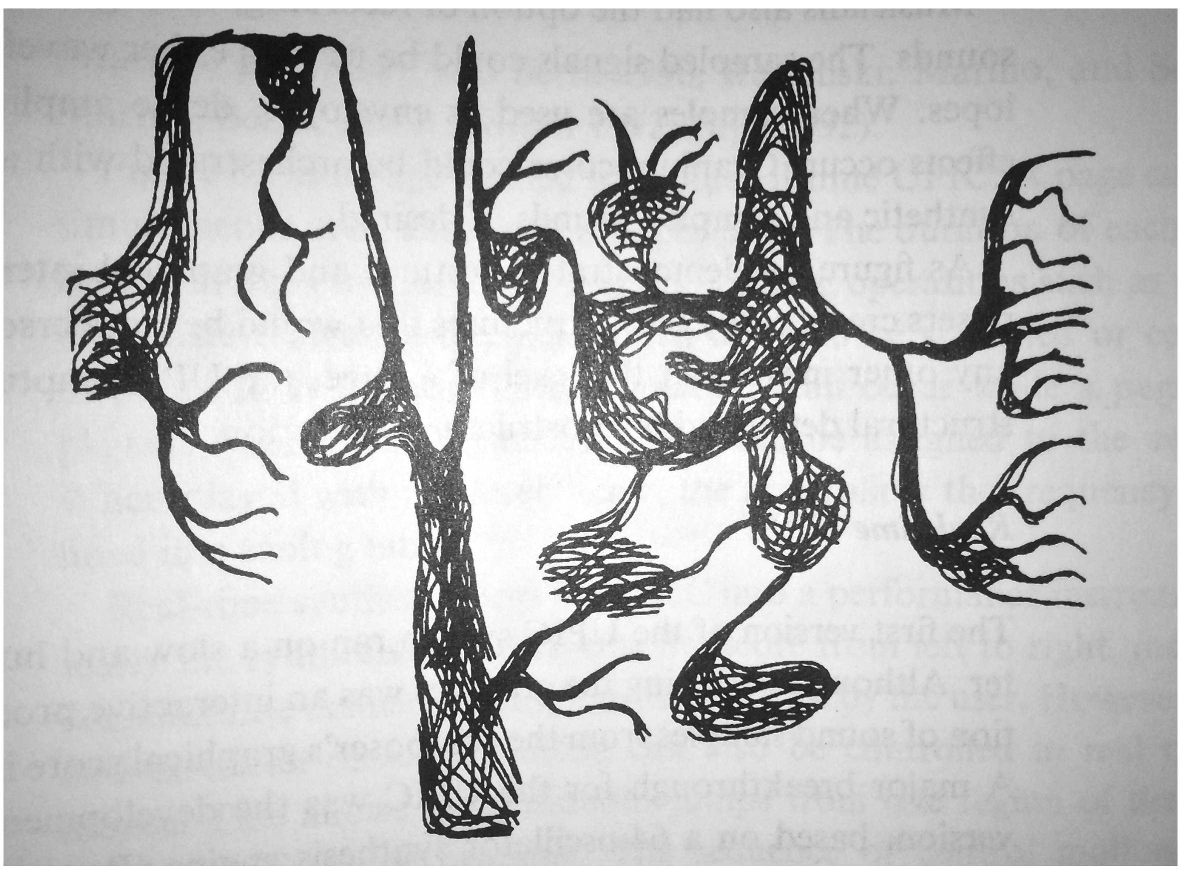


Fig. 2.2a Graphical score of Iannis Xenakis's *Mycenae Alpha* created using UPIC (Roads 1996, 331).[5]

5. See discography: (Xenakis 2001).

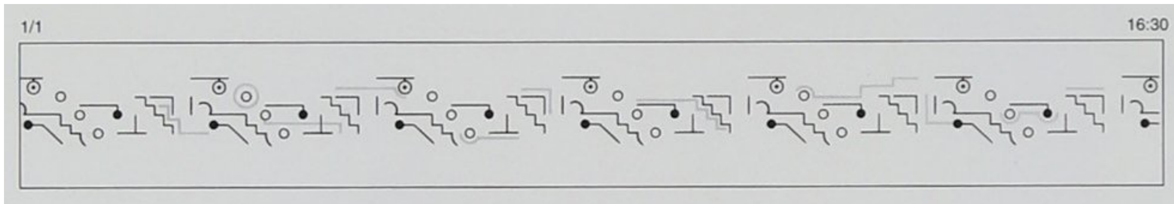


Fig. 2.2b Graphic score of *Ambient #1: Music for the airports* By Brian Eno. (Eno 1978).[6]

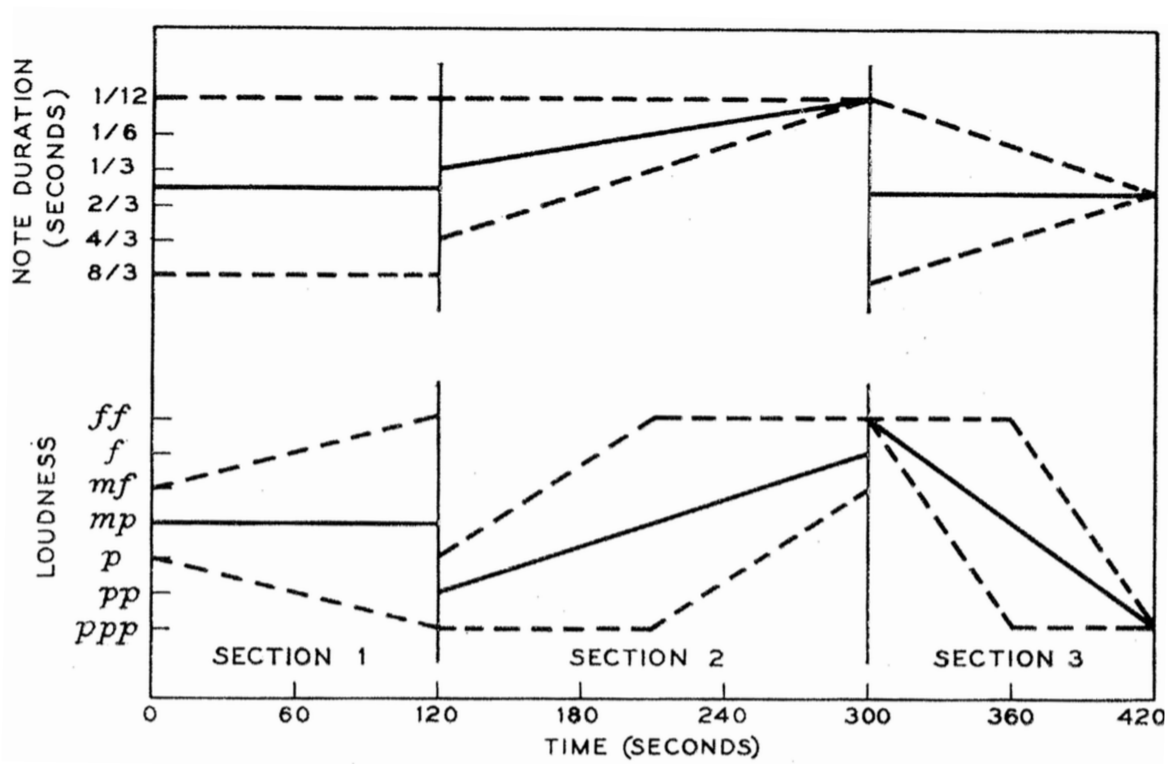


Fig. 2.2c Section of a score prepared for a study by J.C. Kenny. The average values for note-duration and loudness are shown by the solid lines as functions of playing time. The allowable range of variation of these parameters is shown by the dashed lines surrounding the solid lines (Mathews 1963).

As of now, graphical representation seems to be the only genuine solution for notation of electroacoustic music as many approaches were made to improve its deficiencies. Building on prior work, GNMISS (Graphic Networked Music Interactive

6. See discography: (Eno 1978).

Scoring System) for example was developed to address the scoring problem of 'comprovisation' of electroacoustic music through Internet2, using a 'directed dramaturgy' approach (Dudas 2010). Although using hand-drawn scores that could be reduced to a piece of paper were useful for earlier telematic works, Ian Whalley admits that there were limitations within aesthetic approach adopted.[7] Whalley argues that while problems of electroacoustic sound representation are not unique to networked sound art, problems of electroacoustic notation are partly because "electroacoustic music gestures can be independent of pitch yet also expand on traditional pitch-based gestures" (Whalley 2014). Another problem is that electroacoustic music has few common 'instruments' and not always sounds are associated with identified sources. Regarding the GNMISS, Whalley suggests that, beyond being "an artistic choice that the technology facilitates", an open system and common communication method (without any use of complex words or two-dimensional matrix models) may further involve audiences in the performer-to-audience experience as well as allowing alteration of details of a piece by participants and adjust to others, in real time in an ongoing dialogue: "...it may also show something of the dramatic structure of works in a way that could be quickly grasped, allowing further audiences engagement the narrative experience" (Whalley 2014).

As Whalley further describes, the GNMISS has four visual layers that illustrate the structure of work on a circle to efficiently represent the structural relationships between affective parts in a space (Fig. 2.3).

---

7. Telematic Art is a form of performance art where performers are in remote places communicating over telecommunication facilities.

One layer maps emotions to colours based on associated words as a primary basis for gesture and timbre representation. A second layer gives musical motives and frequency information for participants to follow. A third allows for more detailed indication of gesture and sound archetypes through representative symbols… Finally, an inside layer represents macro key centers. For timing, the circle score turns in clock time with the current playing position always at noon, and a central metronome shows speed independent of clock time. (Whalley 2014)

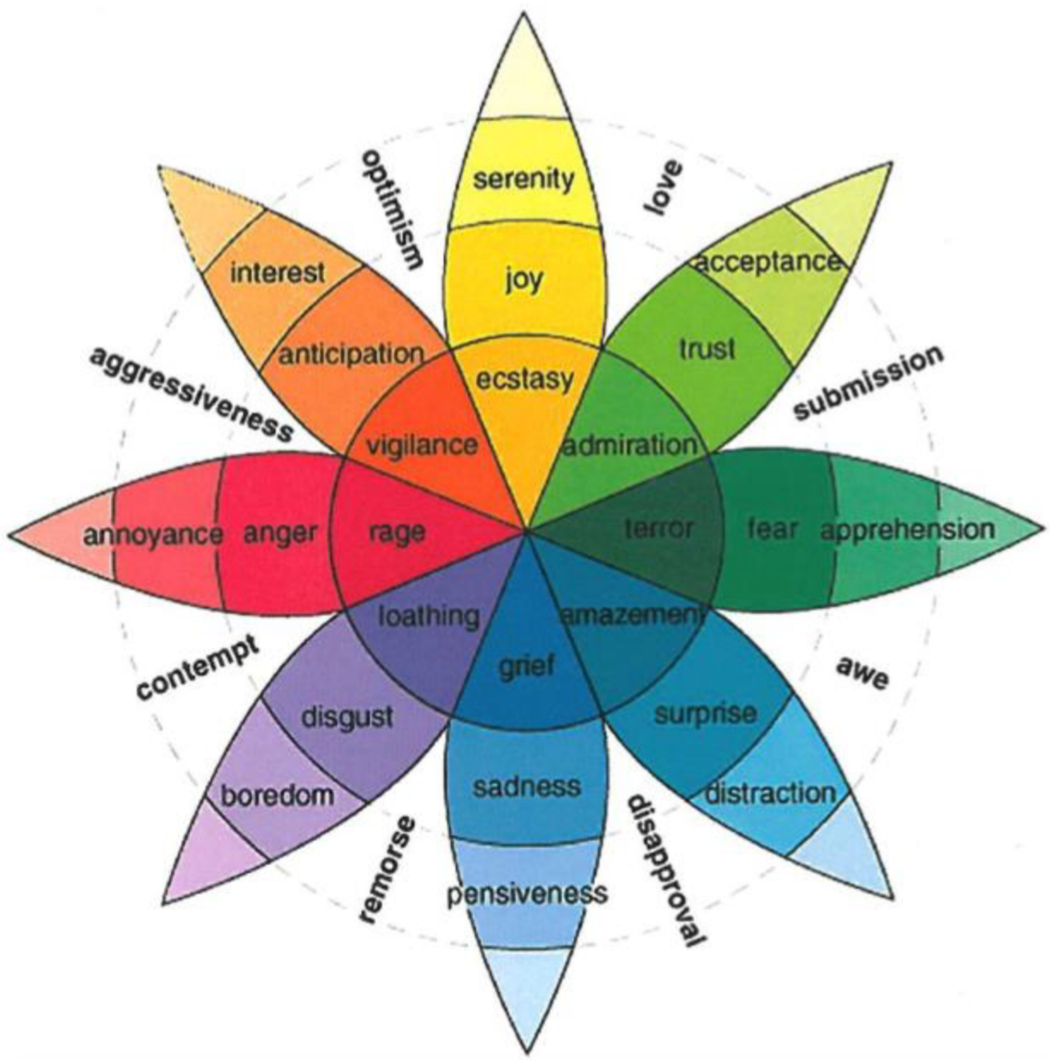


Fig. 2.3a GNMISS score excerpt from Ian Whalley's *Plutchik* (2001).

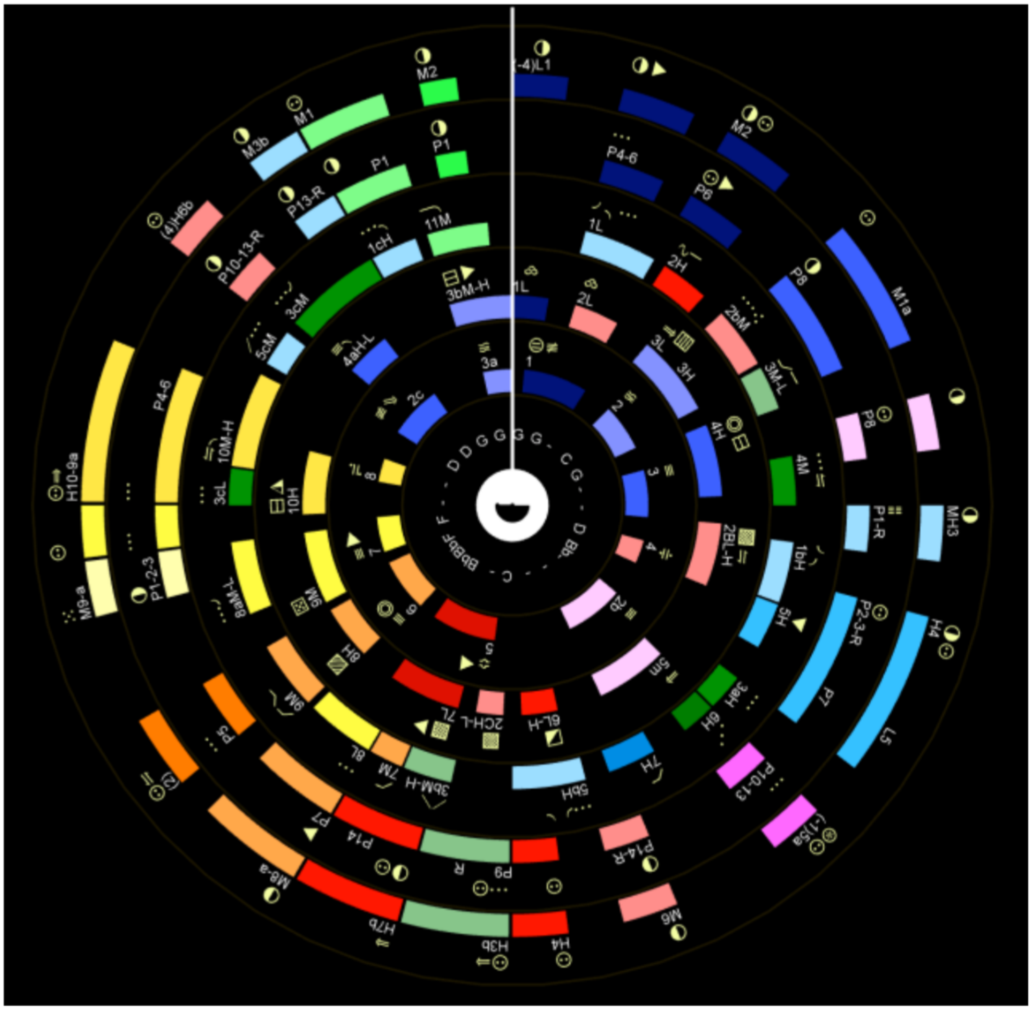


Fig 2.3b GNMISS score excerpts from Ian Whalley's *Sensai na Chikai* (2012).

While GNMISS was initially set out to address practical timing problems in coordinating network-distributed participants, it implemented an affective way to approach timbre and gesture. It was developed to reduce complex ideas to a simple representation in a practical way that offer a good deal of flexibility in producing very complex outcome by people and machine players equally. Beside being approachable to players perhaps the usefulness of the system is in its ability to enhance the communication of electroacoustic telematic works to audience as well as performers through the public projection of the scores during performance (ibid.).

2.3 - Reasons for visual representation of music

In the age of computers, notation in the digital domain is blurred more than ever. There is a seamless move between visual and sonic information, and there is no difference between sound representation and realization. While in Xenakis's UPIC the visuals are produced by the composer, on another occasion we may see an 'aural score' intended to represent a visible 'reading' of the sounds of a piece. For instance, the graphic score of György Ligeti's *Artikulation* was created by Rainer Wehinger after the recording of the piece. In this case the score is no longer a set of instructions for performance, but a visual aid for listening to the piece (Fig 2.4).[8]

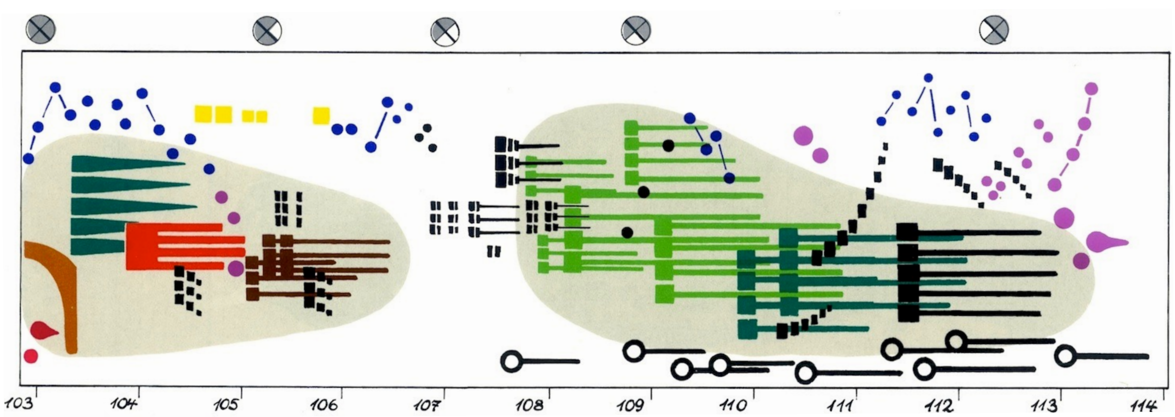


Fig 2.4 Graphic score for György Ligeti's *Artikulation* (Wehinger and Ligeti 1970).

According to Pierre Couprie multimedia has gradually asserted its presence within the research domain in the recent years: computer softwares have been developed which are now being used by researchers and now music analysis can be found as a simultaneous combination of sound, graphics, and text, etc. As a consequence, there are no longer two distinct disciplines of electroacoustic music just

8. See discography: (Ligeti 1984).

as analysis, and representation. Couprie argues that graphical representation seems to "constitute a real tool for analysis and for publication of electroacoustic music: henceforth analysis and representation will be inseparable. Such an attitude will no doubt be very beneficial for music research" (Couprie 2004). He also suggests that, "representation could be a great pedagogical tool, not only to guide the listening of neophytes but also as an analytical tool, particularly appropriate to electroacoustic music" (ibid.).

According to logician, mathematician, philosopher, and scientist, Charles Sanders Peirce (1839-1914), in the 'language of signs' iconic and symbolic functions have different interpretive qualities (Peirce 1955, 98-119). Couprie in a similar fashion to semiotician Claudine Tiercelin (2002) describes these two functions by illustrating iconic and symbolic functions of graphical symbols in music scores:

> This means that our graphical objects will have iconic functions (Fig 2.5a) — the link between graphical qualities and the sound criteria they represent are relatively intuitive: for example the shape of the dynamical envelope — or symbolic functions (Fig. 2.5b) — the graphical form will be the result of a very precise coding of sound criteria. (Couprie 2004)

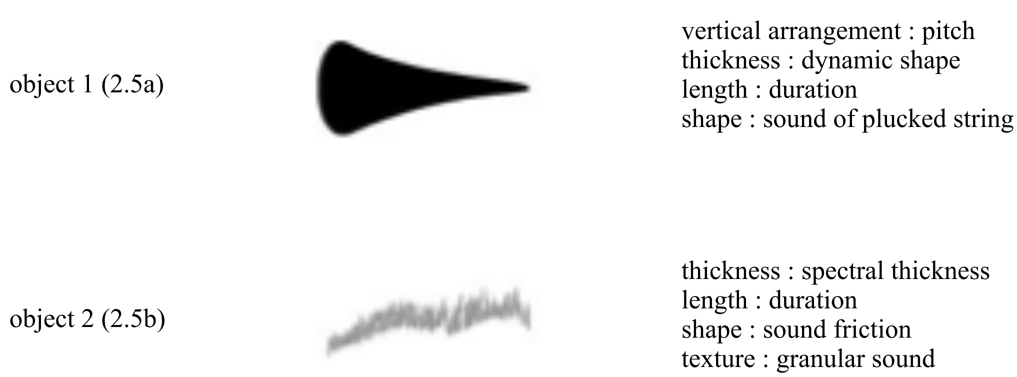


Fig. 2.5 Iconic and Symbolic representation (Couprie 2004).

While iconic representation is much more intuitive and more accessible to a larger potential public, symbolic representation demands familiarity with the legend and is often complicated to read. Yet with symbolic representation it is possible to achieve great precision and great analytical complexity while remaining as clear as possible. As a result, despite being complex to decode, researchers and specialist may prefer symbolic representation because it represents undeniable advantages in the transmission of analysis with its didactic and aesthetic qualities (Fig. 2.6).

Fig. 2.6 Relationship between image and sound in two of the graphical objects extracted from Ivo Malec's *Reflet* (Couprie 2004).

Couprie then concludes that graphical representation could prove to be an analytical tool, as well as an analytical method, plus it can be used to assist 'listening education' since creating visuals provokes an enrichment of listening (ibid.).

Meanwhile, in his book *Principles & Practice of Electronic Music* (1973), Gillbert Trythall suggests two simple reasons for having a score for electroacoustic music, "to make a record of the working process, and to coordinate electronic music with live performers or other media" (Trythall 1973, 164). According to Trythall most of the scores are plots of sounds and sound transformations against time, he then outlines some of the most important elements that can be shown in music notation (see appendix B).

2.4 - 'Code' as a musical score

Trythall argument that musical scores can be used as "a record of the working process" in music production has a similar counterpart in the field of computer science as an essential part of any code is being used to embed programmer-readable annotations inside a computer program by adding 'comments' to the source code. By having descriptive comments, the code itself can become a music notation which also performs and sketches out the piece.

In computer generated algorithmic compositions, static text results in continuous musical events that evolve in time. Yet unlike traditional music notation, code does not naturally come with a representational timeline and according to Thor Magnusson, it is practically impossible to create graphic notation of code's functionality as it will be executed in time (Magnusson 2014). Meanwhile, he argues that since musical notation typically followed the tradition of defining events on a linear timeline, the language of music scores are mostly event-based as opposed to time-based:

> However, it should be noted that chronographic timelines of any kind, such as historical events where time is mapped directly to space, were uncommon in medieval times when the foundation of our current musical notation was established, and only begin to appear in the mid-eighteen century. (ibid.)

In a computer code, instructions can be scheduled by the program itself as timed events, triggered by the incoming data from network of other computers or through human interaction. At the same time, the possibility of adding code into timelines of compositional softwares especially Digital Audio Workstations (DAW) is becoming more common (i.e. Scripter functionality in Logic Pro X, or Max for Live within Ableton Live) which offer strong narrative dimension to the score and music alike.

As mentioned earlier, musical notation is mainly a system of abstractions, affording certain types of expression but excluding others. Any system of abstraction decides upon a perspective of what is important to express and decides what to include or exclude (Bowker and Star 2000; Seeger 1958). For practical reasons coding programs such as Processing, SuperCollider, Max/MSP, and Pure Data (Pd) exclude timelines from their design, but the desire for representation of linear time is so strong that third-party add-ons have been created to reestablish timelines in many of them. Arguably, representation of the timeline may be an important element in musical composition, as the compositional thought process can be simplified by arranging code events on a timeline by using graphical 'code scores'. In this context, implementing textual and graphical representations in a score is not only useful both to the performer and the audience, but allows for more abstract compositional arrangements; higher levels of dynamic scoring, and better support for real-time composition.

Originally conceived of as "a musical piece with an embedded micro-language for live coding performance of microtonal drone music", *Threnoscope*, composed by Thor Magnusson, is an example of a system that implements both representational notation and a prescriptive code score (Fig. 2.7). The primary mode for performance in *Threnoscope* is by means of live coding in a micro-language created on top of the SuperCollider, where musical events are created, shaped and terminated through the textual interface of the code (Magnusson 2014).

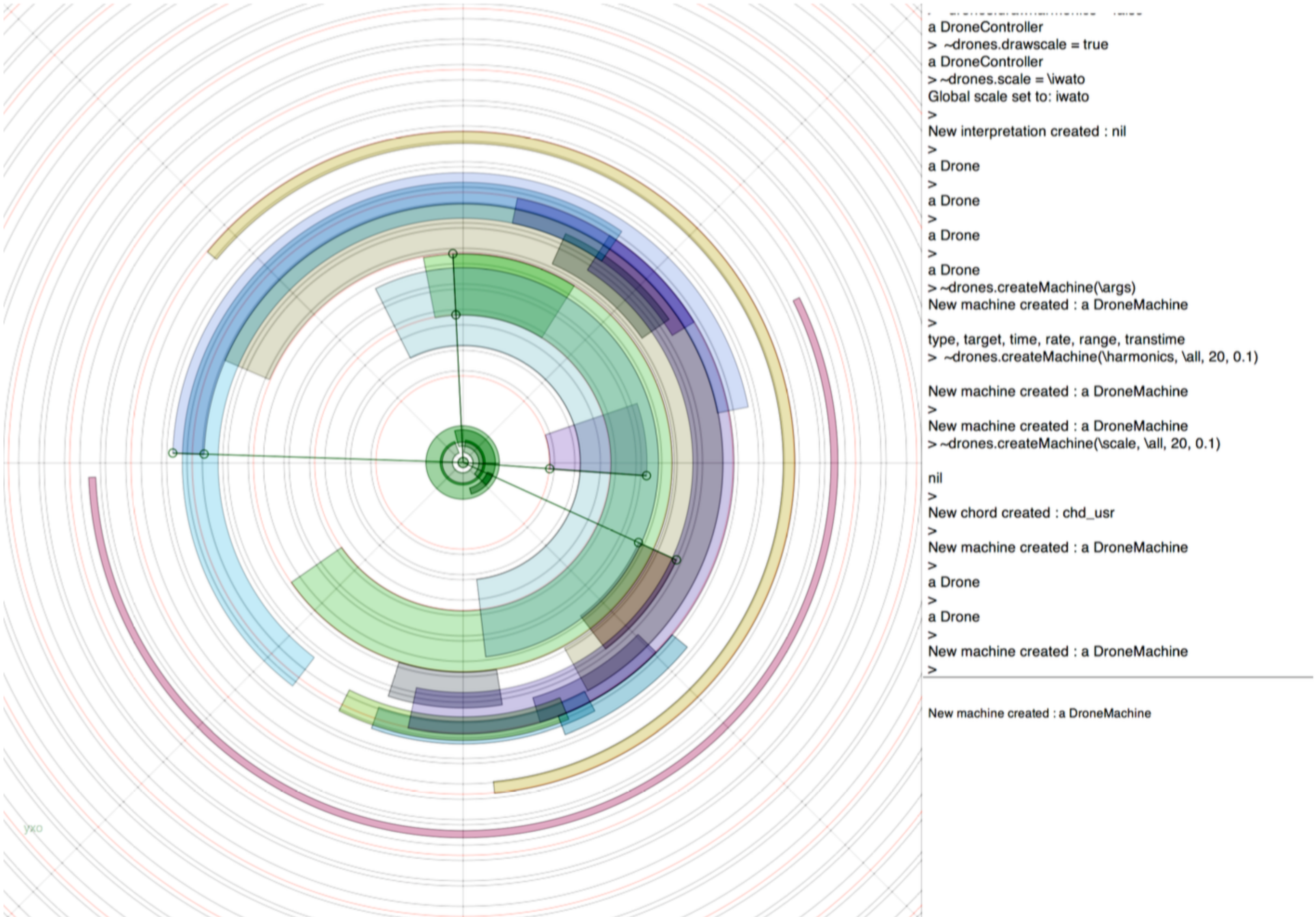


Fig. 2.8 Score excerpt of Thor Magnusson's *Threnoscope.*

2.5 - Using graphical scores in Pure data

According to Miller Puckette, the original idea in developing Pure Data (Pd) was "to include also a facility for making computer music scores with user-specifiable graphical representation … Pd is designed to offer an extremely unstructured environment for describing data structures and their graphical appearance. The underlying idea is to allow the user to display any kind of data he or she wants to, associating it in any way with the display" (Puckette 2002). Data structures in Pd are built from four different data types: scalar floats, symbols, arrays, and lists. Pd also introduces a graphical data structure with a facility to attach shapes and colours to the data while the user can visualize and/or edit the data in real-time (see appendix A).

Although Pd is one of the most popular applications, so far the possibility to create user defined graphical data structures were not widely used by media artists, and as a result there are still problems that can only be found and fixed if more artists and musicians in the Pd Community actually start to discover these possibilities. In his article "128 Is Not Enough - Data Structures in Pure Data", Frank Barknecht presents some examples of how data structures can be used in Pd. According to Baknecht, "Pd's data structures most naturally fit the needs of preparing and playing graphical scores" (Bacnecht 2006). In his composition *Solitude*, composer and Pd developer, Hans Cristoph Steiner, used Pd's graphical data structures to control every aspect of the sounds that is being generated in realtime by Pd (Fig. 2.9).[9] Steiner tries to combine minimalist textures with rapid transformation of samples of Duke Ellington's "Solitude" by using data structures to edit, display, and sequence the score at the same time. In

---

9. See discography: (Steiner 2005).

the score, time flows from left to right, each colour represents a sample, and each sample is controlled by two arrays which controls sample playback, amplitude and panning (Steiner 2010).

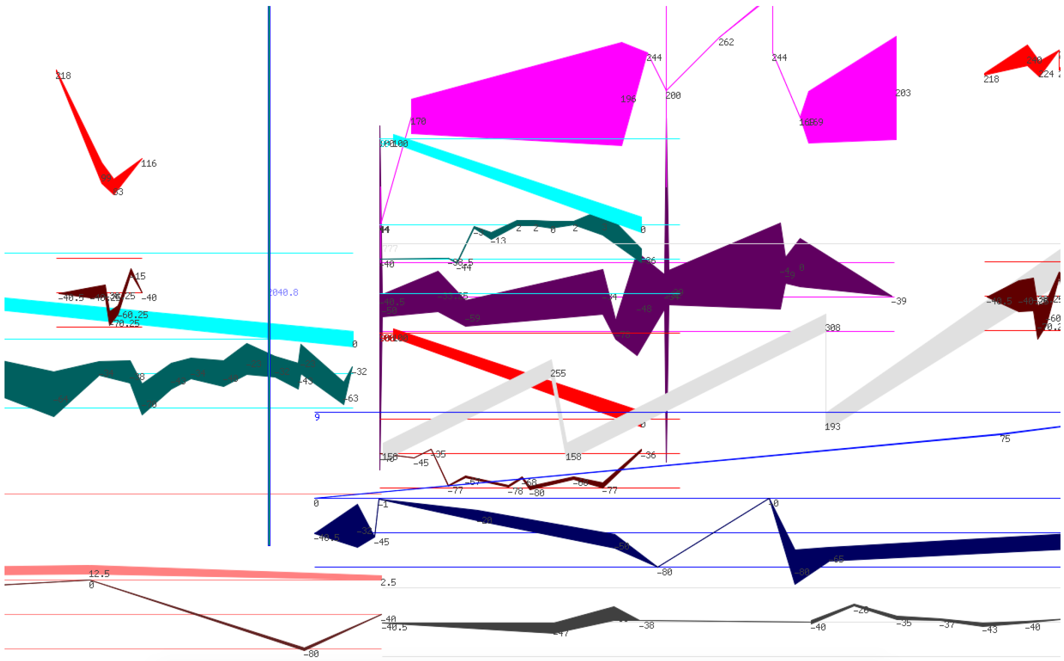


Fig. 2.9 Example of graphical score in Pd Score excerpt of Hans Cristoph Steiner's *Solitude* (Steiner 2010).

Furthermore, Frank Barknecht presents more examples in his paper which shows that Pd's data structures can also be used to interact with a score in realtime,[10] or to implement costume GUI (Graphic User Interfaces) elements,[11] visualize abstract concepts,[12] or even create graphical illustrations.

---

10. Frank Barknecht's *Pong* uses the classic video game "Pong" to manipulate data structures in real-time.

11. Chris McCormick's GUI objects for Pd.

12. Orm Finnendahl created interactive Pd patches to explain sampling theorem, granular synthesis, etc.

Chapter 3: Methodology

This chapter deals with the method by which the interactive music for this thesis was worked out. As we saw in chapter 1.5, mobile smart devices are powerful tools for bringing interactive art to the public. Since smartphones and tablet devices are widely popular nowadays, each year a large number of applications are being developed that turn these mini-computers into a complete musical instrument.

As mentioned earlier, in order to synthesize sounds with computers we need a very fast program and a powerful simple language (Mathews 1970). The most widely used Digital Signal Processing (DSP) musical programming software is certainly Barry Vercoe's text-based Csound (Boulanger 2000) which is very well adapted for batch processing, and it handles polyphony nicely. But since Csound belongs to the so-called Music N languages, it is not very well developed for real-time control structures (Mathews 1970). Another open-source text-based alternative is James MacCarthney's SuperCollider, which is explicitly designed for real-time use. On the other hand, block diagram compilers with graphical interfaces are by far the most popular sound synthesis programs. Being also the first graphical compiler, among all GUI-based (Graphical User Interface) sound synthesis programs, Max/MSP, according to both beginners and system managers running multi-user, multi-purpose computer labs, is better supported and documented than Pure Data (Pd). Cycling74's Max/MSP is the commercial sibling of Pd that can transform a computer into a complete musical instrument compatible of live performance. According to Miller Puckette, Max and Pd allow almost anyone to synthesize 'uninteresting' timbres instantly but, "making interesting timbres is much more difficult and requires much additional knowledge" (Puckette 2007). Max and Pd are not truly compatible, but it is possible to take knowledge of Pd and apply it in Max/MSP and vice versa, and even to port patches from one to the other.

Since its appearance in 1996, Pd has been popular in computer music circles. Beside being a great audio engine, its instantaneous interactive nature makes it an excellent tool for prototyping interactive digital instruments, also it is a great audio engine for games. According to Peter Brinkmann (2012), as processors become more powerful, game developers have a choice of using sophisticated audio processing algorithms to synthesize music and sound effects instead of using canned samples. App developers have been connecting games to Pd by using the network capabilities of Pd in the past, but the use of networked objects in Pd introduces some friction into the patching process. To achieve a complete separation of concerns between audio development and app development, Pd can be implemented in the source code by using 'libpd'*.* With its permissive BSD licence,[1] libpd can be added to almost any project: "you can build libpd as soon as you have a C compiler. Ready-made bindings for Java and Objective-C as well as support for Android and iOS help you get mobile apps off the ground in a hurry" (Brinkmann 2012). Libpd embeds Pd as a DSP library that takes input samples and computes output samples. With libpd sound designers can prototype audio components in Pd and when the patch is ready the developer simply adds the Pd patcher to the app resource library.

Libpd is already popular among people who work on mobile applications and recently many applications took advantage of libpd's free open source licence. As a

---

1. BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the redistribution of covered software. The original license were used for Berkley Software Distribution (BSD), a unix-like operating system.

result, Pd can run on a wide variety of devices from phones to computers. Libpd was developed by a community of “impassioned musicians and developers” who according to Peter Kirn (2012) are working as volunteers out of love for what the tool can do. Martin Brinkmann, the core developer of libpd also has a book that walks the reader through implementing libpd in the source code of mobile applications. Also with the emergence of third party applications such as *MobMuPlat*, useful tools for prototyping sound engines are easily available to sound designers. *MobMuPlat*, created by Daniel Iglesia, is a mobile application that allows sound designers prototype interactive graphical user interfaces for libpd by transferring data from the hardware of a mobile device to libpd; data such as tilt, compass, camera, flash, vibration, etc.

3.1 - Quick introduction to Pure Data (Pd)

Music pieces that accompany this thesis were coded in Pure Data (Pd), so in order to fully understand them, one has to learn how Pd works. There are a lot of places to start learning Pd. For example, FLOSS Manuals is an online collection of manuals about free and open source softwares including Pd, or Pure Data’s Community Website among other resources. Miller Puckette’s book, *The Theory and Technique of Electronic Music* which uses Pd’s documentation examples, is another great resource for learning Pd’s high-end capabilities (see appendix A). Another valuable resource for musicians and sound artists is Andy Farnell’s *Designing Sound* (2010) which uses Pd as the only programming tool for realizing the sound design examples in the book.

3.2 - Walk-through of *Riley in Sea*'s code

Written for the York's University's Music department building, Accolade East (ACE), as part of the accompanying compositions for this thesis, *Riley in Sea* is an interactive mobile platform inspired by Terry Riley's *In C* which aims to recreate the piece in a social-interactive context.

In C consists of 53 melodic patterns that have to be played in sequence by a number of performers with different instruments, desirably a group of about 35 according to Riley's performance note. Each performer has the freedom to determine how many times each pattern has to be repeated before moving on to the next. In addition to 'suggestions' on how to play dynamically and trying to diminuendo and crescendo together, Riley underlines that, "it is very important that performers listen very carefully to one another and this means to occasionally drop out and listen." (Riley 1964) The obvious characteristic of *In C* that makes it suitable for our purpose (social interaction) is already implemented in the original composition: "One of the joys of *In C* is the interaction of the players in polyrhythmic combination that spontaneously arise between patterns." (ibid.)

To begin the coding of the piece in Pd, first we need to define the score by using two different tables: one for pitches, and one for rhythmic durations (Fig. 3.1). We can use durations table to sequence notes of each pattern without having a constant beat. By doing so we can also bypass the limitations of plain CMN, in other words, any real number can be assigned to notes' duration; not just fractional durations of the traditional geometric series (1/2, 1/4, …). In order to do this, notes have to be sequenced by triggering each note when one that is already being played has finished its duration.

Although for this specific example we only need beat ratios, and we can define all of the notes' durations in factors of 32nd, the shortest duration that exists in the score. It means that each 32nd note can be defined as 1, 16th notes as 2, and so on. By multiplying these durations with a variable number the user can change the tempo of the piece easily. With this method it is possible to detour the mechanical treatment of common sequencers which mostly are strictly beat-based. To execute rests we assign a constant number (i.e. 1000) to pitch and later create a portal that only lets the acceptable midi note values between 0 and 127 pass through, that means whenever the output is 1000 (rest), the 'makenote' object waits for the duration of the rest and then triggers the next note (Fig. 3.2).

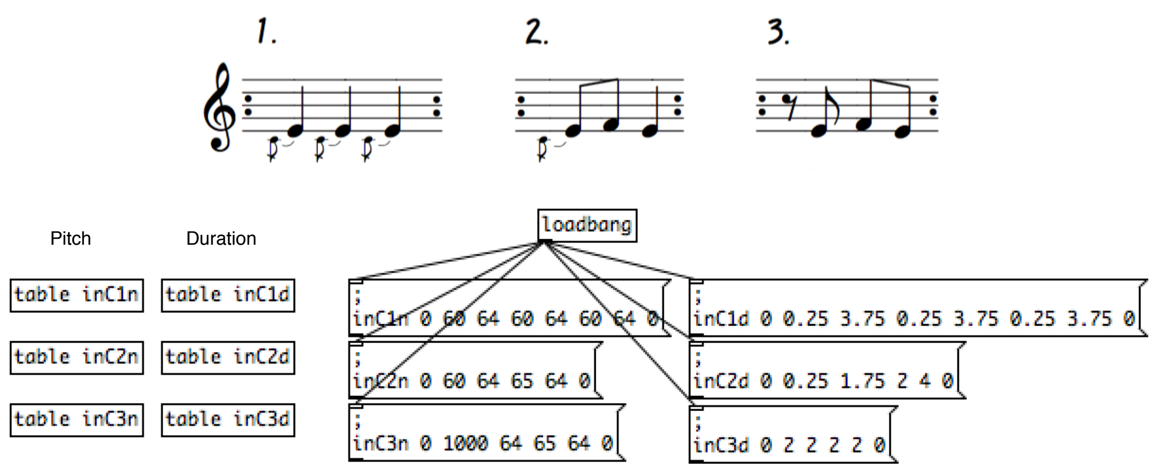


Fig. 3.1 Defining the score in Pd.

To go forward in the score, a zero is added to the end of each table. These zeros will be used to determine whether to repeat the current pattern, or go forward in the score, according to the performer's decision. For now we set a random velocity (loudness) to each note then later we will add a receiver from GUI so that performers can control the dynamics from the device (Fig. 3.3).

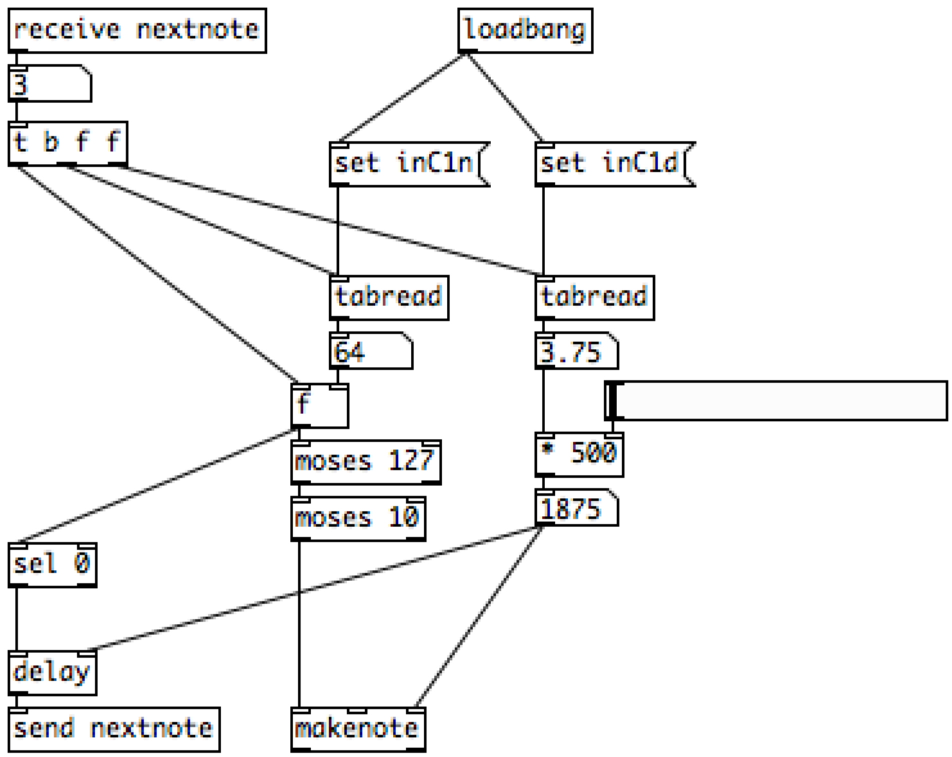


Fig. 3.2 Sequencer created for *Riley In Sea.*

Now that we have the core sequencer, we can choose which parameters performers can actually interact with. In this example performers can control the amplitude of each note by tilting the device, as well as timbre, and tempo within the GUI. Any equation, such as a Gaussian equation, can be used to combine parameters of tilt (x,y) to reach a complex gestural control in order to smooth the output and have a precise control of the dynamics (Fig. 3.4). We can also use the 'shake' signal to trigger next pattern, it means that whenever the user shakes the device, Pd automatically goes forward in the score (Fig. 3.5).

In order to let performers easily learn the gestural control of the dynamics, an ‘array’ was added to the GUI to serve as a real-time visual representation of the dynamics (Fig. 3.6).

Later we can add more elements to the GUI to control other parameters, such as sliders that control sound synthesis parameters (so that each performer can set and control the timbre during the performance). We can also add more gestural control to other compositional parameters, for example we can control the tempo with gestural data of accelerometer (which means that the tempo can be adjusted by moving the device).

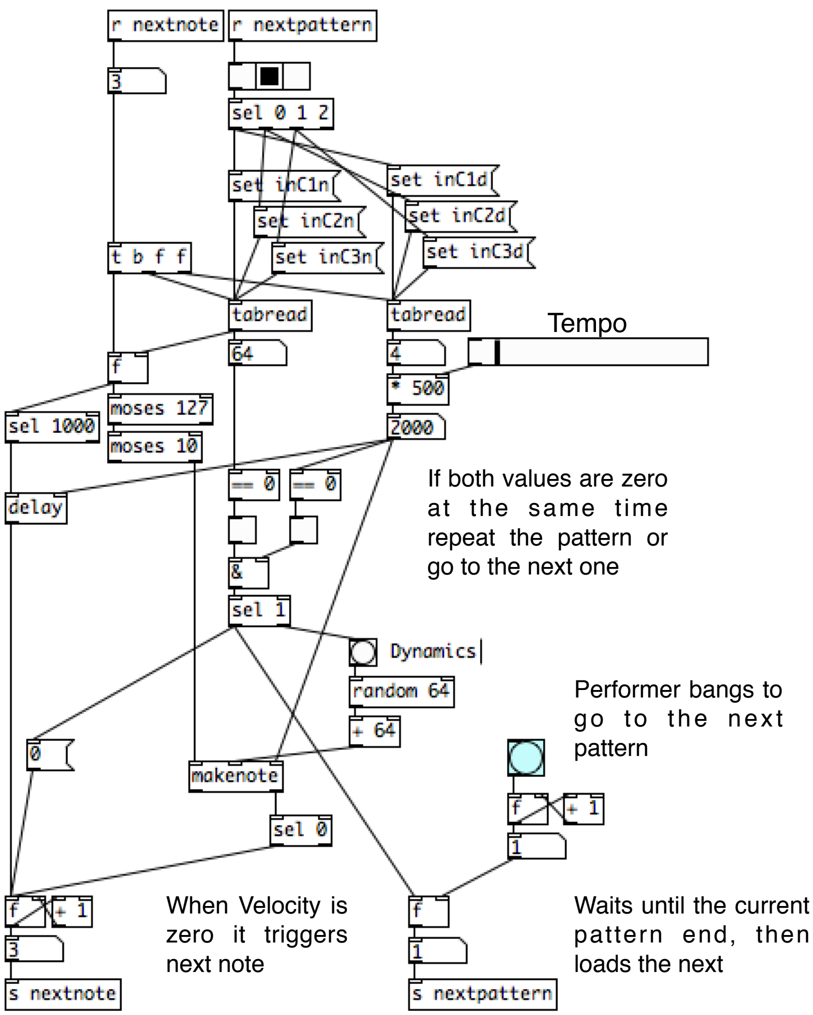


Fig. 3.3 Sequencing patterns and notes.

$$y=\frac{10}{\sqrt{2\pi\sigma}}e^{-\frac{x^2}{2\sigma^2}},\ \sigma=\{1,1.2,\ldots 3\}$$

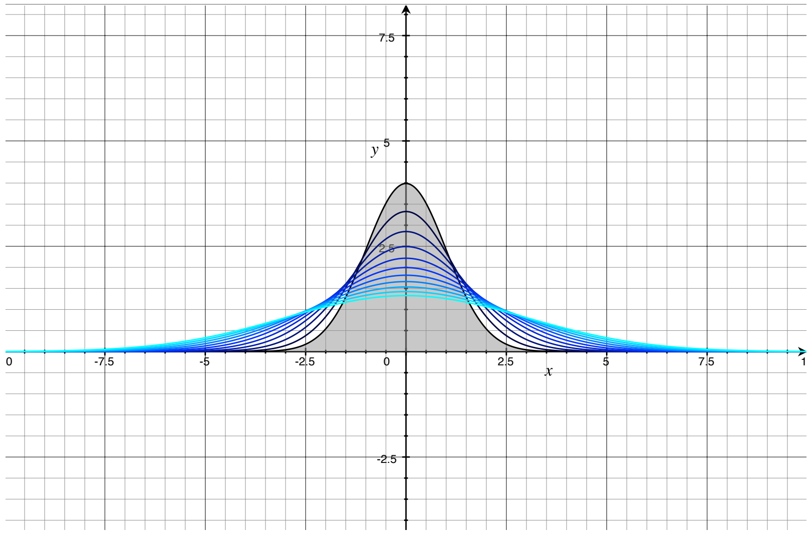


Fig. 3.4 Guassian equation.

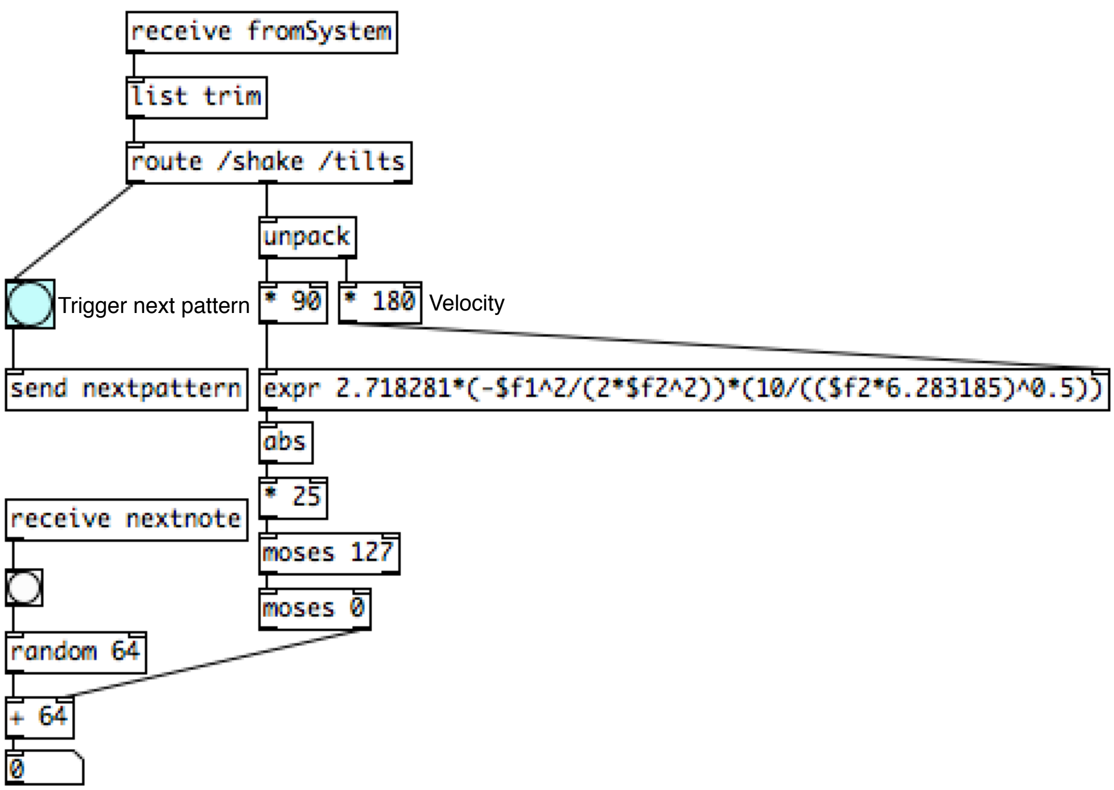


Fig. 3.5 Performance control for going forward in the score and controlling dynamics.

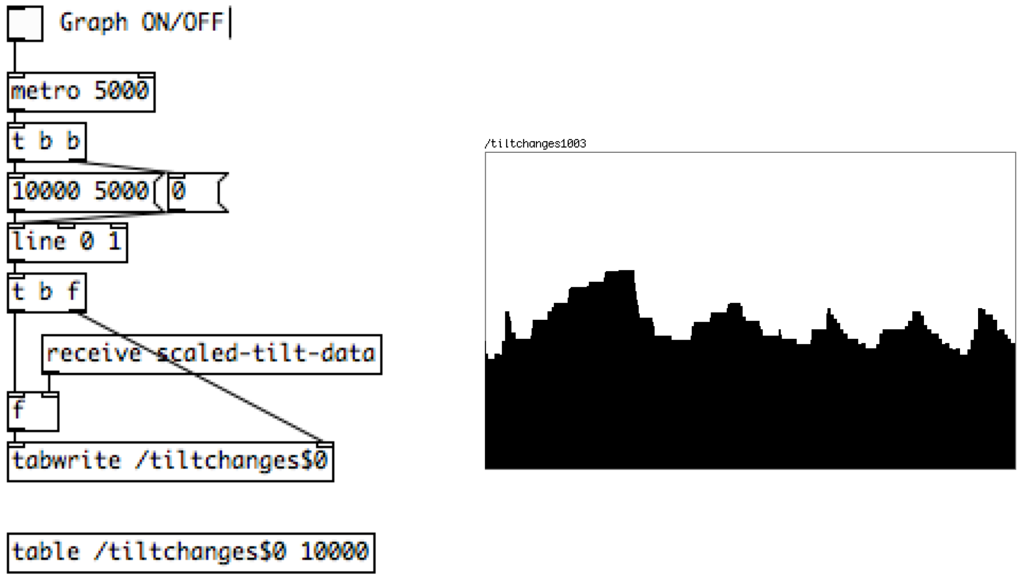


Fig. 3.6 Dynamic's real-time visuals.

At this point we need to design DSP units and send the output of 'makenote' object to produce the sound. Pd offers an infinite number of choices for sound synthesis, but in order to reduce the amount of processing (as the sound of each 'performer' has to be processed and added up to the sound of others), only two DSP units were used in this case; one for the user, and one for the rest of the network.[2] By utilizing a powerful synthesis algorithm we can have many different timbres, specially if we let performers modify their sound in real-time to create a desirable sound. It is also possible to set up a number of templates and randomly assign each user a predefined setup (in case that the user has little knowledge of sound synthesis). In this example a pair of Martin Birkmanns's 'Two Operator FM PolySynth' were used. At this point we only need to create a control surface in the GUI to allow users control the sound

---

1. Using one DSP for the whole network may produce some friction. A better way of doing this is to have a separate DSP unit assigned to each user by using their IP address.

parameters (Fig. 3.7). Later we have to send the data (notes and synthesis parameters) to network so that each performer can receive the data from other users, and with this information their actions will be reproduced virtually with the second synthesizer (Fig. 3.8). We also need to add a mixer to create a volume slider in the GUI. In order to avoid sudden changes of volume, a ramp can be added so that amplitude changes become gradual and less intrusive. Performers can also use this volume slider to produce longer crescendos or diminuendos without using the tilt option and gradually change the dynamics of their performance. At this point it is possible to add more effects such as reverb, low pass filter, or any sound processing module to the sound. By the end we can add a simple automation for fading in and fading out, for starting and ending of the piece (Fig. 3.9).

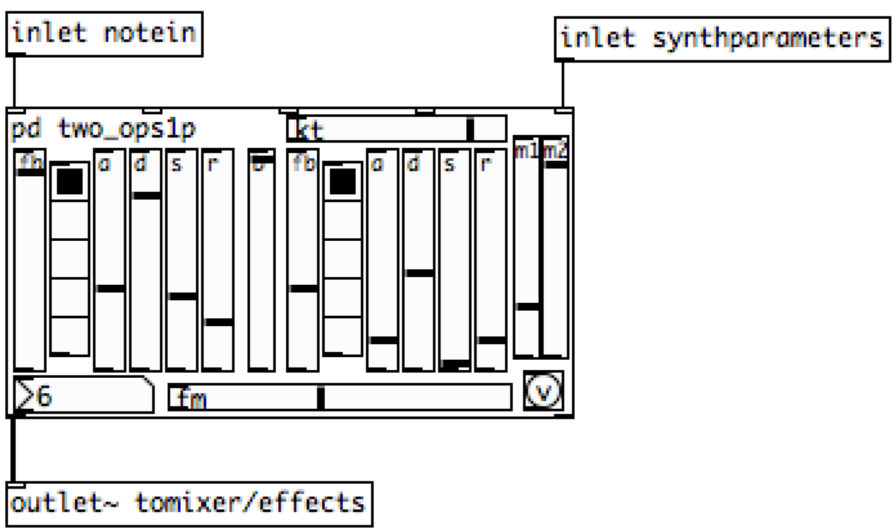


Fig. 3.7 Martin Birkmann's *Two Operator Poly Synthesizer*.

Now we need to create a sub patcher to receive the data from the network in order to recreate the sound of other 'performers'. First we need to send the 'makenote' output (note and velocity) along with the synth parameters to other users. Jascha

Narveson's LANdini, originally written in SuperCollider, is capable of dropping a lot of data via User Datagram Protocol (UDP) multicasting. We send the data to all the users in the network via LANdini except to the device that the data is being sent from (Fig. 3.10) and then create another sub patcher to receive the data from other users (Fig. 3.11).

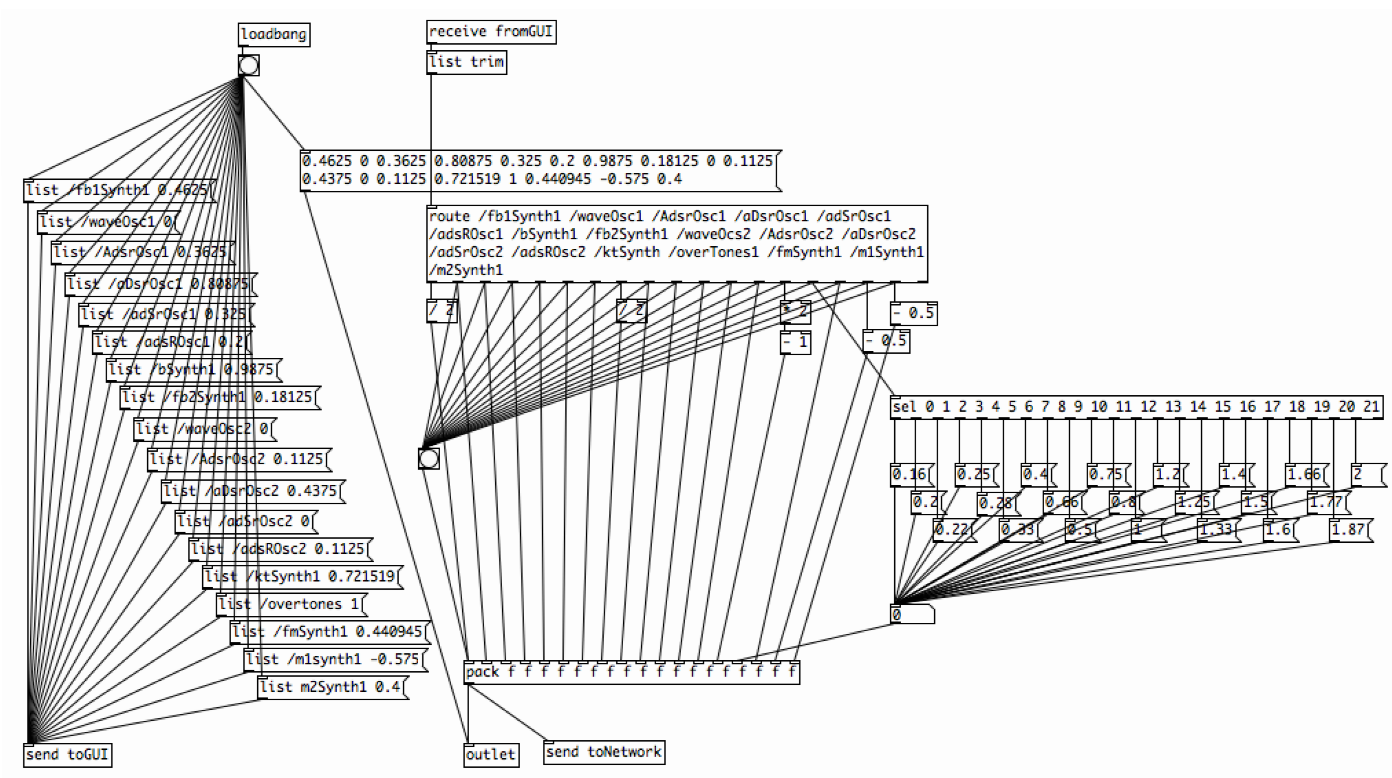


Fig. 3.8 Sending synth data to the network.

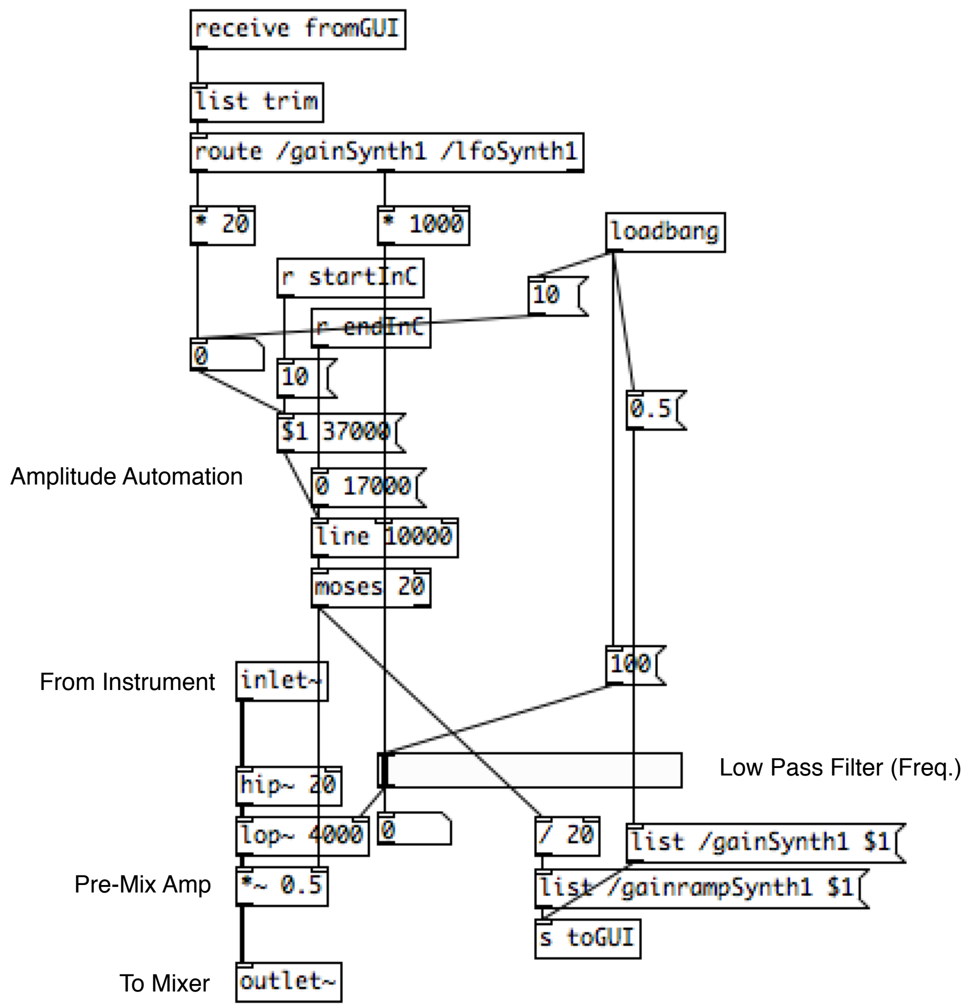


Fig. 3.9 Pre-mixer effects and automation

Now that we have created the sound engine, we can design the GUI itself. First we create a monitor so that each user can see how many users are connected to the system (Fig. 3.12). It is important to schedule all the processes; it means that all the tables has to be loaded first before starting the performance. Here we also created a simple animation to inform the users when to start and when the piece is going to end (Fig. 3.13).

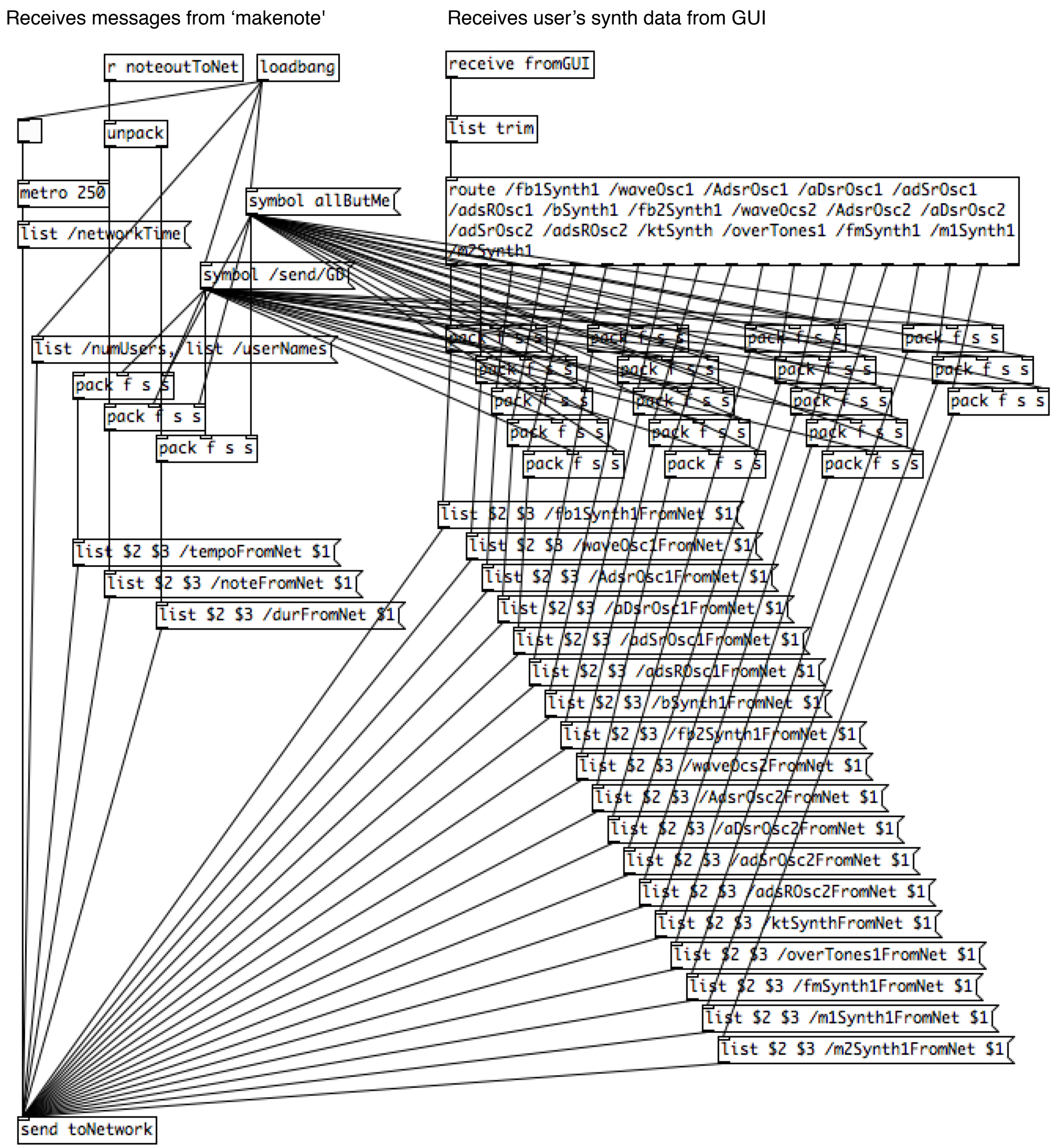


Fig. 3.10 Sending messages to network via LANdini.

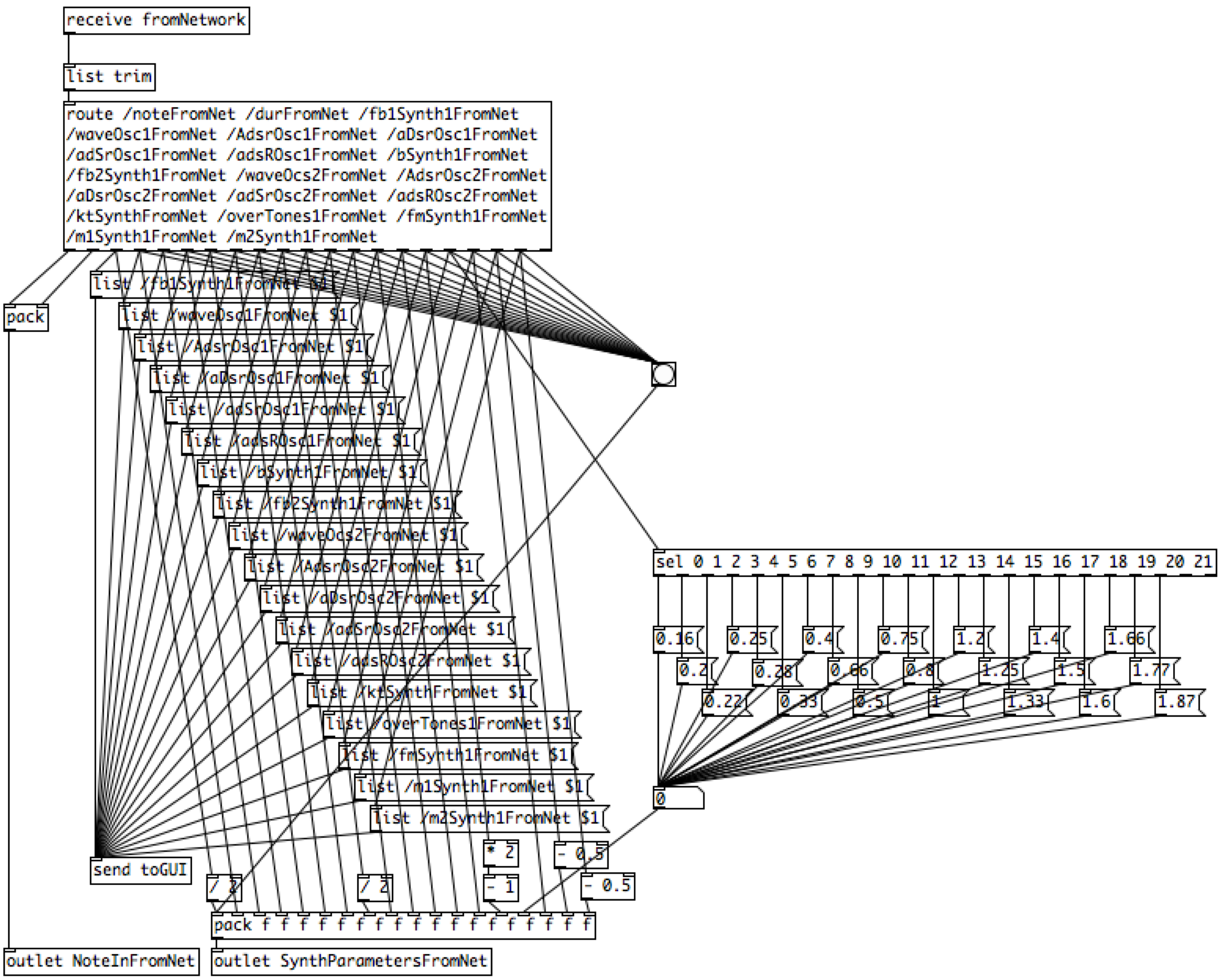


Fig. 3.11 Receiving messages from network via LANdini.

In addition to the parameters for the control surface of the synthesizer, a mirrored image of the same surface is created in the second page of the GUI for the network synthesizer, so that each user can have the visuals of the interaction and see which parameters are being changed by other performers. To keep it simple, in the main page of the GUI we only need a few buttons: to *start*, *end*, and to *go forward* in the score. We also need a slider to control the tempo so that each performer can sync their tempo to the rest of the group. Also, a number box' was added in the GUI so that the user knows which pattern is being currently played. A few sliders have also been added to control the volume, the lowpass filter, as well as a record 'toggle' so that each user can record the whole performance. We can also add some buttons to control the GUI interface itself, for example another 'toggle' to turn off the graph, and a few more slots in case we want to add more features like sound templates, etc. in the future.

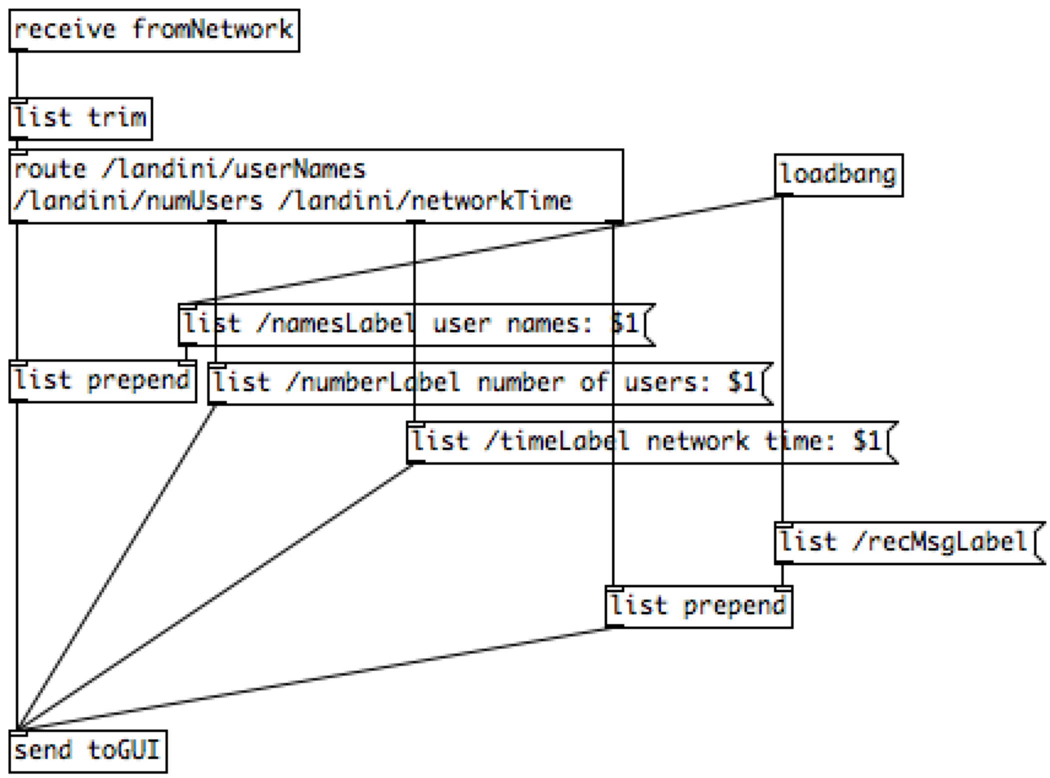


Fig. 3.12 Receiving number of users and user names from the network.

At this point we have created all the components we need. By creating abstractions we can view the whole scheme below (Fig. 3.15). Finally we design the GUI in MobMuPlat editor (Fig. 3.16).

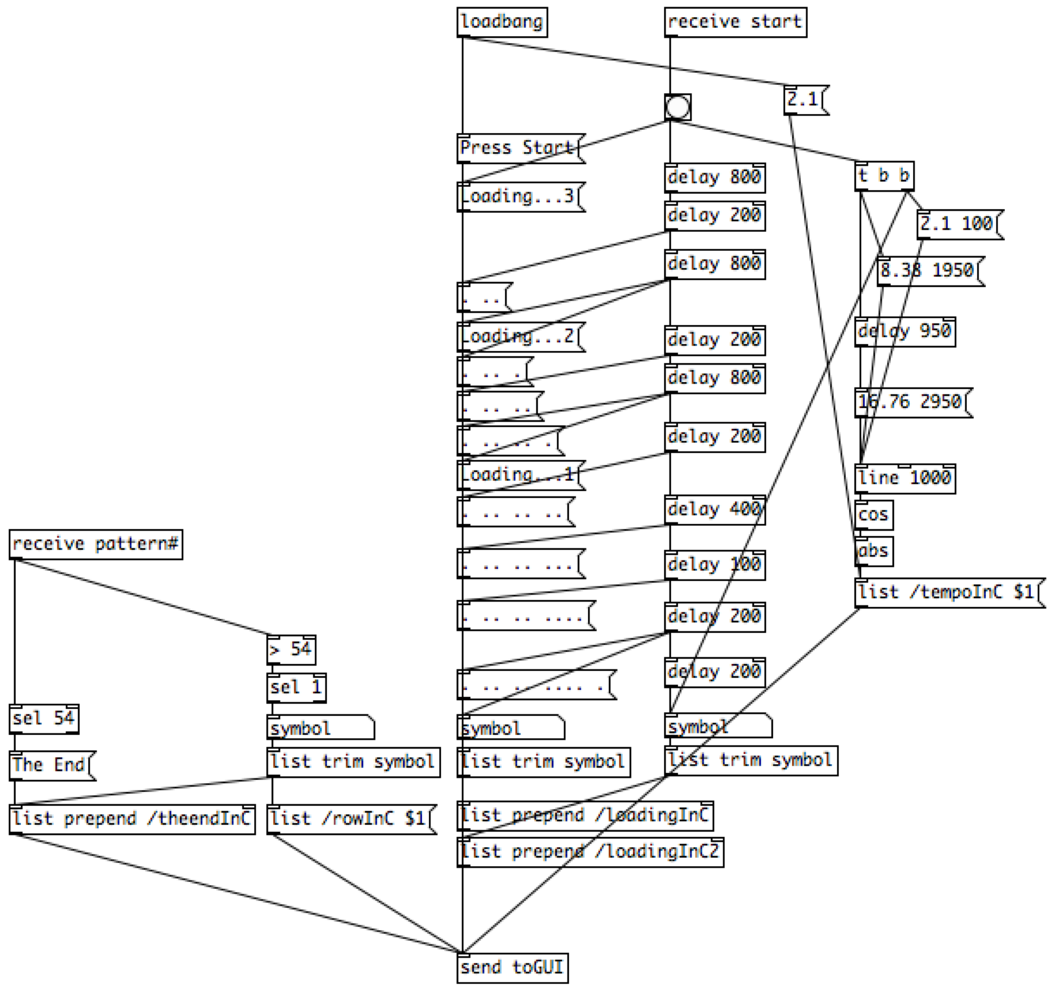


Fig. 3.13 Loading Animations.

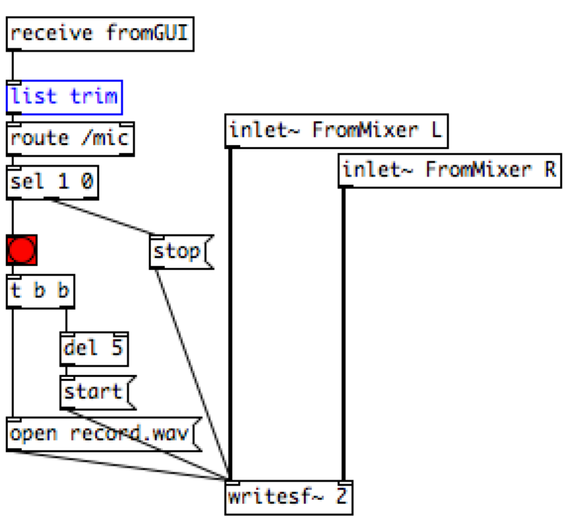


Fig. 3.14 Recording Audio.

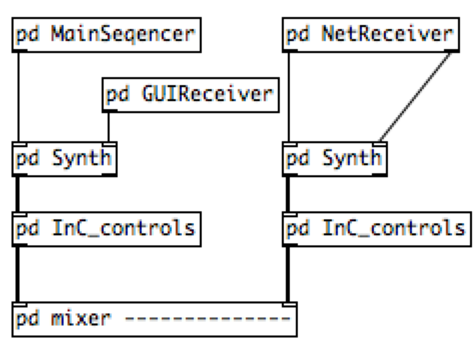


Fig. 3.15 Final stage abstractions.

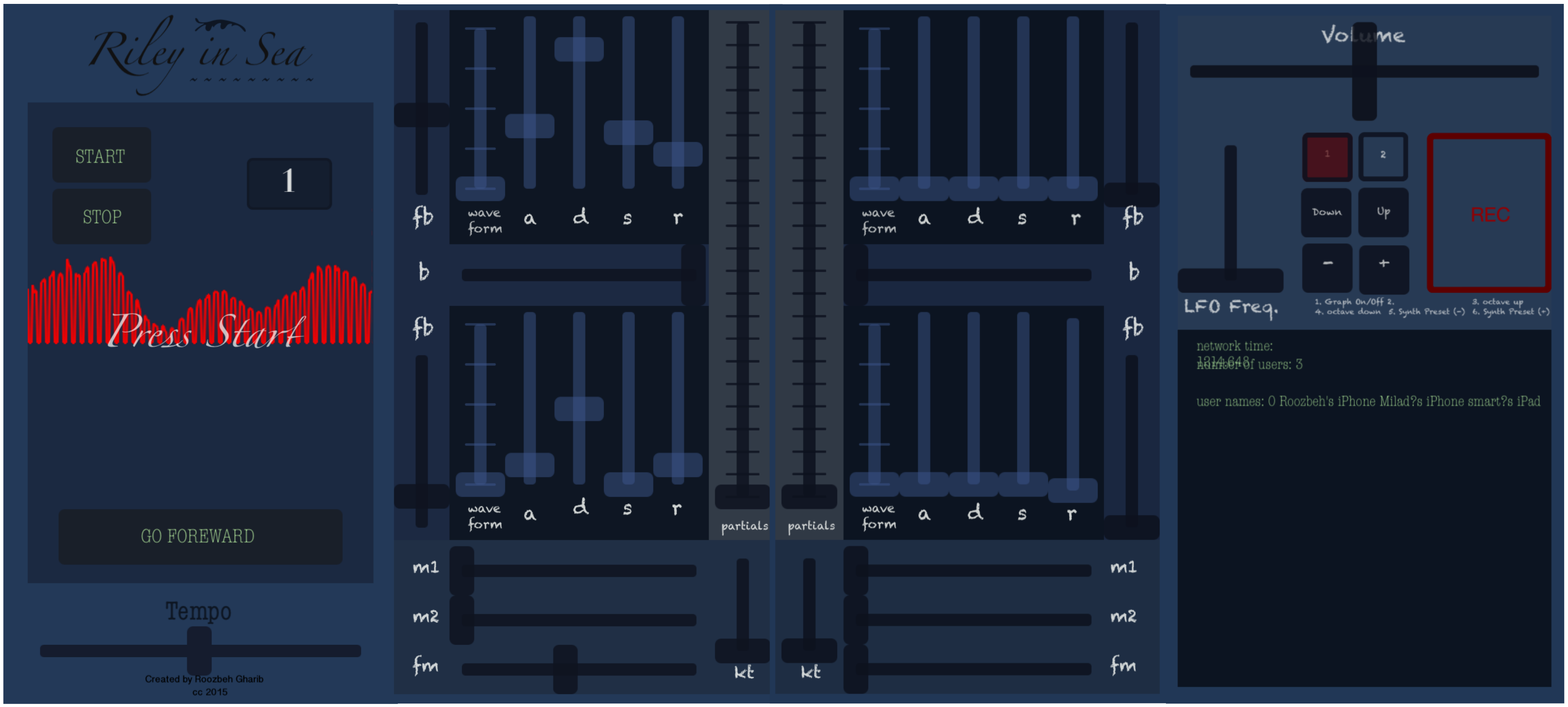


Fig. 3.16 *Riley in Sea*'s GUI interface.

Chapter 4: Computer-based interactive composition for public spaces

The initial idea for this thesis was to create an interactive composing system in which the audience could simultaneously compose and perform by interacting with the system. Some of the elements of this concept are evident in the works of John Cage, Lajaren Hiller, and Iannis Xenakis meanwhile performance with automated electronic systems is an important part of the music of Giuseppe Englert, Morton Subotnick, and many others. Arguably the first real interactive system was built between 1969 and 1972 at University of Illinois by Salvatore Martirano. Martirano built a system that could fully exemplify the concept of interactive composing. According to Chadabe Martirano's *SalMar Construction* consists of, "analog sound generators and modifiers controlled by digital circuits, a console with over 200 touch-sensitive switches, and 24 loudspeakers arranged throughout the performance space. The system automatically generates sounds with different timbres, pitches, and loudness, and routes the sound along four paths through the arrangement of loudspeakers" (Chadabe 1983). In *SalMar Construction* the player can perform in reaction to what is being heard, manipulating aspects of the music, such as pitch, rhythm, tempo, pattern, octave, spatial distribution, and cycling.

Early interactive composing systems were using sequencer-programmable analog synthesis systems which could not be transported to concert halls. By 1977, composers began to compose for and perform with small computers and digital synthesizer systems, which could be carried around. *Solo* (1987) and *Rhythms* (1980) by Joel Chadabe were among the first computer music statements of interactive composition (Chadabe 1982). According to Chadabe, *Interactive Composing,* grown from his work since 1967, which is a two-stage process that consists of creating an interactive composing system, and simultaneously composing and performing by interacting with that system, as it functions (Chadabe 1983).

> Creating the system involves bringing together a programmable computer, synthesizer, and at least one performance device, and programming the computer with algorithms that function automatically and in real time to:
> - Interpret a performer's action as partial controls for the music,
> - Generate controls for those aspects of the music not controlled by the performer,
> - Direct the synthesizer in generating sounds. (ibid.)

According to Chadabe the goal of interactive composing is to engage a performer in an unusual challenging environment:

> An interactive composing system operates as an intelligent instrument —intelligent in the sense that it responds to a performer in a complex, not entirely predictable way, adding information to what a performer specifies and providing cues to the performer for further actions. The performer, in other words, shares control of the music with information that is automatically generated by the computer, and that information contains unpredictable elements to which the performer reacts while performing. The computer responds to the performer and the performer reacts to the computer, and the music takes its form through that mutually influential, interactive relationship. The primary goal of interactive composing is to place a performer in an unusually challenging performing environment.(ibid.)

4.1 - Interactive music composition using mobile phones

According to Nathan Bowen, mobile phones have been one of several kinds of digital tools that allow artists to bring out musical performance and sound art outside the concert hall or gallery.[1] For example John Eacott's *Intelligent Street* (2004) represents an SMS-based interactive music environments in which users collectively affect a shared soundscape via text messages while resulting changes were played over loudspeakers placed over a high-traffic public walkaway. Although using participants' locations as a musical controlling parameter were used before GPS technology become available in mobile phones,[2] implementing GPS in musical context is directly affordable with current smart phones. GPS technology can be utilized to expand the concert hall or performance space, on a larger geographic scale. Choloniewski's compositions *GPS* (2000), and *Trans* (2001) are pieces that pre-dates GPS-enabled mobile phones, but exemplify this approach of 'spread out' concert hall. With GPS-enabled devices equipped with headphones, 'sound walks' have become more popular because of the accurate location tracking. Sound walks allow participants to explore terrain via 'augmented' soundscape, with digitally recorded sounds corresponding to specific geographic locations (Bowen 2013). For example Layla Gaye's *Sonic City* (2003) incorporated GPS locative devices among several other sensors to allow users interact with urban landscape (Gaye 2001). Another example is Mark Shepard's *Tactical Sound Garden* (2007) that uses the garden metaphor to establish a presence between users separated by time but sharing the same geographical location. In Shepard's project

1. Other tools being networks, broadcast media, and the internet.

2. In experiments by Tanaka and Gemeinboeck (2006) participants physical positions within a concert space were deduced by determining each phone's signal strength as received over a GSM cell antennae (Tanaka and Gemeinboeck 2006).

users ‘plant’ or ‘prune’ sounds at a specific urban location. In his dissertation Nathan Bowen (2013) argues that, “Mobile phones can allow communities to emerge through shared sound in like manner to an online forum, which allows conversation to take place without regard to physical location or proximity through time.”

In the study by George Essl and Michael Rohs (2009), authors present a detailed summary of the technical capabilities and limitations of mobile handheld devices. Essl and Rohs argue that, “in the same sense that a good composer of orchestral music has to know the capabilities and limitations of orchestral instruments”. DMI designers and composers must be familiar with the characteristics of mobile handheld devices, in order to make proper design decisions. According to their study, the different kinds of musical interactions that onboard sensors support, include optical tracking of markers and grids using phone’s camera, optical movement detection, acceleration sensing, magnetic field sensing, gyroscopes, touch and multitouch screen, capacitive proximity sensing, and microphones. Yet complex synthesis algorithms or memory-intensive musical processes are not viable for phones at this time.

Another current trend in mobile music development is the creation of modular toolkits and programming interfaces permitting real-time sound synthesis which according to Bowen (2013) are clear attempts to attract more composers and programmers to mobile music community. Mobile phones can also operate as controllers rather than as a primary sound source: TouchOSC, Mrmr, Control, C74, MobMuPlat, and OSCemote are popular apps that transmits Open Sound Control (OSC), and in some cases MIDI messages while some permit custom layout design at the same time.

4.2 - Presentation of compositional framework

As mentioned earlier the compositional idea for this thesis was to engage the listener in an experience that involves some degree of interactivity, which means the physical response of the listener drives the sound, and vice versa. This coupling happens naturally in physical activities that involves making sound, such as singing, or playing a musical instrument where one is being involved with the sound in a complete understanding of the casual process. Even sometimes, on a higher level, the body responds involuntarily to sound as in ecstatic listening which creates a sensation of shivers, called frisson in physiological terms, localized in the back, neck and shoulders of an aroused listener. Now we know that ecstatic listening involves release of peptides by the brain's hypothalamus, in other words hearing can be medication (Huron 2002). Furthermore, designing sonic interactive systems that physical gestures can control the sound may lead to new possibilities for novel applications in movement learning such as sports, dance pedagogy, and rehabilitation (Françoise et al. 2015).

Given the ubiquitous nature of smartphones, most people now have the opportunity to interact with music. In a study by Chao, Chen, and Chen (2014) authors discovered that interacting with classical music can improve the preference of music and that people admire classical music more after experiencing their system which adjusts the tempo by analyzing gestures using a smartphone. Authors argue that, "in order to attract most people, regardless of their abilities, the interaction approach should be the most familiar one possible and the device should be popular and nearly ubiquitous." (ibid.)

The aim of this thesis was to create a mobile application designed for a selected public space, which in this case was the York University Keele Campus located in Toronto, Canada. To experience the work visitors must use their smartphone (or tablet computer) to access the existing audio tour for a number of buildings around the Harry W. Arthurs Common, the primary open space of the campus and the symbolic heart of the University. Being an ideal setting for strolling, socializing, contemplation, circulation and celebration, the Commons can serve quite well for functioning as an interactive environment where visitors connect and cooperate in a musical setting. By using the University's wi-fi network visitors can connect and collectively listen to the sound they are making or create an ensemble to perform a piece of music. By analyzing the GPS data, the application enables participants to consciously use their movement to 'play' different areas. The Commons has been divided into smaller regions to retrieve elements of an audio narrative at each specific location.[3] From the compositional point of view, the place becomes the canvas for a narrative experience on site which creates a changed awareness of situation. For example, coordination of the central pond in the middle of the Commons area was used to trigger a recorded sonic narration of the weather. Using regular stereo sound, pre-recorded ambient sounds were used to create a 3D audio scene by reproducing psychoacoustic sound localization and distant cues. Listening with headphones on, participants hear birds spread around the area, occasionally flying around their head, as gradually the weather changes and rain starts with remote thunderstorms. Overdubbed with piano phrases extracted from Miles Davis solos on "So What" by using first order Markov chains, the final result is a different,

3. Although the system is capable of incorporating visual elements as well as audio, this project was restricted to audio material.

arguably relaxing, sound environment in an ‘augmented’ reality which participant can walk out, or return to hear the left over.

The core part of the patcher uses the GPS data to trigger different compositions by solving whether the current location of the user lies inside, or outside the defined polygons (surrounded buildings and smaller areas inside the Common region). This was done by using a simple method which calculates the area of triangles formed by each edge’s endpoints of a polygon and the test point (user’s location). If the sum of the area of created triangles is zero, the point is inside; if not, it is outside (Fig. 4.1).[4]

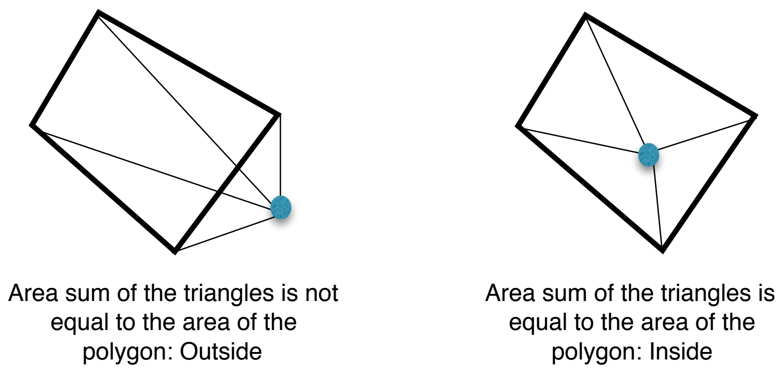


Fig. 4.1 PIP (Point in Polygon) test.

Beside the GPS detection, another sub-patch were used to identify the physical activity of the user. Using the accelerometer data it detects whether the user is walking or running. As soon as the user starts to run, it switches to another composition written for the ‘jogging’ mode.

Each subarea of Harry Arthurs Common can be viewed as a fragment of a bigger composition. While interacting with the location and controlling different aspects of the

4. A better way of doing this is by using ray casting algorithm.

music such as tempo, dynamics, timbre, etc., in order to listen to the whole composition, participants should walk through all the areas. (Fig. 4.2)

Beside the Commons piece which uses pre-recorded material, the rest of the pieces are algorithmic compositions that use all of the Rowe's dimensions of interactive systems.[5] Each piece have been used to experiment with different deterministic or non-deterministic compositional methods such as modal counterpoint, tonal harmony, serial music, stochastic, and aleatory composition. The code of each fragment can be seen in chapter 4.3.

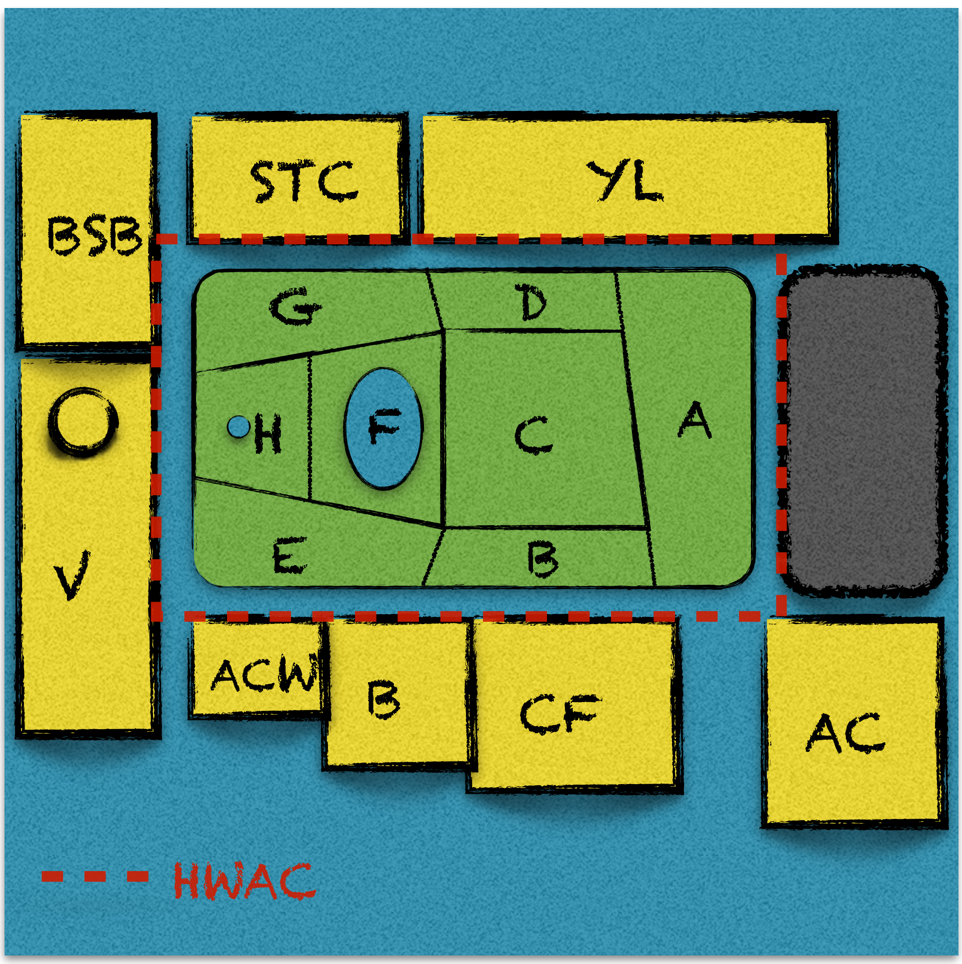


Fig. 4.2 Composition map of the Keele Campus.

5. Transformative, generative, sequenced, score-driven, performance-driven, instrument paradigm, and player paradigm.

4.3 - Presentation of the code

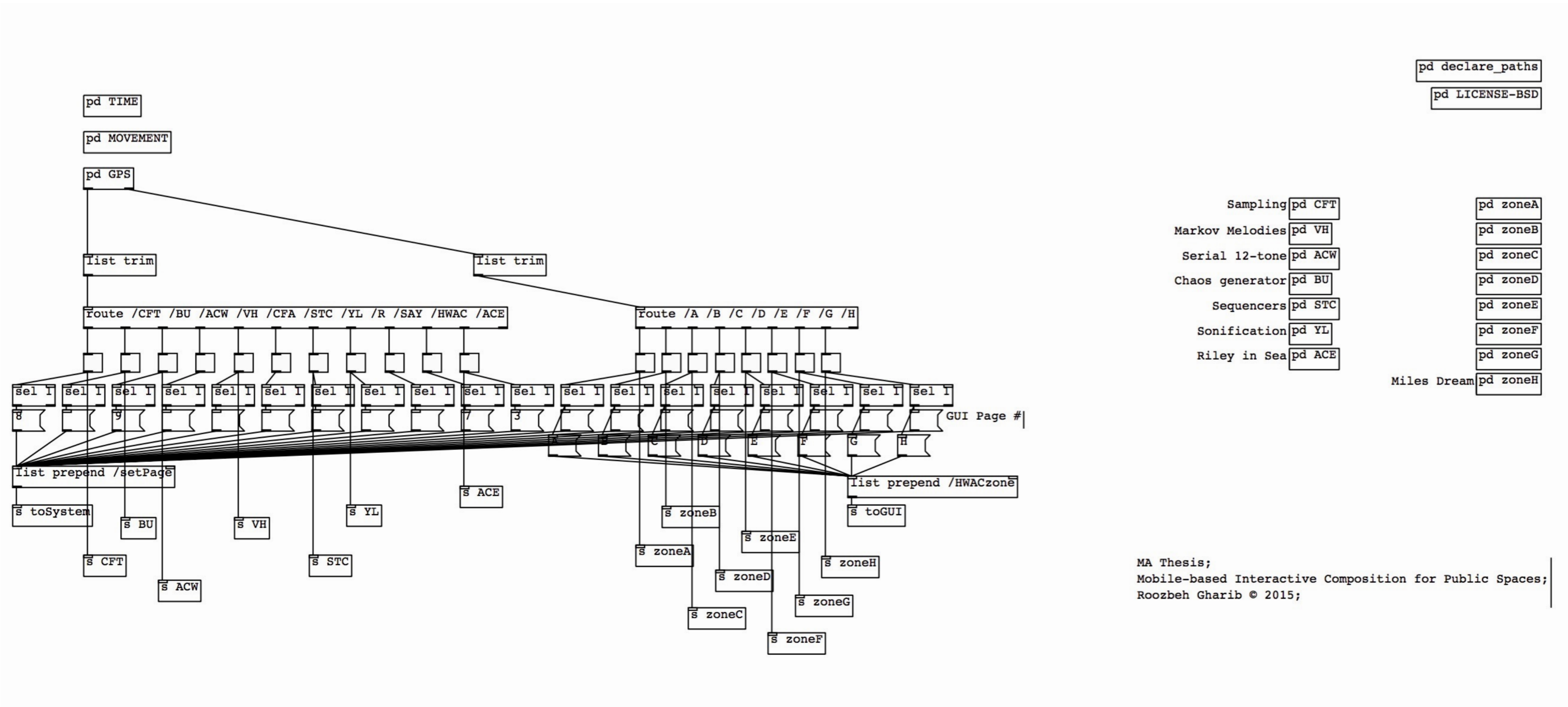


Fig. 4.3 Main patcher which triggers sub patchers for each composition.

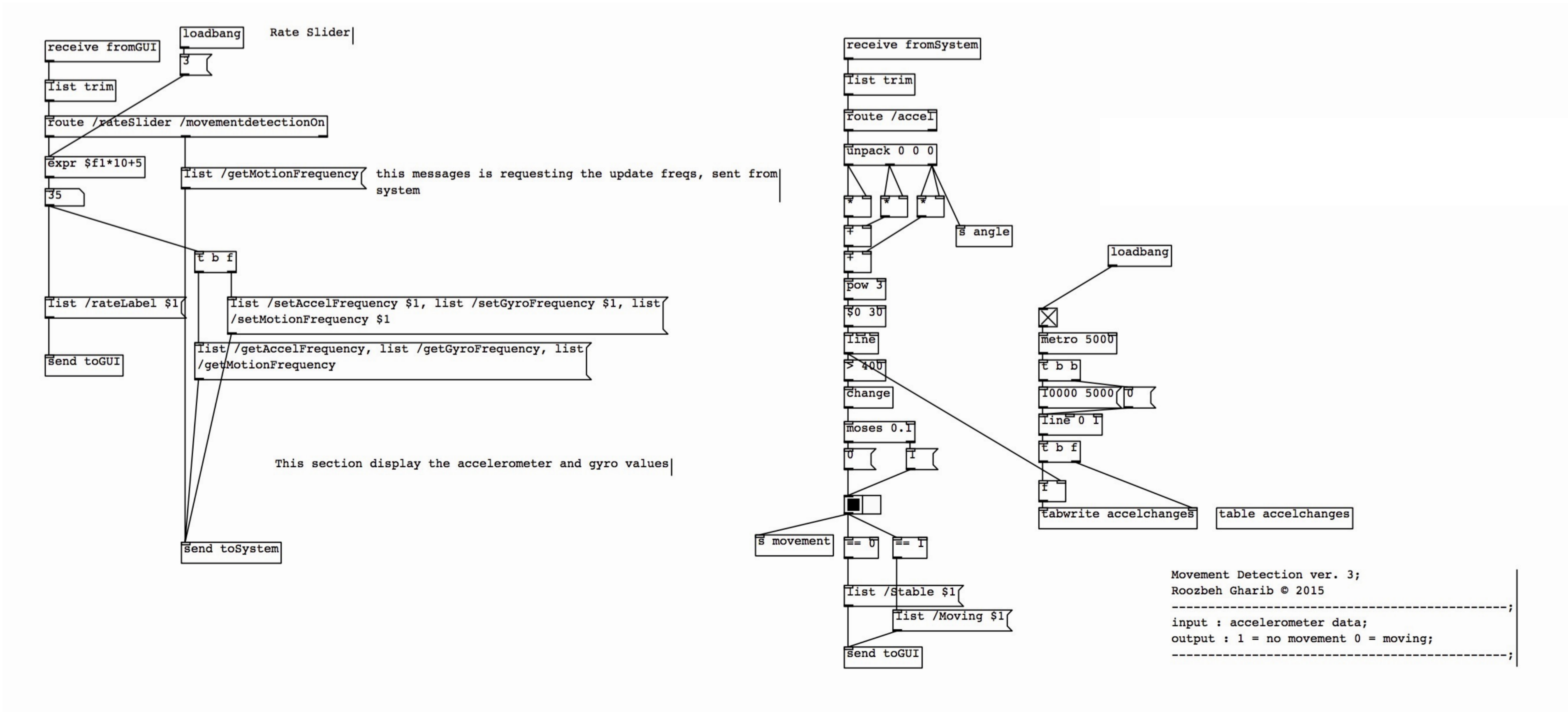


Fig. 4.4 Movement Detection: This patcher triggers the jogging mode.

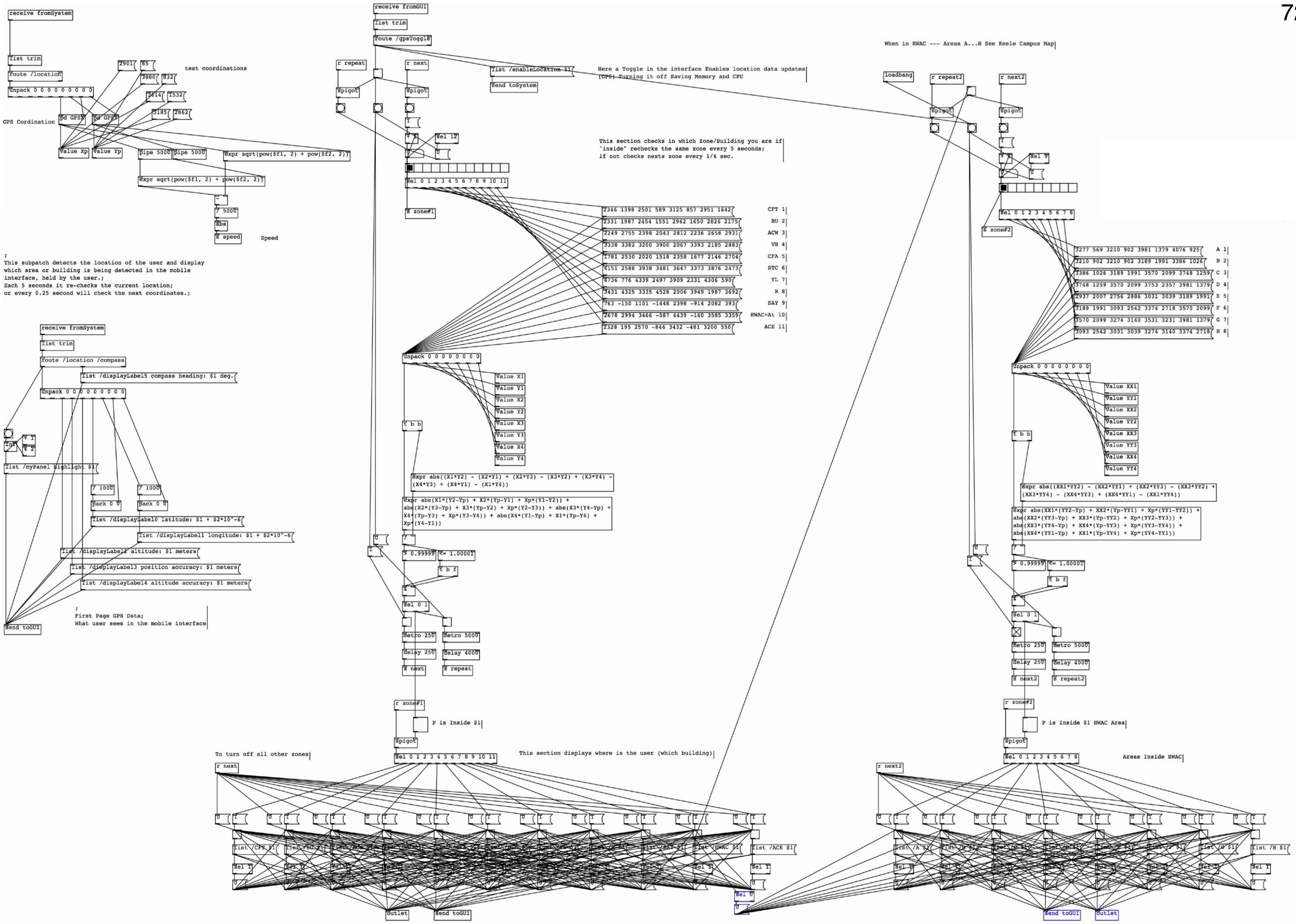


Fig. 4.5 GPS location detector.

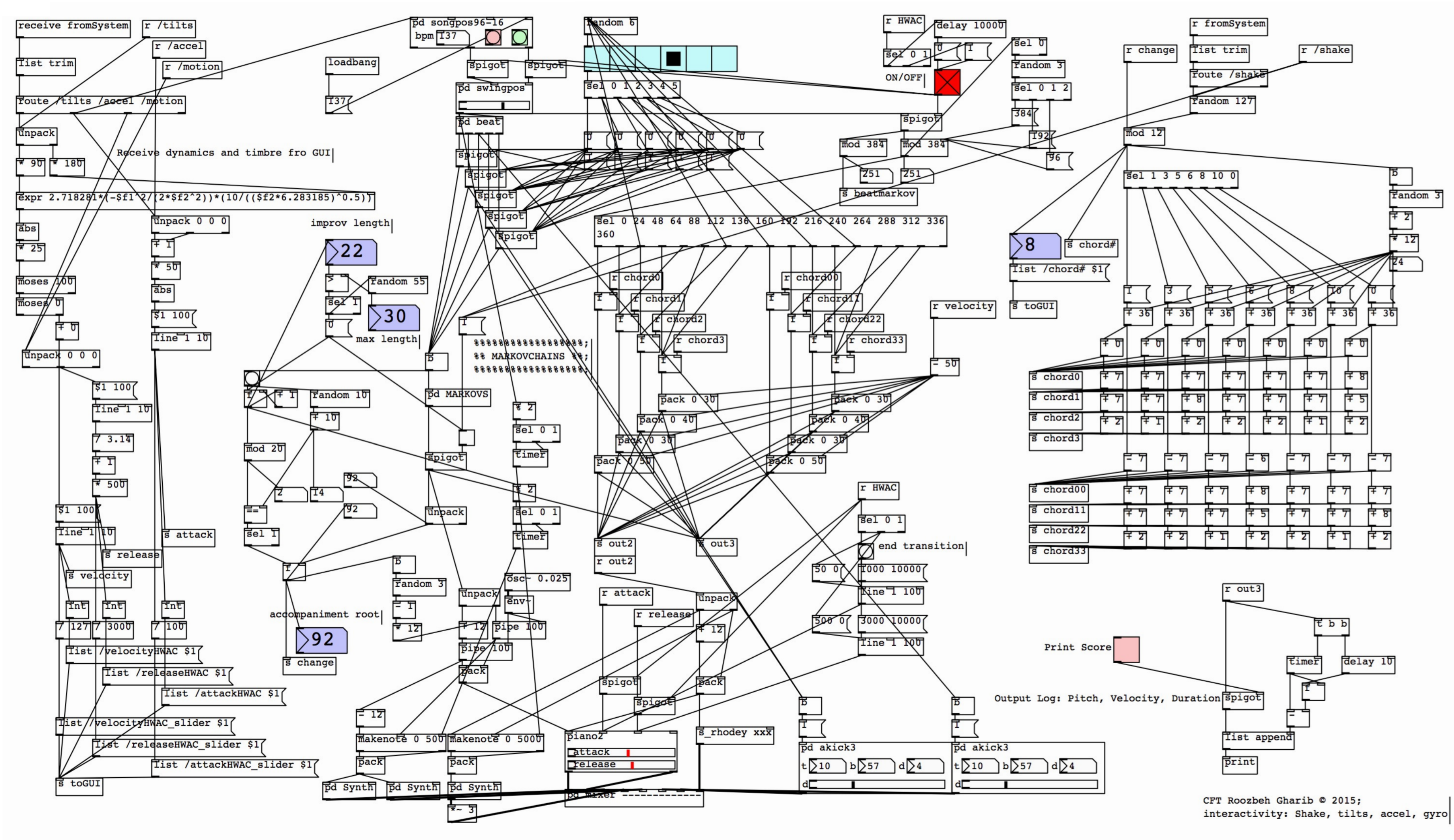


Fig. 4.6 VH: Composition based on melodic generation using first order Markov chains.

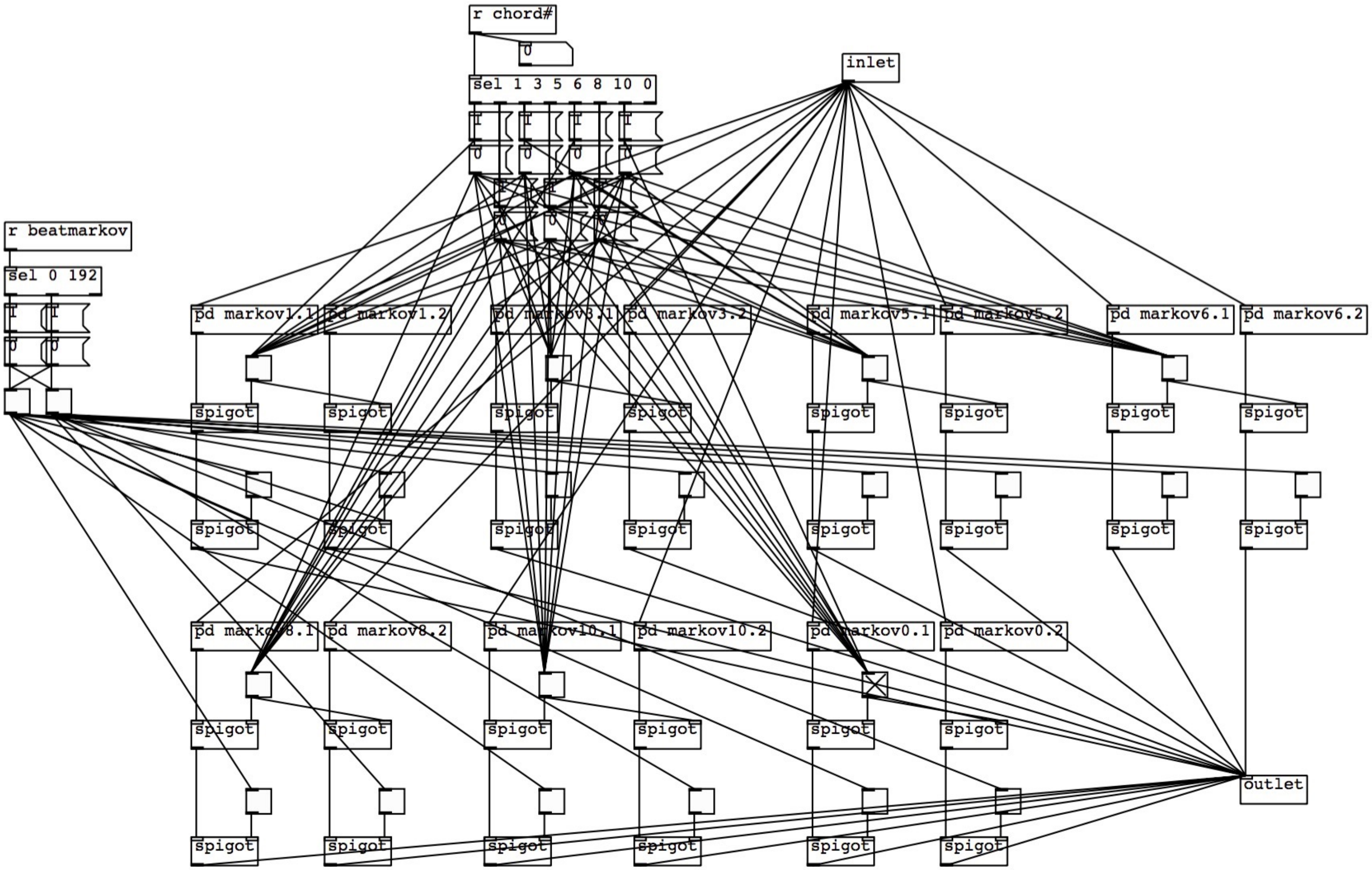


Fig. 4.7 ‘MARKOVS’ sub patcher used in HWAC

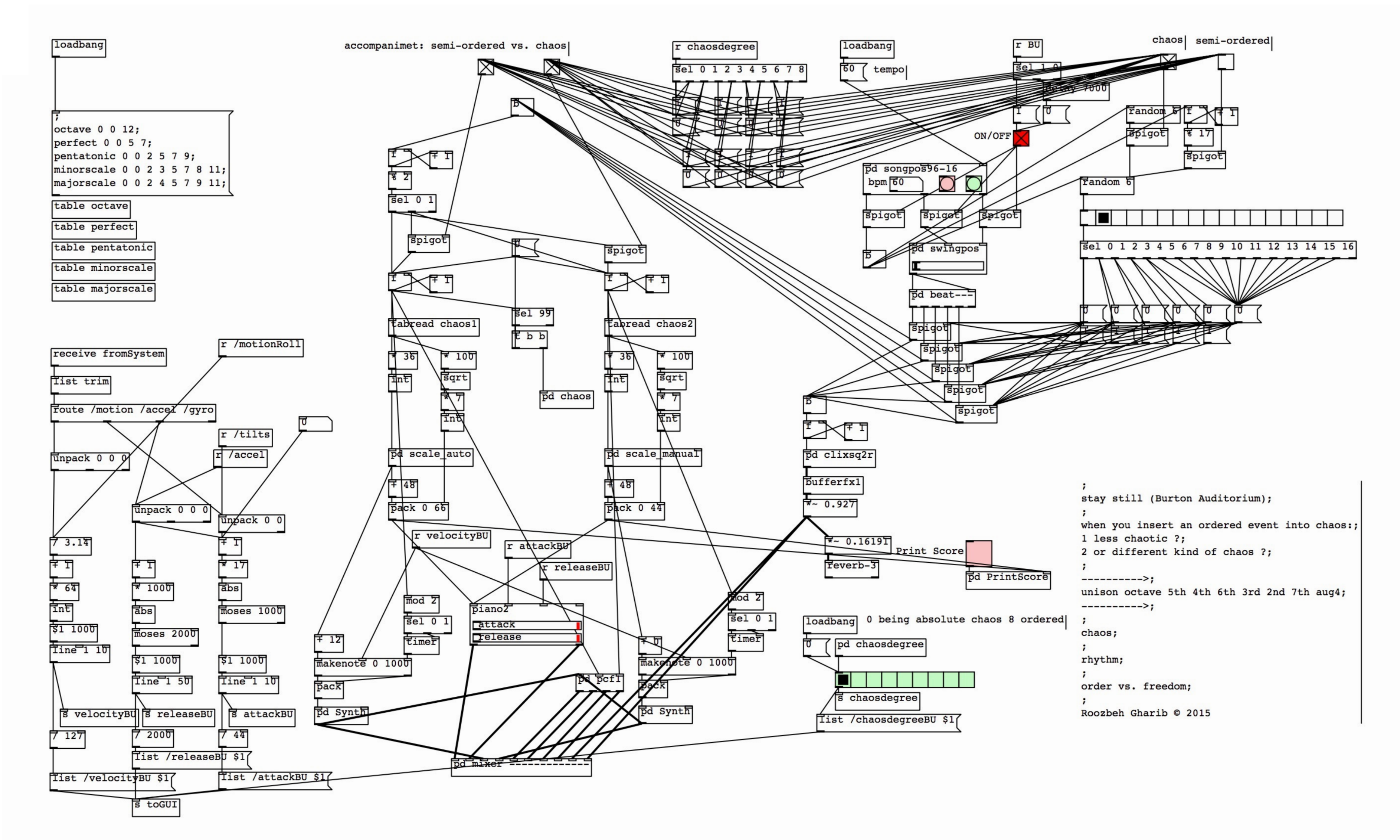
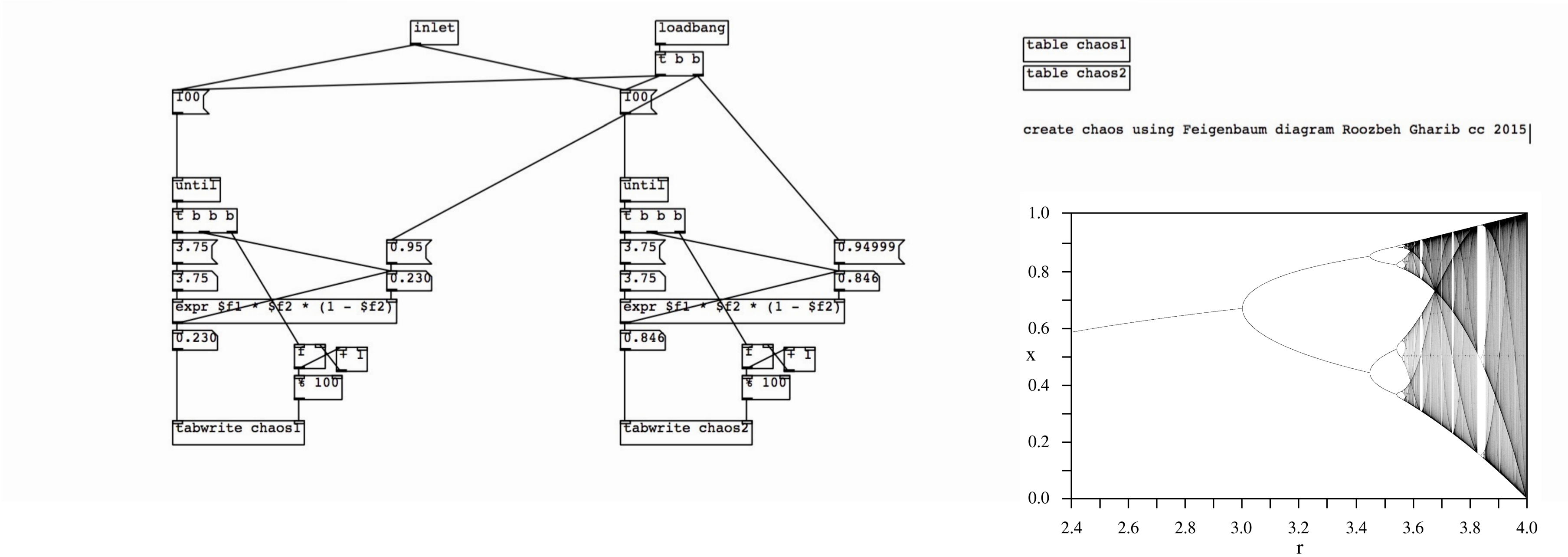


Fig. 4.8 BU: Composition based on chaos theory using Feigenbaum diagram.



$$x_{n+1} = rx_n(1 - x_n)$$

(Nierhaus 2009)

Fig. 4.9 'chaos' sub patcher used in BU

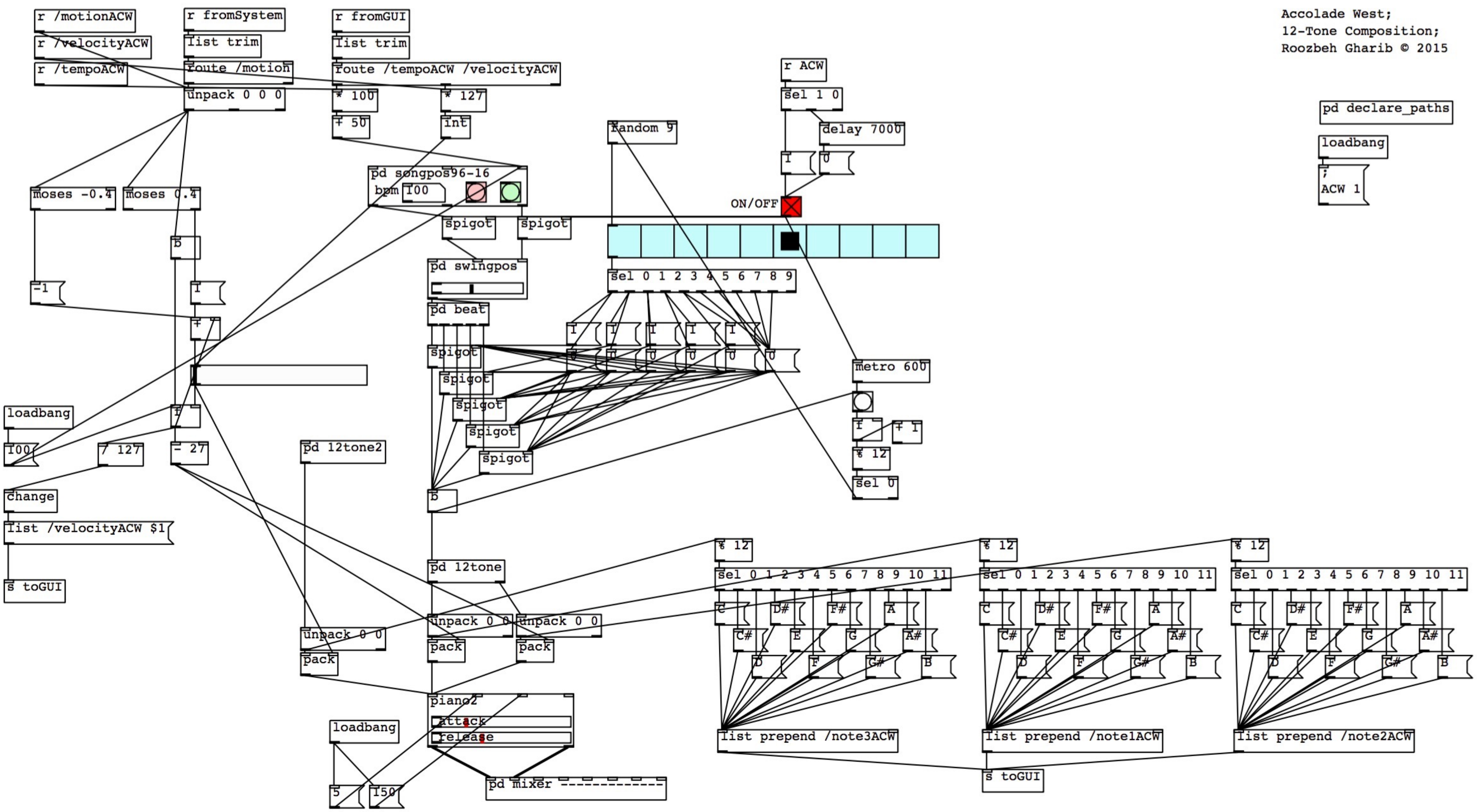


Fig. 4.10 ACW: Composition based on 12-tone technique (Serialism).

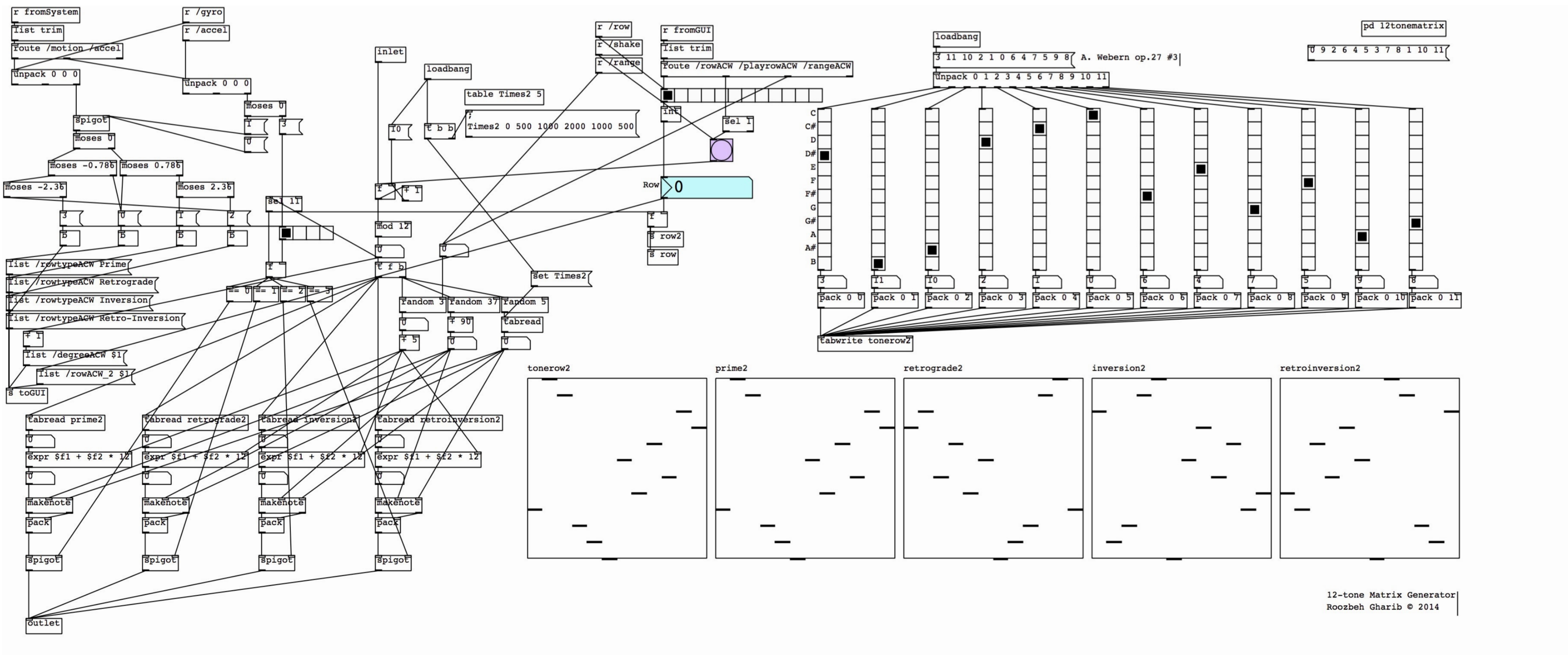


Fig. 4.11 ‘12tone’ sub patcher used in ACW

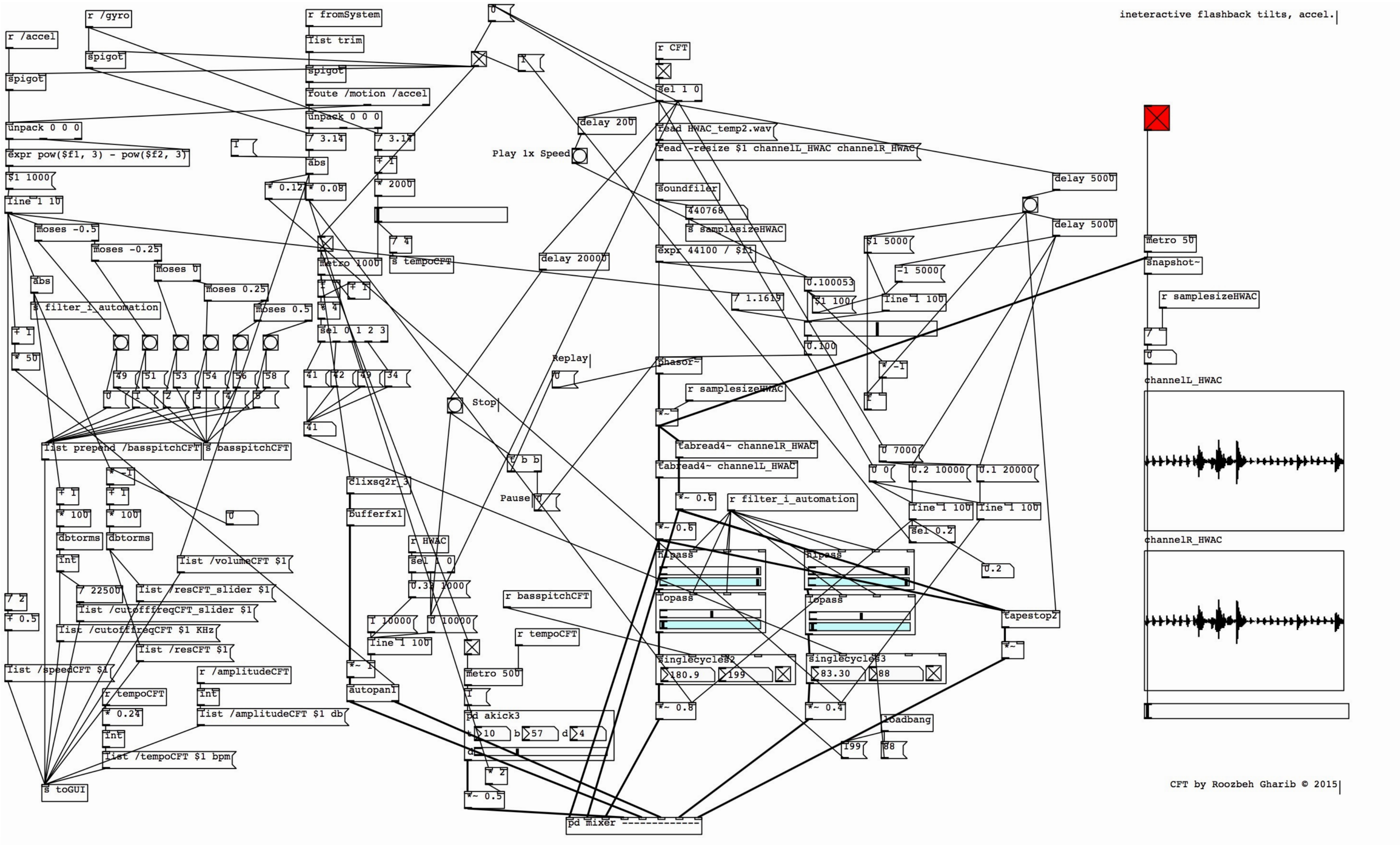


Fig. 4.12 CFT: Composition based on Sampling technique.

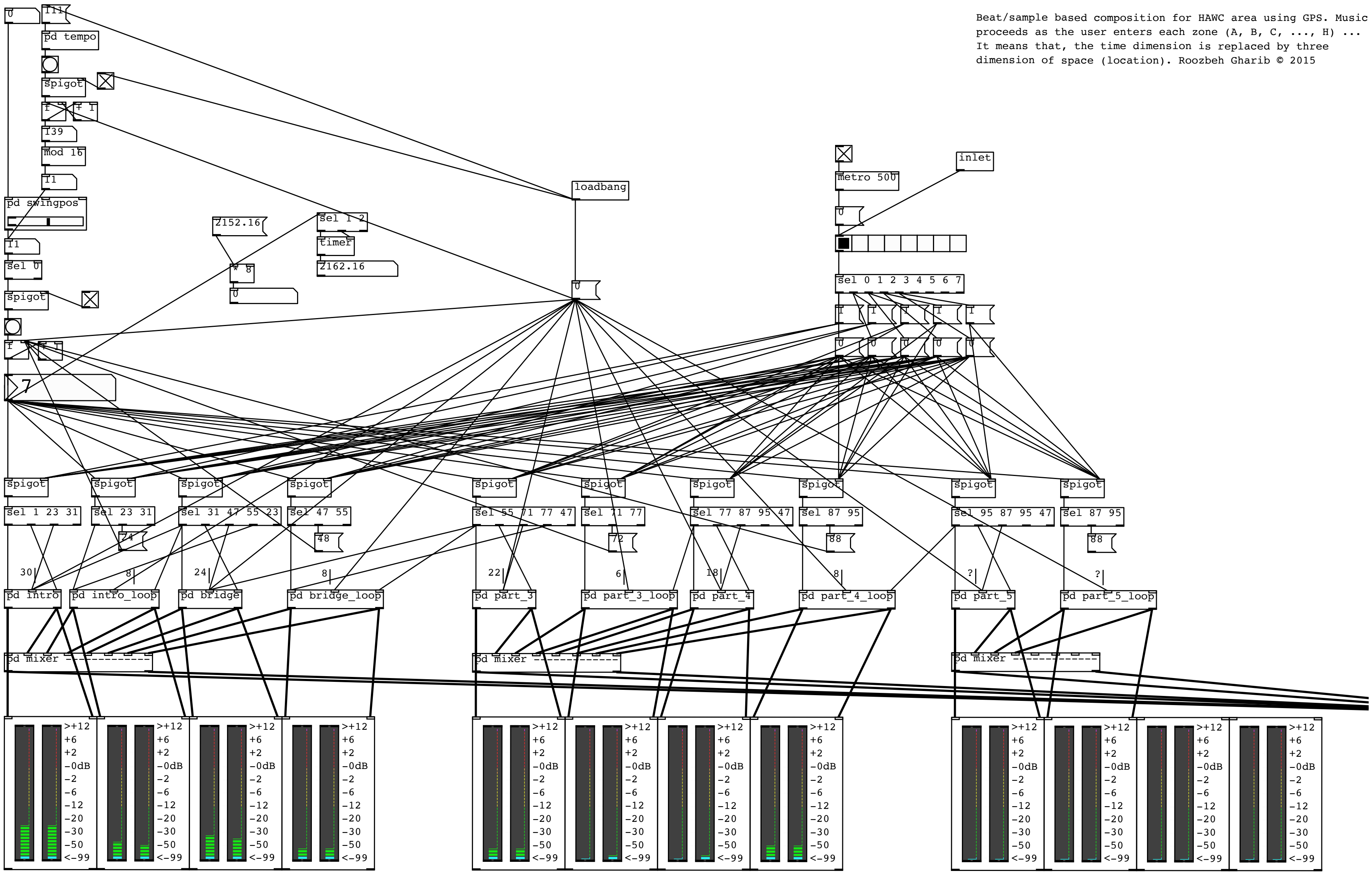


Fig. 4.13 HWAC: Composition using pre composed audio segments.

"Every act of perception, is to some degree an act of creation, and every act of memory is to some degree an act of imagination."[6]

4.4 - Conclusion

Music is an abstract form of art, yet extremely emotional. Without any mediation, music can express inner states and feelings, but it has no power to represent anything in particular or external (Sacks 2007, 329).[7] Music among other forms of cultural activities, but perhaps more effectively than any other form of expression, recalls a meaning that lies outside and beyond the self and contributes to the ideas that movements offer, which may create an opposition to the existing social and cultural order. Without saying that such truth-bearing is inherent in music, "it can be utopian and premodern" (Eyerman and Jamison 1998, 24). Social movements create a situation, or a context, where music can recover at least some of it's ancient, truth-bearing role:

> By focusing on the interaction of music and social movements, we (the authors) want to highlight a central, even formative aspect of cultural transformation … We conceive of these relations between culture and politics, between music and movements, as collective learning process … we have sought to identify the knowledge-producing activities that are carried out within social movements … this "cognitive praxis" has affected scientific research programs and professional intellectual identities. Social movements have provided contexts for the politicization of knowledge, and the effects have often been profound on scientific theorizing, disciplinary identities, and even technological development trajectories. (ibid., 7)

---

6. Oliver Sacks (2007)

7. "Music is an art that has no 'meaning': hence the primary importance of structures that are properly speaking linguistic, given the impossibility of the musical vocabulary assuming a simply communicative function." (Boulez 1986, 32)

While personal interest or political misunderstanding can hardly be excluded from among the motivations that informed Futurist participation in fascism[8], employing non-traditional sound-generation techniques as part of the communicative art form was an inevitable outcome of the Futurist movement and Industrial Revolution. The broad improvement in instrument construction and new technical resources had a major stylistic impact on composers and it may be argued that, the Digital Revolution (known as the Third Industrial Revolution) would have a profound irreversible consequences too. The Third Revolution not only allowed artists to record music, build websites, crowd-fund, release, and promote their music online, it also affected the music performance, and the audio recording industry. While on-demand streaming and digital downloads left the music industry in an economic chaos, the evolution of the computer and the ability to record digital audio made music production more affordable to everyone.

It's been argued that music industry has to find new ways of connecting with clients. The creative possibilities offered by technologies including augmented reality, artificial intelligence, wearable technologies, robotics, and 3D printing are pushing the possibilities of coding as a creative art form while artists across all art forms continue to use digital technology within their works - including Merce Cunningham, Robert Lepage, Brian Eno, Aphex Twin, Rain Room, etc. Recently, sound art is finding increasing acceptance in the culture of galleries and museums where artists have sought to find new strategies for presentation of their works: Björk's *Biophilia* (2011) consists of a series of apps linking the album's themes to musicology concepts, Arcade Fire's

8. See (Gentile 2003).

interactive video “The Wilderness Downtown” (2010) utilizes the latest open web technologies to let the audience participate in the visuals allowing an emotional connection, *REWORK* (Philip Glass Remixed) by Scott Snibbe Studio (2013) features eleven interactive music visualizers and an interactive instrument for the Beck/Phillip Glass collaborative remix album, or *Inception - The App* (2010) which transports the movie *Inception* into the real life by offering a dream machine that transforms the world around you into dreamworlds created by the composer, Hans Zimmer, and RjDj.

Although making use of forms of digital technology may not reinvent the music, the experimentation between software developers and musicians will definitely help to ensure that music continues to be at the forefront of the digital curve. For example, partnered with IBM, James Murphy created an algorithmic generated music by using raw tennis data from the US Open Grand Slam Tennis matches in 2014, in which every break point, foot fault, etc. had been programmed to give each match a unique music score.

Although this thesis was written for a neophyte who wants to learn how to compose interactive music using computers, it also tries to raise underlying questions regarding the nature of musical meaning and expression, the position of music in history and culture, the role of music in the context of society at large, and possibly tries to re-evaluate Schopenhauer’s remarks that music is “perceived solely in and through time, to the complete exclusion of space.”[9]

—RG, Nov. 2015

9. See Robert P. Morgan’ s article “Musical Time/Musical Space’ (1980).

Bibliography

- al-Farabi, Muhammad ibn Muhammad ibn Tarkhan. 2009. *Kitab Al-Musiqa Al-Kabir*. [Cairo]: Dar al-Kutub.
- Avicenna.,. 1958. *Le Livre De Science*. Paris: Les Belles Lettres.
- Barknecht, Frank. 2006. '128 is not enough - Data structures in pure data'. In *LAC2006 Conference Proceedings*: 61-66.
- Brinkmann, Peter. 2012. *Making Musical Apps*. Sebastopol, CA: O'Reilly.
- Bittencourt, Marcus Alessi. 2005. 'Acousmatics, Sound Objects And Instruments Of Music'. In *Spark 05 Festival of Electronic Music and Art.* Minneapolis, Minnesota, EUA: University of Minnesota Twin Cities: 46-49.
- Boulanger, Richard Charles. 2000. *The Csound Book*. Cambridge, Mass.: MIT Press.
- Boulez, Pierre, and Jean Jacques Nattiez. 1986. *Orientations*. Cambridge, Mass.: Harvard University Press.
- Bowen, Nathan. 2013. 'Mobile Phones, Group Improvisation, And Music: Trends in Digital Socialized Music-Making'. Ph. D, The City University of New York.
- Bowker, Geoffrey C., and Susan Leigh Star. 1999. *Sorting Things Out: Classification and Its Consequences*. Cambridge, Mass.: MIT Press.
- Brown, Chris, John Bischoff, and Tim Perkis. 1996. 'Bringing Digital Music To Life'. *Computer Music Journal* 20 (2): 28. doi:10.2307/3681328.
- Cage, John. 1969. *Notations*. [New York]: [Something Else Press].
- Chadabe, Joel. 1984. 'Interactive Composing: An Overview'. *Computer Music Journal* 8 (1): 22. doi:10.2307/3679894.
- Chao, Wilber, Kuan-Ting Chen, and Yi-Shin Chen. 2014. 'Music Interaction on Mobile Phones'. in Ooi, Wei Tsang. *Advances In Multimedia Information Processing - PCM*: 153-162.
- Collins, Nick. 2003. 'Generative Music And Laptop Performance'. *Contemporary Music Review* 22 (4): 67-79. doi:10.1080/0749446032000156919.
- Collins, Nick, and Julio d' Escrivan Rincón. 2007. *The Cambridge Companion To Electronic Music*. Cambridge: Cambridge University Press.
- Cope, David. 2000. *The Algorithmic Composer.* Madison, Wis.: A-R Editions.
- Cope, David. 1996. *Experiments In Musical Intelligence. Madison*, Wis.: A-R Editions.
- Cope, David. 1977. New Music Composition. New York: Schirmer Books.
- Couprie, Pierre. 2004. 'Graphical Representation: An Analytical And Publication Tool For Electroacoustic Music'. *Organised Sound 9* (01). doi:10.1017/s1355771804000147.
- Dahl, Luke, and Ge Wang. 2010. 'Physical metaphors in designing mobile music performance'. In *Proceedings of the International Conference on New Interfaces for Muiscal Expression, 2010.*
- Dean, R. T. 2009. *The Oxford Handbook Of Computer Music*. Oxford: Oxford University Press.
- Dudas, Richard. 2010. '"Comprovisation": The Various Facets Of Composed Improvisation Within Interactive Performance Systems'. *Leonardo Music Journal* 20: 29-31. doi:10.1162/lmj_a_00009.
- Eno, Brian. 1978. *Ambient 1: Music for Airports*. Virgin Records Ltd. 1978.

- Essl, Georg, and Michael Rohs. 2009. 'Interactivity For Mobile Music-Making'. *Organised Sound* 14 (02): 197. doi:10.1017/s1355771809000302.
- Eyerman, Ron., and Andrew Jamison. 1998. *Music and Social Movements: Mobilizing Traditions in the Twentieth Century*. Cambridge, [England] ; New York: Cambridge University Press.
- Farnell, Andy. 2010. *Designing Sound*. Cambridge, Mass.: MIT Press.
- Françoise, Jules, Norbert Schnell, Riccardo Borghesi, and Frédéric Bevilacqua. 'MaD'. *interactions* 22.3: 14-15. doi:10.1145/2754894.
- Gaye, Layla, Ramia Mazé, and Lars Holmquist. 2003. 'Sonic City: The Urban Environment as a Musical Interface'. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression(NIME-03):* 109-115. Montreal, Canada, 2003.
- Gentile, Emilio. 2003. *The Struggle For Modernity: Nationalism, Futurism, and Fascism*. Westport, Conn.: Praeger.
- Goldstein, Paul, and P. B Hugenholtz. 2013. *International Copyright: Principles, Law, and Practice*. 3rd ed. Oxford ; New York: Oxford University Press.
- Her J., and Hamlyn J. 'Meaningful Engagement: Computer-Based Interactive Media Arts in Public Space'. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Arts and Technology, 2010*: 17-24.
- Huron, David. 2002. 'A six-component theory of auditory-evoked emotion'. In *Proceedings of the 7th International Conference on Music Perception and Cognition. Sydney, Australia, 2002.*
- Jalāl al-Dīn Rūmī, and Coleman Barks. 2003. *Rumi*. New York, NY: HarperOne, HaperCollins.
- Kanach, Sharon E. 2012. *Xenakis Matters: Contexts, Processes, Applications*. Hillsdale, N.Y.: Pendragon Press.
- Kane, Brian. 2015. 'Brian Kane'. Browsebriankane.Com. http://browsebriankane.com.
- Kane, Brian. *Sound Unseen: Acousmatic Sound in Theory and Practice*. New York: Oxford University Press, 2014.
- Kent, Allen, and Jack Belzer. 1994. Encyclopedia Of Computer Science And Technology. New York, NY [u.a.]: Dekker.
- Kirn, Peter. 2012. 'Libpd » About'. Libpd.Cc. http://libpd.cc/about/.
- Lansky, Paul. 1990. 'A View From The Bus: When Machines Make Music'. *Perspectives Of New Music* 28 (2): 102. doi:10.2307/833010.
- Lewis, G. 1994. 'Personal Paper and conversation'. In Winkler, Todd. 1998. *Composing Interactive Music.* Cambridge, Mass.: MIT Press.
- Lyndon-Gee, Cristopher. 2007. in Varèse, Edgard, et al. *VARESE: Orchestral Works, Vol. 2 - Ameriques / Equatorial / Nocturnal / Ionisation.* Naxos Digital Services US Inc. 2008.
- Magnusson, Thor. 2014. 'Scoring With Code: Composing With Algorithmic Notation'. *Organised Sound* 19 (03): 268-275. doi:10.1017/s1355771814000259.
- Manaris, Bill, and Andrew R Brown. 2014. *Making Music With Computers*. Boca Raton, FL: CRC Press.
- Manning, Peter. 2013. *Electronic And Computer Music*. New York: Oxford University Press.

- Mathews, M. V. 1963. 'The Digital Computer As A Musical Instrument'. *Science* 142 (3592): 553-557. doi:10.1126/science.142.3592.553.
- Mathews, M. V., and Arthur Roberts. 1970. 'The Technology Of Computer Music'. *Phys. Today* 23 (8): 60. doi:10.1063/1.3022299.
- Miranda, Eduardo Reck, and Marcelo M Wanderley. 2006. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Middleton, Wis.: A-R Editions.
- Morgan, Robert P. 1980. 'Musical Time/Musical Space'. CRIT INQUIRY 6 (3): 527. doi: 10.1086/448063.
- Muller, L., E. Edmonds, and M. Connell. 2006. 'Living Laboratories For Interactive Art'. Codesign 2 (4): 195-207. doi:10.1080/15710880601008109.
- Nierhaus, Gerhard. 2009. *Algorithmic Composition: Paradigms of Automated Music Generation.* Wien ; New York: Springer.
- Peery, J. Craig, and Irene W. Peery. 1986. 'Effects Of Exposure To Classical Music On The Musical Preferences Of Preschool Children'. Journal Of Research In Music Education 34 (1): 24. doi:10.2307/3344795.
- Peirce, Charles S, and Justus Buchler. 1955. *Philosophical Writings Of Peirce*. New York: Dover Publications.
- Pratella, Ballila. 'Technical Manifesto of Futurist Music'*. I*n Manning, Peter. 2003. *Electronic and Computer Music.* New York: Oxford University Press, 2013: 6.
- Puckette, Miller. 'Using pd as a score language'. In *Proceedings of International Computer Music Conference. September 2002a*: 184-187.
- Puckette, Miller. 1991. 'Something Digital'. *Computer Music Journal* 15 (4): 65. doi: 10.2307/3681075.
- Puckette, Miller. 2007. *The Theory And Technique Of Electronic Music*. Hackensack, NJ: World Scientific Publishing Co.
- Reighley, Kurt B. 2000. *Looking For The Perfect Beat*. New York: Pocket Books.
- Rokeby, David. 1996. 'Lecture For "Info Art", Kwangju Biennale'. Davidrokeby.Com. http://www.davidrokeby.com/install.html
- Roads, Curtis, and John Strawn. 1996. *The Computer Music Tutorial*. Cambridge (Mass.): The MIT Press.
- Roads, Curtis. 1997. *Musical Signal Processing*. Lisse [Netherlands] ; Exton, PA: Swets & Zeitlinger.
- Rowe, Robert. 1993. *Interactive Music Systems*. Cambridge, Mass.: MIT Press.
- Sacks, Oliver W. 2007. *Musicophilia: Tales of Music and the Brain*. New York: Alfred A. Knopf.
- Schaefer, John. 1987. *New Sounds: A listener's guide to music.* New York: Harper & Row.
- Seeger, Charles. 1958. 'Prescriptive and Descriptive Music-Writing'. *Musical Quarterly* XLIV (2): 184-195. doi:10.1093/mq/xliv.2.184-195.
- Sica, G. 1997. 'Notations and interfaces for musical signal processing'. In Roads, Curtis. 1997. *Musical Signal Processing*. Lisse [Netherlands]: Swets & Zeitlinger.

- Stockhausen, Karlheinz. 1968. *Kontakte: Elektronische Musik, Nr.12*. London: Universal Edition.
- Steiner, Hans-Cristoph. 2010. 'Solitude'. At.Or.At. http://at.or.at/hans/solitude/.
- Tanaka, Atau, and Petra Gemeinboeck, 2006. 'A Framework for Spatial Interaction in Locative Media'. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME06), Paris, France, 2006:* 26-30.
- Thompson, Robert S. 2014. 'Eric Lyon: Designing Audio Objects For Max/MSP And Pd'. *Computer Music Journal* 38 (2): 80-84. doi:10.1162/comj_r_00243.
- Tiercelin, C. 2002. 'La Sémiotique philosophique de Charles Sanders Peirce'. In Hénault, Anne, David Savan, Drude von der Fehr, and Jean Claude Pariente. 2002. *Questions De Sémiotique*. Paris: Presses universitaires de France.
- Trochimczyk, Maja. 2001. 'From Circles To Nets: On The Signification Of Spatial Sound Imagery In New Music'. *Computer Music Journal* 25 (4): 39-56. doi: 10.1162/014892601528l5288.
- Trythall, Gilbert. 1973. *Principles And Practice Of Electronic Music*. New York: Grosset & Dunlap.
- Varèse, Edgar. 'Newyork Telegraph Newspaper, 16 May 1916'. In Manning, Peter. 2013. *Electronic and Computer Music.* New York: Oxford University Press: 7.
- Wehinger, Rainer, and György Ligeti. 1970. *Ligeti: Artikulation; Elektronische Musik*. Mainz: B. Schott's Söhne.
- Wilson, Scott, Nick (Nicholas) Collins, and David Cottle. 2011. *The SuperCollider Book*. Cambridge, Mass.: MIT Press.
- Winkler, Todd. 2000. 'Audience participation and response in movement-sensing installations'. In *Proceedings of International Computer Music Conference, 2000.* International Symposium on Electronic Art (ISEA) Paris, France.
- Winkler, Todd. 1998. *Composing Interactive Music*. Cambridge, Mass.: MIT Press.
- Whalley, Ian. 2014. 'GNMISS: A Scoring System For Internet2 Electroacoustic Music'. *Organised Sound* 19 (03): 244-259. doi:10.1017/s1355771814000235.
- Xenakis, Iannis. 1992. *Formalized Music*. Stuyvesant (N.Y.): Pendragon.
- Zicarelli, David. 1987. 'M And Jam Factory'. *Computer Music Journal* 11 (4): 13. doi: 10.2307/3680237.
- Zicarelli, David. 2002. 'How I Learned To Love A Program That Does Nothing'. *Computer Music Journal* 26 (4): 44-51. doi:10.1162/014892602320991365.

Discography

- Behrman, David., Maggi Payne, Arthur Stidfole, and David R Gibson. 1978. *On the Other Ocean: Figure in a Clearing*. [New York, N.Y.]: Lovely Music.
- Cage, John, 2014. *Early Electronic and Tape Music (Cage)*. Hong Kong: Naxos Digital Services US Inc.
- Carlos, Wendy., et al. 1999. *Switched-on Boxed Set*. Minneapolis, MN: East Side Digital
- Chadabe, Joel. 1981. *Rhythms. (Jan Willimas, percussionist.)* [New York, N.Y.]: Lovely Music, Ltd. LP 1301.
- Cope, David, et al. 1994. *Bach By Design: Computer Composed Music : Experiments in Musical Intelligence*. [California]: Centaur Records.
- Eno, Brian. 1978. *Ambient 1: Music for Airports*. Virgin Records Ltd. 1978.
- Graugaard, Lars, et al. 1996. *Lars Graugaard Plays Pieces for Flute and ISPW*. [Denmark].
- Hiller, Lejaren, et al. 1989. *Computer Music Retrospective*. Mainz, Germany: Wergo.
- Lancino, Thierry., et al. 1995. *Thierry Lancino, Tamas Ungvary*. Mainz, West Germany: Wergo.
- Ligeti, G. 1984. *Ligeti*. Mainz, W. Germany: Wergo.
- Pierce, John Robinson., Max Mathews, David Lewin, James Tenny, S. D. Speeth, and Newman Guttman. 1983. *Music from Mathematics Played by IBM 7090 Computer and Digital to Sound Transducer*. DECCA Records.
- Steiner, Hans-Cristoph. 2004. *Solitude*. At.or.at. N.p. http://at.or.at/hans/solitude.
- Stockhausen, Karlheinz, David Tudor, Christoph Caskel, and Gottfried Michael Koenig. 1992. *Kontakte: Für Elektronische Klänge, Klavier Und Schlagzeug*. Mainz, Germany: Wergo.
- Subotnick, Morton., Joan La Barbara. 1999. *The Electronic Works*. [S.l.]: Mode Records.
- Varèse, Edgard, et al. 2008. *VARESE: Orchestral Works, Vol. 2 - Ameriques / Equatorial / Nocturnal / Ionisation.* Naxos Digital Services US Inc.

Appendix A: Quick Introduction to Pd

(From the book: *The theory and technique of electronic music* by Miller Puckette)

Pd documents are called *patches*. They correspond roughly to the boxes in the abstract block diagrams shown earlier in this chapter, but in detail they are quite different, because Pd is an implementation environment, not a specification language.

A Pd patch, such as the ones shown in Figure I, consists of a collection of *boxes* connected in a network. The border of a box tells you how its text is interpreted and how the box functions. In part (a) of the figure we see three types of boxes. From top to bottom they are:

- a *message box*. Message boxes, with a flag-shaped border, interpret the text as a message to send whenever the box is activated (by an incoming message or with a pointing device). The message in this case consists simply of the number ``21".
- an *object box*. Object boxes have a rectangular border; they interpret the text to create objects when you load a patch. Object boxes may hold hundreds of different classes of objects--including oscillators, envelope generators, and other signal processing modules to be introduced later--depending on the text inside. In this example, the box holds an adder. In most Pd patches, the majority of boxes are of type ``object". The first word typed into an object box specifies its *class*, which in this case is just ``+". Any additional (blank-space-separated) words appearing in the box are called *creation arguments*, which specify the initial state of the object when it is created.

- a *number box*. Number boxes are a particular type of *GUI box*. Others include push buttons and toggle switches; these will come up later in the examples. The number box has a punched-card-shaped border, with a nick out of its top right corner. Whereas the appearance of an object or message box is fixed when a patch is running, a number box's contents (the text) changes to reflect the current value held by the box. You can also use a number box as a control by clicking and dragging up and down, or by typing values in it.

In Figure I (part a) the message box, when clicked, sends the message ``21" to an object box which adds 13 to it. The lines connecting the boxes carry data from one box to the next; outputs of boxes are on the bottom and inputs on top.

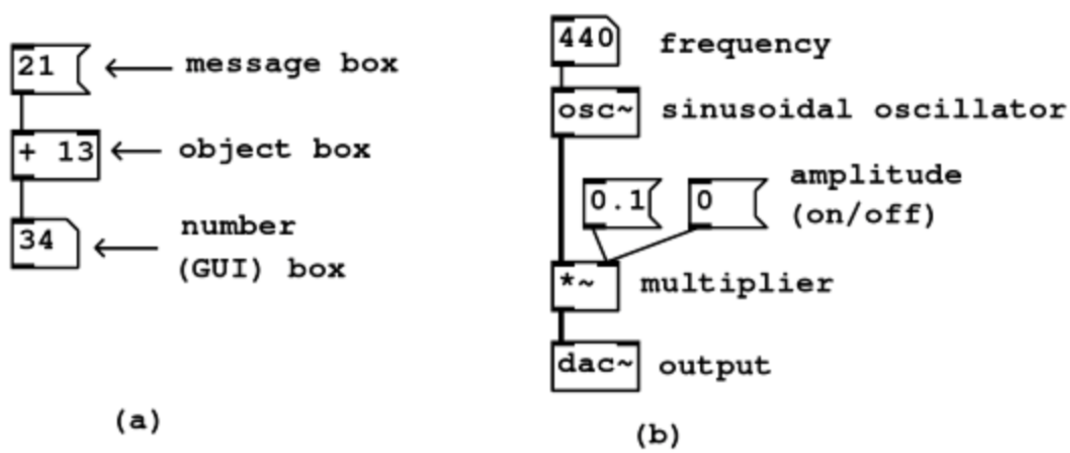


Figure I. (a) three types of boxes in Pd (message, object, and GUI); (b) a simple patch to output a sinusoid.

Figure *l* (part b) shows a Pd patch which makes a sinusoid with controllable frequency and amplitude. The connecting patch lines are of two types here; the thin ones are for carrying sporadic *messages*, and the thicker ones (connecting the oscillator, the multiplier, and the output dac~ object) carry digital audio signals. Since Pd is a real-time program, the audio signals flow in a continuous stream. On the other hand, the sporadic messages appear at specific but possibly unpredictable instants in time.

Whether a connection carries messages or signals depends on the box the connection comes from; so, for instance, the + object outputs messages, but the *~ object outputs a signal. The inputs of a given object may or may not accept signals (but they always accept messages, even if only to convert them to signals). As a convention, object boxes with signal inputs or outputs are all named with a trailing tilde (``~") as in ``*~" and ``osc~".

Appendix B: Important elements of electronic music

(From the book: *Principles and Practice of Electronic Music* by Gillbert Trythall)

ELEMENTS	POSSIBLE SPECIFICATIONS
<u>SOUND SOURCES</u>	
Waveforms: • sine • sawtooth • rectangular - pulse (duty cycle) square • triangle	Exact frequency, from the frequency counter Approximate frequency, by pitch reference General registers, with reference to high, medium, low (and shadings of those terms)
Noise: • white noise • colored noise	Center frequency and bandwidth
Silence • unrecorded tape • leader tape	
Music cocrète	Sources of natural sounds
<u>DURATIONS</u>	
• plotted against time: • plotted against space:	Beginnings and ending points specified in seconds, or in pulses In inches of tape, in centimeters of tape

<u>DYNAMICS</u>	
• plotted against time:	Meter readings on studio or tape recorder equipment Traditional notation (ppp to fff)
	Notation with descriptive words (*loud, soft,* etc.) Envelope attack and decay times with maximum intensities, and description of curves (linear, exponential, other)
<u>ELECTRO-MECHANICAL EFFECTS</u>	
• plotted against time:	Reverberation and echo duration loudness spring, tape, or other unit tape speed and tape echo Speed change half double variable Mixing input sources output routing

ELECTRONIC EFFECTS	
• plotted against time:	Filtering frequency and type: high-pass; low-pass; bandpass; band-reject Ring Modulation input frequencies output frequencies Frequency Modulation vibrato frequency of control signal frequency of modulated signal glissandi
	Amplitude Modulation envelopes rise, steady rate, decay times tremolo frequency of control signal frequency of modulated signal Timbre Modulation filter used frequency of control signal frequency of modulated signal
TAPE EFFECTS	
• plotted against time:	Tape reversal Looping length playback speed Magnetic particle removal razor blade acetone-moistened cotton swab