

Mobile music: a portfolio of works exploring adaptive music generation in embedded and mobile devices.

Elise Marie Plans

Royal Holloway, University of London

Submitted as part of the degree of

PhD in Composition 2021

Declaration of Authorship

I, Elise Plans, hereby declare that this thesis and the work presented in it is entirely my own.
Where I have consulted the work of others, this is always clearly stated.

Signed: Elise Plans Date: 25th August 2021

Statement of Authorship

The Breathing Stone

In the Chelsea & Westminster version of the Breathing Stone, the audio engine was created by Robert M Thomas, and the 'packs' were composed by Elise Plans using Thomas's audio engine.

The Bristol Breathing Stones used a version of Thomas's audio engine that had been modified by Elise Plans.

The Breathing Stone was conceived within the REACT Hub Sandbox programme, funded by the AHRC. After this initial funding, a private company, BioBeats continued the project for the Chelsea & Westminster Hospital and the REACT Festival in Bristol in 2015.

Ominator

The visuals and user interface for the *Ominator* app were coded by David Plans.

Bespia

The visuals for the *Bespia* app were coded by Joel Barker and David Plans.

Nautilus

The user interface was coded by David Plans.

Montague

The visuals and user interface were coded by David Plans.

Cantoo

The visuals were coded by David Plans.

The UI (user interaction) and UX (user experience) for *Ominator*, *Nautilus*, *Montague*, *Cantoo* and *Elerem* were coded by Elise Plans.

Other Funding

To assist with this research, a Francis Chagrin Award organised by Sound and Music¹ allowed me to purchase a test device which was also used for submission to the examiners.

¹ Sound and Music is a national organisation for new music in the UK <https://soundandmusic.org/>

Acknowledgements:

Firstly I must thank my PhD supervisor, Nina Whiteman, for her endless patience and incredible skill at pinpointing exactly where improvements could be made, and also for being a calming voice throughout my doctoral student experience. I learned an incredible amount from listening to Nina presenting her work, and our supervision sessions together.

My previous supervisor, Brian Lock, started me on this journey and was always encouraging without being afraid to tell me when something wasn't right. I am grateful for his wisdom and insight into composition that he shared through our supervision sessions.

I am indebted to Robert M Thomas and Davide Morelli, with whom I had the privilege to work at BioBeats. Numerous conversations with them about music, and more specifically adaptive music, were the original inspiration for this thesis.

Thank you also to many other members of the Royal Holloway collective: Shzr Ee Tan for being a tour de force that lifts up everyone around her, Tom Parkinson who consistently managed to make me feel like I knew what I was doing, and somehow made everyone else believe it too! Tim Summers, for always finding time to talk about interesting things, and imparting a teaching style that I will forever be trying to emulate. Nathan James Dearden always had a cheerful word to share, no matter which stressful situation was thrown at him that week! Shanika Ranasinghe, Lucy Harrison, Tom Wilson and the rest of my RHUL PG cohort for all the important conversations held in corridors, cafés, train stations, and on social media through which I felt supported in myriad ways as we navigated the doctoral highs and lows.

I also wouldn't have been able to do this PhD without the seeds sown by Simon Waters and Monty Adkins at the University of East Anglia, so I am also grateful to them for dragging me kicking and screaming through my undergraduate and MMus degrees!

I must also thank my close friends, Nick, Al and Sarah for carrying me through with their love and succour, always ready with a party or bridge session to remind me about the important things in life.

My sons Thomas and Ollie were only eight and seven when I started this, and have managed to reach near-adulthood without much intervention from me. They have been very patient and non-judgemental, even when there were more ready meals and junk food for dinner than was strictly healthy for growing musculoskeletal bodies.

My parents provided me with both emotional and practical support. In the early days, they stepped in without hesitation to provide childcare and later on, there was always a beer in the fridge and an attentive ear for me to offload my stress and self-doubt.

And finally, my partner David. The list of ways in which he helped and supported me through this would be a weighty tome, but most importantly I am grateful to him for stubbornly continuing to believe in me in ways I cannot even believe in myself.

Abstract

Generative and procedural music is well-established in various fields (games, installations, experimental performance) of computer-based compositional practice, but only in the last decade and a half has the mobile device become available as a suitable platform for its reproduction.

For this research, I aimed to create a portfolio of generative pieces firstly for an embedded mobile device called the “Breathing Stone” then as apps for mobile devices. Some of the pieces, *Montague* and *Cantoo* work as standalone procedural works, the rest, *Ominator*, *Bespia*, *Nautilus* and *Elerem* take varying degrees of input from sensors on the mobile device to modify musical and structural parameters in the compositions.

The resulting portfolio presents a way of looking at composing for the medium of the mobile device as distinct from trying to build or compose within a standardised framework, as each composition is created from its own unique audio engine. The accompanying commentary explores the existing body of work in this field, as well as themes of well-being and ‘dynamic’ music where a piece of music is designed to sound different each time it is heard.

Although a portfolio of mobile works was successfully produced, several difficulties and limitations were found when composing for the mobile device as a medium. Several of these impediments are discussed in the concluding paragraph which hopes to enlighten future composers in this field attempting similar compositions in the mobile device space.

Portfolio of Works

1. *Breathing Stone* (At the Chelsea and Westminster Hospital SAL Unit) June 2015

Capture of each of the seven packs can be found here:

Pack 1: <https://soundcloud.com/eplans/stonepack1>

Pack 2: <https://soundcloud.com/eplans/stonepack2>

Pack 3: <https://soundcloud.com/eplans/stonepack3>

Pack 4: <https://soundcloud.com/eplans/stonepack4>

Pack 5: <https://soundcloud.com/eplans/stonepack5>

Pack 6: <https://soundcloud.com/eplans/stonepack6>

Pack 7: <https://soundcloud.com/eplans/stonepack7>

(Please note, these recordings were taken from the original Pure Data patches, designed to run at a sample rate of 22.05kHz in order to work smoothly on a Raspberry Pi Compute module, so the audio quality will not be comparable to the more typical 44.1 kHz or 48 kHz.)

2. *Breathing Stone* (Bristol Rooms Exhibition) November 2015

Video capture of the exhibition can be found here:

<https://youtu.be/xwWUdOQq1cg?t=55> (starting at 55 seconds).

Instructions on how to use the accompanying device and following apps is included as Appendix 1, with a printed copy included in the iPhone boxing.

3. Ominator (iOS App) October 2017

Installed on accompanying device and available in the App Store:

<https://apps.apple.com/gb/app/ominator/id1131534867>

Please note, Ominator will only work with wired headphones, not wireless ones.

4. Bepia (iOS App) October 2019

Installed on accompanying device and available in the App Store:

<https://apps.apple.com/gb/app/bepia/id1479330865>

5. Nautilus (iOS App) May 2020

Installed on accompanying device and available in the App Store:

<https://apps.apple.com/gb/app/nautilus-by-elise-plans/id1512587880>

6. Montague (iOS App) February 2021

Installed on accompanying device and available in the App Store:

<https://apps.apple.com/gb/app/montague-by-elise-plans/id1539495463>

7. Cantoo (iOS App) (in App Store Review) 2021

Installed on accompanying device and available in the App Store:

<https://apps.apple.com/gb/app/cantoo-by-elise-plans/id1530407161>

8. Elerem (iOS App) (unpublished) 2021

Installed on accompanying device.

Suggested Listening Times

App	Time	Notes
<i>Breathing Stone (C&W)</i>	14m	7 pieces 2 minutes each.
<i>Breathing Stone (Bristol)</i>	2m18s	YouTube video.
<i>Ominator</i>	5m	
<i>Bespia</i>	6m	
<i>Nautilus</i>	30m	
<i>Montague</i>	45m - 1 hr	<i>Montague I</i> = 5 minutes approximately <i>Montague II</i> = each 'piece' is between 7 and 16 minutes. <i>Montague III</i> = 8 minutes approximately.
<i>Cantoo</i>	12m	
<i>Elerem</i>	20m	Listen while moving about.
TOTAL	122m*	*Actual time can vary significantly.

Table of Contents:

[Declaration of Authorship](#)

[Acknowledgements:](#)

[Abstract](#)

[Portfolio of Works](#)

- [1. Breathing Stone \(At the Chelsea and Westminster Hospital SAL Unit\) June 2015](#)
- [2. Breathing Stone \(Bristol Rooms Exhibition\) November 2015](#)
- [3. Ominator \(iOS App\) October 2017](#)
- [4. Bospia \(iOS App\) October 2019](#)
- [5. Nautilus \(iOS App\) May 2020](#)
- [6. Montague \(iOS App\) February 2021](#)
- [7. Cantoo \(iOS App\) \(in App Store Review\) 2021](#)
- [8. Elerem \(iOS App\) \(unpublished\) 2021](#)

[Suggested Listening Times](#)

[Table of Contents:](#)

[Definition of Terms and Glossary](#)

[Definition of terms for the purposes of this thesis:](#)

[Glossary of terms:](#)

[Introduction](#)

[Background](#)

[Rationale](#)

[Mobile Music](#)

[Dissemination of Generative Music](#)

[Scope of Thesis](#)

[App Categories:](#)

[Generative Music Apps](#)

[Tools for Making Music](#)

[Enhancement Apps](#)

[Games](#)

[Well-Being Apps](#)

[Promotional Apps](#)

[Writing Music Vs Writing Frameworks](#)

[Research Aims](#)

[Chapter 1 - Contextualisation](#)

[Procedural and Electronic Music](#)

[Interactive Music on Mobile Devices](#)

[Human Computer Interaction](#)

[Data Sonification](#)

[SUMMARY](#)

[Chapter 2 - Overview of Compositional Methods](#)

[Foundations of Compositional Approach](#)

[Compositional Approaches of Game Music Composers](#)

[The Medium affects the Output](#)

['Creating' Instruments](#)

[Designing Compositional Processes](#)

[Common Abstractions:](#)

[Humaniser](#)

[Mixer](#)

[Drum Machine](#)

[Compositional Process](#)

[Technical Equipment](#)

[Chapter 3 - Compositions for the Breathing Stone \(June 2015\)](#)

[Chelsea & Westminster Hospital Stone](#)

[Piece for Two Breathing Stones and a Drone](#)

[Chapter 4 - Interactive Apps](#)

[Ominator: a Generative Sonic/Visual App](#)

[Bespia](#)

[Nautilus](#)

[Instruments:](#)

[Wood](#)

[Wave](#)

[PitchPad](#)

[Manipulation of materials:](#)

[micRecord](#)

[tapDel](#)

[reEcho](#)

[reWave](#)

[Empty](#)

[Discussion and future development](#)

[Elerem](#)

[Elerem - Composition](#)

[Arpeggiator](#)

[Vocalisations:](#)

[Mallet Sounds:](#)

[Limitations](#)

[Chapter 5 - Procedural Works](#)

[Montague](#)

[Montague I](#)

[Montague II](#)

[Montague III](#)

[Cantoo](#)

[Process](#)

[Chapter 6 - Conclusion](#)

[Technical limitations influencing the work](#)

- [1. Processing power of smartphones.](#)
- [2. Size of Apps](#)
- [3. Sensor data from smartphones are limited by their APIs](#)
- [4. Limited programming skills of a composer.](#)
- [5. Limitations of Pure Data when embedded as libpd](#)
- [6. Limitations of procedural composition when process-oriented](#)
- [7. Apple's own aesthetics limit possibilities](#)
- [8. Coding frameworks change](#)

[Final Summary](#)

[Bibliography](#)

[Appendix 1: Instructions for Accessing Portfolio Apps](#)

[iPhone and App Instructions](#)

[App Instructions](#)

[Ominator](#)

[Elerem](#)

[Bespia](#)

[Appendix 2: Score and Instructions for the Breathing Stone.](#)

[Appendix 3: Diagrams of Algorithms used in Elerem Mallet Instrument](#)

[Melody Fragment 1:](#)

[Melody Fragment 2:](#)

[Melody Fragment 3](#)

[Dyad Fragment 1:](#)

[Dyad Fragment 2:](#)

[Appendix 4: Example of microtonal calculations and playback in Pure Data](#)

[Appendix 5: Dataflow from Montague III](#)

- [1. Equal Temperament Player](#)
- [2. Microtonal Scale 13-ET](#)
- [3. Microtonal Scale 16-ET](#)
- [4. Chords](#)
- [5. Bass notes](#)
- [6. Countermelody](#)

Definition of Terms and Glossary

Definition of terms for the purposes of this thesis:

Procedural music: music generated by processes. For the context of this thesis, I will use the term 'procedural' music to include algorithmic music.

Adaptive music: music that takes inputs or cues to influence and modify music, and/or adapts to listener behaviour.

Generative music: encompassing both of the above, music that is generated in real-time, not pre-composed.

Interactive music: also incorporates adaptive music but tends to be used more for installations, or where deliberate (as opposed to passive) actions are required to affect the music outcomes.

Listener: in the context of 'interactive listening' where the boundaries of user/listener/player are blurred, standard definitions for the word 'listener' can become inadequate. In this thesis, I have used the words 'listener' and 'user' almost interchangeably, with a slight emphasis on more utilitarian contexts for 'user' and more musically-orientated contexts for 'listener'. I felt the word 'player' would be confusing as this is often to describe sound reproduction algorithms in my dataflow.

Glossary of terms:

Abstraction: a Pure Data patch that is saved separately but called within a parent patch. This enables the patch to be used in different systems.

AIFF and WAV: uncompressed audio file formats.

Album app or app album: in many of the texts consulted for this thesis, this term is not standardised. I have chosen to use 'album app' unhyphenated, though where I have quoted others I have left it as written, for example, 'app album'.

API: Application Programming Interface, a set of protocols that allow programs to communicate with each other.

Audio engine: a system that produces music or sounds programmatically.

Dataflow: In the context of this thesis this is used to mean the organisation of data through a series of computations, specifically in the audio programming environments such as Pure Data or Max/MSP.

DAW: Digital Audio Workstation.

DSP: Digital Signal Processing.

FLAC: Free Lossless Audio Codec, an audio file format that uses lossless compression.

FMOD: a group of software products from Firelight Technologies often used to develop game audio and interactive audio.²

GUI: Graphical User Interface, allows a user to interact with a program or app.

Libpd: the embeddable library version of Pure Data.

Max/MSP: visual coding software maintained by Cycling '74.³

MIDI: Musical Instrument Digital Interface, a digital communication protocol for musical parameter exchange. Where I have referred to notes as MIDI numbers, I have used numeric values as this is neater than writing out words for lists of notes.

Modulo: a mathematical operation which gives the remainder when two numbers are divided (for example six modulo four would give the number 2).

Objective C: a programming language often used to build iOS applications.

² <https://www.fmod.com/>

³ <https://cycling74.com/>

OGG: refers to 'Ogg Vorbis' which is a compressed audio file format.

Pad: a sustained sound usually made by combination of sounds to provide a background texture typically generated by a synthesiser.

PD: Pure Data the visual coding software developed by Miller Puckette.⁴

Spigot: In this context 'spigot' refers to an object in Pure Data which, like a tap, uses a binary switch to allow or stop a stream of data passing through.

Subpatch: a Pure Data patch containing code that is used within a parent patch.

Swift: a programming language often used to build iOS applications.

Synth: an instrument where the sound is made by the manipulation of oscillators.

Timebase: a universal clock built in Pure Data (in the context of this thesis) to synchronise events across PD patches.

UI: User Interface elements that deliver the user experience, such as the control panel for the user to interact with an app, a slider, or a number box to relay information.

UX: User Experience, a design term that relates to the mechanics of how an app will be used

Vanilla: the original version of a programme without any additional components or extras.

WWise: Wave Works Interactive Sound Engine is software developed by Audiokinetic that is often used to develop game audio and interactive audio.⁵

xCode: a program in Mac OS used to build other applications.⁶

⁴ <https://puredata.info/>

⁵ <https://www.audiokinetic.com/products/wwise/>

⁶ <https://developer.apple.com/xcode/>

Introduction

Background

My compositional practice is driven by the design and building of processes. I design processes to realise compositional ideas in dataflow programming environments such as PureData or Max/MSP. I use Pure Data components in the same way that an instrumental composer might employ the forces of a soloist, an orchestra or an ensemble to realise an idea, whereby I map processes and the flow of data within and between them as a score.

There is a legacy of process-based music(s) particularly in the 20th century when composers such as Xenakis, Stockhausen, and Reich produced works that used processes to build whole compositions. In some of their work, the process itself was paramount, for example, in experimental pieces where music emerged from the process itself, without a prior sound or musical shape chosen by the composer.⁷ In other pieces, the process was used to realise a predetermined sound or compositional idea. Xenakis often curated the use of process in his works, suggesting that he was trying to realise specific musical ideas, rather than let the music be dictated by the algorithm:

We have seen that, in his musical implementation of cellular automata, Xenakis introduces a lot of changes, and “intervenes” consistently. This is always the case when he uses formal procedures: stochastics, symbolic logic, game theory, group theory, sieve theory, dynamic stochastic synthesis. All the Xenakis specialists, when working in the field of the concrete analysis of works using formal procedures, have noted that the composer takes liberties with the formal models, and introduces “licences”, “gaps” (écarts in French). In other terms, his use of formalization is mediated through manual interventions.⁸

⁷ For example, Stockhausen, ‘Plus-Minus’ (London, 1963).

⁸ Makis Solomos, ‘Cellular Automata in Xenakis’s Music. Theory and Practice’, *The International Symposium Iannis Xenakis*, 2006 <<http://hal.archives-ouvertes.fr/hal-00770141/>>.

Like Xenakis, I do not use process for the sake of process but rather to achieve a compositional vision that precedes the process design, and to allow for different possible outcomes when a piece of music is played more than once.

I use Pure Data for process development as I am familiar with its syntax and can quickly prototype algorithms. However, PD patches are only typically reproducible within the software in which they are created, making public dissemination of these works difficult, requiring the installation of compiled software in order to listen to a piece of music. This limits potential distribution channels to other PD users or to installation work, not to mention also needing to include any sound files needed in the transfer, which can often be sizable. The dissemination issue has dogged generative music, as highlighted by Alex Bainter in his *Medium* article: 'In 1996, only a die-hard Eno fan or a total nerd would probably have bothered to pop a floppy disk into their computer to listen to generative music. Now, generative music systems are just a click away, right on the same device we're already using to listen to recorded music.'⁹

Smartphone applications are easy to download and must be approved by Apple to be made available in the App Store, making them a trusted method of delivery. They are also disposable: it is easy to install and then delete an app that is not wanted. As a result, smartphone applications have become a useful method for disseminating process music composed in PD.

This method of distribution, like any media, comes with limitations and demands in the same way that a seven-inch single could only fit a limited number of minutes of recorded music in its vinyl grooves. These limitations and demands have shaped the music in my

⁹ Alex Bainter, 'The Future of Generative Music: Why the World Is Finally Ready for Endless Music Systems', *Medium*, 2019 <<https://medium.com/@alexbainter/the-future-of-generative-music-e19b6722deb2>> [accessed 14 August 2021].

portfolio in ways that will be described in more detail in the following chapters. They include the limits of processing power on a mobile or embedded device, and meeting demands of the reviewers within Apple, as they choose whether an app is appropriate to Apple's own values and contracts with their users, values which a mobile music composer must meet notwithstanding compositional intention.

Compositional limitations can also be useful and encourage more in-depth investigations into the possibilities of a sound, and how it can be varied. A brief example in my own work is trying to make the most of the limitations of app size, by finding ways to exploit a sound file: playing them at different speeds for example (in *Nautilus*, *Montague*, and *Elerem*) or looping them to produce a continuous sound (*Elerem*, *Nautilus*).

Rationale

Early in 2012, I had the opportunity to work with a startup, BioBeats,¹⁰ building a mobile application for mental health. Their initial work focused on cardiovascular biofeedback (generating music out of heartbeat signals captured from the camera on the phone). Later versions guided the user through biofeedback breathing exercises, designed to relieve stress. I was initially tasked with writing music that reacted to cardiovascular data, and later on (building on work by composers Robert M Thomas¹¹ and Joe Hyde¹²) showing whether the user was successfully breathing diaphragmatically (from the belly). The music had to become more complex rhythmically and timbrally if the user succeeded at breathing deeply, and simpler and duller if not. This concept was then developed into a device called the *Breathing Stone* through support and funding from the AHRC and React Hub in Bristol.¹³

¹⁰ BioBeats Ltd was a startup who created therapeutic apps for health and well-being, recently merged with digital health company, Huma: <https://huma.com/>

¹¹ Robert Thomas is an Adaptive Music Composer: <http://dizzybanjo.com/>

¹² Joe Hyde is a composer and Professor in Music at Bath Spa University <https://www.josephhyde.co.uk/>

¹³ REACT Hub: <http://www.react-hub.org.uk/projects/objects/breathing-stone/>

This device was capable of taking 3-point ECG, the data from which could be fed back into the music, creating biofeedback to enhance the experience. Several pieces written for the *Breathing Stone* have been included within my supporting portfolio. The *Breathing Stone* and mobile application work I did at BioBeats established many of the working methods that I then developed and honed throughout the course of this research. While the *Breathing Stone* had hard limits in terms of time and size (which will be discussed later), the early BioBeats apps generated free-form (although genre-based), endless music from smartphone-captured heartbeat data,¹⁴ and demanded that I build music-generation systems that assumed endless, continuous play for each 'patch' (each piece offered in the app).

Since this first foray into adaptive music for biofeedback, mobile phone technology opened up a range of data input possibilities that I could feed into adaptive music, several of which I have incorporated into the compositions for my portfolio. As well as investigating the use of generative music in these applications, I explore compositional methods for standalone pieces whose focus is simply to evolve on each listening. As such, I am not using algorithms to make experimental music, to see how the processes would play out, rather to generate many variations of a single piece. The influence of working within digital therapeutics can be felt throughout my work; all the pieces in this portfolio make assumptions that I originally encountered in therapeutic work: piece length is determined by listener choice and is therefore by default endless; since there isn't a determined length, the music must contain enough variety to be engaging, but cannot contain radical gestures that

¹⁴ Natalie Robehmed, 'An App That Makes Music From Your Heartbeat', *Forbes Magazine*, 2013 <<https://www.forbes.com/sites/natalierobehmed/2013/08/27/an-app-that-makes-music-from-your-heartbeat/?sh=3112d2a742eb>> [accessed 14 August 2021]; Tom Cheshire, 'BioBeats App Creates Generates Custom Music from Your Heart Rate', *Wired*, 2013 <<https://www.wired.co.uk/article/your-heart-is-a-dj>> [accessed 14 August 2021].

imply finality (in the process necessarily acquiring stylistic aspects of ambient and drone-based music); the presence and availability of sensors and user data imply function, that is, the music can (and one could argue, should) adapt to the listener's environment and her data in order to deepen the level of engagement with the music. This aligns with the 'phone as the new Walkman' that the Kyoto Generative Music Workshop identified as a new paradigm in 2011: 'We were particularly interested in the later approach that is able to connect the listener to environment, and consider that the iPhone could become the Walkman for generative music because of their mobility, programmability and default sensors.'¹⁵

Mobile Music

Generative and procedural music(s) have a long history but have only recently combined with the concept of 'mobile music' where the listener's environmental and physical domain are used by the composition's inner audio engine. Adaptive music (a term we will use here to describe music that is procedural/generative in nature but whose dynamics are influenced by listener/environmental variables), has emerged alongside the development of smartphone technology, whereby people carry a vast array of data-gathering sensors and significant processing power with them almost constantly. This technological paradigm is giving rise to music that can adapt to the listener's particular situation and location (or physical state or mood, and change as those variables evolve. In 2006 a report was published on the specific subject of 'Mobile Music' with information gathered from a series of workshops held since 2004 that encompassed the whole range of possibilities made available by smartphones: 'Mobile Music is a new field concerned with

¹⁵ T. Kaneko and K. Jo, 'Generative Music Workshop,' *2011 Second International Conference on Culture and Computing*, 2011, pp. 179-180.

musical interaction in mobile settings using portable technology. It goes beyond today's portable music players to include mobile music making, sharing and mixing.¹⁶ A fuller definition and contextualisation of mobile music and its relevance in my research is given in the [Research Aims](#) section below.

Dissemination of Generative Music

Previous listening paradigms have used physical media such as tapes or discs which could be inserted into a corresponding device to play music stored on the medium.

Digitisation, leading to standardised compression formats such as mp3, Ogg, and FLAC files has overtaken the use of physical media; music tracks now exist as files that can be played on computers, laptops, smartphones, or dedicated music players. Generative music, on the other hand, does not yet have a standardised system for dissemination:

Though interactive audio works with music as the primary focus do exist, most musicians and composers are still highly unlikely to release liquid music in the same way that they might release static music. One reason for this is that there is simply no standard way in which to do so, while another may be the lack of commercially successful precedents.¹⁷

Although this specific style of adaptive/interactive music dissemination is fairly recent (within the last two decades), there is a significant and emerging history of interactive music for mobile devices, with landmark works such as *Bloom*, *Biophilia*, *Fantom*, and *Reflections*¹⁸. Many of these works take advantage of the quickly expanding processing

¹⁶ Lalya Gaye and others, '2006: Mobile Music Technology: Report on an Emerging Community', in *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*, ed. by Alexander Refsum Jensenius and Michael J. Lyons (Cham, 2017), 253–65.

¹⁷ Keith Hennigan, 'Music in Liquid Forms: A Framework for the Creation of Reactive Music Recordings', ed. by Adams Martin And Simon (unpublished PhD, Trinity College, Dublin, 2018) <http://www.tara.tcd.ie/bitstream/handle/2262/83259/Music%20in%20Liquid%20Forms_KeithHenniganPhDThesis.pdf?sequence=1>. 10.

¹⁸ Brian Eno, Peter Chilvers and Opal Ltd, *Bloom*, 2008 <<https://itunes.apple.com/gb/app/bloom/id292792586?mt=8>>; Björk and Second Wind Ltd, *Biophilia*, 2011

power of smartphones, combined with access to its sensors and data streams to use the device itself as a medium, with musical works made available as apps. It is interesting to note that if we consider that apps are in fact software computer programs, this sort of compositional practice then simply extends early computer music experimentation where pieces existed as computer programs such as the early works of Max Mathews, an early pioneer of computer music.¹⁹

It is likely that adaptive/mobile music will gain popularity as processing power on mobile devices increases and frameworks for composers are developed and standardised (more on this in the 'Writing Music Vs Writing Frameworks' section later in this chapter). In the same way that the Official Charts Company launched a download sales chart as an addition to the record sales in 2004, perhaps future app music will see its own recognition, as the ever-changing music industry adapts.²⁰

There is plenty of procedural music in existence, mostly living as game soundtracks, app soundtracks, installations and artistic works, or within live coding events, but any attempt to capture these moments is purely a static recording, or an example of one possible iteration of the music, rendering moot its purpose to be different each time.

A piece epitomising the context in which my work sits, released halfway through this PhD project in 2017, is the album app *Reflection* by Brian Eno.²¹ Eno, who is often cited as having coined the term 'generative music', has seen decades of his compositions evolve through different media: from *Generative Music 1*, which lived as a computer installation, the soundtrack to *Spore* (a commercially successful video game), the interactive app, *Bloom*,

<<https://itunes.apple.com/gb/app/björk-biophilia/id434122935?mt=8>>; Surge and Falter Ltd and Massive Attack, *Fantom*, 2016 <<https://itunes.apple.com/us/app/fantom-sensory-music/id1062360670?mt=8>>; Brian Eno, Peter Chilvers and Opal Ltd., *Reflection*, 2017 <<http://www.brian-eno.net/>>.

¹⁹ Max V. Mathews, 'An Acoustic Compiler for Music and Psychological Stimuli', *Bell System Technical Journal*, 40/3 (1961), 677–94.

²⁰ John Williamson and Martin Cloonan, 'Rethinking the Music Industry', *Popular Music*, 26/2 (2007), 305–22.

²¹ Eno, Chilvers and Opal Ltd., *Reflection*.

and finally, *Reflection*, which does not require physical interactions, but uses other data such as time of day to vary the music.²² My own entry to generative compositions also began in a different context of what could be called ‘functional music’, i.e. music designed for a different purpose than pure listening pleasure, in music for health apps and devices. Through building compositions with an external purpose, I was able to gain the skills and hone processes to progress to pieces that can be considered standalone musical works.

The invention of the mobile cassette player in the late 1970s heralded a new method of music consumption, and this paradigm is still strong, though the technology has moved through various physical media (compact discs, minidiscs, mp3 players, and now smartphones). Most music reproduction systems today expect to be mobile-first (meaning designed for mobile use in mind as the first choice of delivery medium) and play from music streaming services. This ubiquity, along with advances in technology that allow greater processing power, led to the mobile device being a perfect medium through which to distribute procedural music.

CD singles were a significant influence on my listening habits during the 1990s leading to a desire to obtain singles that included remixes of the same track (Figure 1 shows examples of singles with remixes). My practice is driven by this concept, by the urge to listen to different iterations of a piece of music, by remix culture. As opposed to static (one remix at a time) remix culture, in real-time procedural music the differences between iterations might be chosen by indeterminacy, in the style of composers such as John Cage, or by environmental and biophysical factors such as time of day or heart rate following more recent examples such as Brian Eno and Endel Music.²³ It is for this reason that, as a

²² Nick Collins and Andrew R. Brown, ‘Generative Music Editorial’, *Contemporary Music Review*, 28/1 (2009), 1–4; Maxis, *Spore* (2008) <<https://www.spore.com/>>; Eno, Chilvers and Opal Ltd, *Bloom*; Eno, Chilvers and Opal Ltd., *Reflection*.

²³ Endel© Sound GmbH, ‘Endel’, *Endel* <<http://endel.io/>> [accessed 11 June 2019].

composer focussed on creating procedural music, I am producing a collection of pieces that will offer altered versions on each listen, a parallel that has already been drawn in the mobile music community from the previously mentioned 2006 report: ‘Many mobile music projects draw on earlier popular electronic music movements such as remix and DJ-culture’.²⁴

Remix Fascination

Remixes in streaming software (screenshot captured in January 2020)

CD single from 1996, with seven remixes

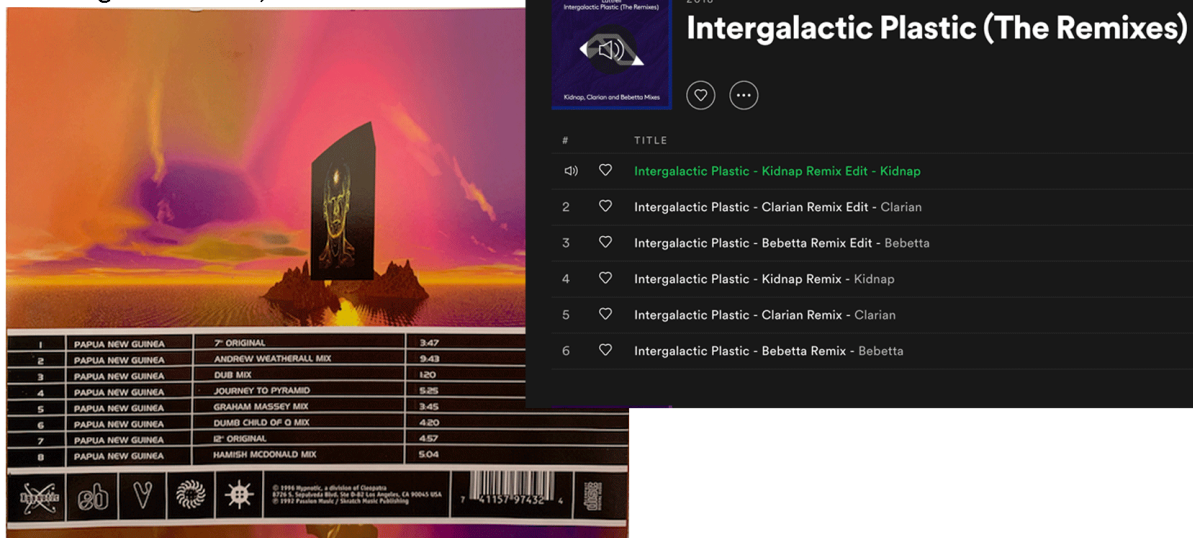


Figure 1 Image of a CD cover and a Spotify playlist to show examples of releases with several remixes.

Scope of Thesis

In this thesis, I have tried to balance the amount of detail on computational process with a more overall explanation of compositional decision-making, to reflect this research sitting within musical practice, as opposed to the field of computer science. As such, illustration of the underlying mechanisms or dataflow computational frameworks, or programming

²⁴ Lalya Gaye and others, ‘Mobile Music Technology: Report on an Emerging Community’, in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, NIME ’06 (Paris, France, France, 2006), pp.255.

paradigms related to the algorithmic components themselves, will be omitted in favour of focusing on the compositional aspects of writing music. Reference to purely technical aspects will thus only be made when directly relevant to composition.

Types of Musical Apps



Figure 2: Diagram to categorise types of musical apps, with apps most relevant to this thesis in the middle.

App Categories:

In order to accurately describe the paradigm in which my work sits, it is necessary to examine and categorise similar apps that have influenced and led to the musical work as an app. There are many types of apps available that involve music in some way. Figure 2 shows examples of the apps pertinent to my research in the centre, and apps that fall beyond the scope of my work around the outside.

Generative Music Apps

These apps can take inputs through the mobile device, both active (for example using the touchscreen interface) and passive (taking the time of day from the device as a data source).

For apps that use biofeedback such as heart rate variability (HRV) to generate music for health applications, there is a much smaller pool of works, with *Endel* being a prominent example.²⁵ This does take a variety of data from a mobile phone's array of sensors to generate music to compliment the users' activity, and interestingly, is listed in the iPhone store under the 'Health and Fitness' category.

Tools for Making Music

There are many apps available that act as software synthesisers, drum machines, effects banks, and DAWs (Digital Audio Workstations). These apps are not within the scope of my research, as they are made to allow the user to create their own works, and in many cases as a replacement for real, physical devices. However, a couple of apps that challenge this demarcation include *Biophilia* by Björk, and *iDaft* by Daft Punk which do provide the user with tools to make music, but where the audio samples used by the interface are identifiably Björk and Daft Punk sounds.²⁶ These apps blur the distinction between creator and consumer, in a meta-creative paradigm where the composer offers a pre-composed bed of sound that the user then plays with in a semi-compositional way.

Enhancement Apps

These types of apps align very closely with my research, in that they often take external inputs, such as location markers, image capture, or GPS location to trigger samples or modify sonic material. However, while they use audio in various capacities, which enhances

²⁵ Endel© Sound GmbH, 'Endel'. <<https://endel.io/>> [accessed 11 June 2019].

²⁶ Björk and Second Wind Ltd, *Biophilia*; Daft Punk and Anders Svendsen, *iDaft Jamming*, *Daft Punk Edition*, 2009 <<https://apps.apple.com/us/app/idaft-jamming/id308712434>>.

the user experience the app is designed to deliver, I have excluded them from my research in that they are not designed specifically to engage users in a musical way, and are mostly spoken information, or environmental recordings designed to augment and complement the user's situation.

Games

Early in my research, I engaged with game soundtracks, as generative music has found a comfortable home in this environment. Preliminary investigations were made using a simple game that mapped user perceptions and biometric data such as frustration and danger to musical parameters.²⁷ I shifted my focus towards applications that used similar paradigms (data-driven user interaction where music had by necessity to be generative and adaptive to user behaviour), using the shared techniques and philosophies that apply to generative music on the whole, and learning from the game prototypes I co-created.

Well-Being Apps

There is a large crossover with my work and well-being apps, where music is used to train or focus a user in some therapeutic activity (e.g. breathing technique or meditative practice). Some of the examples listed in Figure 2 hint at generative music, for example, *Relax Melodies Sleep Sounds* allows the user to combine different samples of relaxing sounds (such as the sound of rain or wind chimes) to create a soothing background drone that assists the user with falling asleep.²⁸

²⁷ Elise Plans, Davide Morelli and David Plans, 'AudioNode: Prototypical Affective Modelling in Experience-Driven Procedural Music Generation', 2015
<https://web.archive.org/web/20170610155005id_/http://www.ccgworkshop.org:80/2015/Plans.pdf>.

²⁸ Ipnos Software Inc., *Relax Melodies Sleep Sounds*, 2009
<<https://apps.apple.com/us/app/relax-melodies-sleep-sounds/id314498713>>.

Promotional Apps

I have not included music apps of a promotional nature, which are released to promote a song or album. These are apps, usually including some sort of music player, that often have links to merchandise purchasing, group chats or forums for the fanbase, videos, real-time lyric scrolling, and static remixes of songs (reminiscent of CD singles / EP vinyl). An example of this is the *Piano Ombre* app by François & The Atlas Mountains, released in 2014, where exclusive tracks could be ‘unlocked’ by sharing user-captured photos of the sun.²⁹

Writing Music Vs Writing Frameworks

There already exist some projects working towards establishing a framework to enable artists that normally work with linear music to develop apps with generative capabilities.

‘RJDJ’ founded by Michael Breidenbruecker (who co-founded Last.fm) was a company pioneering in this space, using their framework to develop apps with many different collaborators, including Little Boots, Imogen Heap, and Hans Zimmer, as well as their own app which included ‘scenes’ composed by various adaptive composers.³⁰

‘PD Party’ and its android equivalent ‘Droid Party’ and ‘MobMuPlat’³¹ are apps that allow you to run PD patches on mobile devices, which moves towards a solution to the problem of distribution. However, these still tend to be used only by contributors to the Pure Data (and associated) communities.

The artist Gwylim Gold worked with a team including Mick Grierson from Goldsmiths, University of London, to create the ‘Bronze’ format, and successfully released an

²⁹ Stuart Dredge, ‘Domino Launches Piano Ombre App for François & The Atlas Mountains Album’, *Music:)Ally*, 2014
<<https://musically.com/2014/03/24/domino-launches-piano-ombre-app-for-francois-the-atlas-mountains/>> [accessed 28 July 2021].

³⁰ RjDjme, ‘RjDj The Mind Twisting Hearing Sensation’ (2008)
<<https://www.youtube.com/watch?v=0fxQFly1BQA>> [accessed 5 August 2021].

³¹ <https://droidparty.net/> and <https://apps.apple.com/gb/app/pdparty/id970528308>

album, *Tender Metal* in the form of an album app, where: ‘the sounds are not bound together in the way they are in conventional formats. In Bronze every sound is subject to a set of laws with a new and unique track being generated in realtime on every playback’.³²

More recently, an AHRC-funded research project titled: *The Commercialisation of Interactive Music* led by Prof. Justin Paterson at the University of West London has been conducting detailed research on the album app format and creating a framework whereby artists can embed their work into an interactive app³³.

These frameworks may prove invaluable to expand and promote this field, by attracting more established artists who want to experiment with this form. However, these groups are approaching the field from an engineering perspective, using the composition languages as a foundation with which to create tools for other composers to use, often simplifying the composition (programming) language so that composers who aren’t familiar with it can nonetheless interact with it and produce work using its paradigms. Instead, I use the programming language as the fabric of the composition, from a musical perspective, to create musical works in and of themselves, allowing for more detailed and individualised exploration. It is not my aim to master the computer science of audio and signal processing, and deal with macro architectures, but to compose with sound at its most malleable level, where my algorithms **are** the arrangement of rhythm, pitch and timbre, and the synthesised or sampled sounds are my instrumental forces. My compositional choices will be explained in more detail in Chapter 2 - Overview of Compositional Methods.

³² DIY, ‘Soap Box: Welcome To The Bronze Age’, *DIY* (2011) <<http://diymag.com/archive/soap-box-welcome-to-the-bronze-age>> [accessed 8 October 2019].

³³ Justin Paterson and Rob Toulson, ‘The Commercialisation of Interactive Music’, University of West London Research <<https://www.uwl.ac.uk/research/research-centres-and-groups/prism-music-and-screen/commercialisation-music>> [accessed 8 October 2019].

Research Aims

My compositional practice investigates the processes whereby music can emerge out of systems. By systems, I refer to algorithmic processes that I use to build a composition. It is these processes that fascinate me, their ability to create form and structure over time often as a result of very simple rules. My work is comprised of lots of little processes working together. The clockwork nature of how my compositions emerge out of this procedural amalgamation is difficult to outline as a methodical framework, because it often emerges out of an evolving sense of how these components help each other: rhythmic, melodic, timbral and textural components are frequently explored at the microscopic level, where I work on that particular process as though it, in and of itself, were a composition. Then, their interplay is explored in the context of some larger compositional aim. Sometimes I have a brief for this, such as is the case in my industrial work, where I am given parameters and variables that influence the compositional systems I make. An example is the sonification of heart rate and its variability for biofeedback applications, where I am constrained by variables that force me to focus on timbral and spatial qualities of the music (as the rhythmic and formal counterpoints are limited).

The processes I explore often relate to synthesis and algorithmic systems: rhythmic and timbral components and their interplay give rise to form in sequences and patterns that are procedural, or **generative**. Mozart's dice games are often cited as an example of early procedural music, in that the order of small sections of score are decided aleatorically,³⁴ but more seminal works from a time where procedural music was being further explored,

³⁴Stephen A. Hedges, 'Dice Music in the Eighteenth Century', *Music & Letters*, 59/2 (1978), 180–87.

include Steve Reich's *It's Gonna Rain* and Terry Riley's *In C*. Brian Eno described these works as **generative**, in that they generate music by following a procedure.³⁵

Another important focus in my practice is to compose pieces that sound different on each playback while still recognisably being the same piece. Phil Hennigan calls this 'liquid music' and describes it as 'any (recorded) musical work which is composed with the deliberate and inherent potential to differ from one playback or iteration to the next'.³⁶

David Collier offers a more detailed description: 'indeterminate compositions that are composed to be listened to on the move. Music created specifically to be listened to while a listener is moving around that have some element of chance built into it so that it adapts to the listener and/or their environment'.³⁷

From a theoretical perspective, my compositional practice could be understood from other aesthetic frameworks such as Umberto Eco's definition of the 'Open Work':

Every performance *explains* the composition but does not *exhaust* it. Every performance makes the work an actuality, but is itself only complementary to all possible other performances of the work. In short, we can say that every performance offers us a complete and satisfying version of the work, but at the same time makes it incomplete for us, because it cannot simultaneously give all the other artistic solutions which the work may admit.³⁸

The music I want to create incorporates both Hennigan's and Collier's definition: music composed with the deliberate potential to differ on each playback, intended to adapt to listener behaviour. Both of these definitions have foundations in Hosokawa's seminal

³⁵ Brian Eno, 'Generative Music', *In Motion Magazine*, 1996 <<https://inmotionmagazine.com/eno1.html>> [accessed 19 July 2021].

³⁶ Hennigan, 'Music in Liquid Forms: A Framework for the Creation of Reactive Music Recordings'.

³⁷ Collier, David 'What Is Mobile Music? – Mobile Music #5 //', David Collier, 2012 <<http://davidcollier.ie/what-is-mobile-music/>> [accessed 6 November 2019].

³⁸ Eco, Umberto, 'The Poetics of the Open Work. 1962', *Participation: Documents of Contemporary Art*, 2006, 20–40

work *The Walkman Effect* which provides an insightful analysis of some of the social and cultural impacts of portable, personal music brought about by the new technology of personal cassette players, as distinct from other portable music systems such as car stereos or ghettoblasters.³⁹

I am drawn to the concept of minimalism in music, with the music of Terry Riley, Philip Glass, Meredith Monk, and Else Marie Pade capturing an aesthetic of music that I find fascinating, that is to build complexity out of simplicity. It is perhaps not such an accident that this genre should appeal to me, particularly in the setting in which I am compelled to create my work - on small processing units: their self-imposed limitations demanding simple rule-based structures. This point has been made by Agnes Guerraz and Jacques Lemordant in their chapter in *From Pac-Man to Pop Music*, edited by Karen Collins: 'Indeterminate adaptive digital audio meets the requirements of mobile phones with limited memory and digital audio mixer. We believe that the experiences of minimalism and contemporary musical forms will be fundamental in building rich audio for multimedia'.⁴⁰

The next chapter will look more closely at the contextualisation of my work within existing practice. Chapter 2 will examine my compositional methods in a general sense, before more detailed explorations into the individual works in chapters 3, 4 and 5, where I discuss my portfolio pieces for the *Breathing Stone*, interactive apps and procedural apps respectively. In Chapter 6, I reflect on the portfolio and the limitations that shaped its progress, along with contributions my work has made, and lessons I can take forward into my future practice.

³⁹ Shuhei Hosokawa, 'The Walkman Effect', *Popular Music*, 4 (1984), 165–80.

⁴⁰ Agnes Guerraz and Jacques Lemordant, 'Indeterminate Adaptive Digital Audio for Games on Mobiles', in *From Pac-Man to Pop Music*, ed. by Karen Collins (Farnham, UK, 2008).

Chapter 1 - Contextualisation

Procedural and Electronic Music

With the exception of the synthesisers and production techniques of popular music I listened to as a child, I was unaware of the existence of electronic music as a genre until the mid-1990s. As a teenager in 1993, I was introduced to *Ambient Works 85 - 92* by Aphex Twin,⁴¹ and remember being profoundly moved by it, in particular the sound palette used, which seemed so distinct from anything I had previously heard. It was around this time that I also discovered the music of Björk, who released *Debut* in July 1993⁴², and again was fascinated by the instrumentation, in terms of the sounds used as instruments on this first solo album.

Other than some brief experiments with Cubase (a popular Digital Audio Workstation) in my school lunch breaks (Music Technology was just being introduced to my school by 1994), I did not encounter electronic composition tools until, as an undergraduate, I was introduced to ProTools, and Max/MSP (among other sound editing software). This was when I began to explore electronic music in earnest, as well as learning its developmental history through the works of Stockhausen, Delia Derbyshire, Daphne Oram and many others, which included procedural music.

There is a large body of research on procedural music, and the different contexts in which it lives. Xenakis describes various approaches to the application of algorithms in composition in his book *Formalized Music: thought and mathematics in composition*.⁴³ Wooller et Al brought together many texts in their article *A framework for Comparison of*

⁴¹ Aphex Twin, *Selected Ambient Works, 85-92* (1992).

⁴² Björk, *Debut* (1993).

⁴³ Iannis Xenakis, *Formalized Music: Thought and Mathematics in Composition* (New York, 1992).

*Process in Algorithmic Music Systems*⁴⁴ with the conclusion that there are three main categories: analytical, transformative, and generative. Karen Collins has written in-depth on procedural music, and although mostly within the context of games, much of her writing can be applied to procedural music in general. In *An Introduction to Procedural Music in Video Games* she also lays down some distinctions between the different types:

The non-linear, variable elements in the sonic aspects of gameplay (what I have elsewhere defined as *dynamic audio*; see Collins, 2007) can be loosely divided into *interactive* and *adaptive* audio. Interactive audio refers to sound events directly triggered by the player, affected by the player's input device (controller, joystick, and so on). ...*Adaptive* audio events, on the other hand, are unaffected by the player's direct actions, although they are inevitably affected by indirect actions.⁴⁵

Collins distinguishes between music which is affected by *intentional actions* and that which is guided by unintentional actions. With regard to mobile devices, one might add a third option whereby the action is listener-driven (or is caused by the smartphone's natural interaction with the user or her world), but not directly intended by the listener as an action to affect the musical outcome. Examples of this are found in the *Inception* app (RJDJ, 2010)⁴⁶ where the sound is affected by the speed at which the phone is moving (ie whether you are in a moving vehicle, on foot, or stationary) and in *Reflections* by Brian Eno where the time of day affects the harmony.⁴⁷ This is the difference between swiping a phone deliberately through the air to change a sound (an idiom found in many 'lightsabre' simulation apps) as opposed to the movement of the phone because you are taking a walk (such as in my piece,

⁴⁴ Rene Wooller and others, 'A Framework for Comparison of Process in Algorithmic Music Systems', ed. by Burraston David and Edmonds Ernest (presented at the Generative Arts Practice, 2005), 109–24.

⁴⁵ Karen Collins, 'An Introduction to Procedural Music in Video Games', *Contemporary Music Review*, 28/1 (2009), 5–15.

⁴⁶ The *Inception* app is no longer available in the app store in the UK, being no longer supported, but UJam have taken on the project <http://inception-app.com>.

⁴⁷ Described here by Peter Chilvers who worked on the app with Eno: <https://brian-eno.net/reflection/>

Elerem). This small distinction becomes a significant difference in terms of listener experience.

Interactive Music on Mobile Devices

The evolution of mobile music apps has been documented in several papers, notably in the writings of Joel Chadabe⁴⁸ and in the introduction to the variPlay project.⁴⁹ In a previous article, I created a timeline of release dates of the apps that I felt particularly achieved the combination of reactive music and portability that I aim to emulate within my own practice (see Figure 3). Björk and Eno's early contributions differ slightly from the rest in that their works rely on deliberate user interaction through physical gestures on the touchscreen of the device. Björk's work borders on the category of music-making tools rather than a composition, as described in the introduction, pushing it outside of the scope of this thesis. However, the soundscape is identifiably Björk's sound and could be considered as paradigm-shifting, negotiating a new medium to convey music to an audience in the mainstream more successfully and before other artists, as described by Nicola Dibben in her article *Visualising the App Album with Björk's Biophilia*.⁵⁰

⁴⁸ Joel Chadabe, 'The Role of Apps in Electroacoustic Music: A New Dimension for Music through Tablets and Other Devices', *Organised Sound*, 20/01 (2015), 99–104; Joel Chadabe, 'Remarks on Computer Music Culture', *Computer Music Journal*, 24/4 (2000), 9–11.

⁴⁹ Justin Paterson, Rob Toulson and Russ Hepworth-Sawyer, 'User-Influenced/Machine-Controlled Playback: The variPlay Music App Format for Interactive Recorded Music', *Arts & Health*, 8/3 (2019), 112.

⁵⁰ Nicola Dibben, 'Visualizing the App Album with Björk's Biophilia', in *The Oxford Handbook of Sound and Image in Digital Media*, Oxford Handbooks (Oxford, UK, 2014), 682–706.

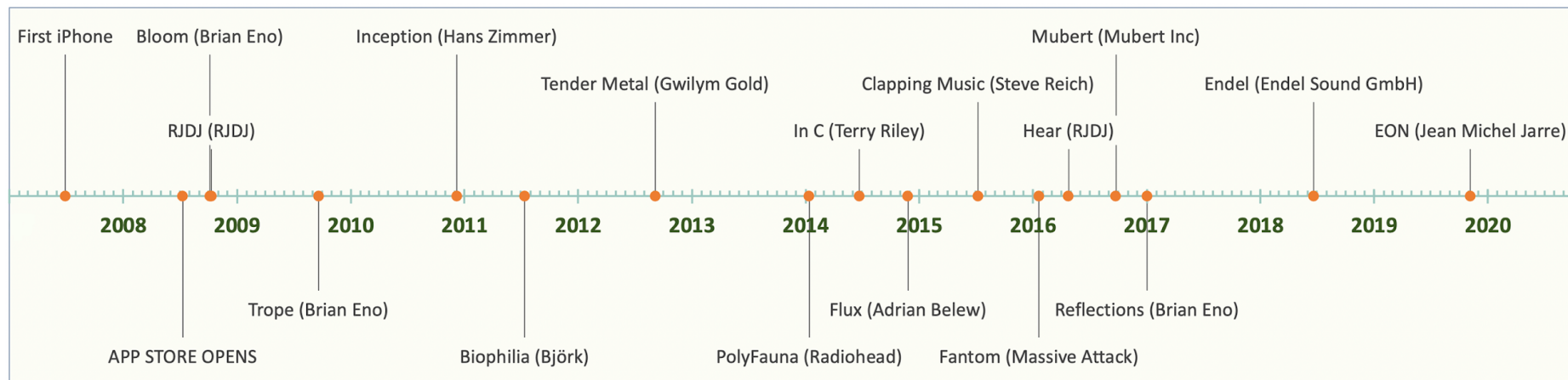


Figure 3 *Timeline of Seminal works, an updated version of the original in* **Composer in your Pocket: Procedural Music in Mobile Devices.**⁵¹

⁵¹ Elise Plans, 'Composer in Your Pocket: Procedural Music in Mobile Devices', *Musicology Research Journal*, 2017, 51.

Inception (which was released to accompany the film of the same name) and *Fantom* both take data from smartphone sensors, such as how fast it is moving (i.e. is the user in a moving vehicle or walking, or standing still) what time of day it is, and even which season. Additionally, you can use the camera in *Fantom* to affect the music in different ways and even, if you have a wearable connected, use your heart rate to modify the music. Both these apps were made possible in part by the adaptive music composer Robert M Thomas as part of RJDJ whose work is also a main inspiration for this thesis, and whose mentorship has been invaluable in discovering my own working process.

Further descriptions of the apps in Figure 3 can be found in *Composer in your Pocket: Procedural Music in Mobile Devices*.⁵²

Although I have given examples from the popular domain, there have been academic groups working within this same paradigm, notably: theses from Phil Hennigan (previously quoted) and Adam Jansch, and David Collier's work from 2012.⁵³ Collier released an interactive app, *In Your Own Time* which makes use of several sensors on the phone, including pace detection. Adam Jansch explores various aspects of 'the open outcome record', exploring installations and purpose-built devices as media for generative music, culminating in his app *Futures EP*.⁵⁴

Larger projects within the academic domain tend towards app frameworks for composers or coders to create music within, such as the variPlay project or the BRONZE

⁵² Plans, 'Composer in Your Pocket: Procedural Music in Mobile Devices', 51.

⁵³ David Collier, 'In Your Own Time' (2012)

<http://davidcollier.ie/wp-content/uploads/2012/08/4_InYourOwnTime.pdf>; Hennigan, 'Music in Liquid Forms: A Framework for the Creation of Reactive Music Recordings'; A. Jansch, 'Towards the Open Outcome Record: A Portfolio of Works Exploring Strategies of Freeing the Record from Fixity', 2011 <http://eprints.hud.ac.uk/id/eprint/16730/8/One_man_band_x_n_-_cycle_2.m4v>.

⁵⁴ A. Jansch, 'Futures EP', *AdamJansch* <<https://www.adamjansch.co.uk/code/futures-ep>> [accessed 16 June 2020].

system, previously mentioned in the introduction.⁵⁵ These include a focus on the commercial possibilities of generative music apps and could be the precursors to the standardisation that is often needed before something can be scaled to industry-level acceptance.

Although these are pushing in the right direction, they have their drawbacks, as Hennigan points out:

...the proposition of the BRONZE format (see Chapter Three) as a new format for music, though laudable, falls short due to the nature of the work: offering only a single inherent style of generative music, this programming imparts a limited aesthetic onto any musical work written for it. This is commendable in the production of a single work, but less so as further works are created with the same tools.⁵⁶

Human Computer Interaction

As mobile devices are designed to act as an interface between humans and software, there are a lot of overlapping interests between music apps and the Human Computer Interface (HCI) community. This includes apps that make music via deliberate manipulation of the device, through actions that can be determined using the device's gyroscope, or via a touchscreen, for example. While there is a large body of work associated with turning the mobile device into a musical instrument, as mentioned previously, there is an aspect of HCI that is directly relevant to how my compositional practice developed and why it emphasises adaptivity to passive data and sensors, and that is data sonification.

⁵⁵ Paterson, Toulson and Hepworth-Sawyer, 'User-Influenced/Machine-Controlled Playback: The variPlay Music App Format for Interactive Recorded Music', 112; M. Grierson, 'Gold + Goldsmiths = BRONZE: Mick Grierson Explains the Bronze Project... – Blog.DoC' <<https://www.doc.gold.ac.uk/blog/?p=208>> [accessed 8 October 2019].

⁵⁶ Hennigan, 'Music in Liquid Forms: A Framework for the Creation of Reactive Music Recordings'.p.3.

Data Sonification

Data Sonification is in itself an interdisciplinary field, drawing on cognition and perception studies, auditory scene analysis, timbre perception studies and more. In her chapter *Sonification ≠ Music*, Carla Scaletti asks for a distinction between data sonification, and data-driven music, stating: ‘a computer musician has both the skills and the opportunity to pursue both music composition and data sonification. By keeping those two activities separate, by defining them differently, a musician has an opportunity to pursue each of them in different contexts, with knowledge and experience gained from each pursuit informing and enhancing the other.’⁵⁷

The data that is sonified in my compositions is not intended for diagnostics or scientific analysis, but purely to connect the listener to that which it is representing, for example hearing your own heartbeat, or your HRV levels. Using music to guide a listener through a deep breathing exercise (a common meditative practice), while also sonifying their heartbeat, instantly conveys the benefits of the exercise, even if they have their eyes closed. *Bespia* is part of an ongoing experiment into how adaptive music(s) can be helpful in biofeedback mechanisms for digital therapeutics aimed at ameliorating the symptoms of anxiety and depression.

SUMMARY

This chapter has attempted to define my practice within the existing narrative of generative music on mobile devices. It brings together the various facets that contribute to the specific compositions I aim to create: generative music in general and music that responds to indirect external influences of the listener (such as pace or heart rate) as

⁵⁷ Carla Scaletti, ‘Sonification ≠ Music’, *The Oxford Handbook of Algorithmic Music*, 2018, 363.

detected through a handheld device. I have emphasised my compositional aim to create works that sound different with each iteration. I have listed some of the previous works that have inspired me and also acknowledged neighbouring fields that also influence my work such as the Human Computer Interface and data sonification communities.

Chapter 2 - Overview of Compositional Methods

This chapter will focus on general methods that apply throughout the works in the accompanying portfolio. More detailed analysis and reflection on composition methods will be given in the individual descriptions of the works in Chapters 3, 4, and 5.

Foundations of Compositional Approach

If composition is fundamentally the organisation of sound(s) in time, then my method of using a timecode system in PD, where I send a universal count around the patch that can be used to trigger sounds in time, is not so different from arranging notes on a score that will be performed along a timeline. The speed of the piece can be changed in the same way a tempo can be adjusted.

An instrumental composer might think carefully on which instruments to use, to achieve the desired timbral quality, whether influenced by a desired mood or emotion, a brief given by a commission, or even practical accessibility to the instruments. In my artistic process, I create these sounds electronically, either through real-time synthesis, or by creating samples to be triggered in PD, or by sampling real instruments or voice, to achieve the sound I want for the piece. Much of my time is spent designing synthesised sounds within PD, to create interesting and morphing sounds using common Digital Signal Processing (DSP) techniques. The skills in sound design needed to achieve this were seeded in my electroacoustic work for my master's degree, and then developed through working with professionals in industry such as Robert M Thomas, and also consulting various practical texts, such as the indispensable 'Designing Sound' by Andy Farnell, 'Loadbang' by Johannes Kriedler, and 'Designing Sound with Pure Data' by Tony Hillerson⁵⁸.

⁵⁸ Andy Farnell, *Designing Sound* (Cambridge, MA, 2010); Johannes Kriedler, *Loadbang : Programming Electronic Music in PD* (Hofheim, 2009); T. Hillerson, *Programming Sound with Pure Data* (Dallas, 2014).

Examples of some of the processes I work with to create the ‘materials’ for my compositions are set out in the table in Figure 4.

Temporal	A-Temporal
Sequence of pitches	Designing a synth sound using DSP.
Sequence of percussive sounds	Designing a sound by manipulating a sample
Repetition (using timebase or delay)	

Micro	Macro
Building a process to make pitch choices from a list	Designing a process to Structure a piece over time
Building a process to continuously pan a sound	Designing a process to modulate the sound of an instrument over time
Building a process to make a list of pitches	Designing user experience, how the listener will need to interact in ways other than listening (if at all).
Creating melodic material	How to sound out a melody

Musical	Non-musical
Building fade envelopes	Building a weighted random object
Designing sounds with different timbral qualities	Calculating playback speed for microtonal playback of pitched samples.
	Designing a ‘mixer’ system to adjust the volume of different sounds relative to each other.

Figure 4 Morphology of processes used in composing the works for the accompanying portfolio to this thesis.

Compositional Approaches of Game Music Composers

Guy Whitmore, an established video game composer, has also considered the possibility that the format could be applicable in autonomous works: ‘In 1999 Guy set out on

his own as a free-lancer, expanding the range and styles of games he had scored. He then took a chance with startup Bootleg TV, founded by Robert Fripp of the band King Crimson. There he explored the possibilities of non-linear music as a standalone format'.⁵⁹ More recently in a webinar in 2018, Whitmore stated that he has been creating an album, using the WWise software (typically used to create game soundtracks):

I think we tend to think of variable, or generative music as, 'oh it creates a completely new piece of music'. I'm actually more interested in, 'how do I create an interesting piece of music and then have it vary enough to keep it interesting over several listens, many, many listens and you discover new things the more times you listen to it'. And that's the concept behind this *Song of Coqui* project and the ultimate idea is, I hope, at some point, it's been years, to have an iOS or android release of this as a little player.⁶⁰

Whitmore has entertained the concept of a procedural album (that modifies itself with each iteration) for a while but is still looking for an outlet. Having worked in game audio, using WWise, his method appears to be more aligned to DAW-style composition (aligning blocks of sound in a grid), but with the added component of multi-possibility (aleatorically choosing from pre-written segment versions) enabled by the WWise interface. Whitmore's choice to distribute this long-standing project as an app mirrors my own needs for a dissemination channel that can go beyond the mere multi-choice reactivity that frameworks like WWise can give an established game composer.

⁵⁹ Chance Thomas, *Composing Music for Games: The Art, Technology and Business of Video Game Scoring* (New York, 2017). pp 326.

⁶⁰ This is transcribed from a webinar: Guy Whitmore, 'Why Music Must Evolve Along With Digital Media', *Twitch.tv*, 2018 <<https://www.twitch.tv/audiokinetic/video/288292691>>.

The Medium affects the Output

Like any medium, the app form shapes the music, in the same way that vinyl records influenced the recorded material they reproduced. Songs were carefully organised on an LP as the tracks on a vinyl album were almost always left to play in order (rather than being able to skip easily with a remote control, or automatically ‘shuffle’ the order). Seven-inch singles limited songs to around three to four minutes, as this is what could physically fit onto that size of disc. In turn, the CD’s 740MB data limit dictated album lengths in the 1990s⁶¹.

‘Album apps’ so far have tended to contain fewer ‘single’ tracks, than would previously have been considered an album (defined by the number of tracks that could fit on an LP or CD). Considering the size of the average music collection, if all music releases became generative, a smartphone would need to hold four thousand to six thousand apps!⁶² In addition, audio files require storage space: an uncompressed mono WAV file sampled at 48 kHz, with a bit depth of 16 that lasts just five seconds, is approximately 450kB, so ten five-second mono samples would already be 4.5MB, with stereo being twice that. Apps can thus quickly become very large, once other files essential to the app’s function are added in. Examples of app sizes are shown in the table in Figure 5.

⁶¹ There are numerous references to this, including Mark Katz, *Capturing Sound: How Technology Has Changed Music* (Berkeley, 2010).

⁶² Mohsen Kamalzadeh, Dominikus Baur and Torsten Möller, ‘Listen or Interact? A Large-Scale Survey on Music Listening and Management Behaviours’, *Journal of New Music Research*, 45/1 (2016), 42–67.

App	Size (MB)
Bloom (Brian Eno)	18
Trope (Brian Eno)	38
Inception (Hans Zimmer)	157
Biophilia (Björk)	1200
Tender Metal (Gwilym Gold)	373
PolyFauna (Radiohead)	159
In C (Terry Riley)	10
Flux (Adrian Belew)	599
Clapping Music (Steve Reich)	36
Fantom (Massive Attack)	147
Mubert (Mubert Inc,)	13
Reflections (Brian Eno)	172
Endel (Endel Sound GmbH)	226

Figure 5 Table to show examples of mobile application sizes, correct at time of writing, but change rapidly with different releases and updates.

As the size of audio files within apps similarly affects the overall size of the app, difficult choices must be made on quality versus file size. It is possible to fit a lot more audio within an app's overall file structure as mp3 files, but this comes with a loss of quality. Alternatively, one can choose an uncompressed format such as WAV or AIFF but include less material overall. Beyond these, there are other choices and constraints to deal with: should CD quality encoding be good enough (44.1kHz, 16 bit depth)? Will mono files work or are stereo files needed? It is worth bearing in mind at this stage that the quality of the sound at the listener end also ranges wildly from earbuds from a pound shop to high-quality headphones from Bose or Beyerdynamic.

One of the ways to avoid large file sizes is to utilise DSP to create musical material, rather than playback of pre-recorded samples. Then there is only the need to be aware of demand on the CPU of the playback device. In most of my works, I combine the two

approaches, as I feel that adding sampled sounds can broaden the timbral palette, especially as the music will be accessed through what is essentially a digital object (as opposed to a more analogue playback system). It is worth mentioning that ‘natural sounds’ can be synthesised, aiming to recreate specific sounds using Fast Fourier Transform (FFT),⁶³ but to be able to recreate their complexity with any degree of success takes up a lot of processing power and very complex DSP processing.

‘Creating’ Instruments

My ‘instrument’ creation methods can be summarised in the following groups:

1. I use standard DSP tools, synthesis (additive, subtractive) amplitude modulation, frequency modulation, filtering, enveloping, following guidance from Andy Farnell, Tony Hillerson and the Floss Manuals,⁶⁴ which are themselves based on DSP principles from Puckette, Collins, Roads, Dodge and Jerse, and Steiglitz.⁶⁵
2. I sample a live instrument, and then manipulate it using common audio processing techniques, such as playback speed, time-stretching, filtering, reversing, looping, and granular synthesis. For example, in *Montague I*, a long-held piano note is played backwards to build a rising sound foundational layer underneath the repeating chord layer. Its playback speed is also varied to change pitch with the changing harmonies. Changing the playback speed can reveal a wider and more interesting array of sonorities not heard in standard playback, as artefacts of the digitised audio contribute audibly to the reproduced sound. I

⁶³ Fast Fourier Transform is a way of representing a waveform in the frequency domain.

⁶⁴ Farnell, *Designing Sound*; Hillerson, *Programming Sound with Pure Data*; Derek Holzer and others, ‘Floss Manuals’, *Floss Manuals - Pure Data* <<http://write.flossmanuals.net/pure-data/introduction2/>> [accessed 7 June 2020].

⁶⁵ Miller Puckette, *The Theory And Techniques Of Electronic Music* (Singapore, 2007); Nick Collins, *Introduction to Computer Music* (Chichester, UK, 2010); Curtis Roads, *The Computer Music Tutorial* (Cambridge, MA, 1996); Charles Dodge and Thomas A. Jerse, *Computer Music: Synthesis, Composition, and Performance*, 1st edn (New York, 1985); Ken Steiglitz, *A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music*, Addison-Wesley Professional Computing Series (Menlo Park, 1996).

am particularly fond of hearing vocal samples played at different speeds, as the sonic quality of the sound both recalls a physical human body, but at the same time sounds unearthly, as if transcending the body. Examples in the popular domain include *Concrete Walls* by Fever Ray, and *Know Where* by Holy Other.⁶⁶

3. I generate a sound in Logic Pro, combining different ‘instruments’ made from a variety of Logic’s instrument bank, and other sample libraries available through Kontakt,⁶⁷ and then manipulate that, as I would a live sample (using the same techniques described above).

4. I also created a bank of sampled single piano notes, played at a normal volume, and then played softly⁶⁸. I have almost exclusively used the softly-played version of this sample bank, as I find the slightly muffled sound of a softly played piano extremely comforting, and it has become a popular style, made prominent by composers such as Olafur Arnalds and Nils Frahm adding a felt layer to their pianos,⁶⁹ as corroborated in this Twitter exchange between composer/musicians Tom Parkinson and Zubin Kanga (Figure 6):

⁶⁶ *Concrete Walls* appears on the album ‘Fever Ray’ by Fever Ray, (Rabid, 2009) and *Know Where* is on the EP ‘With U’ by Holy Other (Tri Angle, 2011).

⁶⁷ Kontakt is a software sampler allowing access to many sample libraries within a DAW.

⁶⁸ These samples were recorded at my home on a Yamaha B3 piano in December 2016.

⁶⁹ Olafur Arnalds and Nils Frahm, *Trance Frenzy* (2016)

<<https://www.erasedtapes.com/release/eratp081-olafur-arnalds-nils-frahm-trance-frenzy>>.



Figure 6 Screenshot of a Twitter exchange showing a discussion of the popular ‘soft’ piano sound.

Designing Compositional Processes

A significant aspect of my composition process is solving compositional procedures algorithmically, as some aspects of music that we take for granted in more established reproduction methods, such as playing from a score, must be programmed to work within a dataflow language such as PD. An example of this is the timebase system (Figure 7), which I built to be used as an analogue to bars and beats in nearly all of my pieces. Numbers counting up to multiples of four (for duple time) are sent around the patch via the ‘send’ object. This stream of numbers can be received at many locations around the parent patch,

and the 'select' object then used to trigger events as needed. These counts tend to go quite high - up to 64 or even 96 to allow for detailed subdivisions such as demisemiquavers, or triplet counts. The bar count is useful for larger structures, and for organising when sounds will be active or silent, and again can be received within other processes, where they can be cycled locally, for repeating patterns that carry over several bars.

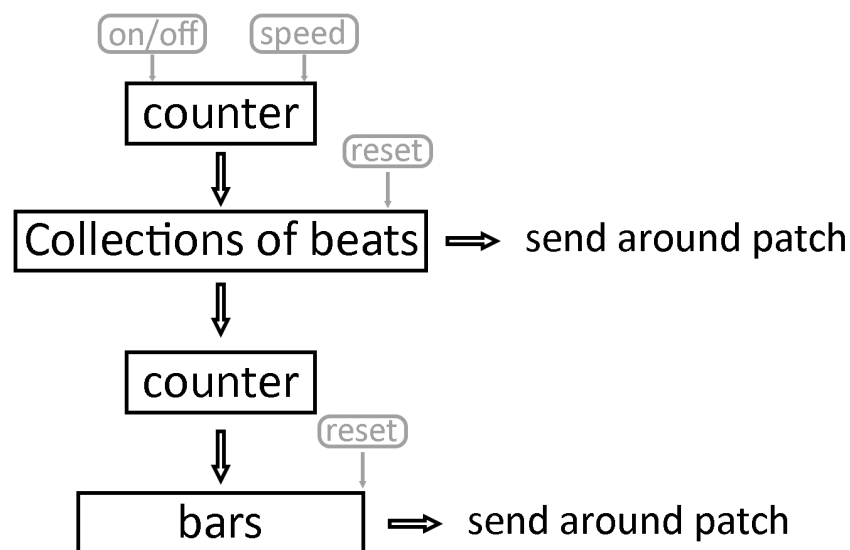


Figure 7 Diagram of the Timebase system used across all the Pure Data patches in the portfolio.

A 'select' object is used in PD to choose which numbers out of the number stream to send triggers on (which in PD are called 'bangs') as shown in Figure 8.

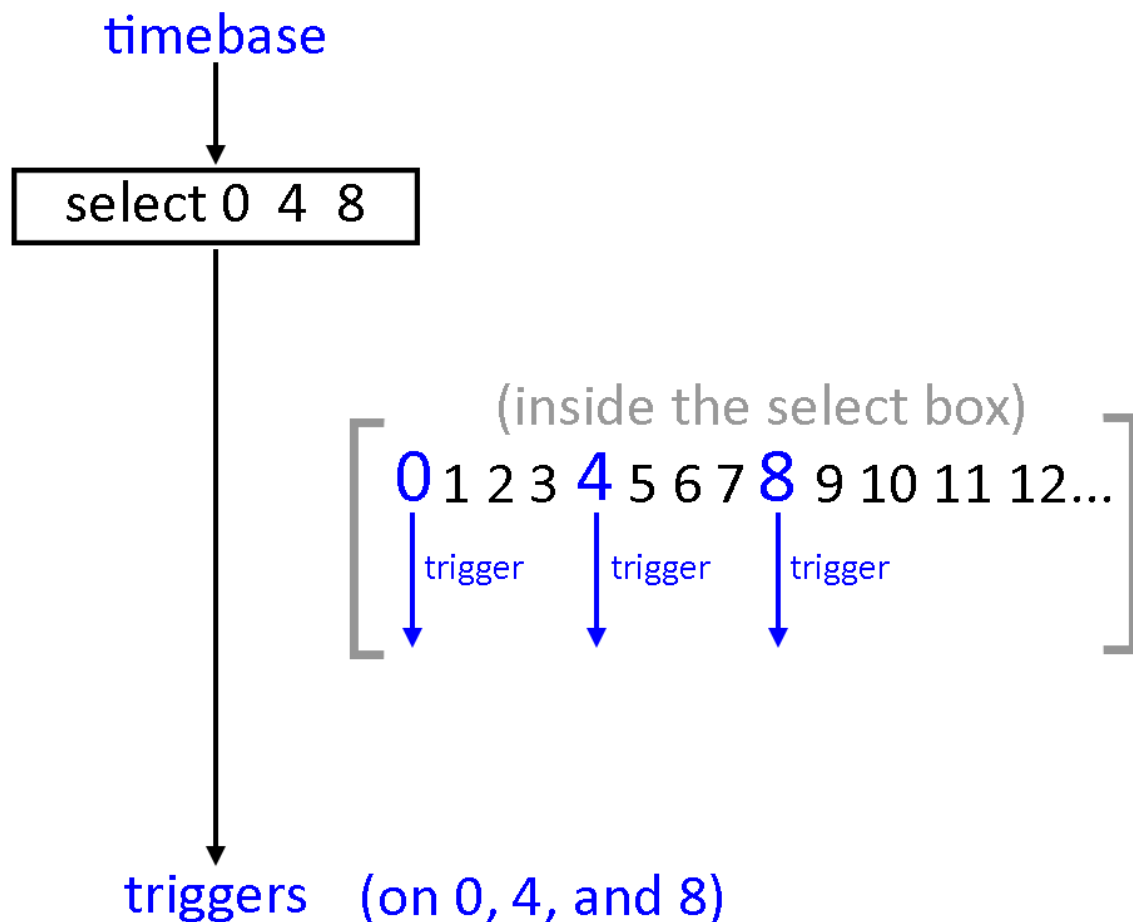


Figure 8 **Diagram to show how the timebase abstraction functions within the Pure Data patches.**

This way of *prescribing* rhythm is reminiscent of the *describing* method of box-notation, or clock-diagrams, as used by Godfried T. Toussaint in many of his articles on rhythm.⁷⁰

Other common algorithms that appear in many of my pieces include: a humaniser for rhythm, a humaniser for volume, a mixer, a sample-playback engine, and a drum machine. These are built in PD and are often saved as *abstractions* that can be used in other situations. Abstractions in PD are patches that can be saved externally to the main patch, and then reused within other patches. This saves having to build the same algorithm over

⁷⁰ Godfried T. Toussaint, 'The Rhythm That Conquered the World: What Makes a "Good" Rhythm Good', *Percussive Notes*, 2 (2011), 52.

and over again, and also allows for starting parameters to be set. They can have inputs and outputs just as a main patch, which can be connected within a parent patch. They are analogous to the modules of a modular synthesiser and can be connected in a similar way. There follows a list of my most commonly used abstractions, which have been constantly revised and improved over the course of my research. Although there are abstractions made by other digital musicians, and even specialist objects that are available within libraries (called *externals*), I have mostly used my own for two main reasons: firstly, the embedding wrapper for PD, *libpd*, is designed to be used with the vanilla version of PD, and bringing in *externals* risks compiling errors at the app-building stage, and secondly, solving dataflow challenges myself afforded me abstractions that are uniquely useful to the specific situations for which they are needed (a need that invariably stems from compositional ideas I'm attempting to realise), rather than having to accommodate and adapt to an approximation designed for a slightly different situation in the context of someone else's compositional framework. Often it is through solving dataflow problems that I find my inspiration as I hit on a sound I like, either by finally solving the problem, or accidentally when my experimentation takes me on a tangent. As an example, when experimenting with building a drone using voice samples, I accidentally triggered all the samples to play at different pitches at the same time producing a sound reminiscent of the chords at the start of *Secret Journey* by The Police.⁷¹ I used the results of this accident to create an instrument where I could control these pitches, which forms the drone sounds in *Elerem*.

During the composition process I often refer to pre-existing works for inspiration, for example when working out the pitch algorithms for the microtonal piece *Montague II*, I listened to, and then read about Ben Johnston's *Sonata For Microtonal Piano*.⁷² I also aurally

⁷¹ *Secret Journey* is a track on the 1981 album by *The Police*, *Ghost in the Machine* (1981).

⁷² John Jeffrey Gibbens, 'Design in Ben Johnston's Sonata for Microtonal Piano', *Interfaces*, 18/3 (1989), 161–94.

dissect almost all the music I listen to, and pick out the elements of it that I like, most often pitch relations, such as the notes of an arpeggiator from an EDM track, but also rhythmic details such as the off-kilter rhythmic repetition in Anna Meredith's *Nautilus*.⁷³ Winifred Phillips describes her experience of this type of listening process in her book *A Composer's Guide to Game Music*, a technique that she refers back to Aaron Copland's *What to Listen For in Music*: 'First, I attempt to pick apart the arrangement, identifying the instruments and their activities'.⁷⁴ As a child, I remember riding in the back of my parents' car listening to The Police's *Greatest Hits* on my Walkman, picking out the bass line, and the chord sequence, the sound qualities of the drum kit (how crisp, and how much reverb) and on which notes Sting would allow the rasp in his voice to come through. In early attempts at composition, my listening habits followed through into my composition practice, bringing this sort of deconstructive listening to a compositional framework that focuses on the juxtaposition of individual elements being played against each other, each with their own variation, as opposed to works with larger overall structural features. Again, it is the development of these smaller juxtapositions and processes that evolves the overall structure of my whole pieces.

Common Abstractions:

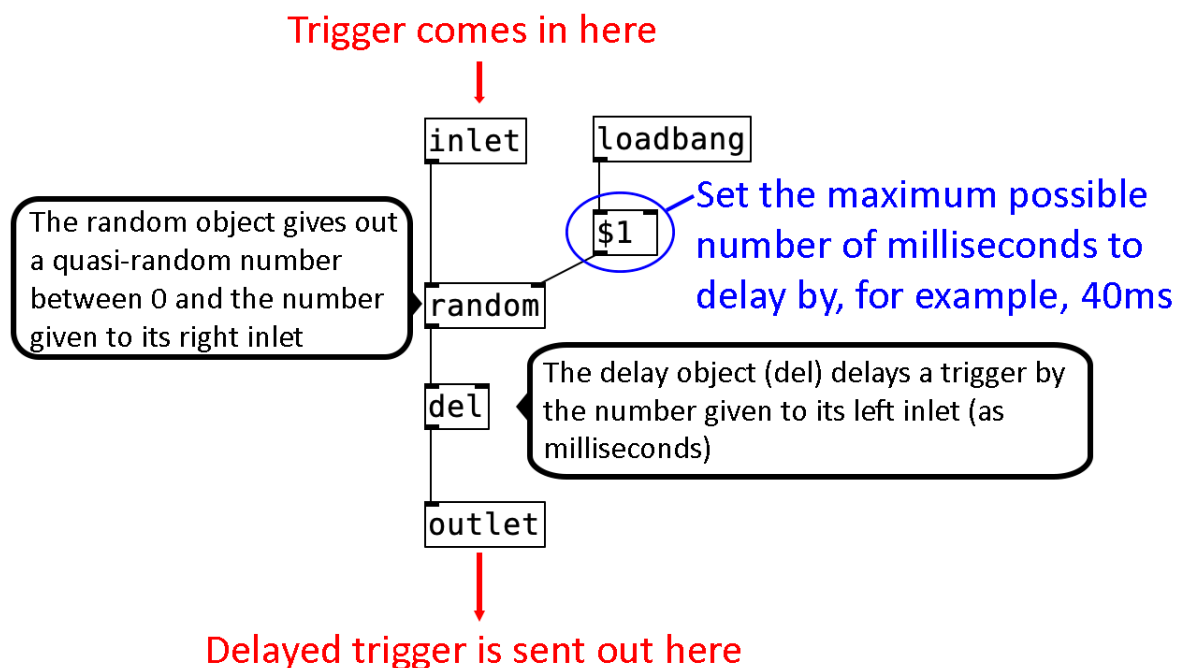
Humaniser

When writing pieces for piano samples, I have found it useful to vary the playback trigger very slightly in time, to give a less robotic, and more human feel to the music, where a human would not be able to press the keys with exact timing. In practice this works by delaying the trigger a random number of milliseconds up to a prescribed limit (Figure 9). This

⁷³ *Nautilus* is a track on the album Anna Meredith, *Varmints* (Moshi Moshi, 2016).

⁷⁴ Winifred Phillips, *A Composer's Guide to Game Music* (Cambridge, MA, 2014). p.23

limit can be set differently depending on how loosely the rhythm can feel, appropriate to the piece. Although it is difficult to measure how many milliseconds of delay a listener can perceive, the general consensus is that around 25-30ms the delay becomes audible, and some research into this can be found in *Early change detection in humans as revealed by auditory brainstem and middle-latency evoked potentials* by Slabu et Al,⁷⁵ though this detection depends on many variables, including volume and context. This approximate measure is reflected in my pieces, where I tend to set the maximum delay at around 40ms, for loose timings.



*The \$1 allows for the number to be set at creation of the abstraction, so in the parent patch, for 40ms maximum delay, it will look like this: `time_humaniser 40`

Figure 9 **Diagram to show how the Time Humaniser abstraction works in the Pure Data patches.**

⁷⁵ Lavinia Slabu and others, 'Early Change Detection in Humans as Revealed by Auditory Brainstem and Middle-Latency Evoked Potentials', *The European Journal of Neuroscience*, 32/5 (2010), 859–65.

The same technique can be applied to the volume of the playback, as a human player would not play a note at exactly the same volume level each time. Therefore, I often use a volume humaniser (a random multiplier to vary the volume between set parameters) with my piano samples, with the parameters set according to the needs of the piece.

This does not completely solve the problem of sampled notes sounding inhuman, so for some pieces I created a 'swell' algorithm, whereby whole phrases of notes can be made to increase and decrease subtly in volume in the same way a human player might subconsciously, or consciously lean into a phrase.

Mixer

Putting lots of different modules together with different outputs will inevitably require some sort of volume balance between the different layers of music. Therefore having a mixer where the inputs can be adjusted in comparison to each other is almost essential. As I am building a piece, I can add more inputs. Finally once the general balance is found, the mixer can be automated, to allow for textural variations throughout the piece, and changed prominence for different voices.

Drum Machine

I have made a drum machine abstraction that essentially plays one-shot samples of different drum kit sounds, appropriate to EDM-style music. The drum machine is set up to respond to the timebase system of the parent patch (and therefore sync with other patches) and has sample players, and effects units such as delay and reverb built-in, whose parameters can also be varied. The abstraction allows for different samples loaded into it, which allows for different soundworlds for different pieces. This abstraction was built with the exact balance of complexity and simplicity I feel is needed for the pieces I write: it is not so complicated

that it requires intricate re-programming for each new piece, but it has enough parameter variability to keep the output varied and interesting.

In particular, I use an algorithm to create changing rhythmical patterns, which can be applied to various different instruments within the drum machine (kick, snare, or hi-hats for example). It takes the stream of timecode from the parent patch and uses a combination of the modulo and select objects to send a trigger to the sound file player. By changing the parameter of the modulo object, one can create interesting rhythmic patterns. The parameter can be varied at random, in time to the beat, or even changing in relation to the rhythmic output, by feeding back the number to the choosing selector (Figure 10).

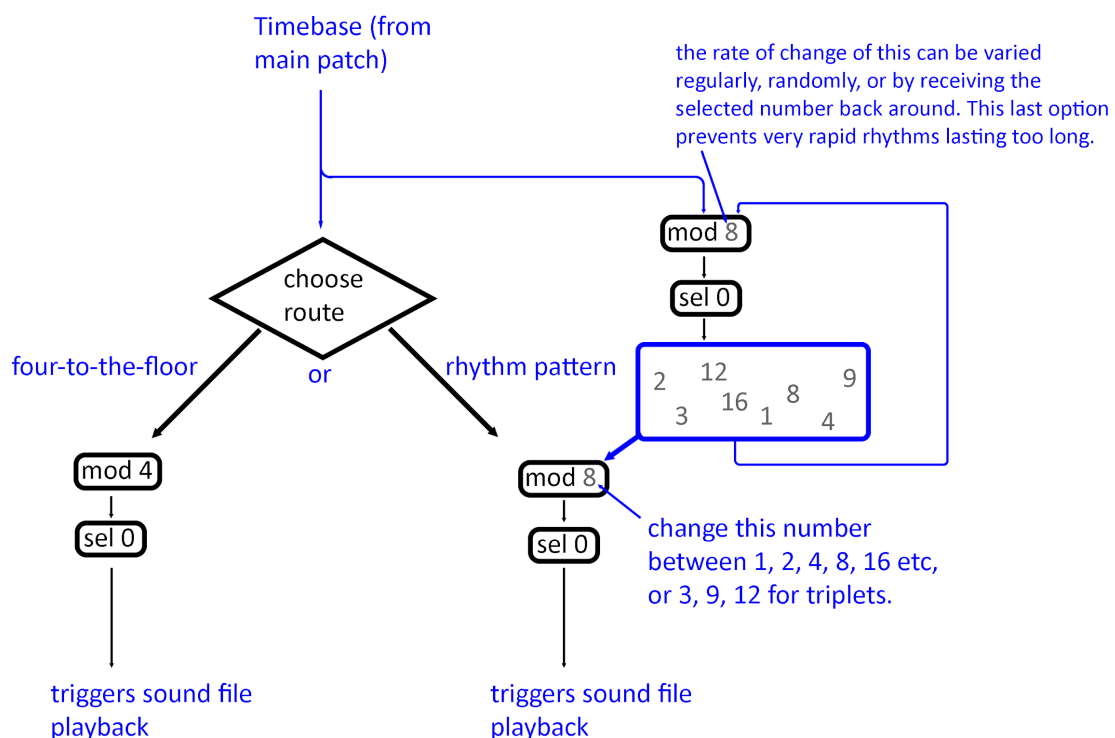
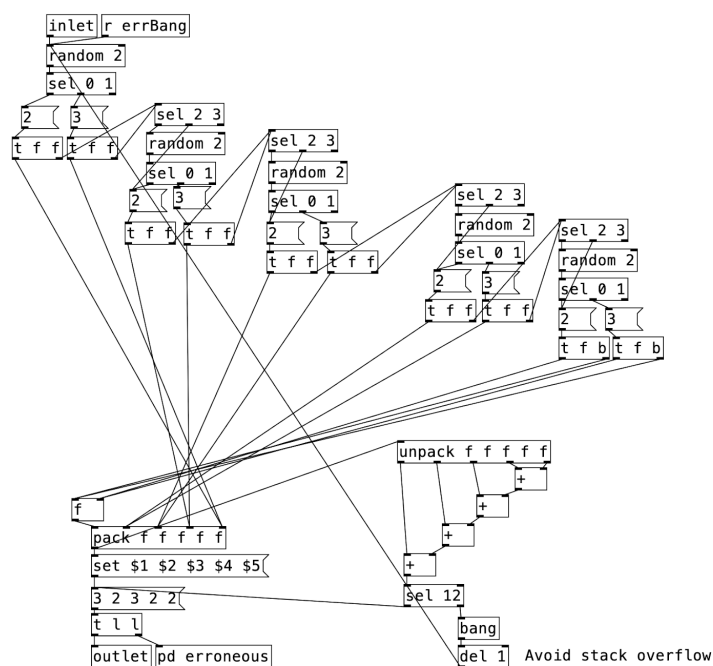


Figure 10 Diagram to show rhythm generator for a kick drum. This type of generator was used in Elerem.

Although these abstractions were all created in response to the sound of my pieces, it is clear that these have parallels with common features in standard DAWs such as Logic, where you can humanise notes in MIDI files for example, to sound more like a human player.

Some of my explorations might seem tangential; in one case, I became interested in exploring the different pentatonic scales and so created an algorithm to play different scales in parallel to see how they sounded against each other (Figure 11).



The octave notes are repeated to emphasise the tonal centre. Complete randomisation doesn't work, because you need to state that chromaticisms can't happen: if a "3" is selected, then the next note cannot be a "4".

On analysing the intervals between the notes, we can see that it is a combination of +2 or +3 semitones between notes. Also, the total addition must add up to twelve, and there must be two +3s and 3 +2s, e.g. +2 +3 +2 +2 +3.

However, the +3s cannot appear consecutively.

This patch produces a set of numbers for creating a pentatonic scale. It doesn't allow consecutive 3s. It checks the output against the number 12, and only pushes out the result if it is equal to 12.

In order to produce the scale, add these numbers consecutively to the root as a midi note.

The only erroneous scale produced by this is 0 3 5 7 9 12, because the 3s are only consecutive if you loop around. Hence a subpatch has been made to deal with this special case called "pd erroneous".

Figure 11 Screenshot of the pentatonic scale generator Pure Data patch.

However, it is often the case that such compositional query explorations can then yield interesting sounding abstractions that can be used in pieces. For example, in exploring how a single click can be varied rhythmically, and in frequency using filters, an engaging rhythmic algorithm emerged, which was used in the piece *Elerem*.

Compositional Process

Stitching many of these systems together to make a piece of music is an organic process that often involves solving small problems or dealing with revealed gaps in a

particular musical parameter, whether pitch range, or timbre, or texture. Often other processes will be brought in from previous experimentation that can fit this void perfectly or be modified slightly to do so. It is also at this point that thought is given to the structure of the work. This can often be heavily tied up with the UX (user experience) of where the music is to be embedded. For example, the pieces for the *Breathing Stone* had to be two minutes long, constrained by the length of the breathing exercises. This was compositionally very restrictive but also straightforward for structure design, as the length was fixed. It is much harder to structure pieces such as *Elerem*, where there is no fixed listening time: structures must be more fluid and more subtle in order to allow for the listener to stop at any time.

Importantly, towards the end of the composition process, some thought is given to making the code as streamlined as possible, both in terms of file size and processing power. This is to allow the app to function in a mobile device whilst taking up as little processing and space as possible. All unnecessary GUI objects are removed, and samples are pared down as much as possible. Sometimes whole “instrumental parts” are removed at this stage, if they are not contributing significantly to the piece as a whole. This is similar to the editing stage of any creative process, where the body of the work is finished, and needs to be refined, and shaped from a macro perspective. It is at this stage that I will move to the xCode software to write the code that will play the PD patch on an iOS device. As this is not strictly a compositional process, I will not go into detail other than to say I relied heavily on tutorials from Professor Rafael Hernandez’s YouTube Channel, and Stack Overflow.⁷⁶ Once I have a working prototype app, where necessary data is flowing between the mobile device sensors and the music, there follows a test period where any necessary tweaks are made for the app

⁷⁶ Rafael Hernandez is a Professor at California State University, and his libpd for iOS tutorials can be found here: <https://youtu.be/uOdt-2Byj20>. Stack Overflow is a public forum for coders and developers: <https://stackoverflow.com/>

to run smoothly and intuitively, including any musical changes. For example, while testing *Elerem*, it became clear that some of the kick sounds didn't really fit with the mood of the piece and were removed. Brian Eno talks about this stage lasting a long time on his website:

When I make a piece like this, most of my time is spent listening to it for long periods - sometimes several whole days - observing what it does to different situations, seeing how it makes me feel. I make my observations and then tweak the rules. Because everything in the pieces is probabilistic and because the probabilities pile up it can take a very long time to get an idea of all the variations that might occur in the piece.⁷⁷

Technical Equipment

Pure Data was chosen as the underlying music programming software initially because of the composer's familiarity with it. It is relatively easy to combine PD into other software architectures via the libpd library, allowing for the composer to have more insight into the physical and practical applications of the connection between external factors and the music. The embedding of PD via libpd was made possible by Peter Brinkmann and others, 'Embedding Pure Data with Libpd', in *Proceedings of the Pure Data Convention* (2011), ccxc <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.2510&rep=rep1&type=pdf>>. members of the PD community.⁷⁸ **Logic Pro** is a common DAW used for arranging and producing music. Logic was used to record and produce live-sampled sounds and to create sounds by combining library samples to make sound files for playback in PD. **Adobe Audition** shares many of the capabilities with Logic Pro but includes some tools that are useful for manipulating audio. In particular, I used the spectral frequency display to assist

⁷⁷ Brian Eno, 'Brian Eno Discusses Reflection', *Brian Eno* <<https://brian-eno.net/reflection/>> [accessed 8 April 2021].

⁷⁸ Peter Brinkmann and others, 'Embedding Pure Data with Libpd', in *Proceedings of the Pure Data Convention* (2011), ccxc <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.2510&rep=rep1&type=pdf>>; P. Brinkmann, *Making Musical Apps*, O'Reilly and Associate Series (Sebastopol, CA, 2012).

with several different decisions, and to resolve questions such as: does a single sound that is required to fill a sound space, such as a pad, contain enough frequency span across the audible spectrum? **SoundHack** is a piece of software that manipulates audio files, in various ways, including vocoding, convolution and time-stretching.⁷⁹ It is very useful for producing sounds with interesting timbres that can be played in PD. **xCode** was used to embed the work into an iOS app. It was used to build a prototype that was trialled repeatedly to check that the sensor data, mappings and ranges made sense, then refined to make the final app.

⁷⁹ SoundHack is an open-source audio editing tool: <http://www.soundhack.com/>

Chapter 3 - Compositions for the *Breathing Stone* (June 2015)

Link to recordings: <http://bit.ly/stonemusic>

Chelsea & Westminster Hospital Stone

(Seven tracks of procedurally-generated music each lasting two minutes)



Figure 12 *Photograph of a Nurse with the Breathing Stone at the Chelsea and Westminster Hospital.*

My brief for the *Breathing Stone* in this hospital context was to compose relaxing music that would be different with each use of the stone, and that would guide the user on a deep breathing exercise, which has been proven to lower blood pressure and also reduce

the measurable symptoms of stress.⁸⁰ In addition, all the breathing exercises were to be twelve breaths long, with five seconds on the inhale, and five seconds on the exhale, following breathing exercises described in early therapeutic research.⁸¹ Two breath samples at the start of the piece acted as a practice breath, and to make the patient aware that the exercise was about to start. The *Breathing Stone* was initially funded by the AHRC within the REACT Hub Sandbox programme. It is a handheld device capable of taking cardiovascular readings while teaching patients diaphragmatic breathing. Its purpose is to help the patients manage symptoms such as pain, nausea, and anxiety. Music plays a unique role within it, as biofeedback to connect the user to the interface, but also as the breathing guide. There are electrodes on its surface that capture electrocardiogram data (ECG) from which the heart rate could be sent to PD running on a Raspberry Pi compute module contained within the body of the stone. The *Breathing Stone* was tested at an exhibition at the Chelsea & Westminster Hospital in the Surgical Admissions Lounge (SAL).

In its first prototype stage, the music on the stone was a series of samples composed by Joseph Hyde which were triggered procedurally within the firmware. In subsequent versions of the stone, PD was run within a python wrapper, allowing much more compositional freedom. An audio engine was developed by Robert M Thomas, which could be used to trigger samples procedurally at the individual note level (rather than samples of pre-recorded phrases). This framework provided a common sound palette for all the pieces to be played on the stone, with distinct instrumentation.

It is worth clarifying that the samples referred to in Joseph Hyde's work were samples of composed musical material, as opposed to Robert Thomas's approach of creating samples

⁸⁰ William J. Elliott and Joseph L. Izzo Jr, 'Device-Guided Breathing to Lower Blood Pressure: Case Report and Clinical Overview', *MedGenMed: Medscape General Medicine*, 8/3 (2006), 23.

⁸¹ J. F. Beary and H. Benson, 'A Simple Psychophysiologic Technique Which Elicits the Hypometabolic Changes of the Relaxation Response', *Psychosomatic Medicine*, 36/2 (1974), 115–20.

of instruments playing individual pitches (either live-sampled, or created using Logic Pro). These could then be triggered individually in the audio engine to allow for procedural variation at the note level. Naming these pitched sound files as their equivalent MIDI numbers enables the audio engine to call up the correct pitch in PD, so that it can be triggered to play at set times, using a global timebase system, to keep all the sample players in time with each other (described in more detail in Chapter 2).

The audio engine's sample collection provided the following instrumentation: "Darkbass", "darksynth", "glassbowl", "hang", "softpiano", and "stringslo". In addition, a choice of three live-sampled inhale sounds and three exhale sounds provided some variation for the breathing sounds playback. Once the music was added, the variations in the music playing concurrently were sufficient to distract from what could potentially be a gratingly repetitive sound, needed to guide the breath.

The heartbeat sounds were the most problematic, as heartbeats are irregular, particularly when a person is relaxed, so there could have been a conflict between the regularity of the musical rhythms against random beats triggered by the heart rate data coming from the ECG. There was much debate within the team regarding heartbeat and how the ECG data could be used in real-time to modulate the music and lighting sequences. I suggested that the heartbeat should be haptic only, as this creates a sensual separation between breathing rhythms and heartbeat. This was achieved by removing most of the high frequencies from a kick sound, so that it would simply move the speaker cone, which is deliberately heavy enough to give a gentle jolt. This way the user could "feel" the connection, without it jarring against the music, whilst also freeing up frequency space in the mix for the other instruments.

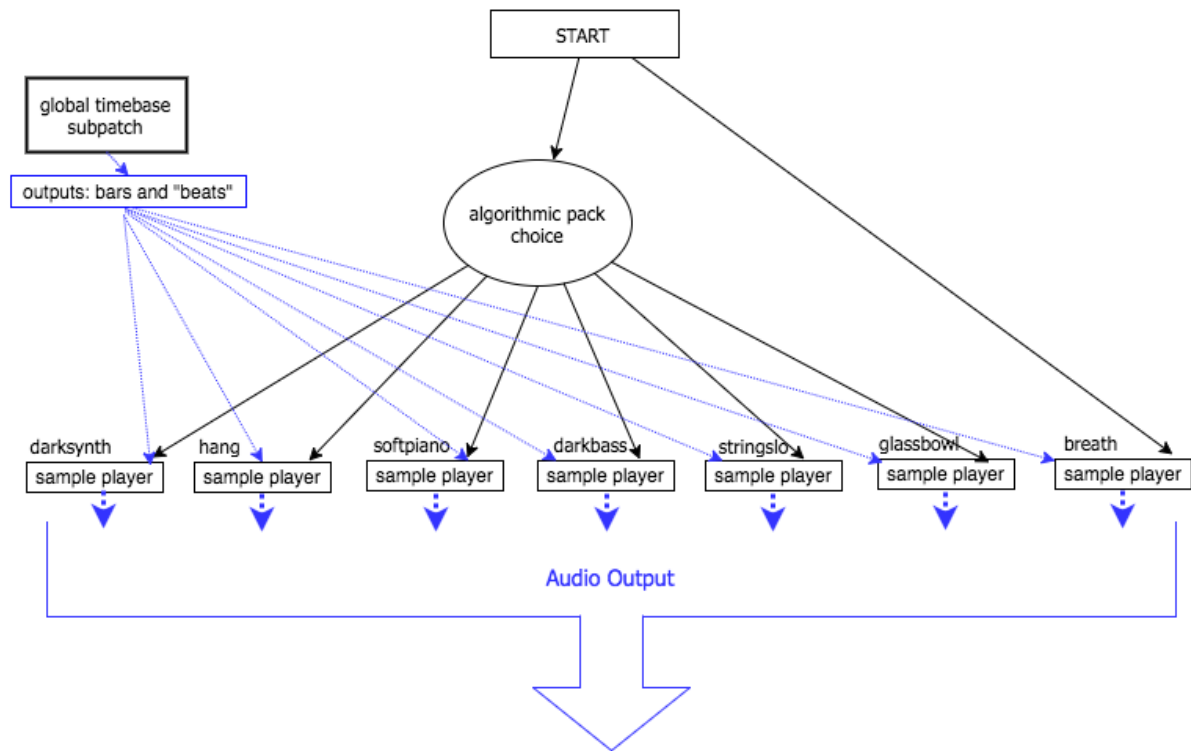


Figure 13 *Diagrammatic overview of Robert Thomas's audio engine used in the Chelsea & Westminster Breathing Stone compositions.*

Compositionally, this was an extremely restrictive brief. The duration of the pieces had to be exactly two minutes, with a distinctive binary feel to the breath in, breath out guide, with the aim to be relaxing music that would appeal to a large tranche of the population. I composed seven 'packs' which acted like scores for the instrument samples within the audio engine that Thomas had created.

In general, I made all the tracks slow-moving, with largely diatonic harmonies. I chose a soft pad for nearly all the tracks, with slow onset to create enveloping sounds that swell in time with the breath pattern (the note samples for the 'pad' instrument are all five seconds long), and act as a constant within the music, then other musical elements are introduced over this bed, to provide interest. My compositional approach to the Breathing Stone pieces

drew on my own personal understanding of the elements of music that are considered relaxing, and also on what I have observed in others, for example, the music choices of yoga and Thai Chi instructors for the relaxation section of their sessions. Elements include using repeating slow-moving harmonic progressions, simplistic harmonic relationships and consonant melodic material, avoiding dissonance and complex rhythms. Some of these elements have been investigated in empirical studies that looked at observed perceptions of harmony and dissonance, and others measured the effects of music on bodily signals such as heart rate and skin conductance.⁸² The harmonic progressions and melodies were mostly composed at the piano, which I then translated into numbers and rhythms for the audio engine. An example of the dataflow in one of the packs for the 'darkbass' instrument in Figure 14 shows how the timebase is controlled to play different collections of pitches in different bars, with a randomisation of the occurrence of one of the pitches.

⁸² Examples include Dave Elliott, Remco Polman and Richard McGregor, 'Relaxing Music for Anxiety Control', *Journal of Music Therapy*, 48/3 (2011), 264–88. and collected studies in Donald A. Hodges and David C Sebald, *Music in the Human Experience: An Introduction to Music Psychology* (2011) pp. 178-190.

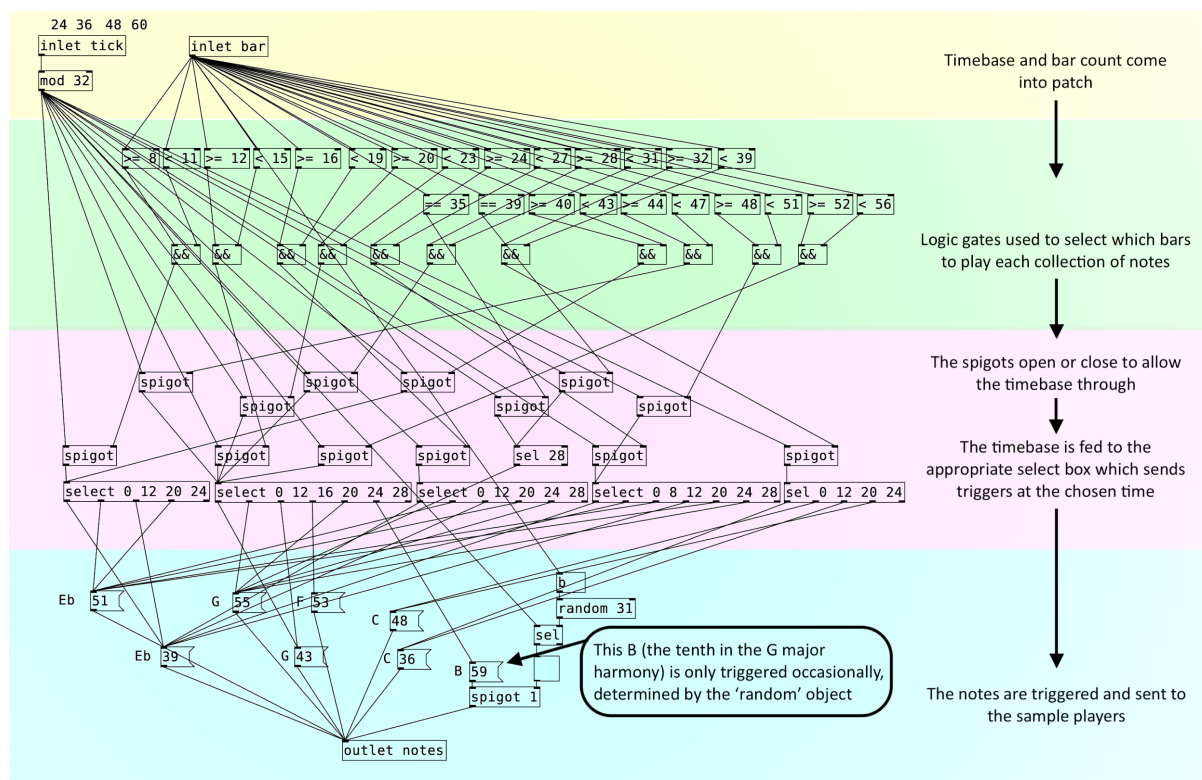


Figure 14 Annotated screenshot of the dataflow for the 'darkbass' instrument in Pack 4 for the Chelsea & Westminster Breathing Stone.

Piece for Two *Breathing Stones* and a Drone

I was asked to write a piece for two *Breathing Stones* and a drone for the REACT Festival in Bristol in November 2015. This was a showcase of the REACT projects organised into "rooms" where a couple of the *Breathing Stones* were to be put into a "breathing room" (the score and instructions are included as Appendix 2).

Unlike the stones deployed within hospitals, where one person could be waiting with the stone for a long time and might use it repeatedly, the Bristol festival installation would have people walking through, potentially only using the stone once or twice. This affected the composition, in that there was less need for the music to sound different with each iteration.

Two chords in the music guide the user when to breathe in and out and, as before, this installation required the stones to each have a breathing exercise lasting twelve breaths, with two practice breaths before the music began. There needed to be more than one stone, to cope with the flow of people coming through the breathing room installation in Bristol.

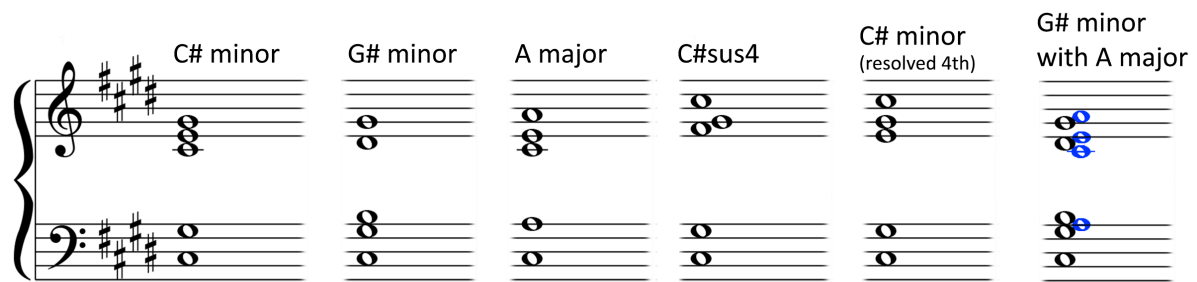


Figure 15 Chords over a C sharp drone used in the Breathing Stones for the REACT Festival in Bristol 2015

Breathing Stone one: C sharp minor, G sharp minor and A major.

Breathing Stone two: C sharp minor, with an alternating F sharp to guide the breathing.

The stones had to be able to work both independently, and together, and both over the drone. This made it very difficult to modulate the music, as it was unpredictable how far along the score one stone would be, before the other one was started. In order to avoid remaining in the same key for the whole two-minute exercise, stone one mostly goes along with the C sharp/G sharp binary chords but does eventually move into an A major chord near the end of the exercise, with a view to breaking the gloomy feel to the minor keys. This still worked over the C sharp drone, but also fit with the C sharp minor chord of the other stone.

The other stone remains mainly focused around C sharp minor, oscillating between the straight chord (the unembellished minor triad) and adding the fourth note of the scale to indicate breath in and out, with melodies over the top played by the hang⁸³ that contain

⁸³ A metal handpan instrument, in this case synthesised in Logic Pro.

some harmonic ambiguity, but still oscillate between two strong tonal centres. The second stone, with the least harmonic movement, plays snippets of melody with random note elements that make the melodies sound different each time.

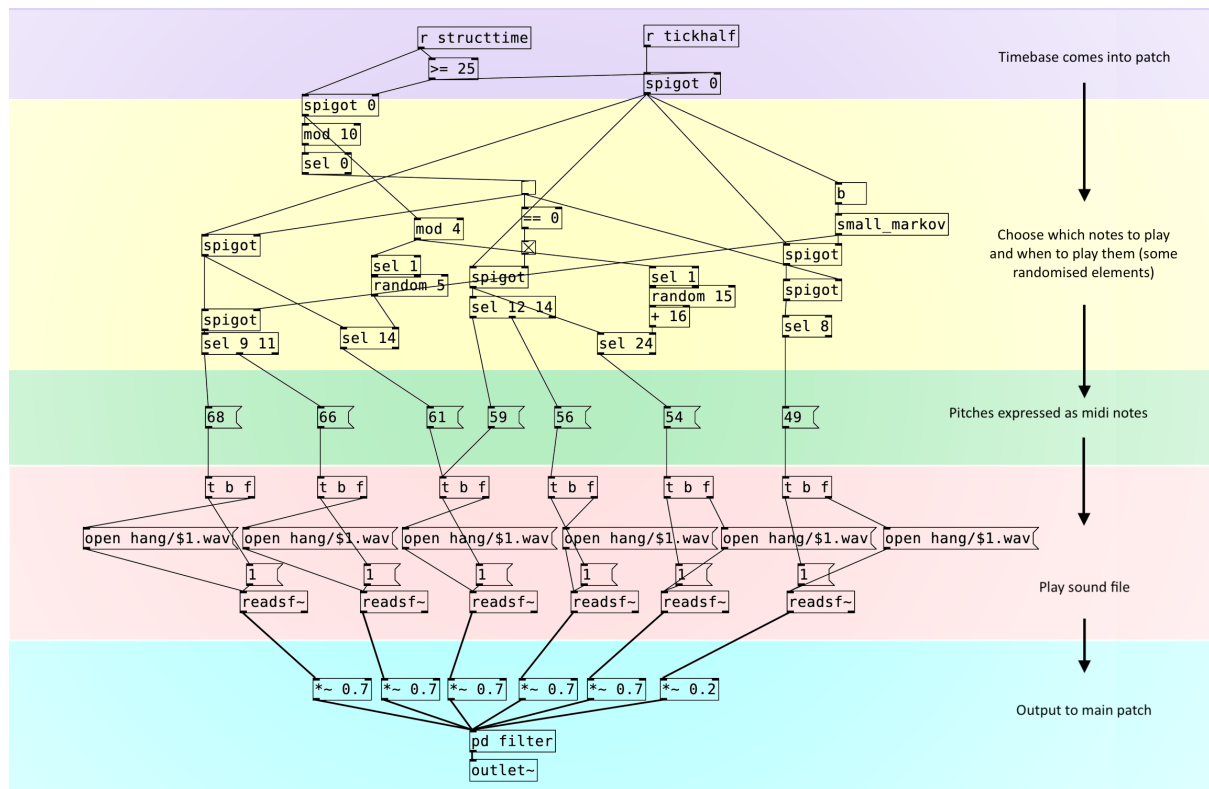


Figure 16 Annotated screenshot showing the dataflow of the Hang-player subpatch in Piece for Two Breathing Stones and a Drone.

Figure 16 shows the melody notes played on the hang, and how they are chosen. Although the notes are represented by MIDI numbers, MIDI is not being used, it is simply a convenient way to label sound files and call them up when needed. The number 68 sounds as a G sharp, 66 is an F sharp, 61 is a C sharp and so on.

Although I kept the hang sound, I modified the audio engine to play samples that I created myself in Logic Pro. These were synthesised sounds made by layering some of Logic's own modules. These 'pad' sounds perform the slow chords, with a second, deeper bass sound adding a greater range from halfway through the piece, in line with the overall

evolving shape. In the second stone, the melody notes are picked out from guitar-like samples that I created with Logic Pro.

The *Breathing Stone* was a good introduction into the generative music process, where I developed the audio engine for the Bristol stones installation, incorporating more of my own dataflow, from what I had learned from studying Robert Thomas's original engine. It allowed me to write not just dataflow 'scores' for his engine, but to create my own engine. This became a springboard for all the generative music programming in subsequent pieces in this portfolio.

The music composition for the *Breathing Stone* has parallels with Robin Rimbaud's (also known as Scanner) compositions for the Philips Wake-Up Light (2009) where the music composition was bound up in the aesthetics of a physical device, and where music was a component (to a lesser or greater extent) of the overall experience.

The stone's popularity at the Chelsea & Westminster Hospital indicates that procedural music could be used effectively in therapeutic devices: one of the visitors at the Breathing Rooms exhibition described her son's reaction to the Breathing Stone:

Thomas, who is very physically limited by his cerebral palsy very much enjoyed lots of the rooms. In particular the *Breathing Stone* was very therapeutic for him. Music and lights are very motivating for him (despite the fact that he has limited hearing and some visual impairment). He was able to hold the stone very well and it really relaxed him.⁸⁴

⁸⁴ The visitor was happy for us to use her feedback but wished to remain anonymous.

Chapter 4 - Interactive Apps

In this chapter I will discuss in more detail the pieces in my portfolio that have interactive elements, *Ominator*, *Bespia*, *Nautilus*, and *Elerem*.

Ominator

Ominator is an iOS app that generates music adaptively by listening to vocal input, augmenting their sustained humming (which the app instructs them to perform). It developed as a result of working through many iterations of different exercises, where I was trying to develop richer sustained pad sounds using synthesis techniques in PD, following Thomas Heatherwick's concept of "critical making" whereby the processes were tried in evolving iterations until a work was created.⁸⁵ Early sketches for this work included *ToneTweak*, an app created to enhance open string playing for beginners learning to play violin.⁸⁶ Harmonically, the app was an exploration of appealing chord combinations around a single note, effectively harmonising the open string into a piece of music. There were many things that were unsatisfying about this early version, but the two main issues that I chose to improve are that the user has to tell the app which string is being played by pressing a button, and the synthesised sounds sounded thin and underdeveloped.

By working on several iterations of synths, the sonorities used in *Ominator* are much richer and rewarding, than earlier attempts. I adapted synths from Andy Farnell's tutorials as well as other synthesis examples found through internet searches.⁸⁷ By using the fiddle~ object in PD, a "key note" or root can be detected via the device's microphone. (The

⁸⁵ Matt Ratto, 'Critical Making: Conceptual and Material Studies in Technology and Social Life', *The Information Society*, 27/4 (2011), 252–60.

⁸⁶ This app was never published, but video capture of the early prototype can be found here: <https://youtu.be/OZoPZnQwi2w>

⁸⁷ Farnell, *Designing Sound* (Cambridge, MA 2010) and <http://drymonitis.me/code/>

“fiddle~” object analyses the incoming sound and outputs a pitch as a frequency (or midi number)). A lot of work went into the timbral quality of the music, experimenting with synthesised tones. Then layers of activity were added, such as a pulsating synth and a simple generative tune over the top. Ominator harmonises the listener's humming, effectively creating stable, consonant sound around whatever pitch the listener might hum, but leaving harmonic rhythm in their control, in an intended balance between interest and adaptive complexity (with input from the listener).

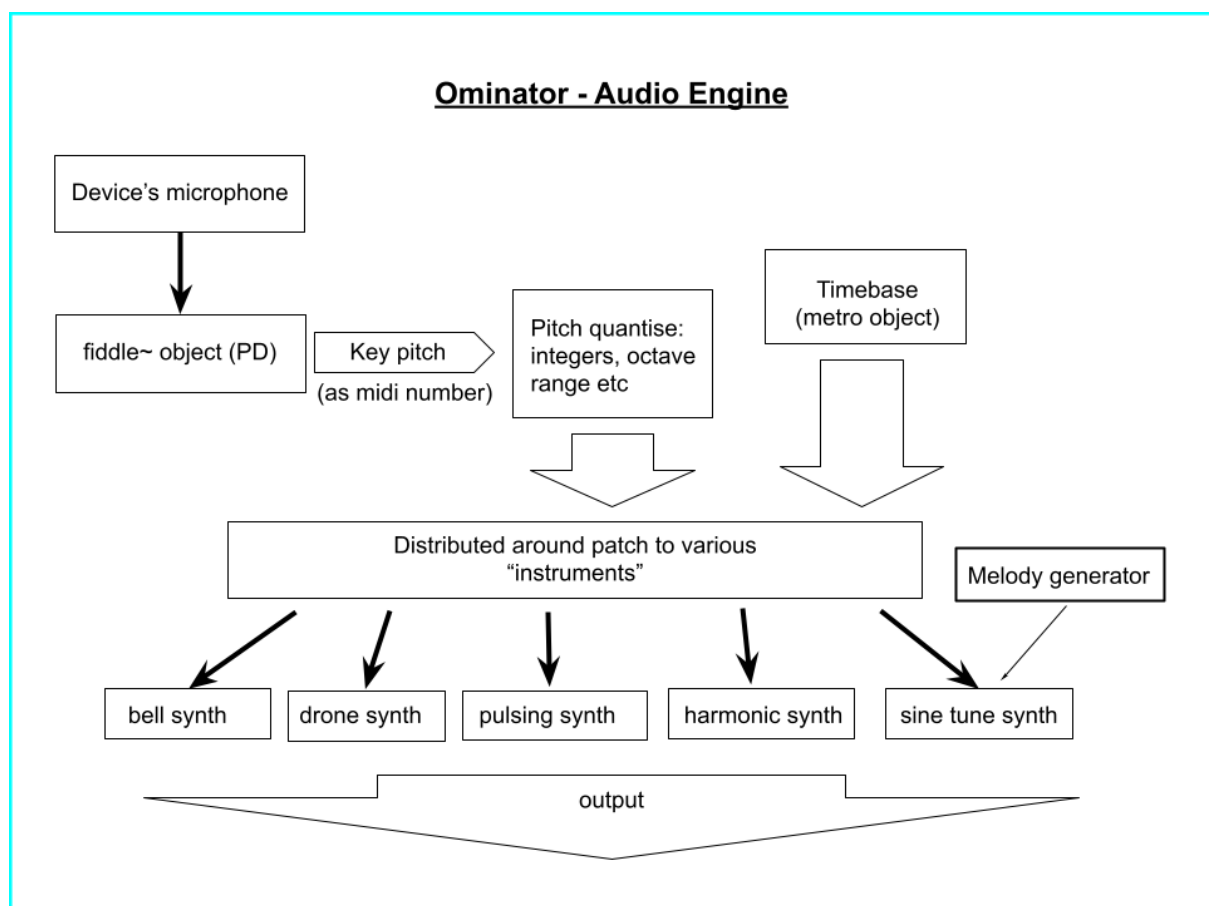


Figure 17 A dataflow representation of the audio engine for Ominator.

The portion of subpatch shown in Figure 17 is added into a patch that also plays the octave and the fifth of the root (as determined by the fiddle~ object) at regular intervals, so that while some of the melody is randomised, there is also some fixed regularity. This is dependent on the input remaining on a static pitch, determined by the user's humming. This

overall generated tune is played on two pure sine oscillators slightly de-tuned, with a fifty millisecond slide between notes, producing a portamento effect, sounding like a high whistle over the droning textures below.

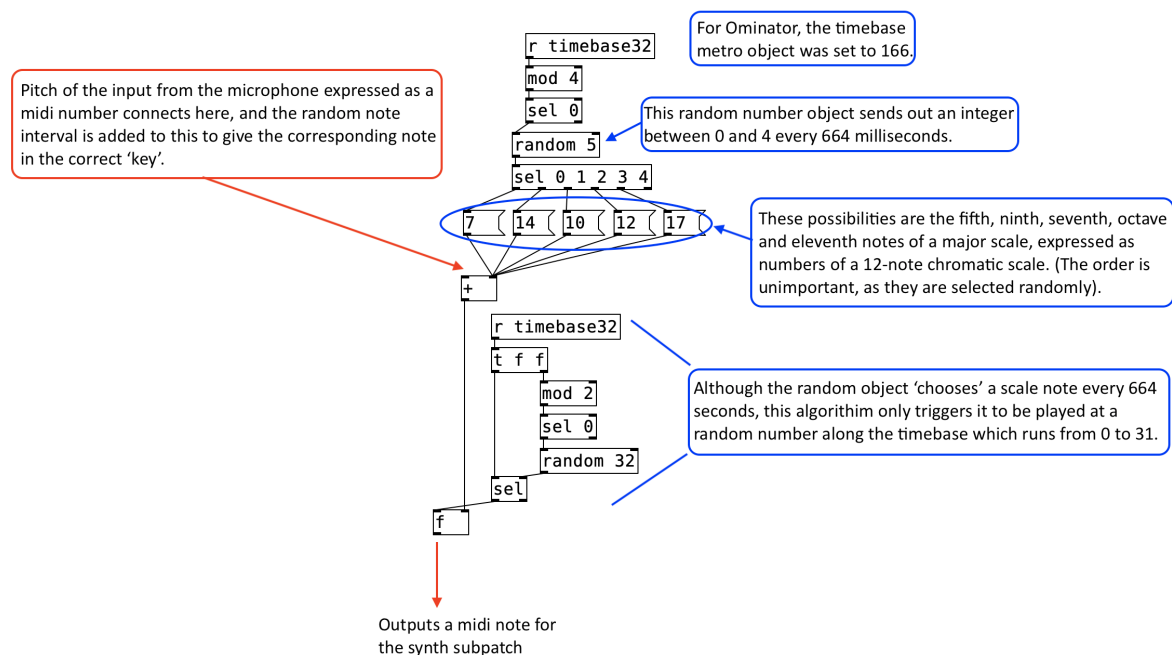


Figure 18 Annotated screenshot to show the dataflow of the random element of the sine melody in Ominator.

One of the limitations of using iOS apps to disseminate music that will be discussed in more detail in the concluding chapter of this thesis is that apps are transient by nature, intended to exist for only a short time by necessity of the rapid changes and updates in operating system and hardware which are typical in smartphone ecosystems. The tools for the creation of apps are constantly updating, and therefore to keep an app viable for usage past the updates of the various components requires a lot of time, an issue not found in fixed media. It is not necessary, as an example, to keep updating or rewriting a symphony just to enable its continued existence. Many apps will eventually simply crash if not maintained, depending on which aspect of the code no longer functions within the new

apple frameworks, or the code libraries that they rely on. In his article *The history of the album-app: creativity galore but commercially tough*, Stuart Dredge remarks on this issue discussing the commercial viability of album-apps: ‘These apps were software that could (and, indeed, would) be broken by operating system updates from Apple and Google – leading to their removal. That’s a shame: as a part of this industry’s history, album-apps are disappearing in a way that, say, videos from MTV’s 80s heyday are not’.⁸⁸

Ominator is no exception and was released before the major development in the apple framework whereby permissions were needed to grant access to various input mechanisms on the mobile device, such as the camera and the microphone. At the time of writing, *Ominator* still works on an iPhone 6 and iPhone XS, but will not work with bluetooth headsets, which is problematic for the later iPhone models that do not have a 3.5mm headphone socket (though an adaptor can be used into the ‘lightning’ socket). Headphones are essential to use with this app, as the microphone will simply ‘listen’ to the device’s own output from the onboard speakers without them.

I investigated fixing this, with a view to releasing an updated version, but the version of the Open Frameworks library, used for the visual component of *Ominator* no longer works with the current version of xCode, and it becomes easier to start afresh with newer, or updated libraries, so a simple update to fix one problem is not possible. As such, *Ominator* is a good example of the impermanence of music-as-app as a medium through which to disseminate music that is at least partially generated by algorithmic processes (and thus needs use of a computational framework for its logic layer).

I would also like to reintroduce the chordal progressions that were present in the original *Tone Tweak* prototype, which were lost when this developed into a meditation app,

⁸⁸ Stuart Dredge, ‘The History of the Album-App: Creativity Galore’, *Music:|Ally*, 2016
<<https://musically.com/2016/07/25/history-album-app-creativity-commercially/>> [accessed 6 April 2021].

as well as to introduce microtonal possibilities, rather than the integer MIDI quantisation that is currently implemented.

Ominator, as with other app efforts in the course of this PhD portfolio, allowed me to develop composition processes and techniques that are applicable beyond their deployment within the mobile operating system for particular app objectives. These are applicable in building music systems/compositions in any framework that needs to generate music dynamically and adaptively, such as in game engines, and therefore are an important and valuable part of the learning and development trajectory of my career as a composer.

Bespia

The music for *Bespia* aims to provide a relaxing audio training-ground for interoception.

Interoception can be defined as the sense of the physiological condition of one's own body.⁸⁹

As such, interoception is a key, yet understudied, aspect of an individual's well-being, with compromised interoceptive abilities linked to a variety of clinical disorders. The majority of research conducted on interoception so far has focused on different versions of heartbeat biofeedback tasks.⁹⁰ In *Bespia*, the heartbeat sonifications need to be accurate in time, but the surrounding music must be engaging enough to keep the user interested, and calming enough to allow for concentration. My compositional approach to the music in *Bespia*, as with the *Breathing Stone*, was to draw on my experiences and observations of others' uses of music to manage anxiety, for example listening to music in headphones while travelling.

As music was often used to mask external sounds of busy stations and airports, I chose slow-moving washes of sound across the frequency spectrum (the pad sounds) that would

⁸⁹ A. D. Craig, 'How Do You Feel? Interoception: The Sense of the Physiological Condition of the Body', *Nature Reviews. Neuroscience*, 3/8 (2002), 655–66; A. D. Craig, 'Interoception: The Sense of the Physiological Condition of the Body', *Current Opinion in Neurobiology*, 13/4 (2003), 500–505.

⁹⁰ R. Schandry, 'Heart Beat Perception and Emotional Experience', *Psychophysiology*, 18/4 (1981), 483–88.

block external noises and direct the listener's attention towards a more internal, calming experience. Although my approach was more trial and error, for example adjusting the attack of the sounds as described below, my explorations align with certain ecological theories of perception as can be found in *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning* by Eric Clarke.⁹¹

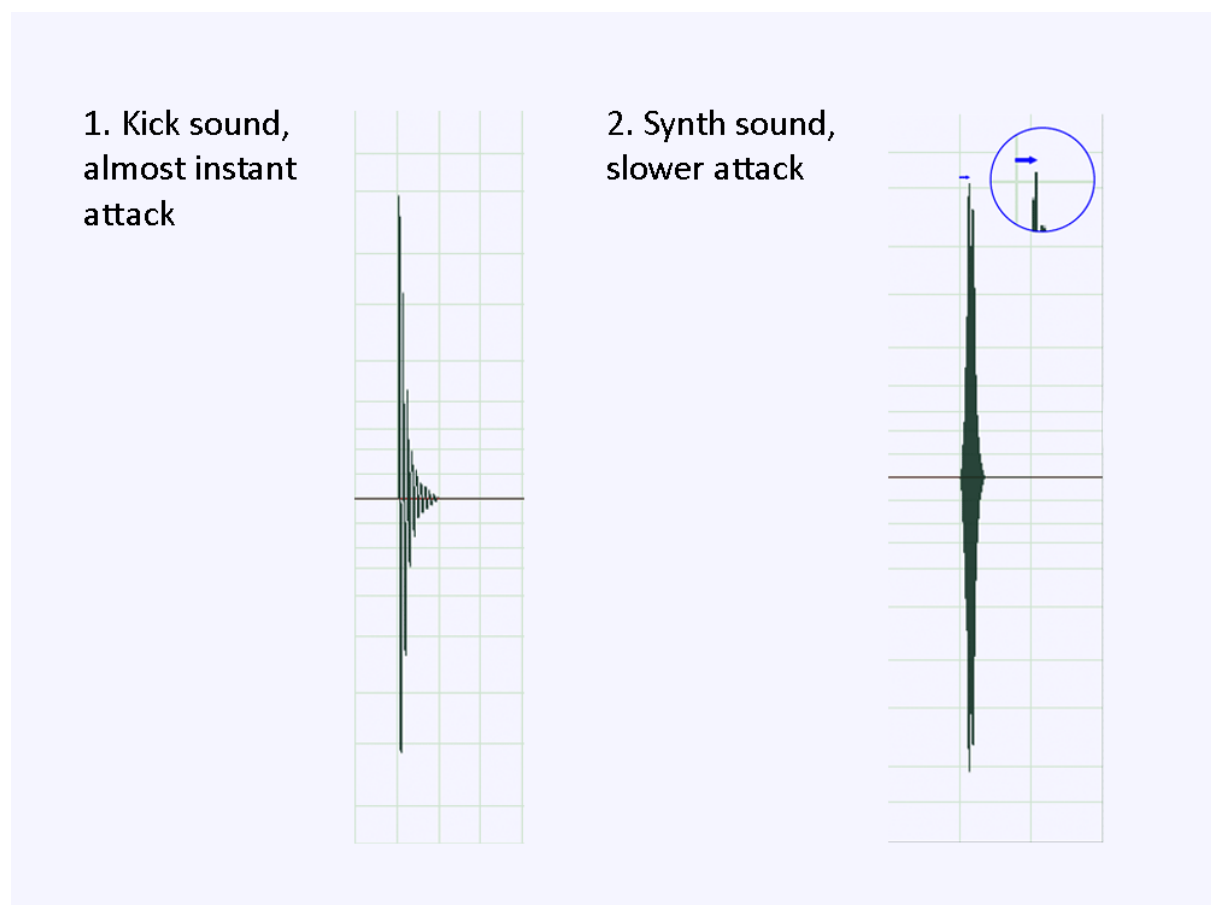


Figure 19 **Waveform images comparing attacks in the heartbeat sounds used in Bessia.**

Through previous work with BioBeats, the concept of sonifying the heartbeat has been explored in several ways. For accuracy of perceived timing, a kick sound with a sharp attack works well. However, this sound can quickly become harsh and grating; uncondusive to a wellbeing app where mental relaxation is key. Sounds with a slower attack, which are

⁹¹ Clarke, Eric, *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning* (Oxford, 2005).

gentler on the ear were also explored, as shown in Figure 19. In *Bespia* the heartbeat sound is made of a combination of a synthesised sound (a low-frequency oscillator passed through several envelopes) and a drum loop sample that was sped up beyond recognition using SoundHack. The audio engine chooses a random sample from a selection of eight sounds that were created in this way, and also varies the speed of the sample playback to produce slight variations in the sonic quality each time. When combined with the synthesised sound, the sonic representation of the heartbeat is rich in timbre and variety.

In order to immerse the user into the app, and encourage longer use-times, pads are sustained underneath the heartbeat sonifications which are warm, rich, and constantly evolving. Over the course of my doctorate, making drones sounds has been a constantly revisited query, and looking back on my cpu-intensive, thin-sounding attempts in the prototype *ToneTweak* (2014), they now seem a long way from the drones in *Bespia*. First of all, there are several layers, which means simply varying the relative volumes of these layers introduces a timbral change. Secondly, in *Bespia*, they are created by processing sampled sounds. Only one sample needs to be used for each pad, as the pitch can be manipulated through playback speed, which keeps the overall size of the app smaller than when using individual samples of each pitch. These samples are generated in Logic Pro, or use recordings of actual sounds, such as stringed instruments or voice, and then processed.

Variations of *Bespia* have been used in clinical apps for research into the effectiveness of interoceptive (body signal perception) training to reduce stress (as measured by Heart Rate Variability) showing the potential for my compositions to have an impact in clinical therapies. In hindsight, it might have been useful to include more variations within the music, particularly in texture, but also in timbre through instrumentation changes as, particularly with longer usage the unchanging soundscape could become tiring on the

ear. In particular, several different options were available for the heart beat sonification, such as having only the pitched sounds, or only a single note (rather than a dyad) or even just allowing the feedback to be purely visual and haptic. Exploring these variations would allow for greater interest and structural organisation over longer periods of time.

Nautilus

Nautilus was inspired by the creature of the same name seen in a BBC documentary on mathematics,⁹² a mollusc-type creature who lives in the dark in the ocean. This piece intends to create a space of quiet and stillness for the listener that imitates the seashell resonance of a Nautilus, whereby ambient noise from the surrounding environment is transformed by the reverberation and amplification within the cavity of the shell. In the piece, sounds from the listener's external environment are taken through the device's microphone and treated sonically so that they seem distant and non-threatening. The creative concept was to build a cocoon of sound to build a protective barrier against auditory assaults in the real world, to fulfil the idea that instead of hearing raw sounds, it might be possible to have the world 'filtered' to a more bearable soundscape, in effect allowing the listener to travel from an acoustically perceptive state (the outside soundworld) to an interoceptive state (the inside soundworld). This journey from externality to interiority is achieved not just by alienating the sounds through reverb, but by transforming so they become an entirely new soundscape, that is unique to the individual's acoustic environment but also created dynamically for their ears only. In this way, I wanted to give the listener a piece of music that acts as a tool as well as a composition, allowing it to be used for music listening pleasure but also as a prophylactic barrier between the listener and her world when that world is a source of anxiety.

⁹² M. Du Sautoy, 'Numbers (Episode 1)', *The Code* (2011) <<https://www.bbc.co.uk/programmes/b012xppj>>.

The layers of sound, created in subpatches in *Nautilus* consist of:

Instruments:

Wood
Wave
pitchPad

Manipulation of materials:

micRecord
tapDel
reWave
reEcho
Empty

Instruments:

Wood

The wood sound is a synthesised sound that was developed early on in my research as an adaptation of the sword-clang sound in Tony Hillerson's *Programming Sound with Pure Data*.⁹³ It consists of a sine wave oscillator processed through different attack and decay envelopes.

Wave

This is a synthesised sound I made using a combination of detuned sine wave oscillators and a sawtooth oscillator. It has a slow attack so that it appears softly, in contrast to the more percussive sound of the 'wood' synth. When it is triggered, it's sound is recorded in a buffer so it can be replayed at random times, bringing in some repetition, and familiarity to the listener.

PitchPad

This sound was created in Logic Pro, combining several virtual instruments, intended to make a sound reminiscent of people humming. The sample will be randomly triggered to play at different speeds to alter the pitch and timbre of the sound. This sound can also be

⁹³ Hillerson, *Programming Sound with Pure Data*.

triggered deliberately if the pitch-detection object (sigmund~)⁹⁴ detects a pitch sustained for more than one point three seconds, for example someone humming into the microphone. This is not made explicit in any instructions to the user, as it is intended as something to be discovered on experimenting with the app.

Manipulation of materials:

micRecord

This algorithm is key to the sound of the app. This records the incoming audio into a series of buffers that are then replayed at different speeds. These replays are also constantly panning around, giving the impression that these sounds are moving past you.

tapDel

This is a simple tap delay with four taps which are randomised to occur between one and thirty seconds after the actual sound, at various diminished volumes. This delay sits on the mic input, so everything detected by the microphone will be processed in this delay line.

reEcho

ReEcho constantly records the last five seconds of all the output, every five seconds. This includes all the various input sounds, delayed sounds, and replayed sounds (at their various speeds). It is used for the 're-Wave' trigger described below.

reWave

This plays the sound recorded in the wave buffer (mentioned above) whenever the 'empty' trigger is received (described below). The output of this playback is sent through another sigmund~ (pitch-detection object) and this pitch is fed to the pitchPad playback

⁹⁴ Sigmund~ is an object in PD that detects the fundamental frequency of audio coming into its inlet.

after a delay of five seconds. This means the pitchPad sound will be triggered to play the pitch of the last 'wave' pitch stored in the buffer.

Empty

I wanted a way to keep some sound going, even if there was very little being detected by the microphone, so I made an algorithm that 'listens' for silence. After twenty seconds of no sounds above a fixed threshold, an 'empty' trigger is sent, which triggers the reWave table to play, which in turn will trigger the pitchPad sound. If there is no sound (above a threshold) after forty seconds, the 're-echo' will trigger, which plays back the reEcho buffer at one of three slower speeds. As all of this sound is recorded into reEcho, there is potential for something to sound every twenty and forty seconds, even if there is no input from the microphone.

Discussion and future development

Nautilus's output is put through a long reverb to create a wash of sound, and to remove the 'directness' of the input, enhancing the cocooning effect that was my objective. I used a vanilla object, rev3~ for this, which has a distinctive sound. The most significant setting for this object is the "liveness" setting which was set to 98%, almost maximum, but allowing headroom to prevent any overloading, and to allow for older sounds to die away (albeit slowly). The resonance of this effect is rarely felt in nature other than perhaps in caves which contributes to the 'cocooned' feeling. In future implementations of this type of sound cocoon, it would be good to try other reverb objects and paradigms, if they can work within libpd to experiment with different sounding spaces.

This piece is very sparse and spacious, partly inspired by Pauline Oliveros's theories on "Deep listening" and "Sonic Awareness"⁹⁵ and in particular, Von Gunden's observations on *Meditation IX The Greeting* where he notes that:

'...musicians who are not singers are hardly aware of their voice and often find singing embarrassing. This lack of attention to the natural voice is a pity, and it is fortunate when a musician discovers his or her own voice by accident. I think that when Oliveros wrote "The Greeting" she was beginning to realise that finding one's voice was important.'⁹⁶

This in particular inspired me to encourage humming / singing into the microphone, triggering the pitchPad to sound if a pitch is sustained for more than a second, in order to bring focus to the listeners voice in an amplified, but also a performative way. With this in mind, this piece responds to vocalisations, encouraging input, which is then modified and replayed to the listener.

Elerem

Elerem is inspired by the tracks I was listening to while running to improve my fitness and well-being. I found that I needed the driving beat of electronic dance music (EDM) to keep me moving, to fight the desire to stop, and wanted to bring elements of this listening experience into my own composition practice. This developed into a piece of adaptive music designed to be affected by the listener's movement. It consists of various musical elements that are layered to make fuller or sparser textures.

The music takes in data from the phone's accelerometer to trigger changes in the texture and feel of the piece. Example video: <https://youtu.be/7MwICZG4ilk>. It assesses "activity" on three levels: still, active and very active:

⁹⁵ P. Oliveros, *Deep Listening: A Composer's Sound Practice* (Bloomington, 2005).

⁹⁶ Heidi von Gunden, 'The Theory of Sonic Awareness in the Greeting by Pauline Oliveros', *Perspectives of New Music*, 19/1/2 (1980), 413.

Very Active: if there is constant movement, the kick drums will play four-to-the-floor rhythm patterns, to accompany you as you are walking, running, or dancing.

Active: if movement is sporadic, the drum patterns are varied, creating an upbeat feel, to provide interest.

Still: if the listener is motionless, the beats gradually fall away to a lighter rhythmic pattern, eventually falling away, leaving only calming mallets and vocalised pitches, with swirling pads in the background.

***Elerem* - Composition**

Elerem, like *Cantoo*, has existed in various forms since almost the start of the portfolio-building process, and has evolved along the way. As in *Nautilus*, *Elerem* has musical layers, but in *Elerem* these can be likened to ‘tracks’ used in a DAW to make EDM music: there is a drum machine, a pad, an arpeggiator, a vocal component, and then a synthesiser giving mallet-type sounds. Some of these layers are described below.

Arpeggiator

This is set up to play notes selected from the Dorian scale on the root note in triplets. The broken triad varies by selecting random notes from the scale, changing one note in the triad at a time (to give the effect of the chord constantly evolving).

A root note is sent around the whole patch (as a midi number), from which all the other notes are calculated. Initially, this was fixed at D flat, as the vocal sample I used was ‘ooh’ sung to a D flat. Later I amended the patches to allow for changes in key, adding much more scope for variety in the music.

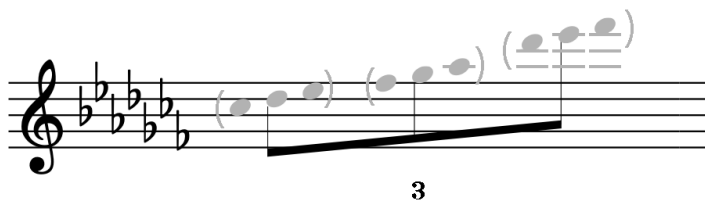


Figure 20 **Scored representation of triplet note possibilities in Elerem.**

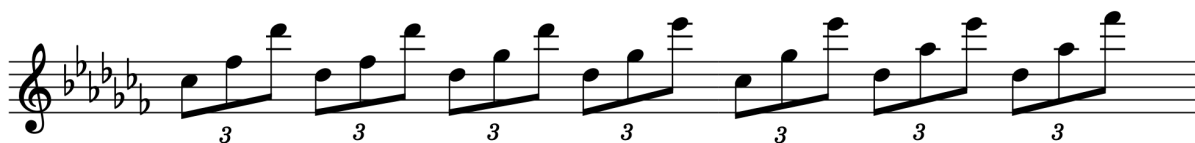


Figure 21 **Scored representation of examples of arpeggiator choices in Elerem.**

Vocalisations:

This element is an example of how to make the most of digital signal processing in small devices, where space is restricted. A single sample (a raw WAV file lasting three seconds and using 300kB of storage) of a female voice singing a single tone (D flat) is played back at different speeds to change the pitch, from which a range of pitches then becomes available. This method of pitch changing, rather than time-stretching (which tries to preserve the timbral qualities of the sound and involves removing or looping minute samples of the sound) introduces timbral artefacts similar to those originally made using magnetic tape recorders by varying the playback speed. Within the context of electronic instrumentation, the artefacts produced by this method of playback compliment the style, reminiscent of the manipulation of the opera singer Inva Mula's voice by Eric Serra for the film *The Fifth Element*.⁹⁷ In *Elerem* the pitches for the vocal sample are chosen randomly from the dorian scale on the root note, to fit diatonically with other elements in the piece. The sample is played more often when the app's 'still' state is triggered, and less often when there are other more active elements being triggered (such as the drum machine).

⁹⁷ E. Serra, *Diva Dance* (1997).

Mallet Sounds:

This element of the piece is a sine wave oscillator synthesised using amplitude modulation. In early stages of the composition, this was always a fixed pre-composed phrase, but due to limitations in processing power (explained below) this was later developed so that the pitch and rhythm of these sounds are controlled by several algorithms, as described by the diagram in Figure 22.

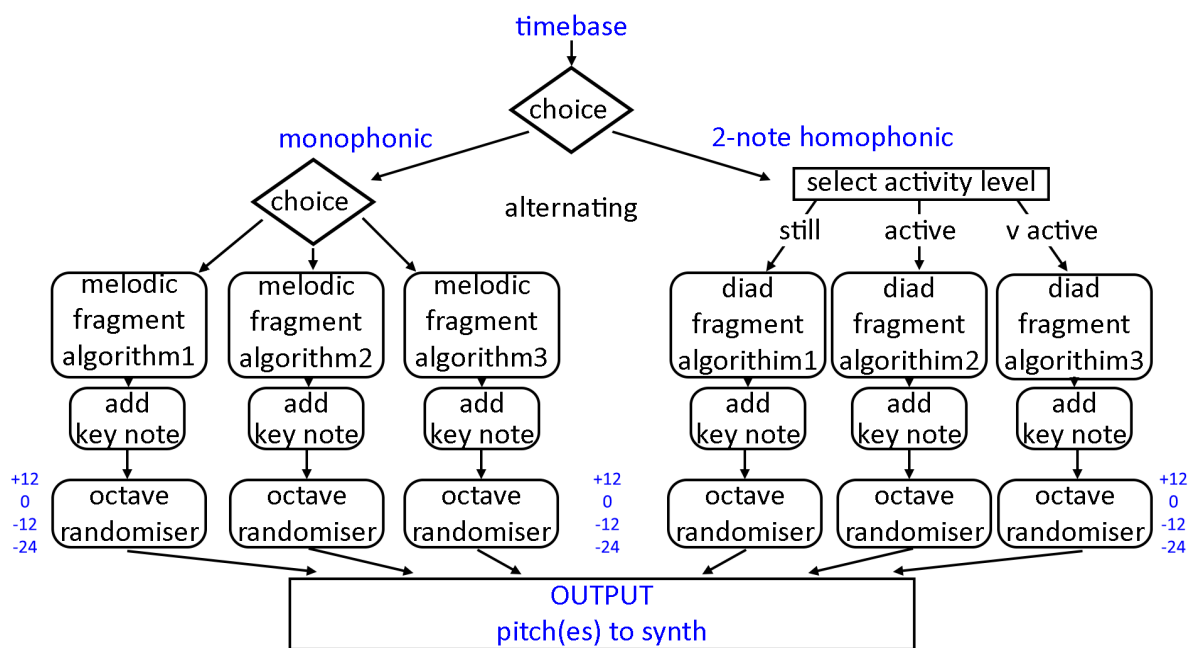


Figure 22 **Diagram to show the dataflow of the mallet sound algorithms from Elerem.**

The ‘choice’ elements in the dataflow are also chosen algorithmically: the main choice at the top of the diagram alternates every time there is a ‘song change’ so that a contrast is made between monophonic melodic lines and chordal (two-note) patterns between ‘songs’. A ‘song change’ message is sent around the whole patch every three minutes, to change several elements including the samples in the drum machine, the root note, the delay timings and others, to create a different ‘version’ of the track. The second choice box in the diagram is chosen by a random object. An example of the algorithms that determine the actual notes and rhythms can be seen in Figure 23, with the rest documented

in Appendix 3. The mallet sounds are also processed with a multi-tap delay,⁹⁸ the timing of which is also modified with each ‘song change’ message, adding another layer of rhythmic variation.

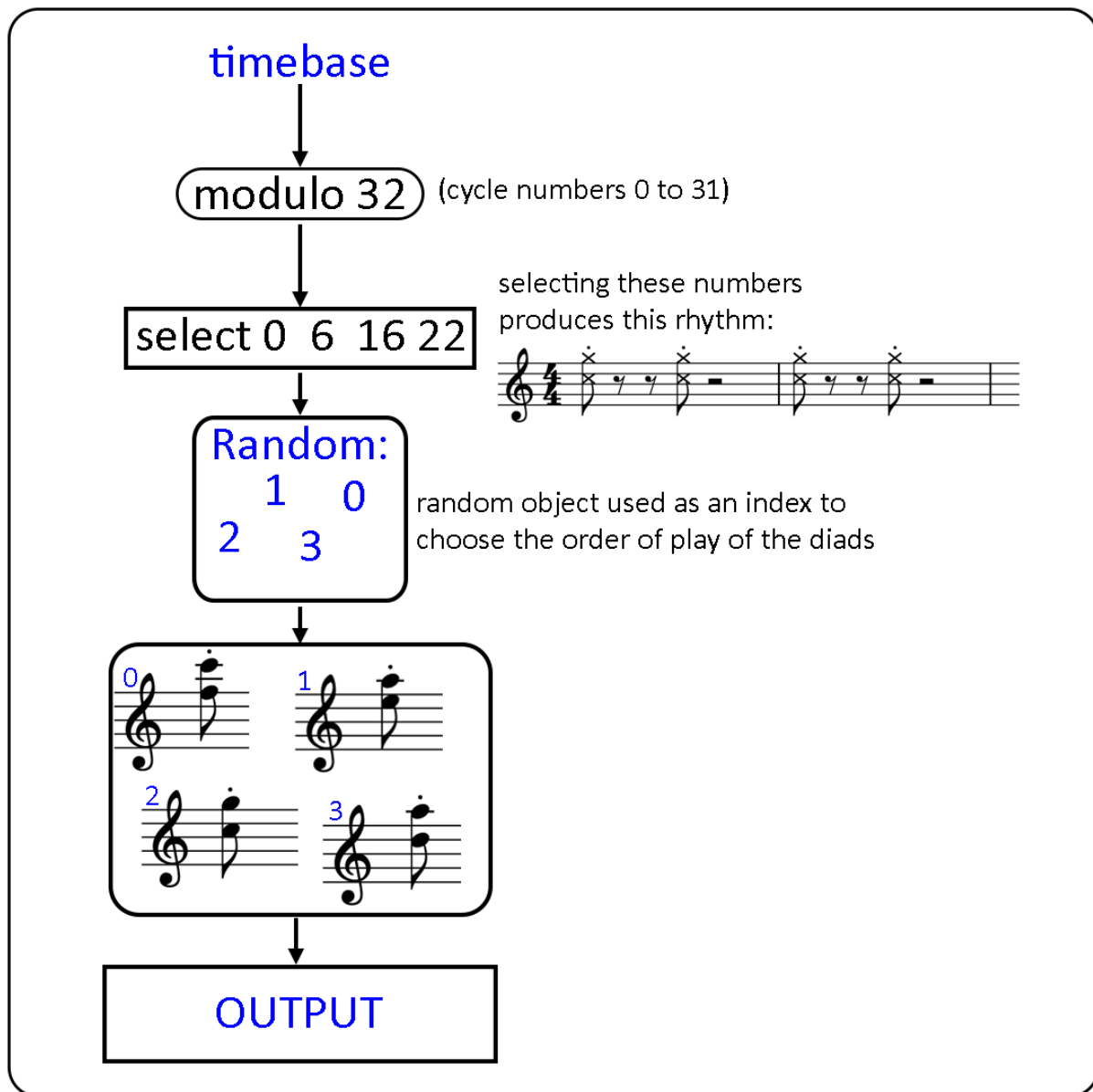


Figure 23 **Diagram to show an example of a two-note chord-choosing algorithm in Elerem.**

Limitations

A significant development occurred in 2019 after I developed a synth in PD that uses voice samples and is much richer than previous sounds. I also developed a drum machine

⁹⁸ Multi-tap delay is where several instances of the delayed sound can be played at set times for rhythmic effect, like having several ‘read heads’ on a magnetic tape player that can be spaced along the tape.

that plays loops, and sections of loops rather than just one-shots (a term used to indicate a single sound activated by a trigger, for example the sound of a snare hit, as opposed to a loop which can include a variety of sounds within a rhythmic pattern). I had intended to merge these richer sounds into the established structure of the original *Elerem* sketches. However, once this was all compiled and tested on an iPhone 6, it became clear that the processing power on the device was unable to cope with the amount of DSP being used to realise the piece (the music stuttered and suffered 'drop-outs'). At this stage in the composition process, I stripped out many of the layers of musical activity, and developed more variety within the remaining layers, for example in the mallet layer, as described above.

In *Elerem*, I used the 'coreMotion' library by Apple with the methods given in the API to pass the raw accelerometer data into PD. My attempts to funnel this data into my music involve only very basic mapping and would benefit from expertise in the areas of pace-detection and coding. Jeska Buhmann provides much more detailed and scientific explorations into movement with music in her thesis *Effects of Music-based Biofeedback on Walking and Running*.⁹⁹

⁹⁹ Jeska Buhmann, 'Effects of Music-Based Biofeedback on Walking and Running', 2017 <<https://biblio.ugent.be/publication/8540383>>.

Chapter 5 - Procedural Works

In this chapter I will discuss in more detail the pieces in my portfolio that are purely procedural in nature, and do not have any interactive features, *Montague* and *Cantoo*.

Montague

On attending a *Minimalism Unwrapped* session at King's Place in London in January 2015, I was particularly inspired by one of Stephen Montague's pieces performed live on the day, *Paramell Va*. I was attracted by the textures produced by the piano scoring, but also its harmonic content. The piece begins with alternating chord clusters at a rapid pace, which gives a static timbre but this is then varied by changing the dynamics and articulations (using the sustain pedal). Grand gesture chords spread to the ends of the piano's pitch range begin to be interjected towards the middle of the piece, which then builds to a loud climax with these loud chords at the extremes of the piano's range played in the same rhythmic pattern as at the start of the piece.

I recorded my own sample bank of piano notes that could be triggered individually in PD, to create a piano-player style instrument. I liked the idea that the samples would capture the particular tuning of the piano at that moment, and also the mechanical sounds around the piano key action.

In August 2016, Olafur Arnaulds and Nils Frahm released *Trance Frenzy*, which is an improvisation on various keyboards, producing a distinctive soft piano sound by close-mic recording, and normally extraneous sounds such as the pedal movement, the performers adjusting their position on the seat, or humming along, all clearly audible. (This sound is previously discussed in Chapter 2.)¹⁰⁰ The soundworld sits very much in the modern ambient

¹⁰⁰ Arnaulds and Frahm, *Trance Frenzy* (Erased Tape Records, 2016).

soundscapes of artists such as Max Richter and Jon Hopkins, which themselves sound like evolutions of earlier piano styles of Graham Fitkin's *Piano Pieces* or Philip Glass's piano works from the 1990s, as well as drawing inspiration from Morton Feldman, John Cage, and more recently, David Lang, Rachel Grimes, and Matti Bye.

Montague is a set of three pieces:

Montague I

Stephen Montague's *Paramell Va* consists of fast repeating cluster chords alternating between the right and left hands with subtle variations in pitch, providing a constant sonority within which the variation of dynamics and articulation is explored. The harmonic content of the piece moves quite slowly, its repetitive nature reminiscent of trance tracks I heard in clubs in the nineties and reminiscent of other pieces with this type of texture including *Music for Eighteen Musicians* by Steve Reich and *Juanita:Kiteless: To Dream of Love* by Underworld.¹⁰¹

My objective in *Montague* was to move harmonies gradually through a static sonority, emulating the textures mentioned above, but with ever-evolving possibilities. I wrote a chord palette keeping the harmonies as open as possible, including mostly ninths and thirteenths (see Figure 24).

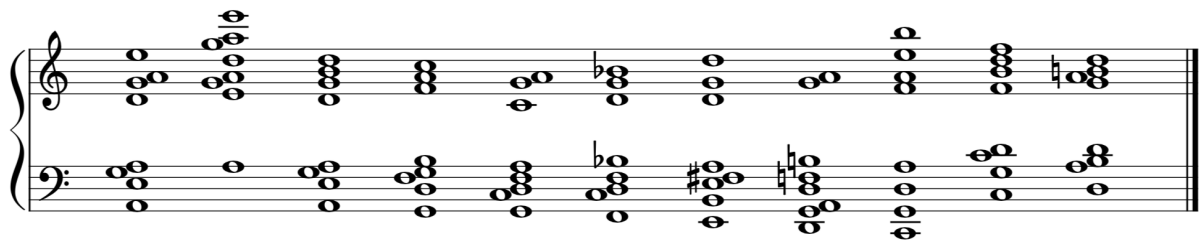


Figure 24 *Scored representation of the chords used in Montague I. These can be played in any key, in any order.*

¹⁰¹ In the Underworld track, the texture is first heard at 3'40" and heard most clearly from 12' 11". *Music for Eighteen Musicians* (1978); Underworld, *Second Toughest in the Infants* (1996).

I wanted to elicit a sense of focus by breaking out the homophonic chords into a right-hand/left-hand repeating rhythm, as occurs in *Paramell Va*. Danish neuroscientists have explored the feeling of focus and joyful play that playing triplets against duplets gives, a feeling I recall from learning to do this as a child; euphoric that I could play what felt like a contradiction between my own two hands.¹⁰² The triplet against duplets feels comforting, and the steady beating between the two hands where the listener can choose to single out one hand or to listen to them both in a more complex pattern seems to focus the mind. Moving through the chord selection randomly is programmatically simple, but I wanted better variety in the chord structure rather than just the set I had chosen, so I also added an algorithm that would change one of the notes in the chord either up or down by a tone before moving onto the next chord, in order to blur the harmonic language and create more complex pitch groups.

Once the piece was at the stage where it could run infinitely, I felt there was still a lack of detail in some frequency ranges. To solve this, I added digitally modified piano sounds in the bass range; a reversed sample of a long low piano note relevant to the chord chosen. The sound of a struck instrument played backwards builds and then ends suddenly at the strike. I timed the arrival of the end of the note to occur just before the chord changes to emulate a swell in sound just before the harmonic change.

In the higher frequency range, I added a sample of 'string piano', an extended technique whereby certain notes are depressed silently and sustained with the pedal before sounding the strings with a finger swipe, to produce a specifically pitched sound, rather than

¹⁰² Peter Vuust, Line K. Gebauer and Maria A. G. Witek, 'Neural Underpinnings of Music: The Polyrhythmic Brain', in *Neurobiology of Interval Timing*, ed. by Hugo Merchant and Victor de Lafuente (New York, NY, 2014), 339–56.

a chromatic collection of notes. This provides a sparkling in the higher frequencies and adds a contrasting timbre to the piano chords.

This piece includes the humaniser component, both for volume and timing as described in the Chapter 2. Additionally, I have inserted a small algorithm I call “hole-maker” which randomly blocks or allows a trigger through, to give the chords a more improvised sound, as if a pianist is playing lazily, and not always sounding all the notes in the chord. Finally, the tempo of the piece changes over time, so there is a constant feeling of either being gently pushed to go faster or slowing down to relax. Currently, this tempo change is arbitrary, lacking the prescience of larger musical structures and gestures to link to, that would be possible in a piece of music of finite length. It could be interesting in future iterations to control this parameter in a way that is more integrated with either the music or the user’s experience.

Montague II

In listening to film and game music, it becomes readily apparent how film composers write memorable themes that they then use repeatedly throughout a film or game, manipulating and developing them to generate hours of music. I wanted to understand whether a basic procedural music system could be created for theme generation and development. To do this, I developed a system in PD whereby the musical notes from a theme based on a simple note series would be used to generate chords, and then further material through inversion, augmentation, and retrograde. The rhythmic material is also varied so that a theme will sound with a different rhythmic pattern each time it is played.

I composed a set of note-series as set out in Figure 25. As more themes were written, it became noticeable that they could be ordered according to level of chromaticism, and because of the way the pieces are generated using the notes of the theme, this makes quite

a difference to the overall feel and tone of the music. I wanted to give this control to the listener, choosing the mood of the piece using a slider to select the level of chromaticism. This is the reason the first theme only contains perfect fifths and octave intervals, in order to avoid the possibility of any chromatic intervals at all. The seventh theme in the list is randomly generated using notes from the five different pentatonic scales.

When the theme is chosen, the rhythm of the theme is assigned from a set of eleven predetermined rhythms. Randomly generated rhythms featured in my initial experiments, but these proved to be too ‘non-human’ sounding: random processes often produce rhythms that do not contain the nuanced referential rigour that human musicians and composers normally imbue in them, and which human listeners expect. These were therefore rejected in favour of pre-composed rhythms from a set in the same way as the tonal themes. However, as each theme is not tied to the same rhythm each time it iterates, each theme will potentially sound with up to ten other iterations.

Once the theme and rhythm have been selected, they are repeated for coherence. The variety and interest is provided by the accompaniment, which is separated out into “bass notes” “LH” and “RH”. Then the theme is developed using the techniques described above.

Themes For Montague

Pre-composed themes*

Elise Plans

1

0 7 12 0 7 0 12 7

2

0 5 7 12 10 7 5 7

3

0 3 5 10 5 3 0 7

4

3 5 10 5 3 7 0 7

5

0 12 5 2 7 5 10 7

6

0 7 10 12 14 7 10 12

7

'pentaGen' (Theme generated based on pentatonic scales)

8

12 14 10 7 3 5 9 7

9

8 12 10 7 5 0 2 4

10

12 11 7 9 4 7 2 4

* These examples are given with D as the tonic but may be played in any key

11

12

13

14

15

16

17

18

Figure 25 *Scored representations of pre-composed themes for Montague II. The chromatic scale degrees (represented here with D as the tonic) are shown numerically beneath the stave.*

In order to add a degree of randomness, the object [sigmund~] is used to detect the pitch of the outputted notes and create a countermelody. If single notes are inputted to sigmund~, the note-matching is accurate and an exact copy would sound as a countermelody, whose sonority is varied through octave transpositions. However, if more than one note feeds into the 'sigmund' object, it confuses the pitch-detection system and throws out some strange pitches which can be outside of the tonal space of the piece, providing a quirky contrast to the rest of the piece, which is more tonally predictable.

On repeated listening, the structure of *Montague II* becomes evident across the different iterations, and coherence of its texture emerges. A subpatch dictates when the texture should build up, and when it should be sparse. The same mechanism makes decisions on the degree of theme development, and when the countermelody should sound. As part of the structuring of this system, at the end of sixty-four bars,¹⁰³ 'pdChange' is triggered. This will bang (trigger) the theme system, and if the chromaticism level hasn't changed, a new piece will play with the same theme, but with a different rhythm, thereby developing a new piece generatively.

Montague III

Montague III is a great example of how the tools I work with influence my composition. Whilst experimenting with timestretch and pitch variance on sample playback techniques, I realised I could work with the pitch of the piano sample set. So long as the playback speeds were not too far from the original, the notes could sound almost indistinguishable from true sampled notes whilst retaining their timbral origin. This inspired an investigation into microtonal piano, and the works of Ben Johnston.¹⁰⁴

¹⁰³ The term bar is used to describe the time it takes for the timebase to run from 0 to 64, although musically this sounds more like three or four musical bars. This is to allow for the possibility of quick subdivisions in the beat.

¹⁰⁴ Gibbens, 'Design in Ben Johnston's Sonata for Microtonal Piano', 161–94.

I built a system for playing microtonal scales as an equivalent to my established system of playing equal-tempered piano notes (see Appendix 4) to explore the richness of microtonal interplay. To facilitate this, I simplified the melodic, harmonic and rhythmic features whilst maintaining the relaxed, improvisatory feel that pervades the *Montague* collection, but still expanding its pitch repertoire.

I started out with simply rising and falling scales at different speeds and rhythms, which already started to capture the *Montague* aesthetic, whereby the focus was on the sound of the microtonal intervals sounding against each other, but this was not sufficiently thematically engaging.

I then decided that the melodic content could be developed with repeating dyads that were complemented by expanding ranges of notes in the other voice parts. This was more interesting, but ultimately still too repetitive. In particular, the clashing sonorities of the microtones didn't fall together as often as I wanted. Investigating instead the harmonic components that most appealed to me, I focused on their pitch relationships so that I could then work towards placing their occurrence in closer temporal proximity. With these established, I was able to combine the previous experiments in pattern evolution by alternating two-note motifs interplayed to form melodies. These are complemented by a countermelody that is generated every eight bars from a collection of notes that have been recently sounded in the other layers, and then repeated four times. These layers are controlled by a matrix that changes the density of the texture by turning the audio output of these layers on and off. The dataflow for this piece is described in more detail in Appendix 5.

In *Montague III*, more than the other pieces that use the same piano sample library, I have allowed the full sample to sound out in most cases, so that the sounds around the actual note can be heard, the hammers clicking back into place as the note is released, and

also the background hiss in the recordings contributes to the texture, most noticeable for its absence when the texture becomes very thin, and the background level diminishes.

Cantoo

Cantoo is designed to form a sonic cocoon in which slowly evolving timbral and harmonic variations of a core drone are procedurally offered to the listener. As well as volume and pitch variations of its different elements, spatialisation of the different voices using procedural panning adds an extra dimension to the constantly moving sound. The compositional remit for *Cantoo* was to create an endlessly evolving soundworld that appears to constantly move away from a harmonic centre whilst at the same time offering the listener a seemingly static, cohesive listening experience. Its aim is to arrive at an auditory cognitive dissonance where movement is found in stillness.

Cantoo was built in PD, using samples recorded in my home. The main materials came from recordings of a saxophone, a recorder, and my voice each sounding a sustained single note. A synthesised pad sound was also built in Logic Pro,¹⁰⁵ and saved as a WAV file for use within PD.

The preliminary patches for *Cantoo* were started in 2016 and were early experiments in building PD-based sample players. I had been struggling to produce rich pad sounds in some of my previous work using dataflow alone; looped sample playback looked like a good way to achieve better sonic quality than pure synthesis, allowing me to focus on compositional decision-making as opposed to the intricacies of synthesis in dataflow tools. In order to make the sample player perform as smoothly as possible, I used the PD tutorial patch as a starting point and modified the dataflow of the core sample player so the looping

¹⁰⁵ A pad in this context is a sustained drone sound such as often used in electronic dance music to fill a harmonic or atmospheric space. In this case it is based on a single pitch, but harmonic overtones can be heard as a thickening of the texture.

would be almost imperceptible. I then worked on layering several sample players alongside each other in a programmatically efficient way. I wanted to control the pitch and volume variations procedurally, so I created message handlers, small dataflow components that can pass instructions between sample players and logical components, to trigger compositional events.

The samples themselves were very much dependent on instruments available to me at the time, as I was keen to reach a point where I could focus on musical/compositional decisions. I recorded a D on a recorder, a saxophone, and my own voice, afterwards recording further tones and octaves in the saxophone and voice. Along with a pad sound created in Logic Pro by blending various synthesised and pre-recorded material, the final set of materials contained seven core ‘instruments’, or sets of discrete sonic material. These were given names for use in the dataflow logic, and for message-handlers to be able to identify.

Instrument	Dataflow name
Pad	pad
Recorder	rec
Saxophone (hard tone)	saxhard
Saxophone (soft tone)	saxsoft
Saxophone (octave higher)	saxhi
Voice (low octave)	voxhi
Voice (higher octave)	voxlo

Once this working framework was in place, the notes could be played simultaneously at any chosen pitch within the PD samplers. On first listening to the sampler interplay, I found it had interesting, enveloping (or cocoon-like) qualities reminiscent of *9 Beet Stretch* (2002), a

piece by Leif Inge¹⁰⁶ where a recording of Beethoven's Ninth Symphony was time-stretched to last twenty-four hours, that I first came across in the early 2000s via a website that streamed the audio in its entirety. I found *9 Beet Stretch* had mesmerising qualities whereby a listener can easily listen for many hours, as the texture and harmony vary very slowly but create (by way of stretching fast-moving harmonic/rhythmic interplay to slowly emerging sonic texture) movement that propels it inevitably forward. This solidified my instinctive hopes for *Cantoo* into a potential working aesthetic.

Another point of inspiration came from *Cantus in Memory of Benjamin Britten*, by Arvo Pärt (1977). I was inspired by the way in which Pärt evolves a simple process of following the descending A minor harmonic scale to create wonderfully rich harmonic movement. It seemed natural to apply a similar technique to my pitches, choosing to use the dorian scale instead.

Process

Cantoo has two sections: a beginning and an endless second section. At the start of the piece, seven synthesised instruments start playing a D: pad, rec, saxhard, saxsoft, saxhi, voxlo, and voxhi. A representational score for the start (section one) of *Cantoo* can be found in Figure 27 at the end of this chapter.

Section One

After fifteen seconds, one instrument slides down one note of the dorian scale. After another fifteen seconds it slides another note, while a different instrument starts its descent.

Every fifteen seconds, the instruments descend a single note, taking ten seconds to do so, with a new instrument starting its descent each time, until all of them have

¹⁰⁶ Leif Inge and others, *9 Beet Stretch* (2002) <<http://www.xn--lyf-yla.com/about9.htm>>.

descended by at least one note (except the pad which maintains the tonic) covering all the notes in the dorian scale.

Section Two

Once this state has been reached, the instruments continue to slide randomly between notes that are near to them. Each instrument's turn to move is decided aleatorically, but the note to which it will move is decided based on a weighted random system, which also considers the current note.

I wanted to create some variety in the chordal harmony but not so much as to be noticed by the listener as a jarring sonic event, disturbing the cohesive, hypnotic drive of the piece, so I wanted the pitch slides to cause as little disruption as possible, by lowering the volume of the instrument before it slides, and also only sliding by one or two tones at most.

Figure 26 shows the seven possible pitches represented by the numbers 50, 56, 59, 66, 75, 83, and 88.¹⁰⁷ For the middle-lying pitches, the weighted randomisation is most

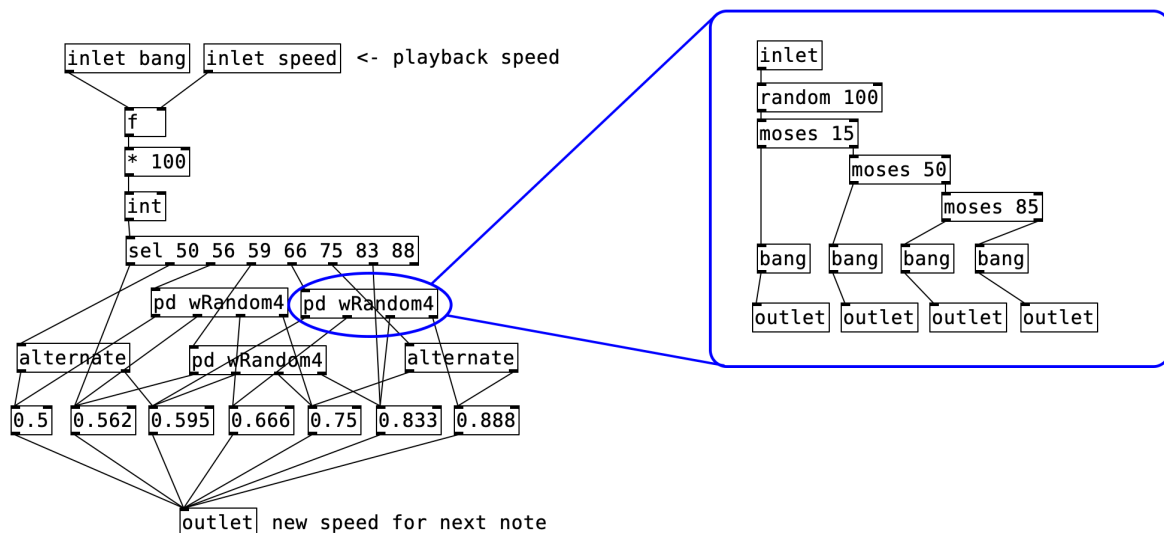


Figure 26 Annotated screenshot to show the pitch chooser from Cantoo using the 'Moses' object in PD which splits a stream of numbers at the number given as its argument.

¹⁰⁷ (These are the playback speed figures multiplied to give integers)

likely to choose the note immediately adjacent to the current one, (35% chance) on either side, but it is also possible that the next note-but-one (15% chance) up or down might be chosen, thus larger leaps in pitch are less likely to occur.

Finally, the instrument panning was implemented by varying the volumes of the right and left channels over time, to give the effect that each instrument is moving around inside the listener's head, creating a sense of space within the evolving texture, in which its movement can be felt.

With *Cantoo*, I wanted to hear how the timbre would change depending on the relative volumes of the different instruments in combining as one evolving texture. I built a matrix to vary the volumes of the instruments, which allowed for enough sophistication within it to create volume changes relative to each other. This matrix was gradually refined over time, to enable smooth pitch transitions. For example, the volume of the instruments that are about to change pitch will dip programmatically, so that the sliding sound is low in the mix.

A listener knows that a note cannot be played indefinitely on a saxophone without taking a breath, for example, but in *Cantoo* the note is elongated to the point where it is possible for the listener to forget that the sound is coming from a saxophone, particularly when hidden within a thick texture, but the odd rasp of the reed occasionally acts as a reminder of the original sound. In Andrew Lewis's *Scherzo* at 1' 42" the child's voice becomes inhumanely sustained with dramatic effect.¹⁰⁸ This plays with the three temporal phases of 'onset, continuant and termination' as defined by Denis Smalley in *Spectromorphology*,¹⁰⁹ which itself builds upon the historical influence of musique concrète

¹⁰⁸ Andrew Lewis, *Scherzo* (1992-1993) <https://electrocd.com/en/piste/imed_13125-1.6>.

¹⁰⁹ Denis Smalley, 'Spectromorphology: Explaining Sound-Shapes', *Organised Sound*, 2/2 (1997), 107–26. page 113.

and the work of Pierre Schaeffer in the 1940s with concepts such as manipulating found sounds to become unrecognisable.¹¹⁰ In *Cantoo* the ‘continuant’ is distorted so that the termination never arrives. The onset is occasionally audible as the sample loops, depending on its depth within the texture.

The format and procedural ambit of this piece can be compared to La Monte Young’s work from the 1960s when he formed the Theatre of Eternal Music, a collective that experimented with sounds and drones in particular,¹¹¹ in that it is designed to be listened to for substantial periods of time and is essentially endless. This once again reinforces the idea that minimalist works lend themselves to mobile/procedural music works, as stated in the Research Aims section of the Introduction to this thesis.

However, while the appropriation of a smartphone as a procedural/adaptive music player seems a perfect choice to me as a composer, this does not completely align with Apple’s App Store values for their customers. *Cantoo* initially failed app store review as it was deemed to have ‘minimal functionality’ and goes on to explain: ‘We found that the usefulness of your app is limited by the minimal amount of content or features it includes’.¹¹² However, I treated this as just another limitation to writing music for this medium, and developed more engaging visuals with a view to building a successful release, citing apps like Eno’s *Reflections* and Loscil’s *Adrift* both successfully released in the App Store with minimal visual effects and touchscreen interaction.¹¹³

¹¹⁰ Pierre Schaeffer, *In Search of a Concrete Music* (Oakland, CA, 2012).

¹¹¹ LaMonte Young’s own words on his work at this time can be found in La Monte Young and Marian Zazeela, ‘Notes on The Theatre of Eternal Music and The Tortoise, His Dreams and Journeys’, *MELA Foundation*, 2000 <<https://www.melafoundation.org/lmy.htm>>.

¹¹² From the App Store Connect Resolution Centre, [accessed 2/08/2021]

¹¹³ Eno, Chilvers and Opal Ltd., *Reflection*; Loscil, *Adrift*, 2015 <<https://apps.apple.com/gb/app/loscil-adrift/id1045265166>>.

Cantoo

(A Representational Score of the first few minutes)

Elise Plans

Static semibreve = 15 seconds, glissandi slowly over 10 seconds to next note.

Recorder

Saxophone (Hard)

Saxophone (Soft)

Soprano (High)

Vox High

Vox Low

Pad

5

random choice *

Rec

Sax hard

Sax soft

Sax hi

Vox hi

Vox lo

Pad

* individual instruments move to a nearby note in the dorian scale, determined by weighted randomisers.

Figure 27 A representational score of the beginning of Cantoo

Chapter 6 - Conclusion

Over the course of my doctoral research, I have produced a body of work in the field of generative mobile compositions that engage some of the structural challenges that arise in music designed to be infinitely long. In *Montague I* and *Cantoo*, the chords never resolve and produce a continuous soundscape. In *Nautilus*, new material is constantly generated using input from the microphone to keep refreshing the palette of sounds washing around within the soundworld. In *Montague II*'s structural framework, each loop is slightly different, and can also be changed by the listener choosing different themes. In *Elerem* the piece is constantly moving through different combinations of the available elements, and the energy level of the piece is related to the physical activity level of the listener. In *Ominator*, the user's own voice determines the harmonic centres at any given moment.

I have maintained the theme of variation through iteration, in that all of the pieces in my portfolio, with the exception of some of the earliest pieces for the *Breathing Stone*, will play out differently on each iteration. This aligns with the aesthetic I am trying to achieve as outlined in the introduction, aptly described by Brian Eno on his work *Reflection*:

My original intention with Ambient music was to make endless music, music that would be there as long as you wanted it to be. I wanted also that this music would unfold differently all the time - 'like sitting by a river': it's always the same river, but it's always changing. But recordings - whether vinyl, cassette or CD - are limited in length, and replay identically each time you listen to them. So in the past I was limited to making the systems which make the music, but then recording 30 minutes or an hour and releasing that. REFLECTION in its album form - on vinyl or CD - is like this. But the app by which REFLECTION is produced is not restricted: it creates an endless and endlessly changing version of the piece of music.¹¹⁴

¹¹⁴ Eno, 'Brian Eno Discusses Reflection'.

As a result of my work within digital therapeutics, I have contributed adaptive music for mobile applications focused on mental health in five experimental studies so far where my music is a core part of biofeedback experiences within the therapeutic interventions.¹¹⁵ In these studies, my contribution focused on creating adaptive music engines that reacted to cardiovascular and respiratory data in order to provide biofeedback loops for interoceptive training, thereby bringing the use of process-based, adaptive music into the clinical domain, in which its functional role becomes part of signals and sensors that make up the design of an intervention. Both in the case of the Breathing Stone, and in the case of these five mobile-based clinical studies, it became clear that music can play a significant role in the design of body-signal therapeutics for mental health. In future research, I will examine the granular aspects of this kind of application for adaptive music in collaboration with cognitive neuroscientists, to further investigate aspects of frequency and form that could be useful in different therapeutic settings. Future collaborations with researchers in the fields of music psychology and neuroscience would improve the rigour for compositional decisions in these works for well-being to ensure intended psychological and physiological effects in the listener (such as inducing a relaxed state, or increasing HRV).

¹¹⁵ N. R. Hemmings and J. M. Kawadler, 'Development and Feasibility of a Digital Acceptance and Commitment Therapy-Based Intervention for Generalised Anxiety Disorder: Pilot Acceptability Study', *JMIR Formative*, 2021 <<https://formative.jmir.org/2021/2/e21737/>>; Sonia Ponzo and others, 'Efficacy of the Digital Therapeutic Mobile App BioBase to Reduce Stress and Improve Mental Well-Being Among University Students: Randomised Controlled Trial', *JMIR mHealth and uHealth*, 8/4 (2020) <<http://dx.doi.org/10.2196/17767>>; Jamie M. Kawadler and others, 'Effectiveness of a Smartphone App (BioBase) for Reducing Anxiety and Increasing Mental Well-Being: Pilot Feasibility and Acceptability Study', *JMIR Formative Research*, 4/11 (2020) <<http://dx.doi.org/10.2196/18067>>; Olga Chelidoni and others, 'Exploring the Effects of a Brief Biofeedback Breathing Session Delivered Through the BioBase App in Facilitating Employee Stress Recovery: Randomised Experimental Study', *JMIR mHealth and uHealth*, 8/10 (2020), e19412; David Plans and others, 'Use of a Biofeedback Breathing App to Augment Poststress Physiological Recovery: Randomised Pilot Study', *JMIR Formative Research*, 3/1 (2019) <<http://dx.doi.org/10.2196/12227>>.

Technical limitations influencing the work

Through the course of producing this portfolio, I came across many limitations particular to the medium of mobile devices, which are described below.

1. Processing power of smartphones.

While I have previously discussed the benefits of ubiquitous portable computers to mobile generative music, the earlier models of smartphones still have limited CPU capacity. This affected my work directly in *Elerem* where the patch I had intended simply required too much DSP (digital signal processing) to be able to run on an iPhone 6, for example, my chosen test devices. I had to modify the work to be less intensive computationally, which affected the overall musical content (as described in Chapter 4).

2. Size of Apps

As outlined in the Methodology section, app size can be an issue, as smartphones can have storage as small as 16GB, and sound files require a lot of data. However, storage in mobile devices is increasing rapidly, which when combined with mobile internet developments for streaming possibilities (such as 5G technologies) and lossless audio compression technologies (such as Free Lossless Audio Codec, or FLAC) could mean that audio file size within apps may no longer be an issue in the near future.

3. Sensor data from smartphones are limited by their APIs

Apple's APIs are designed to allow developers to access data from the smartphone's sensors and tools, or to use them as a data source, but using them in a different way requires the coding skills of a dedicated computer scientist. For example, most smartphones have a camera and a light that can be used as a flash, but there is no API to use the camera and flash to measure pulse. When this was coded by developers at BioBeats, it was an

innovative hack that built functionality on top of that provided by Apple in order to take someone's heart rate through the mobile device.

4. Limited programming skills of a composer.

Apple provide detailed explanations of how to use the various sensors on their devices to access the data they collect.¹¹⁶ However, this information is directed at developers and programmers with a background in computer science, and difficult to implement for someone without this skillset. This resulted in many frustrating hours of adapting code to acquire the data I needed for my pieces, particularly for *Elerem* as described in Chapter 4. There are many things that could be improved across the apps in this portfolio, for example a fade out on exiting the apps would significantly improve the user experience, but this is beyond my current level of coding proficiency. This opens up a channel for future collaboration with developers and computer scientists, allowing for even greater creative scope.

5. Limitations of Pure Data when embedded as libpd

Peter Brinkman developed libpd as a wrapper that makes PD embeddable into other programming frameworks.¹¹⁷ PD is constantly being developed, with more objects being added into the vanilla version, but there are also many libraries of “externals” which are objects that expand PD's capabilities, or expedite tasks by compacting all the necessary computation needed into a single object. Whilst libpd must evolve alongside new objects and libraries, it cannot possibly encompass the multitude of external objects that the international PD community contributes. This means all my patches were made with PD

¹¹⁶ <https://developer.apple.com/documentation/>

¹¹⁷ Brinkmann and others, 'Embedding Pure Data with Libpd', ccxci.*Proceedings of the Pure Data Convention* (2011).

vanilla, and all the algorithms were written from scratch, rather than being able to use an external providing ready-made rich synthesis, for example.

6. Limitations of procedural composition when process-oriented

My compositions rely on decision-making that is coded. Sometimes this will be randomised, but sometimes there are complex processes which take several conditions into account to arrive at a decision. A possibly logical next step would be to use artificial intelligence (AI) or machine learning (ML) to process the decision-making, but this would remove some of the 'control' of the outcome, which is desirable in some situations, but has not been a feature of the work in this portfolio, where the outcomes are strictly managed.

7. Apple's own aesthetics limit possibilities

In order to publish work in the App Store, all apps undergo a review process, to check whether various technical stipulations are met, but also aesthetic conditions which are Apple's own values. Where I feel that a mobile device is the perfect medium to disseminate procedural music compositions, Apple might consider that the iPhone as a medium demands visual and gestural interaction. This meant I couldn't just publish pieces of music as apps with a blank screen, and thought and time had to be given to the user interactions, and the visuals of the apps, an area outside of my field of expertise.

8. Coding frameworks change

Just as PD and libpd are constantly evolving, other coding frameworks are too, so that apps built in 2016 will no longer work in 2020 as different operating system coding parameters have changed and systems become deprecated. I couldn't have built the earlier apps without following online tutorials, especially from Raphael Hernandez in his YouTube

series *Learning libpd for iOS*,¹¹⁸ However, these tutorials were last updated in 2015, and many aspects of the code have been superseded. This makes keeping the apps up-to-date very time-consuming, which also points at the transient and ephemeral nature of art that uses software frameworks: as those frameworks evolve, pieces written in previous versions of the same frameworks are either updated, or become obsolete and disappear as they can no longer be reproduced. This has prompted me to investigate more stable development platforms, such as WWise or FMOD, which can also be used for multi-platform releases.

Final Summary

While I enjoyed composing in Pure Data, I underestimated the technical requirements needed to build my compositions into smartphone/embedded applications. For future work in this area I would choose to collaborate with developers with specific app-building knowledge sets, rather than do all of the coding myself. However, the skills I developed through this research have provided me with a solid grounding in how to compose music within interactive frameworks and have primed me to work with technical frameworks for composition in games, installations and interactive media.

I have not explored machine learning (ML) or artificial intelligence (AI) in my compositions so far, as I have wanted to retain control over the output of my systems; I use individual processes as compositional elements rather than aiming for complete autonomy in the systems. However, it is clear that ML and AI techniques can add compelling dimensions to generative music and will be a useful expansion to my compositional palette going forward. Additionally, there are many other tools other than PD that can be used to make generative music such as Audiokit, Csound, and Juce which I have not pursued for this

¹¹⁸ <https://www.youtube.com/user/cheetomoskeeto>

thesis, favouring a language I know well over experimenting with other possibilities.¹¹⁹

WWise and FMOD are designed for creating game audio and easily lend themselves to making generative music, as discussed in Guy Whitmore's work in Chapter 2. Using these different tools may also allow for cross-platform app building, to include Android apps. The Android operating system was neglected for the purposes of this thesis as it introduced much larger coding complexities, as there are many more variables within the Android community of devices, and even less standardisation across different makes and models of smartphone than within the Apple range. Building for Android was forfeited in favour of concentrating on the music composition aspect of the portfolio.

I have not commented on the visual aspects of my pieces, nor dwelt on the user interfaces/user experience issues (UI and UX) that are inherent in app-building. This allowed me to focus on compositional aspects in this commentary. All of my apps have simplistic UX and UI, with the exception of *Ominator* and *Bespia* for which I consulted the expertise of David Plans, who built more complex visualisations for those specific pieces. *Elerem* has no design for its UX and UI, and will require a design-based collaborator before it can be released in the App Store.

My portfolio makes a number of contributions to the body of generative mobile music, extending its remit into more listener and environment-focused play and interaction, as well as creating new avenues for adaptive mobile music as a functional component of future digital therapeutics. I use the smartphone as a set of performing forces whose ability to synthesise and compute evolving organisations of these forces in real-time allow for listener-influenced variations in performance, in a direct implementation of the Kyoto Generative Music Workshop's 'phone as the new Walkman' paradigm, described in the

¹¹⁹ <https://audiokit.io/>, <https://csound.com/>, and <https://juce.com/>.

introduction. As well, my work has explored the role of the listener not merely as a factor of interactive power (she who owns the device and presses ‘play’), but more significantly as a player whose agency is closer to that of a co-compositional force. In *Ominator*, *Elerem*, *Montague* and in all pieces where cardiovascular data is gathered the listener doesn’t just have ultimate power to start and finish the piece; they can alter its course and change foundational aspects of its content such as their harmonic centre, tempo, and thematic element introduction and withdrawal. Whilst even early pieces of process music such as Stockhausen’s *Plus Minus*¹²⁰ explore the role of the performer within the process (instigating the process to generate an iteration of the piece), and many have explored it since, my work brings the listeners’ own body, her movement, voice, and internal signal as well as its perception, into not only the internal compositional reasoning of each piece, its algorithms, but also into more radical, compositional choice. I have also reaffirmed the therapeutic use of adaptive music as a recurring theme in many of my pieces, including the use of vocalisation to encourage wellbeing (in *Ominator* and *Nautilus*). The subject of singing giving many health benefits is well documented,¹²¹ and its incorporation into well-being apps is a promising area for further exploration.

In this way, I feel this portfolio represents an advance in music(s) that can involve listener interaction as part of their very makeup, in effect using the smartphone and its sensors as instrumental levers to engender a much more profound connection with the listener.

¹²⁰ Stockhausen, ‘Plus-Minus’.

¹²¹ Examples can be found in Clift, S. and Hancox, G. (2001) ‘The perceived benefits of singing: findings from preliminary surveys of a university college choral society’ in *The Journal of the Royal Society for the Promotion of Health*, 121(4), 248-56, and Stacey, R. Britten, K. and Kerr, S. (2002) ‘Singing for health: an exploration of the issues’ in *Health Education*, 102(4), 156-62.

Bibliography

Aphex Twin, *Selected Ambient Works, 85-92* (Apollo (R&S) Records, 1992)

Arnalds, Ólafur, and Nils Frahm, *Trance Frendz* (Erased Tape Records, 2016)

<<https://www.erasedtapes.com/release/eratp081-olafur-arnalds-nils-frahm-trance-frendz>>

Bainter, Alex, 'The Future of Generative Music: Why the World Is Finally Ready for Endless Music Systems', *Medium*, 2019

<<https://medium.com/@alexbainter/the-future-of-generative-music-e19b6722deb2>>

[accessed 14 August 2021]

Beary, J. F., and H. Benson, 'A Simple Psychophysiologic Technique Which Elicits the Hypometabolic Changes of the Relaxation Response', *Psychosomatic Medicine*, 36/2 (1974), 115–20

Björk, *Debut* (Elektra Records, One Little Indian, 1993)

Björk, and Second Wind Ltd, *Biophilia*, 2011

<<https://itunes.apple.com/gb/app/björk-biophilia/id434122935?mt=8>>

Brinkmann, P., *Making Musical Apps*, O'Reilly and Associate Series (Sebastopol, CA, 2012)

Brinkmann, Peter, Peter Kirn, Richard Lawler, Chris McCormick, Martin Roth, and Hans-Christoph Steiner, 'Embedding Pure Data with Libpd', in *Proceedings of the Pure Data Convention* (2011), ccxc

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.2510&rep=rep1&type=pdf>>

Buhmann, Jeska, 'Effects of Music-Based Biofeedback on Walking and Running', 2017

<<https://biblio.ugent.be/publication/8540383>>

Chadabe, Joel, 'Remarks on Computer Music Culture', *Computer Music Journal*, 24/4 (2000), 9–11

———, 'The Role of Apps in Electroacoustic Music: A New Dimension for Music through Tablets and Other Devices', *Organised Sound*, 20/01 (2015), 99–104

Chelidoni, Olga, David Plans, Sonia Ponzo, Davide Morelli, and Mark Cropley, 'Exploring the Effects of a Brief Biofeedback Breathing Session Delivered Through the BioBase App in Facilitating Employee Stress Recovery: Randomized Experimental Study', *JMIR mHealth and uHealth*, 8/10 (2020), e19412

- Cheshire, Tom, 'BioBeats App Creates Generates Custom Music from Your Heart Rate', *Wired*, 2013 <<https://www.wired.co.uk/article/your-heart-is-a-dj>> [accessed 14 August 2021]
- Clarke, Eric, *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning* (Oxford, UK, 2005)
- Clift, S. and Hancox, G. (2001) 'The perceived benefits of singing: findings from preliminary surveys of a university college choral society' in *The Journal of the Royal Society for the Promotion of Health*, 121(4), 248-56
- Collier, David, 'In Your Own Time' (2013)
<http://davidcollier.ie/wp-content/uploads/2012/08/4_InYourOwnTime.pdf>
- , 'What Is Mobile Music? – Mobile Music #5 //', *David Collier*, 2012
<<http://davidcollier.ie/what-is-mobile-music/>> [accessed 6 November 2019]
- Collins, Karen, 'An Introduction to Procedural Music in Video Games', *Contemporary Music Review*, 28/1 (2009), 5–15
- Collins, Nick, *Introduction to Computer Music* (Chichester, UK, 2010)
- Collins, Nick, and Andrew R. Brown, 'Generative Music Editorial', *Contemporary Music Review*, 28/1 (2009), 1–4
- Craig, A. D., 'How Do You Feel? Interoception: The Sense of the Physiological Condition of the Body', *Nature Reviews. Neuroscience*, 3/8 (2002), 655–66
- , 'Interoception: The Sense of the Physiological Condition of the Body', *Current Opinion in Neurobiology*, 13/4 (2003), 500–505
- Daft Punk, and Anders Svendsen, *iDaft Jamming, Daft Punk Edition*, 2009
<<https://apps.apple.com/us/app/idaft-jamming/id308712434>>
- Dibben, Nicola, 'Visualising the App Album with Björk's Biophilia', in *The Oxford Handbook of Sound and Image in Digital Media*, Oxford Handbooks (Oxford, UK, 2014), 682–706
- DIY, 'Soap Box: Welcome To The Bronze Age', *DIY* (2011)
<<http://diymag.com/archive/soap-box-welcome-to-the-bronze-age>> [accessed 8 October 2019]
- Dodge, Charles, and Thomas A. Jerse, *Computer Music: Synthesis, Composition, and Performance*, 1st edn (New York, 1985)

- Dredge, Stuart, 'Domino Launches Piano Ombre App for François & The Atlas Mountains Album', *Music:)Ally*, 2014
 <<https://musically.com/2014/03/24/domino-launches-piano-ombre-app-for-francois-the-atlas-mountains/>> [accessed 28 July 2021]
- , 'The History of the Album-App: Creativity Galore', *Music:)Ally*, 2016
 <<https://musically.com/2016/07/25/history-album-app-creativity-commercially/>> [accessed 6 April 2021]
- Elliott, Dave, Remco Polman, and Richard McGregor, 'Relaxing Music for Anxiety Control', *Journal of Music Therapy*, 48/3 (2011), 264–88
- Elliott, William J., and Joseph L. Izzo Jr, 'Device-Guided Breathing to Lower Blood Pressure: Case Report and Clinical Overview', *MedGenMed: Medscape General Medicine*, 8/3 (2006), 23
- Endel© Sound GmbH, 'Endel', *Endel* <<http://endel.io/>> [accessed 11 June 2019]
- Eno, Brian, 'Brian Eno Discusses Reflection', *Brian Eno* <<https://brian-eno.net/reflection/>> [accessed 8 April 2021]
- Eno, Brian, Peter Chilvers, and Opal Ltd, *Bloom*, 2008
 <<https://itunes.apple.com/gb/app/bloom/id292792586?mt=8>>
- Eno, Brian, Peter Chilvers, and Opal Ltd., *Reflection*, 2017 <<http://www.brian-eno.net/>>
- Farnell, Andy, *Designing Sound* (Cambridge, MA, 2010)
- Fever Ray, *Fever Ray* (Rabid, 2009)
- Gaye, Lalya, Lars Erik Holmquist, Frauke Behrendt, and Atau Tanaka, '2006: Mobile Music Technology: Report on an Emerging Community', in *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*, ed. by Alexander Refsum Jensenius and Michael J. Lyons (Cham, 2017), 253–65
- , 'Mobile Music Technology: Report on an Emerging Community', in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, NIME '06 (Paris, France, France, 2006), 22–25
- Gibbens, John Jeffrey, 'Design in Ben Johnston's Sonata for Microtonal Piano', *Interfaces*, 18/3 (1989), 161–94
- Grierson, M., 'Gold + Goldsmiths = BRONZE: Mick Grierson Explains the Bronze Project... – Blog.DoC' <<https://www.doc.gold.ac.uk/blog/?p=208>> [accessed 8 October 2019]

- Guerraz, Agnes, and Jacques Lemordant, 'Indeterminate Adaptive Digital Audio for Games on Mobiles', in *From Pac-Man to Pop Music*, ed. by Karen Collins (Farnham, UK, 2008)
- von Gunden, Heidi, 'The Theory of Sonic Awareness in the Greeting by Pauline Oliveros', *Perspectives of New Music*, 19/1/2 (1980), 409–16
- Hedges, Stephen A., 'Dice Music in the Eighteenth Century', *Music & Letters*, 59/2 (1978), 180–87
- Hemmings, N. R., and J. M. Kawadler, 'Development and Feasibility of a Digital Acceptance and Commitment Therapy–Based Intervention for Generalised Anxiety Disorder: Pilot Acceptability Study', *JMIR Formative*, 2021
<<https://formative.jmir.org/2021/2/e21737/>>
- Hennigan, Keith, 'Music in Liquid Forms: A Framework for the Creation of Reactive Music Recordings', ed. by Adams Martin And Simon (unpublished PhD, Trinity College, Dublin, 2018)
<http://www.tara.tcd.ie/bitstream/handle/2262/83259/Music%20in%20Liquid%20Forms_KeithHenniganPhDThesis.pdf?sequence=1>
- Hillerson, T., *Programming Sound with Pure Data* (Dallas, 2014)
- Hodges, Donald A., and David C. Sebald, *Music in the Human Experience: An Introduction to Music Psychology* (New York, 2011)
- Holy Other, *With U* (Tri Angle, 2011)
- Holzer, Derek, Adam Hyde, Joao Pais, Alexandre Porres, Luka Princic, Hans-Christoph Steiner, and others, 'Floss Manuals', *Floss Manuals - Pure Data*
<<http://write.flossmanuals.net/pure-data/introduction2/>> [accessed 7 June 2020]
- Hosokawa, Shuhei, 'The Walkman Effect', *Popular Music*, 4 (1984), 165–80
- Inge, Leif, Anders Vinjar, Kjetil Matheussen, and Bjarne Kvinnesland, *9 Beet Stretch* (2002)
<<http://www.xn--lyf-yla.com/about9.htm>>
- Ipnos Software Inc., *Relax Melodies Sleep Sounds*, 2009
<<https://apps.apple.com/us/app/relax-melodies-sleep-sounds/id314498713>>
- Jansch, A., 'Futures EP', *AdamJansch* <<https://www.adamjansch.co.uk/code/futures-ep>>
[accessed 16 June 2020]
- , 'Towards the Open Outcome Record: A Portfolio of Works Exploring Strategies of Freeing the Record from Fixity', 2011
<http://eprints.hud.ac.uk/id/eprint/16730/8/One_man_band_x_n_-_cycle_2.m4v>

- Kamalzadeh, Mohsen, Dominikus Baur, and Torsten Möller, 'Listen or Interact? A Large-Scale Survey on Music Listening and Management Behaviours', *Journal of New Music Research*, 45/1 (2016), 42–67
- Kaneko, T. and Jo, K. 'Generative Music Workshop,' *2011 Second International Conference on Culture and Computing*, 2011, pp. 179-180, doi: 10.1109/Culture-Computing.2011.54.
- Katz, Mark, *Capturing Sound: How Technology Has Changed Music* (Berkeley, 2010)
- Kawadler, Jamie M., Nicola Rose Hemmings, Sonia Ponzo, Davide Morelli, Geoffrey Bird, and David Plans, 'Effectiveness of a Smartphone App (BioBase) for Reducing Anxiety and Increasing Mental Well-Being: Pilot Feasibility and Acceptability Study', *JMIR Formative Research*, 4/11 (2020) <<http://dx.doi.org/10.2196/18067>>
- Kreidler, Johannes, *Loadbang : Programming Electronic Music in PD* (Hofheim, 2009)
- Lewis, Andrew, *Scherzo* (1992-1993) <https://electrocd.com/en/piste/imed_13125-1.6>
- Loscil, *Adrift*, 2015 <<https://apps.apple.com/gb/app/loscil-adrift/id1045265166>>
- Mathews, Max V., 'An Acoustic Compiler for Music and Psychological Stimuli', *Bell System Technical Journal*, 40/3 (1961), 677–94
- Maxis, *Spore* (2008) <<https://www.spore.com/>>
- Meredith, Anna, *Varmints* (Moshi Moshi, 2016)
- Music for Eighteen Musicians* (1978)
- Oliveros, P., *Deep Listening: A Composer's Sound Practice* (Bloomington, 2005)
- Paterson, Justin, and Rob Toulson, 'The Commercialisation of Interactive Music', *University of West London Research* <<https://www.uwl.ac.uk/research/research-centres-and-groups/prism-music-and-screen/commercialisation-music>> [accessed 8 October 2019]
- Paterson, Justin, Rob Toulson, and Russ Hepworth-Sawyer, 'User-Influenced/Machine-Controlled Playback: The variPlay Music App Format for Interactive Recorded Music', *Arts & Health*, 8/3 (2019), 112
- Phillips, Winifred, *A Composer's Guide to Game Music* (Cambridge, MA, 2014)
- Plans, David, Davide Morelli, Stefan Sütterlin, Lucie Ollis, Georgia Derbyshire, and Mark Cropley, 'Use of a Biofeedback Breathing App to Augment Poststress Physiological

- Recovery: Randomised Pilot Study', *JMIR Formative Research*, 3/1 (2019) <<http://dx.doi.org/10.2196/12227>>
- Plans, Elise, 'Composer in Your Pocket: Procedural Music in Mobile Devices', *Musicology Research Journal*, 2017, 51
- Plans, Elise, Davide Morelli, and David Plans, 'AudioNode: Prototypical Affective Modelling in Experience-Driven Procedural Music Generation', 2015 <https://web.archive.org/web/20170610155005id_/http://www.ccgworkshop.org:80/2015/Plans.pdf>
- Ponzo, Sonia, Davide Morelli, Jamie M. Kawadler, Nicola Rose Hemmings, Geoffrey Bird, and David Plans, 'Efficacy of the Digital Therapeutic Mobile App BioBase to Reduce Stress and Improve Mental Well-Being Among University Students: Randomised Controlled Trial', *JMIR mHealth and uHealth*, 8/4 (2020) <<http://dx.doi.org/10.2196/17767>>
- Puckette, Miller, *The Theory And Techniques Of Electronic Music* (Singapore, 2007)
- Ratto, Matt, 'Critical Making: Conceptual and Material Studies in Technology and Social Life', *The Information Society*, 27/4 (2011), 252–60
- RjDjme, 'RjDj The Mind Twisting Hearing Sensation' (2008) <<https://www.youtube.com/watch?v=0fxQFly1BQA>> [accessed 5 August 2021]
- Roads, Curtis, *The Computer Music Tutorial* (Cambridge, MA, 1996)
- Robehmed, Natalie, 'An App That Makes Music From Your Heartbeat', *Forbes Magazine*, 2013 <<https://www.forbes.com/sites/natalierobehmed/2013/08/27/an-app-that-makes-music-from-your-heartbeat/?sh=3112d2a742eb>> [accessed 14 August 2021]
- Scaletti, Carla, 'Sonification ≠ Music', *The Oxford Handbook of Algorithmic Music*, 2018, 363
- Schaeffer, Pierre, *In Search of a Concrete Music* (Berkeley, CA, 2012)
- Schandry, R., 'Heart Beat Perception and Emotional Experience', *Psychophysiology*, 18/4 (1981), 483–88
- Serra, E., *Diva Dance* (Virgin Records, 1997)
- Slabu, Lavinia, Carles Escera, Sabine Grimm, and Jordi Costa-Faidella, 'Early Change Detection in Humans as Revealed by Auditory Brainstem and Middle-Latency Evoked Potentials', *The European Journal of Neuroscience*, 32/5 (2010), 859–65

- Smalley, Denis, 'Spectromorphology: Explaining Sound-Shapes', *Organised Sound*, 2/2 (1997), 107–26
- Solomos, Makis, 'Cellular Automata in Xenakis's Music. Theory and Practice', *The International Symposium Iannis Xenakis*, 2006
<<http://hal.archives-ouvertes.fr/hal-00770141/>>
- Stacey, R. Britten, K. and Kerr, S. (2002) 'Singing for health: an exploration of the issues' in *Health Education*, 102(4), 156–62.
- Steiglitz, Ken, *A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music*, Addison-Wesley Professional Computing Series (Menlo Park, 1996)
- Stockhausen, 'Plus-Minus' (Universal Edition, London, 1963)
- Surge and Falter Ltd, and Massive Attack, *Fantom*, 2016
<<https://itunes.apple.com/us/app/fantom-sensory-music/id1062360670?mt=8>>
- The Police, *Ghost in the Machine* (A&M, 1981)
- Thomas, Chance, *Composing Music for Games: The Art, Technology and Business of Video Game Scoring* (New York, 2017)
- Toussaint, Godfried T., 'The Rhythm That Conquered the World: What Makes a "good" Rhythm Good', *Percussive Notes*, 2 (2011), 52
- Underworld, *Second Toughest in the Infants* (Junior Recordings Ltd, 1996)
- Vuust, Peter, Line K. Gebauer, and Maria A. G. Witek, 'Neural Underpinnings of Music: The Polyrhythmic Brain', in *Neurobiology of Interval Timing*, ed. by Hugo Merchant and Victor de Lafuente (New York, NY, 2014), 339–56
- Whitmore, Guy, 'Why Music Must Evolve Along With Digital Media', *Twitch.tv*, 2018
<<https://www.twitch.tv/audiokinetic/video/288292691>>
- Williamson, John, and Martin Cloonan, 'Rethinking the Music Industry', *Popular Music*, 26/2 (2007), 305–22
- Wooller, Rene, Andrew R. Brown, Eduardo Miranda, Joachim Diederich, and Rodney Berry, 'A Framework for Comparison of Process in Algorithmic Music Systems', ed. by Burraston David and Edmonds Ernest (presented at the Generative Arts Practice, 2005), 109–24
- Xenakis, Iannis, *Formalized Music: Thought and Mathematics in Composition* (New York, 1992)

Young, La Monte, and Marian Zazeela, 'Notes on The Theatre of Eternal Music and The Tortoise, His Dreams and Journeys', *MELA Foundation*, 2000
<<https://www.melafoundation.org/lmy.htm>>

Appendix 1: Instructions for Accessing Portfolio Apps

(Hard copy included in iPhone Boxing).

iPhone and App Instructions

Operating instructions for accompanying iPhone 6 phones containing my PhD portfolio. (These instructions are designed for users who are unfamiliar with Apple iPhones.)

- To turn the phone on, press and hold the button on the right-hand side of the phone.
- To open/unlock the phone, press the round button at the bottom of the iPhone 6s's screen.
- The passcode to unlock the phone is **161803** (the first six digits of the golden ratio). The phone has been set up to not require passcode re-entry for 1 hour, so you should just be able to press the home button to unlock it within an hour of entering the passcode.
- To exit an app, double-click the round button at the bottom of the screen, which should then display a slightly smaller image of the app or apps that are open. To exit an app, place a finger on it and swipe upwards.
- The volume-adjust buttons are on the left-hand side of the phone.
- To close the phone, press the button on the right-hand side of the phone.
- To completely switch the phone off, hold the side button until a slider appears on the top of the screen, and swipe this slider to the right.
- All the portfolio apps will work without internet. Please do not connect to wifi, as operating system updates may cause the apps to not function.

App Instructions

ALL APPS ARE DESIGNED TO BE HEARD THROUGH HEADPHONES

Ominator

Ominator only works with wired headphones, not wireless ones.

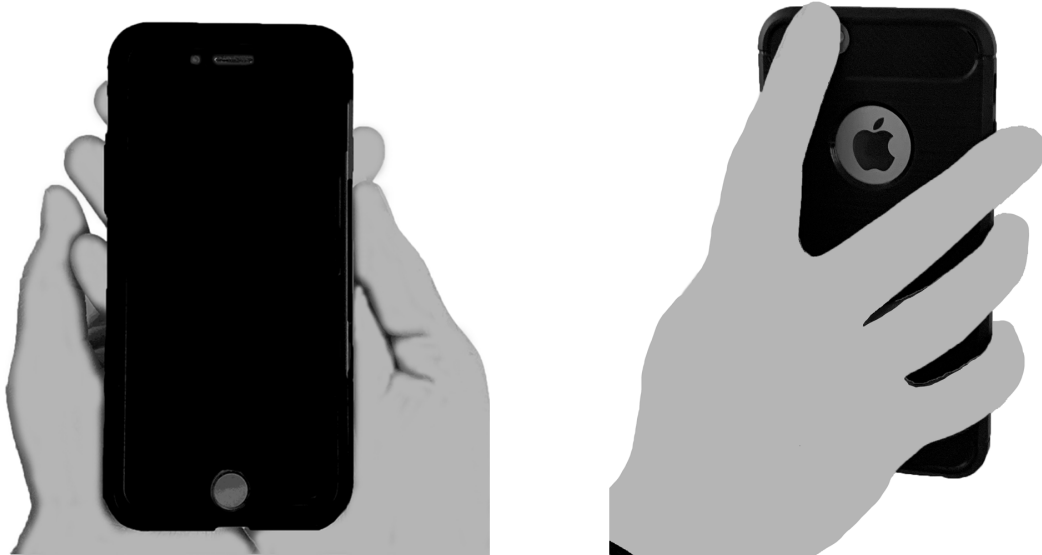
Please press the round 'OM' button to start the music.

Elerem

Elerem works best if the phone is held vertically, or placed vertically in a pocket.

Bespia

In order to detect your heartbeat, please hold the phone in your lap parallel to the floor, face-up, with your finger covering the camera and flashlight at the back.



Appendix 2: Score and Instructions for the *Breathing Stone*.

Piece for two Breathing Stones, one or two audience members and a drone

Set up a drone in C#. It can be any drone but should have evolving sonorities over time, and be played very quietly so that it is only just heard in the space where the stones are.

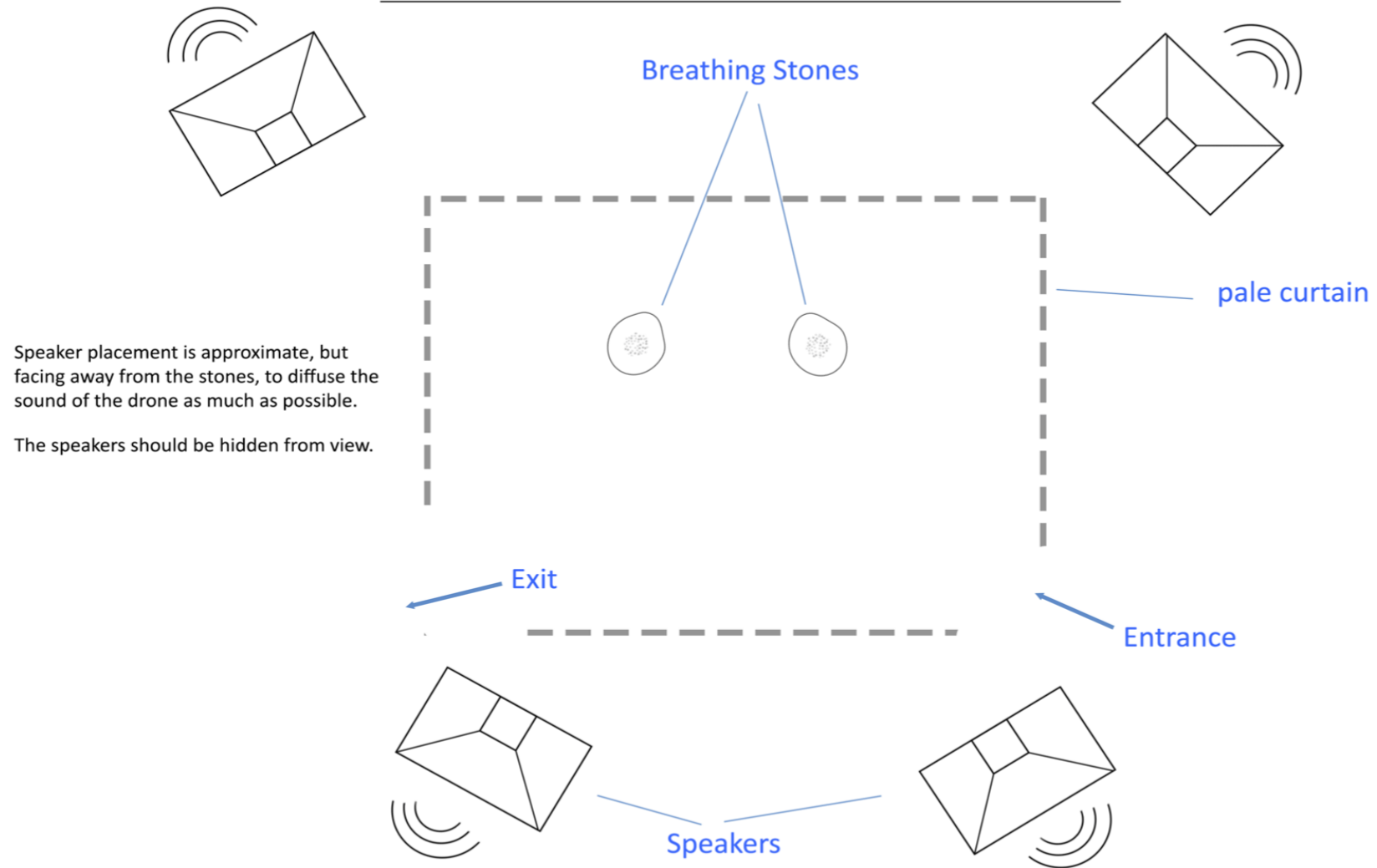
Place the Breathing Stones near each other, so that if two people are using them simultaneously, they can each hear both stones.

The users should start the stone when they are ready, and follow the breathing exercise, which encourages them to take deep breaths into their belly for just over 2 minutes. (Twelve breaths, 5 seconds inhale, 5 seconds exhale, with a couple of practice breaths at the beginning.)

Participants in the piece can choose whether to go again or to let another audience member take a turn using the stone.

The scores of the music inside the stones will be preserved as pure data patches.

Diagram to show position of stones and drone-playing speakers.



The Breathing Stone

A device that helps you manage stress and pain by helping you breathe deeply.

How to use the Breathing Stone

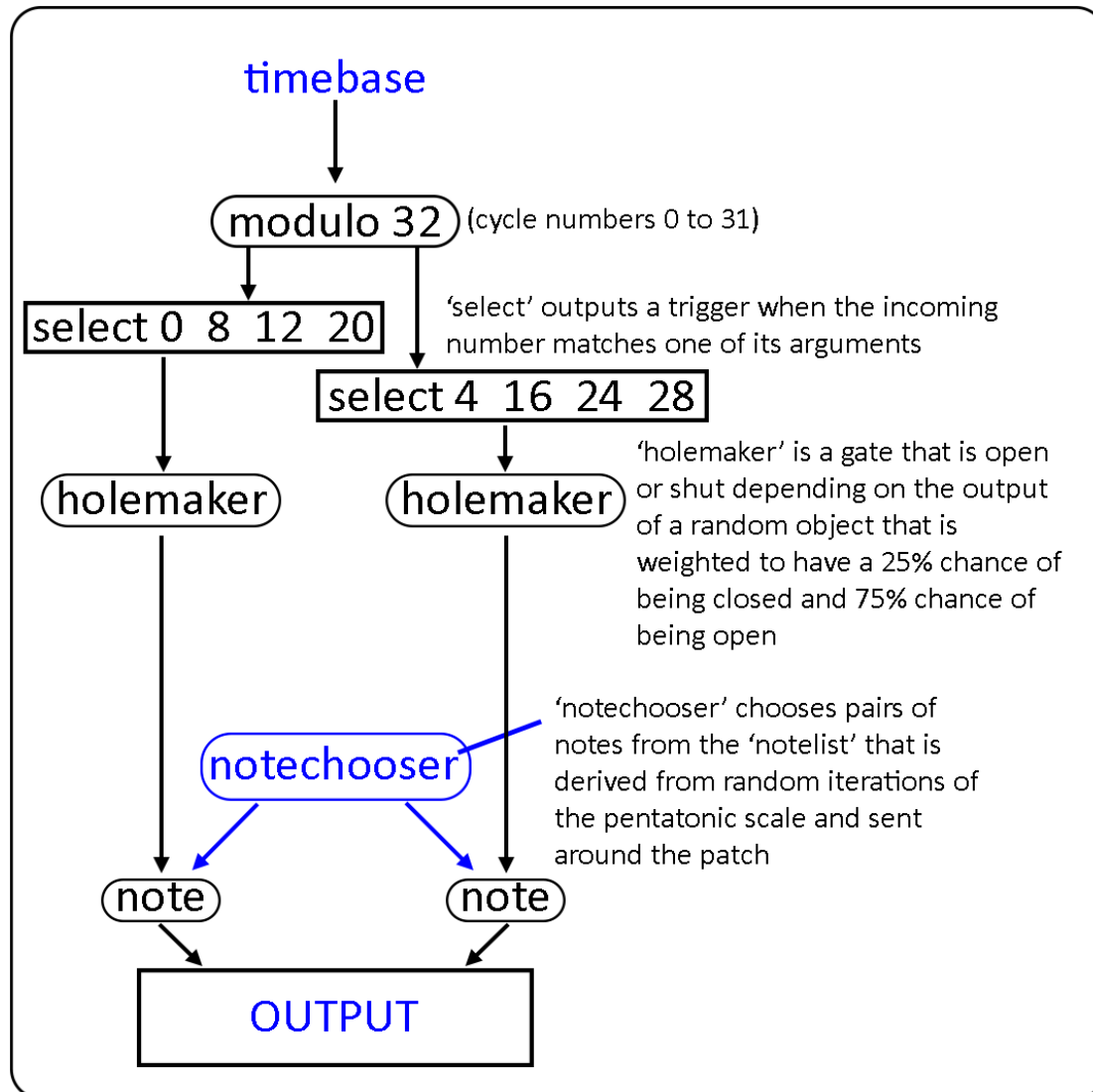
1. Press the power button at top right.
2. Hold the stone in your hands. Make sure the left and right steel sensors are touching the palms of your hands, and the tips of your fingers are on the bigger, top center steel plate.
3. After a few seconds, lights and music will start.
4. When the stone is blue, breathe in deeply through your nose, into your belly.
5. Breathe out when the stone turns white.
6. The music will change with your breathing pattern to help guide you.



The Breathing Stone helps you use a simple breathing exercise that encourages relaxation and helps manage stress and pain. As you hold it, it collects heart data from the sensors on the right, left, and back of the stone. It then makes music and light to guide your breathing. If you breathe from the belly (and not the chest), you will stimulate the vagus nerve, an important element of the parasympathetic nervous system. When stimulated, it counteracts your sympathetic nervous system, which causes stress by activating your fight-or-flight response.

Appendix 3: Diagrams of Algorithms used in *Elerem* Mallet Instrument

Melody Fragment 1:



Melody Fragment 2:

Melodic Fragment Algorithm 2 plays pre-composed phrases, choosing randomly from a set of 4. Although they are shown here below with the key note of G, they will be transposed to whatever key the patch is currently in. The numbers along the top show which numbers of the timecode trigger the note, to give the rhythm (from a timebase of 0 to 63) and the numbers along the bottom show the chromatic distance from the key note. When added to the key note these produce pitches to match the current key of the piece. The output goes to the octave randomiser in the parent patch.

The image displays four musical staves, each representing a different melody fragment. Each staff is in bass clef with a 4/4 time signature. Above each staff are timecode numbers (0-63) indicating when notes are triggered. Below each staff are chromatic distance numbers (integers) relative to the key note G. The fragments are numbered 1 through 4 on the left.

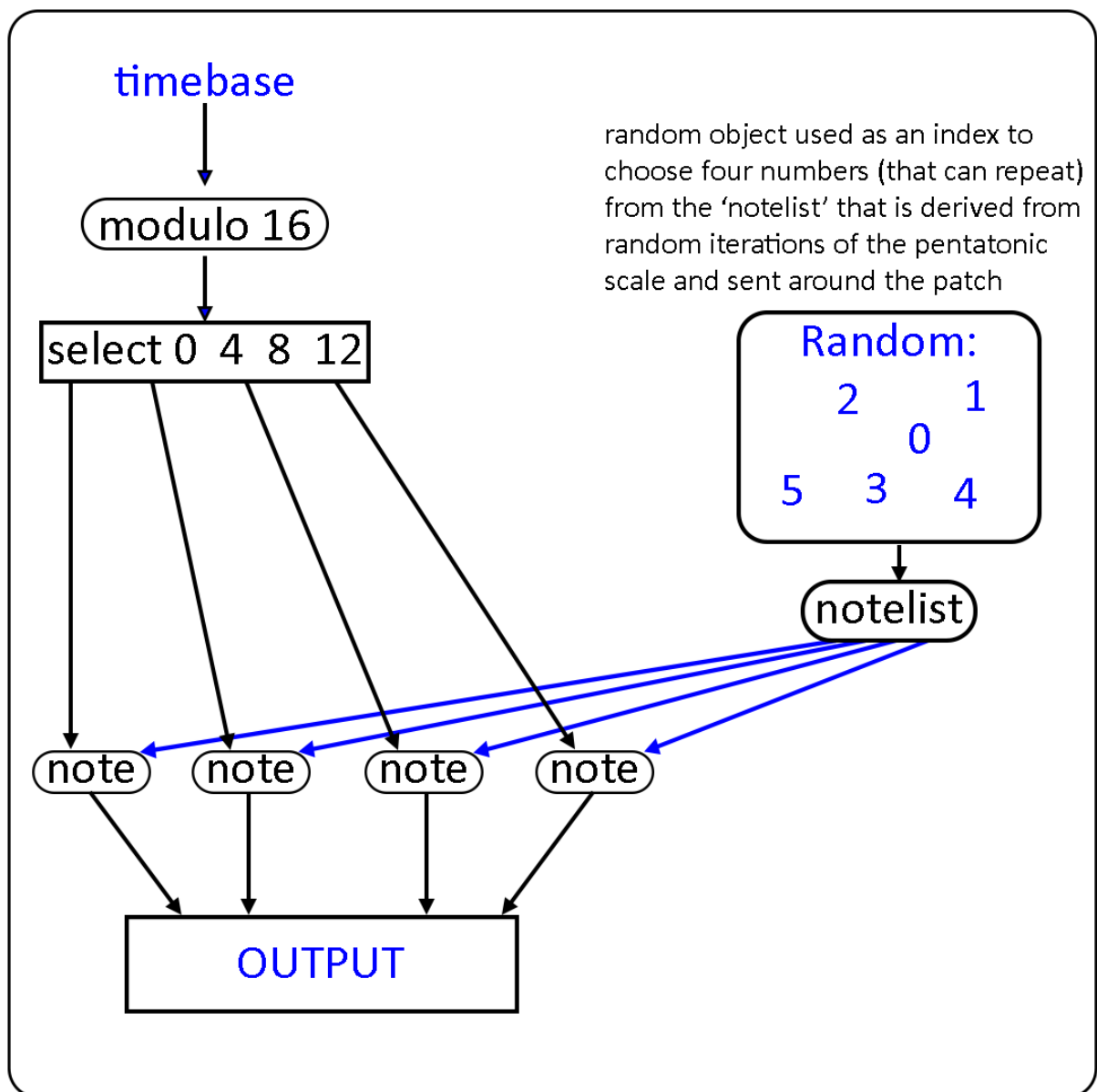
Fragment 1: Timecode: 24, 26, 28, 30, 31. Chromatic distance: 0, 0, 0, -2, 0. The melody consists of five notes: G (0), G (0), G (0), F (0-2), and G (0).

Fragment 2: Timecode: 24, 26, 28, 30, 31. Chromatic distance: 0, 0, 4, -2, 0. The melody consists of five notes: G (0), G (0), B (4), F (0-2), and G (0).

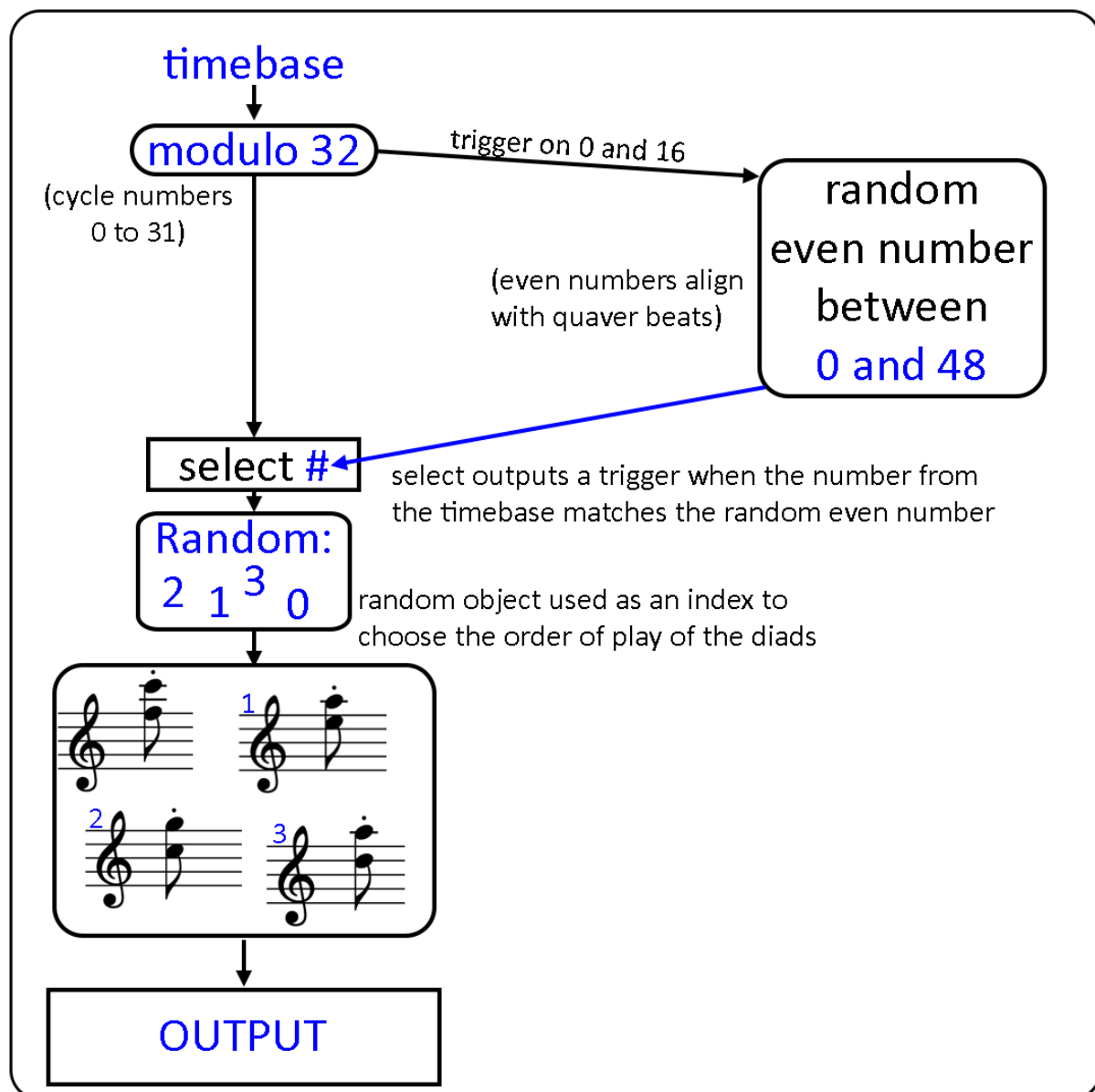
Fragment 3: Timecode: 52, 54, 56, 60, 0, 2, 4, 6. Chromatic distance: 12, 12, 10, 7, 0, 0, 0, 0. The melody consists of eight notes: G (12), G (12), F (10), E (7), G (0), G (0), G (0), and G (0).

Fragment 4: Timecode: 2, 4, 6, 8, 12, 14. Chromatic distance: 0, 0, -2, 2, -2, 0. The melody consists of six notes: G (0), G (0), F (0-2), G (2), F (0-2), and G (0).

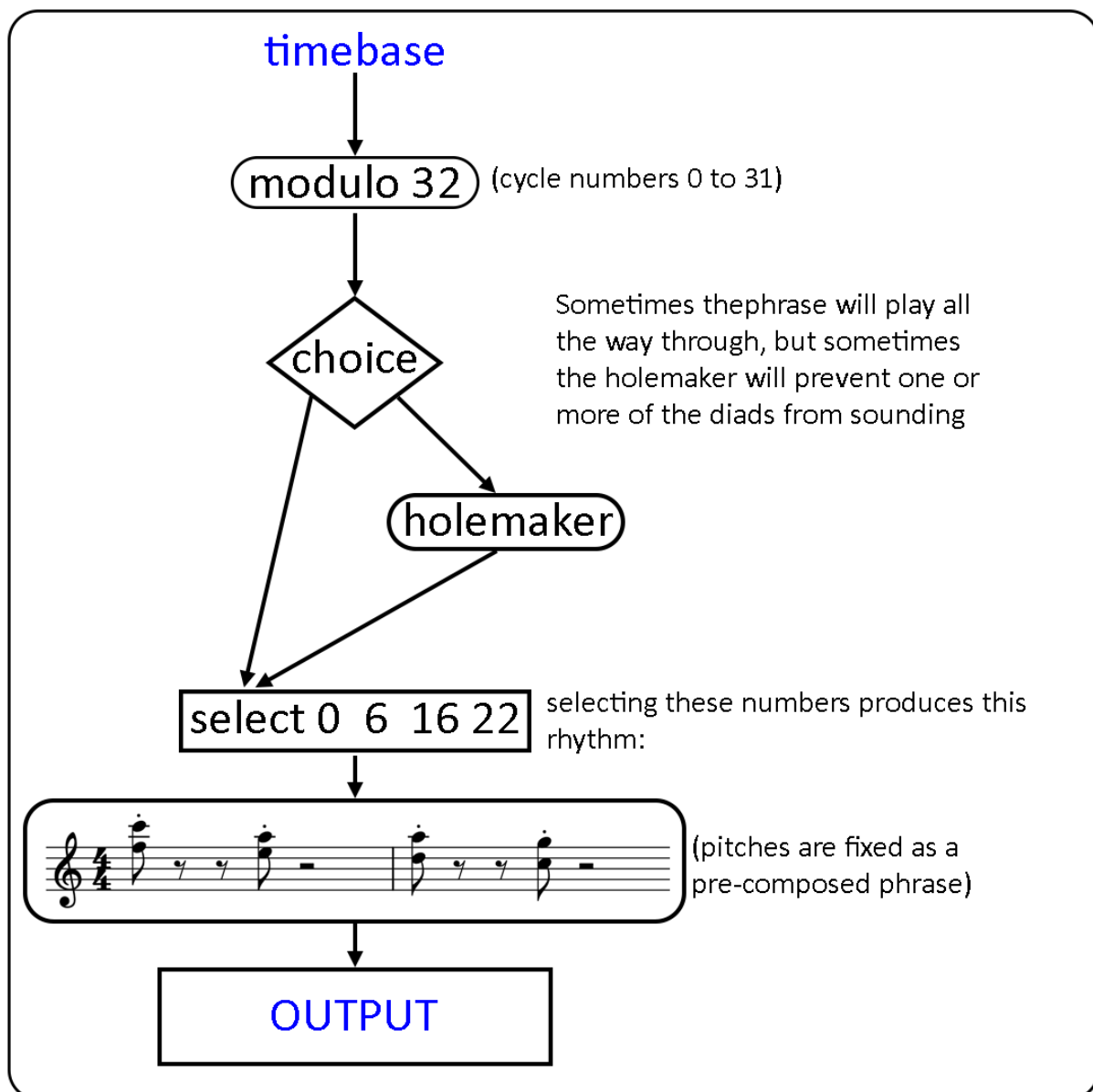
Melody Fragment 3



Dyad Fragment 1:

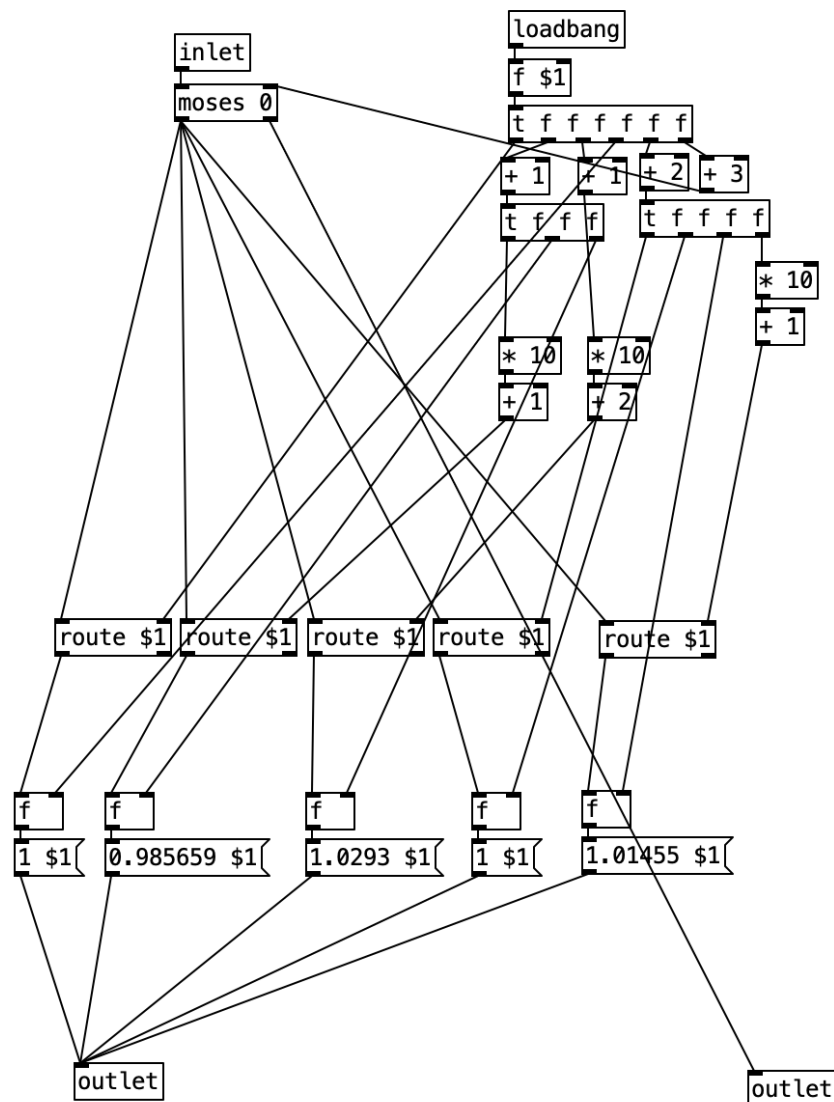


Dyad Fragment 2:



Appendix 4: Example of microtonal calculations and playback in Pure Data

Midi	Note	Frequency (Equal)	Frequency (8ve/16)	Ratio	Midi note and ratio required to produce microtone	New midi note names	New Midi notes names from 21 to 102		
57	A3	220	220	220	57	57	21	521	822
			229.7402321	1.014545335	58 / 1.01454533	581	221	522	831
58	A#3	233.0818808					222	531	84
58	A#3	233.0818808	239.9117012	1.029302237	58 x 1.02930224	582	231	54	851
59	B3	246.9416506	250.5334996	1.014545335	59 x 1.01454533	591	24	551	852
60	C4	261.6255653	261.6255653	261.6255653	60	60	251	552	861
			273.2087187	1.014545335	61 / 1.01454533	611	252	561	87
61	C#4	277.182631					261	57	881
61	C#4	277.182631	285.304702	1.029302237	61 x 1.02930224	612	27	581	882
62	D4	293.6647679	297.9362203	1.014545335	62 x 1.01454533	621	281	582	891
63	D#4	311.1269837	311.1269837	311.1269837	63	63	282	591	90
			324.9017521	1.014545335	64 / 1.01454533	641	291	60	911
64	E4	329.6275569					30	611	912
64	E4	329.6275569	339.2863816	1.029302237	64 x 1.02930224	643	311	612	921
65	F4	349.2282314	354.307873	1.014545335	65 x 1.01454533	651	312	621	93
66	F#4	369.9944227	369.9944227	369.9944227	66	66	321	63	941
			386.3754753	1.014545335	67 / 1.01454533	671	33	641	942
67	G4	391.995436					341	642	951
67	G4	391.995436	403.481779	1.029302237	67 x 1.02930224	672	342	651	96
68	G#4	415.3046976	421.3454435	1.014545335	68 x 1.01454533	681	351	66	971
69	A4	440	440	440	69	69	36	671	972
			459.4804643	1.014545335	70 / 1.01454533	701	371	672	981
70	A#4	466.1637615					372	681	99
70	A#4	466.1637615	479.8234024	1.029302237	70 x 1.02930224	702	381	69	1001
71	B4	493.8833013	501.0669993	1.014545335	71 x 1.01454533	711	39	701	1002
72	C5	523.2511306	523.2511306	523.2511306	72	72	401	702	1011
			546.4174373	1.014545335	73 / 1.01454533	731	402	711	102
73	C#5	554.365262					411	72	
73	C#5	554.365262	570.609404	1.029302237	73 x 1.02930224	732	42	731	
74	D5	587.3295358	595.8724407	1.014545335	74 x 1.01454533	741	431	732	
75	D#5	622.2539674	622.2539674	622.2539674	75	75	432	741	
			649.8035042	1.014545335	76 / 1.01454533	761	441	75	
76	E5	659.2551138					45	761	
76	E5	659.2551138	678.5727632	1.029302237	76 x 1.02930224	762	461	762	
77	F5	698.4564629	708.6157461	1.014545335	77 x 1.01454533	771	462	771	
78	F#5	739.9888454	739.9888454	739.9888454	78	78	471	78	
			772.7509506	1.014545335	79 / 1.01454533	791	48	791	
79	G5	783.990872					491	792	
79	G5	783.990872	806.963558	1.029302237	79 x 1.02930224	792	492	801	
80	G#5	830.6093952	842.690887	1.014545335	80 x 1.01454533	801	501	81	
81	A5	880	880	880	81	81	51	821	



1
0.985659
1.0293
1.01455

This patch needs to have a midi number in its creation argument, which will be the lowest of three midi numbers that will be used to create the 4 microtonal notes of the 1/16th equal temperament scale.

the first note is the same as a midi note in equal temperament, so is played at speed 1

the second note is the next midi note up played at a lower speed, so creation argument +1, played at 0.985659 speed.
($y / 1.01455 = y * 0.985659$)

The third note is the creation argument + 1 played faster, at 1.0293 speed

The fourth note is within the equal temperament scale, so is the creation argument + 2 played at speed 1

The fifth note is the creation argument + 2 played at 1.01455 speed.

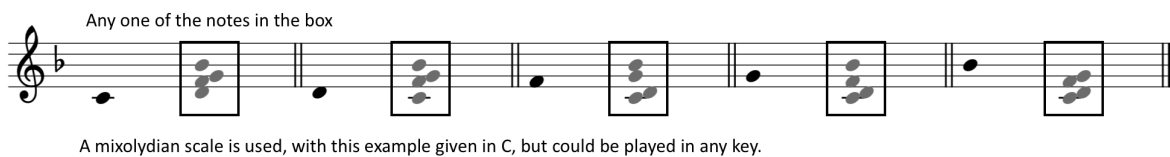
Appendix 5: Dataflow from *Montague III*

Montague III is made up of six 'players'. These are as follows:

1. Equal temperament player
2. Microtonal scale 13 (which fits 13 equally-tempered notes to an octave)
3. Microtonal scale 16 (which fits 16 equally-tempered notes to an octave)
4. Chords
5. Bass notes
6. Countermelody

1. Equal Temperament Player

This patch picks two notes randomly from a choice of the tonic, supertonic, sub-dominant, dominant, and (flattened) leading note of a mixolydian scale. These notes are sounded alternately, triggered by a rhythmic algorithm described below. The same notes cannot be picked twice (to avoid a single note repeating).



The rhythmic algorithm takes in the timebase and triggers on the number two, in the sequence. However, the sequence is shortened using the modulo object to either 3, 4, 6, 8, 12 or 16 at random, resulting in a constantly changing rhythmic pattern¹²². The length of this sequence normally changes every 24 ticks of the timebase, but occasionally changes after only 4 ticks (this is decided by a weighted random algorithm). A change is also triggered within one second, if the modulo of 3, 4 or 6 was chosen, to prevent rapidly alternating notes lasting too long, that would create a jarring hijacking of a melody that is normally shared across the three instruments.

These detailed algorithmic structures allowed me to achieve the balance of repetition against change that drives my compositional style.

2. Microtonal Scale 13-ET

This patch is very similar to the equal temperament patch, with the main difference that the notes are not randomly chosen, but start two degrees apart (on the microtonal scale) and increase step-by-step, and then decrease again. These steps aren't conjunct. The rising pitches follow the 2nd (starting note), third, fifth, seventh and eighth degrees of the

¹²² This is a variation of the rhythm generator algorithm described in chapter 2, Compositional Methods.

microtonal 13-ET scale. The falling pitches follow the tonic (starting note), -2, -4, -5, -7 and -8 degrees of the (13-ET) scale. These notes were chosen through experimentation alongside the equal temperament and the 16-ET scale, to build desired melodic and harmonic clusters.

3. Microtonal Scale 16-ET

This patch is also similar to the other two main players, except these notes are derived from the notes playing in the equal temperament player in the following way. The table index from the equal temperament scale is taken and added to, again with selected degrees of the scale (through listening and choosing the combinations that most appealed to me), then these are cycled up and down, using the same method as in the 13-ET player. The rhythm generator for this patch is the standard algorithm described in Chapter 2.

4. Chords

This patch was composed to add another layer of textural possibility, and is the only one that is homophonic. Two notes sound simultaneously, and are repeated at different timings. Below, the possible chords are shown in the key of C as an example (in the mixolydian mode) but when played within the piece, the key is determined in the master patch.



These are the same note-pairs as in the main equal temperament player, to reinforce a tonal centre. The rhythm of these chords is selected from a range of ten possibilities. The timebase provides a stream of numbers cycling from 0 to 48, and the choice to sound the chords is chosen from pairs of numbers within this range. Numbers that are multiples of three will sound as triplets against the regular timebase.

5. Bass notes

This patch was revised many times, to find the desired balance between coherence and chance. Initially, the bass played only the root note of the 'key' of the piece, which quickly became uninteresting. The note a perfect fifth above the root was added, but while still grounding the piece pitch-wise, did not add much interest. I then decided to use the pitch-detection object, Sigmund~ on the output of the whole piece, to catch whichever notes had recently sounded, and feed this back into the bass notes (using an octave adjuster to keep the notes within the lower range of the piano). The bass notes alternate between the tonic and whichever note is detected at the time. Rhythmically, the notes can sound on any multiple of four in the timebase, effectively tying the whole piece to a solid rhythmic structure. However, there is also an algorithm which acts as a 'holemaker' in the pattern,

occasionally preventing the notes from sounding, thus preventing the bass notes' pattern from becoming too monotonous.

6. Countermelody

This patch 'collects' the notes from the other players and plays them through the 13-ET playback engine (octave adjusted to play the higher range of the piano). Here, I have allowed for the complexity of the dataflow to take over the choice, in that the 'collection' of notes is actually the indices of the note tables, before note-lookup, or the root note is added, so the fact that the indices are being applied to the 13-ET scale make little sense. However, the shape of what has come before can still be detected, even if the actual pitches are not relatable.

Initially, the notes were captured, and played back only once, which led the piece to err too far on the side of chaotic chance, so repetition was implemented to give the listener a familiar phrase to enjoy before the piece moves on. The countermelody repeats four times before being silenced and capturing new notes.

Rhythmically, the notes sound as semiquavers, or the smallest rhythmical division (a tick), following an initial quaver. However, within the process, there is room for notes to 'drop out' (for example, after note-lookup, and adding octaves and the root, the note might fall out of range for playback, which will lead to a silence where a note would have played). This can introduce rhythmical quirks.