Portfolio of Original Compositions

Dynamic Audio Composition via Space and Motion in Virtual and Augmented Environments

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Humanities

2015

José Ignacio Pecino Rodríguez

School of Arts, Languages and Cultures

Table of Contents

Table of Tables

Table of Figures

Final Word Count: 15615

Portfolio of Musical Works

Interactive

Swirls	(2012)	*Concert Hall Interactivity*	Indefinite
Alice: Elegy to the Memory of an Unfortunate Lady[1]	(2012)	*Locative Audio*	indefinite
Alcazabilla	(2013)	*Adaptive Locative Audio*	indefinite

Non-interactive

Singularity	(2013)	*Procedural Composition*	05:23*
Apollonian Gasket	(2014)	*Procedural Composition*	06:32*
Boids	(2014)	*Procedural Composition*	07:11*

** These are the durations of the demonstration recordings included in the portfolio media files. The duration of the actual live performances of the pieces will generally be within a 5% margin of these values.*

Software Development

SonicMaps for iOS/Android	(2012)	*Locative Audio Editor/Player*	N/A

List of Performances and Presentations

Concert Performances, Soundwalks and Interactive Installations

BOIDS Concert Performance	*NYCEMF 2015* *New York City EA Music Festival*	New York, US	Jun 2015
XYZ[2] Interactive Installation	*NIME 2015* *New Interfaces for Musical Expression*	Baton Rouge, US	May 2015
BOIDS Concert Performance	*Sines & Squares Festival* *The University of Manchester*	Manchester, UK	Oct 2014
APOLLONIAN GASKET Concert Performance	*Salford Sonic Fusion Festival* *University of Salford*	Salford, UK	Apr 2014
SINGULARITY Concert Performance	*XX Festival Punto de Encuentro* *AMEE (Spanish EA Music Association)*	Málaga, Spain	Nov 2013
SINGULARITY Concert Performance	*MANTIS Fall Festival 2013* *The University of Manchester*	Manchester, UK	Oct 2013
ALCAZABILLA Soundwalk	*Locative Audio 2013* *The University of Manchester*	Málaga, Spain	Apr 2013

[1] Hereinafter referred to as "Alice"

[2] XYZ is an interactive implementation of Singularity, Apollonian Gasket and Boids.

SWIRLS Concert Performance	*Audio Mostly 2012* *Ionian University*	Corfu, Greece	Sept 2012
ALICE Soundwalk + Concert *Performance*	*Interactive Audio-Game Showcase* *Novars Research Centre*	Manchester, UK	Jun 2012
SWIRLS Concert Performance	*MANTIS Spring Festival 2012* *The University of Manchester*	Manchester, UK	Mar 2012

Papers Publications, Seminar Presentation and Invited Talks

'PORTFOLIO PRESENTATION'

Seminar Presentation	*Composers Forum* *University of Huddersfield*	Huddersfield, UK	Jan 2015

'SONICMAPS: TECHNICAL REPORT, CHALLENGES AND OPPORTUNITIES'

Symposium Presentation	*Meine Stadt, meine Klänge (ZKM)* *Zentrum für Junst und Medientechnologie*	Karlsruhe, Germany	Dec 2014

'SPATIAL AND KINEMATIC MODELS FOR PROCEDURAL SOUND IN 3D VIRTUAL ENVIRONMENTS'

Conference Proceedings	*ICMC 2014* *International Computer Music Conference*	Athens, Greece	Sept 2014

'NON-CONVENTIONAL 3D VIRTUAL INSTRUMENTS FOR ELECTROACOUSTIC COMPOSITION AND THE EXPLORATION OF SOUND GESTURE AS A RESULT OF REAL-TIME COMPLEX PHYSICAL BEHAVIOUR'

Seminar Presentation	*Procedural Audio Now!* *Skype HQ in Chancery Lane*	London, UK	Jul 2014

'LA ESCULTURA SONORA'

Seminar Presentation	*XX Festival Punto de Encuentro AMEE* *Asociación Música Electroacústica de España*	Málaga, Spain	Nov 2013

'SONICMAPS: CONNECTING THE RITUAL OF THE CONCERT HALL WITH A LOCATIVE AUDIO URBAN EXPERIENCE'

Conference Proceedings	*ICMC 2013* *International Computer Music Conference*	Perth, Australia	Aug 2013

'LOCATIVE AUDIO FOR SOCIAL RESEARCH'

Postgraduate Lecture	*Master in Visual Anthropology* *The University of Manchester*	Manchester, UK	Apr 2013

'LOCATIVE AUDIO: SOUND AND GEOLOCATION SYSTEMS'

Postgraduate Workshop	*Master in Visual Arts and Multimedia* *Universidad Politécnica de Valencia*	Valencia, Spain	Dec 2012

Contents of the USB Flash Drive

A USB flash drive is provided along this written commentary, comprising all media files for the pieces presented in this portfolio. This drive includes a digital copy of this document and a folder containing the six original compositions and the SonicMaps source code. With the exception of Locative Audio works, each piece is generally presented in multiple formats, which are organised as three separate subfolders: 'Source Code', 'Builds', and 'Demonstration Video Recording'. For performance reasons, it is highly recommended that the contents of the USB drive are copied to a faster HD or SSD drive before executing any of the files.

The 'Source Code' folder includes the original Unity project and JavaScript/C# scripts, and any other source material specific to the piece. This is a version intended for developers with some functional knowledge of the Unity development framework.

The 'Builds' folder contains stand-alone live versions of the pieces for OSX and Windows systems, that can be run without a copy of the Unity development framework. When it applies, it also includes the external audio engine files (e.g. SuperCollider or Ableton Live files), and any additional software (e.g. Pure Data patch or Android .apk application) that might be required to run the system. Set-up instructions and software requirements are also provided in a text file.

Finally, a high resolution demonstration video recording of the piece is provided for convenience.

In the case of Locative Audio compositions (*Alice* and *Alcazabilla*), a demonstration of the GPS soundwalk is presented as a video file. In addition, the actual interactive soundwalks can be experienced on location using the SonicMaps iOS/Android app, which is available at http://sonicmaps.org. The necessary links to load the pieces in SonicMaps are shown in Table 1, although an offline copy of these text files is also included in the USB drive along with the corresponding audio files.

Title	SonicMaps Link	Location
Alice	http://sonicmaps.org/u/1/projects/Alice.txt	Heaton Park, Manchester, UK
Alcazabilla	http://sonicmaps.org/u/1/projects/Alcazabilla.txt	Calle Alcazabilla, Málaga, Spain

Table 1. Links to Locative Audio works

Abstract

Electroacoustic music is often regarded as not being sufficiently accessible to the general public because of its sound-based abstract quality and the complexity of its language. Live electronic music introduces the figure of the performer as a gestural bodily agent that re-enables our multimodal perception of sound and seems to alleviate the accessibility dilemma. However, live electronic music generally lacks the level of detail found in studio-based fixed media works, and it can hardly be transferred outside the concert hall situation (e.g. as a video recording) without losing most of its fresh, dynamic and unpredictable nature.

Recent developments in 3D simulation environments and game audio technologies suggest that alternative approaches to music composition and distribution are possible, presenting an opportunity to address some of these issues. In particular, this Portfolio of Compositions proposes the use of real and virtual space as a new medium for the creation and organisation of sound events via computer-simulated audio-sources. In such a context, the role of the performer is sometimes assumed by the listener itself, through the operation of an interactive-adaptive system, or it is otherwise replaced by a set of automated but flexible procedures. Although all of these works are sonic centric in nature, they often present a visual component that reinforces the multimodal perception of meaningful musical structures, either as real space locations for sonic navigation (locative audio), or live visualisations of physically-informed gestural agents in 3D virtual environments. Consequently, this thesis draws on general game-audio concepts and terminology, such as procedural sound, non-linearity, and generative music; but it also embraces game development tools (game engines) as a new methodological and technological approach to electroacoustic music composition. In such context, space and the real-time generation, control, and manipulation of assets combine to play an important role in broadening the routes of musical expression and the accessibility of the musical language.

The portfolio consists of six original compositions. Three of these works–*Swirls*, *Alice - Elegy to the Memory of an Unfortunate Lady,* and *Alcazabilla*–are interactive in nature and they required the creation of custom software solutions (e.g. *SonicMaps*) in order to deal with open-form musical structures. The last three pieces–*Singularity*, *Apollonian Gasket,* and *Boids*–are based on fractal or emergent behaviour models and algorithms, and they propose a non-interactive linear organisation of sound materials via real-time manipulation of non-conventional 3D virtual instruments. These original instrumental models exhibit strong spatial and kinematic qualities with an abstract and minimal visual representation, resulting in an extremely efficient way to build spatialisation patterns, texture, and musical gesture, while preserving the sonic-centric essence of the pieces.

Declaration

I hereby declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or any other institute of learning.

Copyright Statement

The following four notes on copyright and the ownership of intellectual property rights must be included as written below:

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's policy on Presentation of Theses.

Acknowledgements

I would like to thank my supervisor Professor Ricardo Climent and Professor David Berezan for their constant support and guidance throughout my doctoral study and dissertation process. I am also grateful to Marta, my parents José and Mª Dolores, and my sister Ana for all of their love and unconditional support.

I would also like to express my gratitude to the Unity community, which was an invaluable source of knowledge and advice.

This research was funded by the School of Arts, Languages and Cultures and undertaken at NOVARS Research Centre, The University of Manchester, UK.

Written Commentary

1 Research Enquiry and Motivation

Like many artists before me, much of my creative work emerges from the observation and admiration of nature. Indeed, it will never cease to amaze me how the complexity of our reality, including abstract constructs such as music itself has its origins in a space-time singularity occurring approximately 13.8 billion years ago where nothing more than a vast amount of energy and a few fundamental laws initially existed. For this reason, many of the compositions presented in this portfolio are an attempt to emulate that creational activity by using some of the same elegant mechanisms that nature employs in order to create complexity out of simplicity (e.g. recursion, symmetry), eventually leading to surprising emergent phenomena that serve my purpose of musical expression. However, these pieces are also inspired on the exploration of the ultimate fabric of reality, thanks to the opportunities arising with recent virtualisation technologies (e.g. augmented and virtual reality), and on how these innovative ways to present audio-visual information might improve the impact these compositions produce on the listener. Therefore, as a starting point, two research questions are proposed:

1) Is it possible to increase the liveness, expressiveness and accessibility of electroacoustic music by using virtual or real space as a means for the dynamic generation and organisation of sound?
2) In that context, how can space and motion function as the main structural parameters?

All these compositions also have in common the development environment that has been selected for their making. I decided to switch from the conventional tools I had traditionally been using when composing electroacoustic music—which I found occasionally limiting or not transparent enough—to a software framework that would not impose any constraints to my creative impulse and would allow me to translate any possible musical idea into actual prototypes and finished works[3]. This usually means creating everything from scratch (e.g. materials, tools and transformational processes), but there lies, in my opinion, the beauty of this compositional approach.

2 Theoretical Framework: A Comparative Analysis

Before addressing the individualities of each piece, a general theoretical framework, containing the main underlying concepts, is presented. This is generally achieved through comparative analysis of opposing terms or ideas, including, but not limited to:

[3] Please see Chapter 2.2 for details about game engines.

- Augmented space vs virtual space.
- Interactive vs non-interactive.
- Acousmatic (unimodal/transmodal perception) vs audio-visual (multimodal perception).
- Cross-modal transfer vs cross-modal enhancement.
- Interactive instrument (compositional tool) vs actual composition.
- Open-form vs flexible linear form.
- Locative Audio (sonic topography) vs Procedural Composition (virtual instrument).

2.1 Real and Virtual Space as a Compositional Canvas

2.1.1 Extending Space

By definition, space is "a boundless, three-dimensional extent in which objects and events occur and have relative position and direction"(Encyclopaedia Britannica, 2015). We normally use this term to refer to the physical extent in which we live, and which is subject to Newtonian laws. Such an extent, and everything that exists within, is what we generally perceive as physical reality. However, with the introduction of computer-generated 3D environments, it is also possible to simulate an instance of space that conforms to the former definition and provides a new methodological framework where *virtual* location, motion, and gesture can be employed to generate musical structures. The virtualisation of sound-generating processes not only allows composers to simulate real-world sonic behaviours and musical instruments, but it also opens doors to creative experimental models that are just too difficult/expensive to implement in real life, or might even not be feasible using real world physics.

Nevertheless, virtual space and real space are only the two ends of a larger abstraction, commonly known as the *Reality-Virtuality Continuum* (Milgram et al., 1995, p. 283). Between these extremes we can find a range of mixed realities sharing elements from both ontological levels (Figure 1).

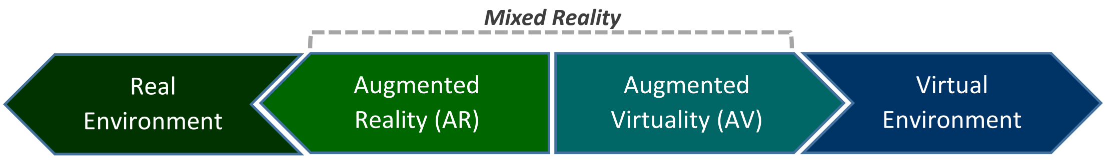


Figure 1. Milgram's Reality- Virtuality Continuum

More specifically, this portfolio deals with multiple manifestations of augmented and virtual reality, in which visualised space is either real or computer-generated. The first case is exemplified by two of the compositions included in the portfolio, *Alice* and *Alcazabilla*, where digital audio assets are integrated into specific locations of a real environment, using geolocated virtual sound-sources and GPS-enabled mobile devices for *sonic navigation*[4]. In these two pieces, the listener visually perceives a given real environment that is aurally being augmented (using headphones) to construct artificial, but meaningful, cross-modal relations (Figure 2).

Figure 2. Augmenting real space using virtual sound-sources (Locative Audio)

The rest of the Portfolio works take place in fully virtual environments that can be experienced in an interactive (*Swirls*) or non-interactive way (*Singularity*, *Apollonian Gasket* and *Boids*). In all cases, space is the main structural parameter and a central compositional aspect to this portfolio (see Table 2).

Augmented Space (Locative Audio)	*Alice*	Interactive (Sonic Navigation)
	Alcazabilla	
Virtual Space	*Swirls*	
	Singularity	Non-interactive (Procedural Composition)
	Apollonian Gasket	
	Boids	

Table 2. Classification of works in terms of space modality and interactivity

[4] This is the baseline for the SonicMaps-Locative Audio application that will later be discussed.

2.1.2 Motion, Gesture and Multimodal Perception

The use of spatial considerations in music composition became especially prominent in the 20th century[5], and most particularly, in electroacoustic music and sound art. This interest in space can be justified not only for its capacity to create depth, clarity, and other desirable musical features, but also for its structural and semantic functions[6].

Musical *gesture*, either physical or imagined, is also possible as a result of a bodily action that takes place in space, whereas musical *motion* can be seen as a conceptual metaphor of physical motion (Johnson & Larson, 2003). For instance, Trevor Wishart asserts that “musicians have always implicitly accepted some kind of (spatial) metaphorical or analogical link between perceived pitch and height. We speak about ‘high’ pitches and ‘low’ pitches and about melodic lines or portamenti going ‘upwards’ or ‘downwards’.”(Wishart, 1996, p. 191). This assertion somehow relates to Smalley’s ideas about motion in spectral space (Smalley, 2007, p. 45), and is actually made explicit in one of the pieces of this portfolio (*Boids),* where the height of a number of flocking particles, in 3D virtual space, is mapped to the frequency of their associated sound oscillators. Thus, some of the works proposed here are an attempt to transcend this metaphor by visually presenting the gestural agents of sound in the form of physically-informed virtual objects.

It is generally accepted that “stimuli from one sensory channel can enhance or alter the perceptual interpretation of stimulation from another sensory channel” in a process known as *cross-modal enhancement*[7] (Serafin, 2014, p. 2).

The extent of this cross-modal interaction can be appreciated, for example, in an interesting experiment by Schutz and Lipscom, who demonstrated that a percussionist playing a marimba has the ability to modify the perceived duration of auditory notes, depending on the visual information derived from different stroke gestures (Schutz & Lipscomb, 2007).

But most importantly, further studies indicate that we are more likely to gain the listener’s attention when two or more sensory channels are involved in the communication process, as compared to unisensory stimulation. This can be partially explained by the fact that mental processing and reaction times are significantly faster with multimodal experiences, so we are able to process more information. As a result, a stronger, richer, and more coherent perceptual signal can be obtained (Hecht et al., 2005; Andreassi & Greco, 2013).

---

[5] Please see Harley’s dissertation about space and spatialisation in contemporary music (Harley, 1994).

[6] As reflected in a survey about contemporary trends in the use of space in electroacoustic music (Otondo, 2008).

[7] This idea contrasts with Smalley’s *transmodal perception* (Smalley, 2007, p. 39) or Serafin’s *cross-modal transfers*, where “stimulation in one sensory channel leads to the illusion of stimulation in another sensory channel”(Serafin, 2014, p. 2).

These findings suggest that an audio-visual representation of sound events might improve the accessibility and understanding of electroacoustic music when the nature of these sounds is particularly abstract. The reason for this is not just the associated *added value* (Chion, 1994, p. 4,21), but our increased perceptual response to multimodal stimuli. In the case of musical structures with a strong spatial component, this is particularly relevant, given that spatial positions and trajectories can be better resolved as a result of the cognitive integration of visual and auditory information into a unified event. For example, if we consider a computer-generated 3D scene that is being rendered on a conventional 2D screen, our visual perception of depth and motion is usually compromised by a number of factors[8]. Therefore, aural information has an important role in clarifying and interpreting the visual scene, and vice versa (Ecker & Heller, 2005). This is one of the reasons why, in this portfolio, all compositions employing virtual space[9] are presented as audio-visual experiences, even if the sonic content is predominant, and some of them could have been delivered as acousmatic works[10].

By doing so, it is also possible to provide an innovative solution to Wishart's concern about the perception of motion in acousmatic music (Wishart, 1996, p. 202):

> *"… our aural discrimination of spatial position is not so refined as, for example, our discrimination of pitch. Particularly where the virtual acoustic space is projected on a limited number of loudspeakers (e.g. stereo or quadraphonic projection). Advances in computer simulation of spatial position or the general development of multi-loudspeaker concert halls may soon improve this position."*

2.1.3 Audio-visual Features

The last three compositions of this portfolio (*Singularity, Apollonian Gasket,* and *Boids*) present several points for discussion regarding their audio-visual nature. First, the visual component has not been "fabricated" by the composer as a separate entity. Instead, it is automatically generated, in real time, applying physical or mathematical models running in a 3D simulation engine. Synchresis (Chion, 1994, p. 63) does not emerge from an artificial or manual re-association of image and sound (e.g. film sound post-production), but as a natural consequence of the simultaneous generation of audio and video, resulting from spatial and kinematic events[11].

---

[8] E.g. frame rate, screen resolution, angle of vision, etc.
[9] *Swirls, Singularity, Apollonian Gasket* and *Boids.*
[10] Except for *Swirls*, which is interactive and relies on visual elements.
[11] Somewhat similar to what happens in real-world events.

Secondly, these pieces propose structural models that are intended to work as non-conventional 3D virtual instruments, so any elements not directly related with the production of sound and musical gesture are generally avoided or minimised. For this reason, the listening situation virtually resembles that of a live concert performance, where the audience visualises the gestures of a pianist pressing the keys, or a violinist drawing the bow. The main difference being here that the role of the performer is substituted by a semi-generative algorithm acting upon the virtual components of our experimental instrument.

The visualisation of virtual gestural agents (which also are sonified) has intentionally been performed using relatively simple geometric objects[12], in pursuit of a degree of abstraction in the audio-visual discourse. Such a minimal display also aims to emphasise the sonic-centric essence of the works, so that visual information does not become overly dominant. As previously suggested, visuals are mainly used to inform the auditory perception of space and motion through multimodal stimuli. In fact, the above-mentioned abstract quality of the audio-visual material can also be seen as a legacy of the author's background experience in acousmatic music—where there is usually room for personal interpretation or imaginary narratives. Therefore, the portfolio seeks an optimal balance between: the intention to improve the cognition of space and gesture in EA music, which here involves a partial disclosure of sound-sources; the need to increase the expressiveness and accessibility of sound-based[13] music (Landy, 2007, p. 21–35; Weale, 2006); and the wish to preserve the abstract quality of music as a desirable feature.

In the case of *Alice* and *Alcazabilla*, visual information is otherwise originated from a pre-existing real environment, as users navigate the physical space, so audio-visual cross-modal interactions are artificially devised by triggering selected sounds at specific locations. However, an accurate synchronisation between sound events and visual cues is technically hard to achieve, due to the limited accuracy of the GPS tracking sensor. Consequently, the focus is not so much on physical and musical gesture, but on the interactive exploration of augmented space via *asynchronous* sounds. The nature of these sounds is generally non-diegetic, in contrast with the above pieces based on virtual 3D instruments, although incidental or calculated synchresis effects might occur, resulting in a diegetic interpretation of sound.

Swirls presents a more elaborate and polished visual environment that is neither pre-existing nor automatically generated, but manually created with a clear aesthetic and functional intention. In this piece, we find original artwork, dynamic illumination, and particle effects, all of which aim to

[12] Cubes, rings and spheres with plain textures.
[13] See (Landy, 2007, p. 17) for a definition of sound-based music.

highlight multiple aspects of sound, reinforcing the interaction models, and ultimately embellishing the overall listener experience. Nonetheless, some of these elements (i.e. the swirls and branches) also possess an important structural function, so that the piece could not exist without them[14]. Moreover, it is observed that the interaction model and the global appearance respectively resemble the gameplay and aesthetics of some conventional video games (e.g. *Flower*[15]), although musical context is generally different (*Flower* uses tonal orchestral music, while *Swirls* follows the aesthetics of timbre-based EA music), and no explicit extra-musical narrative is here proposed.

In general terms, decisions on the adequacy of any sound for a particular one-to-one synchronous visual/audio event is primarily taken attending to its spectromorphological qualities and its capacity to convey a meaningful synchresis effect for the perceived gesture. These sounds could be classified into what Smalley defines as a third-order and remote surrogacy because their sources and causes cannot be easily inferred from the sound itself (Smalley, 1997). Sources are only artificially devised through the proposed audio-visual contract, yet the abstract nature of sound maximises the exploration of the electroacoustic medium.

2.1.4 Open Form versus Flexible Linear Form

One of the initial research questions of this PhD thesis investigates how virtual or real space can be used as a structural parameter to organise sound; however, this question can also be reformulated in terms of *listening modes* and *time* management.

Essentially, this portfolio presents two different ways of dealing with the concept of time: one considers time as an implicit property of space and motion (as explicitly stated in the equation: *space* = *velocity* x *time)*, so the time elapsed between two sound-events will depend on the speed with which the listener moves. Per contra, the other way makes use of time as an explicit variable that can be controlled using timers and scheduled procedures (see 4.2). In both cases, time lapses are not necessarily fixed, but flexible to a greater or lesser extent, and no traditional timelines—like those found in most conventional audio work stations, e.g. Pro Tools—are being used.

For instance, *Swirls* and both locative audio compositions—*Alice* and *Alcazabilla*—present sounds that have been meticulously arranged over virtual or real space, according to a predetermined structure—the one provided by the paths and branches in *Swirls*, or the selected outdoor

[14] Please see Chapter 3.1 for details about interaction design and technical implementation.
[15] http://thatgamecompany.com/games/flower/

locations in *Alice* and *Alcazabilla*. Therefore, these pieces have been conceived as open-form interactive compositions, with indefinite durations.

Conversely, *Singularity*, *Apollonian Gasket*, and *Boids*, are examples of non-interactive compositions with a *quasi-fixed* macroscopic form, and a linear discourse that is apparently similar to the sort of linearity we can observe in fixed-media/acousmatic works. However, as discussed below, the procedural compositional techniques employed provide them with a certain degree of microscopic variability that fixed-media compositions do not generally possess, unless some further manipulation is performed at a later stage—e.g. live sound diffusion[16]. For this reason, their total duration is very much approximate, with a few seconds variance.

The distinction between structure and form, along with the role of indeterminacy and chance procedures in contemporary music, was notably manifested by John Cage in his general aesthetics when considering up to four distinct compositional elements, i.e. *structure*, *material*, *method,* and *form* as the basic pillars of his indeterminate works (Boulez & Cage, 1995, p. 39). The connection between these four elements is then clearly and elegantly encapsulated by Nye Parry in the following quotation (Parry, 2014, p. 4) :

"…form results from method acting on materials in a given structure."

The actual perceived form of any of the pieces of this portfolio—either at the macroscopic or at the microscopic level—is just one of the many possible outcomes arising from their respective structures, the final form depending on how the specific methods are applied to the actual materials.

In this regard, the effective relation between space and time can also be explained by virtue of the existing dichotomy between structure and form, since the listener's temporal perception of the piece will mostly depend on how the spatial structure defined by the composer is materialised, as a particular form, during its performance (e.g. total duration, or elapsed time between sound events). In other words, these works propose structures that take place in space, while resulting forms exist in time.

If we consider the listener's point of view, it might also be helpful to differentiate between two possible perspectives or listening modes in connection with motion, space, and our perception of time and musical form. The first one proposes a sonic topography where static sound-sources can be found as we, the listeners, move around it. That is the case, for instance, of the above-

[16] Please see Mooney's evaluation of sound diffusion systems for the live performance of electroacoustic music (Mooney, 2006).

mentioned open-form interactive compositions[17]. On the contrary, the second listening mode is a passive one, where the listener is now static, and it is the agents of sound who are actually moving—e.g. the spinning rings in *Apollonian Gasket*, or the flocking particles in *Boids*. In this second setting, the position of the camera and virtual listener are both generally fixed, and the latter is always situated at the centre of the scene so as to capture an immersive sonic image of the surrounding objects[18]. The type of motion also differs from one listening mode to another because the first one involves a change in the listener's location (horizontal translation motion), whereas the second one is a gestural movement associated to the active components of a 3D virtual instrument, which is anchored in front of a static listener.

These ideas are directly linked to the previously discussed metaphor of physical motion, which Johnson & Larson now extend into a 'moving music' metaphor, and a 'musical landscape' metaphor (Johnson & Larson, 2003, p. 69–71). Once again, at least in the case of a musical landscape (static sound-sources with a moving listener), our implementation is not metaphoric but a real one, using 3D Euclidean space.

2.2 Game Engines as a Creative Framework

In recent years, developments in what is known as 'game engines' have been significant. Game engines were originally devised for video games development, but given their power and versatility, they have become increasingly popular in various creative contexts, especially since the release of open-source and free alternatives of some of the most popular engines now available (e.g. Blender, Unity or Unreal Engine 4).

Essentially, a game engine is a software framework that comprises at least a graphics rendering engine, a physics engine, some scripting language, and an audio engine, although it will usually also include an animation engine, support for human interface devices (e.g. joystick), an AI system, and many other additional features. Thanks to all of these features game engines are very convenient creative tools, particularly when working with interactive models, simulation, or real-time representations of space and motion.

For this reason, we have seen how a number of composers and sound artists have embraced the use of game engines as a compositional tool, particularly in the field of interactive music. Notable examples can be found in the works of Andrew Dolphin (Dolphin, 2011), Ricardo Climent (Climent, 2009, 2014), Julio d'Escriván (d'Escriván, 2011, 2014) or Rob Hamilton (Hamilton, 2014).

[17] *Alice*, *Alcazabilla* (augmented space navigation), and *Swirls* (virtual space navigation).
[18] Details on actual multichannel rendering (e.g. 5.1) and the virtual listener will be discussed in a later section.

From this state of things, the present research and portfolio make extensive use of these technologies in an attempt to extend the language and practices of electroacoustic music, while addressing some of the aforementioned issues. For this purpose, the game engine of choice was always Unity[19] (version 4.x), and these are some of the areas explored:

Open-Form Composition

- Use of augmented space for EA music composition (Locative Audio), including 3D spatial audio and adaptive sound (Dynamic Content Server).
- Development of adequate tools for Locative Audio composition (SonicMaps app for iOS/Android).
- Sonic navigation of virtual space in the concert hall with multi-user interaction.

Non-interactive Composition

- Use of spatial and kinematic models/algorithms as non-conventional 3D virtual instruments for procedural sound.
- Procedural composition of non-interactive EA music (Quasi-fixed/Live-Fixed Media) using C#/JavaScript scripting.

2.2.1 Musical Instrument/Compositional Tool versus Actual Composition

At this point, an additional question arises in regards to the demarcation of the boundaries between musical instrument, compositional tool, and actual composition, especially when the actuators involved (i.e. instrument/tool maker, composer, performer, listener) may be the same person, as is the case in some of the works discussed below.

This issue was already considered by Dolphin when positioning his interactive "sound toys" somewhere in the intersection between these three entities (Dolphin, 2011, p. 37–38). However, the ultimate identity of some of these works is generally unclear and open to interpretation, as is the role of the listener/player, who most of the time is taking compositional decisions through the playful exploration of the sound toy.

In my opinion, the ambiguity arises when trying to equate the manipulation of a novel musical instrument or compositional tool with the exploration of an open-form composition. If the nature of the musical system[20] is not clear enough, it might be difficult to discern: for example, if a listener is performing a pre-existing open-form composition, or if s/he is consciously or

[19] http://unity3d.com (Accessed 23 March 2015)

[20] A musical system being any set of things working together with a musical intention, so it may well include a musical instrument (physical system made of material components) or a musical composition (abstract system made of musical ideas).

unintentionally improvising[21] a musical piece by operating (or just playing with) the provided instrument/tool.

Attending to the etymological origin of the word "compose", we note that it is derived from the Latin word "componere" which means "to put together" (com- "together" + ponere "to put"). Accordingly, music composition can be defined as the intentional act of putting sounds together into a particular coherent structure. Furthermore, the separation between performance and composition may be explained in terms of form and structure, as discussed above, thus form being the temporal materialisation of a composed structure through a performative act, which usually involves some kind of instrument.

Therefore, the answer to this issue probably lies on the nature of components comprising the musical system. For instance, a luthier making a violin is indeed "putting together" (composing) a number of parts into a coherent structure, but the resulting object cannot easily be regarded as a musical structure, because no sounds have been employed for its making. This is otherwise considered an instrumental structure—the violin—with a connate potential to produce sounds; that is the case, for example, of the non-conventional instrumental models used to compose the last three pieces of this portfolio: *Singularity*, *Apollonian Gasket* and *Boids*. Nevertheless, when designing a musical instrument, its designer will normally have a clear picture of the variety of sounds that the instrument will be capable of, so the components are assembled accordingly. That means that a direct relation between the instrument's structure and the associated sounds seems to exist; therefore, the "luthier" could somehow be referred to as a "sound designer"[22]. In that scenario, an instrument is now seen as a "potential sound" (or group of sounds), suggesting that the distribution and organisation of a number of instruments (potential sounds) into a coherent structure can also, by extension, be considered a type of music composition. This is the case, for example, of the spatially distributed sound generators that create the musical structure of the interactive work *Swirls*. Actual sounds do not exist, "a priori", in the structure of this piece, but they are procedurally generated, "a posteriori", as players navigate the multiple sections of a predetermined (composed) structure. The composer has previously decided what sounds will be available in each section (group/cluster of swirls), and how these sections are interconnected with each other, in order to dictate the possible sequences (i.e. the "structure" of the piece); hence the players have only control over the particular order and tempo in which the different sections will be performed (i.e. the "form" of the piece).

[21] According to the ideas exposed in 2.1.4, an improvisation can be regarded as a compositional act where structure and form are being created virtually at the same time, and by the same person or agent.
[22] This idea directly connects with Jordà's dissertation about constructing tools for musical performance and improvisation using computers (Jordà, 2005).

2.2.2 Spatial Sound Using Virtual Sound-Sources

The importance of space and motion for this portfolio has already been made explicit in the previous chapters. The following section now discusses the practical implementation of spatial sound using game engines, as an alternative to traditional spatialisation techniques[23] where the number of audio channels of a composition usually needs to be decided beforehand, since this decision has technical and aesthetic consequences for the conception, delivery, and performance/playback of the piece.

The spatialisation of sounds in the pieces of this portfolio is thus conceived as a direct consequence of the actual 3D positions and the physical movement of a number of virtual objects acting as sound-sources in the simulation environment. Consequently, the number of audio output channels can be easily modified at any time, and the signal energy distribution will automatically be calculated according to the relative position of the previous objects with respect to a reference object known as the audio-listener, which functions as a sort of virtual omnidirectional microphone within the scene[24]. This feature facilitates the creation of multiple versions of a piece with different numbers of output channels (e.g. stereo, quadraphonic, 5.1, octophonic), while avoiding the usual phase-related issues associated with typical up-mixing or down-mixing processes (e.g. 5.1 to stereo) in conventional fixed-media. Furthermore, and what is most important, it is suggested that the object-based physical representation and manipulation of sound-sources facilitates the spatial conception of these works, improving the coherence and clarity of sound in terms of spatial motion and localisation, even when the visual component is hidden to the audience.

The automated calculations involved in the audio signal distribution for a particular multichannel configuration can normally be executed by the game engine itself using its spatial audio engine[25]. This calculation takes into consideration the distance and the angular position (azimuth and elevation) of the sound-source with respect to the audio-listener, so the audio signal is attenuated and panned accordingly, in a similar way to what happens in a real world situation. The closer the sound-source is to the audio-listener, the louder it is, and vice versa.

However, the internal audio engines included in these development environments (e.g. FMOD in Unity) usually have significant limitations in terms of audio signal processing capability, since, at the time of writing, they are primarily designed to playback recorded audio samples, and perform

---

[23] E.g., the manual panning of an audio signal using a mixer console or a conventional DAW such as Pro Tools.

[24] Analogous to the speaker-independent, full-sphere surround technique found in Ambisonics (Fellgett, 1975).

[25] That is the case of *Alice*, *Alcazabilla,* and *Singularity*.

simple transformations such as pitch/duration variations (resampling), amplitude scaling, and, in the best-case scenario, basic audio filters and reverb zones. Consequently, it is common practice to overcome these limitations by integrating into the system an external audio processing software (e.g. MAX/MSP or Supercollider[26]) with greater flexibility and capabilities to perform complex audio processing tasks (e.g. real-time physical modelling synthesis).[27] Both applications can then be interconnected using the OSC[28] protocol via a local network, whether they are on separate computers or sharing a single machine.

When the internal audio engine is overridden, its 3D spatial audio functionality, as previously discussed, is inevitably lost, and needs to be re-implemented. The solution considered in some of the pieces of this portfolio is to translate the game engine's Vector3 positions (x, y, z) into angular positions (r, θ), and subsequently send this data to Supercollider where sounds can be panned into a custom array of loudspeakers using, for example, the VBAP object (Pulkki, 1997). The downside of this approach is that our piece will no longer work as a single standalone application, but it will depend on external software to generate sounds. This is, however, a distribution issue and not a compositional one.

2.3 Dynamic Audio and Structural Levels

Along with the crucial role that the previous spatial considerations have in the development of this research, a strong focus has also been on the very essence of sound production itself, and how sound-sources/processes are assembled and presented to an audience in an appealing and innovative way.

A common feature can be observed throughout the portfolio: sounds and music are dynamically created, as opposed to what we experience when listening to fixed-media compositions. That means that every time one of these pieces is performed, the sonic outcome is to some extent different and not exactly reproducible. This is indeed a natural feature of live music, regardless of the genre, and it is generally believed that the listening experience benefits from it, mainly because what we hear is somehow unique, and we can witness how music is being created[29]—which is actually related to the earlier discussion on the multimodal perception of sound and music.

[26] SuperCollider is a programming language for real time audio synthesis and algorithmic composition. Please visit http://supercollider.github.io/ for further information.
[27] This is the case of *Swirls*, *Apollonian Gasket*, and *Boids*.
[28] Open Sound Control specifications available at http://opensoundcontrol.org/ . This portfolio uses the UnityOSC C# classes interface by Jorge Garcia: https://github.com/jorgegarcia/UnityOSC
[29] An informal but interesting commentary on this subject can be found at http://kidsmusiccorner.co.uk/2010/02/01/why-is-live-music-so-much-better-than-a-cd/

The term dynamic audio has been defined by Collins as "non-linear, variable elements in the sonic aspects of gameplay" and can be effectively divided into interactive and adaptive audio (Collins 2009, p.5).

In the case of interactive compositions, variability is an inherent property, subject to the choices made by the users interacting with the system. On the other hand, the non-interactive compositions obtain their indeterminate quality from the dynamic implementation of sound and upper musical structures. In all circumstances, analysis of these pieces is considerably facilitated if a structural differentiation is introduced. Therefore, we will generally consider up to three different levels.

The first and lowest level is the inner structure of the sound itself[30], whose ultimate morphology can be automatically set, for example, using a synthesis model fed by real-time spatial and kinematic data (e.g. procedural audio works), or it can simply be obtained by modifying a few basic parameters of a pre-existing sound recording (e.g. *Singularity*, where pitch, amplitude, and panning of selected micro-samples are constantly being adjusted).

The second level emerges from the concatenation or superimposition of multiple sound-sources or processes which have the capacity to produce evolving textures, motifs, and other simple musical ideas; we will refer to this level as the *microstructure* of the piece. In the interactive work *Swirls*, it is the users/listeners who introduce the indeterminacy factor, since each individual decides when and which sound-source is to be triggered—sometimes in a collaborative way—within a particular area or section of the piece. By doing so, they are actively specifying the actual form of that precise section.

Alternatively, the dynamic interpretation of the microstructure of non-interactive works is completed by running certain generative/kinematic algorithms (e.g. a cellular automata algorithm in *Singularity,* a flocking algorithm in *Boids*), or a distinct fractal/kinematic model (e.g. *Apollonian Gasket*). These algorithms and models govern the base behaviour of the system components (e.g. the selection of active cells in *Swirls,* the movement and distribution of the rings in *Apollonian Gasket,* or the discrete trajectories and velocities of each individual particle in *Boids*); but they are also complemented by a number of random elements so as to increase the complexity and variety of the final musical outcome (please see details below about the implementation of these models and random methods).

A special situation is observed in the case of locative audio compositions, as their microstructure can be seen as predominantly static, because of the internal logic of large pre-existing audio files.

[30] Please see Roads (2004) for a comprehensive discussion about microsound.

However, if the fixed sequence of sounds included in any of these audio file is sufficiently fragmented over several adjacent audio zones, human interaction at a microscopic level can again be re-enabled. In that case, the composer is just relocating the compositional job, from the arrangement of multiple sounds into a linear audio file, to the organisation of those same sounds into spatially distributed sound areas (non-linear microstructure). A balance between these two strategies is generally desirable in this category of locative works.

Finally, the third upper level, or *macrostructure*, contains the larger sections of a piece, whether these are aggregates of static sound-sources in 3D virtual space (spatial structure), or a flexible sequence of instructions in a C# script, concatenating a large number of sound events/processes into a single time-domain macroscopic unit (temporal structure).

Notice that we are again making a differentiation between structure and form, where form–that is, the ultimate musical and sonic outcome perceivable by the listener–arises from the dynamic interpretation of the existing structures (at any level), using any of the given methods. This is true no matter who the interpreter or performing agent is: a human, or a computer running a generative/random algorithm. In all cases, the underlying sonic or musical structure has been designed and created by the composer.

For classification and clarification purposes a glossary of terms is provided at the end of this thesis in order to define some of the terms previously mentioned, as they will also be employed during further analysis of the pieces below.

3 Interactive Open-Form Composition

The following two chapters will discuss three open-form compositions focusing on the idea of Sonic Navigation, as the act of navigating the space with sound as the main referential framework and purpose. Sonic Navigation can be conducted both in virtual and augmented space, and it involves the active exploration of a spatial musical structure containing a number of static virtual sound-sources. The interaction model will differ in each case, depending on the nature of the navigated space (i.e. virtual or real), so custom solutions will therefore be proposed.

3.1 Concert Hall Interactivity — *Swirls*

3.1.1 Introduction

Concert Hall Interactivity is the descriptive name given to the musical experience proposed in the piece *Swirls*. This work explores the creative possibilities of an interactive music composition based on a 3D virtual environment that can be navigated by members of an audience, in a concert hall, using smartphones as wireless input interfaces. The concert situation connotes a narrative discourse with an approximate fixed duration, and a traditional distribution of the members of the audience sitting in a medium or large concert hall—as opposed to a sound installation where visitors are free to move around without time constraints.

A custom interactive system was developed for this purpose, including an Android application enabling the interaction of a few members of the audience with the musical structure, no matter where in the room they are situated.

This project presented a number of technical and compositional challenges in regards to time management and good continuity. Specifically, it was required that the piece could be performed in less than 10 minutes, a self-imposed target duration which is a common length for computer music festivals. Compositionally-speaking the problem to solve was also to design a system where the musical outcome was coherent, sonically rich, and engaging for the non-interacting members of the audience who function as a control group. These goals were achieved thanks to the particular structure of the piece, and the introduction of specific methods, system behaviours, and constraints in the interaction model, as described below.

3.1.2 Structure and Interaction Model — Virtual Space Sonic Navigation

As a starting point, the structure of the piece is built upon a series of paths and "swirls" (spiralling patterns), distributed along a bespoke bi-dimensional tree-like shape, which is displayed on the virtual terrain of a computer-simulated 3D environment. Each of these paths corresponds to one of the branches of the tree, and the main axis (trunk) is progressively broken up into smaller sub-branches containing multiple clusters of swirls (Figure 3).

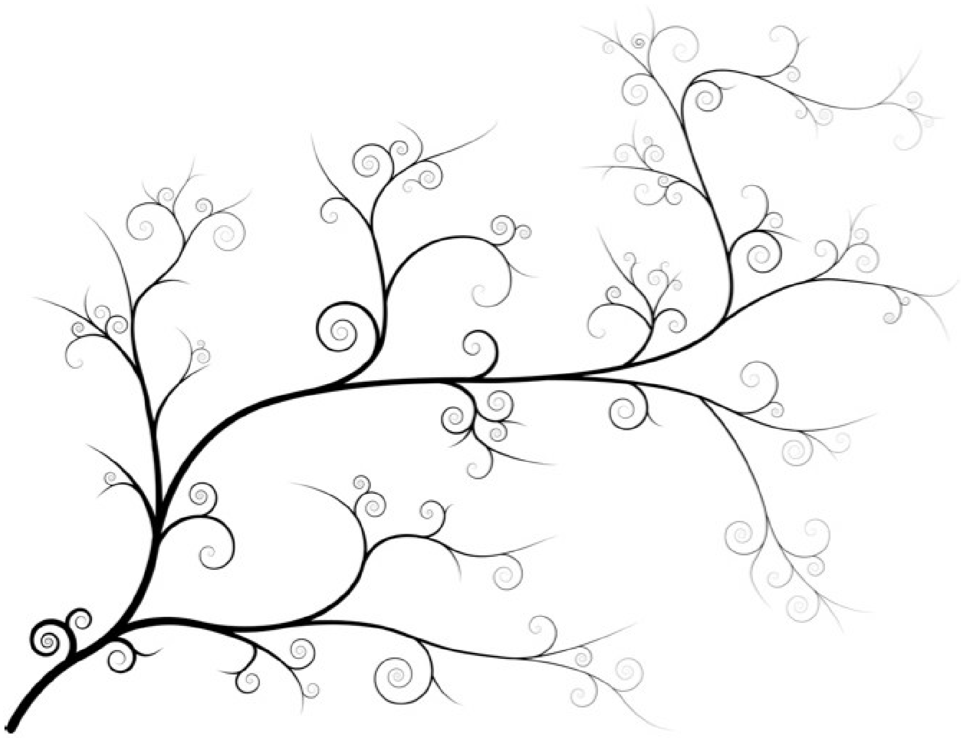


Figure 3. Swirls spatial structure (original artwork)

Participating members of the audience[31] are then invited to freely navigate the proposed structure, where sound events are triggered whenever any of the users—represented as floating spheres of coloured light—enter or hover the proximity of a swirl (Figure 4). In order to move their corresponding light spheres, users simply need to tilt their smartphone into the right direction, so that the accelerometer sensor reading can be obtained and sent to the server (further technical details below).

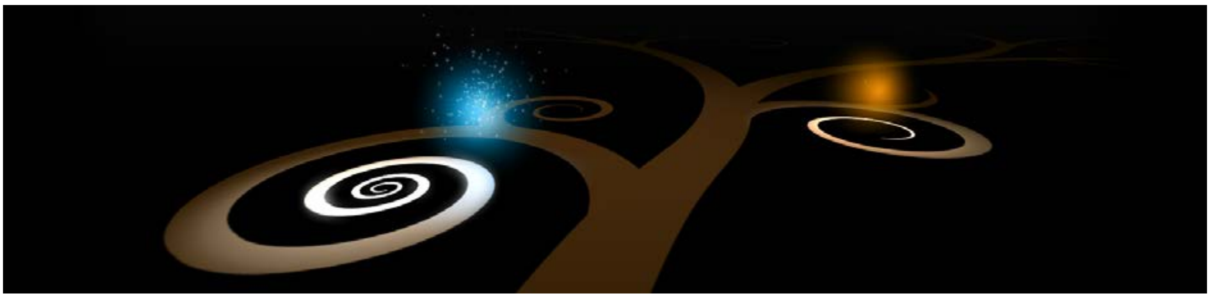


Figure 4. Users hovering swirls to play sounds

[31] Up to three audience participants in the current implementation, as proposed for the premiere; however, the actual maximum number of participants is only limited to the capacity of the local network and the processing power of the machine running the simulation, so it could be considerably higher.

In this scenario, users have direct input access to the music system, and the interaction model could be classified as *performative*. This is equivalent to saying that the users' intention when approaching any of the swirls is to produce an imminent sonic outcome. The instrumental function of the swirls is thus evidenced, and the overall structure can be seen as an aggregate of discrete musical instruments organised over space, as discussed earlier.

An emerging soundscape, pre-composed with several non-visible sound-sources, can be found at the end of every main path/branch, as an alternative sonic interaction model where users need to rely completely on their ears to trigger the corresponding sounds, in an almost acousmatic experience. The piece will be complete once the users have revealed all the sounds comprising the soundscape, which is indicated by automatically fading out the audio and video signals.

3.1.3 Agency, Constraints and Progression

Given the freedom granted to the performing members of the audience, one of the main challenges is finding valid strategies to predict user's behaviour and maintain a sense of progression towards the completion of the piece. It might happen that a particular user decides to stay for too long in a specific area, progresses too fast, or simply gets lost beyond the limits of the visible paths. In any of these situations, an additional issue arises: the point-of-view camera might not be able to follow the movements of all users at the same time, and the listening reference might also be ambiguous. The proposed solution to these potential issues includes the following points:

- Setting the camera to focus on the centre of mass defined by all the light spheres. The audio-listener (invisible object) is equally set at this point, and will move with it.
- Automatic camera zoom (in and out) depending on the distance between the spheres, so no one is left outside the field of view.
- Introducing a mist effect in the scene, so that any user moving too far away from the rest of participants will no longer see the corresponding sphere, adding a motivation to remain together.
- Finally, scripting an attraction force, which is proportional to the distance from the center of mass. The further a sphere is from the barycenter, the stronger this force will try to push it back towards the group (this will only happen beyond a certain distance threshold). Thus, if two out of three users decide to move their spheres forward, the third participant will be forced to follow them.

As a result, collaborative strategies are expected to emerge from the participants involved, suggesting a form of swarm Intelligence behavior (Dorigo & Birattari, 2007). The average user will tend to move forward, once the sonic possibilities of a cluster of swirls have been fully explored and exhausted. Therefore, regardless of the chosen path, the piece might eventually come to an end within the expected time frame. The final duration will mostly depend on the length of the existing paths and the number of swirls to be explored, but a more accurate estimation can be obtained during rehearsals.

3.1.4 Sound Materials and Technical Implementation

Every local cluster of swirls (neighbouring swirls) presents a specific typology of sounds, provided by the corresponding associated sound generators running in Ableton Live (external audio engine). These are generally abstract sounds, although they can be categorised into two main groups: *bell-like* sounds—with a fast attack and prompt long release—and *sustained* sounds—with a variable attack, indefinite sustain time (as long as the user hovers the swirl), and a generally shorter release.

All these sounds are procedurally synthesised[32] according to multiple "game" parameters such as:

- Identification of the light sphere entering the swirl area. Variations of a same sound generator are employed to produce a similar but distinct timbre depending on who is triggering the sound.
- Size of swirls. This parameter has a direct effect on the relative pitch of the associated sound, although the actual value is returned using a random process to avoid exact repetition.
- Velocity of the light spheres when entering or hovering a swirl area. It has multiple implications on the spectromorphology of the resulting sound, including brightness, intensity (amplitude) and attack times (ADSR envelope).
- Time elapsed since a swirl area is entered. Considered only for sustained sound-shapes in order to slowly evolve some parameters of sound, while the user remains in this area.

The nature of sound will also depend on the particular synthesis method being employed, ranging from simple FM/additive synthesis (i.e. Ableton *Operator*), to a more complex physical-modelling synthesiser (i.e. Ableton *Collision*). A number of effects were also introduced into the audio signal chain to modify the properties of sound. Both synthesisers and effects processors include automated parameters reacting to incoming data for real-time dynamic sound.

In contrast, all sounds composing the final soundscape areas are not synthetic but original recordings[33], and they have been implemented using Unity's spatial audio engine. The listening

[32] Please see the Glossary of Terms for a definition of Procedural Audio.

mode is no longer performative, but mainly explorative, in a similar fashion as it happens in the locative audio works discussed below. Light spheres also possess a characteristic sustained sound employing the same internal spatial audio engine, so their movements can aurally be tracked.

The link between Unity (game engine), Ableton Live (external audio engine), and the custom Android application (user input interface) is facilitated by a single PureData patch that routes the multiple available data streams (Figure 5).

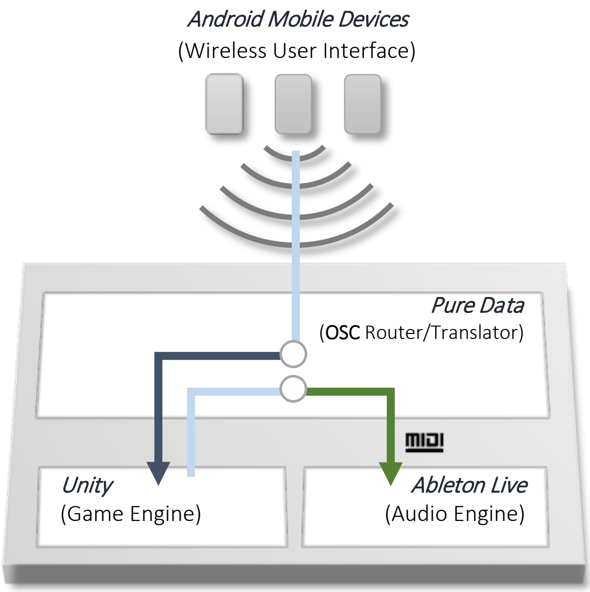


Figure 5. Swirls system overview

For instance, the incoming OSC messages from the game engine need to be adequately translated into MIDI messages for their use in Ableton Live. This can be easily achieved thanks to the Pure Data "noteout" and "ctlout" objects, as shown in Figure 6.

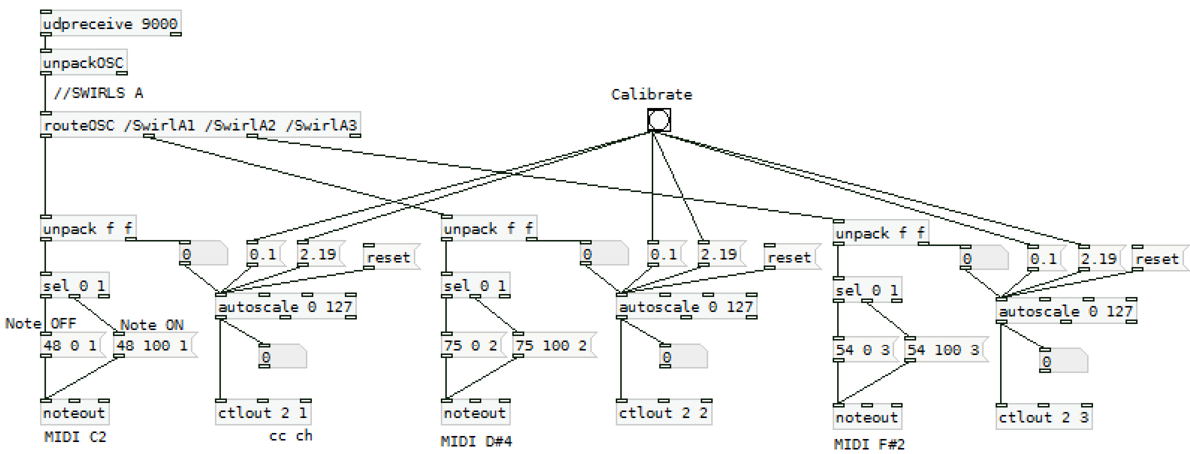


Figure 6. OSC to MIDI translation in Pure Data

[33] With the exception of a vocal choir glissando and a Gregorian chant, both of which are pre-cleared samples included in the *Symphony of Voices* collection by Spectrasonics.

The custom Android application consists of a simple initialisation screen, where the network connection settings can be entered (IP address and UDP port); and a main control screen, which simply displays a short message ("Tilt your device to navigate") to instruct the users on how to operate their corresponding light spheres, while implicitly suggesting the right orientation of the device. The main screen also has a distinct background colour to identify the associated light sphere (Figure 7).

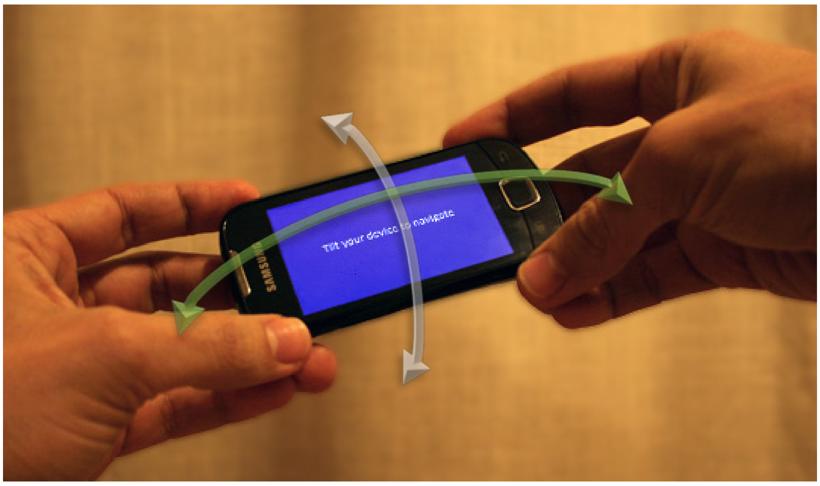


Figure 7. Android device as a wireless input interface

As previously indicated, this application uses accelerometer sensor readings to detect how much is the device being tilted on the main orientation axis. Those values are sent to the Pure Data patch as OSC messages, and subsequently formatted and routed to the game engine so that we can apply motion forces to the corresponding light sphere (Figure 8). This procedure has proven to be an effective, intuitive control method, with a minimal learning curve, which is essential for the concert hall experience here suggested. It also reinforces the performative nature of the interaction by linking manual gesture, on-screen movement, and sonic output, as the user has control over the amount of force applied to the light sphere, and how fast will this sphere approach a particular swirl.

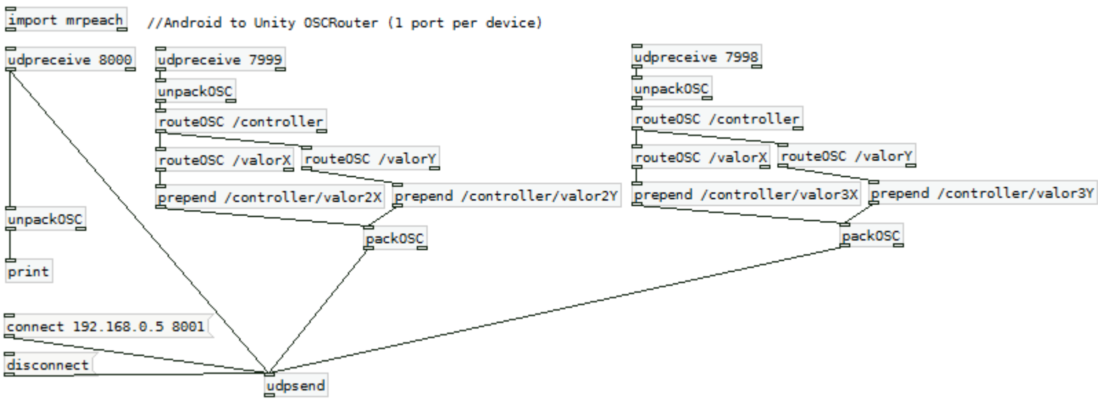


Figure 8. Routing and formatting of accelerometer data in Pure Data

3.2 Locative Audio — SonicMaps

3.2.1 Introduction

Location-based audio, as a compositional practice, is the creation of situated listening experiences where a virtual layer of auditory information extends, modifies, or enhances our perception and understanding of a particular place. Through this augmentation process "composers can recover memories of a particular place, can produce sonic alternatives to repositories of visual information; and even attempt to forecast desired futures through sound"(Climent, n.d.).

As an audio-visual experience, it is often considered an extension of film and gaming music/audio because it shares numerous stylistic elements with them (Paterson & Conway, 2014, p. 4,6)[34]. However, additional similarities can be observed in connection with soundscape composition (Schafer, 1977; Truax, 1992) and acoustic ecology (Wrightson, 2000), as sound serves to mediate the relationship between individuals and their environment through a conscious and immersive listening experience.

Locative audio composition uses real physical space as a medium for the organisation and presentation of sound materials, here presented as geolocated virtual sound sources; therefore, resulting spatial structures are strongly influenced by the landmarks and particular characteristics of the selected locations[35] (e.g. the streets of a town or city, the paths or natural features of a park or forest, etc.). These augmented environments are then explored and navigated using physical movement as an interface utilising GPS-enabled mobile applications which track our geographic position to playback audio files as users walk into predefined areas. Thus, users are allowed to create their own sonic version or remix of the placed sounds by choosing specific paths and by adjusting their walking speed[36].

The initial motivation for this research strand arises in the context of the Manchester's Sonic Meta-ontology[37] event (June 2011), which included a number of locative audio works by NOVARS[38] composers using the NoTours[39] audio-guides system. This tool allowed the geolocation of audio files using GPS tracking on an Android device, although a number of technical limitations

---

[34] For example: the typical division of sound materials into background sound (providing emotional mood), sound effects (reflecting the space) and dialog; the simultaneous presentation of diegetic and non-diegetic sound; or the interactivity and the characteristic sense of immersion of game audio in order to fully experience the media narrative.

[35] Please see details below about the particular structure of the two locative audio compositions here presented

[36] See Behrendt's discussion about locative media and walking as remixing (Behrend, 2012).

[37] http://acusmatica.7host.com/locative/2011June/index.html

[38] NOVARS Research Centre, The University of Manchester, UK.

[39] http://www.notours.org/

and reliability issues occasionally hindered the creative and listening activities. Therefore, in order to respond to these limitations/issues I created SonicMaps: a new sound geolocation platform with a focus on the implementation of 3D spatial sound for locative audio (increased immersion), workflow improvements, and new content sharing/publishing options for better usability and accessibility.

3.2.2 SonicMaps as a Compositional Tool

'SonicMaps Editor' is the iOS/Android application that I started to develop in 2012 in order to facilitate the creation and deployment of Locative Audio compositions (i.e. *Alice* and *Alcazabilla*). It works in conjunction with an online publishing platform[40], where users can easily share their projects (Figure 9). The SonicMaps open platform currently includes more than 500 registered users and 223 published projects around the world as of the writing of this commentary.

Figure 9. SonicMaps website (projects page)

[40] http://sonicmaps.org

The iOS/Android application basically works by displaying the user's position on a map (using the GPS sensor) along with a number of sound-sources, represented as circular areas, to be discovered as the user walks.

In order to geolocate sounds, I implemented a geospatial positioning system by translating the geographical coordinate system used by GPS mobile devices (WGS84), into Unity's Vector2 Cartesian coordinates (x,y). This was completed using the Mercator[41] projection as I wanted to use Google Maps images to render the navigation display. However, everything had to be scaled down given the single-precision floating-point format that Unity uses for object transform positions[42]. That means you can only have seven digits to set a coordinate and anything longer than that will be truncated with a consequent loss of valuable information. Additionally, I tried to avoid using absolute coordinates from a fixed frame of reference and opted for a relative positioning system that takes the first GPS coordinates (on application start) as a relative frame of reference. This value is stored on every session, so if the application is run somewhere else and the initial reference point changes, all objects positions can be recalculated without exceeding the seven digits limitation.

When starting the SonicMaps Editor application, it will first try to determine the user's current position. The message "Searching for Satellites" is displayed on the screen so location services needs to be enabled on the device before we run the application. It is also necessary to check that GPS signal is strong enough; otherwise, after 30 seconds, the user is asked to check location settings and restart SonicMaps[43]. Once the current location has been set, the main user interface layout is displayed.

A map tile for the current location is obtained from the Google Static Maps[44] service using a Wi-Fi or 3G internet connection, and it works as a texture for a simple plane mesh whose dimensions conform to the considered scale. On the other hand, the user's position is displayed on screen as a small blue arrow that reads the incoming GPS and compass sensor signals and updates its transform (position and Euler angles) correspondingly. If the user walks "off the screen", the message "Updating Position" will automatically pop up, and the map will be re-centred.

Tall buildings, mountains, and other objects often interfere with or interrupt the GPS signal from available satellites. Furthermore, Google maps are not always 100% accurate so they might present significant displacements from the actual terrain. SonicMaps tries to minimise these

[41] http://alastaira.wordpress.com/2011/01/23/the-google-maps-bing-maps-spherical-mercator-projection/
[42] Actual scale factor is 1:100
[43] When connected to Wi-Fi (without GPS enabled), the application might still be able to get our location, but this will result in much less accuracy.
[44] https://developers.google.com/maps/documentation/staticmaps/

issues by offering what I defined as "on-site" editing, that is to say, being able to edit projects from a mobile device, as the user walks and explores the environment.

In order to create a new sound area, users only need to tap the map right at the point where they want it to be located and press the "New Area" button in a resulting window. A grey circular area is then instantiated. If we touch this newly created area, it will turn red (editing mode) and the sound properties windows will be displayed (Figure 10). In this new window the user can set the URL for the corresponding audio file, toggle between 3D/2D listening modes, activate the loop playback, set the volume, etc. The size of the sound area can also be modified with a simple touch-and-drag gesture.

Sounds are triggered whenever the arrow representing the user enters any cylindrical mesh containing an audio file (sound area). Both the user position arrow and the sound area mesh include a Collider component which is used by Unity's physics engine to detect collisions. More precisely, the cylinder's collider is enabled as a trigger, so it is possible to code different sonic behaviours by using the OnTriggerEnter, OnTriggerStay and OnTriggerExit Unity functions.

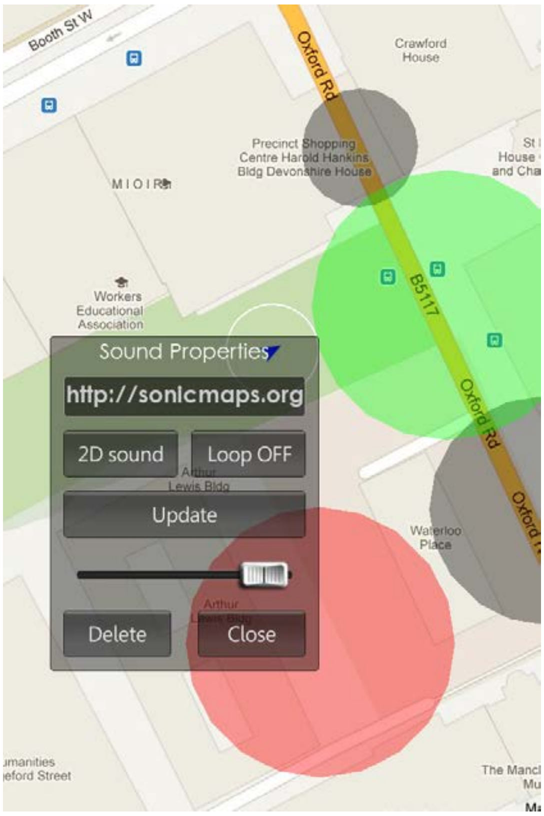


Figure 10. Sound areas editing in SonicMaps Editor

However, actual audio files are not initially stored in the mobile device but online, using a personal server or file hosting service. Therefore, sound areas only contain URLs to online audio files, avoiding the hassle of manually transferring these files into every device before we can play a project. Every project is saved as a JSON text file containing serialised data about the sound areas and their properties[45]. This feature makes content publishing and sharing extremely easy with the help of a dedicated database which is also publicly displayed on the projects page of the PHP-enabled SonicMaps website.

The GPS location services updates the user position every half a second, although transitions are smoothed over time using linear interpolation. The sensor's accuracy is graphically displayed using a white error-range circle around the position arrow. In some devices this accuracy may be as good as 5 metres or less.

45 http://www.json.org/

The solid state compass sensor is called every 50ms in order to get orientation data. The user position arrow rotates to match the new heading and surrounding sounds are panned accordingly. Noise data from the compass sensor due to random movements, e.g. when handling the mobile device as we walk, is filtered so only substantial rotations are visualised.

The Unity game engine uses an implementation of the FMOD[46] audio library for the creation and playback of interactive audio. This library includes an Audio Listener component (virtual microphone) to output spatial audio resulting from any audio source in a scene. SonicMaps attaches the audio listener to the user position arrow so sounds are perceived from this point of view. Likewise, Audio Source components can be found in every sound area the user creates and will be assigned a transform position (Vector3). Therefore, audio signals experiment a customisable amplitude drop-off (volume attenuation) over distance and are panned into the stereo sound field according to their relative angular position.

Three playback modes have been devised as a result of combining a spatial audio mode (3D/2D) with a mono or stereo audio file:

- 3D/mono
- 2D/stereo or mono
- Pseudo-3D/ stereo.

A 2D sound is just a common stereo or mono audio file which is played with no spatial qualities, just as they would be heard from any regular music player.

The 3D property relates to the spatial sound feature we discussed in Chapter 2.2.2. However, all 3D sounds in Unity are forced into mono by default, for easier panning and a more effective spatial localisation. This limitation can be partially corrected by emulating distance attenuation with a simple custom JavaScript to produce a stereo pseudo-3D audio source, with no panning or directional properties, whose volume increases as the user gets closer to the centre of the defined area.

Audio clips can also be looped, which is useful when the user needs to maintain a sustained sonic texture within an area and it is not possible to predict how long will the user stay in that particular location. Loops also help to keep audio file size small, which is strongly recommended when relying on a slow and sometimes expensive 3G internet connection.

Another important issue, especially when working with 2D sounds, is that whenever a listener exits a sound area, the corresponding audio clip might suddenly stop playing and produce

[46] http://www.fmod.org/

undesirable artefacts (clicks, etc.). Nevertheless, a two- to three-second fade-out script is activated on trigger exit, providing a more consistent termination.

Every sound area can be explained as a combination of two separate components: a buffering area and a playback area (Figure 11).

By default, every sound area includes a link (URL) to a sample audio clip[47] that will start downloading whenever the listener enters the buffering area. This buffering area guarantees that the sound is ready to play by the time the actual playback area is reached. The size of the buffering area is proportional to the size of the playback area, and once the download is complete the latter turns green as a notification.

Figure 11. Audio buffering and playback areas in SonicMaps

Any audio file contained in a sound area will automatically start playing any time its playback area is entered. When the listener exits the playback area, if the sound is still playing, it will rapidly fade out and stop playing. The fade-out time is no longer than one to two seconds and it has been implemented to avoid sonic artefacts. An external audio editor can be used if we need longer fade-outs or any fade-in. SonicMaps does not apply any fade-in effect to sounds when entering an area. However, a natural fade-in effect is perceived when using the 3D audio option, since amplitude depends on how close the listener is to the centre of the sound area.

Once all sound areas have been created, the project can be saved and published along with some descriptive metadata into a MySQL database which is accessible for other users via the SonicMaps website.

In summary, the main innovative features of this system are:

[47] http://sonicmaps.org/sample.mp3

- *3D spatial sound*. The SonicMaps Editor was built in Unity making use of its spatial audio engine; once a sound is fixed onto a particular location, its amplitude and stereo distribution (panning) will always be consistent with the position and orientation[48] of the listener. This feature contributes to a more immersive sonic experience and a better integration of sound sources into the scene. An illusion of diegetic sound is thus achieved.
- *On-Site editing using mobile devices*. Unlike browser-based editors, sounds can be geolocated on-site using the same mobile application that playbacks published projects. There are two main advantages to this approach: first, buildings reflections of GPS signal producing inaccurate/displaced location readings can be taken into account when positioning sound areas; and second, the suitability of one or multiple sounds for a specific location, and their exact position, can be better assessed when being on-site, from an immersive perspective.
- *Online publishing and audio files hosting*. Soundwalks can be directly published to and downloaded from the online SonicMaps database using the app. There is no need to manually copy audio files to the mobile device before playing a project; these are hosted online by the user and introduced into a project as public links for progressive on-demand download (3G/4G mode), or full download (Wi-Fi mode).

The original paper discussing the SonicMaps platform can be found in Appendix A.

3.2.3 Sound Materials and Compositional Approach

Similarly to soundscape composition, the sound materials here employed can be categorised somewhere along the continuum between the following two extremes: 'found' sounds, and 'abstract' sounds. The former usually consists in transparent presentations of recorded sound, while the latter generally involves synthesised sound, or deep transformations of existing recordings, so the sound source is unclear or no longer recognisable. In both cases, the multimodal nature of locative audio compositions originates a number of new causal connections between these sounds and their visualised environment, given our tendency to 'source bonding' (Smalley, 1997, p. 110–111). It is the composer's function to predict or anticipate such associations.

The assignment and distribution of sounds over space is mostly done according to a tension-release criterion, in order to ensure that some chosen paths can deliver optimised flow of musical information. It is understood that "optimal" means a sequence of sounds where tension and

[48] Thanks to the magnetometer sensor.

release are alternating as the users/listeners walk along a path. This dynamic balance can be achieved by controlling the amount of information the listener receives at every stage of the soundwalk, and avoiding long periods of excessive or too little information (e.g. consecutive similar sections). According to information theory (Shannon, 1948), this amount of information is directly related to our expectations and the degree of uncertainty involved in a particular event. Consequently, it is advisable, for example, to seek the right balance between 'found' and 'abstract' sounds, as the amount of information, in terms of sound-source uncertainty, is considerably higher in 'abstract' sounds than in easily recognisable 'found' sounds (e.g. birds singing, water flowing, and especially the human voice itself). Adjusting the number of overlapping/overlaying sounds (sound density) is also a good way to control the information flow. As we raise the number of simultaneous sound sources, the amount of information that the listener needs to process, and the associated tension, will increase accordingly. These virtual sound sources are provided as circular playback areas that can be arranged in numerous different ways using the custom mobile application discussed above (Figure 12).

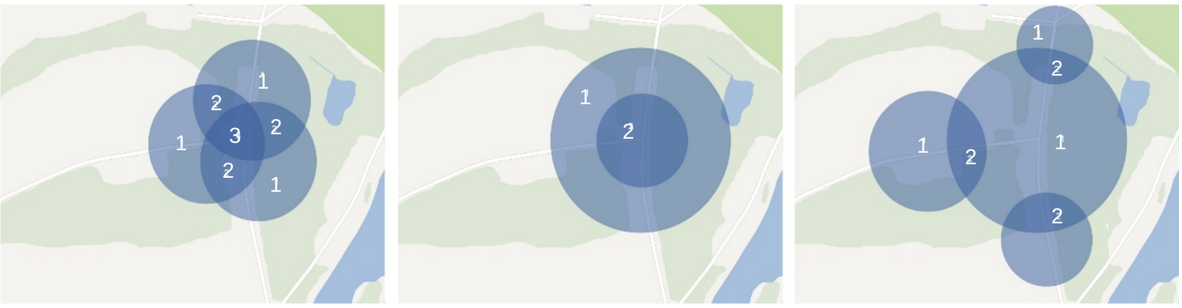


Figure 12. Examples of sound areas overlapping/overlaying (numbers indicate density)–not actual maps.

Additionally, the emerging relationships between situated sounds and visual landscape possess the ability to alter the amount of information that a particular sound provides. Ambiguous sounds with multiple possible explanations (in terms of source identity) might see their uncertainty levels reduced, when a perceived visual element suggests a particular interpretation. This phenomenon can be observed, for example, in the piece *Alice*, when associating processed bell-like spectromorphologies, with cows wearing cowbells and grazing on the grass of Heaton Park; or a notably abstract granular texture, with frogs croaking on the park's lake. Without the visual or contextual aid, the information/uncertainty provided by these sounds would be significantly higher. For that reason, it is suggested that the contextualisation of abstract sound-based music, using locative audio techniques, is a useful compositional resource to increase the accessibility of this type of music to wider audiences, as demonstrated in the 'Locative Audio 2012' event discussed below.

The length of the audio files contained in sound areas is another relevant aspect to be considered, as they should be adjusted to match the size of the corresponding circular area, taking into consideration the expected time that it would normally take to walk through that space. However, it is not always possible to predict the user's behaviour, who might stay longer than expected in a particular sound area; consequently, we might choose to loop our sounds until the user exits the corresponding activation circle.

3.2.4 Alice: Elegy to the Memory of an Unfortunate Lady

Alice was the first piece completed using the SonicMaps Locative Audio tool. The site chosen for this soundwalk is Heaton Park[49] (Manchester, UK); hence, the piece explores the history of the park, its natural landmarks, and a particular popular legend about the ghost of a girl named Alice who is said to wander the rooms of Heaton Hall—a 18th century country house once inhabited by Sir Thomas Egerton and his family. In addition, this piece includes excerpts from the poem *Elegy to the Memory of an Unfortunate Lady*[50] by Alexander Pope (1717), about a young woman who commits suicide because of a thwarted love; therefore, a possible connection between the popular legend and the lady of the poem is here suggested.

As previously indicated, the spatial structure of this work is constrained and provided by the local topography and the landmarks/features of the park (Table 3), including natural elements (e.g. hills, boating lake, water streams, woods), a number of buildings (e.g. Heaton Hall, Town Hall Colonnade, Temple rotunda observatory, stables), and the footpaths which link all previous elements.

1	East entrance	6	Old Tram Station/Museum	11	Stables and Farm Centre
2	Grazing cattle	7	Horse-drawn tram	12	Woods and cross paths
3	Town Hall Colonnade	8	Footpath loop/open fields	13	Dell Gardens and pond
4	Boating lake	9	Heaton Hall (country house)	14	Water stream
5	The woods	10	The Temple (observatory)	15	Play ground

Table 3. Map legend including landmarks/features in Heaton Park

Lower and higher tension areas alternate along the structure, generally using footpaths in open fields as low tension areas, and paths within woodlands as high tension zones (Figure 13), although an exception to this can be found around Heaton Hall, given its historical and narrative relevance.

---

[49] Detailed information about the park is available at http://www.heatonpark.org.uk/HeatonPark/Default.aspx

[50] The poem is recited and recorded by Professor John Richetti (University of Pennsylvania), who kindly authorised its use within this piece. A full public copy of the original poem is available at http://www.poetryfoundation.org/poem/174157

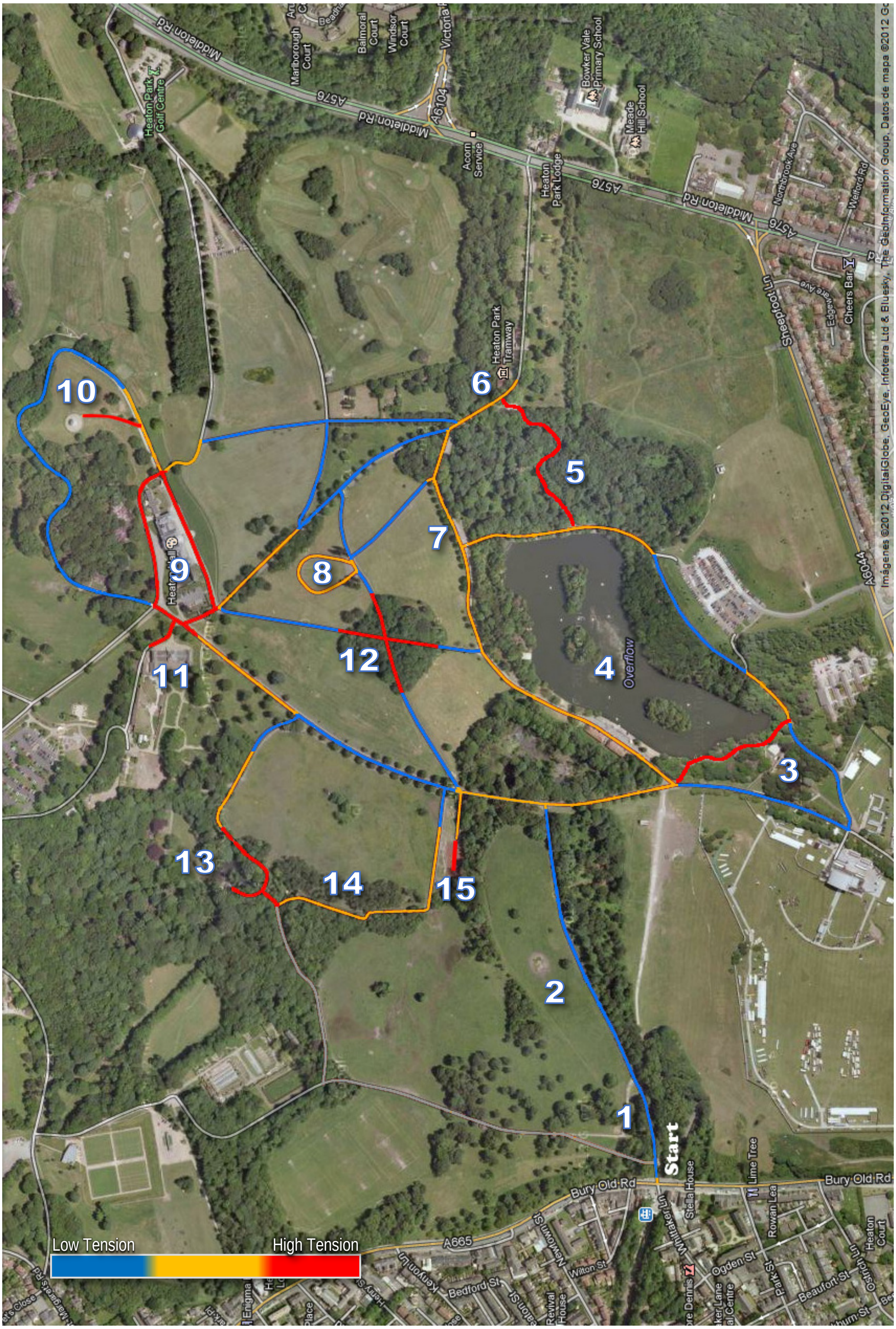


Figure 13. Footpaths, landmarks and tension levels in Alice

Unlike *Swirls*, there are no strategies to limit the duration of the piece, and no goals in terms of linear progression (e.g. the final soundscapes at the end of every path in *Swirls*); users are free to roam around the park as long as they want. The listening mode is no longer performative but explorative in nature.

Given the open-form navigation model, the selected excerpts from the poem (usually just a few verses) will not normally follow the linear narrative sequence of the original text; therefore, the subject of the poem is only slightly suggested, as pieces of a puzzle that need to be put together with the help of the rest of sounds.

An extension to this work, exploring the possibility of connecting a soundwalk with a concert hall experience (Appendix A), was presented at the *Interactive Audiogame Showcase* organised by NOVARS Research Centre on 29 June 2012. This version was now based on Whitworth Park (Manchester, UK) and implemented an OSC module into SonicMaps in order to send live user location data to a server in the concert hall, where a virtual representation of the park was being projected on a large screen. This not only allowed the audience in the concert hall to experience the sounds of the park as they were being triggered by the person doing the real soundwalk, but also introduced a new virtual agent —represented on the screen as a white sphere of light and controlled by a performer— that would be perceived by the distant user as an invisible mobile sound source wandering around the park, intending to be 'Alice' (i.e. the ghost) (Figure 14).

Figure 14. Connecting real and virtual space in Alice

3.2.5 Adaptive Locative Audio (Dynamic Content Server)

Given that we are relying on the cross-modal interaction between aural and visual information, and the establishment of meaningful relationships between our virtual sound-sources and existing physical locations, this relationships might be affected by any significant changes in the visual landscape–the most obvious and frequent variation being the transition between day and night.

For that reason, I decided to develop a Dynamic Content Server (DCS), consisting in a 'Unix Cron Job' (automated scheduled service)[51] that periodically reads incoming system time and date and calculates when the sun is due to set or rise at a specific location. Subsequently, it runs a custom PHP script including instructions to replace some of the active audio files in a SonicMaps project with the appropriate daytime or night-time version. This solution can indeed be regarded as an 'adaptive locative audio' system.

Sound variations are stored in separate folders (Figure 15) which can be accessed by the PHP script in order to pick the appropriate version and copy it into the project's active folder, so it can be used by the SonicMaps app.

This new approach using local time or date, also introduces the idea of 'sound-event geolocation' (vs simple 'sound geolocation'), suggesting that now sounds are not just located in a spatial dimension, but also in a temporal one. Therefore, beyond the delivery of scheduled variations of a sound, we might also decide to make a single sound (no variations) only available at specific times or days.

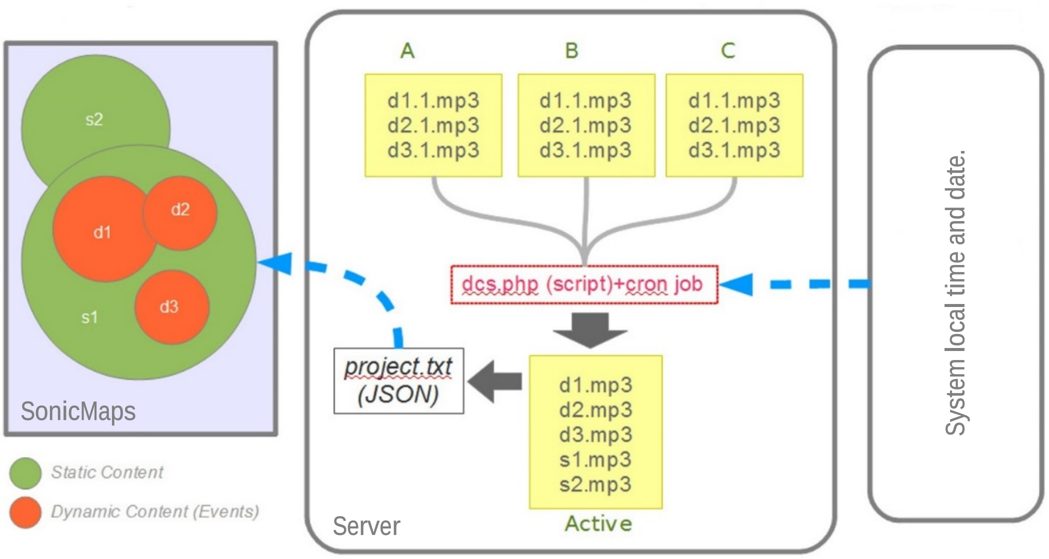


Figure 15. Dynamic Content Server for Adaptive Locative Audio

[51] A description of the Cron job scheduler is available at http://en.wikipedia.org/wiki/Cron

3.2.6 Alcazabilla

The following work was composed in response to the Locative Audio 2013 Valencia+Network Festival[52], an international large-scale collaborative project curated by Professor Ricardo Climent and sponsored by SALC's Research Network Fund and The University of Manchester. This event involved 12 different cities[53] where simultaneous soundwalks were taking place around a common theme (sonic chronologies).

Alcazabilla was commissioned for the Málaga soundwalk[54] (Spain), which I was also responsible to coordinate as an event manager and technical advisor. The piece is named after one of the most iconic and touristic streets of the city of Malaga —Calle Alcazabilla— where the soundwalk is situated.

It is inspired by the history, landmarks, and traditions of this fascinating place, and especially, on a poem by Adriana Schlittler (Table 4), presenting her personal literary vision of the place: "ideas of erosion, time and contemplation", which reflect the more than 2000 years elapsed since this part of the city was first inhabited.

Original text[55] (Spanish)	Approximate English translation
Me quedaré aquí *en estas ruinas de teatro* *en esta piedra* *Me quedaré aquí porque* *padezco sus surcos y me hieren sus grietas* *Aparezco inerte ante tanto derrumbarse* *¿Acaso es el tiempo un ensayo para la mutilación?* *Me quedaré aquí* *hasta que el cuerpo sea escombro* *y el vientre a la carroña* *una ofrenda*	*I will stay here* *in these theatre ruins* *on this stone* *I'll stay here because* *I suffer its grooves and cracks which hurt me* *I look inert in front of so much collapsing* *Is time a trial for mutilation?* *I will stay here* *until the body is rubble* *and my belly* *an offering for carrion*

Table 4. 'Me quedaré aquí'; a poem by Adriana Schlittler–the potential unifying element.

[52] Please visit http://acusmatica.7host.com/locative/2013/indexval.html for further details.
[53] Valencia (Spain), Virginia (USA), Manchester (UK), Málaga (Spain), Avignon (France), Linz (Austria), Grenoble (France), Volos (Greece), Murcia (Spain), and Tampere (Findland).
[54] Please see Appendix C for full size brochure of the event including soundwalk map. *Alcazabilla* is number 16 on the map.
[55] Reproduced with explicit permission of the author.

Therefore, the original sound materials which I recorded for this piece are in part explained with regard to the major landmarks found in Calle Alcazabilla (see centre image in Figure 16), but also by the use of spoken word[56] (poem excerpts) and a number of granular erosion/collapse sounds (sand, stones, rocks) related to the poem's content. These latter sounds are present throughout the piece, unifying the different sections as a coherent whole.

Calle Alcazabilla is approximately 150m long and the prominent landmarks considered for the piece are:

- *Casa Hermandad Cofradía de los Estudiantes (1)*. This building holds religious processional thrones carrying images of Christ and other Biblical characters. Thundering sounds of traditional snare drums are played on the streets during religious processions (Semana Santa). The piece exhibits multiple modified version of this sound using spectral delays, filters and reverse tape effects. Although the *Alcazabilla* takes place in Málaga, the original source recordings for these sounds were taken during Seville's Semana Santa in 2010.
- *Roman garum factory (2)* (under pavement and visible through crystal pyramid). Garum was a fermented fish sauce used as a condiment in the cuisines of ancient Greece, Rome, and Byzantium. Sea related sounds are suggested in the piece for this area. These are original recordings taken in "La Malagueta"–Malaga's most popular beach.
- *Picasso Museum* (3). Holds a collection of art by Malaga born Pablo Picasso. The artist is mainly known for his paintings, although his sculptural works are equally significant. This is evoked in the piece through sounds of rocks being scratched, crushed stones, etc.
- *El Pimpi (4)*. Traditional restaurant and winery serving Málaga wines. A place where popular flamenco artists used to play for decades. Processed original recordings of flamenco tap dancers hitting the floor are employed as a characteristic percussive material[57].
- *Centro de Interpretación del Museo Romano (5)*. Small building exhibiting the history and heritage of Roman "Malaca" (Málaga in Latin language). A Roman amphitheatre (section B) dominates Calle Alcazabilla and inspires the poem by Adriana Schlittler, which is located here.
- *Alcazaba de Málaga (6)*. The Alcazaba is a palatial fortification built by the Hammudid dynasty in the 11th century. Placed on a hill beyond the Roman archaeological site, it is

[56] Special thanks to Marta Strojek, Silvia Guerrero, and Danny Saul for reciting the poem.
[57] Recorded with permission of "Tablao Flamenco Ana María" in Marbella (Spain) during a live night flamenco session.

only suggested in the piece by a distant Adhan prayer in Arabic that overlaps with other materials[58].

The piece is organised into four separate sections (A1, A2, A3, A4), plus a voice-only section including the original poem at the Roman amphitheatre (B). Each section uses sound materials informed by the nearest building or archaeological remains, as described above. Transitions between sounds and sections are implemented using progressive crossfades, as a result of the overlapping of sound areas (Figure 14—right) and the spatial implementation of sound (i.e. 3D audio engine with signal amplitude attenuation over distance).

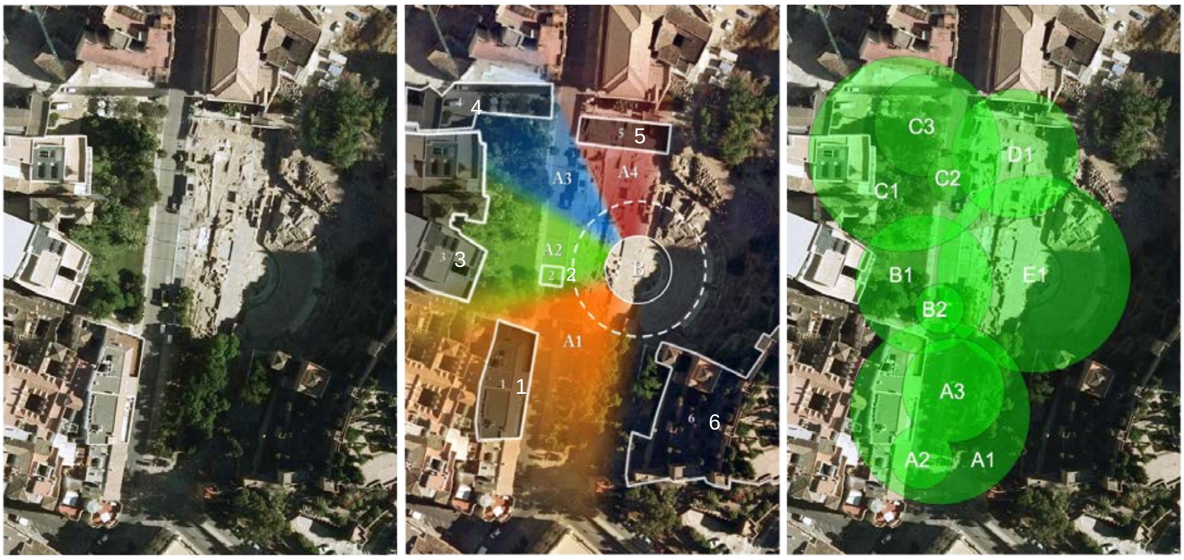


Figure 16. Calle Alcazabilla (left); landmarks and main sections (centre); actual sound areas in SonicMaps (right).

Moreover, the original poem in Spanish language was translated and recorded (by male and female voices) in several different languages (i.e. English and Polish), with the intention of highlighting the multi-cultural heritage and current touristic development of the city of Málaga; but most importantly, as a resource to increase the variety of timbres through phonetic differentiation. It also increases the accessibility of the message of the poem to audiences with no knowledge of the Spanish language and it was used to articulate a vocal counterpoint between different versions of the text (i.e. Spanish vs English or Polish, female vs male).

Finally, 'day' and 'night' variations of some of the sounds proposed in the piece have been introduced using the Dynamic Content Server as discussed above. In general, these are background textures with variable energy or activity levels, so as to modify the overall atmosphere and character of the piece depending on light conditions (e.g. sounds of sea waves).

[58] An excerpt of the Adhan in Masjid al Haram (prayer) is reproduced with permission of iwebaudio.net

3.2.7 Final Comments

The new features introduced with the SonicMaps platform, and more specifically the SonicMaps Editor application, has taken the concept of locative audio one step further in terms of immersiveness, flexibility and accessibility for wider audiences. The 2013 Locative Audio Malaga event (see Appendix C) is an example of this, although it also proved the limits of the current technology. For instance, the number of audio zones included in one single project depends on the size of the associated audio files and their location, so the number of simultaneous active areas (those whose corresponding audio files have been downloaded and loaded into memory) cannot exceed the device's available memory[59]. Even if audio files are downloaded as MP3 files, they are decompressed by Unity's audio engine before playback, increasing their size considerably. This limitation led to occasional crashes on low-end devices and the need to restart the application.

A second issue resulting from this experience was the need to instruct the audience on how to use their devices in order to perceive the correct stereo image of sound. Given that the magnetometer sensor is used to read the device's orientation, the smartphone/tablet needs to be hold horizontally and aligned with our head to perceive the right sound panning. Morover, the device cannot be set in background mode[60] (screen disabled) for SonicMaps to work properly, so battery drain can be an issue with larger projects, ultimately limiting the length of any potential soundwalk[61]. This issue was addressed during the previously mentioned event by carrying multiple external battery packs with USB connections that any participant could use in case of a low battery warning.

Despite the benefits of on-site editing using the SonicMaps Editor application, some users have expressed their desire to have an alternative web-based editing tool that could be used to create a soundwalk using their computer at home, especially when weather conditions do not facilitate using a mobile device outdoors. This option could indeed complement the existing procedure as a way to create the basic layout of sounds of a project, which might later be fine-adjusted using the current on-site mobile editor.

From an aesthetic point of view, when planning a soundwalk and playing its sounds using the SonicMaps application, it is important to consider the type of headphones to be employed, as we can find closed models (those which block or cancel external noise/sound) and open models (those with no isolation capability). This is fundamentally a question about what the external

[59] Around 1GB or less in most mobile devices at the time of writing.
[60] This is a of the Unity game engine.
[61] On average, a fully charged device will run SonicMaps over one hour before the battery is completely drained.

environmental sounds are with respect to our musical work: unwanted noise or predictable/usable materials that can be integrated into the piece. In urban locations with a disturbing omnipresence of traffic noise I generally opt for closed headphones, but other environments such as the park where *Alice* takes place certainly call for open headphones that will allow the sounds of nature (e.g bids, water streams) to be perceived along the pre-composed materials included in the piece.

Finally, in regards to the structure of the piece and how this structure is initially perceived and navigated by the user, I have observed that the visualisation of sound areas on the screen works as a first guide for the user to take decisions on where to go and what to expect during their walk. This behaviour worked well in most situations as it doesn't reveal the content of the sound areas but it provides a reference to stay within the limits of the piece. However, in other scenarios it was sometimes useful to provide an additional printed map indicating the most relevant locations or a preferable path that a group of people should walk together as a part of a scheduled event (see Appendix C as an example of this).

4 Non-interactive Procedural Composition

The second research strand of this portfolio brings back an entirely virtual environment as demonstrated in *Swirls*, although the interactive practice is now consciously abandoned in order to deliver a new form of compositional discourse where, similarly to fix-media works, the composer retakes control of most of the creative and interpretative processes that determine the ultimate outcome of a piece —as discussed in Chapter 2.1.4 and 2.3 of this thesis.

This new approach is again focused on the expressive possibilities arising from the use of space and motion in an audio-visual electroacoustic music composition—especially in terms of embodied sonic gestures and multimodal perception—along with the dynamic generation of audio assets, and the procedural implementation of flexible music structures. The latter will be referred to as 'Procedural Composition'.

In this context, audio and video are created simultaneously as a result of the real-time manipulation of 3D objects in the simulation environment; therefore, the level of synchrony and causal connection between what we can see and hear are particularly remarkable, although the quality and quantity of auditory information is generally intended to exceed the visual one (sonic-centric works) .

In order to achieve these goals, the next three pieces—*Singularity, Apollonian Gasket,* and *Boids*—respectively employ three different physical and/or mathematical models that despite their non-musical nature, can be turned into novel virtual music instruments for the articulation of physical and musical gesture via custom instrumental techniques.

The creative workflow involved is also divided in three separate stages—and ideally formalised in an equal number of scripts:[62] first, the model implementation and its sonification; second, the exploration of functional control methods; finally, the application of the previous control methods into a coherent sequence of instructions (score). These stages are not by any means limited to this musical practice, but common in many other contexts, as we are just making a music instrument, learning how to play it, and finally writing music to be performed with this instrument (Figure 17).

[62] A literal separation between the three creative stages as individual scripts is only performed in *Boids*. *Singularity,* for example*,* combines implementation algorithms and control methods in one single script.

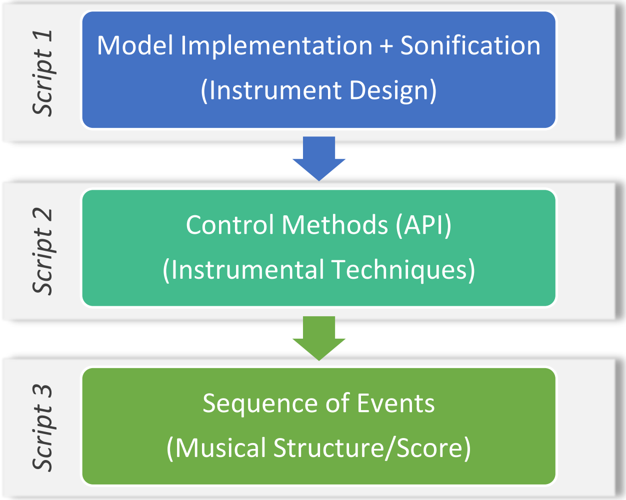


Figure 17. Creative workflow in Singularity, Apollonian Gasket, and Boids

4.1 Recursive and Emergent Systems as Non-conventional 3D Virtual Instruments: Spatial/Kinematic Models for Procedural Sound

The first stage involves the implementation and sonification of the chosen model as an active gestural agent, using the game engine. For that purpose, material components (virtual 3D objects) need first to be provided, along with the algorithms containing the rules that govern their behaviour within the instrumental structure. These algorithms are implemented as a first JavaScript or C# script, exposing a number of characteristic variables (public variables) that can be used by other scripts (or methods within that script) to control the instrument in a further stage.

Please notice that the 3D objects that constitute these instruments do not always need to be manually arranged in advance (e.g. as it happens with the rings of the Apollonian Gasket), but they can also be provided as 'prefabs'[63], which are later instantiated (i.e. cloned and activated) at run-time by the model's defining algorithm; for example, the cubes in *Singularity* or the spherical particles in *Boids*.

In particular, the three models here exposed are (Figure 18):

1) A cellular automaton based on Conway's *Game of Life*[64].
2) A fractal set of circles known as *Apollonian Gasket*[65].

---

[63] Prefabs are ready-to-use assets that are not initially in the scene but in a repository. For further information about Unity prefabs please visit http://docs.unity3d.com/Manual/Prefabs.html.
[64] Extensive information about Game of Life including more than 766 known patterns is available at http://www.conwaylife.com/wiki/Conway%27s_Game_of_Life (Accessed 09 April 2015)
[65] http://mathworld.wolfram.com/ApollonianGasket.html

3) An artificial life, flocking algorithm using an extended version of Craig Reynolds' *Boids* rules.[66]

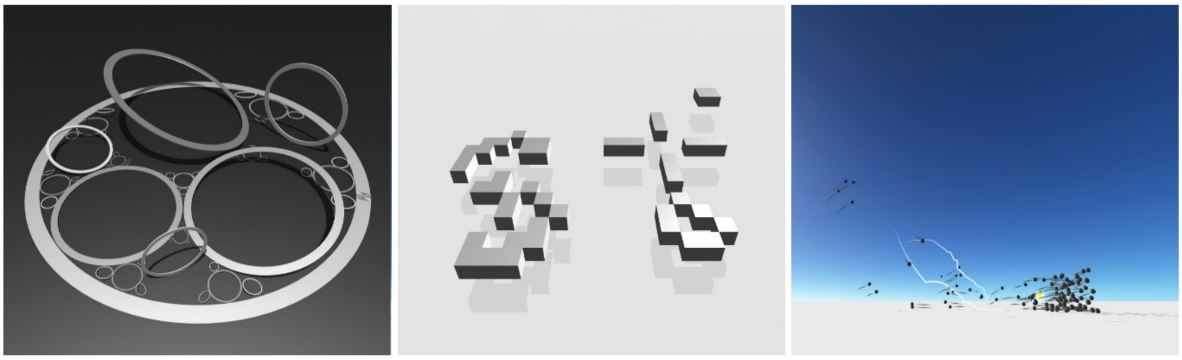


Figure 18. Non-conventional virtual 3D instruments in action: Apollonian Gasket (left), cellular automaton (centre), and flocking particles (right)

The second and third models make use of physically-informed virtual objects as gestural agents of sound; therefore, the position and movement of these objects will determine the musical output (micro and macro structures), and most of the properties of sound (e.g. timbre, pitch, amplitude, spatialisation).

In the first model, however, the perception of movement is just an illusion resulting from the activation/deactivation of the cells that make up the automaton.

In all cases, audio is procedurally generated, in real time, in response to specific changes or events taking place in the implemented model's structure.

Moreover, all models were selected based on meeting the following criteria[67]:

- ✓ Complex behaviour can be obtained from simple rules, either as a recursive or emergent process.
- ✓ The model enjoys spatial and kinematic features which are inherently pronounced and maximise the exploration of gesture and the surrounding 3D environment (spatialisation).
- ✓ Minimal visual representation prioritises aural perception.

The specific rules and features of each of the selected models are shown in Table 5.

[66] A comprehensive repository of online resources is available at http://www.red3d.com/cwr/boids/
[67] The reasons for these constraints, in terms of audio-visual features, accessibility, etc., were discussed in more detail in Chapter 2.1.3

	Complexity via Emergence		Complexity via Recursion
Base Model	**Cellular Automaton**	**Flocking Behaviour**	**Fractal Structure (Apollonian Gasket)**
Piece Title	*Singularity*	*Boids*	*Apollonian Gasket*
Rules	- Any live cell with fewer than two live neighbours dies (under population). - Any live cell with two or three live neighbours will live on to the next generation. - Any live cell with more than three live neighbours dies (overcrowding). - Any dead cell with three live neighbours becomes a live cell (reproduction).	- Separation: steer to avoid crowding local flock mates. - Alignment: steer towards the average heading of local flock mates. - Cohesion: steer to move toward the average position (centre of mass) of local flock mates. - Avoidance: steer to avoid predators. - Location: stay within the stated limits (e.g. 3d box).	- Recursively generated from triples of circles where each circle is tangent to the other two. 3D rings implementation: - Rotation - Precesion - Collision
Features	- Dissemination of cells over a 2-dimensional space. - Desirable musical characteristics: growth, decrease, chaos-order alternation, etc. - Excellent motif generator A source of algorithmic micro spatialisation. - Capable of Spatial Concatenative Synthesis.	- Swarm Intelligence Behaviour (AI) - Extensive exploration of 3D space thanks to flock's kinematics. - Controllable balance between order and chaos with multiple possible states or configurations. - Sound texture might emerge as a result of summing individuals (Additive Synthesis)	- Fractal Geometry. - Quasi-symmetric distribution of circles. - A 3D version is possible turning circle into rings, resembling the kinematic behaviour of a spinning coin or Euler's Disk. - The precession rate of the disk's axis of symmetry accelerates as the disk spins down (finite-time singularity).

Table 5. Rules and features of three spatial/kinematic models

4.1.1 Singularity — Cellular Automata

This work is titled after the Singularity hypothesis—the emergence of artificial intelligence because of technological exponential growth—as the automata used in the piece are a good example of emergent self-organising system. Compositionally speaking, it is also inspired by the 17th and 18th centuries British Industrial Revolution, using audio recordings of textile manufacturing and steam powered machines, taken at MOSI[68], as sound source materials.

[68] Museum of Science and Industry, Manchester, UK.

Several large arrays of very short samples (~ 10-30 ms) have been manually extracted from these recordings, so whenever any of the cells of an automaton becomes active, a random member of the corresponding array is selected for playback (using Unity's internal audio engine) (Figure 19).

Additional sound materials, with a similar sonic character, are also employed in the middle section of the piece (section B in Figure 20). These were obtained from vocal recordings of stop/plosive consonants (e.g. 'p', 't', 'k') in Spanish and Japanese language[69].

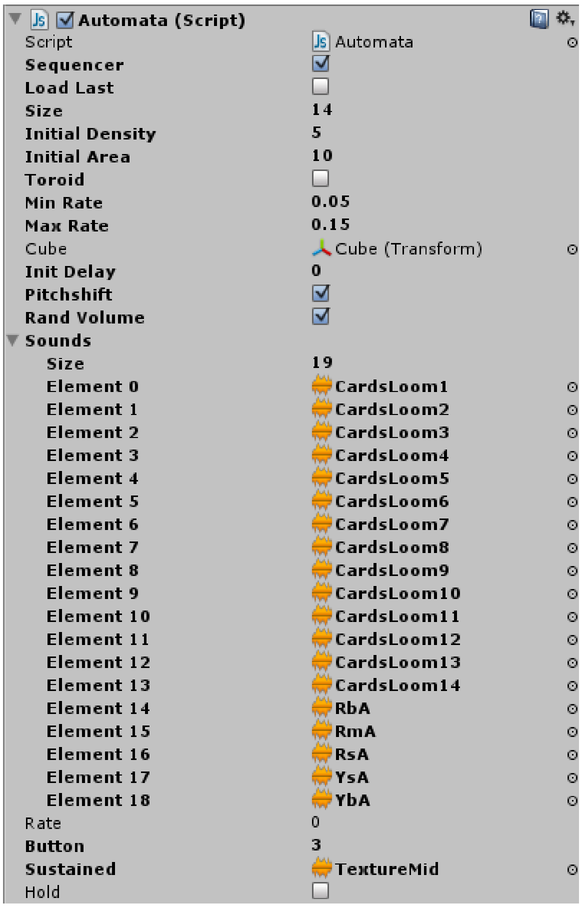


Figure 19. Example of an automaton's inspector window in Unity Editor (model implementation script)

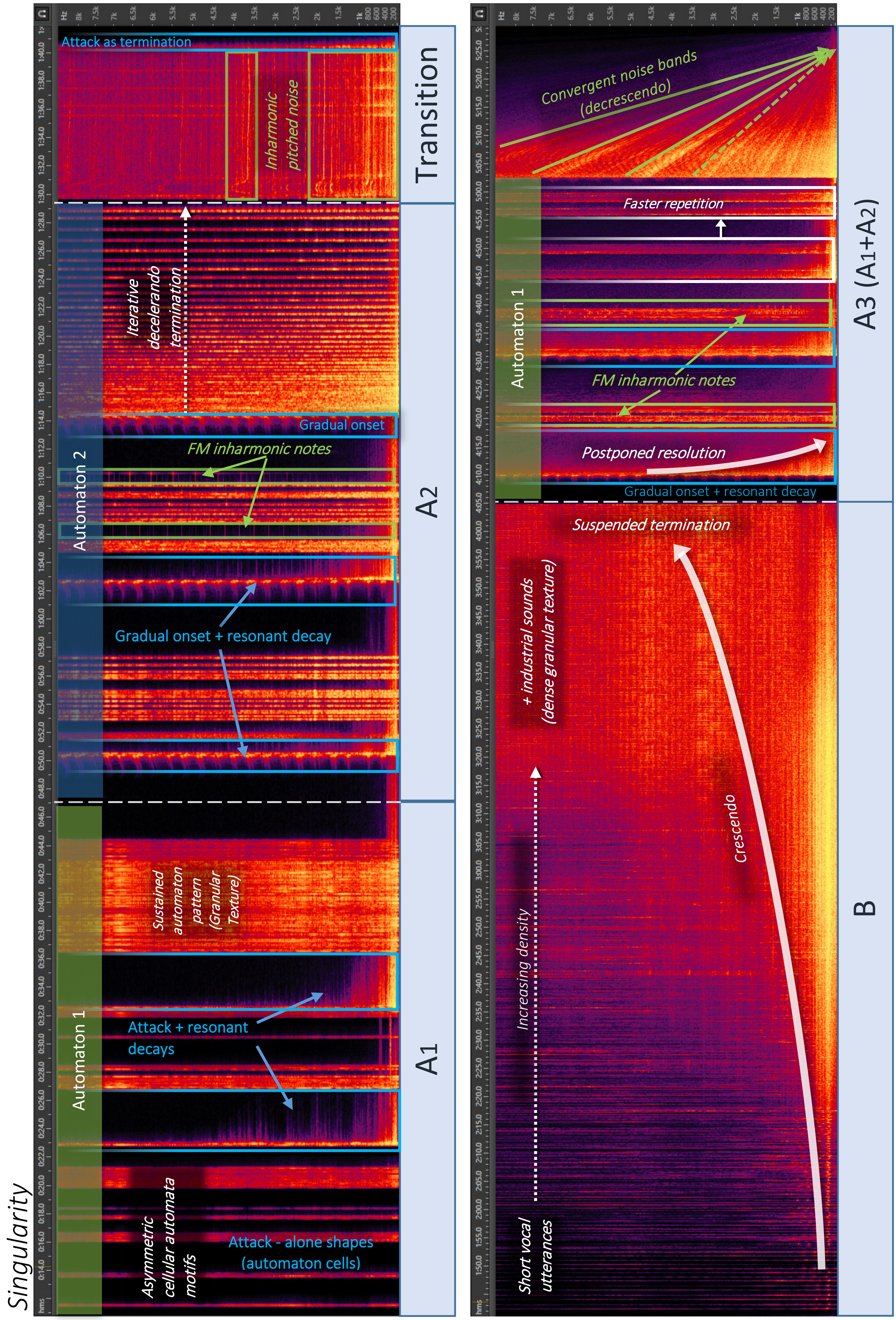
All previous micro-samples are accompanied by longer processed sounds of piano effects (e.g. string damping metallic effect) which accentuate the onset of new automata after transitory silences.

The characteristic growth and dissemination of the automaton's cells over the 2-dimensional space will produce a series of rhythmic and spatial motifs, which can be controlled through the automata's initial configuration and evolution rate—as their behaviour is complex but deterministic. This rate does not have a continuous fixed value, but it randomly oscillates between a minimum and maximum value, in order to introduce enough rhythmic variation.

We are effectively using the cellular automata as a generator of musical patterns (microstructure) that can be later used to compose the macrostructure of the piece (see 4.2). When the automata's evolution rate (discrete temporal steps) is fast enough, a concatenative synthesis effect might also be perceived.

[69] I would like to thank Silvia Guerrero and Haruka Hirayama for their collaboration in this recordings.

Figure 20. Singularity - Spectral Analysis



Apollonian Gasket — Fractal Structure

The geometry of the Apollonian Gasket is presented in this piece as an extension of the original bi-dimensional fractal object (Figure 21), where circles have been substituted by three-dimensional rings, in order to enable its kinematic behaviour.

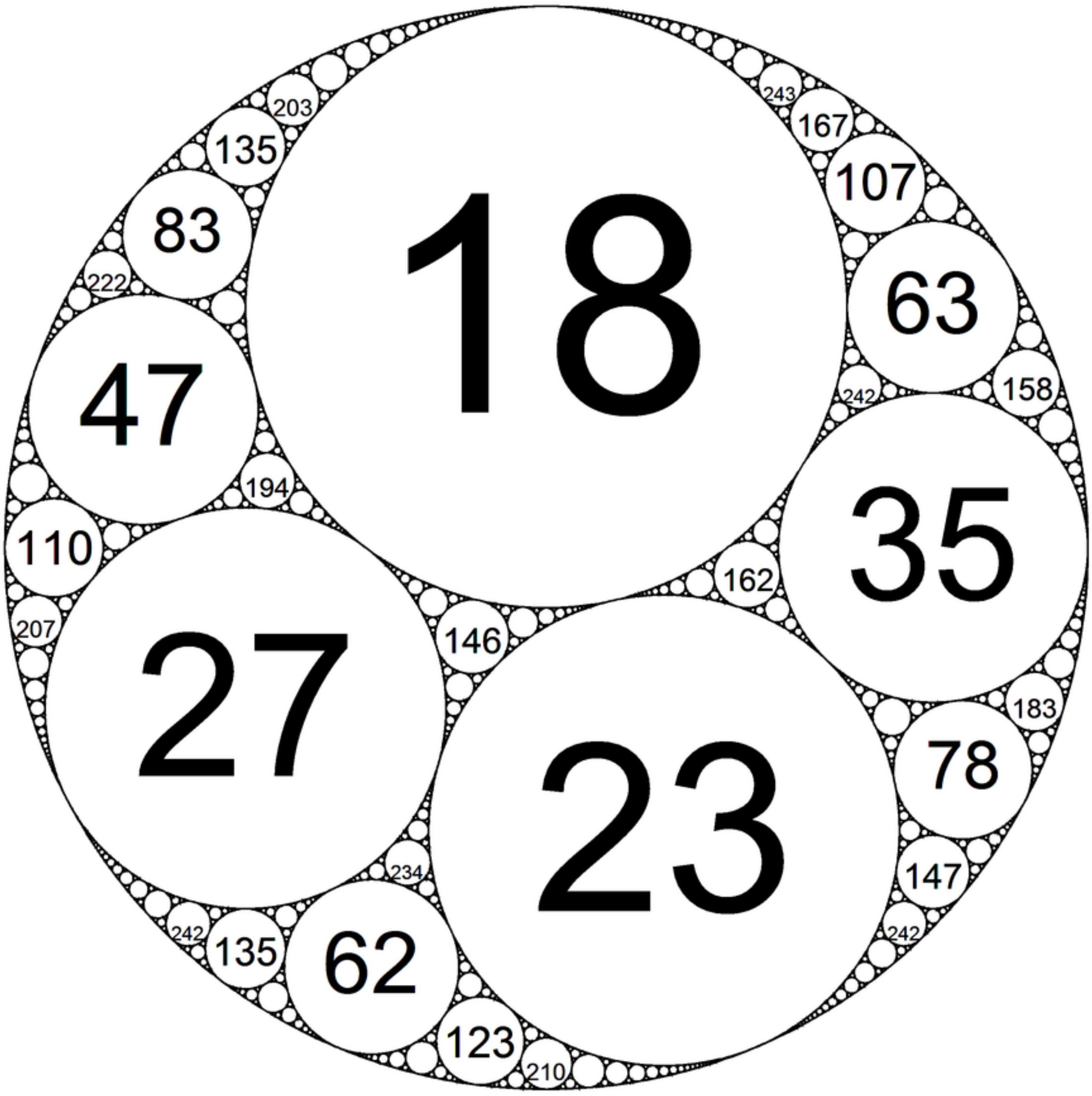


Figure 21. The original Apollonian Gasket; circles curvatures are indicated.

In particular, three distinct types of motion or events can be observed (Figure 22):

- *Rotation*. Circular movement around the ring's main symmetry axis.
- *Precesion*. Change in the orientation of the rotational axis around a vertical axis (secondary rotation). A connection between rotation and precesion speed might exist.
- *Collision*. A small percussive element is sometimes attached to the ring, so it will hit the floor at intervals producing a resonant collision sound as a result of the rotation-precesion movement.

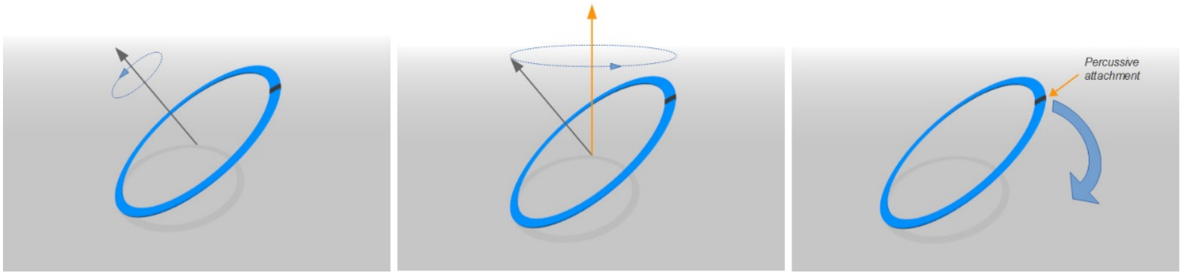


Figure 22. Apollonian Gasket kinematics: rotation (left); precesion (centre); collision (right);

A number of synth definitions have then been created in Supercollider to provide the corresponding sounds for the previous gestures, including FM/AM synthesis, physical modelling synthesis, noise generators or BPF/LPF filters. These are originally simple sounds (e.g. sine oscillators) where amplitude is usually modulated according to the rings' rotation and precesion rate; however, their frequency spectrum will progressively become more complex (via additional frequency or phase modulation) as the energy of the associated rings is increased. Some of the input variables used to feed these sound generators are shown in Figure 23.

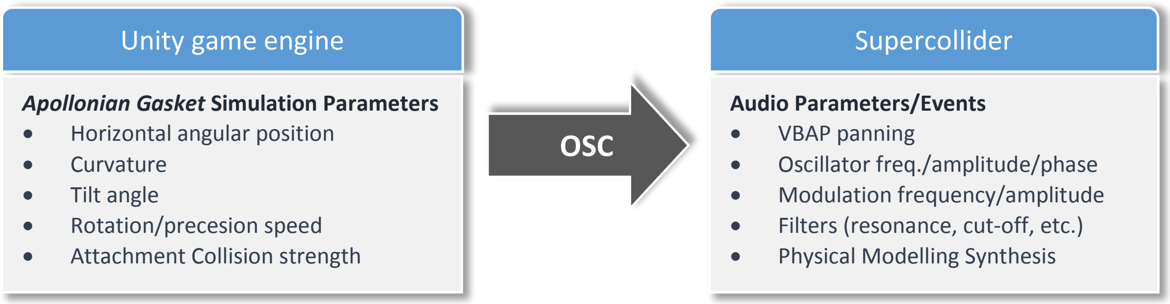


Figure 23. Apollonian Gasket data mapping (sonification)

Given that ring curvatures are being used as a variable to feed sound parameters, an asymmetric geometry, given by the initial values 18, 23 and 27 (see Figure 21), has been selected in order to avoid excessive regularity and repetition.

Furthermore, this particular fractal distribution provides a quasi-symmetric layout with three main groups of rings placed on the front-left, front-right and rear-centre of the scene (Figure 24). The audio listener is placed at the approximate centre of the set (ring 146), providing an optimal listening perspective that is also useful when mapping the multichannel output to a sound diffusion system such as MANTIS[70].

[70] http://mantis-novars.blogspot.co.uk/

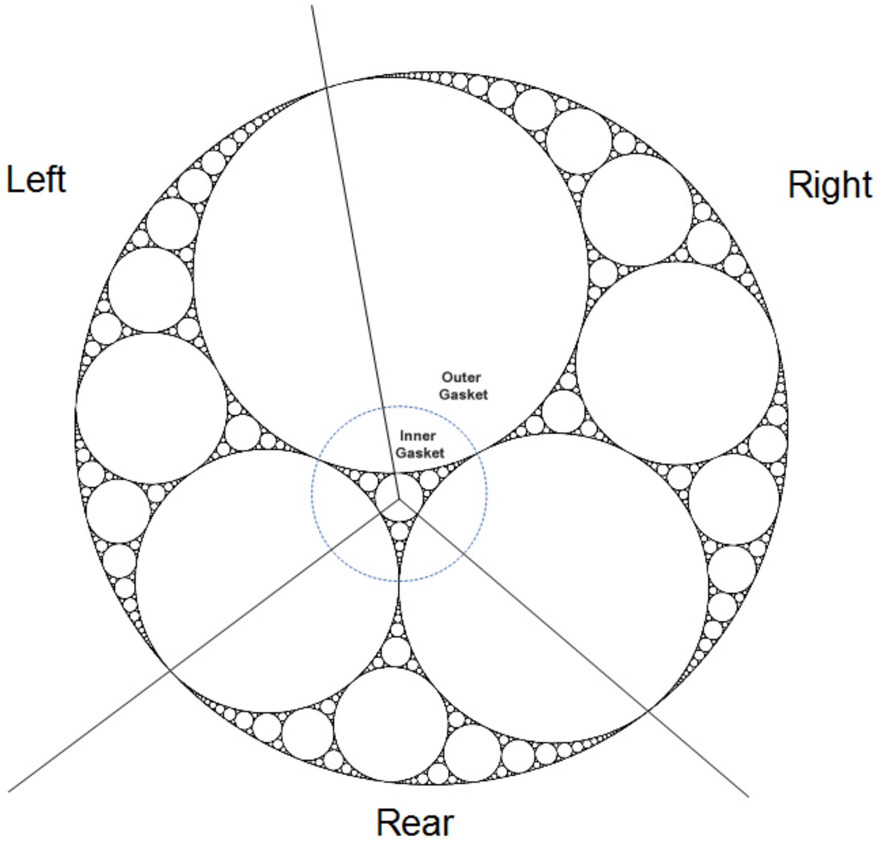


Figure 24. Symmetry/grouping and spatialisation in Apollonian Gasket

To some extent, the final outcome is considered to be a virtualised instance of sound sculpture intermedia.[71]

The structure of the piece is represented in terms of its spectromorphological features in Figure 25.

4.1.2 Boids — Flocking Behaviour

The last piece of this portfolio is an attempt to combine and consolidate the most innovative methods and ideas that were revealed in the previous two works; for example, the algorithmic implementation of a complex system with emergent properties—as seen in Singularity's cellular automata—and the procedural generation of sound using kinematic data and synthesis definitions in an external audio engine—as demonstrated in Apollonian Gasket.

For that purpose *Boids* starts from an artificial life flocking algorithm of the same name, popularised by Craig Reynolds in 1986, although its basic rules—as stated in Table 5—are here extended to introduce some boundaries in the space where flocking agents can move (e.g. height, radius), and the figure of the 'predator', as a hostile object that needs to be avoided.

[71] An art form which is also characterised by the use of tri-dimensional space and the organisation of matter in order to produce sound (Grayson, 1975).

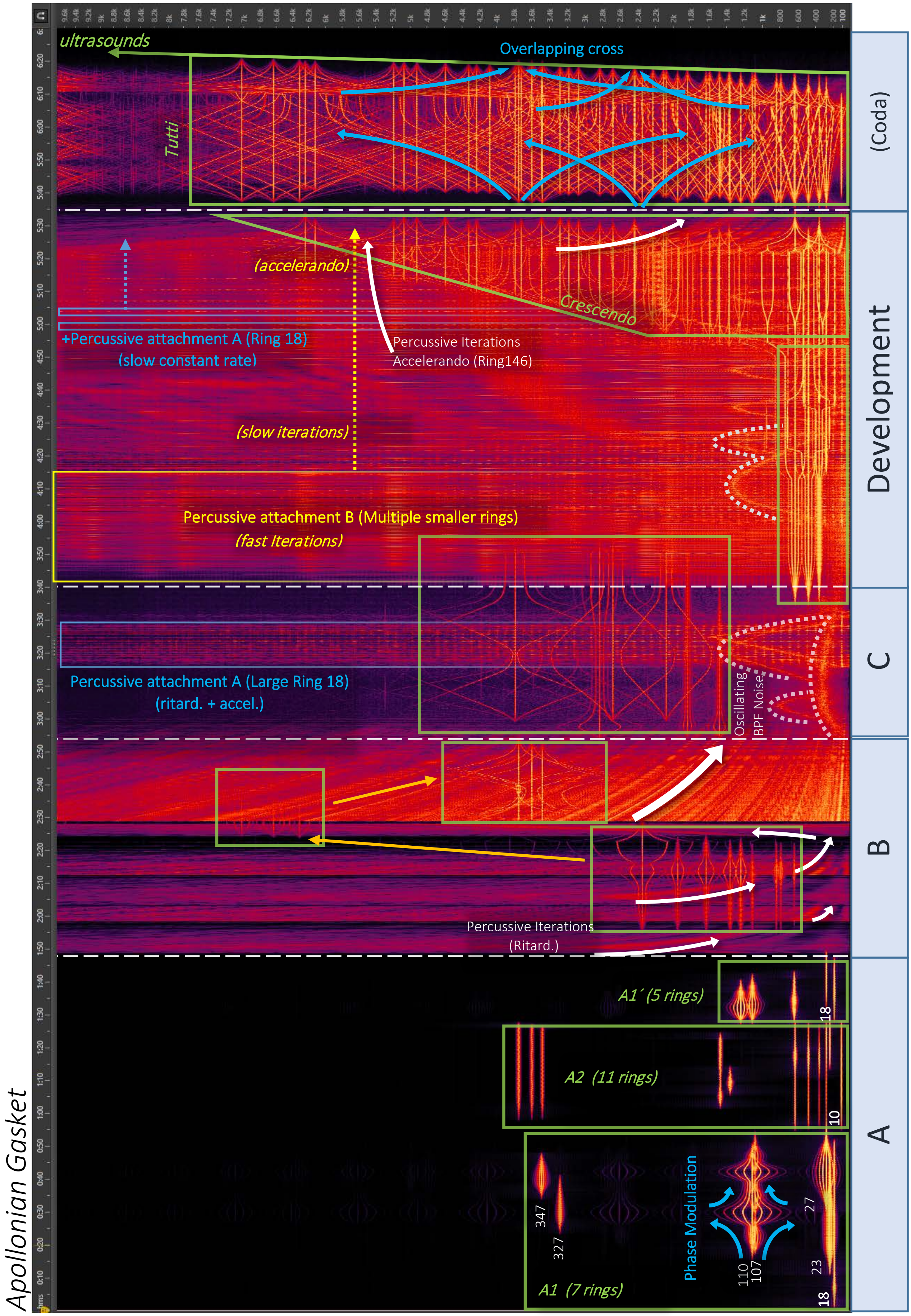


Figure 25. Apollonian Gasket - Spectral Analysis

Again, sounds are enabled by mapping multiple simulation variables into a number of audio parameters in the external audio engine (Figure 26).

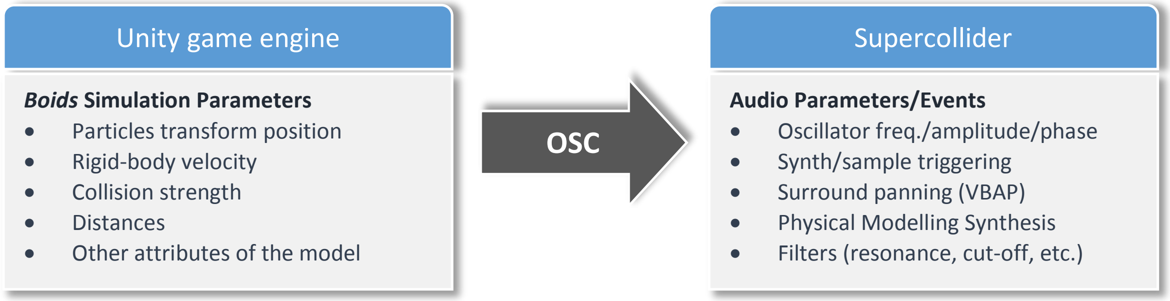


Figure 26. Boids data mapping (sonification)

The current version includes 100 individual particles (agents); every particle is assigned an instance of a sound generator (synth definition) in Supercollider (Figure 27), which is continuously updated using the particle's kinematic data.

```
(
SynthDef(\simple, {|out=0, freq=1, azi= 0, ele=0, spr=10, gate= 1, cutoff=0,revSend=0.18, effectBus=56,
neighbors=0.01, bVol=1|
    var exciter1, exciter2, sig,noise, panned, env,gen;

    env = Env.asr(5,1,10,-2);
    gen = EnvGen.kr(env, gate, doneAction: 2);

    //component1 (SAW)
    exciter1 = LFSaw.ar(Lag.kr(freq, 0.2),0,(0.03+(cutoff/8000))*bVol);

    //component2 (NOISE)
    noise = WhiteNoise.ar(Lag.kr(cutoff*cutoff/500 + cutoff/500,0.1));
    exciter2 = BPF.ar(noise,Lag.kr(100+freq,0.2)/2, 0.01+Lag.kr(2*neighbors/(1+(cutoff*cutoff)), 0.2));

    //Total Signal
    sig=RLPF.ar((exciter1+exciter2)*gen,Lag.kr(cutoff,0.2)*100+50,1,1);

    //Surround 8.0 Output
    panned = VBAP.ar(8,sig, d.bufnum, azi, ele, spr);
    [0, 1, 2, 3, 4, 5, 6, 7].do({arg bus, i; Out.ar(bus, panned[i])});

    //FX Send Output
    Out.ar(effectBus, sig*revSend);

}).add;
)
```

Figure 27. Synth Definition of flocking particles in Supercollider

A sawtooth wave oscillator and a white noise generator are at the core of the main synth definition, providing enough partials for a rich and bright sound. Since frequency is mapped to the height of the particles, any vertical movement will be perceived as a glissando effect. In addition, the audio signal is sent to a low-pass filter, whose cut-off frequency is linked to the particle's velocity, so the sound becomes brighter as the particle moves faster, and vice versa. Surround spatialisation is again automatically performed by the VBAP object using angular positions, and a reverb bus has been added to smooth and homogenise the final texture.

As a result, a rich and dynamic additive synthesis effect is achieved, and the nature and complexity of the emergent sonic texture (spectral density and distribution) is thus ultimately representative of the flock's state and internal configuration.

Nevertheless, not all sounds are generated via synthesis; for example, the crackling sounds of predators chasing particles are otherwise produced using audio recordings of different sized tree branches being split (Figure 28 and 27). These are very short samples, similar to those presented for the cells of the cellular automata in *Singularity*, and equally arranged into arrays for random playback.

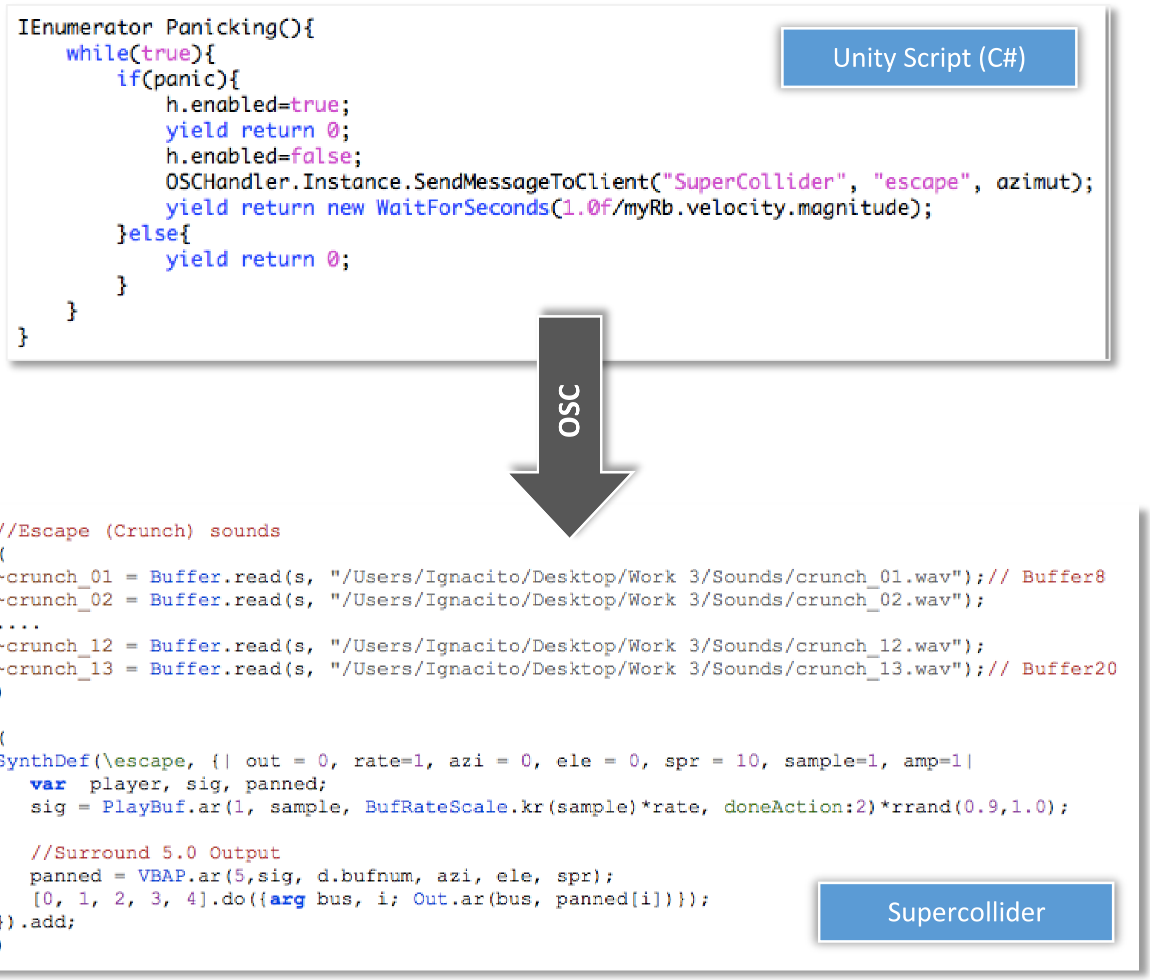


Figure 28. Predators sound implementation (partial code)

Figure 29. Boids recording session (predator sounds)

A similar sound is produced when any of the particles collide against the ground. This sound has a percussive wood-like quality which was obtained by striking pairs of those same tree branches segments. Further variation and transformation is achieved by pitch shifting these audio samples according to the strength of the collisions against the ground.

The granular nature of the texture emerging from these dissected audio recordings contrasts with the sustained audio texture of the synthesised flock movement, providing a good source for sonic articulation.

On the other hand, the flocking behaviour is eventually stopped during the piece, giving way to a geometrical distribution of the agents in space (e.g. parallel circles, helices), which is markedly contrasting in terms of sound harmonicity, spectrum stability, etc. (Figure 30)

Figure 30. Flocking behaviour (left) and geometric distribution (right) in Boids

In this regard, the overall macrostructure of the piece is partially explained as an oscillating balance between chaos and order, harmonicity and inharmonicity, symmetry and irregularity. This is apparent when analysing the spectrogram of an audio recording of the piece, as shown in Figure 31.

* *Only the fundamental frequencies of the sawtooth oscillators are considered for analysis purposes.*

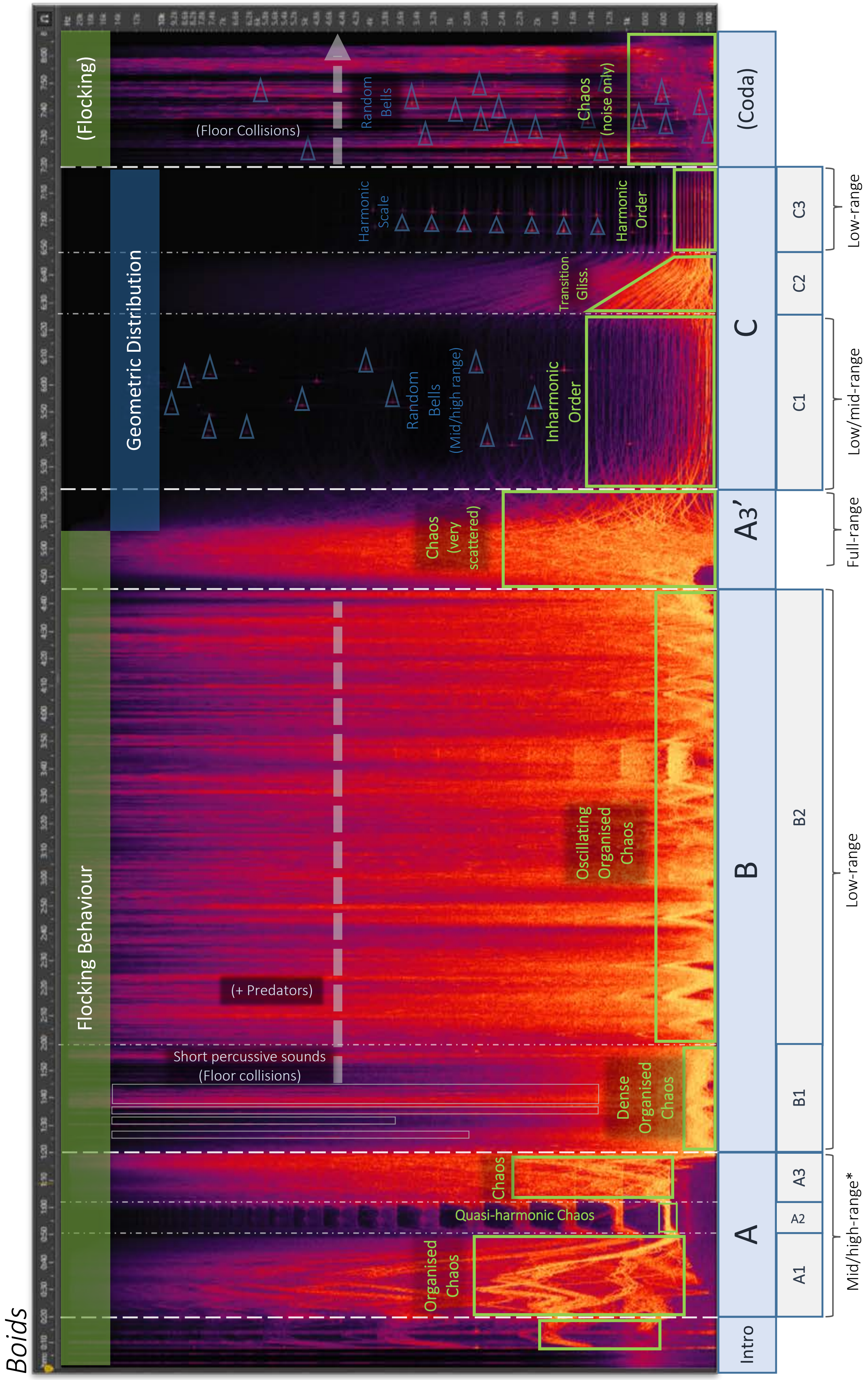


Figure 31. Boids - Spectral analysis.

4.1.3 Instrumental Techniques (Control Methods/API)

After completing the 3D implementation and sonification of the instrumental model, the expressive potential of the resulting instrument is explored, and formalised as a set of control methods or API (Application Programming Interface). From a traditional point of view, each method functions as an instrumental technique, and has a direct impact on the properties and articulation of sound, as it has access to the public variables that characterise the instrumental model.

Once a control method has been defined, it can be called from the main events sequence (generally a third script) and reused as many times as needed, avoiding excessive code duplication.

For example, the following code snippet shows a method to set the energy of a spinning ring in *Apollonian Gasket* (Figure 32).

```
//Change Ring Precesion Energy over time
IEnumerator SetEnergy (GameObject ring, float targetEnergy , float seconds){
    float t = 0.0F;
    var precesion = ring.GetComponent<Precesion>();
    float currentEnergy = precesion.energy;
    while (t <= 1.0F) {
        t += Time.deltaTime/seconds;
        precesion.energy = Mathf.Lerp(currentEnergy,targetEnergy, Mathf.Lerp(0.0F, 1.0F, t));
        yield return 0;
    }
}
```

Figure 32. Control method example (SetEnergy)

It is also possible to write a method to produce an accelerando or ritardando effect in *Singularity*'s automata (Figure 33).

```

function ChangeRate(targetMin:float,targetMax:float,seconds:float){
    var t = 0.0;
    var currMin= minRate;
    var currMax= maxRate;
    while (t <= 1.0) {
        t += Time.deltaTime/seconds;
        minRate = Mathf.Lerp(currMin,targetMin, Mathf.Lerp(0.0, 1.0, t));
        maxRate = Mathf.Lerp(currMax,targetMax, Mathf.Lerp(0.0, 1.0, t));
        yield;
    }
}

```

Figure 33. Control method example 2 (ChangeRate)

We can observe how both previous functions take three input parameters, and they can be later used by respectively calling:

```
StartCoroutine (SetEnergy (ring23, 1.3f, 5.0f));
```

or,

```
a.ChangeRate (0.08, 0.3, 10);
```

The first example (in C# programming language) will progressively increase or decrease (depending on current values) the energy of ring23 over a five-second period until the target value (1.3) is reached. In the second example (in Unity JavaScript) the minimum and maximum progression rates of automaton 'a'—which should previously be referenced—will be similarly modified over a 10-second period.

4.2 Procedural Thinking and Musical Structure: Sequencing via Scripting

The compositional strategies above illustrated how musical events and complex processes that determine different aspects of the evolution of a piece can be encapsulated into well-defined procedure-based descriptions using a scripting programming language. This is a form of control abstraction that allow us to submerge the details of music production and focus on higher level manipulations (Greenberg, 1987).

Therefore, the musical macrostructure of these works can be described as a temporal sequence of procedures (Figure 34), which also works as a musical score for its interpretation by the game engine (using JIT compilation[72]). Every time we need to reuse a particular function/method, we just need to provide its name and any customisation parameters, so the final sequence remains considerably short and neat, facilitating the visualisation of the emerging structure and the compositional activity.

As previously discussed (2.1.4), time can be managed using coroutines; for example(C#):

```
yield return new WaitForSeconds (3.0f);  // Pause the execution of this thread during 3 seconds
```

Or introducing some indeterminacy:

```
yield return new WaitForSeconds (Random.Range (1.0f, 5.0f)); // Pause for 1 to 5 seconds
```

But it is also possible to subordinate the execution of one method to the completion of an earlier method. An example of this can be observed in lines 195-198 of Figure 34, where section C (from line 205 onward) will only start once all predators in Section B have extinguished, which, incidentally, is not a fixed duration process, but depends on random circumstances as a result of the flocking behaviour.

[72] For details about Just-In-Time compilation in Unity please visit http://docs.unity3d.com/412/Documentation/ScriptReference/index.Script_compilation_28Advanced29.html

```
//SECTION B ********

//Cluttered Flock
StartCoroutine(SmoothBoxBottom (0.5f,5.0f)); //Decrease Box Bottom
StartCoroutine(SmoothBoxTop (30.0f,5.0f)); //Change Box Top
StartCoroutine(SmoothRandVel(1.0f,3.0f)); //Decrease RandVel
boids.SetBoxArea(50.0f); str="50";
yield return new WaitForSeconds (1.0f);

StartCoroutine(SmoothDrag (1.5f,2.5f,10.0f)); //Change Drag
yield return new WaitForSeconds (5.0f);

StartCoroutine(SmoothBoxBottom (1.5f,5.0f));
StartCoroutine (SmoothTrailTime (0.3f, 6.0f)); //Reduce trails length
yield return new WaitForSeconds (6.0f);

//Predator1
inception.BigBang(1);
boids.SetPredator ();
yield return new WaitForSeconds (4.3f);

bubbles=GameObject.FindGameObjectsWithTag("bubble1");
yield return 0;

bubbles[0].GetComponent<Explode>().FreePredator();
yield return 0;

predators = GameObject.FindGameObjectsWithTag("Predator");
yield return 0;
StartPredator (0);
yield return new WaitForSeconds (20.0f);

//Predator2
inception.BigBang(1);
boids.SetPredator ();
yield return new WaitForSeconds (2.3f);

bubbles=GameObject.FindGameObjectsWithTag("bubble1");
yield return 0;

bubbles[0].GetComponent<Explode>().FreePredator();
yield return 0;

predators = GameObject.FindGameObjectsWithTag("Predator");
yield return 0;
StartCoroutine(Harakiri (0));
StartCoroutine(Harakiri (1));
foreach(GameObject predator in predators){
    predator.GetComponent<Predator>().mates =
    GameObject.FindGameObjectsWithTag("Predator");
}
StartPredator (1);

//Flock increases energy/area slightly
StartCoroutine(SmoothBoxArea(80.0f,5.0f));
StartCoroutine(SmoothDrag (0.1f,0.5f,5.0f));
StartCoroutine(SmoothRandVel(1.1f,3.0f));
StartCoroutine(SmoothBoxBottom (1.0f,10.0f));
StartCoroutine(SmoothBoxTop (50.0f,5.0f));
yield return new WaitForSeconds (30.1f);

//Wait until all predators die
while (deadCount!=2) {
    yield return 0;
}
yield return new WaitForSeconds (3.0f);

//SECTION C ********

//Flock transition towards geometry
//Move camera further
StartCoroutine(SmoothCamera(new Vector3 (50f, 5.0f, -300.0f), 20.0f));
print ("Starting Transition 1 (5s)");
StartCoroutine(SmoothBoxArea (3.0f,5.0f));
StartCoroutine(SmoothDrag (2.1f,2.5f,15.0f));
StartCoroutine(SmoothBoxBottom (25.0f,10.0f));
StartCoroutine(SmoothBoxTop (250.0f,1.0f));
StartCoroutine(SmoothRandVel(3.0f,15.0f));
yield return new WaitForSeconds (5.0f);
print ("Setting BoxArea to 150 over 3s and wait 7s more");
StartCoroutine(SmoothBoxArea (150.0f,3.0f));
StartCoroutine (SmoothTrailTime (4.0f, 10.0f));//Increase trails
yield return new WaitForSeconds (10.0f);

//GEOMETRY SPIRAL
print ("Starting Geometry (1.5s)");
StartCoroutine(SmoothBoxBottom (2.0f,5.0f));
boids.SetDrag(1.0f,1.5f); str4 = "1"; str5 = "1.5";
yield return 0;
geoState=true;
freezeState=true;
yield return 0;
boids.FixGeometry(true);
boids.SetShape("helix");
helixRadius="15";
boids.SetRotRadius (float.Parse(helixRadius));
boids.SetRotSpeed(0.2f); str7="0.2";
geoHeight=0.14f;
yield return 0;
StartCoroutine(SmoothRandVel(0.02f,20.0f));
StartCoroutine(SmoothDrag (3.0f,3.0f,6.0f));
boids.SetSpeedLimit (30.0f);
yield return new WaitForSeconds (1.5f);
```

Figure 34. Example of Procedural Composition (C# sequence excerpt from Boids)

Consequently, time between procedures is not necessarily fixed, but takes into account random elements and the complexity of the models employed, hence the dynamic, flexible form and duration of these works.

It should be mentioned that the last three procedural compositions of this portfolio have also been implemented as interactive works, by providing a custom human user-interface with direct access to the instruments' control methods. For that purpose, the procedural score and computer interpreter were simply omitted. An example of this is an online implementation of the sonic cellular automata used in *Singularity*, where users can trigger random patterns or introduce the initial configuration of the automaton by activating the corresponding cells in a web-browser 'grid' interface[73]. A later and more complex interactive installation, presenting the three virtual instruments here discussed, will also be exhibited at NIME 2015[74] (LSU Glassell Gallery) under the title *XYZ*. These interactive presentations highlight the separation between the original musical instrument and the actual composition based on this instrument, as discussed in Chapter 2.2.1.

[73] Available at http://ignaciopecino.com/automata.html

[74] New Interfaces for Musical Expression Conference, Baton Rouge Louisiana, 31 May - 28 June 2015.

5 Conclusion

In brief, the first part of this portfolio has illustrated how making use of game engine technologies, virtual and augmented space can be employed as a compositional canvas where sound sources are organised into an effective musical structure. The resulting sonic landscape was physically and sonically navigated by indoors (concert hall interactivity) and outdoors (locative audio) audiences, thanks to bespoke interfaces that highlighted the roles of embodied motion and gesture in our multimodal perception of sound. The active audio-visual exploration of sound materials, increased the levels of engagement of the participants with the author's compositional language, which was ultimately more accessible. This engagement was also reinforced in *Swirls* by adopting a number of stylistic and gameplay elements from popular video games culture.

On the other hand, the second part of the portfolio presented three procedural compositions based on spatial and kinematic models implemented as non-conventional 3D virtual instruments with a minimal visual content. It can be observed how these final pieces present some similarities with fixed-media/acousmatic composition, not just because of the abstract sonic language employed, but because of the linearity of the proposed macrostructure and the absence of a human interpreter—in contrast with Locative Audio compositions and *Swirls*. However, the dynamic real-time generation of sounds, and the flexible, live interpretation of a procedural score (using a computer), are also typical features of live musical practices. Therefore, this intended duality has been helpful to address the initial research questions by combining the best of both worlds in aid of a more expressive non-interactive compositional practice. Furthermore, an important research outcome has been the formalisation of the different creative stages involved in procedural composition (as executable programming code instructions or scripts) which clearly separate instruments, methods, and musical score.

The systems implemented in this portfolio of compositions exhibited a sonic and musical output that was effectively dynamic in nature, either by virtue of the use of interactive open-form structures, or thanks to the live execution of generative algorithms and kinematic models for procedural audio. In all cases, space and motion were the fundamental structural parameters that mediated the production of sound, which resulted in a particular audio-visual discourse that takes into consideration the importance of bodily gestures in our experiencing of music.

Future work is intended to address two distinct aspects of this portfolio. First, the spatial gap between a frontally situated 2D screen and those sounds coming from side or rear speakers is usually overcome by our perceptual/cognitive capacity to associate simultaneous visual and sound events (synchresis effect). However, a more immersive visual presentation of the latest procedural pieces, using virtual reality technologies such as the Oculus Rift[75] headset, should help to improve the correlation between perceived visual and auditory information, which is currently compromised by the use of a non-immersive conventional screen, especially when rendering multichannel surround sound. A similar principle applies to the rendering of 3D spatial sound, currently implemented using Unity’s audio engine or the VBAP object in Supercollider. It is suggested that the localisation of virtual sound sources could be enhanced using alternative methods such as binaural 3D audio synthesis[76] or Ambisonics. Finally, the distribution of the works is currently non-trivial when the execution of an external audio engine (e.g. SuperCollider) is required. In that regard, new solutions to integrate all components of the system into a single easy-to-use executable, are to be explored.

[75] https://www.oculus.com/

[76] For example, the 3Dception Audio Engine. Information available at http://twobigears.com/

Glossary of Terms

Some of these terms have been borrowed from game-audio theory, where they are largely accepted, although many of them are also common in other disciplines. Moreover, a same term might also be found elsewhere in conjunction with the words "audio/sound", or "music/composition", depending on the structural level considered.

Adaptive Music

This is a popular term in the game audio industry, denoting a musical system that is able to adapt to specific changes in contextual parameters of the game (e.g. transitions between day and night), by providing new sounds or music that is appropriate to the new context. A particular case exists when music adapts to the listener and his real-world environment using built-in sensors in mobile devices (camera, microphone, accelerometer, GPS, etc.); this is what has recently been referred to as Reactive Music[77].

Dynamic Audio

In the context of this research, dynamic audio refers to sounds and music which are not completely linear and/or fixed, as in fixed-media compositions, but they possess the potential to change along multiple successive iterations.

Generative Music

This term describes any music that has been generated using an autonomous not-human system, and is ever-different and changing. According to its most restrictive definition no input is required, so it can be regarded as non-interactive, although it is sometimes accepted that the initial conditions of the system are set before the execution starts (Farnell, 2007, p. 3).

Open Form Composition

A musical work where the different component sections are not arranged following a predetermined linear sequence, but they can be performed or navigated in any arbitrary order.

Procedural Audio

The term "procedural audio" can be defined as "non-linear, often synthetic sound, created in real time according to a set of programmatic rules and live input"(Farnell, 2007, p. 1)— as opposed to recorded sound. Procedural audio reduces the need for large amounts of disk-stored audio data, but it demands greater CPU processing power for its operation. It is a relatively recent term, and it is focused on content generation, particularly, "when there is too much content to create, when we need variations of the same asset, and when the asset changes depending on the game

[77] A well-known advocate of reactive music is app developer RjDj (Reality Jockey Ltd.). Please visit http://rjdj.me/ for further information.

context" (Fournel, 2012). Hence, it "emphasises the interplay between inputs, algorithms, and the expressivity and correctness of the outputs" (Compton et al., 2013). In other words, procedural audio possesses a similar role to adaptive music, but at a sonic level.

Script (programming)

"A scripting language or script language is a programming language that supports 'scripts', programs written for a special run-time environment that can interpret (rather than compile) and automate the execution of tasks that could alternatively be executed one-by-one by a human operator". (Anon, 2015)

Bibliography

Andreassi, J. L. and Greco, J. R. (2013) ‘Effects of bisensory stimulation on reaction time and the evoked cortical potential’, *Physiological Psychology*, 3(2), pp. 189–194.

Anon (2015) ‘Scripting language’, *Wikipedia,* [online] Available from: http://en.wikipedia.org/w/index.php?title=Scripting_language&oldid=654965736 (Accessed 7 April 2015).

Arnold, D. (1959) ‘The Significance of “Cori Spezzati”’, *Music & Letters*, 40(1), pp. 4–14.

Behrendt, F. (2012) ‘The sound of locative media’, *Convergence: The International Journal of Research into New Media Technologies*, 18(3), pp. 283–295.

Boulez, P. and Cage, J. (1995) *The Boulez Cage Correspondence*, Nattiez, J.-J. (ed.), Cambridge University Press.

Chion, M. (1994) *Audio-vision: Sound on Screen*, Columbia University Press.

Climent, R. (2009) *Ho - A Sonic Expedition to Vietnam*, [online] Available from: http://acusmatica.7host.com/electroacoustic/rik_06.html (Accessed 8 April 2015).

Climent, R. (n.d.) ‘Locative Audio’, [online] Available from: http://locativeaudio.org/ (Accessed 25 March 2015).

Climent, R. (2014) *‘Putney’ for game-audio*, [online] Available from: https://vimeo.com/109376169 (Accessed 8 April 2015).

Collins, K. (2009) ‘An Introduction to Procedural Music in Video Games’, Contemporary Music Review, 28(1), pp. 5–15.

Compton, K., Osborn, J. C. and Mateas, M. (2013) ‘Generative methods’, In *The Fourth Procedural Content Generation in Games workshop, PCG*, [online] Available from: http://www.fdg2013.org/program/workshops/papers/PCG2013/pcg2013_6.pdf (Accessed 20 April 2015).

d’ Escriván, J. (2011) *Terra Nova, for orchestra, laptop ensemble and computer game player*, written for the Anglia Sinfonia, Cambridge.

d’ Escriván, J. (2014) ‘Videogames as digital audiovisual performance’, in *xCoaAx 2014 Proceedings of the Second Conference on Computation Communication Aesthetics and X*, Porto, Portugal, pp. 248-269, ISBN 9789897460364.

Dolphin, A. (2011) ‘A Portfolio of Original Compositions’, SARC, Queen’s University, Belfast.

Dorigo, M. and Birattari, M. (2007) ‘Swarm intelligence’, *Scholarpedia*, 2(9), p. 1462.

Ecker, A. J. and Heller, L. M. (2005) ‘Auditory - visual interactions in the perception of a ball’s path’, *Perception*, 34(1), pp. 59–75.

Encyclopaedia Britannica (2015) ‘space | physics and metaphysics’, *Encyclopaedia Britannica*, [online] Available from: http://www.britannica.com/EBchecked/topic/557313/space (Accessed 26 February 2015).

Farnell, A. (2007) ‘An introduction to procedural audio and its application in computer games’, In *Audio Mostly Conference*, pp. 1–31, [online] Available from: http://www.cs.au.dk/~dsound/DigitalAudio.dir/Papers/proceduralAudio.pdf (Accessed 20 April 2015).

Fellgett, P. (1975) ‘Ambisonics.Part one: general system description’, *Studio Sound*, 17(5), pp. 20–22, 40.

Fournel, N. (2012) ‘Procedural Audio: An Interview with Nicolas Fournel | Designing Sound’, [online] Available from: http://designingsound.org/2012/06/procedural-audio-an-interview-with-nicolas-fournel/ (Accessed 20 April 2015).

Grayson, J. (1975) *Sound sculpture: a collection of essays by artists surveying the techniques, applications, and future directions of sound sculpture*, A.R.C. Publications.

Greenberg, G. (1987) ‘Procedural Composition’, in *International Computer Music Conference Proceedings*, Illinois, USA, Computer Music Association, pp. 25–32, [online] Available from: http://quod.lib.umich.edu/cgi/p/pod/dod-idx/procedural-composition.pdf?c=icmc;idno=bbp2372.1987.004.

Hamilton, R. (2014) ‘Perceptually Coherent Mapping Schemata for Virtual Space and Musical Method’, Stanford University, [online] Available from: http://www.researchgate.net/profile/Robert_Hamilton13/publication/262805044_Perce ptually_Coherent_Mapping_Schemata_for_Virtual_Space_and_Musical_Method/links/543c23eb0cf204cab1db88a3.pdf (Accessed 10 February 2015).

Harley, M. A. (1994) ‘Space and Spatialization in Contemporary Music: History and Analysis, Ideas and Implementations’, Montreal, McGill University, [online] Available from: http://www.moonrisepress.com/dissertation.html (Accessed 2 March 2015).

Hecht, D., Reiner, M. and Halevy, G. (2005) ‘Multi-Modal Stimulation, Response Time, and Presence’, *PRESENCE 2005*, p. 269.

Johnson, M. L. and Larson, S. (2003) ‘“ Something in the Way She Moves”–Metaphors of Musical Motion’, *Metaphor and symbol*, 18(2), pp. 63–84.

Jordà, S. (2005) ‘Digital Lutherie Crafting musical computers for new musics' performance and improvisation’, MTG, Universitat Pompeu Fabra, Barcelona, [online] Available from: http://mtg.upf.edu/node/449 (Accessed 18 July 2015)

Landy, L. (2007) *Understanding the Art of Sound Organization*, Cambridge, Mass, MIT Press.

Milgram, P., Takemura, H., Utsumi, A. and Kishino, F. (1995) ‘Augmented reality: A class of displays on the reality-virtuality continuum’, In *Photonics for industrial applications*, International Society for Optics and Photonics, pp. 282–292, [online] Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981543 (Accessed 1 March 2015).

Mooney, J. R. (2006) ‘Sound diffusion systems for the live performance of electroacoustic music’, University of Sheffield, Department of Music, [online] Available from: http://www.james-mooney.co.uk/research/sound_diffusion/JM_Sound_Diffusion_Systems.pdf (Accessed 10 March 2015).

Otondo, F. (2008) ‘Contemporary trends in the use of space in electroacoustic music’, *Organised Sound*, 13(01), [online] Available from: http://www.journals.cambridge.org/abstract_S1355771808000095 (Accessed 11 February 2015).

Parry, N. (2014) ‘Navigating Sound’, In Collins, K., Kapralos, B., and Tessler, H. (eds.), *The Oxford Handbook of Interactive Audio*, Oxford University Press, [online] Available from: http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199797226.001.0001/oxfordhb-9780199797226-e-002 (Accessed 1 March 2015).

Paterson, N. and Conway, F. (2014) ‘Engagement, Immersion and Presence’, In Collins, K., Kapralos, B., and Tessler, H. (eds.), *The Oxford Handbook of Interactive Audio*, Oxford University Press, [online] Available from: http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199797226.001.0001/oxfordhb-9780199797226-e-016 (Accessed 1 April 2015).

Pecino, I. (2014) ‘Spatial and Kinematic models for Procedural Sound in 3D Virtual Environments’, In *Proceedings of the International Computer Music Conference*, Athens, Greece, Michigan Publishing, pp. 836–843, [online] Available from: http://hdl.handle.net/2027/spo.bbp2372.2014.131 (Accessed 9 February 2015).

Pecino, I. and Climent, R. (2013) ‘SonicMaps: Connecting the Ritual of the Concert Hall with a Locative Audio Urban Experience’, In *Proceedings of the International Computer Music Conference*, Perth, Australia, Michigan Publishing, pp. 315–320, [online] Available from: http://hdl.handle.net/2027/spo.bbp2372.2013.044 (Accessed 9 February 2015).

Pulkki, V. (1997) ‘Virtual sound source positioning using vector base amplitude panning’, *Journal of the Audio Engineering Society*, 45(6), pp. 456–466.

Roads, C. (2004) *Microsound*, New Ed edition. Cambridge, Mass., MIT Press.

Schafer, R. M. (1977) *The Tuning of the World*, Knopf.

Schutz, M. and Lipscomb, S. (2007) ‘Hearing gestures, seeing music: Vision influences perceived tone duration’, *Perception*, 36(6), pp. 888–897.

Serafin, S. (2014) ‘Sonic Interactions in Multimodal Environments’, In Collins, K., Kapralos, B., and Tessler, H. (eds.), *The Oxford Handbook of Interactive Audio*, Oxford University Press, [online] Available from: http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199797226.001.0001/oxfordhb-9780199797226-e-014 (Accessed 2 March 2015).

Shannon, C. E. (1948) ‘A mathematical theory of communication’, *The Bell System Technical Journal*, 27(3), pp. 379–423.

Smalley, D. (2007) ‘Space-form and the acousmatic image’, *Organised Sound*, 12(01), p. 35.

Smalley, D. (1997) ‘Spectromorphology: explaining sound-shapes’, *Organised Sound*, 2(2), pp. 107–126.

Truax, B. (1992) ‘Electroacoustic Music and the Soundscape: The inner and the Outer World’, In Payner, J. (ed.), *Companion to Contemporary Musical Thought*, pp. 374–398.

Weale, R. (2006) ‘Discovering How Accessible Electroacoustic Music Can Be: the Intention/Reception project’, *Organised Sound*, 11(02), p. 189.

Wishart, T. (1996) *On Sonic Art*, Psychology Press.

Wrightson, K. (2000) ‘An Introduction to Acoustic Ecology’, *Soundscape: The Journal of Acoustic Ecology*, 1(1), pp. 10–13.

Yoh, M.-S. (2001) ‘The reality of virtual reality’, In *Seventh International Conference on Virtual Systems and Multimedia, 2001. Proceedings*, pp. 666–674.

Appendix A

Conference Paper 1

ICMC 2013

Perth, Australia

SONICMAPS: CONNECTING THE RITUAL OF THE CONCERT HALL WITH A LOCATIVE AUDIO URBAN EXPERIENCE

Ignacio Pecino
NOVARS Research Centre,
University of Manchester, UK

Ricardo Climent
NOVARS Research Centre,
University of Manchester, UK

ABSTRACT

Physical space is often used as a means for sound organization. Furthermore, our environment can be sonically augmented, highlighting hidden aspects of it or proposing a whole new reinterpretation. On this basis, we present SonicMaps, a new locative audio tool and complete solution for sound geolocation. Using a number of sensors on mobile devices, the SonicMaps application virtually places sounds into real space providing panning and amplitude information. These sounds are played back as we walk into predefined areas which can be overlapped into layers according to compositional purposes. A custom Dynamic Content Server (DCS) automatically updates audio files for selected locations according to a set of environment variables (time, temperature, etc.) creating a dynamic piece of music that emphasizes the link between sound materials and their assigned physical location. Additionally, a OSC module has been implemented in order to send users positions and sensor data to a remote server. This data can then be musically interpreted in a concert hall to establish creative links between distant spaces.

1. INTRODUCTION

1.1. Extending Space

By definition, space is a boundless, three-dimensional extent in which objects and events occur and have relative position and direction. We usually use this term referring to the physical extent in which we live, and is subject to Newtonian laws. Such an extent and everything that exists within, is what we perceive as Reality. However, the introduction of computer-generated 3D environments, although being simulated and not necessarily governed by these natural laws, also offers an instance of space that conforms to the given definition. This “Virtual” type of space can then be explored and navigated using multiple interfaces.

However, "Virtual Space" and "Real Space" are only the two ends of what we know as the "Virtuality Continuum" (Fig.1). Between these ends we find a wide mixed reality that shares elements from both ontological levels. We can conceive, for instance, a virtual environment where sounds events are being streamed, in real time, from a remote location in the real world. This situation could be described as Augmented Virtuality.

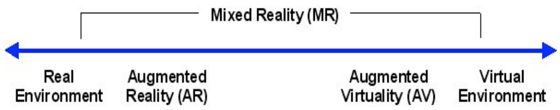


Figure 1. The Virtuality Continuum.

On the other hand, real space can also be augmented incorporating elements from the digital domain, which are not originally present but are displayed using some kind of interface (Augmented Reality). This is the baseline for the sound geolocation tool we will be describing here.

1.2. Locative Audio

In that context, Locative Audio can be defined as a type of Augmented Reality based on sonic experiences (Augmented Aurality). It works by adding a layer of auditory information to our real-world environment and consequently modifying or enhancing our perception and understanding of a particular place. Nowadays, this can be achieved by using existing sensors on mobile devices, like GPS or a solid state compass, which track our position/orientation and render this layer accordingly. Frauke Behrendt discussion about ‘the sound of locative media’ sets a clear framework to explain how we can listen to space by walking and thus remixing a number of placed sounds [1] . Such an approach contrasts with other geotagged audio projects, where a collection of field recordings are just displayed on a website using the Google Maps API or other mapping solutions [2].

1.3. Networked Performances.

Recent internet communication speed improvements has also led to a whole new category of musical practice known as networked music performances [3]. These projects often involve a number of geographically displaced artists who interact in real-time to produce a musical output. A notable example of a networked audio system is JackTrip (CCRMA, Standford University) where several remote audio streams can be mixed together from a master server [4]. However, several limitations arise when high quality audio streams need to be sent and synchronized over the internet, given the large amount of data involved. An alternative networking procedure can be achieved by using a lighter and more flexible musical data format like OSC, where no real audio is required , but a series of short messages.

This protocol, in combination with the SonicMaps locative audio tool, opens a door to the possibility of linking not just performers, but multiple distant spaces in new imaginative ways. For instance, we will discuss how a locative audio urban experience, in the terms earlier described, can be connected to a live Concert Hall event, establishing meaningful relationships between these two spaces and their audiences.

2. SOFTWARE DESIGN

SonicMaps was developed using the Unity3D Game Engine[1] and takes advantage of its 3D spatial audio engine, along with its rendering and physics engine, scripting languages (JavaScript, C#), and multi-platform publishing capability. Currently, SonicMaps is available on iOS and Android devices while user content management and support is provided from a PHP enabled website[2].

In order to geolocate sounds, a geospatial positioning system was implemented by translating the geographical coordinate system used by GPS mobile devices (WGS84), into Unity's Vector2 Cartesian coordinates (x,y). This was completed using the Mercator projection as we wanted to use Google Maps images to render our navigation display. However, everything had to be scaled down given the Single-precision floating-point format that Unity uses for object transforms. That means you can only have 7 digits to set a coordinate and anything longer than that will be truncated with a consequent loss of valuable information.

Additionally, we avoided using absolute coordinates from a fixed frame of reference and opted for a relative positioning system that takes the first GPS coordinates (on application start) as a relative frame of reference. This value is stored on every session, so if we run the application somewhere else and the initial reference point changes, all objects positions can be recalculated while numbers are kept below the 7 digits limitation.

2.1. User Interface

When starting SonicMaps, it will first try to determine our current position. The message "Searching for Satellites" is displayed on the screen so location Services needs to be enabled on the device before we run the application. It is also necessary to check that GPS signal is good enough; otherwise, after 30s user is asked to check location settings and restart SonicMaps. Sometimes, when connected to Wi-Fi (without GPS enabled), the application might still be able to get our location, but this will result in much less accuracy.

Once our current location has been set, the main UI layout shows up (Fig.2).

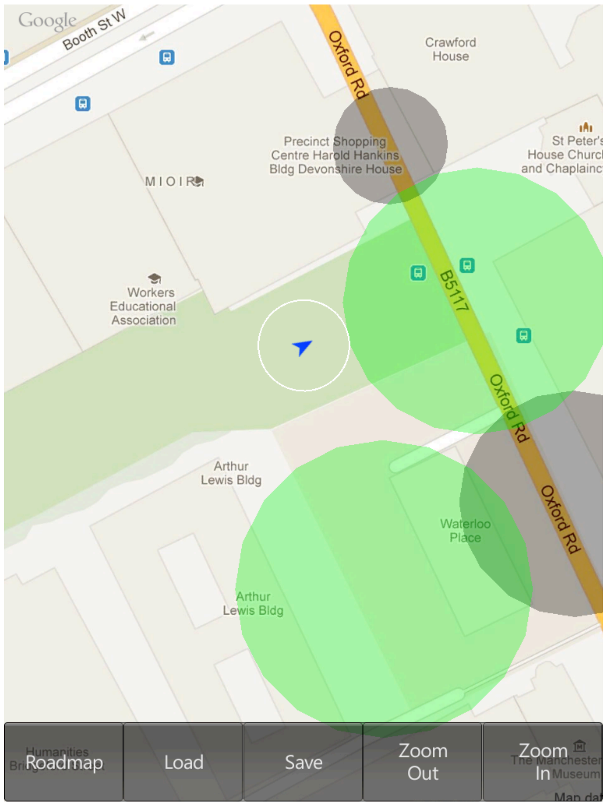


Figure 2. SonicMaps User Interface.

A map tile for the current location is obtained from the Google Static Maps[3] service using a Wi-Fi or 3G internet connection, and it serves as a texture for a simple plane mesh whose dimensions conforms to the considered scale. On the other hand, user position is displayed on screen as a small blue arrow that reads the incoming GPS and compass sensor signals and updates its transform (position and Euler angles) correspondingly. If we walk "off the screen", the message "Updating Position" will automatically pop up, and the map will be centered at our new position.

Sounds are triggered whenever this arrow enters a cylindrical mesh containing an audio file (sound area). Both, the user position arrow and the sound area mesh include a Collider component which is used by Unity's physics engine to detect collisions. More precisely, the cylinder's collider is enabled as a trigger, so we are able to code different sonic behaviors by using the OnTriggerEnter, OnTriggerStay and OnTriggerExit JavaScript functions.

[1] http://unity3d.com/
[2] http://sonicmaps.org
[3] https://developers.google.com/maps/documentation/staticmaps/

However, audio files are not initially stored in the mobile device but online, using a personal server or file hosting service. Therefore, sound areas only contain URLs to online audio files, avoiding the manual process of having to transfer these files into every device before we can play a project. Every project is saved as a JSON[4] text file containing serialized data about the sound areas and their properties. This makes content publishing and sharing extremely easy, with the help of a dedicated database.

2.2. Sensors implementation

2.2.1. GPS tracking

Thanks to GPS Location Services, user position is updated every half a second, although transitions are smoothed over time using linear interpolation. The sensor's accuracy is graphically displayed using a white error-range circle around the position arrow. In some devices this accuracy may be as good as 5 m or less.
Tall buildings, mountains, and other objects, often interfere with or interrupt the GPS signal from satellites and Google maps are not always 100% accurate so they might present significant displacements from the actual terrain [5]. SonicMaps tries to minimize these issues by offering what we have defined as “on-site” editing, that is to say, being able to edit projects from a mobile device, as we walk, and ensuring that sounds are just at the desired location. Therefore, positioning inaccuracies are taken into account while editing so unwanted displacements are not perceived during content playback. This method is also useful to decide if sound materials are aesthetically or functionally suitable for a particular place, by actually being there as we edit.

2.2.2. Solid state compass

The solid state compass sensor is called every 50ms in order to get orientation data. The user position arrow rotates to match the new heading and surrounding sounds are panned accordingly. Noise data from the compass sensor due to random movements, e.g., when handling the mobile device as we walk, is filtered so only substantial rotations are visualized.

2.3. Audio Engine

2.3.1. Spatial 3D sound.

The Unity Game Engine uses an implementation of the FMOD audio library for the creation and playback of interactive audio. This library includes an Audio Listener component (virtual microphone) to output spatial audio resulting from any audio source in a scene. SonicMaps attaches the audio listener to the user position arrow so sounds are perceived from this point of view. Likewise, Audio Source components can be found in every sound area we create and will be assigned a transform position (Vector3). Therefore, audio signals experiment a customizable amplitude drop-off (volume attenuation) over distance and are panned into the stereo sound field according to their relative angular position. Multichannel configurations[5] are also possible but they are not currently supported on mobile devices.
Three playback modes have been devised as a result of combining two fundamental variables:

- 3D/mono
- 2D/stereo or mono
- Pseudo-3D/ stereo.

A 2D sound is just a common stereo or mono audio file which is played with no spatial qualities, just as they would be heard from any regular music player.
The 3D property relates to the spatial sound feature we discussed previously. However, all 3D sounds in Unity are forced into mono by default, for easier panning and a more effective spatial localization.
This limitation can be partially corrected by emulating distance attenuation with a simple custom JavaScript to produce a stereo pseudo-3D audio source, with no panning or directional properties, whose volume increases as we get closer to the center of the defined area.
Audio clips can also be looped, which is useful when we need to maintain a sustained sonic texture within an area and it is not possible to predict how long will the user stay in that particular location.
Loops also help to keep audio file size small, which is strongly recommended when relying on a slow and sometimes expensive 3G internet connection.
Another important issue, specially when working with 2D sounds, is that whenever a listener leaves a sound area, the corresponding audio clip would suddenly stop playing and produce undesirable artifacts (clicks, etc.). Nevertheless, a 2-3 sec fade out script (depending on frame rate) is activated on trigger exit, providing a more consistent termination.

2.3.2. Sound Areas and audio buffering

Every sound area can be explained as a combination of two separate components: a buffering area and a playback area (Fig.3).

By default, every sound area includes a link URL to a sample audio clip (http://sonicmaps.org/sample.mp3) that will start downloading whenever we enter the buffering area. This buffering area guarantees that the sound is ready to play by the time we reach the actual playback area. The size of the buffering area is proportional to the size of the playback area, and once the download is complete the latter turns green as a notification.

[4] http://www.json.org/

[5] Quad, 5.1 and 7.1 surround.

2.3.3. *Audio Playback*

Any audio file contained in a sound area will automatically start playing any time we enter its playback area (Fig.3). When we exit the playback area, if the sound is still playing, it will rapidly fade-out and stop playing. The fade-out time is no longer than 1-2 seconds and it has been implemented to avoid sonic artifacts. An external audio editor can be used if we need longer fade-outs or any fade-in. SonicMaps does not apply any fade-in effect to sounds when entering an area. However, a natural fade-in effect is perceived when using the 3D audio option, since amplitude depends on how close we are to the center of the sound area.

Figure 3. Audio buffering and playback areas.

2.4. Sound Areas editing

To create a new sound area, users just need to tap the map right at the place where they want to locate it and press a "New Area" button in a resulting window. A gray circular area is then instantiated.

If we touch this newly created area, it turns red (editing mode) and the sound properties windows are displayed (Fig.4).

In this new window user can set the URL for the corresponding audio file, toggle between 3D/2D modes, activate the loop playback, set the volume, etc. The size of the sound area can also be modified with a simple touch and drag gesture.

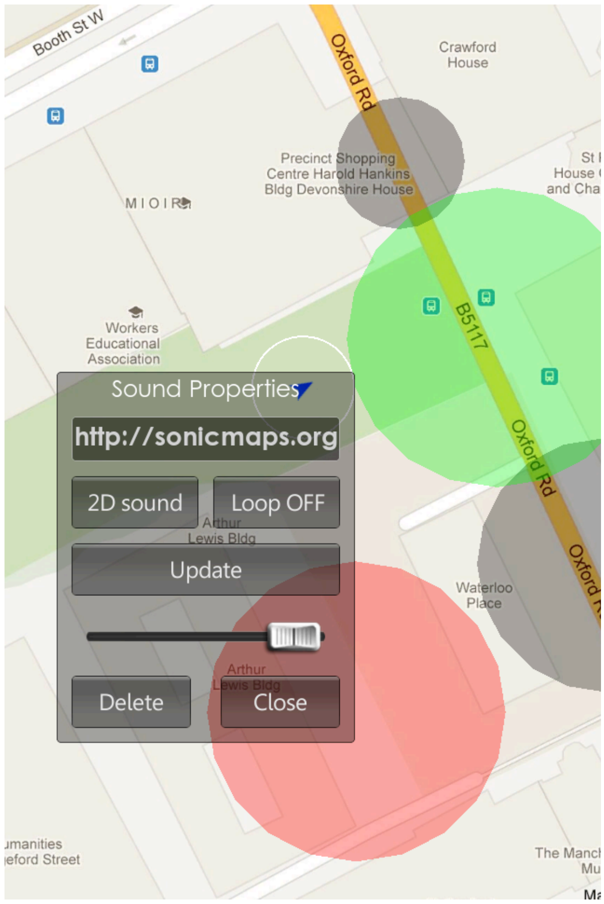


Figure 4. Sound Areas editing.

2.5. Project publishing. A public platform for locative audio content.

Once all sound areas have been created, the project can be saved and published along with some descriptive metadata into a MySQL database which is accesible for other users via a web page.

Figure 5. SonicMaps Locative Audio project publishing

To load a published project, users need to provide a link to the corresponding JSON text file (available in the web page) so SonicMaps can re-create all the sound areas in that project.

This system described allows the user to execute all necessary steps from one single mobile device without the need of USB data cables or any external equipment.

The website and database was designed to serve as an open platform for free locative audio content with a focus on usability and accessibility.

2.6. Dynamic Content Server (DCS). Sound-Events Geolocation.

If we turn our attention to the content itself and its relation with the environment we will soon realize that this living spaces are not static but dynamic; they evolve over time and affect the meaning of our sounds and vice-versa [1]. Consequently, under changing environmental conditions, a 'placed sound' might lose part of its meaning and suitability or even suggest something different of what it was intended.

We tried to resolve this problem by using a dynamic content server that provides adapted versions of a sound depending on the current environmental conditions (time, date, temperature, etc.).

This process relies on a Unix cron job[6], which periodically calls a php script on the server. This script reads the current conditions from the system status (time, date) and other online data feeds (weather conditions) and automatically selects the most adequate sounds for every sound area in a SonicMaps project. The update is achieved using a folders system to organize the available resources while the php script copies the right files into an active folder that can be accessed by the mobile application (Fig.6).

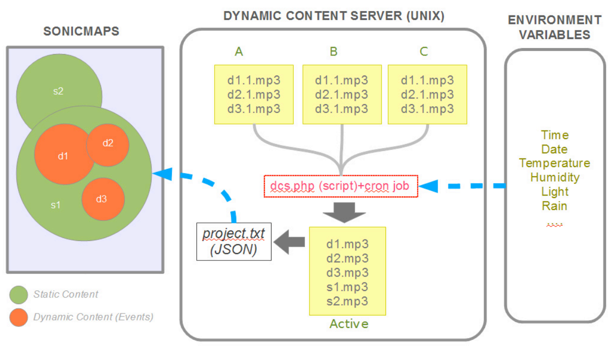


Figure 6. Dynamic Content Server

From this new approach, emerges the idea of Sound-Event Geolocation vs (simple) Sound Geolocation. Sounds are now placed at a specific point in space and time according to our compositional needs.

[6] http://livecronjobs.com/

2.7. OSC Module.

In order to facilitate network communications between the locative audio tool and any external software capable of handling OSC messages[7], we have implemented a OSC module for SonicMaps.

This module sends the user's position to a remote IP address via UDP packets so multiple clients are able to connect simultaneously (Fig.7). However this particular approach requires that the specified port number is open in the server machine and port forwarding has been enabled (when behind a router).

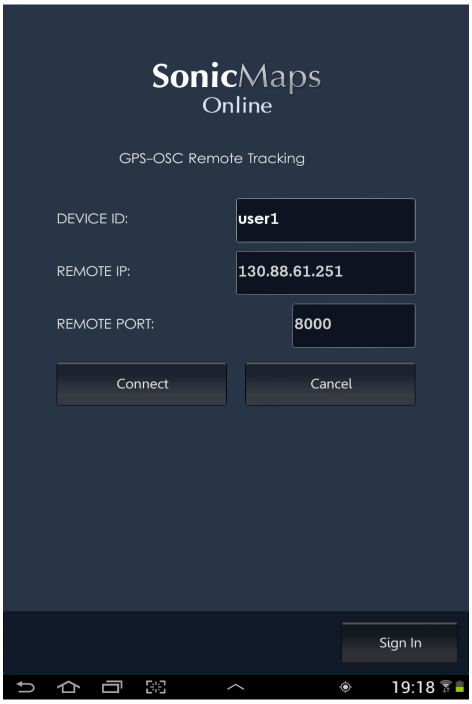


Figure 7. SonicMaps OSC network setup

The OSC messages being sent are formatted as follows:

/userName/GPS_absolute lon lat

/userName/XY_relative transform.position.x transform.position.z

where *GPS_absolute* is the actual longitude and latitude from the GPS sensor and *XY_relative* are the Cartesian coordinates in Unity's world units using the frame of reference we mentioned earlier. Notice that the Y coordinate is obtained from *transform.position.z* given that Unity uses the Y axis as height, so horizontal movements are described on the XZ plane.
Whether we choose one or the other depends on how we intend to make use of this information and what is the software running on the other side of the network.

3. CONNECTING SPACES

SonicMaps might be used as a stand-alone compositional tool and playback system for locative

[7] http://opensoundcontrol.org/

audio. Nevertheless, the implementation of network capability via OSC suggests a number of more elaborated forms of networked and immersive sound exploration. In particular, we were interested on the possibility of connecting sonically augmented outdoor urban spaces, with the more traditional concert hall or museum experience, proposing a creative framework for composers and sonic artists to construct a dual interlinked event.

3.1. SonicMaps + MAX

An interesting sound installation, *Kinesthesia* (Ivica Ico Bukvic, 2012), exploring these ideas, was presented during an Immersive Audio-game Showcase in the University of Manchester. The piece analyzes the human geospatial motion of multiple participants, whose cumulative actions form a subconsciously collaborative data stream devoid of time and space. The ensuing data stream was broadcast from Blacksburg (United States) to Manchester (United Kingdom) where it is reconstructed inside an autonomous meta-instrument and presented to an audience as a persistent spatially-aware installation.
Up to eight people with mobile devices running SonicMaps were asked to walk across the Virginia Tech Campus, experiencing a custom soundwalk. Meanwhile, the OSC module was sending their positions to a remote computer in Manchester running a MAX patch that used the incoming geolocated data to feed several sound objects and shape a multichannel musical output.

3.2. SonicMaps + Game Engine (Unity3D)

It is also possible to connect a physical real space and a computer-generated virtual representation of that space for a sonic-centric networked interaction.
For example, the piece *Alice: Elegy to the Memory of an Unfortunate Lady* (Ignacio Pecino, 2012), uses a Unity3D virtual model of Whitworth Park (Manchester) to display the current position of a remote user walking on the real physical park. This virtual environment is then presented to an audience in a concert hall so it is possible to experience, in a synchronized way, the same sounds the remote performer is triggering as he walks outdoor using the SonicMaps mobile application.

Figure 8. SonicMaps + Unity3D visualization (Alice).

However the concert hall version benefited from a 5.1 speakers configuration, while the mobile device could only provide stereo sound, which reinforces the idea that connecting different types of space (real, augmented, virtual, indoors, outdoors) and taking advantage of their specific nature, might lead to satisfactory multifaceted experiences.

4. RELATED WORKS

A relevant work by composer Andy Dolphin in collaboration with programmer Kingsley Ash, *Urbicolous Disport* (2012), is an audio-visual installation exploring connections between urban sounds and interactive abstract sound toys generated with the Unity3D game engine. In their own words:

"Urbicolous Disport is an interactive, generative sound-toy installation in which participants capture, manipulate and play with the sounds of the city. Participants use mobile devices to collect sounds from around the city, which are then streamed, reworked and transformed to form the source materials in an original sound-toy application presented in an installation setting. The work therefore exists as a collaboration between the sound recordists and the installation participants, allowing them to explore the sonic environment of the city in different ways and collaboratively influence the resulting audio-visual output". [8]

An example of locative sound using a smart phone application is the National Mall (2011), based around an outdoor park in Washington DC [6]. However, this immersive experience is restricted to that proposed environment, so no connections are established with further networked spaces.

5. REFERENCES

[1] F. Behrendt. "The sound of locative media", *The International Jouornal of Research into New Media Technologies*, Convergence, vol. 18 no. 3, pp. 283-295, 2012.

[2] J.Palacios. "Mapa sonoro de Asturias". Available at: http://mapasonoru.com/mapa.php (accessed 15 February 2013)

[3] A.Renaud, A. Carot, P.Rebelo. "Networked Music Performance: State of the Art", *AES 30th International Conference*, paper 16, Saariselkä, Finland, March 2007.

[4] J.P. Cáceres, C.Chafe. "JackTrip: Unde the Hood of an Engine for Network Audio", *Proceedings of the International Computer Music Conference*, Montreal, Canada, 2009.

[5] M.Modsching, R.Kramer, K.T.Hagen. "Field trial on GPS Accuracy in a medium size city: The influence of built up", *Proceedings of the 3rd workshop on positioning navigation and communication (WPNC'06)* , pp. 209-218, 2006.

[6] Bluebrain (2011) "The national mall by BLUEBRAIN: The first location-aware album". Video. Available at: http://vimeo.com/24250620

[8] http://www.urbicolousdisport.org.uk/

Appendix B

Conference Paper 2

ICMC 2014

Athens, Greece

SPATIAL AND KINEMATIC MODELS FOR PROCEDURAL AUDIO IN 3D VIRTUAL ENVIRONMENTS

Ignacio Pecino
NOVARS Research Centre,
University of Manchester, UK
joseignacio.pecinorodriguez@postgrad.manchester.ac.uk

ABSTRACT

In the last years an increasing interest can be observed for developments in game engine technologies as a versatile creative tool. In particular, the possibility to visualize and simulate real-time complex physical behaviors facilitates the design and implementation of 3D virtual music instruments and the exploration of sound gesture as a result of their kinematic and spatial properties. This paper describes two case examples in the form of linear compositions based on non-conventional instrumental designs where audio is procedurally generated using custom-built APIs in the game engine scripting language (Unity3D-Javascript/C#). Sound events are also organized as a sequence of flexible code instructions, resulting in a quasi-fixed piece duration with subtle timbral variations over multiple playbacks. In both cases, the model presented shows inherit spatial characteristics, which are useful in order to build spatialization patterns in a multichannel loudspeakers configuration and emphasize the strong causal connection between the visual and sonic aspects of these works.

1. INTRODUCTION

1.1 Game Engines as Multidisciplinary Creative Environments.

The need for rich development environments, especially in the field of sound art and interactive media, has led to the incorporation of new tools [1] [2] which, although originally devised for the video game industry, have proven to be an extremely convenient and affordable solution for many sound artists.

This is easily understandable as video games development often requires similar software and hardware capabilities as many of the interactive works proposed by sound artists, including graphics rendering, time management methods, an audio playback engine, input interfaces, etc.

In some occasions, this type of software can also be used for prototyping complex art installations where an impact assessment is required [3], yet the usage of game engine technologies goes far beyond real-world simulation models, allowing for innovative and original approaches to instrumental design and composition where virtual tridimensional space can be considered as the main structural variable that shapes the musical output.

Notable examples of compositional applications of a game engine [4] can be found in the works of Robert Hamilton [5], Ricardo Climent [6] or Andrew Dolphin [7] [8], whose “sound toys” are a very interesting demonstration of interactive tools with a focus on space and real time synthesis (in MAX/MSP) using control data from a 3D engine (Unity). However, it is possible to appreciate how most of these and other works using game engine technologies are essentially interactive in nature, as they need some sort of live user input in order to produce sound, whether it is coming from a member of the audience (e.g. a sound installation) or the artist himself (during a performance or at the studio).

In contrast, the two cases described later in this paper are an attempt to reconcile the more traditional approach of fixed-media and acousmatic composition with the emerging trends and procedural techniques that these new development environments have to offer. In particular, the potential for 3D objects manipulation, integrated physics (collisions detection, gravity, etc.), along with the built-in scripting languages, are exceptionally helpful when mapping dynamically generated spatial and kinematic data into the audio engine of our choice (e.g. SuperCollider [9]), while sound events can be programmatically sequenced over time from a custom script, analogously to how we organize sound materials or MIDI events in a DAW (e.g. Protools).

1.2 Procedural Audio

Given the real-time nature of most processes running in a game engine, a procedural approach to sound generation is possible and desirable, even in a non-interactive compositional context. An audio sample playback engine is generally provided in every game engine, but these are usually very limited in their functionality, permitting only basic volume and pitch controls. Consequently, it is often advisable to replace the default internal audio engine with a more specific and powerful software for real time audio

Copyright: © 2014 Ignacio Pecino. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

processing such as Supercollider or MAX/MSP. Both sides of the system (game engine and external audio software) are then interconnected using a standard communication protocol such as OSC [10].

Procedural audio [11], as opposed to recorded sound, is created in real time according to a set of rules or algorithms (sound as a process), reducing the amount of raw audio data needed to play a music piece although considerably more demanding in terms of CPU usage.

1.3 Live Coding vs. Sequenced Code.

The use of live coding (on-the-fly programming) as a form of musical expression is an increasingly popular practice within the music community. Supercollider and Chunk [12] are well-known examples of object-oriented programming languages that allow live interaction with code and improvisation.

On the other hand, the scripting languages that most game engines include (JavaScript and C# in Unity, Python in Blender, etc.) do not permit code modifications at runtime but, in most situations, they are still capable of managing time efficiently, as long as we do not require sample accurate audio processing.

In the two case examples proposed here, the scripting programming language is utilized in three different ways, corresponding to the three stages of the proposed compositional method.

First, it is required to implement the basic structural design of our virtual instrument, including the characteristic variables and components of the system.

Secondly, it allows us to define a set of functions or subroutines describing what our virtual instrument is able to perform (instrumental techniques) as result of the manipulation of the previous variables and objects.

Thirdly, as we mentioned earlier, all these functions (API) can be called as an organized sequence of events to create the temporal structure of the final musical piece (specific examples will be presented later in this paper).

Across all these three stages, minor random elements are introduced into the instrumental design and related algorithms, so the final output will be a piece with an approximate total duration that exhibits a macroscopic deterministic behavior while presenting interesting small variations at a microscopic level which are not directly determined by the composer (in contrast with conventional fixed-media pieces).

2. VIRTUAL 3D INSTRUMENT IMPLEMENTATION

Efforts to build virtual 3d instruments for musical interaction can be found prior to the emergence of game engines technologies [13], but the workflow involved in this process and the quality of the final output has benefited from the general features present in this accessible and “ready-to-use” development framework.

2.1 Space as the starting point

The two pieces we are about to discuss next, make extensive use of these virtual 3D environments, proposing non-conventional instrumental models with strong spatial attributes. The first piece (“Singularity”) uses a sonification of cellular automata to explore the growth and dissemination patterns of its cells as a motif generator for spatial concatenative synthesis. On the other hand, the second piece studies the kinematic behavior and sonic potential of a set of virtual rings, organized as a fractal geometric construct known as Apollonian Gasket [1], which is recursively generated from triples of circles where each circle is tangent to the other two. Given the elements involved—materials/physical objects, sound and movement—this second piece displays manifest similarities with generic sound sculpture works [14].

Nevertheless, none of the instruments employed in these pieces imitate any real-world conventional instrumental design, but suggest more abstract prototypes based on mathematical models. Thus, the focus is not on simulation but on the use of space and the possibility of composing original pieces using bespoke experimental instruments that might otherwise be difficult and expensive to implement in a real-world situation.

2.2 The Audiovisual Dilemma

It seems to be clear that the possibility of real-time visualization and manipulation of 3D objects within a game engine is certainly helpful when designing and implementing 3D virtual instruments, as we are establishing direct causal relationships between the operation/behavior of the simulated physical system and the following sonic output. However, since the main goal of the proposed method is to deliver a non-interactive music composition, it is not so clear what the nature of the final piece should be: acousmatic or audiovisual, especially when the music language used resembles that of the electroacoustic acousmatic tradition. In other words, should the audience be presented with a visualization of the virtual instrument performing the piece?

In principle, both points of views are equally valid (with their pros and cons) but, given the spatial attributes of the pieces being discussed, I tend to believe that the sonic gesture is better explained when we are able to see the object and movement involved, in the same way as a performer and his instrument can be observed in a traditional concert situation. Yet categorizing the piece as “audiovisual” might not always be correct, as the visual side of it does not necessarily have an aesthetic intention, in the way the musical output does.

Furthermore, the visuals of the mentioned pieces have deliberately been kept simple and minimalistic, avoiding

[1] http://en.wikipedia.org/wiki/Apollonian_gasket

unnecessary elements not related with the sonic mechanisms and reinforcing the original musical concept.

2.3 Audio Sources and Spatialization

One of the advantages of using game engines to play sound materials is that the number of final audio channels is not fixed, not even needs to be considered during the compositional process. Consequently, spatialization is mostly based on the movement and positions of virtual objects (audio-sources) in the 3D virtual environment. Therefore, sounds are automatically panned by the built-in spatial audio engine according to the current loudspeaker setup and the audio-listener object position (also included in the scene). Once the piece is completed, we are able to switch the desired multichannel output configuration from a list of available options, usually including stereo, quadraphonic, 5.1 and 7.1 surround.

Alternatively, if we decide to integrate an external audio engine, such as one of the previously discussed, this spatial functionality is lost and needs to be re-implemented. The solution considered in one the mentioned examples (Apollonian Gasket) is to translate Unity's Vector3 positions (x,y,z) into angular positions (r, θ) and send this data to Supercollider where sounds can be panned into any speakers array using VBAP [15]. The downside of this approach is that our piece will no longer work as a single standalone application but it will depend on external software to play our sounds. This is, however, a distribution issue and not a compositional one.

3. SINGULARITY: SONIC AUTOMATA

A sonic implementation of a cellular automaton is presented along this piece[2], consisting in an orthogonal grid of square cells (represented as cubes) that "live or die" according to a given set of rules known as Game of Life [16].

Figure 1. Screenshot of cellular automata in Singularity

The rules are quite simple but powerful enough to originate complex emergent phenomena. The originated shapes and patterns depend on the initial configuration of the automaton and evolve over time (using discrete time intervals/ steps) spreading across the different areas of the grid (see Figure 1). This behavior is particularly interesting from a compositional point of view, since it exhibits "life form" characteristics: growth, decline, chaos-order alternation, etc., which are structurally suitable for a musical discourse.

Also, the dynamic spatial dissemination of the living cells can be used to generate spatialization motifs within a surround loudspeakers configuration. In this case all sounds are played within Unity using the default audio engine.

3.1 Sonification of the Cellular Automaton

In order to translate the successive cells states into sonic gestures, an array of very short audio samples (<100ms) dissected from original recordings at the Museum of Science and Industry of Manchester, was used. Then, a piece of JavaScript code selects a random sample and plays it whenever a new cell comes to live. This sound is played at the cell's position, contributing to the surround sonic image perceived from the point of view of the audio-listener object at the center of the scene.

The rapid concatenation of micro-samples across a bi-dimensional space produces a macroscopic textural effect that can be categorized as a form of spatial concatenative synthesis, as long as the evolution rate is fast enough; although sometimes this rate is also kept relatively low and variable, favoring the apparition of distinct rhythmic patterns.

3.2 Software Control Interface

A number of variables are defined in the automaton's code, so we can modify its properties and state; for example: size (number of cells in the grid), initial density (how many cells are likely to be alive on start), initial area (size of the sub-grid where cells can be initially instantiated), pitch-shift (option to slightly detune audio samples), random-volume, random-rate, etc.

All the cells are initially disabled (dead state) but we can activate them from the main sequence script using the cell "state" property as follows:

```
var automaton_A: GameObject;
function Awake(){
        //Reference to the control script in the A automaton
        a = automaton_A.GetComponent(Automata);
}
function Start(){
        a.rate=0.1;
        a.state[10,12]=true;
        a.state[11,13]=true;
        a.state[9,13]=true;
}
```

The previous example would initiate a particular automaton configuration, which would evolve under the given rules (on a separate script) until a stationary state is achieved or all the cells are dead. As we can observe, the reference to the corresponding cell is provided by its (x,y)

[2] Recording of the piece available at http://vimeo.com/79344883

position in the grid (bi-dimensional array) while other properties can also be modified here.

In addition, specific functions are also created as methods to control the automaton's behavior, particularly, tempo related functions such as Accelerando, Ritardando, etc. The evolution rate (time between consecutive steps) can be fixed or randomly selected between a minimum and maximum rate. The next example shows a routine that progressively changes these values over a given period of time:

```
function ChangeRate(targetMin:float,targetMax:float,seconds:float){
   var t = 0.0;
   var currMin= minRate;
   var currMax= maxRate;
   while (t <= 1.0) {
      t += Time.deltaTime/seconds;
      minRate = Mathf.Lerp(currMin,targetMin, Mathf.Lerp(0.0, 1.0, t));
      maxRate = Mathf.Lerp(currMax,targetMax, Mathf.Lerp(0.0, 1.0, t));
      yield;
   }
}
```

3.3 Creating the final sequence

Once the instrument with its custom control interface is ready, we need to select a number of automata configurations as possible candidates for the final sequence, taking in consideration their musical potential and functionality within our piece. This process is analogous to the act of selecting sound materials for an acousmatic piece, although we are now working with shorts space-timbre motifs generated by virtue of the fast concatenative action of our cellular automata.

The sequence is possible thanks to several time related instructions such as the *yield* instruction of the Unity scripting language. For instance, if we need to wait 1.5s before a new motif is generated, it is just a matter of typing the following line:

```
yield WaitForSeconds(1.5);
```

Using a *while* statement, it is also possible to link two events so the second one will only take place once a certain condition involving the first event is met. For example:

```
//automaton A is ritardando
//Wait until its minimum rate is above 0.85
while (a.minRate<0.85){
   yield; //Wait 1 frame
}
//Play motif in automaton B
automata_B.SetActive(true);
b.state[1,1]=true;
b.state[1,2]=true
b.state[3,2]=true;
//etc.
```

Other subroutines are also created when multiple code threads need to run in parallel, facilitating the operation of simultaneous automata within our scene from a single master script (sequencer).

4. APOLLONIAN GASKET: A VIRTUAL SOUND SCULPTURE

As we mentioned earlier, this second case example explores the kinematics of a set of rings geometrically arranged as an Apollonian Gasket. This mathematical object was chosen not just because of its spatial characteristics, but also because of its simplicity and recursive elegance. Moreover, the kinematic model of the rings movement has multiple similarities with a popular sound toy known as Euler's Disk[3] and also resembles the natural behavior of a spinning coin.

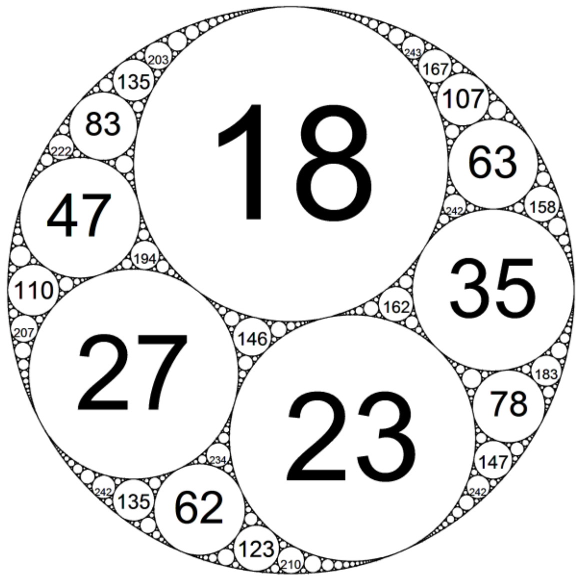


Figure 2. Spatial distribution of the Apollonian Gasket

The original Gasket is a bi-dimensional object as we can see in Figure 2, but it has been extended for this piece[4] into a third vertical dimension, in order to allow a richer kinematic experience (see Figure 3).

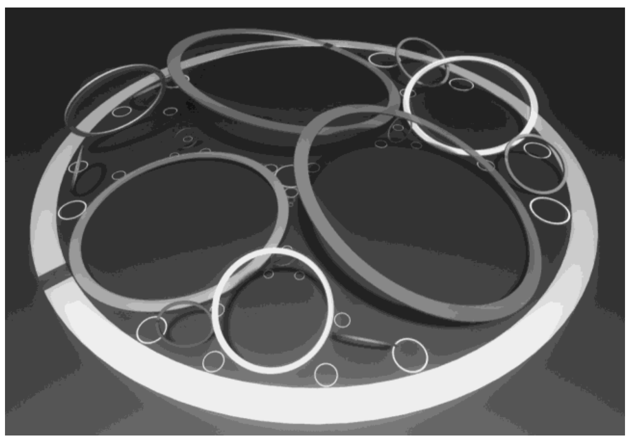


Figure 3. Screenshot of the 3D Apollonian Gasket Virtual Instrument.

[3] http://www.youtube.com/watch?v=ug2bKCG4gZY

[4] Recording of the piece available at http://vimeo.com/83995079

Given that the curvatures of the component rings are being used as a variable to feed sound parameters, I opted for an asymmetric shape given by the initial values 18, 23 and 27, thus avoiding excessive regularity and repetition.

4.1 Sonification of Kinematic Data

Audio is generated procedurally, in real time, using synthesis techniques that respond to specific changes in physical variables in the virtual objects (rings).

These objects are animated using a custom Unity script and the consequent kinematic data is sent to an external audio engine (Supercollider) via OSC[5] messages. Again, this approach permits a tight interconnection between the visual and sonic elements of the piece, since audio-video synchrony is nearly perfect and the movement/gesture of the virtual object can be effectively translated into sound.

The algorithms used in Supercollider include different types of synthesis such as Physical Modelling, FM/AM or Granular Synthesis and the input variables used to feed these generators include: ring curvature, tilt, rotation speed, precesion speed, collision strength, horizontal angular position (azimuth), etc.

One of the advantages of this kind of procedural sound generation is, for instance, the high number of sound variations that can be achieved with very light algorithms. For example, the .scd file used by SuperCollider to generate all the sounds in this piece, despite it contains more than 1500 lines of code, it only weights 30Kb (plus another 30Kb on the Unity scripting side). Also, these sonic variations are responsive and necessarily linked to the virtual object's kinematic variables, while a similar complex response using raw recorded audio would take a considerably larger amount of data.

4.2 Control Interface

Similarly to how we did in the first case example (Singularity), we need some kind of software interface to regulate the mechanics of the proposed instrument.

The system suggested is now based on an "energy" variable that determines the kinematic state of the rings. Then, we can introduce energy variations by calling the following function (C#):

```
IEnumerator ChangeEnergy (GameObject ring, float targetEnergy,
float seconds){
        float t = 0.0F;
        var precesion = ring.GetComponent<Precesion>();
        float currentEnergy = precesion.energy;
        while (t <= 1.0F) {
                t += Time.deltaTime/seconds;
                precesion.energy=
                Mathf.Lerp(currentEnergy,targetEnergy,
                Mathf.Lerp(0.0F, 1.0F, t));
                yield return 0;
        }
}
```

However, before any energy is introduced into the system, the multiple rings need to be initialized, both in Unity and SuperCollider. For example (Unity code):

```
public void StartRotor(){
        arcTan = Mathf.Atan(transform.position.x/transform.position.z);

        //Surround Panning
        if(transform.position.z>=0){
                azimut = (arcTan*180/Mathf.PI);
        }else{
                if(transform.position.x>=0){
                        azimut = 180 + (arcTan*180/Mathf.PI);
                }else{
                        azimut = (arcTan*180/Mathf.PI)-180;
                }
        }
        //Send OSC Messages
        List<object> values = new List<object>();
        values.AddRange(new object[]{curvature, azimut});
        OSCHandler.Instance.SendMessageToClient("SuperCollider",
        "start"+curvature.ToString(),values);
        rotorSynth=true;
}
```

The previous functions can obviously be reused as many times as needed from the master sequence script:

```
aro18.GetComponent<Precesion>().StartRotor();
StartCoroutine(ChangeEnergy(aro18,10.0F,5.0F));
```

These are just two example of the Apollonian Gasket API functionality, but a number of other methods can also be called to easily modify any required property.

4.3 Instrumental Techniques

The Apollonian Gasket instrument presents at least three distinctive timbres, depending on the characteristic movements/behaviour of the rings in 3D space, namely:

a) *Rotation*. Circular movement around the ring's main symmetry axis (Figure 3).

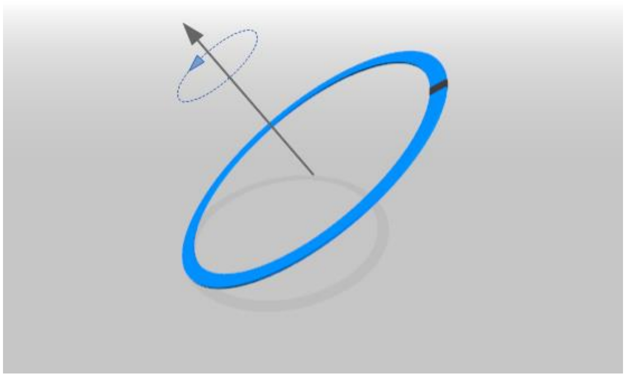


Figure 3. Rotation movement

[5] Using UnityOSC (https://github.com/jorgegarcia/UnityOSC)

b) *Precesion*. Changes in the orientation of the rotational axis around a vertical axis (secondary rotation). Rotation and precesion speeds are often related (Figure 4) and the consequent movement is translated into sound using real-time noise filtering and recursive modulation techniques in SuperCollider.

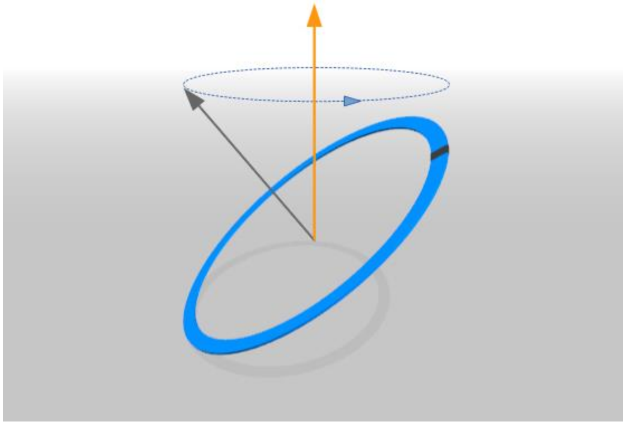


Figure 4. Precesion movement

c) *Collision*. A small percussive element is sometimes attached to the ring in a way that it will hit the floor at intervals, producing a collision sound as a consequence of the rotation/precesion movement (see Figure 5). In that case, the ring's tilt angle and the relative velocity of the collision can be used to feed several input parameters of an associated physical modelling synthesizer (Karplus-Strong UGen). Furthermore, the contact points generated by the physics engine provide the necessary vector data to position our collision sound using VBAP (see Section 2.3).

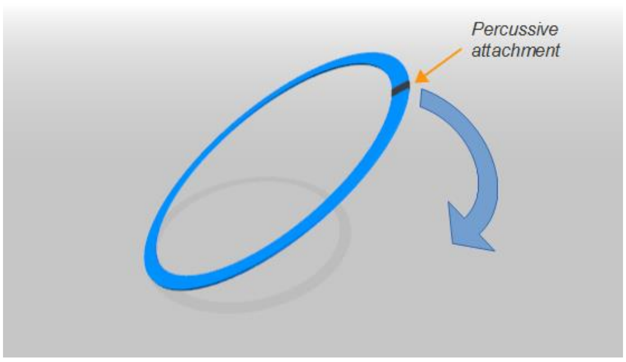


Figure 5. Collision gesture

4.4 Spatialization and Sound Diffusion

The geometrical distribution of the rings, according to the mathematical fractal structure, provides a quasi-symmetric layout with three main groups of rings placed on the Front-Left, Front-Right and Rear-Center (see Figure 6). The audio listener is then placed at the approximate centre of the set (ring146), providing an optimal listening position that can later be reproduced in the concert hall when playing the piece in a multichannel speakers setup (e.g. MANTIS[6] system). Every sound event is panned in Supercollider using the VBAP object, allowing for azimuth and elevation data on any array of speakers. This approach reinforces the original idea of channel independent composition where sound objects are spatialized in virtual space, regardless of the final number of output channels.

Furthermore, the Apollonian Gasket presents a sort of inside-outside separation given by the three original tangent circles, as all remaining circles will either be inside or outside of these three circles. The inner set is considerably smaller in size but has the same number of rings and similar spatial distribution as the outer set. This in/out characteristic can be easily used to create two separate multichannel buffers or stems (e.g. 4+4) and send them to different groups of speakers in the sound diffusion system (close vs. distant speakers) allowing for further exploration and manipulation of the piece during the live concert performance [17].

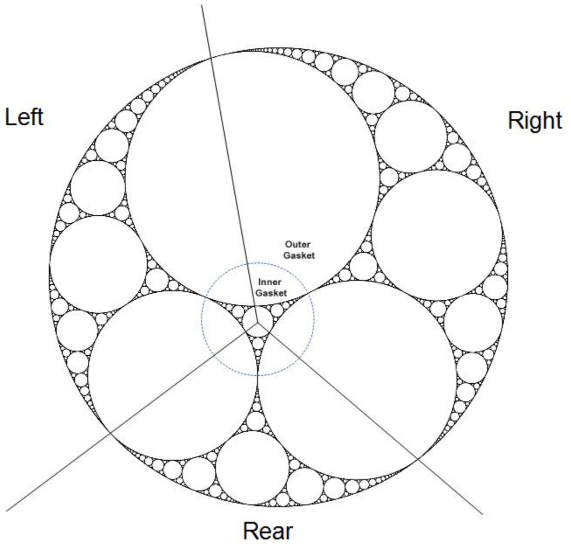


Figure 6. Spatial distribution of the Apollonian Gasket

5. CONCLUSIONS

Subsequent analysis of the ideas and techniques proposed in these two works seem to suggest that all the usual phases of music creation (e.g. instrument design/implementation, exploration of instrumental techniques, performance and composition) are also present within a game-engine environment and can be formalized using a programming language. This is indeed one more example of the existing virtualization trend that can be observed in many other disciplines; usually driven

[6] www.novars.manchester.ac.uk/mantis/

by economic factors, but focused here on extending the limits of musical expression.

We can also appreciate how real-time procedural sound generation is not necessarily linked to interactive media but may also be adopted for compositional purposes, for example, when a fixed macroscopic output is pursued while maintaining a certain degree of microscopic randomness or organic behaviour. Using code to organize our sound-generating processes (in contrast with traditional DAW timelines) has proven to be an effective way to achieve this goal, as it permits a flexible implementation and reinterpretation (playback) of these functions.

On the other hand, by extending our notion of space to the virtual domain, we have been able to explore sonic gesture as a result of simulated physical actions and spatial distribution/dissemination models that can be consolidated as viable 3D virtual musical instruments.

The complexity and level of detail of the processes involved are only limited by the actual processing speed of our CPU/GPU, but it is also possible to use separate networked machines for specific tasks (e.g. physics engine vs. audio engine) when the proposed model requires more processing power.

6. FUTURE WORK

By the time of writing this paper, all live performances of these pieces have been carried out in a concert hall using multiple loudspeakers and a conventional projection system (large screen at the stage). The most important limitation of this approach is that it requires some effort from the audience in order to apprehend the proposed correspondence between the visual on-screen space and the aural concert hall space. For example, for the perspective view displayed in Figure 1 and 3, objects rendered at the top of the screen would normally be played from the front loudspeakers, while objects at the bottom of the screen would be played from the rear ones (anything in-between is panned accordingly). Therefore, we find a physical separation between both spaces which is only overcome thanks to the listener’s intuition and imagination.

As an optimal future solution, a fully immersive visual system such as Oculus Rift[7] could be adopted. The transition to this new technology should be quite straight forward given that the game engine used for these pieces (Unity) has already been tested with this kind of hardware. Such an approach would be compatible with the current system using loudspeakers and VBAP panning, but alternative audio rendering methods based on binaural or Ambisonics techniques might also be considered.

[7] http://www.oculusvr.com/

7. REFERENCES

[1] “Unity3D”, http://unity3d.com/. Last accessed March 2014.

[2] “Blender.org”. http://www.blender.org. Last accessed March 2014.

[3] I.Bukvic, “Using gaming engine for virtual prototyping and impact assessment of complex interactive art installations”, in *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[4] A.Dolphin, A. “Compositional Applications of a Game Engine”, in *Proceedings of the Games Innovations Conference (GIC)*, London, UK, 2009, pp. 213-222.

[5] R.Hamilton, “UDKOSC: An immersive Musical Environment”, in *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[6] R.Climent, http://acousmatic.org. Last accessed March 2014.

[7] A.Dolphin, “Cyclical Flow: Spatial Synthesis Sound Toy as Multichannel Composition Tool”, in *Proceedings of the International Computer Music Conference*, Ljubljana, Slovenia, 2012.

[8] A.Dolphin, “SpiralSet: A Sound Toy Utilising Game Engine Technologies”, in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Pittsburgh, USA, 2009, pp. 56-57.

[9] “SuperCollider”, http://www.audiosynth.com/. Last accessed March 2014.

[10] M.Wright, A.Freed, "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers", in *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

[11] A.Farnell, “An introduction to procedural audio and its application in computer games”, available online at *http://obiwannabe.co.uk/html/papers/proc-audio/proc-audio.html*, 2007. Last accessed March 2014.

[12] G.Wang, P.R.Cook, “On-the-fly Programming: Using Code as an Expressive Musical Instrument” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Hamamatsu, Japan, 2004.

[13] A.Mulder, S. Fels, K. Mase, “Design of Virtual 3D Instruments for Musical Interaction”, in *Proceedings of Graphics Interface ’99*, Kingston, Canada, 1999, pp.76-83.

[14] J.Grayson. *Sound sculpture: a collection of essays by artists surveying the techniques, applications,*

and future directions of sound sculpture. Vancouver: A.R.C. Publications, 1975. ISBN 0-88985-000-3

[15] V. Pulkki, “Virtual Source Positioning Using Vector Base Amplitude Panning.” J. Audio Eng. Soc., 45(6): 456–466, Jun. 1997.

[16] M.Gardner, “The fantastic combinations of John Conway's new solitaire game "life", *Scientific American* 223, Oct.1970: 120-123

[17] D.Berezan, “Flux: Live-Acousmatic Performance and Composition”, *EMS: Electroacoustic Music Studies Network*, De Montfort/Leicester, 2007.

Appendix C

Málaga Soundwalk Brochure

Locative Audio Festival 2013

Málaga, Spain

PROGRAMA
Locative Audio
Málaga

11.00-11.30 Presentación del evento en el Conservatorio Superior de Música de Málaga (Sala Seminario).

11.30-11.50 Nos trasladamos al cine Albéniz en Calle Alcazabilla, frente a la pantalla gigante situada en su fachada desde donde se iniciará el paseo (ver mapa).

11.50-12.15 Los asistentes recibirán asistencia técnica para configurar sus dispositivos móviles y descargar el proyecto desde un punto de acceso Wi-Fi habilitado para tal efecto.

12.15-13.30 Se efectúa la primera parte del recorrido.

13.30-14.30 Pausa para almorzar (puerto deportivo - ZONA 6 del mapa).

14.30-15.30 Segunda parte del recorrido

OXFORD England

AVIGNON France

VALENCIA Spain

MÁLAGA Spain

VIRGINIA USA

LOCATIVE AUDIO
13 ABRIL 2013
INTERACTIVE SOUNDWALK

LocativeAudio.org

¿QUÉ NECESITAS?

Para disfrutar del paseo sonoro interactivo has de instalar la aplicación gratuita **SonicMaps Player** en un dispositivo iOS(iPhone, iPad) o Android compatible* equipado con auriculares. Una vez instalada, asegúrate de que estás conectado a Internet y emplea el siguiente enlace en la sección de carga de proyectos (LOAD):

http://sonicmaps.org/locativeaudio2013.txt

Si necesitas ayuda, no dudes en consultar a algún miembro de la organización.

ATENCION! Te recomendamos cargar completamente la batería de tu móvil para compensar el uso continuado del sensor GPS. Adicionalmente, tráete el cargador y recarga tu móvil si es necesario en el punto que habilitaremos para ello durante la pausa intermedia.

* La versión Android requiere un procesador ARMv7. Los modelos iOS compatibles son iPhone 3GS, iPhone 4/4S/5, iPad Wi-Fi + 3G, iPad 2 Wi-Fi + 3G/4G, iPad Wi-Fi + Cellular (4th generation) y iPad mini Wi-Fi + Cellular. Require iOS 4.3 o superior. Se necesitan al menos 200 MB de memoria interna disponible para almacenar el proyecto.

COORDINADORES DE LOCATIVE AUDIO NETWORK

AVIGNON. Sylvie Marchand, France. MigAA, European School of Visual Arts in Poitiers / Anguulême
TEMPERE. Ari Koivumäki, Univ. of Tampere, Finland
VOLOS. Nicolas Remy, Volos, Grecia
GRENOBLE. Julie McOisans, Cresson Institue, Grenoble, France
LINZ. Postgrados Master Interface Cultures, Kunstuniversität, Linz, Austria
MÁLAGA. Ignacio Pecino, University of Manchester & SonicMaps.org. Prof.Luis Torres Mancilla, Conservatorio Superior de Málaga.
OXFORD. James Percival, Oxford University
VIRGINIA. Ivica Ico Bukvic, Virginia Tech Institute for Creativity, Arts & Tech
HANOI. Joshua Kopecek, Mathias Rossignol, Dinh Lê Vân and Trí Minh
VALENCIA. Deco Nascimiento, Ricardo Climent, Miguel Molina, Michael Lau

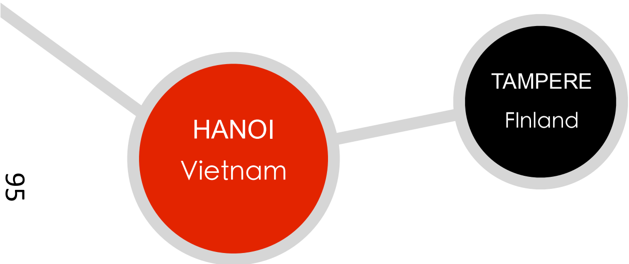
COLABORADORES Y SPONSORS

SALC´s Research Network Fund, Universidad de Manchester / Conservatorio Superior de Música de Málaga (Junta de Andalucía) / SonicMaps.org / Hanoi Soundstuff Festival / Facultad de Bellas Artes, Máster en Artes Visuales y Multimedia, UPV / Virginia Tech Institute for Creativity,Arts & Tech (ICAT) / Málaga Interactiva / VisualK videoproduction.

SonicMaps LocativeAudio

LOCATIVE AUDIO 2013 MÁLAGA INTERACTIVE SOUNDWALK

La tercera edición de Locative Audio + Network llega a Málaga de la mano de Novars Research Centre (Universidad de Manchester), Virginia Tech (USA), Universidad Politécnica de Valencia y el Conservatorio Superior de Música de Málaga, en un evento colaborativo y simultáneo que tendrá lugar el día 13 de Abril en múltiples ciudades del planeta: Valencia, Málaga, Virginia, Hanoi, Avignon, Linz, Grenoble, Volos, Tempere y Oxford. Dicho evento engloba un conjunto de paseos sonoros interactivos (soundwalks), actuaciones en directo y conferencias llevadas a cabo por diversos colectivos de artistas sonoros en torno a la idea “City as Museum / City as Instrument” (la Ciudad como Museo / la Ciudad como Instrumento).



En la edición de Málaga, un grupo de jóvenes poetas y compositores malagueños presentan un proyecto de audio localizado, en el que diversos lugares de la ciudad cobrarán vida sonora a través de poemas y obras electroacústicas originales. Estas piezas podrán ser experimentadas "in situ" mediante el uso de dispositivos móviles a lo largo de un recorrido que, a través del sonido, tratan de describir y resaltar algunos de los rincones de mayor interés turístico y cultural de nuestra ciudad.
Dicha información se presenta al usuario de forma interactiva a través de SonicMaps, una aplicación de geo-localización de audio que obtiene nuestra posición mediante satélites GPS y activa los sonidos correspondientes a medida que caminamos por las diversas zonas. La ciudad se convierte así en una gran sala de conciertos, abierta a todos, y en la que nuestro entorno cotidiano se ve aumentado a través de un conjunto de ideas novedosas. Estas ideas pueden tratar de recuperar historias o recuerdos sonoros del pasado, crear realidades sonoras alternativas o simplemente “adornar” dichos lugares con una capa de información inspirada en los mismos.

ARTISTAS COLABORADORES

1. Plaza de la Merced (Poema: PILAR GONZÁLEZ / Música: BOHDAN SYROYID)
2. Calle del Mundo Nuevo / Jardines de Puerta Oscura (Poemas: INMA BERNILS-PALOMA PEÑARRUBIA, VIOLETA NIEBLA / Música: DIANA PÉREZ CUSTODIO)
3. Plaza del General Torrijos (Poema: MAR LÓPEZ / Música: MAR LÓPEZ)
4. Plaza de Toros (Poema: ALEJANDRO ROBLES / Música: FRANCISCO JOSÉ MARTÍN JAIME)
5. Paseo Ciudad de Melilla (Poemas: GABRIEL NOGUERA, BEATRIZ ROS / Música: CARMEN VIDAL)
6. Paseo de la Farola (Zona Sur) (Poema: JACINTO PARIENTE / Música: IGNACIO PECINO)
7. Paseo de la Farola (Zona Norte) (Poema: CRISTINA SÁNCHEZ)
8. Paseo de los Curas (Poema: ÁNGELO NÉSTORE / Música: ISABÉL ROYÁN GONZÁLEZ)
9. Plaza de la Marina (Poema: MARÍA ELOY / Música: SARA ALMENDROS FLORES)
10. Paseo del Parque (Poema: LAURA CARNEROS / Música: JESÚS ORTIZ)
11. Catedral de Málaga (Poema: CAMILO DE ORY / Música: REYES OTEO)
12. Calle Larios (Poema: VIRGINIA AGUILAR / Música: LUIS TORRES)
13. Plaza de la Constitución (Poema: FRAN ROMERO / MUSICA: IMAT(Juan J. Raposo, Juan J. Espinosa)
14. Calle Granada (Poemas: CRISTIAN ALCARAZ, LAURA FRANCO)
15. Calle de San Agustín (Poema: SILVIA GUERRERO / Música: CHELO LÓPEZ)
16. Calle de la Alcazabilla (Poema: ADRIANA SCHLITTLER / Música: IGNACIO PECINO)