CD Reviews

David Berezan
Allusions sonores
empreintes DIGITALes, 2013
IMED 13122-CD

Martin Bédard
Topographies
empreintes DIGITALes, 2013
IMED 13121-CD

Pierre Alexandre Tremblay (Chen, Laporte, Nicolls, Roche)
La marée
empreintes DIGITALes, 2013
IMED 13123/124-CD(2)

Andrew Lewis
Au-delà
empreintes DIGITALes, 2013
IMED 13125-CD

Reviewed by Valentina Bertolani

Between 2011 and 2013, empreintes DIGITALes issued four compelling composer profiles, following in the tradition of the label. They are devoted to the Canadian composers David Berezan (Allusions sonores), Martin Bédard (Topographies), Pierre Alexandre Tremblay (Quelques reflets) and the English composer Andrew Lewis (Au-delà). Except for the last composer, whose work is represented by compositions spanning from 1987 to 2012, the work of the Canadian composers is showcased through their activity in the past ten years or so.

If at a first listening these recordings might seem very different from one another in style and inspiration, one might argue that they all pay homage to the music of Francis Dhomont and, in particular, to his Cycle du son – a piece also released by empreintes DIGITALes in 2001. Many aspects of these CDs nod to the aesthetics of the French composer; however, David Berezan and Andrew Lewis are the two that make the link between Dhomont's activity and their compositional work more explicit. The former acknowledges in an interview Dhomont's Cycle du son as one of the most seminal listening experiences he ever had [1]. Besides this, Berezan is also the director of the research centre Novars, in Manchester, and the centre itself has been named after one of the movements of Cycle du son. Andrew Lewis has also devoted some of his scholarly work to the analysis this work [2].

Notwithstanding this common interest, the two releases by Berezan and Bédard could not be more different. Berezan's title Allusions sonores is appropriate and indicative of the particularly appealing geography of this disc. In fact, inspirations for Thumbs (2011) and Galungan (2010) come from Balinese sounds, a region that the composer has been familiar with since the 1990s. The wooden floors of Japanese temples and buildings (namely, of Nijō Castle in Kyoto) and sea buoys - their actual sounds and metaphorical meanings - inform Nijō (2009) and Buoy (2011), respectively. Since all these locales are well known to the composer, he is able to capture them in their most mysterious and elemental aspects. Particularly fascinating is Badlands (2008), a rendition of the homonymous region of Southern Alberta, close to Berezan's native city of Edmonton. The piece is a holistic representation of this region that takes into account its culture, its history, and its prehistory: through use of Edmonton's natural sounds and allusions to the city's notoriously rich deposits of dinosaur fossils.

The nostalgic and inspiring atmospheres created by Berezan could not be further removed from the hyperenergetic, restless CD by Martin Bédard, entitled Topographies. Not only is the general allure of the album different; Bédard also gleans material from more industrial and urban atmospheres. The inspirations for his pieces come in a number of guises: from the history of the composer's hometown of Quebec City (Champs de fouilles, 2008); to the symbolic redemption of Quebec City jail, which becomes an occasion to explore musical narratives of freedom and captivity (Grand dehors, 2011); to the symbolic choice of metal as a compositional material, which becomes a metaphor of life and its toils in Métal fatigue (2012). Throughout the album, Robert Normandeau's concept of "cinema for the ears" is very present, such as in Push & Pull (2010), a powerful composition inspired both by the sounds of trains and the song Kid Gloves by Rush (a teenage favourite of Bédard's).
In order to have a real dialogue with popular music, though, you need to listen to Pierre Alexandre Tremblay's Quelques reflets. The pieces presented in this DVD are partially the outcome of practice-based research that Tremblay has been carrying out since 2001. The composer tried to find a synthesis of his previous musical experiences - studio composition, post-free-jazz improvisation, DSP coding, and a strong past background in popular music production - without relying on the physical presence of the performer/improviser. In his own words, Tremblay aims 'to bring critical improvisation practice in the studio to propose embodied post-acousmatic composition' [3]. The DVD includes compositions that reflect on the problems and dilemmas that the pursuit of happiness yields in the modern world. In Reflets de notre société crépusculaire (2009) the composer tries

to reconcile 'his happiness [… and the] recognition of it at the expense of the rest of the planet' (see CD booklet). Les trois petit c… (2010) and Ces énigmes limineuses (2010) are also engaged with the pursuit of happiness: the former focusing on women's resilience, and the latter a reflection on those moments that give us the feeling of a stillness or slowed time amidst the noise and commotion of contemporary life. The description of Walk That Way. Tuesday, Turn (2006), which certainly deserved the Mention it received at the 35th Bourges International Electroacoustic Music and Sonic Art Competition, gives as simple and unambiguous a depiction of the concerns that underpin Tremblay's compositional practice as could be hoped for. Indeed, its simplicity leaves the listener literally breathless: 'Inhale. Exhale. Repeat until life ceases'.

Unlike Tremblay, who explores the burden of simplicity, Andrew Lewis's Au-Delà provides a luminous and hopeful taking on weighty matters. Ascent (1994-97) showcases wide, static sounds derived from the natural soundscape around Bangor University. Similarly, Time and Fire (1987-90, 2013) and Cân (1997) engage with natural forces and soundscapes. In Lexicon (2012) and Scherzo (1992-93), the opening and closing pieces of the CD respectively, the sounds become less abstract with the introduction of children's voices. In fact, the latter piece uses the voices of the composer's daughters and the former, also available in a videomusic version, is based on a poem written by a 12 year old boy about his experience with dyslexia. With these four issues, the record label reinforces its aesthetic consistency. Leaving aside the very inconvenient packaging, which fails to protect the CDs, empreintes DIGITALes successfully navigates the dual roles of being both a safe harbour for the connoisseur of high quality art music, and a talent scout for new and off-the-radar composers.

References

[1] Soundcloud interview with David Berezan. 2013. https://soundcloud.com/sscloud1/sets/interview-with-david-berezan Last accessed 15 March, 2015.

[2] Lewis, A. 1998. 'Francis Dhomont's Novars', Journal of New Music Research 27, no. 1/2: 67.

[3] Tremblay, P.A. /Mixing the Immiscible: Improvisation within Fixed-Media Composition', Proceedings of the Electroacoustic Music Studies Network Conference, Meaning and Meaningfulness in Electroacoustic Music, Stockholm, June 2012, 3. http://www.ems-network.org/IMG/pdf_EMS12_tremblay.pdf

[4] See CD booklet

Book Review

Eric Lyon
Designing Audio Objects in Max/MSP and Pure Data
Middleton: A&R Books, 2012

Reviewed by Christopher Haworth

Cycling '74's Max/MSP, or 'Max' as it is now called, is easily the most widely used multimedia programming software worldwide. Over the past ten years it has achieved an unparalleled popularity amongst artists, designers, creative technologists, audio professionals and educators, far eclipsing its former running partner Pure Data, as well as Supercollider and Csound. The great success of Max owes much to CEO David Zicarelli's continuing efforts to improve the software's learning resources, workflow and documentation. Starting with Max 5 in 2008, and continuing to the present version 7, the company has invested a great deal of effort in the experience of new users, generally making Max programming appear less daunting to the uninitiated. The learning resources for the early and intermediate stages of Max apprenticeship therefore excel, but harnessing Max's more advanced features, especially programming external objects in the C language, has traditionally been a more treacherous path. Those for whom Max is the only programming language with which they are familiar would often complain about feeling stranded, since the available Max resources – the Max API, Software Developers Kit folder, developer forums, and the odd online PDF – all assume knowledge of procedural programming and integrated development environments like Xcode (tools that many initially turned to Max to avoid).

Eric Lyon's Designing Audio Objects for Max/MSP and Pd is written precisely for this audience. It will become an essential textbook for readers whose 'native' language is graphical programming but who wish to develop skills in audio DSP coding. From even the briefest glance at the opening pages it is clear why, despite the clear need for such a book, others may have balked at the task of writing it. Providing a comprehensive introduction to external development for both Max and Pd necessitates keeping multiple combinations of programming environments, C compilers, and operating systems in balance; a hard enough task in itself, but complicated by the violent cycles of change and renewal that Max has undergone over the last decade [1]. A mammoth task then, but as anyone who has encountered Eric Lyon at an ICMC, or used his FFTease externals for spectral analysis and synthesis in

Max will know, he is ideally suited for the job: an accomplished and prolific composer, audio researcher and teacher, and seemingly untiring in energy and zeal. His enthusiasm for all things computer music-related transports completely intact onto the pages of this book. From building a step sequencer (chapter 6, 'a sample accurate step sequencer'), to implementing avant-garde noise generators (chapter 13, 'dynamic stochastic synthesis'), one couldn't wish for a more engaging, more knowledgeable guide.

The book is organised into fifteen chapters, bookended by a foreword by David Zicarelli and an afterword by Miller Puckette. It ships with a CD-ROM containing the source code, compiled externals, and supplemental patches that are used as examples in the book. While it helps to have these files to resort to when things go wrong, I would recommend coding from scratch when following the examples. The learning curve of the book is nicely considered, beginning with exhaustive line-by-line explication of the code, and developing to increasingly abridged accounts that leave more up to the reader. It is complemented by a set of practical exercises that append each section; although easy to bypass, these sections are well considered in terms of the skills they aim to nurture, adding much to what precedes them.

The opening chapter covers the basics: the OS X and Windows development environments (and the use of Terminal in Linux), the Max SDK, as well as general practical considerations to bear in mind concerning external development.

Chapter 2 provides a useful refresher on digital signals, walking the reader through the process of writing it into a buffer using javascript inside Max. The chapter essentially functions to indicate the general level of competency that is required for readers of the book to progress: reasonably fluency with a text-based coding language is expected.

Chapter 3 is probably the crucial chapter in the book, and an invaluable resource for anyone that has ever gazed, bamboozled, at the source code of a Max external. It walks the reader through the entire process of designing, coding, compiling, and testing an external in Max and Pd, providing an extensively annotated line-by-line account of the key elements that make up the anatomy of a Max object: the header files, object structure, class pointer, function prototypes, initialisation routine and perform routine. Lyon's decision to omit any consideration of objects that run at the control and event rate means that the jump from chapter 2, where we discuss filling a buffer in Max, to chapter 3, where we develop a signal multiplication object ('multy~') from scratch, will be steep for most readers. Personally, I appreciated the sole focus on objects that run at signal-rate, but one should expect to return to this chapter a few times over the course of learning to fully absorb its contents.

One of the nice features of the book is that, with the exception of the demo externals of chapter 1 and 3, all of the objects that are described offer functionality that doesn't already exist in the native Max environment. As well as providing useful additions to one's own external library, this strategy serves to push home a smart development ethic: given the required time and persistence, it is only worthwhile implementing ideas that aren't already catered for by Max or the wider development community. Chapter 4, 'Variable Feedback with Delay', perfectly exemplifies this principle by showing how developing externals offers a means to overcome the native limitations of the Max environment. Specifically, it covers the design of a variable delay ('vdelay~') whose lower time limit is unaffected by Max of PD's signal vector size (a restriction that affects tapin~/tapout~, delay~, and delread~/delwrite~), thereby making such sound synthesis techniques as feedback FM possible in Max. Through the example, we address fundamental DSP and coding topics such as dynamic memory allocation, how to account for with changes to the global sample rate, and sample interpolation.

If chapter 4 provides an essential grounding in basic audio processing, then chapter 5 gets the reader started on sound synthesis. In this chapter we go through the process of implementing a new oscillator complete with options to set arbitrary wavetable sizes, as well as perform dynamic waveform allocation (triangle, square, sawtooth) and direct additive synthesis. The chapter introduces new topics such as normalisation and parameter-passing in Max, but by now much of the nuts and bolts of building externals have already been introduced. It allows Lyon to slightly ease up on the detailed explication of the previous two chapters, focusing only on the essentials. After a long and 'improvisatory' chapter on sequencing, where a simple premise – the design of a sample accurate sequencer – is taken in a more exploratory direction through the implementation of novel list manipulation algorithms, chapter 7 introduces non real-time buffer editing and processing. Here, Lyon details how to link to buffers inside Max, print internal buffer information to the Max window, and perform various editing operations including cutting, pasting, normalising and – a welcome thing in the Max world – reversing edits.

Chapter 8 marks the 'advanced' stage of the book. It covers the development of

two externals designed to be used inside the pfft~ system in Max: one, an adaptive noise reduction algorithm whose threshold changes in response to the amplitude of the input signal ('cleaner~'); and the second a spectral scrubber that allows for dynamic playback of spectral frames from a buffer ('scrubber~'). This chapter will be primarily of use to Pd users, firstly because it is not quite as clear why one would want to implement these functions in C rather than stay in Max and Jitter; and secondly because, differently to the other chapters, the rewriting of the external for Pd requires a complete redesign to account for the lack of a Pd equivalent to the pfft~ system. Working through this complex chapter should furnish the reader with a fairly high level understanding of advanced DSP coding that is transferable to VST plugin design, or designing objects for other environments like Supercollider.

Barring a chapter on Xenakis's dynamic stochastic synthesis (chapter 13), the remaining sections either focus on developing the reader's programming skills or on making use of new Max-specific features in external design. Debugging, code optimisation, benchmarking / profiling, and porting code from one platform to another all feature in these later chapters, whilst others focus on adding attributes to your externals (a new feature in Max 5), integrating externals into Max for Live devices, and updating one's externals to make use of the post-Max 6 64-bit processing (this last chapter is included on the accompanying CD-ROM).

In closing, I want to turn to my earlier-mentioned comments about the great challenge a book like this faced: not only in terms of keeping track of overhauls to the various development environments, but also in staying abreast of changes to the target environments. These latter can have the effect of either 1) rendering pre-existing code obsolete, or 2) adding functionality that is not accounted for in the book. Two years after the publication of the book, a revision to the Max SDK was introduced that amply satisfied the first condition; though trivial, it broke every single external in the book, meaning that the code on the CD-ROM now needs to be updated in order for it to compile [2]. Yet it is condition two that presents the biggest threat to the longevity of the book. The introduction of Gen in Max 6 has so improved the experience of programming in Max for advanced users that it is now possible to achieve many of the gains that one would normally build an external for without leaving the Max environment. Among other things, Gen allows one to bypass the signal vector limitation and create single sample delays, whilst also affording more efficient use of CPU resources. It also offers a choice between patching and code, which improves the readability of complex signal graphs. A recent addition has even provided the option to export code for use with other companies' SKDs, such as Steinberg, thereby filling the hole that the discontinuation of Pluggo left.

For the intermediate and advanced users I described in the opening, it may be that the introduction of Gen is enough for them: it has obviated many of the reasons to develop third party externals, and hence, the need for this book. However, just as the changing ecology of Max has direct effects on the skills and literacies needed of its user base, wider technological change can often cause them to return in new guises. Cycling '74 has been slow to interface with the iOS and Android operating systems, for instance, meaning that audio coding in Xcode remains an essential skill for app development. For this reason and others, I predict a long and prestigous shelf life for Designing Audio Objects by Eric Lyon. As the only textbook in the field designed for migrating from graph-based programming to audio coding in text-based languages, the transferable skills it will nurture in audio technologists will prove invaluable.

Notes

[1] First, the version update from 5 to 6, which introduced 64-bit processing to Max; and second, the introduction of Max for Live, the much-vaunted bridge between the worlds of Max and Ableton Live.

[2] The updated CD-ROM package can be found here: http://disis.music.vt.edu/eric/DAOCD_Update_2014/