# DVD Program Notes

## Part One: Thor Magnusson, Alex McLean, Nick Collins, Curators

### Curators' Note

[Editor's note: The curators attempted to write their Note in a collaborative, improvisatory fashion reminiscent of live coding, and have left the document open for further interaction from readers. See the following URL: https://docs.google.com/document/d/1ESzQyd9vdBuKgzdukFNhfAAnGEgLPgLlCe_Mw8zf1Uw/edit?hl=en_GB&authkey=CM7zg90L&pli=1.]

**Alex McLean** is a researcher in the area of programming languages for the arts, writing his PhD within the Intelligent Sound and Music Systems group at Goldsmiths College, and also working within the OAK group, University of Sheffield. He is one-third of the live-coding ambient-gabba-skiffle band Slub, who have been making people dance to their algorithms across Europe since 2001. Alex is janitor of many organizations including TOPLAP, POTAC, dorkbotsheffield, and the placard headphone festival. Further details are found on his Web site (yaxu.org).

**Thor Magnusson** is a musician/writer/programmer working in the fields of music and generative art. His PhD from the University of Sussex focused on computer music interfaces from the perspective of philosophy of technology, phenomenology, and cognitive science. He is a senior lecturer in the School of Art and Media at the University of Brighton. Thor is a co-founder and member of the ixi audio collective. With ixi he has written a variety of musical software and given workshops and talks at key institutions across Europe on the design and creation of digital musical instruments and sound installations. Further details are found on his Web site (www.ixi-audio.net).

**Click Nilson** is a Swedish avant garde codisician and code-jockey. He has explored the live coding of human performers since such early self-modifiying algorithmic text pieces as *An Instructional Game for One to Many Musicians* (1975). He is now actively involved with Testing the Oxymoronic Potency of Language Articulation Programmes (TOPLAP), after being in the right bar (in Hamburg) at the right time (2 AM, 15 February 2004). He previously curated for *Leonardo Music Journal* and the *Swedish Journal of Berlin Hot Drink Outlets*.

### 1. *Overtone*—Sam Aaron

In this video Sam gives a fast-paced introduction to a number of key live-programming techniques such as triggering instruments, scheduling future events, and synthesizer design. Finally, the viewer is shown how a simple musical sequence may be composed and then converted into an intricate phase *à la* Steve Reich. The main body of the video was recorded in one take and features an Emacs buffer for editing text and communicating with Overtone, an expressive Clojure front-end to SuperCollider. Clojure is a state-of-the-art functional Lisp dialect emphasizing immutability and concurrency.

**Sam Aaron** (see Figure 1) is a researcher, software architect, and live programmer with a deep fascination surrounding the notion of communicative programming. He sees programming as a communication channel for descriptions of formalized processes of any kind, be it a business process, a compiler strategy, or even a musical composition. His previous research focused on the design of domain-specific languages in order to allow domain concepts to be communicated and transposed

*Figure 1. Sam Aaron.*

more effectively and efficiently. He has successfully applied these ideas and techniques in both industry and academia. Currently, Sam leads Improcess, a collaborative research project hosted by the Crucible Network for Research in Interdisciplinary Design at the University of Cambridge. The mission of the project is to explore the combination of powerful sound synthesis techniques with tactile and linguistic user interfaces to build new forms of musical device with a high capacity for improvisation.

### 2. *blind date*—Pd~graz

*blind date* is an audiovisual performance that aims at an artistic extension of common patterns in computer programming. Instead of featuring isolated programmers, several artists work concurrently on a single program (a Pure Data, or PD, patch), with all participants operating their own physical keyboard, mouse, and monitor. In earlier performances, these devices were all controlling the same logical interface. The players therefore had to coordinate their programming efforts at an immediate

Figure 2. Pd~graz.

level, because only one person could control the patch at any given point in time. In later performances, such as the one documented in the present video, IOhannes m zmölnig's Peer Data proxy provided each player with an independent logical interface. Nevertheless, the individual performers still need to participate in a common effort in order to produce a (technically and musically) functioning patch. Rather than merely representing a technical tool for generating music, the patch also becomes the primary means of communication between the players.

The performers start with a blank canvas (i.e., an empty patch) and gradually build up and modify a running program in the tradition of "live coding." Besides the resulting audio (and sometimes also video), the patches themselves are also projected into the performance environment. This offers the audience an insight into the programming and communication processes that occur among the players during the performance.

*blind date* has so far been presented at the following occasions: EarZoom Festival, Ljubljana, Slovenia (2009); International Computer Music Conference, Belfast, UK (2008, shown in the video); Second International Pd~Convention, Montreal, Canada (2007); Roxy/NoD gallery, Prague, Czech Republic (2006); Netart Community Congress, Graz, Austria (2005); and the Musikprotokoll Festival, Graz, Austria (2005).

The media art collective **Pd~graz** (see Figure 2) was founded in Graz, Austria, in 2005 and serves as an initiative for the organization of performances, installations, workshops, and publications around the Pure Data programming language. Pd~graz grew out of the Pd Stammtisch, a diverse group of independent artists and academic researchers in Graz, who have shared their fascination for Pure Data in regular meetings since 2003. This soon resulted in a number of group activities, such as the audiovisual installation *create/destroy* (2003) and a twelve-hour audiovisual live improvisation at the Lange Nacht der Musik (2003), both at the ESC Gallery in Graz. In 2004, the collective organized and hosted the First International Pd~Convention in Graz, with support from the ESC Gallery, the Mur.at net art initiative, the Institute of Electronic Music and Acoustics, and the Medienkunstlabor at the Kunsthaus Graz. In the aftermath of this event, Pd~graz was formally founded together with its publishing body, the Pd~ label. The first release was a DVD with artworks presented at the Convention (release 0.1, 2005). The book *bang: Pure Data* (ISBN 978-3-936000-37-5) was published in 2006 in collaboration with Wolke Verlag. It includes articles by developers, artists, and theorists who had participated in the Convention. Pd~graz conducts workshops on Pure Data on a regular basis, such as at the Film and TV School of the Academy for Performing Arts in Prague (2006), the CC in Graz (2007), and the EarZoom Festival in Ljubljana, Slovenia (2009). In recent years, the collective has performed its audiovisual group improvisations *blind date* and *rec.wie.m* on multiple occasions in Austria, Slovenia, Germany, the Czech Republic, the UK, and Canada.

People who have been associated with Pd~graz include (in alphabetical order): Lukas Gruber, Ypatios Grigoriadis, Reni Hofmüller, Florian Hollerweger, Georg Holzmann, Karin Koschell, Manuela Meier, Thomas Musil, Markus Noisternig, Renate Oblak, Michael Pinter, Peter Plessas, Nicole Pruckermayr, Winfried Ritsch, Romana Rust, Uwe Vollmann, Franz Xaver, Ales Zemene, Fränk Zimmer, and IOhannes m zmölnig. However, the line-up of actual participants varies from occasion to occasion. IOhannes m zmölnig and Florian

Hollerweger are the two players in the 2008 performance of *blind date* at the International Computer Music Conference in Belfast, which is shown in the video on the present DVD.

## 3. *King's Anatomy*—Andrew R. Brown

This live coding performance was performed at the Live Coding @ The Anatomy Museum concert at Kings College London in January 2010. It demonstrates the emergent combination of simultaneous computational processes that have distilled through research in computational musicology, and been published by the author in various peer-reviewed publications over several years. In this work, Andrew battles with the processes of succinct music representation as he performs with the Impromptu live coding environment. He deploys probability, recursion, and graph structures in an integrated offensive with his allies in music theory and acoustics in an improvised struggle with time and aesthetics. This and other live coding performances are a vehicle for experimentation of algorithmic and procedural processes and an ongoing struggle for computational expression of music.

**Andrew R. Brown** (see Figure 3) is an active computational artist working in music and visual domains. He is Professor of Digital Arts at the Queensland Conservatorium of Music, in Brisbane, Australia, where his work explores the aesthetics of process and often involves programming of software as part of the creative process. In addition to a history of computer-assisted composition and rendered animations Brown has, in recent years, focused on real-time art works using generative processes and musical live-coding where the software to generate a work is written as part of the performance. He has performed live coding around Australia and internationally, including in London, Copenhagen, and Boston. His digital artwork has been shown in galleries in Australia and China.

*Figure 3. Andrew R. Brown.*

## 4. *sdfsys2min*—Samuel Freeman

The video presented here demonstrates an alpha build of sdf.sys, which is a programmable soundmaking software system being developed within Max/MSP/Jitter. There is a basic text editor for script-based interaction; text is used both by the user to instruct the system and by the system to inform the user. The scripting syntax allows several types of command. First, there are geometry-based commands for specifying, and drawing with, points on the plane. There are also commands that manipulate digital signal processing (DSP) abstractions within the system. These abstractions either write to or read from Jitter matrices using MSP signals. Parameters of the DSP abstractions can be queried and set by text commands as well as being manipulable via graphic user interface (GUI) objects.

*Figure 4. Samuel Freeman.*

This system is being developed as part of my doctoral research which questions the ways in which sound is represented visually in computer music software; how do the ways in which elements of sound are represented affect the ways we choose to manipulate and organize those elements as music? As a composer the emphasis of my work is upon the aesthetics of the software in use and the musics one may produce with it.

**Samuel Freeman** (see Figure 4) makes things to make noise with, and then makes noises with them. These things are made both inside computers, where interactive systems are programmed mostly in Max/MSP/Jitter, and in the more physical realm, where electroacoustic contraptions are hacked together using recycled/recontextualized components. With an experimental approach to performance practice, Freeman regularly appears with Inclusive Improv, of which he is co-founder, and also plays laptop in HELOpg ensemble. Freeman is currently working on a PhD at the University of Huddersfield under the supervision of Michael Clarke and Monty Adkins, supported by the Arts and Humanities Research Council. His work is being documented on his Web site (sdfphd.net).

Figure 5. Graham Coleman.

Figure 6. Evan Hanson.

Figure 7. Mark Havryliv.

## 5. *Short Variations on a Quartet Theme from Ravel*—Graham Coleman

Adapting repertoire offers one possibility for interaction between live coding and traditional music genres. This is an attempt to riff on a classical music fragment that has long buried itself in my musical consciousness. Presented is a first glance at what hopefully will grow and evolve.

Two short phases defined the preparation for this piece. First, realizing the original score fragment in SuperCollider (SC). Second, reacquainting with the pattern and synthesis systems in SC to allow for a basic improvisation. A minuscule hack was introduced for tracking stream player references, as well as some synonyms for patterns.

Thanks to Alex, Thor, and Nick for providing an excuse to realize this. Much thanks to Stefan Kersten as well as members of SC-users for patient guidance in SC. Apologies to Ravel.

**Graham Coleman** (see Figure 5) occasionally live-codes cheesy counterpoint in ChucK and SuperCollider. In addition, he has presented tools for automatic and non-automatic sample composition at the ICMC and DAFx conferences. He is from Athens, Georgia, but is currently pursuing a PhD at the Music Technology Group of Pompeu Fabra University, Barcelona.

## 6. *Life*—Evan Hanson

This is a small demonstration of an idea I've explored recently: The control of sound by unrelated systems. By mapping a logical series of musical tones onto a simple but unpredictable automaton (in this case one of the most well-known, Conway's Game of Life), one can produce music that is at once structured and chaotic.

**Evan Hanson** (see Figure 6) is a recent graduate of the University of Wisconsin, Madison, where he currently works as a computer programmer. His musical pursuits are, like much of his code, driven by curiosity and a desire to create things that are both simple and compelling.

## 7. *The Waiting Room*—Mark Havryliv and Josh Mei-Ling Dubrau

This video represents a distillation of a networked live performance based on John Tranter's poem "The Waiting Room." This poem takes the form of a pantoum, an older Malaysian style of variable length consisting of a pattern of interwoven quatrains. It has a non-rhyming but very strict scheme in which the second and fourth lines of each stanza are replicated as the first and third in the stanza following.

Performance involves one or more computers running our software, P[a]ra[pra]xis, which is a front end for SuperCollider and a set of algorithms that substitute input text with other words. These algorithms play on the notion of the parapraxis, or Freudian slip, and employ a set of user-defined grammatical rules to govern phonemic, phonetic, and lexical substitutions.

As we type, text inputs and manipulations are categorized and reported to a sound-server in SuperCollider, which is programmed to route key/text/word/grammatical/substitution events and their data to synthesis and algorithmic control inputs. This results in a neat range of abstractions, flirting with classical live coding.


*Figure 8. Josh Mei-Ling Dubrau.*

The substitution engine is a weakly typed, late-binding, real-time interpreter; any typed key calls an overloaded function whose parameters are determined by surrounding keys, words, grammatical rules, and substitution rules, which gives the source code the ability to alter itself. This makes it an extremely difficult live-coding language for those wanting precise control over sonic output; it does, however, save on wear-and-tear of parenthesis keys.

**Josh Mei-Ling Dubrau** (b. 1975) (see Figure 8) is a poet and doctoral student at the University of New South Wales, Australia. Her thesis investigates the potential of psychoanalytic theory as a framework for the reading of modern and avant-garde poetry, and her creative work explores the use of multiple systems of delivering meaning within the poem.

**Mark Havryliv** (b. 1981) (see Figure 7) is a composer and doctoral student at the University of Wollongong, Australia. He is especially interested in the musical possibilities of integrating real-time sonification with other disciplines, like game design and creative writing. He looks forward to doing much

*Figure 9. Scott Hewitt.*

more of this once he completes his doctoral project, the Haptic Carillon.

## 8. *Chuck CMJ Quarks*—Scott Hewitt

*Chuck CMJ Quarks* is the title of this screen capture of Scott's daily performance practice. Using just a basic single function with a built-in control loop, the single impulse is reverberated and then spread across the four wide channels of the typical 5.1 DVD setup. The recording aims to illustrate how rich a texture can be built by using and reusing just a simple section of code, the use of multichannel ChucK, and finally, the power of the command line for performance.

**Scott Hewitt** (see Figure 9) is a PhD candidate at Huddersfield University and the Director of the Huddersfield Experimental Laptop Orchestra (HELO). He also curates a Web site on live coding (www.livecoding.co.uk).

## 9. *Do sinusoids dream of electr(on)ic sweeps*—IOhannes m zmölnig

*Do sinusoids dream of electr(on)ic sweeps* is a live-coding performance done in Pure Data. It relies heavily on self-modifying PD patches that interact in an agent-like fashion. The acoustic output is somewhat limited to sine waves. The

*Figure 10. IOhannes m zmölnig.*

visual output is a PD patch going crazy.

**IOhannes m zmölnig** (see Figure 10) is a long-time user of Pure Data. Since 2003 he uses his chosen environment in live coding performances, mainly generating sine waves.

"I always preferred building sequencers to building sequences."

## 10. *Quoth*—Craig Latta

Quoth is a dynamic interactive fiction system I wrote as an experiment in executable natural language. I use it for musical live coding, to make those performances more accessible to audiences. Instead of using a typical programming language and development environment, each of which tend to appear cryptic to the uninitiated, I use English in the simple conversational format of the "text adventure."

Writing code live for an audience presents serious challenges. The most important is interactivity; I need to get results quickly. In a musical performance, continuity is also crucial. Because I work with long-lived musical structures and

Figure 11. Craig Latta.

Figure 12. Julie Dassaud (photograph by Andreas Maria Jacobs).

Figure 13. Eliad Wagner (photograph by Olivia Wagner).

processes, I need to change the system as it runs. The Squeak Smalltalk system underlying Quoth provides these features, and the language is effectively a superset of English.

This video gives a short demonstration of Quoth's conversational interface. Several tactics for increasing interaction speed appear, including anthropomorphization and phrase completion. The system also attempts to create a more readable transcript of past actions by rephrasing what the performer types. The musical objects conversing with the performer are simple MIDI note events and their component parts.

For more information about Quoth, please visit the project Web site (netjam.org/quoth).

**Craig Latta** (see Figure 11) studied music and computer science at the University of California, Berkeley. He discovered Smalltalk programming at the Center for New Music and Audio Technologies, and was inspired by the musical implications of its improvisational development style. A composer and research computer scientist, he is active in the improvised music communities of San Francisco and Amsterdam.

## 11. *Lemuriformes (excerpts)*—lemuriformes (Julie Dassaud, Eliad Wagner, Roel van Dooen, Laurens van der Wee)

*lemuriformes* is a collaborative improvisational performance project in which live coding, painting, and electronic music blend together in one experience. Many issues are addressed in the process, such as the role of the graphical representation of code in live coding performances, live coding as a means of sound transformation (rather than synthesis), human and machine involvement levels in electronic music, and live coding in collaborative performances.

The main reason to start this unusual cooperation was to share a positive and playful experience in which the personal qualities of the collaborators have their roles defined on the fly. From it we have learned that it definitely creates an interesting combination of crosshatched and intersecting auditory and visual palettes, which we will continue to explore.

lemuriformes includes: Julie Dassaud, ink; Eliad Wagner, synthesizer; Roel van Doorn, circuits; Laurens van der Wee, coding.

**Julie Dassaud** (see Figure 12) is a French visual artist, based in Amsterdam, who is involved with various solo and collaborative projects, wherein she explores in particular the relevancy and possibilities of drawing in our contemporary digital age. In this context, she exhibits, for instance, with Img-src, a collective that confuses the viewer's senses by producing analog work that looks digital and the other way around. Her live ink-painting contribution to lemuriformes is her first performing appearance. Besides this, Julie is active as artistic director of the Kulter art space and of the Notations platform for alternative music notations (www.julisso.org/ and www.notations.nl).

**Eliad Wagner** (see Figure 13) is a musician, composer, and programmer. Born in Israel and currently based in Utrecht, The Netherlands, he holds a Bachelors degree in physics and is completing his Master of Music degree at the Music Technology Department of the Utrecht School of the

*Figure 14. Roel van Doorn (photograph by Olivia Wagner).*

Arts. He develops his own approach to electronic sound, mainly based on analog and modular synthesizers and computer programming. His work has been published by Metropolis records (USA), Digital Kranky (Germany), C-sides (Germany) and concrete plastic (UK). He is co-founder of the electronica label ±g6pd records (Israel) (eliadwagner.wordpress.com).

**Roel van Doorn** (see Figure 14) is a multimedia artist living in Rotterdam. At first, his works were focused on sound, but recently this has slowly shifted to audiovisual projects in which he is involved in both the sonic and visual aspects of performances. For Vanilla Riot he programs interactive visual software, and improvises with his visual software as a fourth member of the group. This way, sound and visuals are intertwined in a meaningful way. For his graduation project he designed and produced several sound installations that created sound using solar and wind power that were exhibited during the Duizel festival in Rotterdam in 2009 (www.roelvandoorn.com).

*Figure 15. Laurens van der Wee (photograph by Evelien van Zonneveld).*

**Laurens van der Wee** (see Figure 15) is a sonic designer, composer, and programmer from The Netherlands, currently living in Vilnius, Lithuania. In June 2011 he will obtain his Master of Music degree at the Music Technology Department of the Utrecht School of the Arts. Laurens focuses on autonomous works on the edge of composition and performance, as well as usual and less usual collaborations, ranging from modern dance to live coding improvisation (lemuriformes). His work has been presented in Hong Kong, Canada, USA, Portugal, Spain, and several other countries, at occasions such as ICMC, SMC, and the Hong Kong International Dance Symposium (www.laurensvanderwee.nl).

## 12. *Bal des Ardents*—Benoît and the Mandelbrots

*Bal des Ardents* was performed at the Studiokonzert of the University of Music Karlsruhe at the ZKM_Kubus on 28 January 2011. The piece is inspired by the 618th anniversary of the Bal des Ardents [Ball of the Burning Men]: In 1393 King Charles VI of France threw a grand party to celebrate the wedding of one of the queen's ladies-in-waiting. As a fire broke out, four of his friends were killed. After this incident the King, who had been suffering from mental health issues all his life, went

*Figure 16. Benoît and the Mandelbrots.*

irrevocably mad. In its original length (16 min), the piece starts with the simple development of a ternary rhythm using the Dorian scale with Pythagorean tuning. These decisions were made just before the concert, and were expanded through improvisation. Some cues were planned to reflect the course of the historical event. These cues and other spontaneous directions are communicated by text messages, gestures, facial expressions, and the musical flow itself. The musical process is bound to the coding process, which is performed live using SuperCollider. The complexity of the music increases with the complexity of the code. The course of this historical event is reflected in the transition from melodies and rhythms, inspired by early music, to noise and more abstract sounds. The 6-minute excerpt on this DVD starts at 5′44″ of the original recording.

**Benoît and the Mandelbrots** (see Figure 16) was formed in late 2009 as a laptop band at the ComputerStudio of the University of Music Karlsruhe. Matthias Schneiderbanger and Holger Ballweg joined Patrick Borgeat and Juan A. Romero, who were members of the recently disbanded laptop ensemble Grainface. All members were students at the Institute for Musicology and Music Informatics at that time. This new constellation focuses on live programming and

Figure 17. Davy Smith and Louis McCallum.

Figure 18. No Copy Paste.

improvisation instead of using controllers and pre-composing the music for the performances. The band performs frequently in the Karlsruhe area in a wide variety of venues, from galleries and concert halls to cinemas and bars.

The creative process of the band is influenced by the respective events and venues. Due to MandelClock, a self-developed system based on OSC, part of the open source BenoitLib, the members are always beat-synced and able to communicate with each other. The main intention of the band is to show and establish the laptop as a musical instrument by playing a wide variety of musical genres, including techno, noise, ambient, and experimental avant-garde (www.the-mandelbrots.de).

## 13. *Show us Your Screens*—Davy Smith and Louis McCallum

*Show us Your Screens* is a documentary that provides an introduction to live coding. Established practitioners explain their motivations and methods alongside audience members who have varying degrees of past exposure.

**Davy Smith** (see Figure 17) is a sculptor by trade, using computer vision techniques to explore interactivity in his work.

**Louis McCallum** is a musician, researcher and rookie roboticist. Both are PhD students from Queen Mary's College, University of London, as part of the Media and Arts Technology program. Their current research interests are in mixed reality and electromechanical sound, respectively.

## 14. *Fourfold*—No Copy Paste

The sound is generated using the PD real-time graphical dataflow programming environment. The visuals are created using Fluxus, which is a rapid-prototyping, 3-D graphics, and live-coding environment. The performance mixes live coding with augmented reality (AR) technologies. It starts up with blank slate live coding, then the programmed animation is mixed with the camera feed using AR markers. The location of the AR markers control the sound parameters, their relative spatial location determining how the various audio modules are created and connected. This type of dynamic system makes real-time manual patching possible. *Fourfold* was premiered at the Make Art Festival 2009, Poitiers, France.

**No Copy Paste** (see Figure 18) is a live-coding duo consisting of Agoston Nagy and Gabor Papp. They mix the expressive possibilities of programming languages with computer vision, mathematical models, game controllers, broken rhythms, and raw synthesized sounds. Agoston works with sound in traditional and experimental ways, and Gabor is interested in the aesthetic implications of software. They create dynamic systems for installations and build interfaces for audiovisual live performances by using and developing free and open source tools (ncp.kibu.hu).

## 15. *Less than a Minute for CMJ (v4)*—Miquel Parera Jaques

My main motivation is to reduce the perceptual space to get the sound and its processes as relevant as possible.

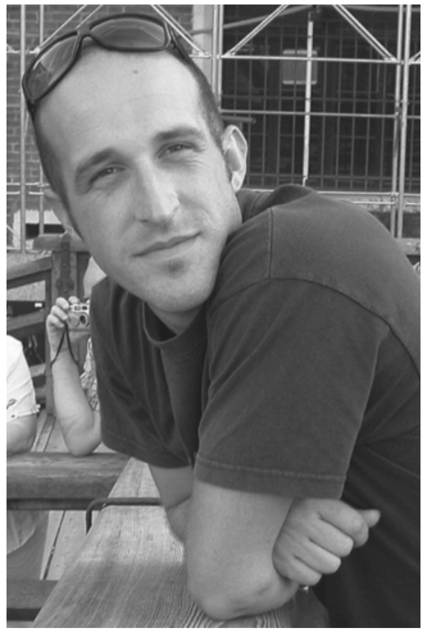
**Miquel Parera Jaques** (see Figure 19) is based in Barcelona.

## 16. *Untitled 12*—Michele Pasin

*Untitled 12* is an investigation of how concise, recursive, and iterative

Figure 19. Miquel Parera Jaques.

Figure 20. Michele Pasin.



procedures can be used to create interesting musical results. Through the use of algorithms that pick random pitches within a deterministic matrix of possibilities, the piece aims at creating a crescendo of musical structures that vary unexpectedly but fundamentally remain within a single, broad musical space. This creates a double effect on the audience. In fact, the listener/viewer is almost subconsciously trying to bring "cognitive" order to the composition (e.g., by reading the code projected on the screens and trying to figure out the model behind the repeating musical patterns), but is in the end always and irremediably surprised by the emergence of random sounds.

This recording of *Untitled 12* was made in January 2010 at King's College Anatomy Museum (London). *Untitled 12* was composed and performed using Andrew Sorensen's Impromptu live-coding environment. This is a freely available software for Mac OS X that allows real-time creation of musical procedures using the Scheme programming language. One of the key features of Impromptu is that it interfaces with Apple Audio Units API, thus allowing the employment of any third-party virtual instrument in a composition. For example, in *Untitled 12* Pasin is making use of U-He's Zebra and Native Instruments's Battery audio units.

**Michele Pasin** (see Figure 20) is an Italian musician and digital creative currently working as a research associate at London's King's College. He graduated in Logic and Epistemology at the University of Venice. In 2004 he moved to the UK and did a PhD in artificial intelligence, focusing on the application of knowledge representation and semantic technologies to humanistic domains. Despite not being his primary academic subject, music has been a constant research interest throughout the years. He has a degree in music theory from Trieste's Conservatory G. Tartini. He then extensively studied classical guitar for a number of years before starting experimenting with less traditional approaches and musical styles, ranging from minimalist electronica to progressive and psychedelic rock. More recently, he discovered the world of algorithmic composition, which presented him with a chance to bring together two of his passions, artificial intelligence programming and music. Currently, while at King's College, he organizes live-coding events and explores with curiosity the musical affordances of algorithmic abstractions.

## 17. *Movement 1 – Pärt*—Alex Ruthmann

*Movement 1 – Pärt* (2010) is an excerpt from a suite of live-coding pieces, *Scratch Etudes*, exploring the live, interactive coding capabilities of the Scratch visual programming environment (scratch.mit.edu). Taking inspiration from the musical organization of Arvo Pärt's *Stabat Mater* and a live coding performance by Andrew Sorensen within his Impromptu software, *Movement 1 – Pärt* utilizes live manipulation of visual code chunks, blocks, lists, and variables through mouse and keyboard control in a creative exploration of the Aeolian mode. An additional minor pentatonic solo layer was performed live over the drone using an IchiBoard sensor interface (bit.ly/ichiboard) developed by Mark Sherman in the Engaging Computing Group at University of Massachusetts, Lowell. The IchiBoard enables melodic and rhythmic performance of the solo line through a button and linear potentiometer with volume controlled by the *z*-axis of the built-in accelerometer.

*Scratch Etudes* was conceived as a set of live coding examples to share with students enrolled in an undergraduate general education course, "Sound Thinking," offered at the University of Massachusetts, Lowell, and for use in workshops with

Figure 21. Alex Ruthmann.

Figure 22. Ben Swift.

middle- and high-school students in computational music. Originally developed for use by children by the Lifelong Kindergarten Group at the MIT Media Lab, Scratch has proven useful as a platform for engaging children in creating computational music and live coding in specific.

**Alex Ruthmann** (see Figure 21) is Assistant Professor of Music Education at the University of Massachusetts, Lowell, where he teaches coursework at the intersection of music, computing, and learning. After graduating from the Performing Arts Technology program at the University of Michigan, he pursued masters and doctoral work in music education at Oakland University. Currently, he is an active collaborator on a National Science Foundation–funded Performamatics project linking computer science with the fine, design, and performing arts and serves as a development consultant on several music education technology projects with companies and research teams in Australia, Norway, and the USA. His research centers around the design, creation, and study of technologies and environments that promote creative music-making, learning, and teaching.

## 18. *Impish Grooves*—Ben Swift

Building something meaningful in three minutes of live coding is a significant challenge. *Impish Grooves* is an exploration in procedural rhythm generation. By reusing code fragments while tweaking parameters, it is possible to layer different percussive parts over one another and quickly produce a polyrhythmic texture. Once the rhythmic pulse is established, its components are available for manipulation, allowing the coder to guide the dynamics of the piece while interacting with the code.

**Ben Swift** (see Figure 22) is a Computer Science PhD student based in Canberra, Australia. His current creative practice revolves around live coding in Impromptu/Extempore. Having initially studied both music and mathematics, Ben is currently trying to convince the Long Beards at Australian National University that his "touchy-feely" research constitutes *real* computer science. Videos of Ben's live coding can be found at vimeo.com/videos/benswift/.

## 19. *Live Writing with the Rumentarium*—Andrea Valle

What does "to play a computer" mean? Actually, for me, playing has always been mapping real-time gesture to sound, that is, involving that specific aspect of cognitive processes known as muscle memory. Indeed, there is a peculiar tension between the hypercognitivization of code writing in live coding on one side, and the the deep embodying of motor learning involved in instrumental playing on the other. But typing is also a specific form of gestural action. As shown by many studies in human–computer interaction, it is possible to achieve very fast typing, that can be compared to percussion playing rates. So, the idea at the basis of my approach is to sonify the typing process by defining different mapping strategies from keys to sound. In this way, typing gestures trigger sounds: it is like, literally, playing a keyboard. The best strategy in order to explore muscle memory is not to use arbitrary symbols, but to exploit the writing process of ordinary languages (to which we have been exposed since childhood). In this way, much faster rates can be reached. This also means that the input data will not be neutral, rather showing a specific structure, depending on the sentence/word structure but also on graphemic patterns that are typical of each language (one can think, for example,

Figure 23. Andrea Valle.

about the difference between Italian and English alphabets and writing systems). Poetry is particularly apt to be typed, as it typically shows complex patterns based on variation/repetition. But, indeed, the text to be typed can be code itself: in this way, the sonification of writing is intermingled with live coding, leading to a form of live code writing, so to say. In the video, the live writing approach is used to drive the Rumentarium, my computationally-controlled, electromechanical percussion ensemble. Electronic sound is added, mapping keys to pitches and including processed sound from the Rumentarium and the typing action. The whole software system is implemented in SuperCollider.

**Andrea Valle** (see Figure 23) (www.cirma.unito.it/andrea), an electric bass player interested in experimental rock and in free jazz, studied music composition with Alessandro Ruo Rui, Azio Corghi, and Mauro Bonifacio while attending master classes by Trevor Wishart and Marco Stroppa. His work as a composer is mainly focused on algorithmic methodologies, in both the electroacoustic and instrumental domains. When composing for acoustic instruments, he is interested in developing compositional methodologies for automatic notation generation and he has participated in the *Notation 21* project (Theresa Sauer, ed., New York: Mark Batty Publisher, 2009). His compositions have been performed at Logos Foundation and commissioned by OSN Rai of Torino. He has worked on multimedia installations, film music (*La fôret rouge*, by Michela Franzoso, a project hosted by Le Fresnoy, 2008; *Rohbauten*, by Eva Sauer, 2009), and, recently, theater (*Cotrone*, by Marcel.lì Antuñez Roca, 2010). His most recent major project is the Rumentarium, an electromechanical, computer-driven, percussion ensemble made of recycled materials.

Andrea is a member of the core unit of AMP2, a collective devoted to free improvisation, and he appears on the album *Hopeful Monster*, issued by Die Schachtel in the Musica Improvvisa box set (2010). He earned a PhD in Semiotics at the University of Bologna and he is currently researcher at the University of Torino, where he is a founding member of CIRMA (Inter-departmental Centre for Multimedia and Audiovisual). He participated in the VEP project, which reconstructed the *poème électronique* in virtual reality. He is a member of the Italian Semiotic Association and of the Italian Music Informatics Association.

## 20. *Dupin's Spaceship*—Graham Wakefield and Wesley Smith

The visual forms are generated from a shape called a Dupin Cyclide, made by inverting a torus through a sphere.

Figure 24. Graham Wakefield.

The sounds are produced by two live-coding performances (played side by side, overlaid on the video). *Dupin's Spaceship* was written and performed using LuaAV.

**Graham Wakefield** (see Figure 24) is a composer of time-based media, and co-author of the LuaAV real-time audiovisual scripting environment. Graham is approaching the end of his PhD at the Media Arts and Technology program at the University of California, Santa Barbara, where he also works as a researcher for the AlloSphere immersive instrument. He is also a developer for Cycling '74.

**Wesley Smith** (see Figure 25) is a computational designer living and working in San Francisco. He is a co-author of the LuaAV real-time audiovisual scripting environment. Wesley is currently pursuing his PhD at University of California, Santa Barbara's Media Arts and Technology program, and works for Cycling '74.

## 21. *Improvisation*—Matthew Yee-King (Live Coding), Finn Peters (Alto Flute)

This is an edited version of a 9-min improvisation recorded on 22 March 2011 at the Goldsmiths Digital Studios, Goldsmiths College, London. Finn Peters plays alto flute, and Matthew Yee-King live-codes. The

Figure 25. Wesley Smith.

programming language is Scaling version 3.3, and the live-coding is carried out in the Emacs text editor under Ubuntu Linux. Four improvisations were recorded during the session, and this piece was chosen as it had the most successful overall structure, allowing it to function as a standalone piece. The other pieces were more in a free improvisation style, featuring less constrained pitch and rhythm sequences. This was the first time Matthew and Finn had played together with the constraints of using an unaffected instrument and from-scratch live-coding. The experience was inspiring, but it appears that a specialized environment for live-coding interactive music systems might need to be developed to improve the standard of the results, or that more practice is required.

**Matthew Yee-King** (see Figure 26) is a lecturer in creative computing at Goldsmiths College as well as a computer music composer, performer, and researcher. His research interests include automated sound synthesizer

Figure 26. Matthew Yee-King and Finn Peters.

programming and interactive music systems. Recent musical activities include development of the technique and practice of live coding in the UK with the TOPLAP collective, and extensive involvement in a brain-wave music project "Music of the Mind" alongside composer Finn Peters. He has performed live internationally and nationally and has recorded many sessions for BBC Radio. His solo music has been released on electronic music imprints such as Warp Records and Richard James's Rephlex Records. Collaborators include Jamie Lidell, Tom Jenkinson (Squarepusher), Finn Peters, and Max de Wardener (www.yeeking.net).

**Finn Peters** is a flautist and saxophonist who studied music at Durham University and took the postgraduate jazz course at Guildhall School of Music. He is a member of the F-IRE Collective and the contemporary classical music group Noszferatu. He has also worked with Two Banks of Four and Matthew Herbert and recorded under the names Bansuri and Finntech. In September 2006, he released his first album, *Suling*, under his own name, having recorded this album with a band that included guitarist Dave Okumu, pianist Nick Ramm, bassist Tom Herbert, and drummer Tom Skinner. It was selected as a Jazzwise album of the year for 2006. In July 2007, the Finn Peters Quintet (or Finntet) beat the competitors in the

Figure 27. Renick Bell.

best jazz group category of the BBC Radio 3 Jazz Awards. In 2008 he initiated the Music of the Mind project, working closely with Matthew Yee-King.

## 22. *Percussion Improvisation*—Renick Bell

This rhythmic three-stage improvisation served as practice with Conductive, a live-coding library for Haskell written by Renick Bell (see Figure 27). As a musical foundation, a 130-BPM electronic dance music style was chosen to allow a clearer judgment of how much live control could be achieved with this environment. Seventy-eight percussion samples are played according to algorithmically composed patterns that were generated shortly before the video begins. The performance consists of executing functions that specify which patterns should be played by selected samples. The functions are sent for evaluation from the Vim text editor (at the top of the screen) to the Glasgow Haskell Compiler interpreter (at the bottom of the screen). The cursor position serves as a hint to which functions

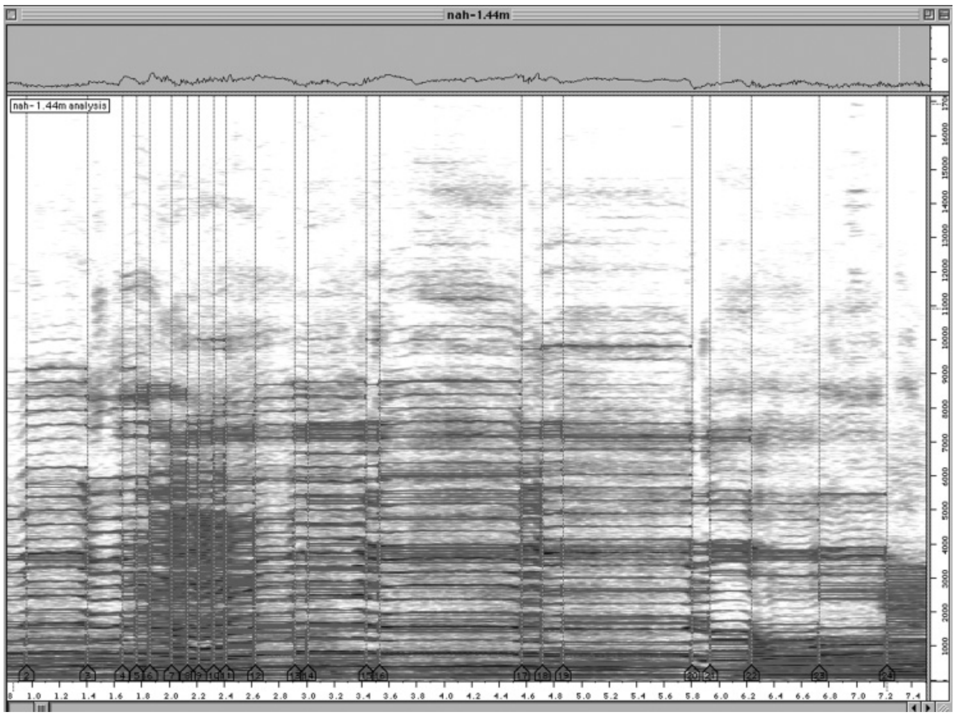



*Figure 28. Sonogram 1.*

are being sent to the interpreter, while the interpreter displays the functions themselves, status messages, and debugging information. This practice revealed the need for even-higher-level functions and regular practice with this new instrument to achieve more-spontaneous, fast-moving improvisations. The need for less-verbose debugging information and more persistently displayed status information also became clear.

**Renick Bell** is a researcher and composer based in Tokyo. He focuses his research on computer systems for live performance. Information on the tools he has developed and his music can be found at his Web site (renickbell.net). An American, he holds an interdisciplinary undergraduate degree from Texas Tech University, where he studied electronic music with Steven Paxton. He also holds a Master of Science degree in Music Technology from Indiana University. He was a doctoral student in Information Systems and Multimedia Design at Naotoshi Osaka's Sound Media Representation Laboratory at Tokyo Denki University.

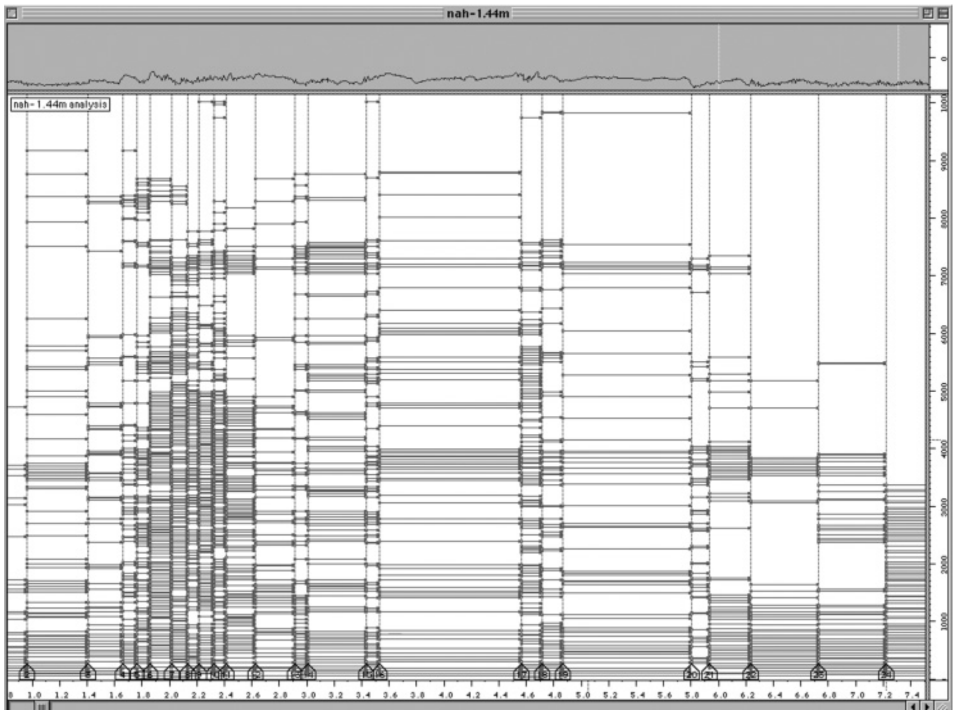
## Part Two: Video and Sound Examples

This section of the disc features video and sound examples to accompany articles appearing in Volume 35 of the *Journal*. Where examples contain more than one element in succession, each individual element has been encoded as a separate chapter, so one may navigate forward and backward through the examples using the Next and Previous Chapter buttons on any DVD player or remote control. Alternatively, the examples will automatically play in sequence with a short pause between each.

### 1. Sound Examples to Accompany the Article "OMChroma: Compositional Control of Sound Synthesis" by Carlos Agon, Jean Bresson, and Marco Stroppa (Volume 35, Number 2)

All examples come from Marco Stroppa's composition *Come Natura di Foglia*, where they are used in the electronic part starting at 6′54″. They were chosen because they constitute an exhaustive overview of the main concepts delved into in the article.

*Figure 29. Sonogram 2.*




1. Original sound: *nahhami*, song from pearl fishermen of Bahrain (Arabian Peninsula).

## Analytical Procedure

a. Sonogram analysis of the sound (FFT Size: 4,096 points; black at –60 dB, white at 0 dB);
b. Discrete temporal segmentation of the sonogram (shown on the disc), according to some compositional principles, based on placing a marker wherever a musically important change is located (although the main notes are marked, there are more markers than notes);
c. Computation of a static spectrum between each group of two adjacent markers. The spectrum contains the partials that are above a certain amplitude and last through the whole time segment (see Figure 28);
d. Construction of a *chroma model*. Each spectrum instantiates a *chroma matrix* containing the following information: as many components as frequencies, one amplitude and frequency per component. Each matrix has a duration corresponding to the inter-onset interval (IOI) between two adjacent markers and a global onset time corresponding to the value of the marker (see Figure 29);
e. Processing and synthesis of the *chroma model*. The synthesis class is chosen; the missing parameters required by that class are given according to a set of compositional rules.

## Synthesis Results

The main compositional issue being the exploration and development of a cognitive spectromorphology of the original sound, an exact resynthesis was not sought for. By *cognitive* we mean the ability to recognize the salient features of the original sound (if known) or of its synthetic development, whereas *spectromorphology* is the control of the characteristics and evolution of the sound over time. It is worth noting that at least two levels of time are used: each matrix has both a position in the sequence of the onsets of the chroma model, as well as its own internal duration. This allows each matrix to end before or overlap with other matrices.

## General Control Strategy

For examples 2–5 a subset of markers was selected (1 2 3 7 10 14 16 19 21 22

23 24 25). The duration of each matrix was stretched across the IOI, so that each event is partially superposed on the next one (to simulate a "legato" effect). The amount of superposition can vary across a model and is determined by a scalar factor computed by looking up a break-point function. Further data for the matrix were computed algorithmically (using Lisp functions).

2. Relatively close additive synthesis of the original sound (similar rhythm, no change of the spectromorphological information coming from the model), with approximately 2–3 side-components per real component. The stereo pan goes from left to right from the beginning to the end of the model. The sonogram (shown on the disc) is of the mono reduction. The overlap between each matrix, the additive structure, as well as the small entry delays of the partials of each matrix, are clearly visible.
3. Same as above, but using formant frequency modulation (one modulating, tuned to the lowest frequency in the model, and several carrier waves, tuned to the upper partials of the model, but adjusted so as to produce a relatively harmonic spectrum). This means that the N1/N2 ratio consists of small integer numbers, plus a little amount of "detuning" added to the computation of the N2. The sonogram (shown on the disc) is of the mono reduction. The typical formant structure and the opening of the formant due to the increase of the modulation index are very apparent.
4. Same spectromorphological profile as Example 2, with the following changes: pan goes from right to left, there are 10–20 sub-components for each partial (it sounds more "clustery"), the spectral amplitude and duration of each partial start as in Example 2 and are progressively reversed, which produces a final chord with the higher frequencies being the longest and loudest. In the sonogram (shown on the disc), the thickness of each partial is clearly visible (compare, for instance, the spectrum at 4.4").
5. Spectral profile similar to Example 2, but with a longer phrase and slower tempo. It starts with the same pitch and develops towards other pitches (shown in the sonogram, on the disc).
6. Same spectral profile as Example 2, with exponential accelerando and a bell-like percussive attack and decay (shown in the sonogram, on the disc).
7. Same profile as Example 6, but with very large clusters (20–100 sub-components per partial), yielding the effect of a cymbal-like sound (shown in the sonogram, on the disc).
8. *Come Natura di Foglia, Canti lontani per voci ed elettronica* [Faraway Songs for Voices and Electronics]—Marco Stroppa. Texts: an ancient prophecy by the Cree Indians and sacred appeals and reports from shamans, translated into English. Performers: Electric Phoenix (Judith Rees, soprano; Meriel Dickinson, mezzo-soprano; Daryl Runswick, tenor; Terry Edwards, bass; John Whiting and Mike Skeet, sound projection). Computer music design: Serge Lemouton. Electronic production: Institut de Recherche et Coordination Acoustique/Musique (IRCAM). Commissioned by Françoise and Jean-Philippe Billarant for IRCAM. This live recording was produced at IRCAM in 1997. It is used here with kind permission of Electric Phoenix. Note: this version of the piece is currently withdrawn. A major revision is being planned.

> The Real People don't think the voice was designed for talking. You do that with your heart/head center. If the voice is used for speech, one tends to get into small, unnecessary, and less spiritual conversation. The voice is made for singing, for celebration, and for healing. (Marlo Morgan, *Mutant Message Down Under*)

*Come Natura di Foglia* was born from a challenge I made to myself: How to bring together my own musical experience and models coming from several distant traditions, such as Tibetan voices, the Bunun choir, a song of a fisherman from Bahrein, or a toaca, a Romanian wooden drum. The relationship should remain understandable, but without any direct quotation of the original sounds. I was fascinated by the deep relationship between the people's culture, the social function of the music, and its insertion into a universal framework governed by the laws of Nature and the Cosmos. I have sought for a musical path that would link these different models and delve into the poetic and emotional dimension of computer-generated sounds, with the hope to carve a sort of "magic" event out of every sound.

The title comes from one of the *Meditations* written during the last ten years of his life by Marcus Aurelius, the great Roman emperor, philosopher, and humanist, who enlightened with his wisdom a decaying empire (Marco Stroppa).

## 2. Sound Examples to Accompany the Article "Parametric Electric Guitar Synthesis" by Niklas Lindroos, Henri Penttinen, and Vesa Välimäki (Volume 35, Number 3)

A complete electric guitar synthesis algorithm based on the digital waveguide approach was developed incorporating novel techniques and other appropriate improvements. The excitation signal produced by plucking a steel string with a plectrum was measured using a piezoelectric pickup. A parametric excitation model consisting of two parts was then proposed: The first part is a filtered noise burst and the second is composed of a parametric simplified pulse that is reproduced with an integrating filter.

The proposed magnetic pickup model is founded on a special feedforward comb filter in which a frequency-dependent delay is implemented using an all-pass filter. The frequency-dependent delay filter is needed to simulate the dispersive wave propagation on the vibrating steel string, which causes the notches of the comb filter to be nonuniformly spaced in frequency.

In addition to these novelties, an improvement to the waveguide string model was introduced: A time-varying loop gain helps to emulate the two-stage decay of electric guitar tones. The gain variation over time is easily calibrated by smoothing a short-term averaged envelope of the first harmonic. The proposed synthesis model also accounts for inharmonicity and beating appearing in electric guitar tones.

Recorded Sound:

1. Recorded guitar pluck: String: 6; Fret: 0; Plucking point: 10 cm; Pickup: Bridge; Level: *mf*.

Synthesized Examples:

2. Synthesized guitar pluck: String: 6; Fret: 0; Plucking point: 10 cm; Pickup: Bridge; Level: *mf*.

Changing Pickups:

3. Synthesized guitar chord: Open E major; Pickup: Bridge.
4. Synthesized guitar chord: Open E major; Pickup: Middle.
5. Synthesized guitar chord: Open E major; Pickup: Neck.

Changing plucking point:

6. Changing plucking point of a synthesized guitar tone: Plucking point moves from 2.6 cm to 32.6 cm on the open sixth string.

Altering dynamics:

7. Altering the dynamics of a synthesized guitar tone: Plucking force from *pp* to *ff* in five steps. The amplitude is normalized, therefore, the pitch drift is the most prominent effect.

Synthetic chords without distortion and with distortion:

8. A synthetic perfect fifth chord without distortion.
9. A synthetic power chord, i.e., a perfect fifth chord with distortion.

Synthetic Riff Without Distortion and With Distortion:

10. A synthetic riff without distortion.
11. A synthetic riff with distortion.

## 3. Sound Examples to Accompany the Article "Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition" by Christopher Ariza (Volume 35, Number 3)

1. In 1955 David Caplin, working at the Koninklijke/Shell-Laboratorium in Amsterdam, made a recording of the Ferranti Mark I* performing a program to generate and synthesize melodic lines from contredances, based on a version of W. A. Mozart's *Musikalisches Würfelspiel*. The synthesis technique, as suggested by Alan Turing in 1951, used the integrated computer loudspeaker and the "hoot" programming instruction. This synthesis system was implemented by Dietrich Prinz. The recording was made by holding a microphone near the computer's loudspeaker and recording to a reel-to-reel analog recorder. The occasional high-frequency noises are a result of the random signal generation; the occasional stutters in the sound are due to the time necessary to transfer data from the magnetic drum storage to the fast access storage. This recording was transferred to analog cassette in the late 1990s and digitized in 2009. No noise reduction or other audio processes have been applied.
2. In 1959 Caplin commissioned Elizabeth Innes, a programmer in the Shell Computer Development Division, to

rewrite the 1955 Mozart Dice Game System and the synthesis routine for the Ferranti Mercury, a significantly faster computer than the Ferranti Mark I*. The output of this computer was recorded using a method similar to that of Example 1. This recording contains frequent clicks due to the starting and stopping of the recorder. This recording was similarly transferred to analog cassette in the late 1990s and digitized in 2009, and no noise reduction or other audio processes have been applied.

3. In 1964 Sister Harriet Padberg, then studying at Saint Louis University, submitted her dissertation, *Computer-Composed Canon and Free Fugue*, with complete score tables defining pitch and duration for five computer-generated, microtonal, polyphonic compositions. The works were the result of an original system that, based on a text string input, created canons and fugues with multiple pitch and rhythmic transformations and a non-equal-tempered, 24-tone, microtonal scale. Although she used the IBM 1620 and IBM 7072 to generate these works, she did not have access to sonic realizations.

   This example is a canon, the fourth composition Padberg provides. The first three compositions all use the same source text as input ("college canon") and thus begin with the same primary voice. This example is unique, and perhaps more musically compelling, in that it uses four voices, rather than two.

   These realizations were made by transcribing the printed score tables by hand into a tab-delimited data table. This table was then processed by a Python script to provide a Csound CSD file. For a clear presentation of the pitch and rhythm structures with a harp-like tone, a simple Csound instrument using the `wgpluck` opcode is used with a quickly attacked envelope. Small amounts of reverb (via the `freeverb` opcode) and stereo panning are employed to support voice separation.

4. This example is the fifth composition Padberg provides, and is the only example of a free fugue. The source input text is "university canon and twentieth century fugue." This example was realized in the same manner as in Example 3.

## 4. Video Examples to Accompany the Article "Virtual Gesture Control and Synthesis of Music Performances: Qualitative Evaluation of Synthesized Timpani Exercises" by Alexandre Bouënard et al. (Volume 35, Number 3)

### *Validation Exercises*

1. Attack Modes. Simulations showing each attack mode independently, at one-third impact location.
   a. Legato: sequence of legato attacks, played at 63 bpm, composed of six two-handed beats, finishing with a right-hand beat.
   b. Tenuto: sequence of tenuto attacks, played at 63 bpm, composed of six two-handed beats, finishing with a right-hand beat.
   c. Accent: sequence of accent attacks, played at 63 bpm, composed of six two-handed beats, finishing with a right-hand beat.
   d. Vertical accent: sequence of vertical accent attacks, played at 63 bpm, composed of six two-handed beats, finishing with a right-hand beat.
   e. Staccato: sequence of staccato attacks, played at 63 bpm, composed of six two-handed beats, finishing with a right-hand beat.
2. Impact Locations. Simulations showing each impact location independently, with legato attacks.
   a. One-third: sequence of impacts at the one-third location, played at 63 bpm, composed of six two-hand one-third impacts, finishing with a right-hand one-third impact.
   b. Center: sequence of impacts at the center location, played at 63 bpm, composed of six two-hand one-third impacts, finishing with a right-hand one-third impact.

c. Rim: sequence of impacts at the rim location, played at 63 bpm, composed of six two-hand one-third impacts, finishing with a right-hand one-third impact.

*Extrapolation Exercises*

3. Attack Modes.
   Simulations showing mixed sequences of attack modes, at one-third impact locations.
   a. Exercise 1: sequence played at 63 bpm, composed alternatively of staccato, legato, tenuto, legato, accent, legato, vertical accent, and legato attacks.
   b. Exercise 2: sequence played at 63 bpm, composed alternatively of legato and accent attacks (3 times), ending with a vertical accent attack.
4. Tempo Variations.
   a. Accelerando-decelerando: accelerando-decelerando of legato attacks at the one-third location, progressively from 63 bpm to 120 bpm and then back to 63 bpm.
5. Impact Locations.
   a. Impact locations: sequence of legato attacks, played at 63 bpm, composed alternatively of location pairs: rim/one-third, rim/center, one-third/center, and finally locations played and sequenced differently by the two hands.

## 5. Sound Examples to Accompany the Article "Experiments in Modular Design for the Creative Composition of Live Algorithms" by Oliver Bown (Volume 35, Number 3)

These short excerpts provide documentation of experimental works using two types of dynamical system—continuous-time recurrent neural networks (CTRNNs) and dynamic decision trees (DTs)—as patterning modules in the context of live algorithms for music (LAMs). In each excerpt, a single dynamical system responds to the input from one or more live performers, through simple feature analysis, and controls a metronome's rate as well as multiple electronic sound modules that are either triggered directly from the dynamical system's output or from the metronome. This work was produced by Oliver Bown while working as a post-doctoral research assistant at the Centre for Electronic Media Art at Monash University, Melbourne, Australia. It was created using a Java-based software library for real-time generative music, called Beads, which was also developed during this period. Beads is freely available under a GNU General Public License (www.beadsproject.net). The specifics of analysis and synthesis modules are not provided in detail. They are operationally relatively opaque, being the products of hacking and tweaking during the compositional process.

Oliver Bown is a British electronic musician and researcher working in Australia (www.olliebown.com). He is one half of the electronic music duo Icarus and a member of the Not Applicable label and artists' group (www.not-applicable.org). He works with improvising musicians in Europe and Australia. He is interested in the application of complex dynamical systems to musical composition, applications of evolutionary ideas to creative music software, and evolutionary approaches to human musical behavior and culture.

1. CTRNN with Finn Peters (flute).
   Recorded live at Cafe Oto, London, August 2009. Part of a concert of Live Algorithms which followed a three-day workshop at Goldsmiths, University of London. Recorded by Sam Britton. The CTRNN controls FM-synthesis and drum-machine modules. Finn Peters is a British flautist and saxophonist who regularly improvises with electronic musicians. His 2010 album *Music of the Mind* explores the use of a brain–computer interface as a compositional tool.
2. CTRNN with Adrian Sherriff (shakuhachi) and Brigid Burke (clarinet).
   Recorded live at the Guildford Lane Gallery, Melbourne, February 2010. Part of a concert series called Hands Free, which explored autonomous music software. Recorded by Oliver Bown. The CTRNN controls drum machine, FM synthesis, and granular sampler modules. Adrian Sherriff is an Australian trombonist, shakuhachi player, and tabla player who performs with live electronics. Brigid Burke is an Australian clarinetist and bass clarinetist who works in improvised and new music. She performs with live electronics and incorporates

live video mixing into her performances. Burke and Bown released an album of improvised music, *Erase* (2011), on the Not Applicable label.

3. DT with Lothar Ohlmeier (bass clarinet).
   Recorded live at the North Sea Jazz Festival, Rotterdam, July 2010. Part of a special event called OK Computer, which explored contemporary approaches to electronic music performance. Live technical assistance and recording by Roy Carroll. The DT directly triggers a sampler playing percussive bass clarinet and timpani samples, and a granular sample player playing textural bass clarinet samples. Lothar Ohlmeier is a German bass clarinetist and saxophonist and a regular member of the Not Applicable Artists.
4. DT with Brigid Burke (bass clarinet).
   Recorded in a studio session at Northern Melbourne Institute of TAFE, Melbourne, December 2010. Recorded by Oliver Bown. The DT directly triggers subtractive synthesis, drum machine, and granular sample player modules.

### 6. Video Example to Accompany the Article "The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments" by Ajay Kapur et al. (Volume 35, Number 4)

Karmetik Machine Orchestra, REDCAT Theater, Los Angeles, California, USA, 27 January 2010.

### 7. Video Example to Accompany the Article "The Man and Machine Robot Orchestra at Logos" by Laura Maes, Godfried-Willem Raes, and Troy Rogers (Volume 35, Number 4)

The video was created to illustrate the various automatons that are discussed in detail in the article. Performer: Godfried-Willem Raes. Robots: aeio, fa, harmo, ob, pp2, psch, puff, qt, thunderwood, toypi, vibi. Filming and editing: Laura Maes. More information on the M&M orchestra can be found at the Logos Foundation Web site (www.logosfoundation.org).

### 8. Video Example to Accompany the Article "Trimpin: An Interview" by Sasha Leitman (Volume 35, Number 4)

This 5-min sequence from the film *Trimpin: The Sound of Invention* (2011, 77 min) shows Trimpin's design and construction of *if VI was IX*, which is on permanent exhibit in Seattle's Experience Music Project. The piece makes use of over 500 instruments, 32 of which play MIDI files continuously. *if VI was IX* was designed, composed, programmed, and constructed by Trimpin over a seven-month period in 1999.

Producer/director: Peter Esmonde. Camera: Peter Esmonde and Elijah Lawson. Audio: Gabriel Miller. Distributor: Microcinema and Participant Observer.

## Part Three: Additional Content

The 2011 DVD includes a DVD-ROM section. To access the materials contained there, the reader will need to place the DVD into a suitable disc drive on a computer.

### 1. Sound Files to Accompany the Article "The Perceived Affective Expression of Computer-Manipulated Sung Sounds" by Freya Bailes and Roger T. Dean (Volume 35, Number 1)

These 40 audio files are all described on pp. 92–95 of the article. In particular, see Table 1 (which omits the filename prefixes S*n*-, where $n = 1$ to 40).

### 2. Data Files to Accompany the Article "An Evaluation of Musical Score Characteristics for Automatic Classification of Composers" by Ofer Dor and Yoram Reich (Volume 35, Number 3)

1. Base files: full_data.arff; full_data_string.arff; full_data_keyboard.arff; _DO_NOT_CLASSIFY_full_info_data.arff.
2. Binary Files: keyboard (280 ARFF files); string (150 ARFF files); both (360 ARFF files).
3. Composer files: composer_*n*.arff (ten files); major_*n*.arff (ten files).
4. Genre files: baroque_*n*.arff (ten files); classical_*n*.arff (ten files).
5. Genre_instrument files: baroque_keyboard_*n*.arff (ten files); classical_keyboard_*n*.arff (ten files).
6. Instrument files: keyboard_*n*.arff (ten files); string_*n*.arff (ten files).