
KENNETH B. MCALPINE
Abertay University
k.mcalpine@abertay.ac.uk

The Sound of 1-bit: Technical Constraint and Musical Creativity on the 48k Sinclair ZX Spectrum

ABSTRACT

This article explores constraint as a driver of creativity and innovation in early video game soundtracks. Using what was, perhaps, the most constrained platform of all, the 48k Sinclair ZX Spectrum, as a prism through which to examine the development of an early branch of video game music, the paper explores the creative approaches adopted by programmers to circumvent the Spectrum's technical limitations so as to coax the hardware into performing feats of musicality that it had never been designed to achieve. These solutions were not without computational or aural cost, however, and their application often imparted a unique characteristic to the sound, which over time came to define the aesthetic of the 8-bit computer soundtrack, a sound which has been developed since as part of the emerging *chiptune* scene. By discussing pivotal moments in the development of ZX Spectrum music, this article will show how the application of binary impulse trains, granular synthesis, and pulse-width modulation came to shape the sound of 1-bit music.

KEYWORDS: *1-bit game music; ZX Spectrum; technical constraint*

INTRODUCTION

For those who grew up gaming on the video game consoles and home computers of the early 1980s, the bleeps of the in-game music were as much a soundtrack to life as were Iron Maiden or Depeche Mode. Indeed, many teen gamers, myself included, spent much more time playing games and absorbing the sights and sounds of those games than we did spinning vinyl. Certainly, the familiar chirp of Rob Hubbard's theme from *Monty On the Run* (Harrap, 1985) on the Commodore 64 (C64) or Tim Follin's ZX Spectrum soundtrack for *Agent X* (Tatlock et al., 1986) has a definite nostalgic appeal, but the game music of that period is of more than just sentimental value, with a legacy that extends into the contemporary musical mainstream.

The early days of video game music are replete with tales of ingenuity and creativity¹, which were driven largely by the constraints of the sound hardware. Microcomputers like the C64, whose specifications offered a degree of audio hardware support used *Programmable Sound Generators (PSGs)*, dedicated sound chips that provided their voice by synthesizing simple waveforms. Other machines, like the ZX Spectrum (Christie 2016), whose computer architecture was constrained by cost, offered no dedicated hardware support at all, and its motherboard-mounted speaker was controlled using a single-bit value on one of the processor's addressable memory ports.

Regardless of whether the sounds were generated by dedicated PSGs or directly by the *Central Processing Unit (CPU)*, the computer hardware offered little in the way of musical expression. At most, PSGs offered only a few channels of polyphony and a prescriptive palette of simple waveforms, while the monophonic 1-bit Spectrum beeper was more restrictive still, providing just a single-channel square wave with no level control. In response, however, there arose from this digital frontier an explosive period of technical creativity as game programmers and musicians (they were often one and the same) coaxed the hardware into performing feats of musicality that it had never been designed to achieve. The methods that were adopted to broaden and expand the musical capabilities of the PSGs were not without cost, however, and their application often imparted a unique characteristic to the sound, which, over time, came to define the aesthetic, if not the style, of the 8-bit computer soundtrack. Here, 8-bit refers to the generation of microcomputers, of which the C64 and ZX Spectrum were part, which used 8-bit microprocessors at their core. This is distinct from the notion of 1-bit music, which uses only a single bit of information to encode volume level or speaker displacement.

The characteristic 8-bit sound that accompanied the video game soundtracks of the early- and mid-1980s has currency through a number of related contemporary subcultures, including the *retrocomputing scene*, a distributed community of enthusiasts who continue to drive development on obsolete computing platforms (Takhteyev & DuPont, 2013), the *demoscene*, a distributed technoculture focused on real-time computer art (Carlsson, 2009), and the *chipscene*, a vibrant lo-fi musical subculture that repurposes obsolete gaming hardware to make music (Paul, 2015). Appearances of that 8-bit style of music in movie soundtracks (see, for example Brian LeBarton's C64 arrangement of Sex Bob-Omb's *Threshold*, which features in the end credits of Edgar Wright's *Scott Pilgrim vs. the World*), television advertisements (Jonathan Dunn's hypnotic theme from the Gameboy version of Ocean's *Robocop* (1990) was used as the basis for Ariston's *And on... and on...* campaign in the early 1990s), and major exhibitions, such as that at the Smithsonian in 2014 (Melissinos, 2014), suggest a growing acceptance of chip music, alongside 8-bit video game art and animation, as a legitimate form of artistic expression, while the adoption of elements of chiptune by major artists like Mark Ronson (Knowles, 2010) suggests the style is more than a niche crossover.

1. For an overview of the early period of video game music see, for example, Collins (2008), and Collins and Greening (2016)

Even Iron Maiden, those stalwarts of the 80s new wave of British heavy metal, have embraced the sound, launching their 2015 album *Book of Souls* with a NES-style game, which features an 8-bit arrangement of the band's "Speed of Light" (Dickenson & Smith, 2015) as the background track.

To understand and fully appreciate the evolution of that sound it is necessary to approach it from a number of different angles. Of course, we can examine the music itself to understand the stylistic influences that helped to shape the sound; however, stylistic analysis alone does not tell us very much about video game music as a media form. Sometimes, stylistic choices were driven by the narrative of the game, so that the music might provide context to game levels. However, as emerging from interviews carried out by the author with video game coders and composers from the early bit through to the -bit era young coders, many of them were keen to turn around their games quickly and with little appreciation or regard for copyrights and intellectual property. Often, composers just reached for the nearest sheet music to hand or arranged whatever vinyl was spinning in the background.

To really understand how video game music functions as media music we must also delve into the source code and the hardware, employing a platform approach to learn more about how the computer architectures and the games that were written for them shaped both the structure of video game music and how it was realized. It is by examining this broader structural context to video game music that we begin to appreciate the challenges facing early game designers, and see how those constraints functioned as a spur for creativity. This, in turn, can shed light on how the aesthetics of early video game music evolved.

The physical design of musical instruments creates affordances and constraints that, to a great extent, shape the music that is written for them; this is as much the case for electronic instruments (including PSGs) as it is for more traditional acoustic instruments. Video game hardware shaped the sound of early video game music by way of the affordances they offered and the constraints that they imposed.

THE AESTHETICS OF CONSTRAINT?

Constraint has long been recognized as a powerful driver for musical creativity. Many cultures express ideas and expectations about how music ought to be performed, and arguably, it is the role of the professional musician both to satisfy and to challenge these expectations by exploring imaginative departures from the norm. Boden explores this idea (1995, p. 95), noting that, "(c)onstraints map out a territory of structural possibilities which can then be explored, and perhaps transformed". Such common understanding of what music is and how it should sound emerges primarily from the musical structures: the form, timbre, harmony, melody, and rhythm of performance, the grammar and vocabulary of which define the conventions of musical style, and the conventions of interpretation and performance that communicate these from com-

poser through performer to listener (see, for example, Ball 2010, for a detailed yet accessible discussion on this topic). It is from these conventions that there arises one of the most delightful aspects of music, an implicit guessing game between composers and their listeners. Composers provide sufficient structure and familiarity for their audiences to anticipate what is coming next at least some of the time, while providing enough novelty to maintain engagement with the listener (Huron, 2006, p 141). Without the shared notion of musical and performative norms that arises from the constraints of musical structures in particular audiences, this guessing game would often not be possible (there is little point or reward in guessing what is likely to follow if all eventualities are possible and equally likely) and it would often be difficult, in this light, to distinguish creative innovation from chance variation.

The constraints of musical form and grammar can be understood as cultural constructs, emerging by consensus and evolving as composers and performers experiment at the boundaries of style as public tastes and fashions change, but other externalities can impose constraints on musical expression, both implicitly and explicitly. For example, while it is more than just indulgent for a musician to write a sixty-four bar intro for the radio mix of a song, it is commercially reckless, since it limits the number of stations prepared to broadcast the track and the amount of airplay the song will receive. In particular, a commercially-aware musician will implicitly impose self-constraint to ensure that their compositions suit their chosen medium.

Perhaps more significantly, the physical design of musical instruments creates affordances and constraints that, to a great extent, shape the music that is written for them; this is as much the case for electronic instruments (including PSGs) as it is for more traditional acoustic instruments. In short, video game hardware shaped the sound of early video game music by way of the affordances they offered and the constraints that they imposed. Of these affordances, of importance was the complete top-down control provided to videogame composers by the sound hardware and its hosting computing platform (R. Hubbard, personal communication, June 9, 2017). This not only allowed detailed control over each and every aspect of the music and its performance (similar to that of Stockhausen's principle of 'total control'—White, 1968, p, 319), but it was also enabled the means by which the territory of structural possibilities could be explored, mapped out by the hardware's affordances, and transgressively pushed against and stepped beyond the boundaries imposed by its constraints.

THE ZX SPECTRUM: A MODEL OF TECHNICAL CONSTRAINT

A strong hobbyist community exists in the UK (see, for example, Kline, Dyer-Witthford & de Peuter 2003, pp. 84-108). Therefore it may be little surprise that the first home computers were sold in the UK as component kits that required considerable time and technical dexterity to assemble. It was against this backdrop that Science of Cambridge (later to become Sinclair Research Ltd.)

launched the Microcomputer Kit 14 (MK14) in February 1978 as a ‘minimum cost computer’ (Science of Cambridge 1978). Science of Cambridge launched the MK14 at a price point of £39.95, something Practical Electronics described as “a landmark of [...] unassailable proportions” (Berk, 1979). While it was relatively cheap and accessible, the MK14 looked positively primitive alongside its contemporaries, the Commodore PET and the Apple II. Nevertheless, the MK14 sold well enough to justify a successor, named the ZX80 for its 3.25MHz Zilog Z80 processor, with an added X to denote a magical X-factor (Tomkins, 2011).

Following the broadcast of the *Mighty Micro* (1979), a groundbreaking documentary series about the developing computer revolution, the British Broadcasting Corporation’s (BBC) Further Education Department began to take an interest in the burgeoning home computer market, and established the BBC Computer Literacy Project, a series of television and radio programs that would be based around a BBC-branded microcomputer. The project was initially scheduled for launch in the autumn of 1981, which left little time for the BBC to develop its microcomputer in-house. Instead, they collaborated with the Cambridge-based firm, Newbury Labs, to draw up a specification for the machine. This spec matched very closely that of Newbury’s NewBrain, the intention being, presumably, that Newbury Labs would pick up the BBC contract. As the project developed, however, Newbury Labs pulled out of the agreement and did not tender a design. The BBC was forced to postpone the Computer Literacy Project and broaden their search for a partner. Sinclair pitched its new machine, the ZX81.

Sinclair lost out on the BBC contract to rivals Acorn, but the ZX81 was picked up and aggressively promoted by the national newsagent chain, WH-Smith, which had an exclusive contract to supply the machine for six months. It sold by the thousand. Growing support from the popular press and a thriving mail-order games network grew the market for the machine, so that when the ZX Spectrum launched the following year, Sinclair had an established user base and many developers selling through a national network of retail outlets.

Free to specify its own components and price point, Sinclair, designed the most compact and powerful computer that they could to a price, undercutting the Acorn-designed 32K BBC Model B by over £200 at launch (Smith, 2011). With the Computer Literacy project giving the machine free marketing by pushing the idea of the home computer as a tool for learning, thousands of parents bought into the idea, giving the cheaper ZX Spectrum a home.

As a consequence of being designed to a low price point, the ZX Spectrum was a very simple machine. Available in two guises, both models had 16K of ROM and either 16 or 48K of RAM. It was also, if one discounts the analogue cassette interface of the ZX81, which could be co-opted to output simple melodies by *POKE*-ing certain memory registers from BASIC², Sinclair’s first machine to feature any kind of onboard sound interface, a motherboard-mounted 22mm, 40 Ohm “beeper” speaker, which provided just a single channel of 1-bit playback across a 10-octave range.

2. BASIC, or Beginners

All-Purpose Symbolic Instruction Code, is a high-level interpreted programming language that provides simple English-like commands that allow the end-user control over certain aspects of the machine’s hardware. POKE was one such Spectrum BASIC command, which allowed users to write data values directly into the machine’s addressable memory registers. By addressing certain memory registers, the ZX81’s tape interface, which used square wave tones to encode and save digital data to analogue cassette tape, could be made to play simple melodies.

To compound matters, the sound commands were managed directly by the main CPU (a Zilog Z80A processor running at 3.5MHz) and a custom Ferranti *Uncommitted Logic Array* (ULA) chip. Without the availability of dedicated sound hardware, calls to the speaker occupied the processor; therefore, while the Spectrum was beeping it was unable to do anything else.

IMPLIED POLYPHONY: A CHANNEL FOR FREE

It is perhaps not surprising then that few of the early Spectrum titles featured very much in the way of sound or music. Typically, games would feature a single-channel melody as a title tune, and only limited in-game sound effects to punctuate key elements of the gameplay. *Chuckie Egg* (Alderton, 1983) is typical of this model, featuring the melody from “Birdie Song (Birdie Dance)” by the Tweets (Rendall & Thomas, 1981), itself a cover of Werner Thomas’s accordion tune, as its title music. Such repurposing of existing musical themes was not uncommon in the early days of gaming. This was as true of graphics and gameplay as it was of music: *Hungry Horace* (Tang, 1982), for example, one of Sinclair’s launch titles, was essentially *PacMan* (Iwatani, 1980) in disguise, and Artic Computing’s *ZX Galaxians* (Wray, 1982a) and *Invaders* (Wray, 1982b) unofficially recreated those arcade classics as closely as was possible on the Spectrum’s hardware. Ben Daglish, a celebrated C64 musician, recalls:

“I had no idea that copyright existed. Quite seriously [...] I really didn’t. When we wrote all the Jarre stuff and all that [...] we had no real idea as a 14- or 15-year-old kid that you couldn’t just take some music that you liked, whether it was Beethoven or whether it was Jean Michel Jarre. We’d just write it down and put it in a computer game” (Burton & Bowness, 2015).

It was one such act of creative appropriation that lay behind Perfection Software’s *Fahrenheit 3000* (Jones & Williams, 1984), a 64-screen platform game. Perfection wanted a title theme that would make an impact as soon as the game loaded, and Peter Jones suggested using Johann Sebastian Bach’s 18th-century composition *Tocatta and Fugue in D minor* which he had heard opening the movie *Rollerball* (Jewison, 1975). Working from the sheet music of Sky’s 1980 cover (rearranged by Kevin Peek), Jones coded a five-minute beeper arrangement in Sinclair BASIC, before Tim Williams converted it to machine code for the final game.

What makes the music in *Fahrenheit 3000* significant is not so much the arrangement, which doesn’t quite stick faithfully to either the Bach or the Sky sources but, rather, the choice of musical material itself. The opening statement of Bach’s fugue is a sequence of semiquavers, which alternate between the melody and an implied pedal point on A. The effect, particularly when played at speed, is to create a sense of two-voice polyphony by using the pedal note to continually reinforce the sense of the tonal center against the melody.



Figure 4: The opening section of J. S. Bach's Fugue in D minor creates a sense of two-voice polyphony by contrasting a repeated pedal note against a melody line.

Jet Set Willy (Smith, 1984) features a similar technique in its arrangement of Beethoven's *Piano Sonata no. 14, Moonlight*. Using a pattern of broken octaves, similar to the left-hand bass patterns of Boogie Woogie or Stride piano, the arrangement creates a sense of continuous movement between melody and accompaniment. The effect is striking, and it is easy to forget that there is nothing more complex here than a sequence of single-channel square wave tones.

Ben Daglish took the idea to its logical extreme with his soundtrack for Gremlin Graphics' *Arkanoidclone, Krakout* (Toone et al., 1987), providing an implied bass, accompaniment and melody, all played at breakneck speed. Importantly, he recalls that part of the joy of working on a Spectrum was the sense of challenge that it gave. It forced composers to look for ways to circumvent its limitations and find novel ways to introduce dynamic movement and musical interest, and often that involved harnessing the power of the computer itself:

“Half the point of writing some of the music that I did, writing it on a computer, was that it meant that I could use notes that were never actually meant to be played by human beings. I could do really fast runs, scales and arpeggios” (Burton & Bowness, 2015).

His earlier port of *Thing Bounces Back* (Kerry et al, 1987) for the Spectrum used a similar approach, but alternates between a bluesy bass vamp in broken octaves and a bright blues melody. The effect works in much the same way a blues harpist will alternate between vamping and soloing to self-accompany, making use of the listener's aural memory and a strong sense of harmonic familiarity with the I-IV-V chord progression.

GRANULAR SYNTHESIS: PLAYERS CAN'T HELP ACTING ON IMPULSE

Artic's *Invaders* was an unofficial clone of Taito's *Space Invaders* (Nishikado, 1978) and features near-identical graphics and field of play to the original coin-op. The soundtrack also mimics the original, which uses a descending, four-note Dorian scale pattern that repeats and gradually speeds up, as the invaders are picked-off by the player.



Figure 5: The descending Dorian scale pattern from Artic's 1982 game, *Invaders*.

The above descending scale sequence plays continuously throughout the game, marking the first use of a continuous non-diegetic soundtrack on the

Spectrum, being released around a year earlier than Bug Byte's *Manic Miner* (Smith, 1983b), whose rendition of Grieg's *In the Hall of the Mountain King* is often credited with this accolade. So how did *Invaders*, and indeed *Manic Miner*, achieve this feat? The solution was to think small.

Granular synthesis is an approach to sound synthesis and manipulation that was posed initially by the Greek composer Iannis Xenakis (1971), who created the composition *Analogique B* from hundreds of splices of tiny fragments of magnetic tape (Robindoré & Xenakis, 1996 pp. 11-12). Conceptually, the idea of treating sounds at times as continuous waves and at others as though they were composed of tiny sound quanta, or grains, opens up many interesting and creative ways of working. For example, two effects that have become commonplace in recent years are the time-stretch and the pitch-shift, which allow for the independent manipulation of tempo and pitch in recorded audio. Usually, these two parameters are inextricably linked: slow down the playback of a sound recording, and the pitch will drop proportionally. Granular synthesis enables the pitch and speed to be processed independently by applying the processing individually to sound grains, before recombining them to construct the final sound output.

Invaders uses granular synthesis as a technical strategy to create an in-game soundtrack that addresses the key limitations of the Spectrum's hardware. Recall that its speaker was controlled directly by the computer's main CPU and ULA, meaning that it was not normally possible to combine both gameplay and sound. Also, because the speaker was 1-bit, controlled via a single pin of the ULA, the speaker was either fully driven or at rest. No intermediate states were addressable, and consequently, there was no level control over the signal, which was a square wave by default. However, a 1-bit device can produce more than a square waveform. It was by recognizing this, and working directly within the software to manipulate the state of the ULA at a low-level, that author William Wray was able to create multiple independent channels of sound within the game.

A single cycle of a digital square wave is little more than a sequence of ones followed by an equal number of zeroes. Repeating this pattern over and over creates a continuous tone whose period, and therefore frequency, is determined by the number of ones and zeroes in each cycle. Increasing the number of ones and zeroes increases the period, and so lowers the pitch, and vice versa. A Fourier analysis (Roads 1996, pp. 1084-1112) of the square wave reveals a well-defined and characteristic spectral signature:

$$\text{Relative magnitude of the } n^{\text{th}} \text{ harmonic} = \begin{cases} \frac{1}{n}, & \text{if } n \text{ is odd} \\ 0, & \text{otherwise} \end{cases}$$

Now suppose that, rather than outputting ones and zeroes in equal measure, one outputs a sequence of ones followed by three times as many zeroes. This is a pulse wave, an asymmetrical version of the square wave. In this case, 25% of the pulse is made from ones, and the rest from zeroes, and so the pulse

wave has a *duty cycle* of 25%. Its tonal characteristics are similar to those of the square wave, although a Fourier analysis reveals a different frequency spectrum, where M is the number of successive ones in the N sample points that represent a complete cycle of the wave:

$$\text{Relative magnitude of the } n^{\text{th}} \text{ harmonic} = \text{sinc}(n) = \frac{\sin\left(\frac{\pi n M}{N}\right)}{\frac{\pi n M}{N}}$$

Continuing in this manner, the number of ones in each cycle of the wave can be reduced further to create smaller and smaller duty cycles, varying the frequency spectrum and tone of the sound, until the beeper is sent just a single positive bit followed by a stream of zeroes. This signal is a binary impulse, and its Fourier transform is a constant. In other words, an impulse contains all possible frequencies at equal magnitude.

It is not possible to hear an impulse on its own, but it is possible to hear its effect on a speaker, the so-called *impulse response*. Any speaker exhibits a degree of inertia, taking a short but finite time to move from rest to maximum displacement and back again, and it is this response that can be heard as a noticeable click. By sequencing a series of binary impulses together separated by short gaps, an *impulse train* emerges, a pitched tone, the frequency of which is determined by the period between successive impulses, and which contains all of the harmonics of the signal at equal strength, as shown in Figure 6.

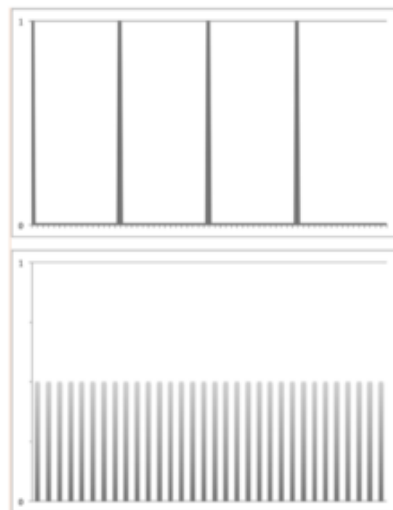


Figure 6 – Outputting an impulse train generates a signal whose frequency spectrum contains an equal amount of energy across all harmonics. The top plot here shows a time domain representation of the sound, showing a series of impulses separated by silence. The lower plot shows a frequency domain representation of the same sound, which has its energy distributed equally across all of its harmonics.

Invaders uses binary impulse train synthesis to create all of the sounds in the game, ensuring that the speaker is tied up for as short a period as possible while still allowing for continuous in-game sound, the game processing taking place in the fractions of a second between impulses. Moreover, the story does not end there. By using clever sequencing of the sounds, similar to that of the implied polyphony discussed in Section 4 above, *Invaders* manages to create multiple sound effects playing synchronously with the underscore.

The first, and most frequent of the game's sound effects is the alien explosion, which is cued whenever a player shot collides with one of the alien invaders. The explosion lasts for approximately 85ms, and is always triggered sequentially with the underscore. If an explosion sound coincides with one of the soundtrack tones, the sound that was triggered first, either the tone or the explosion effect, takes priority, and the subsequent sound is delayed until the first sound has completed. This results in a maximum delay for the explosion sound of around 25ms, which is barely perceptible in the context of the game. For the underscore, however, the worst-case situation could result in a delay of around 80–90 ms, which is enough to cause a degree of jerkiness to the underlying note sequence, although not so much as to cause it to break down.

The second effect is triggered by a bonus mystery ship, which travels across the top of the screen. Here, the sound effect plays continuously while the ship is onscreen, which takes approximately between 6 to 7 seconds, and is created by toggling the speaker on and off at 25ms intervals. During the mystery ship effect, when an underscore tone is due to be triggered, the game stops generating the mystery ship impulses and prioritizes the underscore grain, before picking up the mystery ship sound when the underscore tone has finished. The blip effectively masks the discontinuity in the mystery ship sound effect, creating an illusory continuity of tone in the latter. The final two effects are the player explosion and a level-start siren effect. These are played strictly sequentially, and cause the other elements of the soundtrack to stop playing.

Aside from the slight lumpiness to the underscore caused by the prioritized sequencing of the soundtrack elements, and the curious omission of sound effects for the player ship's laser fire, the game's soundtrack is very effective, not just referencing the original sound effects from the coin-op, but also in creating a real sense of continuous two- or three-channel sound, something that it achieves by the clever handling and sequencing of the short impulse trains.

Extending this idea further, it wasn't long before developers were using the technique to play two simultaneous musical lines by alternating between two or more grain pitches, and that grainy, bubbly quality became firmly established as part of the Spectrum sound. *Rockman* (Carter, 1985), for example, features an arrangement of the first movement of Mozart's *Eine Kleine Nachtmusik*, although the use of 50ms sound grains and lengthy inter-grain silences results in an unconvincing multi-voice effect, in the same way that a slowing a film sequence to below about 15 frames per second spoils the illusion of continuity

of motion, and the viewer becomes aware that they are seeing a series of time-sampled images. More successful was Imagine's port of the Konami coin-op, *Yie Ar Kung Fu* (Beuken & Thorpe, 1985), which uses the effect to play the main game stings in double-octaves, and *Dynamite Dan* (Bowkett, 1985), which uses alternating and arpeggiated grain pitches to recreate Mozart's *Rondo a la Turca*. Durell Software featured two-voice granular music tracks on two of their 1986 releases, *Thanatos* (Richardson, 1986a) and *Turbo Esprit* (Richardson, 1986b).

The music on *Turbo Esprit* is a fine example of the technique. Its Jan Hammer-styled melody complements perfectly the *Miami Vice*-like gameplay.

SINGING TO THE TUNE OF TWO

In 1983, Matthew Smith, a schoolboy from the seaside town of New Brighton in the North-West of England, was loaned a Spectrum by Liverpool-based publisher Bug Byte to develop three games. His first title, *Styx* (1983a), was a fairly simple action maze game based on a single, repeating screen that became progressively more difficult each time the player completed a level. It was his second game, *Manic Miner*, which became a runaway success, making Smith an unlikely superstar, and introduced the Spectrum's first truly iconic character, Miner Willy.

Manic Miner was based on *Miner 2049er* (Hogue 1982), a platform game that featured a Canadian Mountie, Bounty Bob, navigating his way through ten different screens and inspecting each area before his oxygen runs out. Several elements of *Miner 2049er* appear in *Manic Miner* (the underground setting and the oxygen-level as a timer, for example), but in creating Miner Willy, Smith injected a particularly British spin on the game, with an absurd humor to the level and character design, and a Pythonesque boot³, which descends to squash Willy when the game is over.

On loading, the game displays a dynamic title screen showing the sun setting behind an idyllic cliff-top house, below which an animated keyboard plays, pianola-style, the notes of a delightfully-clangorous two-channel rendition of *The Beautiful Blue Danube* by Johann Strauss II. Although the music routine includes an algorithm that uses the note data to display the notes onscreen, the keyboard graphics show a shortened octave (C to E) to the left of middle C, making it almost impossible to use this as a visual point of reference for transcribing the music.

Smith (2014) notes that:

“The game needed music, as I felt it was an integral part of the attraction. The title song, I had an old, simple piano arrangement [of *The Beautiful Blue Danube*] in sheet music so it was easy to transcribe. I did everything as quickly as possible, got the loop running as fast as possible, but I never got too prissy about exact timings”.

3. “Pythonesque boot” is a reference to the surreal animation of a gigantic squashing boot that regularly appears in the television series of the British comedy sketch group, *Monty Python*, in order to segue various sketches

A RAM disassembly of Smith's code reveals that he used impulse trains as the basis of the title music routine. The music was stored in memory as a series of 95 groups, each containing three data bytes. Each triplet corresponds to a separate beat (or sub-beat) in the arrangement, and each is encoded as a duration and a pair of pitch values, or more accurately, as counter values, which are used to calculate the period between successive impulses using a technique known as *frequency divider*, or *divide down synthesis* (Roads 1996, p. 925).

This technique generates a waveform by counting the pulses of a master clock, and triggering an impulse when a chosen divisor (the counter limit) is reached. The counter is then reset and begins again. This generates a periodic impulse train at a frequency that can be calculated as follows:

$$\frac{\text{sample frequency}}{\text{counter limit}}$$

By rearranging the equation, one can calculate the counter limit that corresponds to any given frequency. In the case of *Manic Miner*, the counter is updated on each cycle of the theme-music subroutine, and so the timing of each master clock tick is determined by two factors: the clock speed of the Z80 CPU, which runs at 3.5 MHz, and the length of time taken by the CPU to execute each of the machine instructions in the loop, which can be obtained experimentally. Smith was thus able to construct a frequency table that mapped the notes of the musical arrangement to a series of counter values, and it is these values that provide the note data for his routine.

Smith's music routine uses two counters to calculate two simultaneous impulse trains. The routine writes the two counter values stored in the data triplets into two memory registers, and calculates the period between successive impulses, effectively interleaving the two impulse trains on playback to create two channels of playback. For single melody notes, Smith encoded the pitch as a pair of counter values separated by 1 to create a phasing effect. Chords are encoded as two distinct frequency values. The phasing effect works well, creating a harmonically rich, time-varying tone on the single notes with a characteristic sweeping effect at the beat frequency. However, when the effect is used to trigger two simultaneous distinct pitches, the routine introduces a degree of pitch ambiguity that results from the relative amplitudes of the harmonics of the individual tones.

As noted above, single notes are encoded as pairs of counter values separated by a single unit, the effect of which is to create two binary impulse trains separated in frequency by only a few Hertz. This results in a frequency spectrum that is very close to a harmonic series, as illustrated in Figure 8.



Figure 8: A spectral plot of the two near-coincident impulse trains shows a pseudo-harmonic series, although the concordance between the harmonics of the lower tone, illustrated by the dark bands, and the upper tone, illustrated by the lighter bands, decreases with increasing frequency. This harmonic character makes it easy to identify a definite sense of pitch.

When two impulse trains are interleaved at distinct frequencies, this pseudo-harmonic spectrum breaks down, as shown in Figure 9 below. This spectral plot illustrates a major third interval. As before, the dark bands correspond to the harmonics of the lower tone in the interval, and the light bands to the harmonics of the upper tone. It can be seen immediately that there is no regular structure to these frequency components. The spacing between spectral components is variable, and includes a number of very closely clustered components, which introduces an unpleasant beating to the tone. Also, because each of the harmonics of each tone has equal magnitude, one of the key auditory cues that we normally use to locate and identify pitch, the fundamental, which is usually the strongest of these frequency components, is not evident. Every frequency component therefore arbitrarily becomes the dominant one as the ear focuses in on different regions, creating a very vague and indistinct sense of pitch. The overall effect is to create a sense in the listener of a rough, complex tone, rather than two discrete and distinct pitches.



Figure 9: A spectral plot of two non-coincident impulse trains shows a more complex relationship. There is variability in the spacing between components and some clustering, leading to beating. The uniform magnitude of the components makes it very difficult to identify discrete pitches.

Smith's approach, then, was innovative and, to an extent, very effective. He had managed to move beyond implying polyphony on a macro level, by manipulating the temporal arrangement of fairly large-scale sound grains, to implying it on a micro level by interleaving impulses, the smallest units of binary sound. This took ZX Spectrum music into similar territory to that which was explored by electronic music pioneers like Pete Samson, whose work with MIT's TX-0 and PDP-1 computer systems, explored similar methods some twenty years earlier (Levy, 2010, p 17-18), and suggested a direction for other developers to continue innovating.

PULSE-WIDTH MODULATION

In 1984, Quicksilver's *Zombie Zombie* (White & Sutherland, 1984) became the first spectrum game to address the failings of Manic Miner's two-channel routine and coax two completely independent channels of tunable square waves from the spectrum using *pulse-width modulation* (PWM). As discussed earlier, sending different sequences of ones and zeroes to the beeper allows the creation of a series of related wave shapes, from trains of binary impulses through to pulse waves of varying duty cycle. This idea can be taken one step further by returning to the idea of speaker inertia, which is the notion that a speaker cone cannot change its state discretely and instantaneously. When driven, it takes a short but finite time to reach maximum displacement and must move through all its intermediate states between fully off and fully on. The speaker behaves in a similar, though not identical way, as it returns to rest. Modulating the width of the signals (by varying the amount of time that the speaker is driven relative to the time that it is not) sent to the beeper, the speaker can be driven to intermediate points between off and on, thereby simulating the effect of a continuous analogue voltage. There are, as you might imagine, many ways to achieve this, but the most common method for the Spectrum was to use pre-calculated lookup tables to convert note frequencies to counter values which could be stored in memory and used to synthesize pulse trains in a similar way to the binary impulse trains discussed earlier. Using this form of PWM, the speaker cone could be made to dance in very elaborate ways to create very complex multi-voice tracks. This process tied up the CPU completely, though, meaning that the effect was only possible for the title screen and breaks in gameplay.

The sound routine in *Zombie Zombie* generates two-channels of sound without any volume or timbral control, and is based around an eighth note quantization scheme, with longer notes consisting of multiple eighth notes at the same pitch and triggered sequentially. The game features three main music sequences. The first is a triumphal, march-like setting of *Ten Green Bottles*, which morphs in bar 9 into an unsettling arrangement in parallel augmented 4ths, a reference to the common eighties horror soundtrack trope of the distended children's song or nursery rhyme. The game also features a simple, yet triumphal arrangement of Bizet's *March of the Toreadors* on completion of the

game, and a track that combines White's two-channel routine with the implied polyphony technique described in Section 4, combining bass and a simple arpeggiated accompaniment to create the suggestion of three simultaneous voices.

Having established PWM as a viable approach to music-making on the Spectrum, some games applied the technique with varying degrees of success, while Melbourne House's *Wham! The Music Box* (Alexander, 1985), a fairly sophisticated music sequencer and percussion synthesizer provided users with an easy-to-use graphical interface that would be familiar to users of most digital audio workstations today. The Spectrum's beeper, however, had yet more to give, and it was Tim Follin, a young programmer from St. Helens, in the northwest of England, who really embraced PWM, and took the Spectrum and its 1-bit voice to a whole new level. Follin developed his sound routine on his earliest titles, *Subterranean Stryker* (Follin, 1985), *Star Firebirds* (Follin et al, 1985a) and *Vectron* (Follin et al, 1985b), so that by 1986 with *Agent X*, both his signature sound and his technical implementation, which had reached a channel count of five, along with percussion, enveloping, portamento and phasing, were already very well developed. This did, however, come at the expense of audio fidelity.

In retrospect, Follin's earliest soundtracks showcase the incremental development of both his sound engine and his emerging musical style. The soundtrack for his first Spectrum game, *Subterranean Stryker*, is interesting only insofar as it demonstrates some of his engine's nascent capabilities. It features a single-channel melody line, which drifts stylistically and with little in the way of melodic coherence, the programming equivalent, perhaps, of a guitarist noodling on a fretboard. Beneath the notes, however, can be heard amplitude enveloping, a far-from-trivial task on a speaker that can only be either on or off, and a phasing effect, creating a dynamically-changing timbre, both features that Follin would continue to develop. For his next title, *Star Firebirds*, Follin introduced a portamento effect, creating quite dramatic Emersonian pitch glides in places, but it was *Vectron*, a 3D maze game inspired by the *Space Paranoids* sequence from Disney's *Tron* (Lisberger, 1982), where both the engine and Follin's musical style really begin to shine through. The soundtrack in *Vectron* manages three independent voices during playback and begins with a phased, enveloped synth leading into an electronic fanfare, before a fast blues-scale riff, not unlike the percussive organ lines of Keith Emerson and Rick Wakeman, begins. The score then breaks style, directly referencing Wendy Carlos's original score from *Tron*, before returning to a series of blues-scale sequences.

Follin published his three-channel music routine as a hexadecimal type-in program listing in *Your Sinclair* magazine (Follin, 1987), making it freely available for use in non-commercial programs. The listing contains just 167 lines of code, and the entire routine, complete with note data weighs in at just over 1K in size. The article noted that, at the time, Follin was working on a new 6-channel routine with chorus, bass, echo, portamento and full ADSR, all elements that would turn up in his later soundtracks as his commercial engine continued to develop.

In 1986, with the release of *Agent X*, Follin upped the channel count to 5, although this came at the expense of some audio fidelity. With the processor pushed to its limits, the music is very lo-fi, something Follin acknowledged in an interview with Eurogamer, noting that “It’s hard to actually hear [the music in *Agent X*], I think I’d pushed the processor too far actually!”. Follin’s *Agent X* engine works by using five of the Z80’s registers, sections of RAM inside the main CPU that can be used to store and rapidly operate on frequently-used data, prioritized areas of memory that allow for rapid access by the processor, in a loop, all of which count down from a series of predetermined values to zero. When each loop is complete, it generates a pulse, the width of which determines the speaker level. The constantly shifting pulse-widths affect both the level and timbre, adding noise in the sense that the changing harmonic content introduces an undesirable roughness to the sound and causes tuning problems as the channel count rises.

SUMMARY

That peculiar quality of sound of the ZX Spectrum, its quality of sound, the grungy fuzziness, came to define the sound of the Spectrum for a generation of gamers, becoming an important feature of the style, in much the same way that the warmth of tape saturation came to characterize the sound of recorded music throughout the 1960s and 70s to such an extent that modern developers now devote significant time and resource to create effects algorithms that degrade pristine digital recordings to simulate some of that analogue character.

It was a sound, however, that evolved gradually, through a series of logical steps, each of which is rooted elsewhere in the annals of electronic music history. Interestingly, however, my conversations with those early game music pioneers and game music historians, including Rob Hubbard, Ben Daglish, and Chris Abbott, suggest that these innovations happened independently. These were young, creative programmers looking for a way around a technical problem. In the same way that they weren’t aware of copyrights, nor were they aware of Max Matthews’ and Peter Samson’s innovations in electronic music that had taken place in the preceding decades.

Following the demise of the Spectrum in 1992, 1-bit music continued to feature in many games, largely thanks to the PC speaker, which provided the default sound output for many early PC games. LucasArts’ *The Secret of Monkey Island* (Gilbert, 1990) is a fine example of such early PC soundtracks, using a combination of the techniques outlined above to create an engaging title theme.

With the introduction of dedicated PC soundcards, Frequency Modulation and sample playback synthesis gradually replaced PSGs (Programmable Sound Generators) as the source of video game sound, and video game soundtracks became more cinematic, often increasingly relying on multiple channels with orchestral timbres, both in concept and in execution, and yet the chirpy 1-bit sound continued. Music trackers, such as the DOS-based *Monotone* (Leonard, 2008) and *Pulse Tracker* (Larsson, 2012), put these 1-bit music techniques in the

hands of musicians rather than programmers. Emulators and hacked code allowed a new generation of musicians to continue to push the capabilities of the Spectrum, and demoscene meets and *compos* (competitive events that encourage the creation of sophisticated real-time generative art and music using obsolete and limited hardware) continue to provide platforms for creative performance.

The growth in recent years of open development systems like the Raspberry PI, which was introduced to promote the teaching of basic computer science in schools, has kick-started the same sort of experimental approach to coding that happened during the first wave of the microcomputer revolution. With just a few lines of code and a small Mylar speaker wired to the digital output pin of an Arduino, a new generation of coders has been able to experiment with 1-bit music techniques.

Recent developments in music technology over the last 30 years have seen an explosion in the range and scope of music creation and production tools. Virtualization has taken esoteric studio hardware that previously would have been the preserve of international-class studios and converted them to code, allowing all-comers to build flexible virtual processing racks, driven by carefully designed presets that allow the devices easily to integrate into any production session. Classic synths have similarly been modeled and virtualized, and primed, both with sounds and loopable MIDI sequences, to allow their users to channel the sounds of, for example, Kraftwerk, the Prodigy, or Emerson, Lake and Palmer, with a few simple selections from a drop-down menu. Such is the democratizing effect of this technology that armed with a laptop, a suitable digital audio workstation (DAW) and a little time and enthusiasm, it is possible to create quite authentic-sounding electronic music tracks with relatively little effort. In many respects, this is a very positive development. It has provided a creative outlet for many and has made music making and production more accessible. This accessibility, however, comes at a cost.

Constraint is what the lo-fi sound of the 8-bit microcomputer can provide. With simple, raw waveforms, limited polyphony and few options for dynamic articulation, chip musicians have no option but to go right back to the very basics and address the fundamentals that make music engaging and entertaining.

Historically, scholars such as Amabile, (1983) have argued that too much constraint on creative freedom decreases the intrinsic motivation to create. However, recent work has demonstrated a clear distinction between constraints that obstruct creativity (for example by encouraging conformity, as may be the case when composing new work from preconfigured musical patterns and presets), and those that promote it (see, for example, Stokes, 2005). In addition, recent research has suggested that the “Paradox of Choice” (Schwartz, 2004) can have similarly deleterious effects on intrinsic motivation (Iyengar Lepper, 2000) and originality (Chua Iyengar, 2008). While, on the one hand, it is won-

derfully liberating to have complex in-the-box software solutions that enable musicians to compose, arrange and produce, on the other hand, the tyranny of choice that is presented can be crippling, leading to creative procrastination as one searches for ‘just the right sound’, rather than ploughing on with the process of creation. It is just as Devo sang back in the 80s: “Freedom of choice is what you got; Freedom from choice is what you want,” (Mothersbaugh, 1980).

Constraint is what the lo-fi sound of the 8-bit microcomputer can provide. With simple, raw waveforms, limited polyphony and few options for dynamic articulation, chip musicians have no option but to go right back to the very basics and address the fundamentals that make music engaging and entertaining. There is nowhere for half-formed ideas or weak arrangements to hide. It is electronic music in its most fundamental state; it is about simple ideas expressed well.

In 2003, Malcolm McLaren declared 8-bit to be the new punk (2003). It has that same, lo-fi DIY aesthetic and, just as punk raised a defiant middle finger to the worst excesses of prog rock and glam rock, so too 8-bit and the associated lo-fi subculture stands in stark contrast to the over-produced sound of much of current commercial music. The Spectrum embodies that spirit perfectly and, as a small but vibrant part of the retro computing scene, the demoscene and the chipscene suggest that there are, even now, many new musical chapters to be written in Z80 assembly.

REFERENCES

- Amabile, T. M. (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, 45(2), pp. 357-376.
- Ball, P. (2010). *The Music Instinct: How Music Works and why We Can't Do Without it*. London: Random House.
- Berk, A. (1979, May) MK14 Review. *Practical Electronics*, p. 50.
- Boden, M. (1990). *The Creative Mind: Myths and Mechanisms*. London: Wiedenfield and Nicholson.
- Carlsson, A. (2009). The forgotten pioneers of creative hacking and social networking—Introducing the demoscene. *Re: live*, 16.
- Christie, T. (2016), *The Spectrum of Adventure: A Brief History of Interactive Fiction on the Sinclair ZX Spectrum*, Extremis Publishing.
- Chua, R. Y. J. and Iyengar, S. S. (2008). Creativity as a matter of choice: Prior experience and task instruction as boundary conditions for the positive effect of choice on creativity. *Journal of Creative Behavior*, 42(3), pp. 164-180.
- Collins, K. (2008). *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. Cambridge: MIT Press.
- Collins and Greening (2016) *The Beep Book: Documenting the History of Game Sound*. Canada: Ethonal
- Follin, T. (1987). Star Tip 2. In *Your Sinclair*, Issue 20, p. 55.
- Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge: MIT Press.
- Iyengar, S. S. and Lepper, M. R. (2000). When choice is demotivating: Can one desire too much of a good thing?. *Journal of Personality and Social Psychology*, 79(6), pp. 995-1006.
- Knowles, J. (2010). How computer games are creating new art and music. *British Broadcasting Corporation*. Retrieved from: <http://www.bbc.co.uk/news/10260769>
- Levy, S. (2010). *Hackers: Heroes of the Computer Revolution, 25th Anniversary Edition*. Sebastopol: O'Reilly Media, Inc.

- McLaren, M. (2003). 8-bit Punk. In *Wired*, Issue 11.11. Retrieved from: <http://www.wired.com/wired/archive/11.11/mclaren.html>.
- Melissinos, C. (2014). *The Art of Video Games*, 16 March–30 September, Smithsonian American Art Museum, Washington, D.C.
- Paul, L. (2014). For the Love of Chiptunes. In K. Collins, B. Kapralos, and H. Tessler (eds.), *Oxford Handbook of Interactive Audio*, Chapter 30, pp. 507–530.
- Roads, C. (1996). *The Computer Music Tutorial*. Cambridge: MIT Press.
- Robindoré, B. and Xenakis, I. (1996). Eskhaté Ereuna: Extending the Limits of Musical Thought – Comments On and By Iannis Xenakis. *Computer Music Journal* 20(4), pp. 11–16.
- Schwartz, B. (2004). The tyranny of choice. *Scientific American*, 290(4), pp. 70–75.
- Science of Cambridge. (1978). *MK14 Standard Micro Computer Kit*. Cambridge: Science of Cambridge.
- Sinclair Research Ltd. (1982). *ZX Spectrum Introductory Booklet*. Cambridge: Sinclair Research Ltd.
- Smith, T. (2011). The BBC Micro turns 30: The 8-bit 1980s dream machine. *The Register*. Retrieved from: http://www.theregister.co.uk/2011/11/30/bbc_micro_model_b_30th_anniversary/?page=4
- Stokes, P. (2005). *Creativity from Constraints: The Psychology of Breakthrough*. New York: Springer Publishing Company, Inc.
- Takhteyev, Y. and DuPont, Q. (2013). Retrocomputing as preservation and remix. *Library Hi Tech*, 31(2), pp. 355–370.
- Tomkins, S. (2011). ZX81: Small black box of computing desire. *British Broadcasting Corporation*. Retrieved from: <http://www.bbc.co.uk/news/magazine-12703674>
- White, J. (1968). *Understanding and Enjoying Music*. New York: Dodd, Mead.
- Xenakis, I. (1971). *Formalized music*. Bloomington: Indiana University Press.

AUDIOVISUAL

- Burton, C. & Bowness, A. (2015). Ben Daglish BIT Brighton 2015 Interview (preview). *c64audio*. Retrieved from: <https://www.youtube.com/watch?v=qhv6U8Wm0GY>
- Eurogamer.net. (2015). Code Britannia: Tim Follin. Retrieved from: <http://www.eurogamer.net/articles/2014-01-02-code-britannia-tim-follin>
- Lisberger, S. (1982). *Tron*. USA: Walt Disney Productions.
- Jewison, N. (1975). *Rollerball*. USA: MGM Studios, Inc.
- Smith, M. (2014). *From Bedrooms to Billions*. Anthony Caulfield and Nicola Caulfield, UK: Independent.
- The Mighty Micro. (1979). ITV. 29th October, 20:30.
- Wright, E. (2010). *Scott Pilgrim vs. the World*. USA: Big Talk Films.

GAME MUSIC (BY COMPOSER/DESIGNER/SOFTWARE HOUSE)

- Alderton, N. (1983). *Chuckie Egg*. UK: Elite.
- Alexander, M. (1985). *Wham! The Music Box*. UK: Melbourne House.
- Beuken, B. & Thorpe, F. D. (1985). *Yie Ar Kung Fu*. UK: Imagine Software Ltd.
- Bowkett, R. (1985). *Dynamite Dan*. UK: Mirrorsoft Ltd.
- Carter, D. (1985). *Rockman*. UK: Mastertronic Ltd.
- Follin, M. (1985). *Subterranean Stryker*. UK: Insight Software.
- Follin, M., Wilson, M., & Gough, P. (1985a). *Star Firebirds*. UK: Insight Software.
- Follin, M., Wilson, M., & Gough, P. (1985b). *Vectron*. UK: Insight Software.
- Gilbert, R. (1990). *The Secret of Monkey Island*. USA: LucasArts.
- Harrap, P. (1985). *Monty on the Run*. UK: Gremlin Graphics.
- Hogue, B. (1982). *Miner 2049er*. US: Big Five Software.
- Iron Maiden: Speed Of Light Game*. Last 17, 2017. Retrieved from <http://speedoflight.ironmaiden.com>
- Iwatani, T. (1980). *PacMan*. Japan: Namco Corporation.
- Jones, C. & Williams, T. (1984a). *Fahrenheit 3000*. UK: Perfection Software.
- Kerry, C., Dooley, C., Hollingworth, S., Harrap, P., Holmes, G., Kerry, S., & Duroe, M. (1987). *Thing Bounces Back*. UK: Gremlin Graphics Software Ltd.
- Larsson, F. (2012). *Pulse Tracker v1.02a*. Retrieved from: <http://jackdawinteractive.com/files/programs/pulse.zip>
- Leonard, J. (2008). *Monotone v0.38b*. Retrieved from: <http://www.oldskool.org/pc/MONOTONE>
- Nishikado, T. (1978) *Space Invaders*. Japan: Taito Corporation.
- Ocean (1990). *Robocop*. UK: Ocean.
- Richardson, M. (1986a). *Thanatos*. UK: Durrell Software Ltd.
- Richardson, M. (1986b). *Turbo Esprit*. UK: Durrell Software Ltd
- Smith, M. (1983a). *Styx*. UK: Bug Byte.
- Smith, M. (1983b). *Manic Miner*. UK: Bug Byte.
- Smith, M. (1984). *Jet Set Willy*. UK: Software Projects.
- Tang, W. (1982). *Hungry Horace*. UK: Sinclair Research Ltd.
- Tatlock, J; Tatlock, S and Follin, T. (1986). *Agent X*. UK: Mastertronic.
- Toone, B; Holmes, G; Green, A; Lloyd, T& Duroe, M. (1987). *Krakout*. UK: Gremlin Graphics Software Ltd.
- White, S. & Sutherland, A. (1984). *Zombie Zombie*. UK: Quicksilver Ltd.
- Wray, W. (1982a). *ZX Galaxians*. UK: Artic Computing Ltd.
- Wray, W. (1982b). *Invaders*. UK: Artic Computing Ltd.

SOUND RECORDINGS

- Bach, J.S., re-arranged by Peek, K. (1980). *Toccatà* [Recorded by Sky]. UK: Ariola

- Dickenson, B. & Smith, A. (2015). “Speed of Light” [Recorded by Iron Maiden]. *Book of Souls*. US: BMG Recorded Music.
- Mothersbaugh, M. (1980). *Freedom of Choice* [Recorded by Devo]. US: Warner Bros Records.
- Rendall, F. & Thomas, W. (1981). “Birdie Song (Birdie Dance)” [Recorded by The Tweets]. UK: PRT.
- Sky Writing Ltd. (1980). *Toccata*. UK: Sanctuary Records Group Ltd.

AUTHOR’S INFO

Kenneth McAlpine is a musician, author and academic at Abertay University, Dundee, Scotland, and is a Member of the Editorial Board of *The Computer Games Journal*. His research focus includes video game music and the role of technical constraint in the development of its aesthetic, and the implementation and analysis of real-time adaptive music used in interactive contexts.