# Chips with everything: coding as performance and the aesthetics of constraint. Part of AAH 2016: In and Out of Art History: The Video Games Conundrum, Edinburgh, UK, 7 – 9 April.

## Author Dr Kenneth McAlpine , Abertay University

Last January, I sat down with my two children to watch the Jame Badge, an episode of *Hey Duggee*, an animated series about the adventures of the Squirrel Club, a Scout-like organisation for young animals run by Duggee, an avuncular brown dog.

As my children became absorbed, my attention drifted from the television to the iPad in my lap. Until, that is, I heard the familiar chirp of 8-bit style video game music. I looked up to see Duggee and the Squirrels assume the roles of Donkey Kong and Mario as the cartoon transformed into a takeoff of Nintendo's arcade classic, while my daughter, who normally prefers tactile play with an impossibly-proportioned blonde doll, pointed at the screen excitedly, yelling, "Daddy! Daddy! Listen: It's Mario music!"

Any doubts I might have had about the currency of 8-bit video game music as a pop-culture reference vanished in that moment. Music that works as a cutaway gag in a show aimed at four-year-olds is part of, or very near, the mainstream.

For those of us who grew up with the video game consoles and home computers of the early 1980s, that sound has a definite nostalgic appeal, and certainly, it was a nostalgic wink towards parents that the director, Grant Orchard, intended. He explained to me how the look and feel of the show shares many of the characteristics of those early 8-bit games, with flat, distinctive characters, and planar environments that use a visual and aural grammar that work without the need for any complex backstory. As the episode came together, the gag just seemed like a good fit: A chase scene involving a monkey and basket of fruit? Well, what else but Donkey Kong?

My daughter's reaction, though, suggests that there is something more than nostalgia at play.

Like the trend for retro gaming, which has been embraced by kids too young to have been born when that first generation of 8-bit machines was already obsolete, that 8-bit sound has currency with a new generation of young musicians through the *chipscene*, a vibrant lo-fi musical subculture that repurposes obsolete gaming hardware to make music.

So how did that sound emerge and what is its appeal? We might answer that question by examining the music to understand its stylistic influences, but those early game soundtracks drew on some astonishingly diverse musical material, ranging from Baroque to blues via everything in between.

Stylistic analysis also doesn't give us a complete picture. Video game music, you see, is functional; just think of the 'attract mode' of many arcade games, non-playable demos used to entice the quarters of prospective gamers like a midway caller drumming up trade for a ring toss stall. It is a type of media music, whose form and structure is determined, at least in part, by factors that lie outside the music itself.

As with film underscore, another form of media music, video game soundtracks, arguably, should portray a sonic image of the game's diegetic world to better characterise its narrative and enhance its emotional reach.

The best film soundtracks do that on a subconscious, and often visceral level. A film, though, once cut, is a fixed and fairly linear structure, which provides a definite temporal framework against which to compose, and film composers can draw upon more than a century of musical vocabulary and grammar to help shape and guide that expression. There were few equivalents in the early days of computer gaming, and the interactivity of games makes it particularly challenging even to judge, never mind capture, its changing visual tempo. Nevertheless, some games managed this seemingly impossible feat beautifully, the relentless accelerando of Taito's Space Invaders, for example, creating a feeling of growing tension that is every bit as palpable as any motion picture thriller.

To understand this interplay between the game and the music, we really have to take a platform approach, delving into the source code and the hardware to learn more about how the music was structured and how it was realised. It is only by examining the hardware and how it shaped and supported the media that were created for it, that we begin to appreciate the challenges – both creative and technical – that presented to those early game designers, and how video game music evolved as a result.

**The Atari**

Let me illustrate that idea using a two player arcade game that used blocks for graphics… and for sound.

**<Change slide>**

Now, of course, Pong wasn't the first arcade game. It wasn't even the first arcade game with sound. But it managed to muscle its way into the public consciousness in a way that none of its predecessors had. Before Pong came along, the domain of the video game was the university computing lab: the hardware required to run games ran to tens or hundreds of thousands of dollars, and lay beyond the reach of even the most committed of home enthusiasts. Pong turned that on its head. It was the machine that domesticated video gaming.

Atari had hired Al Alcorn, a young engineer who had previously worked with Atari's founders, Nolan Bushnell and Ted Dabney at Ampex, and set him to work on developing a bat-and-ball game. He bought a black-and-white television from a Walgreens drugstore for 75 bucks and developed the game on a cheap state machine, a jumble of logic chips and diode arrays. About the sound he recalls that:

*I've seen articles written about how intelligently the sound was done, and how appropriate it was. The truth is, I was running out of parts on the board. Nolan wanted the roar of a crowd of thousands. Ted Dabney told me to make a boo and a hiss when you lost a point, because for every winner there's a loser. I said "Screw it. I don't know how to make any one of those sounds. I don't have enough parts anyhow." So I poked around the sync generator to find an appropriate frequency or tone. Those sounds were done in half a day. They were the sounds that were already in the machine.*

By default, then, rather than by design the shape of video game sound had been defined, and it was square.

**<Change slide>**

To capitalise on the success of Pong, Atari had begun work on a home video game console as early as 1975.

Back then, RAM memory was phenomenally expensive — tens of thousands of dollars per megabyte, in fact — and so to keep costs down, the engineers had to devise a way of minimising the memory requirements, and they looked to the display for a solution.

Nearly every console and computer uses what's called a framebuffer, an area of RAM used to

cache a full image of the display in memory before uploading it in its entirety to the screen. Even for a relatively small monochrome display, the VCS would have needed a 1k framebuffer, which would have pushed the cost of the device well beyond its target price point.

Atari's solution was to create a graphics chip that removed the need for a framebuffer completely by building the image directly, a line-at-a-time, on the screen as its electron gun slewed from side-to-side, carrying out the game processing in the moments that the electron gun was turned off as it transitioned between vertical lines. The cost saving on video memory shifted the responsibility for the video display from hardware to the game code, a complex process that is still described in game programming folklore as 'racing the beam'.

**<Change slide>**

Now what, you may ask, has this got to do with music? Well, on this chip, the TIA, alongisde the display mechanism, were two independent audio circuits, also synced to the display, whose output was multiplexed with the display components and sent to the television speakers via an RF modulator.

The oscillators were basic, providing just 1-bit of resolution – that is, the oscillator could be fully on or fully off, with nothing in between – but by sending the chip complex sequences of 1s and 0s, generated in real-time using a linear feedback shift register they were capable of producing a range of different pitched and noise tones. The rate at which these sequences were generated was sync'd to the horizontal video sync counter and sent to a 5-bit frequency divider circuit, giving just 32 possible pitches.

The actual frequency of each tone depended on two factors, the frequency register and the length of the bit sequence — longer sequences increase the period of the wave and reduce the frequency of the tone.

**<Change slide>**

This table shows the tuning chart for one of the VCS's tones, relative to true pitch. What it shows is that although there were, in principle, 32 pitch values to choose from, some of these departed so dramatically from where they should be that they couldn't really be used for music — the TIA was really intended more for effects than it was for musical expression.

There is an interesting parallel to draw here with the music of the late Renaissance and early Baroque. Although nowadays equal temperament is the norm, it was not always so. For

Renaissance composers, the ideal was just intonation, a system of tuning based around the frequency ratios of the harmonic series. Unfortunately, thanks to a quirk of mathematics, it is not possible to build a consistent pitch system around it. While it sounds melifluous when used melodically within the I, IV and V chords of the root key, the moment the music starts to venture further afield it sounds dreadful.

In order to achieve a more usable range of consonance, some of the notes of the scale were detuned or tempered, losing a degree of pitch accuracy in favour of a better intervallic balance overall. By prioritizing different tuning goals, different tuning systems have very different tonal colours and textures as one moves through the 24 major and minor keys, a feature showcased by Bach in his famous 48 Preludes and Fugues.

The composers of the time understood this, and worked to the strengths of the tuning systems they used, and, interestingly, a similar process happened on the Atari as the limitations of the system began to directly influence the aesthetics of its music.

During a presentation at the Game Developer Conference in 2011, David Crane explained that Garry Kitchen, the developer of Activision's *Pressure Cooker*, calculated the nine notes that could be reasonably approximated by the VCS, and marked them on a Casio keyboard. They hired a professional jingle composer to write a melody for the game using only these marked notes.

**<Change slide>**

**The Professional Composer**

This soundtrack, simple though it was, really marks the beginning of the professionalisation of video game music and the role of the game composer. Before then, it was just one of the tasks, alongside gameplay and graphical design and programming, carried out by an individual. But in March 1983, Atari hired Ed Bogas as the world's first fully-professional in-house video game composer.

A San Franciscan, Bogas had begun his musical career as a keyboard player with the psychedelic rock band, *The United States of America*, before moving into scoring for film and television, most notably writing the music for the *Peanuts* series after Vince Guaraldi's death in 1976. Atari's hiring of Bogas, fully a year before Nintendo hired its first full-time composer, Koji Kondo of Super Mario fame, points to the growing influence of music on video games.

**<Change slide>**

In the UK, Rob Hubbard was, like many jobbing musicians, giving piano lessons and backing cabaret acts in clubs to make ends meet. Sensing the opportunity that the home computer revolution would bring, Hubbard bought himself a Commodore 64 and taught himself to program. Initially, he developed some music education software, hoping to make his fortune, before learning the hard way what we all now know: The home computer as a tool for education is a myth perpetuated by children so that their parents will buy the hardware that allows them to play games. It was always thus.

Hubbard shifted to game programming, writing his own graphics routines, and getting to know intimately the machine's hardware in the process. Although he didn't have any initial success with his own games, his publisher was impressed with his music, and he landed his first professional game music commission with Gremlin's *Thing on a Spring*.

**<Change slide>**

From the outset, Hubbard recognised that creative coding was the key to effective music on the Commodore. Its sound chip, christened SID, the three-letter monicker continuing Commodore's tradition of the friendly anthropomorphicisation of its componentry, was orders of magnitude more advanced than anything else on the market, never mind the erratic qualities of Atari's TIA.

The SID chip had been designed by Bob Yannes, who would later found the synth manufacturer *Ensoniq*, and, with proper ADSR envelopes and filters on top of multiple waveforms and analogue audio and controller inputs, the C64 felt like a hardware synth that just happened to come with a computer built around it.

However, the SID still only had three channels of sound, and the code for both the sound driver and the music data had to fight for memory and resource with the graphics and gameplay, making it a real challenge to create rich, detailed arrangements.

Hubbard's approach was to change the way he thought about music. He hand-coded his own 6502 assembly-level music routines to ensure that his music would run as efficiently as it could, and, to squeeze every last drop of performance from the machine, he would customise it for each composition, adding and removing features as needed. He began to think procedurally about musical structure, and considered how he could abstract and nest levels of musical detail, allowing whole compositions to be encoded and generated procedurally as a series of functions rather than as a linear data stream. It was a fairly groundbreaking approach, incorporating

elements of fractal compression to condense and distil the musical essence of his works to a concentrated form that could be diluted and replayed later.

Over the years, I've interviewed many video game composers, and almost all have cited Hubbard as a role model. Most sought to emulate his routines, of which he was fiercely protective, although few could make code sing in quite the same way. At a time when the industry was still professionalising, he brought the work ethic of a grafting club musician. He was, perhaps, the first video game composer, with the possible exception of Koji Kondo, to really elevate the form, creating dynamic, melodic tracks, something he considers to be the defining characteristic of his music, arranged well and with a strong, driving percussion and bass.

I suggested to him, with my tongue nudging my cheek, that in some respects his career trajectory was like that of the Beatles. Both had a conservative period at the start, where they produced safe crowd-pleasers as they found and established their own voice; a confident middle period, adventurous, but aware of their limitations and boundaries, and then a transgressive period, the White Album, very experimental and not always successful. He laughed self-deprecatingly… but then agreed, adding that that drive came from the need to continually innovate. There was always a need to try something different, something new. Partly that came from within, partly it was to stay ahead of the competition, but largely, it came from a facet of game development culture, which evolved from the first-generation hacking culture that preceded it. Neat ideas have currency. Neat ideas that push a system to its very limits have the greatest currency of all.

**<Change slide>**

**The ZX Spectrum**

It was also around this time, in the early 1980s, that Sinclair's ZX Spectrum took the UK by storm. Like the Atari, it had just 1-bit of sound resolution, but musically it was more restrictive still. It had just one channel of sound, and to compound matters, there was no sound hardware support. All of the sound was handled by the main processor, so while the Spectrum was beeping, it couldn't, without some clever coding, do anything else.

It's 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 monophonic title tune, and only limited in-game sound effects to punctuate key elements of the gameplay. Elite's *Chuckie Egg*, for example, is typical of this model, featuring the melody from *The Birdie Song* by the Tweets,

itself a cover of Werner Thomas's accordion tune, as its title music.

That musical diversity that I mentioned earlier, which makes it difficult to pinpoint a particular video game musical style, arose from the bedroom origins of the nascent industry. Quick turnarounds often meant that developers would adapt or lift whatever musical material was close to hand, often with little regard to the copyrights. Ben Daglish, the celebrated Commodore 64 musician, told me that:

*I had no idea that copyright existed. Quite seriously I really didn't. When we wrote all that Jarre stuff, 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.*

And it was everywhere. Sweevo's World, for example, features a beautifully-orchestrated carousel-style medley of marches, beginning with *Blaze Away* before moving into Sousa's *Washington Post March*, while Firebird's *The Wild Bunch*, which is loosely based on exploits of the 19th Century gang of outlaws rather than Sam Peckinpah's 1969 film, raids Ennio Morricone's back catalogue, featuring the pipe organ theme from *La Resa Dei Conti* as the musical segue into the game's showdown sequences.

One such act of creative appropriation lay behind Perfection Software's *Farenheit 3000*. Perfection wanted a title theme that would make an impact as soon as the game loaded, and programmer Peter Jones suggested using Bach's *Toccata and Fugue in D minor*, which he had heard opening the movie *Rollerball*. Working from the sheet music of Sky's cover, Jones coded a five-minute arrangement in Sinclair BASIC, before co-programmer Tim Williams converted it to machine code for the final game.

What makes the music in Farenheit 3000 significant is not so much the arrangement, which doesn't quite stick to either the Bach or the Sky sources, but rather the musical functionality of the piece 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 duophony by using the pedal note to continually reinforce that sense of tonality against the melody, the simplest example, perhaps of *virtual polyphony*.

The approach was used repeatedly from the second generation of Spectrum titles onwards. *Jet Set Willy*, for example, uses a similar technique in its arrangement of Beethoven's *Piano Sonata*

*no. 14, Moonlight*.

**<Change slide>**

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 at play than a sequence of monophonic square waves.

This approach, of sequentially alternating between melody and accompaniment to create a sense of two or more simultaneous parts separated in time, has its roots in the Baroque. In his *Prelude in Bb* from Book I of the *Well-Tempered Clavier*, Bach uses the same technique, here delineating the right-hand triples from the staccato bass. Yes, 350 years before the Sir Clive Sinclair created their medium, Bach was turning out Spectrum-style chiptunes.

**<Change slide>**

Even the musical Commodore posed challenges in this respect. With only three channels available, creating richness and depth was difficult. Martin Galway, nephew of the renowned flautist Sir James Galway, took virtual polyphony one step further, simulating full chords on one channel by rapidly arpeggiating the notes of the chord in sequence, collapsing down perhaps three or four channels into one.

**<Change slide>**

He first used the technique on the raggy soundtrack to Ocean's *Kong Strikes Back!*, and the sound very rapidly became a fundamental part of the video game music sound.

Ben Daglish took the idea to its logical extreme with his soundtrack for Gremlin Graphics' 1987 *Arkanoid* clone, *Krakout*, providing an implied bass, accompaniment and melody, all played at breakneck speed. Part of the joy of working on these machines, he recalls, was the sense of challenge that it gave. It forced composers to look for ways to circumvent constraints and limitations and to find ways to introduce dynamic movement and musical interest. Often, that meant harnessing the power of the computer as performer:

*Half the point of writing some of the music that I did,* he said, *of 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 apreggios.*

This illustrates one of the key differences between storing music in machine code and a traditional musical score. While a score is a very efficient way of communicating the macro elements of music, it relies on the performer interpreting those high-level musical directions and imparting the sort of expressive nuances, those un-notated micro elements of performance, that truly animate it. When the performer is a lump of silicon, any and all of that musical expression must be hard coded in advance. Every aspect of musicality depends on the composer having both an overarching sense of musical development, and the ability to translate it into the technical language of machine expression.

Meanwhile, back on the Spectrum, a young programmer from New Brighton in the Northwest of England would push virtual polyphony in a different direction.

Matthew Smith grew up around the mechanical fairground games in the seaside arcades of the town, and, like many school children, daydreamed of creating his own games, doodling on the graph paper of his maths jotters to sketch graphical designs in pencil. In 1983, he was loaned a Spectrum by Liverpool-based publisher Bug Byte to develop three games. The first, *Styx*, was a fairly simple action maze game. It was his second, *Manic Miner*, which became a runaway success, making Smith an unlikely superstar, and introducing the Spectrum's first truly iconic character.

**<Change slide>**

Manic Miner was based on *Miner 2049er*, a platform game that featured a Canadian Mountie, Bounty Bob, navigating his way through ten different screens before his oxygen runs out. Several elements of Miner2049er 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 humour to the level and character design, and a Pythonesque boot which descends to squash Willy when the game is over.

Smith realised that the restrictions with the Spectrum's sound system made it impossible to simply playback musical tones during the game, so doing would tie up the processor and bring the gameplay to a crashing halt.

Instead, he used an interrupt in the main game loop. Each time it executed, he played a fragment of a note for a fraction of a second — 20ms to be exact. In the remaining 70ms of the game loop, all of the other game code was executed. In effect, Smith was granularising the music sequentially, relying on the illusory continuity of tone to fill in the gaps between the sound grains.

**<Change slide>**

Smith went one step further, though. He realised that he could run sound effects alongside the music by similarly granularising those and slotting them into the spaces between the music grains.

**<Change slide>**

It's a novel solution, but, of course, it's not without cost, because the effect of breaking down the sound into grains like that gives it a disjoint, bubbly quality, and that, in turn, became an important part of the Spectrum sound.

Imagine's port of the Konami coin-op, *Yie Ar Kung Fu*, for example, uses the effect to play the main game stings in double-octaves, and *Dynamite Dan* uses the technique to recreate Mozart's *Rondo a la Turca*.

Durell Software featured two-channel granular music tracks on two of its 1986 releases, *Thanatos* and *Turbo Esprit*. The music on Turbo Esprit in particular is a fine example of the technique, its Jan Hammer-styled melody complementing perfectly the Miami Vice-like gameplay.

**<Change slide>**

**Summary and conclusion**

Now of course, I've only really been able to give a brief overview of some of the key moments in the early stages of development of this style of music. Developers continued to push the capabilities of all three systems, incorporating, for example, pulse-width modulation and sound sampling to create richer, more complex music. Eventually, however, technical progress overtook the sound chip, and with the arrival of Commodore's Amiga in the mid-1980s, and the growth of the PC as a gaming platform, video game music began to converge with film score, becoming more cinematic, using sample playback to create more realistic soundtracks.

The 8-bit sound did not disappear, however. It moved underground as part of the crack scene and later the demoscene, providing musical accompaniments to the digital graffiti that announced the illicit hacking of the copy protection systems on those same Amiga and PC games. In the days of dial-up and Bulletin Boards, with file size again at a premium, the 8-bit chiptune, with its ethos of squeezing maximal performance from minimal resource was the perfect accompaniment.

The style re-emerged in the late 1990s as the Game Boy was hacked and turned into what is, perhaps, the world's most accessible portable hardware synthesiser. Emulators and hacked code allowed a new generation of musicians to push the capabilities of these platforms, and demoscene meets and compos continue to provide a platform for expression.

Movie soundtracks, television advertisements and major exhibitions, at Bletchley and the Smithsonian to name but two, suggest a growing acceptance of chip music, alongside 8-bit video game art and animation, as a legitimate form of cultural and artistic expression. The adoption of elements of the style by major artists like Mark Ronson and Kylie Minogue also suggest that it is more than just a niche crossover. 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 while an 8-bit arrangement of the band's *Speed of Light* plays in the background.

In 2003, Malcolm McLaren declared 8-bit to be the new punk. It has that same, lo-fi DIY aesthetic, and, just as punk raised a defiant middle finger to the worst excesses of prog 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 today's commercial music. Little wonder, then, that one of the most popular slogans around the scene is 'Fuck Pro-Tools'. For those who wear that particular T-Shirt, the shape of music, it seems, is square.

**Transcipt of Author's presentation**