volume 8
february 2006
journal on media culture
ISSN 1567-7745

"Loops and bloops"

Music of the Commodore 64 games

by Karen Collins

volumes
Volume 15 / 2012-2013 —
Volume 14 / 2011-2012 —
Volume 13 / 2010-2011 —
Volume 12 / 2009-2010 —
Volume 11 / 2008-2009 —
Volume 10 / 2007-2008 —
Volume 9 / 2006-2007 —
Volume 8 / 2005-2006 —
Volume 7 / 2004-2005 —
Volume 6 / 2003-2004 —
Volume 5 / 2002-2003 —
Volume 4 / 2001-2002 —
Volume 3 / 2000-2001 —
Volume 2 / 1999-2000 —
Volume 1 / 1998-1999 —
Editorials and Op-Eds —

databases
Alan W. Pollack's Notes On ... —
Markus Heuger's Beabliography —
Rare Pictures from Radio's Past —
Zeezender Discografie —

dossiers
Beatles Studies —
Comics and Cartoons —
History of Radio and Television —
Local and Global Radio —
Offshore Radio Stations —
Popular Music Studies —
Rock Song Anatomy —
Studies in Photography —
Theory and Methodology —

The Commodore 64 was the greatest-selling home computer system of all time, and still draws a large crowd of retro-gamers. Despite its popularity, the music of C64 games has never before been analysed in any academic articles. This article is designed to introduce readers to the music of the C64. Discussing the technical constraints of C64's SID soundchip, Karen Collins shows the conventional ways in which the chip was used, compares some well-known, pre-composed songs which were covered on the C64 in various games and then explores approaches to interactivity and looping in Commodore games music. After comparing the C64 with its contemporaries, she concludes that the Commodore's music was a combination of both technological constraint and musical aesthetics.

1

Constraints on compositions. According to the *Guinness Book of Records*, the Commodore 64 was the best-selling home computer system of all time, selling over thirty million units in its production years from 1982 to 1993. [1] There were approximately ten thousand games released for the C64 over its decade-long reign, and many still remain popular with "retro gamers." One of the attractions to Commodore's games over those of its competitors was their unique musical aesthetic. With screaming guitar-like square wave solos, full-length songs, attempts to re-create traditional "rock band" line-ups in its use of tone channels, and its increased use of percussion, Commodore music was like rock to Nintendo's heavily looped disco aesthetic. There are vast archives of Commodore games music available on the web — notably sites like HVSC, the "High Voltage SID Collection" — and fans debate endlessly the best composers and games music on message boards. "Micromusicians" still use the Commodore to compose on, and relish — rather than regret — the technological constraints it imposed on its composers.

Technological constraints are nothing new to musical composition, although most discussions arising about the subject have centered on twentieth century concerns. Mark Katz discusses how the 78 RPM record led to a standard time limit for pop songs, and how Stravinsky famously tailor-made *Sérénade en LA* for the length of an LP; although he points out, however, that Stravinsky may have been shaped by "his penchant for self-imposed limitations" (Katz, 2004: 3-5). Critiques of "hard" technological determinism as it relates to musical technologies have dominated the discussion (e.g. Taylor, 1993: 27; Théberge, 1997: 160; Katz, 2004), in favour of a softer approach in which the relationship is more of a negotiation. As with other recent approaches to music technology, I would argue that the relationship between technology and aesthetics is one of symbiosis rather than dominance, what Barry Salt (1985: 37) refers to as a "loose pressure on what is done, rather than a rigid constraint."

Micromusicians tend to agree, and embrace the constraints as an important part of the creative process, as Teamtendo intimates: "Working with this limited harmonic vocabulary forces you to be creative, and there are some very pleasant discoveries along the way," [2] or, says Goto80, "it's fun working with such hardcore limits, forcing you to realize your ideas in other ways." [3] In order to explore the constraints of the C64 on composition, I first discuss the limitations of the sound chip, showing the conventional ways in which the chip was used. I follow this with a comparison of well-known, pre-composed songs which were covered on the C64 in various games. I then explore approaches to interactivity and looping in Commodore games music, drawing comparisons between the C64 and its contemporaries.

2

Right: Screenshot of Ghost 'n Goblins (1986); graphics: Chris Butler; program: Mark Cooksey; music: Chris Butler

The Commodore 64 SID. The Commodore 64 (C64) was originally conceived as a games computer, with advanced — for the time — graphics and sound designed to entice consumers scared off by the more business-like PCs. Unlike most games machines of the era, the sound chip — called the Sound Interface Device or SID — was specially designed by Commodore's Bob Yannes in 1981. [4] A three-tone plus noise generator, each tone on the chip could be selected from a range of waveforms-sawtooth, triangle, variable pulse (square wave), and noise. An independent ADSR amplitude envelope generator for each channel enabled the SID to more accurately imitate traditional instruments than other existing chips. The tone oscillators had a range of between 0 and 3995 Hz, approximately the same range as a piano. There were two 8-bit registers for each

channel controlling frequency — meaning 16 bits total, or 65536 frequency possibilities for each voice, so composers could "detune" notes if they wished.

Each tone could also be subjected to a variety of effects and programmable filters including ring modulation — unheard of on other sound chips of the time. The ring modulation option, which allowed the combining of information from two channels (the triangle plus a second waveform), was used commonly for sound effects like bells, chimes and gongs. The registers designed for modulation, however, were rarely used, since software could simulate them, without sacrificing the triangle form, although "for novice programmers they provided a way to create vibrato or filter sweeps without having to write much code." [**5**] Nevertheless, this option made it much easier for non-programming musicians to write for the C64.

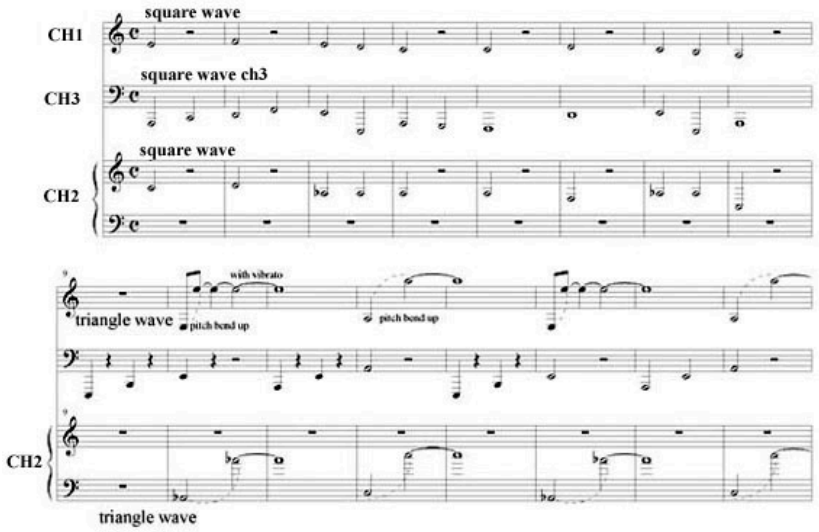


Figure 1: Ghosts 'n Goblins (Elite Systems, 1985) Commodore 64

Ghosts 'n Goblins (Elite Systems, 1985) is an appropriate illustration of the capabilities of the channel usage on the C64 in comparison with its contemporaries. Unlike the same game on the Nintendo, built on short loops, [**6**] the C64 game has only one song. Each channel changes voice throughout and imitates various rock instruments — such as the start of the song (shown above, Figure 1), which begins with square waves resembling electric guitar/electric bass, but which change to a triangle wave with rapid pitch bends and a theremin-like vibrato, extremely similar to the double-octave leaps on the theremin in *Mars Attacks* title theme (1997) by Danny Elfman.

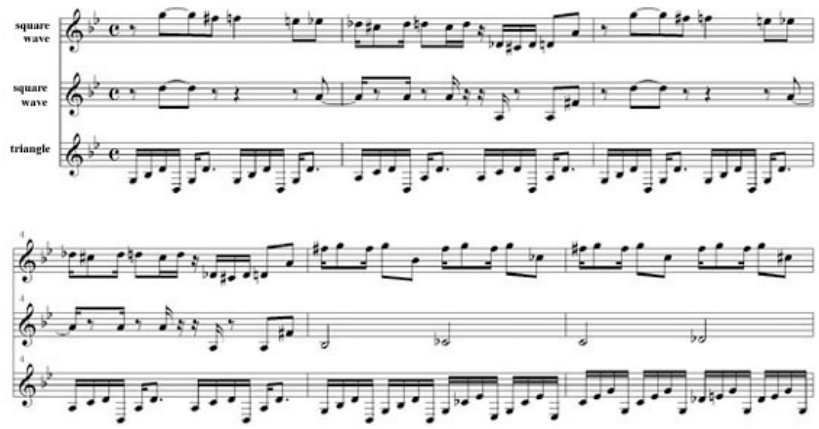


Figure 2: Ghosts 'n Goblins (Capcom, 1989) Nintendo NES

The noise channel — along with presenting white noise commonly used for percussion or sound effects — could also operate as a simple pulse width modulation (PWM) sampler (by modulating the volume of a voice so fast, a 4-bit sampled sound is created). PWM was used for sampling and to simulate an LFO to the volume — creating a tremolo effect, as heard on *Parallax* (Ocean, 1986). Martin Galway was the first to use sampled sounds on the C64, in the *Arkanoid* (Taito, 1987) theme song, as he explains: "I figured out how samples were played by hacking into someone else's code ... It was a drum synthesizer package called Digidrums, [**7**] ... I couldn't really figure out where they got the sample data, just that they were wiggling the volume register, so I tried to make up my own drum sample sounds in realtime — which is the flatulence stuff that shipped in *Arkanoid*." The ability to sample sounds led to the inclusion of somewhat realistic sounding sound effects in many game tunes. *Turbo Outrun* (Sega, 1989), for instance included a "scratch" sound and voice samples, and there were many examples of the samples used for more tonal percussion instruments.

Typically, however, drum sounds were made from a noise channel with a fast decay, and/or a square wave. [**8**] Unlike other sound chips of the time, the channels on the SID chip had the ability to change "voice," or sound wave type, at any point throughout a song, and so unlike on, for instance, the NES, we don't see a standard bass or treble sound, although a fat square wave or a saw wave seems to be most common for bass (see: Collins, 2006).

One of the biggest technological problems for C64 composers was the SID's filters (low pass, high pass, band pass and notch), which would act differently on different versions of the C64 machine. Yannes lamented, "The filter is the worst part of SID ... different lots of SID chips had different cut-off frequency characteristics. I knew it wouldn't work very well, but it was better than nothing and I didn't have time to make it better." [**9**] Attempts to overcome the problem vary. One game, *Beach-Head 2* (Access, 1985), even allowed the user to select the filter settings for the sounds, to prevent the screeching that would occasionally be heard with the wrong filter setting. Some composers chose not to use most filter settings, as Ben Daglish explained, "I tended to use "static" filters as little as possible for exactly that reason — generally, I'd use filter sweeps, which were pretty much guaranteed to have the same effect irrespective of the start/end frequencies." [**10**]

The limitations of memory were another major problem for games composers, along with the fact that the music was coded in assembly language. Commodore 64 software was distributed on three kinds of media: 5.25" floppy, datacassette tape, and cartridges. The datacassettes — the most popular storage medium for games — had built-in audio converters to convert the computer's digital information into analogue sound, though the loading of games on cassette was slower than it would be on floppy. The floppy disks provided a total storage capacity of 170KB, and of this, music was usually limited to between 5 and 10 KB. Rob Hubbard partially overcame the storage issue by arranging music for a game in series of modules, containing a set of songs. [**11**] Each module may contain title music, in-game music and game-over music using the same source code to share instrument tables to save space — that is, each different timbre used in the game was set out in a table in advance, and just called upon when needed.

Each song typically had three tracks (one for each channel), and each track made up of a list of patterns (sequences) and the order in which they were to be played. The code then would refer to specific sections of the module to be called when necessary, reducing the need to repeat any coding that would take up valuable space. As on other early machines, looping was commonly used when space was valuable: "The [song conversion] was a nightmare since it's the tune right from the beginning of the movie [*Short Circuit*] with all the robotic short notes and arpeggios. The tune just built up so massive [sic!] that the poor C64 was short of notes by about 30 seconds into it, so I had to fudge the end a bit and make it repeat, basically." [**12**]

3 **A comparison of songs**. Like other early systems, most Commodore games of the first few years had very little sound — and almost no background music. Of the "Top 100 C64 games," [**13**] about ten percent had no background music at all, and the earliest examples of games — 1983 to 1985 or so — had the least amount of music. For instance, *Boulder Dash* (First Star Software, 1984), *Bruce Lee* (Datasoft, 1984) and *Jumpman* (Epyx, 1983), had no music, apart from a short intro. Typically, music was only used when gameplay did not take place — the clearest examples of this are in the *Winter Games* / *Summer Games* / *California Games* / *World Games* series (Epyx, 1985), in which there was simple music for movie scenes and introductions to events, but the music stopped as soon as the event — and therefore gameplay — was begun.

The Commodore 64 was the home system to use the most cover songs of pre-composed music — in part because its advanced sound chip enabled more recognizable renditions, as the comparison with Beethoven on the VCS (*Acid Drop*) with the C64 *Jet Set Willy 2* version shows (Figures 3 and 4), although a few composers for the C64 favoured original tunes: "Covers, on the whole [were more difficult], simply because I was a perfectionist when it came to things like getting fast guitar solos right note-for-note ... and also because of the arrangement challenge of fitting a "real world" piece of music with drums and bass and strings and everything into three voices." [**14**]

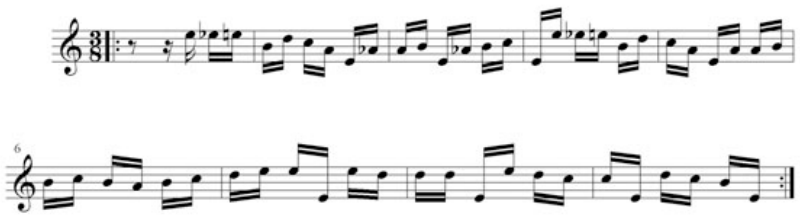


Figure 3: Acid Drop (Salu, 1992) Atari VCS using Beethoven's Für Elise

Western classical music was common in Commodore 64 games — such as Chopin's *Funeral March* (Sonata no. 2 in B flat minor OP 35) in *Zak McKracken* (Chris Grigg, LucasArts, 1988), Holst's *Mars, Bringer of War* and Bach's *Prelude no 2 in C Minor* (BWV847) in *Wicked* (Electric Dreams, 1989), and Rossini's *William Tell Overture*, Offenbach's *Orpheus in the Underworld Overture*, and Strauss' *Blue Danube* in *Microprose Indoor Soccer* (Martin Galway, Microprose, 1988). A comparison of the Commodore with the contemporary Atari VCS shows clearly the superiority of the C64 chip over that of the Atari, which was usually well out of tune (see: Collins, 2005b). The single channel of the Atari compared with that of the C64, as well as the timing issues and tuning problems, shows a marked improvement in the C64's technology.

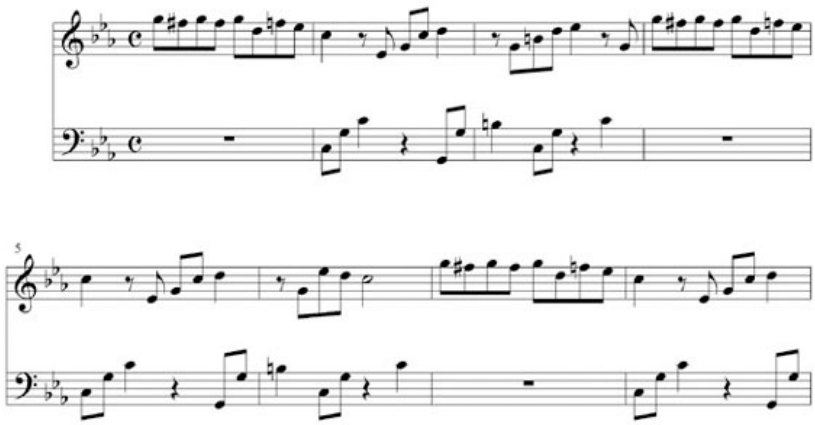


Figure 4: Jet Set Willy 2 (Software Projects, 1985) using Beethoven's Für Elise

More interesting, perhaps, was the mingling of popular and classical songs in games like *Frantic Freddie* (Commercial Data Systems, 1983), which used the Sylvers' "Boogie Fever," several Scott Joplin songs, Paul Simon's "Kodachrome," Beethoven's *5th Symphony*, Queen's "Crazy Little Thing Called Love" and ELO's "Don't Bring Me Down." Traditional American folk tunes of the 19th century were also very common in games, including *Buffalo Bill's Wild West Show*, for example, which used the "Star Spangled Banner" (Sir Francis Scott Key, lyrics, John Stafford Smith, music), "Yellow Rose Of Texas" , "William Tell Overture" (Rossini), "Shortnin' Bread" and "When The Chariot Comes" (traditional spiritual), "Yankee Doodle" (Richard Schuckburgh), "Buffalo Gals, Won't You Come Out Tonight" and "Camptown Races" and "Oh Susanna" (Stephen Foster). Many title themes were based on film music, such as *International Karate*'s use of "Merry Christmas Mr Lawrence" (Ryuichi Sakamoto), and *Super Pipeline*'s "Dance Of The Cuckoos," which was Marvin Hatley's title music for the Laurel and Hardy shorts, as well as Miklós Rózsa and Walter Schumann's "Dragnet Theme" (programmed by Paul Hodgson). The game also used Debussy and Paganini (Taskset, 1983).

It was rare to see the original composers credited in these games, and there was usually no credit given or information on licensing of songs for games at the time, although there were some exceptions, such as Devo's "Some Things Never Change," used in *Neuromancer* (Electronic Arts, 1988) and *California Games*' use of the Kingsmen's "Louie Louie" (Chris Grigg, Epyx, 1987), which both included a credit in the manuals. As Martin Galway commented on his use of Daglish's music for *Arkanoid*, "I'm glad you spotted "Cobra" on the Spectrum, whose tune I was in love with and HAD to use somewhere else...! I figured no-one would complain if I used it a year later on the C64." Cover tunes, then, seemed largely the whim of the games producers, and there was little or no concern for copyright infringements, as Martin Galway tells the story of one choice of cover tune: "I was still freelancing when I worked on that music first in 1984, and I just said to Tony Pomfret "what do you want for the music?" to which he replied "I want the B-side from the Limahl single "Neverending Story" I bought the other day."" [**15**] In other words, music was chosen without much consideration for the limitation of the chips or for copyright.

Left: Screenshot of the Great Giana Sisters (1987); graphics: Manfred Trenz; program: Armin Gessert; music: Chris Huelsbeck

Many early games which did contain background music would lack sound effects, or the music would have to stop for the effect. In fact, some popular games well into the late 1980s had little or no music at all. By the late 1980s and early 1990s — a time when the 16-bit machines (Sega Genesis, Super Nintendo, etc.) began entering the market — games were increasingly more likely to have in-game background music, although some remained without — such as *MYTH* (1989), and *First Samurai* (1989), which had no game-play music on the C64, but did a year later when it was ported to the Super Nintendo. This suggests that the pressure of competing games systems on sales of games for the C64 led to an attempt to adapt to the aesthetic of game audio present in other systems at the time. The fact that Nintendo NES had an — arguably — inferior sound chip, but had more gameplay music is most likely related to the storage capacity of the game cartridges. Whereas most C64 games averaged about 30KB (on cassette), 10KB (on cartridges) or 60KB on a floppy (as mentioned, to a maximum of 170KB on floppy before having to go to a multi-disc game), Nintendo's cartridges held up to 512 KB, and with the use of memory management chips, some games could be expanded, such as *Kirby's Adventure* (Nintendo 1993), which was 768KB — almost four times the capacity of Commodore's games.

Interactivity and looping on the Commodore. Despite memory constraints, there were certainly examples of early gameplay looping on the Commodore, such as *Frogger* (1982), which some people claim was the first Commodore game to use constant background music; a medley of traditional American Civil-War era songs. Generally

speaking, loops on the C64 were much longer than those of the NES — but this could be due to the fact that less songs were included overall, whereas the NES was more likely to have consistent background music. Loops occurred in many different shapes and sizes, as we will see below. It is perhaps useful to bring forth Middleton's concept of musematic repetition and discursive repetition — the "riff" versus the "phrase" (Middleton, 1996). We see both forms of looping repetition occur in games audio, most often at the same time, however we also see a longer loop of the entire "song" or sequence of loops at a larger-than phrase level. For clarity's sake, I will refer to these as microloops, mesoloops and macroloops. For example, a two-note bassline may provide a (musematic) microloop which repeats twice in a two-bar (discursive) mesoloop, which is then part of a longer eight-bar macroloop song which is looped throughout a level of gameplay.

Some composers were adventurous with the coding, and included random number generators into the code which would select from a group of loop options, something not seen on the NES. For instance, *Times of Lore* (Martin Galway, Microprose 1988) used a selection of guitar solos that were randomly selected for the eleven-minute duration of the song. In this way, the game's ten songs — over thirty minutes of music — could fit into just 923 bytes, and sound a lot more varied than it was. [**16**] A similar random generation was used in *California Games* (Epyx, 1987), and in *Rock Star Ate My Hamster* (CodeMasters, 1988), a rock management game which has the band practicing, with a tune which picks from a random combination of sixteen sequences, intentionally out of tune but improving as the band practices (Figure 5).

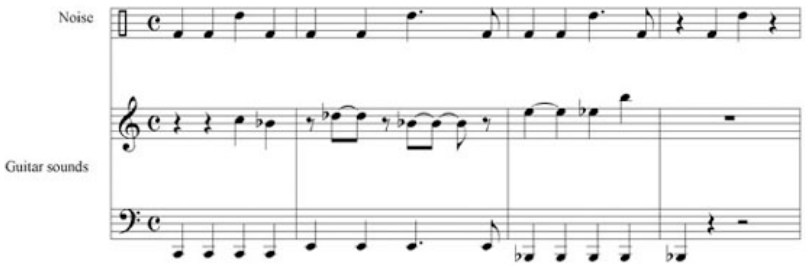


Figure 5: Rock Star Ate My Hamster (CodeMasters, 1988) Commodore 64

In *California Games* (Chris Grigg, Epyx, 1987: Figure 6), the music was similar to the NES version of the game, but it was set to typically stop during gameplay. Rather, we hear an intro to an event, and then just sound effects during gameplay — e.g. hackey sack, frisbee and skating, which have looping background music on the NES. A few events have musical loops — such as surf, which also has Jaws music when the player falls off. The interactive use of looping happens in the halfpipe event, in which an opening sequence plays for seven seconds, and then selects a random sequence to follow — from sixteen possible choices, each twenty-three seconds long — as long as the player stays upright on his/her board. If the player falls, the first segment repeats.

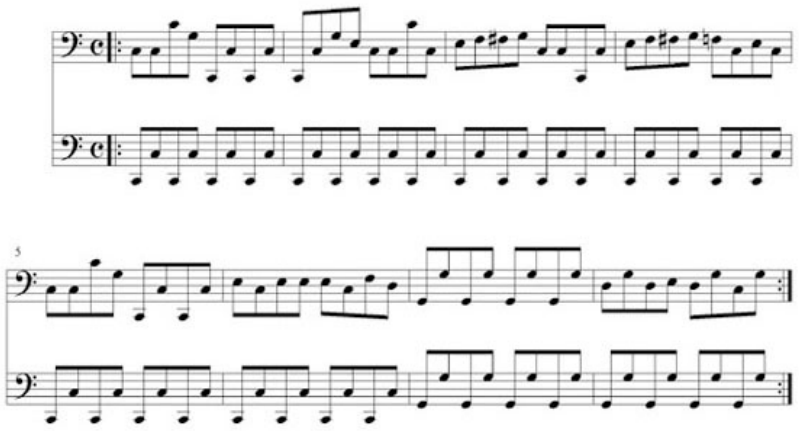


Figure 6: Lazy Jones 8-bar track one (main gameplay) (Terminal Software, 1984)

An effective example of looping on the C64 is the game *Lazy Jones* (Terminal Software, 1984), which had 21 mesoloops, each of which were selected when the character entered or left one of the "room" levels of gameplay. There were 18 rooms in total, and each room had its own 4-bar "song" or mesoloop, which actually played like a segment of one greater macroloop song (the title music). Even if the character left the room at, say, bar 21, the rest of the loop would play before it would transition seamlessly into the theme song. Most of the loops worked well together, in part due to the ragtime-like microlooped basslines, the same timbres used, and the fact that the game only used two channels (Figures 6 through 9).

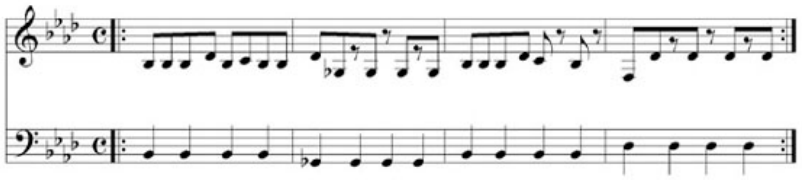


Figure 7: Lazy Jones 4-bar track two (Terminal Software, 1984)

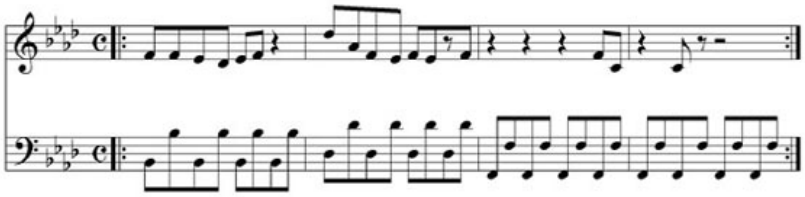


Figure 8: Lazy Jones 4-bar track three (Terminal Software, 1984)

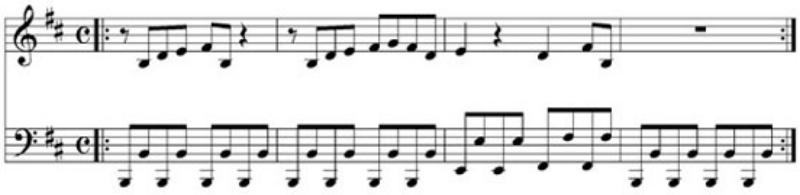


Figure 9: Lazy Jones 4-bar track twenty-one (Terminal Software, 1984)

By the late 1980s and early 1990s, there were several Commodore games which clearly tried to emulate the Nintendo game style, such as *Mayhem in Monsterland* (Apex, 1993), or the *Great Giana Sisters* (Rainbow Arts, 1987), which was so similar to *Super Mario Brothers* that Nintendo successfully sued to have it pulled from stores. Each game contained similar loops for "overworlds" and "underworlds," and "boss" music, in Nintendo style, with longer overworld music, and short boss music loops. *Great Giana Sisters*, for instance, follows Nintendo's looping style with the main overworld level, and *Mayhem in Monsterland* has a short four-bar boss music pattern (Figure 10). It is interesting that these games not only adopted NES-like gameplay and visuals, but adopted a distinctly NES style music, suggesting an associated well-defined aesthetic was involved, rather than technological constraints implied by the sound components.

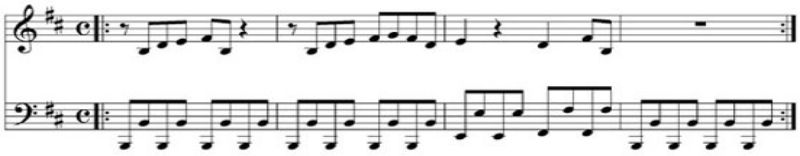


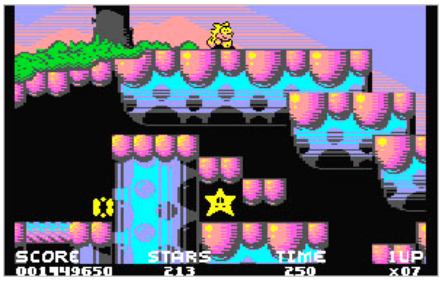
Figure 10: Mayhem in Monsterland boss music (Apex, 1993)

Despite Nintendo's influence, Commodore also maintained a highly unique aesthetic unseen on other games systems, which included incredibly long tracks. *Tetris* (Mirrorsoft, 1987), for instance, was very different than the versions released on the NES, showing this very different aesthetic particular to the C64. Not having any selectable music (which was an option on the NES), Wally Beben composed all original music — one very long (about 26 minutes — 13Kb) track of many segments. In order to save space (likely), certain micro and mesoloops of the track repeat: for instance the bass/percussion line that begins the song repeats just one bar for about half the track, with different melodies coming over top and being layered with various accompaniments. This accumulative form — the gradual building up of a groove by adding sequential units cumulatively (Spicer, 2004) — was closer to the electronic trance music beginning to emerge in the late 1980s than any game music aesthetic of the time.

[optional insert: Tetris SID: note: needs Sidplay or Sidamp to play]

Right: Screenshot of Mayhem in Monsterland (1993); graphics: Steve Rowlands; program: John Rowlands; music: Steve Rowlands

The C64 musical aesthetic. The different musical aesthetic of the Commodore 64, including the use of "guitar-like" sounds and more attempts to re-create traditional "rock band" line-ups in its use of tone channels, and its increased use of noise and PWM for sound effects and percussion is closely tied to the technology:



"*Well, you know, part of that* [sound aesthetic] *is dictated by the fact that you have such limited resources. The way that you have to write, in order to create rich textures, you have to write a lot of rhythmic kinds of stuff. ... it's easier to try to make it sound a lot fuller and like you're doing a lot more if you use much shorter, more rhythmic sounds*." [**17**]

The persistent practice of looping is particularly illustrative of the tensions between technology and aesthetic. As was seen, alternatives to the standard macro-looping practice of games music of the era were certainly available — and, at times, used, as in *Tetris* — although looping remained the most prominent response to a limited amount of memory. Looping, then, was a combination of both aesthetic choices and a pre-determined factor led by technology. Even within games that used looping, as was seen, there were many different responses and approaches to looping, with repetition occurring in many different forms. As Katz discusses in an unrelated matter, "If nothing else, the diversity of responses to repeatability should dispel any notion of strict technological determinism, for such wildly disparate phonograph effects demonstrate that there can be no simple cause-effect relationship between recording technology and the activities of its users" (Katz, 2004: 31). The Commodore 64 clearly offered enough options for composers to explore that its musicians created their own aesthetic out of the constraints imposed upon them, an aesthetic influenced by outside sources, but one which was also unique to the C64.

Notes

1. The real number may rather be in the 20 million range — still an impressive figure (Matthews, 2003). ⋀

2. Katigbak, 2004. ⋀

3. Car, 2002. ⋀

4. Yannes had also helped to engineer Commodore's VIC-20, and would later go on to create the DOC chip for the 16-bit Apple IIGS and to found Ensoniq keyboards. ⋀

5. Yannes in Varga, 1996. ⋀

6. Here, I take loops to mean "self-contained units that may or may not be combined with other loops or non-looping material in a larger structure ... In music production and sound design circles, a "loop" is a bit of audio — usually, though not exclusively, of short length — that can be played back repeatedly and potentially endlessly without noticeable gaps or disruptions between one instance of the loop and the rest" (Stillar, 2005: 199). ⋀

7. Made by Digitunes, who went on to create Protools. ⋀

8. See Taylor, 1993. ⋀

9. Yannes in Varga, 1996. ⋀

10. Composer Ben Daglish in Pouladi, 2004. ⋀

11. The way songs were written can be seen by using "Ransid," a tool for analyzing SID tunes by disassembling the assembly code. There are also a few examples with explanation in **Sid-In Magazine**. ⋀

12. Composer Martin Galway in the **Sid Tune Information List** (STIL v3.4). ⋀

13. According to downloads from **C64.COM**. Retrieved from the internet on October 14, 2005. ⋀

14. Composer Ben Daglish in Pouladi, 2004. ⋀

15. Martin Galway in the **Sid Tune Information List** (STIL v3.4). ⋀

16. Martin Galway on Times of Lore in **SIDfind**. Retrieved from the internet on November 13, 2005. ⋀

17. Beck, 1999. ⋀

References

- Beck, Justin (1999), "Rob Hubbard interview for the Commodore 64 music radio programme 6581SID on KDVS." In: **The Complete Works of Rob Hubbard**. Retrieved from the internet on November 23, 2005.

- Car, Neil (2002), "An interview with Anders Carlsson aka GOTO80." In: **remix64.com**. Interview date: September 18, 2002. Retrieved from the internet on November 17, 2005.
- Collins, Karen (2005a), "From bits to hits: video games music changes its tune." In: *Film International*, 12, 2005.
- Collins, Karen (2005b), "Minor seconds and the musical aesthetic of the Atari VCS." Available on: **Karen Collins: Research and Publications**.
- Collins, Karen (2006), "Repetitive refrain injury? Audio looping in the music of the Nintendo NES." Available on: **Karen Collins: Research and Publications**.
- Katigbak, Raf (2004), "Game on!" In: **Montreal Mirror**, 20, 18 (October 21-27, 2004). Retrieved from the internet on November 14, 2005.
- Katz, Mark (2004), *Capturing sound. How technology has changed music*. Berkeley: University of California Press.
- Matthews, Ian (2003), "The Commodore 64: machine of destiny." In: **www.commodore.ca**.
- Middleton, Richard (1996), "Over and over: notes towards a politics of repetition. Surveying the ground, charting some routes." *Conference paper* for Grounding Music, 1996. Available on: **Forschungszentrum Populäre Muzik: Texte und Materialien zum Studium der populären Musik**. Retrieved from the internet on October 10, 2005.
- Pouladi, Ali (2004), "An interview with Ben Daglish." In: **Lemon 64**. Retrieved from the internet on October 14, 2005.
- Ross, Rob (2001), "Interactive music... er, audio." In: **Gamasutra**. Retrieved from the internet on September 2, 2005.
- Salt, Barry (1985), "The evolution of sound technology." In: Elisabeth Weis and John Belton (eds.), *Film sound: theory and practice*. New York: Columbia University Press.
- Spicer, Mark (2004), "(Ac)cumulative form in pop-rock music." In: *twentieth-century music,* 1/1, 29-64.
- Stillar, Glenn (2005), "Loops as genre resources." In: *Folia Linguistica*, XXXIX, 1-2, 197-212.
- Tagg, Philip (1994), "From refrain to rave: the decline of figure and the rise of ground." In: *Popular Music*, 13, 2 (May 1994): 209-233.
- Taylor, Craig (1993), "Editor's Notes." In: **Cracking**, 5, March 7, 1993. Retrieved from the internet on October 12, 2005.
- Taylor, Timothy (2001), *Strange sounds: music, technology and culture*. New York: Routledge, 2001.
- Théberge, Paul (1997), *Any sound you can imagine: making music / consuming technology*. London: Wesleyan University Press, 1997.
- Varga, Andreas (1996), "Bob Yannes interview." In: **SID In-Depth Information Site**. Retrieved from the internet on October 12, 2005.