Arcade Videogame Interface Aesthetics

Kieran Nolan
GV2 Research Group, School of Computer Science and Statistics, Trinity College, Dublin, Ireland.
kinolan@tcd.ie

Abstract

This paper examines the aesthetic and connective properties of arcade videogame interfaces. It considers the arcade videogame interface as a communicative and creative link that extends beyond play orientated input and feedback mechanisms. With the correct emulation and homebrew tools, videogame platforms that were originally designed as consumer only devices become malleable forms that can be interfaced at artist and developer levels, allowing previously closed hardware and software systems to function as reconfigurable digital materials.

Communication within constraints

Videogame interfaces ideally create a symbiotic connection between the participant human(s) and the computer that are participating in a digitally centered game experience. While an interface is not the game itself, for example a chess game may have a text-based interface, it does fundamentally affect how the game is experienced. [1]

This experience encompasses the complete aesthetic design of the overall product, visually, physically, sonically, and communicatively. Technically the host platform will impose a set of affordances and constraints for the videogame designer, who presents the results of working within these parameters to the end user. These factors include graphics resolution, sound bitrate, processor speed, and the nature of input devices. The physical styling of the hardware device's form factor also bears influence on its game experience, whether used as a portable, domestic space based, or public situated play experience.

The communicative aspect of the interface is critical, as the player must understand the machine, and vice versa. This is summed up succinctly in Chris Crawford's definition of interaction as "an iterative process of listening, thinking, and speaking between two or more actors". [2]

For example, after a player inserts their coin into a videogame at an amusement arcade, they are presented with a text prompt through the machines visual interface to select either single player or two-player mode. To articulate back their choice of single player mode, the player selects the button labeled '1'. An audible chime from the speakers and a change in the visual interface that affirm the chosen option are then relayed back to the human as sonic and visual feedback by the computer. Without this feedback, there is a disjoint in the communication, and as a result the interface is rendered defective.

The constraints that govern the flow of this dialog affect the viewpoints of both the human and machine participants. From the meatspace occupant's point of view, how the computer communicates is limited by its processing power, the fidelity level of its graphics and sonic capabilities, alongside any other feedback mechanism available to it. From its own vantage point the computer perceives an incomplete composite of human sensory faculties via the limited input means assigned to it. If the digital organism's senses are augmented through the addition of extra sensors, for example the ability to detect gestural movement, this subhuman composite evolves closer to resembling the complete human being. This hobbled interpretation of the human in the eyes of the computer based on it's restricted sensor capacities provides a view of an leveled relationship between both parties, comparable to how humans may interpret a suprahuman machine intelligence when it emerges.

A videogame's communicative anatomy comprises of two elements, input and feedback. The input side is most commonly associated with the controller-centric perspective of game interfaces. This view encompasses manual controllers such as the keyboard or joystick, alongside glass based gestural interfaces, in addition to freeform gestural interfaces, for example, Nintendo's *Wii* Controller and Microsoft's *Kinect*. These can be seen as consumer level game interfaces. The game developer interacts with the host technology at a more raw and visceral level, where the inner workings of the computing system are exposed and directly addressed through code.

Feedback forms the second component of the transmission loop. This feedback is most commonly delivered through visual, auditory, or tactile means. The harnessing of the olfactory senses as feedback targets for digitally originated feedback in videogaming has yet to reach consumer level. In game design the vibrating 'rumble' feature of game controllers is common method used to enhance immersion by adding the extra tactile dimension of haptic feedback. For example, The Playstation 2 version of Sony's *Rez* (2001) uses a dedicated tactile feedback called the *trance vibrator* unit to provide feedback accompanying its psychedelic visuals.

In terms of visual feedback, the Graphical User Interface adds a layer above the 'developer only' interface of the command line. This further abstracts human interaction directly with the metal of the computer.

The term GUI is typically used to describe a point and click style operating system interface, but in the videogame context the graphical interface is not limited to mouse control. A visual interface is delivered mainly through the screen, but also can refer to other visual augmentations to a videogaming system, such as the flashing lights on the *APB* arcade cabinet (Atari, 1987). In *APB* the player drives a police car and must arrest a set quota of criminals and traffic law violators on each level. The player must also collect doughnuts to increase their high score. Whenever the player is in pursuit of a perpetrator's vehicle, the siren lights on the cabinet activate, adding to the games immersion.

Galloway cites Vilém who described the screen as "a two dimension plane with meaning embedded in it or delivered through it" [3] This meaning can be textual, image based, both abstract and figurative. Galloway also uses the term of the intraface to describe the ether that holds together the various layers of the visual interface, the interplay of static and diegetic (or dynamic) interface elements.

Static game interface elements are those that cannot be interacted with directly, that exist to solely provide visual feedback. For example, in the game *Asteroids* (1979), the static interface elements are the current score and the number of lives remaining. Dynamic or diegetic interface elements are feedback elements that are incorporated into the game environment itself. In the *Grand Theft Auto* series, damage that is visually manifested on the player's vehicle provides dynamic feedback to the user on how effective a driver they are, rather than relying on a numeric value.

The division between screen based visual interfaces and the manual interface of the hardware controller has become less distinct since touchscreen technology became a viable part of modern consumer electronics. The screen has become a new form of tangible surface, despite its limited tangible feedback capabilities. The ability to reach deep into the screen physically in commercial video games, in a reverse of *Videodrome*'s signature hallucinogenic sequence, can only be done through virtual rather than physical augmentation of the visual display. The glass barrier ensures that games presently remain solid to physical touch, rendering the play field only malleable through extension into the digital space visually through a digital embodiment, or sonically via three-dimensional sound.

While remaining intuitive and clear, a game interface must engender a sense of challenge. This level of difficulty is dependent on the audience as well as the host platform. A casual gamer indulging in a five-minute game on their smartphone requires simple, intuitive controls. On the opposite end of the spectrum are those who thrive from the intrinsic challenge of mastering a complex experience, taking satisfaction in mastering a task that requires more rote learning than intuition. Take for example the text mode visual overload of *Dwarf Fortress*, where the player must invest months to master the game control system, memorising a considerable number of keyboard shortcuts while navigating and understanding a myriad of statistic packed status screens.

As technical advances bring the digital closer and closer to analogue realism, the digital has already bled through into analogue culture. This essay is intended as an introduction to the area of videogame interface aesthetics, reflecting on how the link between people and low fidelity videogame technology has manifested through the pixel art, chiptune, retro gaming, and media art.

Building an interface for play

One of the earliest documented participatory videogames, and interfacing of the general public with computers as a mode of leisure took place in 1958 at Brookhaven National Laboratory, a nuclear research facility in Suffolk County, Long Island. William Higinbotham, a nuclear physicist who had worked on the Manhattan project, and afterwards became a campaigner for nuclear non-proliferation, led the team that built *Tennis for Two*, an interactive exhibit intended for public engagement. "I knew from past visitor's days that people were not much interested in static exhibits, so for that year I came up with an idea for a hands-on display, a video tennis game". [4]

Housed at the display for BNL's Instrumentation Division, *Tennis for Two* used a circular 5 inch diameter oscilloscope screen as a visual display. Auditory feedback was provided through a basic sound effect that is heard when the ball is batted. The use of an oscilloscope as *Tennis for Two*'s graphical display unit enforced a monochromatic, stripped down line drawn aesthetic, that can be seen taken in a more intricate form to the z-axis in *Battlezone* (1980), and *Vib Ribbon* (1999).

Brookhaven National Laboratory's mainframe computers were imposing machines that filled entire rooms. From the general public's point of view, the fact that these new and largely technologies were housed at a nuclear research facility during the cold war added to their ominous nature. Indeed the computer used was an analog computer, programmed laboriously through punch cards. It was a machine built by computer scientists for computer scientists.

By building *Tennis for Two*, Higinbotham introduced play to early human computer interaction, easing any misgivings felt by humans interacting with a new technology for the first time. A vital characteristic of a gaming interface is that it establishes a communicative link between player and machine, which can be measured through user engagement. The successful dialog accomplished by Higgenbotham's team was made evident by the long lines of people who waited in line to play the game during the 1958 BNL visitor's day. However, Higgenbotham didn't seek to commercialise his invention, and Tennis for Two remained largely unknown until 1985 when Nintendo cited Bayer's *Tennis for Two* as the original tennis videogame in response to been sued by Magnavox.

Videogame interface anatomy

Tennis for Two is important in the early lineage of video gaming as it set out the basic template for a two-player arcade game interface. The game's two manual controllers were built using industrial potentiometer knobs for onscreen paddle movement alongside a serve button, each of which was encased in stainless steel boxes connected via cables to the computer. Alongside the laboratory oscilloscope and the bulk of the analog computer, the physical visage of the machine is very much representative of the cold war military industrial complex that is was born from. Its gameplay mechanic and dual controller physical interface, alongside the audio and visual feedback elements were later echoed by Ralph Bayer's *TV Game System* (1966). Bayer's invention was later commercialised for the mass market as the *Magnavox Odyssey* (1972).

In 1985 Japanese game manufacturer Nintendo manufactured and sold their own videogame tennis system, and were sued for patent infringement by Magnavox. Nintendo responded by calling Higinbotham as a witness, arguing that *Tennis for Two* was the original tennis video game. The judge ruled that the since the oscilloscope display was not receiving a standard video signal, it therefore could not be considered a 'videogame'. At the time, a mixture of naivety and legal obfuscation was enough to ensure this minor technicality exempted Tennis for Two from falling under the definition of videogame. However, signal transfer technologies for video display continue to evolve, from the analog radio frequency cable to the binary datastream of digital and internet based television. Despite the judge's ruling in the Mangnavox Vs Nintendo case, the *Tennis for Two* computer was indeed sending instructions to the oscilloscope, and although it did not meet the same technical specifications as a televisual video signal sent to standard nineteen seventies cathode ray tube television set, it nonetheless defined and updated the visual display on the unit.

Atari's *Pong* (1971) took the TV tennis concept and translated it to the amusement arcades, it also (predictably) resulted in Atari been sued for copyright infringement by Magnavox. It was also the second commercially sold coin-operated arcade videogame. The first of which was *Computer Space* (1971), which was also produced by Nolan Bushnell before starting Atari. Similar to Pong it was a close copy of an existing game, a lab experiment called *Spacewar!* that originated in 1962 at Massachusetts Institute of Technology as a tech demo and experiment for the DEC PDP-1 computer. The extravagant molded fibreglass body of the cabinet reinforced the science fiction scenario of the game, even making an appearance in the film *Soylent Green* (1973) as a futuristic home fixture. Despite high production values, *Computer Space* failed to translate to the public space. It's control console and onscreen display elements did not communicate their function clearly to the user, and the game demanded more time to learn than a single play would allow. *Computer Space* had an interface that preformed all of its functions correctly from an engineer's point of view, there was a button for each action required of the player. Unfortunately, the visual mapping of the buttons and the lack of a pre existing visual language for videogames that the layperson could connect with, lead to *Computer Space* failing commercially.

By contrast, the zen like simplicity of the *Pong* interface ensured it's success. Pong's arcade cabinet design is minimal with clean typography, the wooden veneer body and stainless steel control panel striking a balance between the industrial and 1970s home decor. A single line of instructions that reads 'Avoid missing ball for high score' was placed on the front of each cabinet between the two rotating potentiometer controllers. The mapping of the physical controllers to their respective onscreen paddles is made obvious to the player, while turning each knob clockwise or anti-clockwise generated an immediate and appropriate response, changing the vertical position of the player's onscreen paddle. Simple audible tones accompany the sound of hitting the ball, and also missing the ball. The ball itself isn't round, but square, the beginnings of the pixel aesthetic. Pong sold heavily, generating more royalties for Magnavox than the *Odyssey* console had.

Looking inside of a *Pong* cabinet we see the interface construction laid bare. The circuit board hangs attached to the side of the cabinet, with wires reaching outwards to the television monitor, controls, power supply, and coin mechanism. It is unembellished and utilitarian, a nod back towards the industrial feel of BNL's *Tennis for Two*. The outward face of the game presents a polished visage in comparison to the exposed raw components that power it. In effect the cabinet front is a boundary both physically and visually, separating the player from the electromechanical and computational processes that power the interactive game experience. This view behind the curtain is denied to the general public, with only the game technicians and the arcade owner making contact with the core mechanics of the system. While the developer uses machine code writes directly to the metal, here the coin-op owner or technician physically views and interacts with the machine circuitry.

Following on from the *Pong* arcade game was the home television console version of *Pong*, which in turn inspired further copies and adaptations of the TV tennis concept. A simple and refined game interface had kickstarted both the home and arcade videogame markets. This new form of digital engagement would influence the formation of creative and cultural movements in the digital and analog spaces, bringing computer technology from the science laboratory to the domestic space, while influencing thought on speculative scenarios involving the merging of both domains.

The emulator as interface

The original *Pong* arcade machines are now in museums and private collections. If a visitor to the Computerspeilmuseum in Berlin wants to play *Pong* they need to arrive at a set time where they will be allowed to play the delicate artefact with supervisor assistance. For someone who doesn't have access to the original Pong hardware, the next best option for experiencing the original game is through an emulator. An emulator is a piece of software or hardware that recreates a computer system through a different computing platform. For example, the Commodore 64 home computer can be recreated in software form on a modern Apple Macintosh using the emulator *VirtualC64*.

Most classic videogame consoles and home computers exist in emulated form, which are usually created by enthusiasts without any official backing from the original machine creators. Legal difficulties do exist, and to circumvent these issues most emulators are distributed without core proprietary code. Users are typically asked to sign a disclaimer stating that they are going to fill the gap by legally extracting the code from the rom chips or system disks of their own working machine. Yet in practice most users will simply conduct a web search to find the missing files for free through a less than legal source.

Pong is one of the few anomalies in videogame emulation since it wasn't actually programmed but 'built' electronically as an analogue circuit. In regular emulation, the original program code is 'dumped' as a rom file taken direct from the original game circuitry in the case of a cartridge or arcade PCB system, or extracted from the original tape or disc. This program rom is then loaded into an emulator for the chosen classic system, just as an original disk or cartridge is loaded into the original hardware. Recreating an analogue videogame through code is directly comparable to the creation of virtual analogue synthesizers, where the circuit and the voltage flow through its components are modeled through code. *DICE* is one such program that simulates the original *Pong* arcade games on a home computer.

Modern pixels are too square

It is necessary to archive and preserve digital culture that has not already been committed to the online space. Physical machines and data storage mediums are not impervious to time and wear. Bitrot, the phenomena of magnetic storage mediums losing vital sectors of their information due to degradation of their physical makeup, proves that the digital world is not impervious to the environmental forces of the analogue world. However, recreating a classic arcade game on a modern computer works to the extent that an ebook can recreate a print book. Using an original tangible controller with an emulator through a custom adaptor, for example the Atari 2600 paddle control with the Atari 2600 emulator *Stella*, can closer approximate the original experience of using 2600 as it was originally played.

The kindle ebook reader provides an immaculate paper white screen, and the ability to store many books in a compact physical form. Subjectively this can be interpreted as convenience at the expense of character. A paper based book takes up more room, it can be easily torn and the paper fades and changing colour over time. Whether or not these weaknesses of paper books are seen as a hindrance or welcome quirk, they remain a part of the experience of interacting with that medium. Similarly, when a game intended for original viewing through a cathode ray tube based screen is viewed on the immaculate high resolution of a modern computing system, convenience is added, but also part of the original medium's essence is lost.

Admittedly, CRT screens are inconvenient on several levels. Compared to flatscreen monitors, they are bulky in size and weight, and their low resolution makes it difficult to focus on smaller details onscreen. Additionally, the scanline interlace flickering of tube base video screens can induce eyestrain. Also when using an analog signal, if a lower quality video connection cable is used that doesn't separate the red, green, and blue component video signals, the image becomes less defined.

Game designers have positively and creatively harnessed these imposed constraints and colourations. An example of this in effect is shown in the transparent waterfall graphics in *Sonic The Hedgehog* on the Sega Megadrive, made possible by blurring two shades of light grey into a semi-opaque effect. When viewed through an emulator without CRT simulation the effect is lost. [5]

In comparison to Sega's Megadrive, the Atari 2600 was extremely limited in it's visual display options, and was originally designed to display only two player objects on screen at once, in addition to 'bullets'. A machine engineered to play Pong derivatives. To overcome the system's limited sprite display capabilities, programmers relied on the ghosting effect of CRT phosphor and the human eyes persistence of vision. By switching images on and off quickly onscreen, it was possible to create the effect of multiple characters onscreen.

In 2009 a team of developers from Georgia Tech developed a software filter for the *Stella* emulator that simulates a cathode ray tube screen. [5] The parameters of the CRT effect can be adjusted to change the images curvature and blur, distorting the outputted video signal to the viewer's own requirements. This destructive yet aesthetically enhancing form of image filtering can be paralleled to the glitch art movement. If bitrot is the random encroaching of analog environmental variables into the digital world, then the creation of distortion and glitches is a deliberate unsettling of the predictability of digital code execution.

Emulation, the ROM hack, and vintage platform enhancements

Aside from recreating the features of a vintage computing platform, many emulators add extra functionality that serve to expand the recreated platform beyond it's original uses, making direct to metal processes open to the consumer that in turn enable them as creators. *NESticle* version x.xx (1998) by Bloodlust Software emulates the Nintendo NES system. [6] When a .NES rom file is loaded into the emulator it can be viewed in tile mode. Tile mode displays the component graphical elements of the loaded game rom, alongside the HEX values for their assigned memory locations and other binary variables such as their colour palette. If the rom file for *Super Mario Bros.* is loaded and the tile view is enabled, it is possible to locate the graphic tiles that are used to compose the sprite for the game's main character, Mario, and redraw his appearance using the emulator's built in image editing tools. The edited rom file, known as a rom hack, can then be saved and ran as a normal game. It is also possible to take a hacked rom image and transfer it to a physical rom chip, allowing the modified code to run on the original system hardware.

Through *NESticle* and other rom hacking tools, it is possible to edit and experiment with the visual interfaces and control systems of classic video games. Some rom hacks are purely functional and address existing issues in the game, for example, making a Japanese language only game accessible to a wider audience by replacing the original dialog with English language substitutions. Other rom hacks go down a more conceptual route; *Super Mario Clouds* [7] by Cory Archangel is a seminal example of the rom hack as media art. *Super Mario Clouds* is a rom hack of the game *Super Mario Bros.* (1985), reducing the game down to a minimal level, where all visual interface elements are rendered invisible, apart from the sky and clouds.

The early home computer systems such as the Commodore 64 presented the command line of the BASIC programming language as their initial start up interface, encouraging users to creatively code for their machines, however the game consoles were intended solely as playback devices. Emulators recreate this core software interface, but also expand on it.

In the cases of Nintendo and Sega, only licensed developers were permitted to develop for their systems. Another factor that made creating for these platforms so exclusive was that specialist development systems were required for the task, which were not on sale to the general public. In addition, the game cartridges for console systems were developed strictly under license from Nintendo or Sega, although bootleg cartridges and pirate 'backup' copy devices existed on the grey market.

As with the emulator scene, modern development systems for classic videogame hardware are made by enthusiasts with detailed and often self-taught knowledge of the software and hardware architecture of the original platforms. These software interfaces provide access to previously inaccessible aspects of the console platform, and ease the development process for the classic computer systems. For example, *VirtualC64* presents a series of advanced debug options that allow a programmer to examine and modify a game as it is running.

Complimentary to software emulation and homebrew development kits, vintage gaming hardware is augmented into the modern age through the addition of third party peripherals. These devices add functionality such as USB file storage, network connectivity, and high definition enhanced audio and visual output. The HD output options contradict the aesthetic ideal of the CRT software filter, showing that the phosphor blur is a nostalgic as much as aesthetic preference.

Emulators and homebrew software development kits, alongside modern hardware additions for classic videogame systems ensure the continued survival of vintage computer gaming platforms. Authenticity is a prized attribute in emulation, whether exemplified as the fuzzy RF visuals on a tube based television set approximated in *Stella*, or the unseen but consciously valued process of analog circuitry simulation in the *DICE Pong* recreation. In these instances, the interface effects of platform constraints are embraced, but expanded beyond when necessary for creative purposes.

Homebrew videogame music interfaces

In 1989, Nintendo released the Game Boy portable gaming console. Designed by Gunpei Yokoi, it was an evolutionary step beyond Nintendo's Game and Watch series of LCD based portable games, a videogame equivalent of the Sony Walkman, allowing access to a library of interactive games on the go. The system's most basic revision, the Model DMG Game Boy (Dot Matrix) is equipped with four sound channels (two pulse waves, a 4-bit wave channel capable of playing samples, and a noise channel).

Apart from the nostalgic view, there are several reasons that the Game Boy, and low fidelity computer music sound has been embraced by modern musicians. One is the appeal of working within constraints rather than navigating a bewildering interface of feature bloats in modern music software. Another factor is that the Game Boy provides an affordable alternative to pricey electronic music equipment. Finally, the sound from Nintendo Game Boy has a raw, unfiltered edge. This primitive level of sound synthesis is immediately associated with vintage videogames, and has been adopted into the high-end systems that it rebels against.

Visually it can display four colours onscreen simultaneously, and is controlled through a simple four way directional pad alongside four action buttons. While the visual display was improved in clarity over a series of system revisions up to the Game Boy Color (1998), the basic control and audio system remained the same. For over twenty years, the system has endured as a pop culture motif to the point that custom cases are available for high-end smart phones that camouflage them into the likeness of the original Game Boy.

Inevitably the Game Boy has become a popular machine for homebrew software and hardware development, and has received wider embracement as a musical instrument. Although Nintendo has produced it's own music creation programmes, for example the *Trippy-H* sequencer included with the Game Boy Camera, the platform's abilities as a music creation tool have been pushed largely due to the efforts of the homebrew development community. The two most prolific of these creative tools are *Little Sound DJ*, also know as *LSDJ*, and *Nanoloop*.

Little Sound DJ and *Nanoloop* are not endorsed by Nintendo, although Nintendo seemingly tolerates their existence since no legal threats have been made against the application developers. In order to use *LSDJ* or *Nanoloop* with an original Game Boy console, the user must first obtain a grey market 'backup' cartridge, and then transfer the downloaded rom image from their computer to the cartridge. These cartridges have the advantage of allowing the user to save their compositions onto their computer for backup purposes, as well as allowing them to playback and interchange music sequences between Game Boy software emulations and the original hardware.

The first cartridge versions of *Nanoloop* were released by German art student Oliver Wittchow in 1999. *Nanoloop* transforms the Game Boy into a music composition and performance instrument, but instead presents the user with a minimal, graphical based interface. This visual interface is based upon a 4 x 4 grid of squares, the user interacts with these shapes using the directional pad and control buttons, tweaking the sound parameters of each of these steps in the loop. Changes made to a square's properties are reflected visually as well as sonically, and the overall interface feel is more exploratory than *LSDJ*. A fine level of control detail is compromised through the graphical interface while improving accessibility.

In 2001 Stockholm based Johan Kotlinski released the first cartridge version of *Little Sound DJ*. As with any rom image, the program can also be used on other portable gaming systems and computers through emulation software, although the creator cautions "Keep in mind that sound emulation can never be 100%, mostly because the low-grade hardware used in Game Boys adds some characteristic noise". [8] Indeed, the original DMG model is the preferred device for live performance and recording of gameboy based tunes.

Little Sound DJ uses a tracker based interface, maximising the level of granular control that the composer has of the Game Boy's music architecture, while fitting comfortably within the visual confines of the platform's 160 x 120 pixel resolution and 4 colour pallet. This level of miniaturisation is made possible through the numeric shorthand of the hexadecimal number system, allowing values above 99 to fit within a two-character space. Navigation is accomplished through the systems four way directional pad and four buttons, although custom peripherals allow further manual control options of the application's sound parameters. The learning curve associated with LSDJ is steep in comparison to its nearest rival Nanoloop, but perseverance in mastering the text heavy interface opens up a high level of granularity and control.

Both *LSDJ* and *Nanoloop* are further augmented through hardware hacks, modifications, and custom peripherals. These mods enhance and expand the Game Boy's hardware interface by adding features such as professional level audio out jacks, backlit screens for improved visibility, and MIDI interfacing to enable the connection of music control devices. The LED hack is an aesthetically as well as functionality motivated mod. It provides useful visual feedback, flashing in synchronisation with the current BPM rate of the device, while enhancing the visual appearance of the Game Boy in when used for playing music live.

In 2003 chiptune music received a boost of mainstream press recognition when Malcolm McClaren, the manager of Sex Pistols, who had been around at the inception of the Punk and Hip Hop subcultures, was interviewed about his involvement with New York's burgeoning chiptune scene. McClaren waxed lyrical about this 'new' subculture in interviews with *Wired*, and *The Guardian*. However his attachment to this scene was soon questioned online through an open letter by GW3M, a member of the Micromusic forum.

"Whilst micromusic.net welcomes interest from the music industry we feel that Malcolm's statements have been at least inaccurate, certainly without acknowledgement of the 25 years of chip music history, and possibly even using ideas and concepts taken from us." [9]

It is indeed important to note that chiptune music didn't begin in the early twenty first century, nor is it confined to the Nintendo platform. Chiptune music has existed as long as the hardware to create it has existed, just existing under different names. The originators include computer game soundtrack composers like Martin Galway and Jonathan Dunne, renowned for their work written for the Commodore 64 SID chip. Music composition on vintage computing platforms has also been pushed on since the early days by game hackers who removed copy protection from commercial video games, then redistributed cracked versions incorporating their own signature theme tunes and 'greetz' to their hacking cohorts in the introduction. The level of technical proficiency needed to compose on a Commodore 64 in the mid-1980s was considerable:

"To have a clearer picture we have to add that those times there were no music composer programs yet. Their tracks were composed as an assembly language program, they mixed sounds "by hand" and coded all filters effects themselves. One can imagine the patience and experience it needed" [10].

The game hacking scene is closely linked to the demo scene. Demos are audiovisual presentations where teams of coders, artists, and musicians work together to produce the most technically and aesthetically impressive presentation possible within the constraints of their host vintage platform. For demos that use modern hardware, the restrictions are imposed, for example the 64-kilobyte memory limit imposed for the Breakpoint 2010 demo contest.

What links the aforementioned three videogame music subcultures, and separates them from the artists using *LSDJ* and *Nanoloop*, are the interface methods employed to create their music. The early computer game musicians, demo coders, and hack intro / hacktro creators all had programming knowledge at their disposal, either individually or as part of a team. The new generation of chiptune music makers benefit from constantly evolving interfaces harnessed to established and deeply explored creative mediums, allowing content creation, while negating the necessity to interface with the platform at developer level.

Interface aesthetics and authenticity

In 2012, Disney released the videogame culture inspired movie *Wreck-It Ralph*. The film centers around Ralph, an antagonist character in fictional classic era arcade game *Fix-It Felix*. Ralph is a non-player controlled character, a feature of the game's dynamic interface. He travels outside of his normal environment across multiple videogame platforms and genres on a quest to find his true self. In an opening scene, the arcade cabinet for *Fix-It Felix Jr.* is shown in an amusement arcade, alongside actual historic cabinets such as *Pong*.

This manufactured association with videogame history is reinforced in the physical space by Disney's creation of a limited number of *Fix-It Felix Jr.* arcade coin-ops. These cabinets provide an experience that simulates the early 1980s arcade videogame, including tactile arcade microswitch controls, an authentic CRT screen with imperfect blurry scanlines, limited audio resolution, and even the character building battle damage of a game presented as a 30 year old museum piece.

Through the manifestation of an interactive artefact, the movie narrative is anchored concretely in our physical universe. The *Fix-It Felix Jr.* cabinet is an emulation of a game that never existed, successfully employing the functionally superfluous, but aesthetically authentic constraints of classic videogame interface aesthetics to enhance the credibility of its claimed vintage. The interface is not the actual game, but it undoubtedly affects how the game is experienced.

References

1. Nick Montfort & Ian Bogost, Racing the Beam: The Atari Video Computer System (MIT Press, 2009), 143.
2. Tom Igoe & Dan O'Sullivan, Physical Computing: Sensing and Controlling the Physical World with Computers, (Premier Press, 2004), 21.
3. Alexander Galloway, The Interface Effect, (Polity Press, 2012), 30.
4. Van Burnham, Supercade: A Visual History of the Videogame Age 1971-1984 (MIT Press, 2003), 28.
5. Ian Bogot, A Television Simulator. Available at: http://www.bogost.com/games/a_television_simulator.shtml.
6. Nathan Altice, 2012. Emulate the Emulators. Meteopal. Available at: http://metopal.com/2012/03/15/emulate-the-emulators/.
7 Cory Arcangel, Super Mario Clouds (Cory Arcangel's Official Portfolio Website and Portal, 2002) Available at: http://coryarcangel.com/things-i-made/supermarioclouds/.
9. gwEm, Open Letter to Malcolm McLaren by gwEm. (Micromusic.net, 2004) Available at: http://micromusic.net/public_letter_gwEm.html.
8. Johan Kotlinski, Little Sound Dj (2015), Available at: http://www.littlesounddj.com/lsd/.
10. Támar Polgár, Freax, (CSW-Verlag, 2005), 90.