http://dx.doi.org/10.14236/ewic/EVA2021.14

Worship the Penguin: Adventures with sprites, chiptunes, and lasers

Jonathan Weinel
School of Computing and Mathematical Sciences
University of Greenwich
London, UK
J.R.Weinel@greenwich.ac.uk

This paper provides a review of recent projects developed through the author's creative practice and activities across multiple computing and games technologies platforms. These include: a 2D game project made in Unity; an Arduino-based laser puzzle; chiptune breakbeat music made on a Commodore 64; the archival of a collection of Amiga demoscene disks; PETSCII graphics; a controller adapter for the Amiga; and a DJ/VJ performance. While playfully exploring new trajectories, these projects broadly reflect on-going themes present in the author's previous work, such as explorations of the aesthetic paradigms presented by vintage computers, 1990s rave culture, and synaesthesia. The paper will address the various challenges and methodologies used to realise these projects; pedagogical considerations; and the pandemic context in which they have been created and presented.

Video games. Unity. Commodore 64. Pixel art. Chiptune. Demoscene. Arduino. Lasers.

1. INTRODUCTION

As discussed in previous papers I have presented at EVA (e.g. Weinel 2019, 2020), and my book *Inner Sound: Altered States of Consciousness in Electronic Music and Audio-Visual Media* (2018), my creative work explores various interwoven themes related to altered states, synaesthesia, visualisations of sound, and the audio-visual aesthetics of 1990s rave culture. Prior to the Covid-19 pandemic I had been working on an Oculus Quest version of *Cyberdream,* a virtual reality (VR) piece which explores the aesthetics of 1990s rave culture and synaesthetic visualisations of sound. During the first lockdown, I consolidated my creative work from the past ten years or so by writing another book, *Explosions in the Mind: Composing Psychedelic Visualisations of Sound* (Weinel, 2022 [forthcoming]).

Following this, I worked on a number of projects, which are relatively small in scale. These works engage with previous concepts, while notably exploring fun and playful dimensions related to 1990s computer aesthetics. They also serve a pedagogical function, since I have been able to use them in my lectures on video games development at the University of Greenwich. In this context I have been using these projects as examples to show various techniques in workshops on programming, sound design and game development. In what follows, I will discuss each piece.

2. *STYLE WARS*

Style Wars (Figure 1) is a 2D video game demo made in Unity. Taking it's name from the classic New York graffiti documentary of the same name (1983), *Style Wars* uses a pixel art style to portray a graffiti artist who must battle through the streets against hoards of hooded characters.

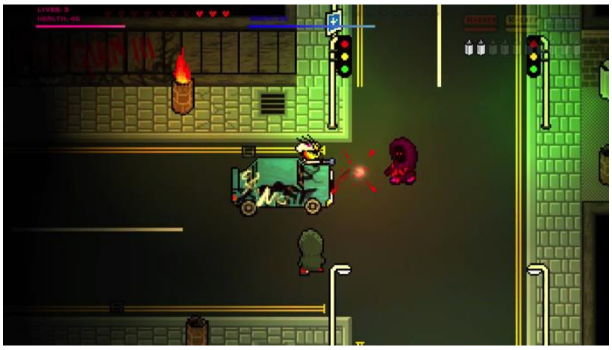

Figure 1: *The Style Wars 2D video game.*

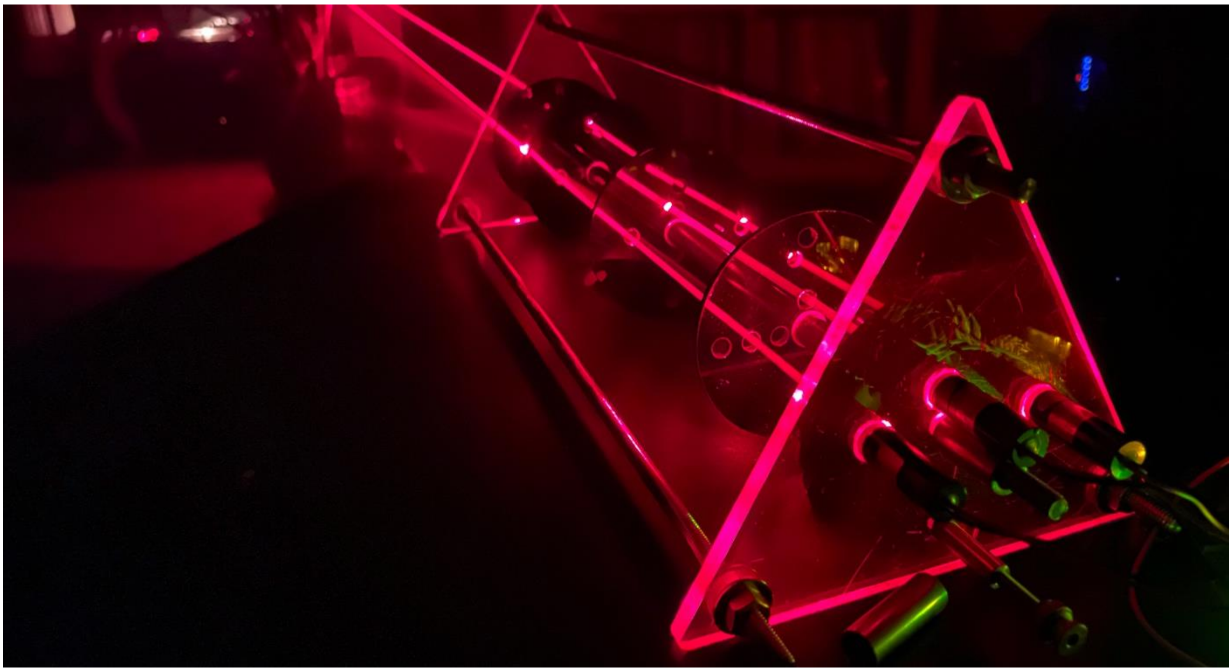

***Figure 2**: The laser puzzle in operation.*

All sprites were made for this game on a Nintendo 3DS using the Inchworm Animation software to quickly design animations using the handheld console's stylus. A top-down environment provides features such as pavement, tarmac, shop windows, damaged vehicles and flaming oil drums. The motif 'Worship the Penguin' appears in spray paint across the walls. The player character is loosely based on the graffiti artist Rammellzee, who created various colourful costumes, and is able to fight with hoards of rival hooded characters by shooting magic or using a sword. The game mechanics involve combat and the collection of keys and other items in order to progress. 2D lighting effects provide a modern take on pixel art style, as used by various other contemporary games that use pixel art.

The soundtrack for *Style Wars* takes inspiration from Yuzo Koshiro's soundtracks for the *Streets of Rage* games (1991–1994). The music was created using Renoise (a modern music tracker) and the Roland Cloud suite of plugins that emulate classic Roland synthesisers, which were popular in the 1980s and 1990s.

Style Wars was made in one week, since this was the only time available to do so, and by design, since intensive creative production in an allocated time-frame such as this can be an interesting way to promote creativity. In terms of its pedagogical use, I have used the piece for demonstrating object-oriented programming concepts.

3. *LASER PUZZLE*

The project shown in Figure 2 is a *Laser Puzzle*. This project was the result of a collaboration with my father (David Weinel, a retired helicopter engineer). David undertook the mechanical construction of the frame based on my sketches, while I devised the electronics and programming. In the context of the pandemic, our collaboration on this project had to be undertaken via a series of telephone calls, emails, and artefacts sent in the post.

A point of inspiration for the laser puzzle was the VR game *Half-Life: Alyx* (Valve 2020), which features multiple spatial mini-games in which the player must manipulate laser beams in various configurations. I wanted to create a physical version of something akin to these puzzles using lasers. Taking *Half-Life: Alyx* as a point of departure, I sketched out a design that would be technically feasible using a relatively simple mechanical construction, an Arduino, three lasers, and three photoresistors.

The resulting artefact is a puzzle, which must be solved by rotating the three central discs mounted on an axle. Each disc contains a series of holes organised into different configurations (Figure 3). Rotating the disc allows one, two, or three of the lasers to pass through the holes in the disc. Rotating all three discs into the correct positions will allow all three lasers to reach the other side of the frame, connecting with three photoresistors (Figure 4). Doing so unlocks the puzzle.

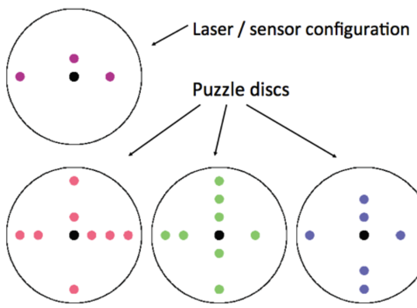

***Figure 3**: The puzzle discs have holes that must be aligned to allow each of the three lasers to pass through and connect with the corresponding sensors.*

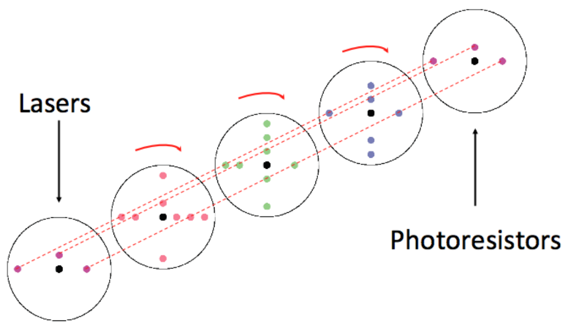

***Figure 4**: Each disc must be rotated in order to find the one correct permutation to unlock the puzzle.*

Holes were placed in various configurations at three distances from the centre of the discs. The mathematical formula for permutations (where the order matters), is n^r, where n is the number of things to choose from, and we choose r of them. This formula allows us to calculate that the discs provide $4^3 = 64$ permutations, of which the orientation of the holes means that only one of these will complete the puzzle. This design was conceived so that the puzzle would not be too easy, though visual feedback, such as seeing the beams pass through the discs in certain positions, should help the user to complete the task.

Various aspects of the laser puzzle can be considered in terms of Norman's (2013) design principles. For instance, Norman discusses the use of sonic feedback in designs, which is implemented in the laser puzzle through the use of a piezo, which emits a clicking sound when each laser connects with its corresponding photoresistor. This click is designed to give 'neutral' sonic feedback; the player knows the laser has connected with the opposite side, but more must be done to complete the puzzle. Red LEDs light up in correspondence with each click. Upon connecting all three lasers, a green LED lights up, and a 'success' sound is emitted, which gives positive emotional feedback to the user through an ascending harmonic sequence of tones. This emotional feedback can be understood as an affective sound design (see Cunningham et al. 2020).

It became apparent that a short 'success' sound only was insufficiently satisfying as a reward for completing the puzzle, and so the project was elaborated through a short laser light show, which is immediately activated when the puzzle is unlocked. I programmed the Arduino to control the piezo in such a way that it emits a series of tones and plays a short tune, which loops four times. Each tone is mapped to one of the lasers, which is triggered in synchronisation, creating a short laser light show in which light and sound are linked.

Challenges encountered during the creation of the laser puzzle included the need for precise alignment between the lasers, holes, and photoresistors, and the need to test and document the device with a smoke machine (in order to make the lasers visible), which tends to sets off the smoke alarms in my flat, which in the pandemic was the only space available to test the *Laser Puzzle*. As noted, the laser puzzle also serves a pedagogical function, a utility that results from the electronic, programming, mechanical, mathematical, and audio-visual mechanisms it exhibits, as well as its use as a vehicle for discussing Norman's (2013) theories of design.

4. *EKTUNE* AT ASSEMBLY

EkTune is a chiptune composition made using a Commodore 64. The piece was originally composed using a Retrogames 'The C64' (a full-size Commodore 64 reissue with functioning keyboard). The Commodore 64 remains a popular choice for composing chiptune music, due to the unique characteristics of the computer's Sound Interface Device (SID). The SID is a programmable sound chip, which provides three audio oscillators, four waveforms (sawtooth, triangle, pulse, and noise), and a filter (Commodore Computer 1982).

***Figure 5**: Work-in-progress screenshot of EkTune during composition with Sid Wizard 1.8 on the Commodore 64.*

The composition is designed based on the typical form of a drum and bass composition. Initially the piece was composed on the Retrogames 'The C64' using Sid Wizard 1.8 (a music tracker, see Figure 5). However, it became apparent that glitch/popping artefacts were being introduced on this platform, due to the system running PAL mode at 60Hz, rather than the usual 50Hz, for reasons of display compatibility. This not only created sporadic glitch/popping problems, but also caused other issues, which became apparent when playing back the track at the correct 50Hz speed on other emulators or actual hardware. Most notably, the tempo was altered, and it transpired that certain sounds were audibly (and somewhat jarringly) different at 50Hz, due to the impact on the Nyquist frequency.

These concerns might not be so important if the composition were to be played back from the Retrogames 'The C64', but since I had decided to submit the composition to a demoparty (an event for computer enthusiasts creating art, music and code on home computers, including vintage systems; see Polgár, 2005), it was necessary that the composition would sound approximately comparable on both the original Commodore 64 hardware and (PC based) emulators.

The solution was to load the composition on to an original Commodore 64, as this provides the most authentic representation of the SID chip. There are two different SID chips in Commodore 64s. Earlier models have the MOS Technology 6581, while later models have the 8580. In this case I used the Commodore 64C, which has the 8580 chip.

Using original hardware was not without its own difficulties, as display compatibility with modern LCD screens is limited, and CRT monitors, which are too big, especially if one's studio is currently the living room! Fortunately, I was able to source a Dell 2001FP monitor, an LCD display, which is compatible with the Commodore 64, as well as the Amiga 1200 which I am also currently using (see Figure 6).

Once the tempo issues and sound design problems of *EkTune* had been resolved on the Commodore 64C, the track was recorded as a .wav file from the original hardware, and converted to a .d64 disk image, which can be played on a Commodore 64 or emulator.

The track was submitted to the Assembly Winter '21 demoparty (Finland), under my Soundcat alias, where it was performed in the streaming music category. The piece is available to listen to online (see Soundcat 2021).

5. ARCHIVING AMIGA DEMOSCENE DISKS

My adventures in vintage computing and demoscene productions have not been limited to my own creative productions. I recently acquired a box of several hundred Amiga demoscene floppy disks from c.1989–1993. The box includes productions from demoparties held in the UK and Europe, including music disks, graphics and slideshows, demo compilations, megademos, multi-part demos and more, by groups such as Scoopex, Kefrens, Cult and many others.

While many of these productions are already archived on Internet sites such as Poeut (https://www.pouet.net/), the box includes some disks with productions which may not be available online. Due to their age, some of the disks no longer work, but I have been working through the box, systematically logging them, and converting them to .adf format, which can then be transferred from the Amiga via compact flash card, and archived on modern computers (Figure 6).

Hopefully this may make a small contribution towards preserving artefacts from the demoscene, which represents an important area in the history of computer art. Indeed, following an initiative by Demoscene – The Art of Coding (http://demoscene-the-art-of-coding.net/), the cultural significance of the demoscene has been recognised by UNESCO, who have awarded the demoscenes in German and Finland 'intangible cultural heritage' status.

***Figure 6**: Archiving demoscene disks on the Amiga 1200, connected to a Dell 2001FP LCD in the living room. The mixing desk connects to KRK Rokit RP5 monitors, mounted on a dinner table with several other monitors and computers.*

Figure 7: *Soundcat64 PETSCII, for Revision 2021.*

6. *SOUNDCAT64* AT REVISION

The Commodore 64 uses PETSCII (PET Standard Code of Information Interchange), a set of 128 characters that includes various letters and symbols (Figure 8). Each character is 8x8 pixels. A global background colour, and individual character colours can be selected using the Commodore 64's pallet of 16 colours.

In the demoscene, PETSCII is used to create artworks. For the Revision 2021 demoparty (Germany), I created the PETSCII graphic shown in Figure 7. This was included in the ASCII/ANSI/PETSCII competition (Revision Demoparty 2021). This graphic extends the interest mentioned earlier in street art and graffiti, using the diagonal PETSCII characters to form lettering. Some Commodore 64 begin with clever transformations of the familiar blue screen, which the computer loads when it is booted. In homage to this, my *Soundcat64* PETSCII shows the lettering exploring through this screen.

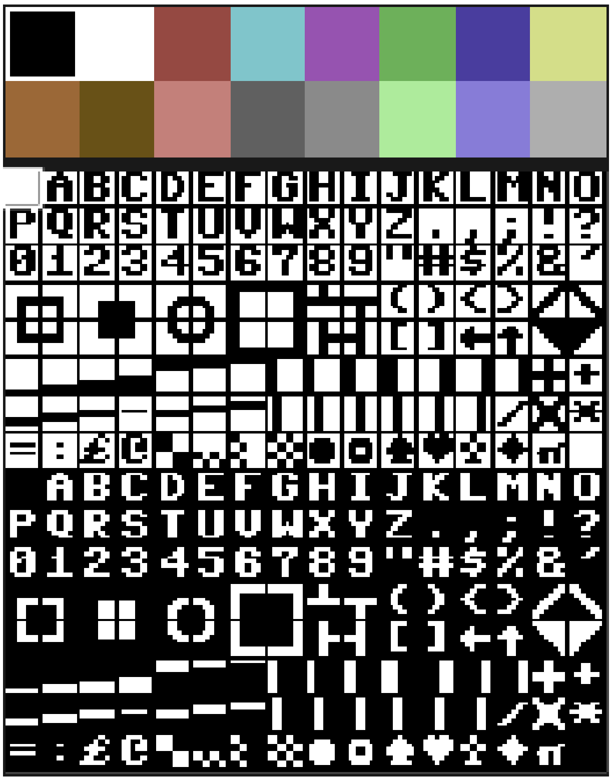

Figure 8*: PETSCII provides a set of 128 characters that can be used with the Commodore 64 colour pallet.*

7. WIRELESS SEGA MEGADRIVE JOYPAD ADAPTER FOR AMIGA

Continuing my exploration of retro video games technologies using modern technologies, for another project I created a controller adapter. Both the Sega Megadrive and the Amiga 1200 use a 9-pin DE9/DB9 connector, where the pin usages have some similarities, but are not identical (see Table 1). Note that the 5v pin is different, and the Megadrive also uses a 'select' pin, which is controlled by the host device in order to read multiple buttons. For instance, setting the select voltage to HIGH enables the B button to be read from pin 6, while setting it to LOW enables the A button to be read from pin 6.

***Table 1**. Pin mappings for the respective controllers.*

Megadrive Controller		**Amiga Controller**	
1	Forward	1	Forward
2	Back	2	Back
3	Left	3	Left
4	Right	4	Right
5	+5v	5	Not connected
6	A / B	6	Button 1
7	Select	7	+5v
8	Ground	8	Ground
9	Start / C	9	Button 2

For convenience, I wanted to use a Retro-Bit SEGA Mega Drive wireless 2.4GHz controller on the Amiga. Initially I found this was possible using a cheap printed circuit board (PCB) available online, which remaps the pins to provide compatibility. This works, however many games benefit from the provision of an autofire, which is often included internally within popular Amiga joysticks such as the Zipstik. I decided to add this to the controller adapter by using an autofire PCB. This worked but requires the player to switch the autofire on and off in situations where autofire is not wanted, such as in games like *Turrican II* where holding down the fire button (rather than repeatedly tapping it) produces a directional beam.

I realised that it would be more useful to utilise the additional buttons of the Megadrive pad, to provide the Amiga button 1 with and without autofire, as shown in Table 2. To do this I created a prototype using an Arduino Uno, utilising a library created by Jon Thysell (2021), which deals with reading buttons from Sega control pad. Figure 9 shows the original prototype. This version provides a potentiometer, which allows the user to control the rate of the autofire, and includes LEDs to show button presses from the controller.

***Table 2**. The remapped button configuration.*

Megadrive Buttons	**Amiga Buttons**	***Turrican II* Functions**
Up	Up	Jump
Down	Down	Crouch
Left	Left	Walk Left
Right	Right	Walk Right
A	Button 1	Directional Beam
B	Button 1 + Autofire	Fire Weapon
C	Button 2	Grenade / Wheel

The prototype version works as intended, so the next step is to create a more compact version housed in a project box. For this I am using a low-cost Arduino Nano compatible board, which will be mounted in a project box for convenience. Since the 8-button megadrive pad has other buttons available, an expanded version of this pad could allow the X, Y, Z and shoulder buttons to be used as other shortcuts for games requiring combinations of buttons, and further modifications can be made using the Arduino code.

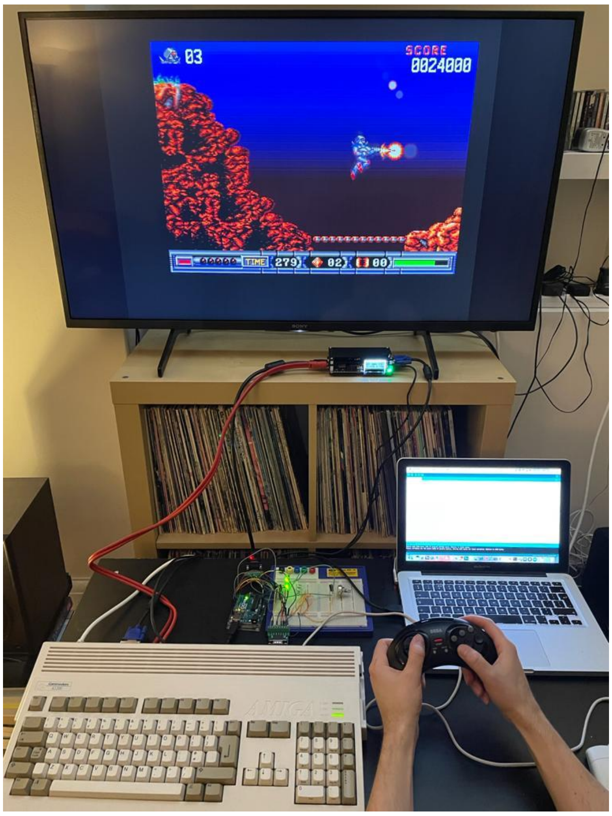

***Figure 9:** The prototype controller adapter on breadboard.*

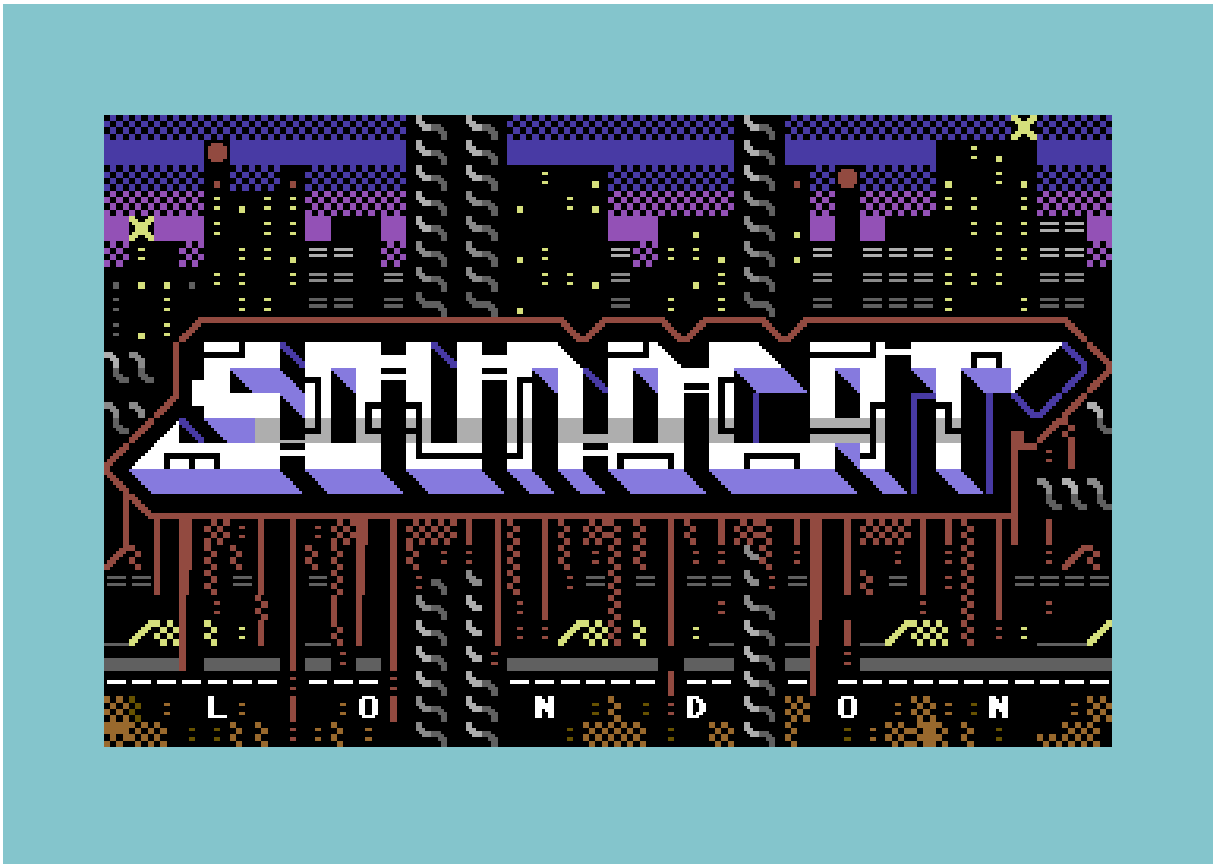

Figure 10: *Soundcat City PETSCII, for Transmission64.*

8. *TKTUNE* AND *SOUNDCAT CITY* AT TRANSMISSION 64

Following on from Assembly Winter '21 and Revision 2021, I produced additional Commodore 64 work for the Transmission64 (2021) demoparty. *TkTune* is a piece of breakbeat music composed for the SID 8580, which incorporates dub sections. I also created another PETSCII entitled *Soundcat City,* which is shown in Figure 10.

9. SOUNDCAT DJ/VJ SET AT ART FUTURA

For the Art Futura London Festival I performed at DJ/VJ set, also under the Soundcat Alias. This extended the DJ/VJ work that I have previously discussed at EVA (Weinel 2019). Earlier versions of this performance utilised bespoke music videos, which were mixed and recorded in the studio using various techniques in analogue and digital video production, such as direct animation on 8mm film and code written in Processing to generate motion graphics. The music consists of a DJ mix of 1990s breakbeat music and beyond.

The Art Futura version of this performance integrates some newer materials not used in the original performance, such as sections with hard trance and acid techno tracks that are paired with new colourful visuals (Figure 11). Due to an earlier SSD drive failure in the Mac computer I normally use for this performance, it was not possible to use Serato DJ Pro with the MixEmergency plugin, as I normally do. Instead some adaptations were made to realise the performance on a Windows 10 laptop with Serato DJ Pro and the Serato Video plugin. Due to the social distancing constraints of the pandemic, this performance needed to be pre-recorded using OBS studio and live streamed via Twitch.

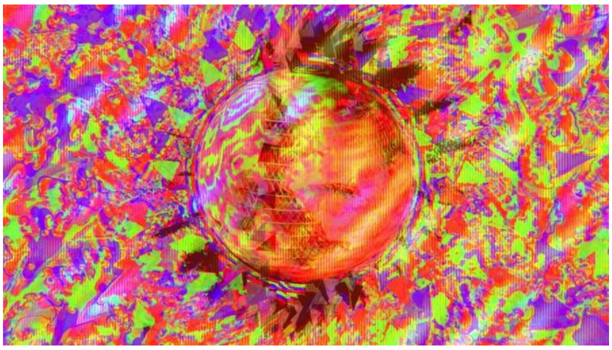

Figure 11: *Still from the Soundcat DJ/VJ performance at Art Futura London Festival, April 2021.*

10. SUMMARY

In this paper I have provided a review of recent creative projects undertaken during a pandemic. The pieces and their dissemination have inevitably been shaped by the circumstances of the lockdowns, and the mental, physical, temporal, and social restrictions these have imposed, but not to the detriment of their design, I think. This context has also required new modes of dissemination. It has been possible to transmit these works via streaming video services, both in remote teaching sessions and at various festivals and demoparties. This is not always entirely satisfactory, since streaming video services do not always provide the framerates or sound quality needed to fully appreciate the works in question. In some cases, the content policies of services like Twitch can also be problematic; for instance, Revision 2021 were banned from Twitch mid-way through the festival, and had to relocate to a different service provided by Chaos Computer Club.

Despite these limitations, online festivals and demoparties do give convenient access to international events where time and cost may otherwise be prohibitive, and where circumstances prevent face-to-face events from taking place. The competition entries for demoparties are also subsequently made available online, so for those with appropriate hardware it is possible to download and experience the original artefact (for my own productions, see Demozoo 2021). At the time of writing I am working on further projects related to those discussed in this paper, and I look forward to presenting my works at the EVA London 2021 conference.

11. REFERENCES

Commodore Computers (1982) *Commodore 64 Programmer's Reference Guide.* Financial Times Prentice Hal.

Cunningham, S., Ridley, H., Weinel, J., and Picking, R. (2020) Supervised Machine Learning for Audio Emotion Recognition: Enhancing Film Sound Design Using Audio Features, *Personal and Ubiquitous Computing*. doi: 10.1007/s00779-020-01389-0

Demozoo (2021) Soundcat demos archive. Online: https://demozoo.org/sceners/121306/

Norman, D. (2013) *The Design of Everyday Things: Revised and Expanded Edition.* Philadelphia: Basic Books.

Polgár, T. (2005) *Freax: A Brief History of the Computer Demoscene*. CSW-Verlag, Winnenden.

Revision Demparty (2021) Revision Online 2021 - Compo - ASCII/ANSI/PETSCII [YouTube video]. Online: https://youtu.be/kdjze7BI9VU (Accessed 18 April 2021).

Silver, T. (1983) *Style Wars* [documentary film]. Public Art Films.

Soundcat (2021) *EkTune* [chiptune music], *Assembly Winter '21: Streaming Music Compo,* 15 March 2021. Version available online: https://youtu.be/4nxJ-QSAjk0

Thysell, J. (2021) Reading Sega Genesis Controllers with Arduino. Online: https://jonthysell.com/2014/07/26/reading-sega-genesis-controllers-with-arduino/ (Accessed 18 April 2021).

Transmission64 (2021) *Transmission64* [online demoparty]. Online: https://www.twitch.tv/videos/999155708 (Accessed 27 April 2021).

Valve (2020) *Half-Life: Alyx* [virtual reality game]. Windows/Steam.

Weinel, J. (2018) *Inner Sound: Altered States of Consciousness in Electronic Music and Audio-Visual Media.* New York: Oxford University Press.

Weinel, J. (2019) Virtual Hallucinations: Projects in VJing, virtual reality and cyberculture. In: Weinel, J., Bowen, J.P., Diprose, G., and Lambert, N. (eds), *EVA London 2019 (Electronic Visualisation and the Arts) 2019*. doi: 10.14236/ewic/EVA2019.57

Weinel, J. (2020) Visualising Rave Music in Virtual Reality: Symbolic and interactive approaches. In: Weinel, J., Bowen, J.P., Diprose, G., and Lambert, N. (eds), *EVA London 2020 (Electronic Visualisation and the Arts) 2020*. doi: 10.14236/ewic/EVA2020.13

Weinel, J. (2022, forthcoming) *Explosions in the Mind: Composing Psychedelic Sounds and Visualisations.* Palgrave Macmillan Studies in Sound.