Sander Jacobsson Skjegstad

The Aesthetics of Demoscene Tracker Music

Bachelor’s thesis in Music Technology
Supervisor: Øyvind Brandtsegg
May 2022

Bachelor’s thesis

NTNU
Norwegian University of Science and Technology
Department of Music

NTNU
Norwegian University of
Science and Technology

Sander Jacobsson Skjegstad

The Aesthetics of Demoscene Tracker Music

Bachelor's thesis in Music Technology
Supervisor: Øyvind Brandtsegg
May 2022

Norwegian University of Science and Technology
Department of Music

NTNU
Norwegian University of
Science and Technology

The Aesthetics of Demoscene Tracker Music

Sander Jacobsson Skjegstad}

May 2022

Bachelor Thesis in Music Technology
NTNU

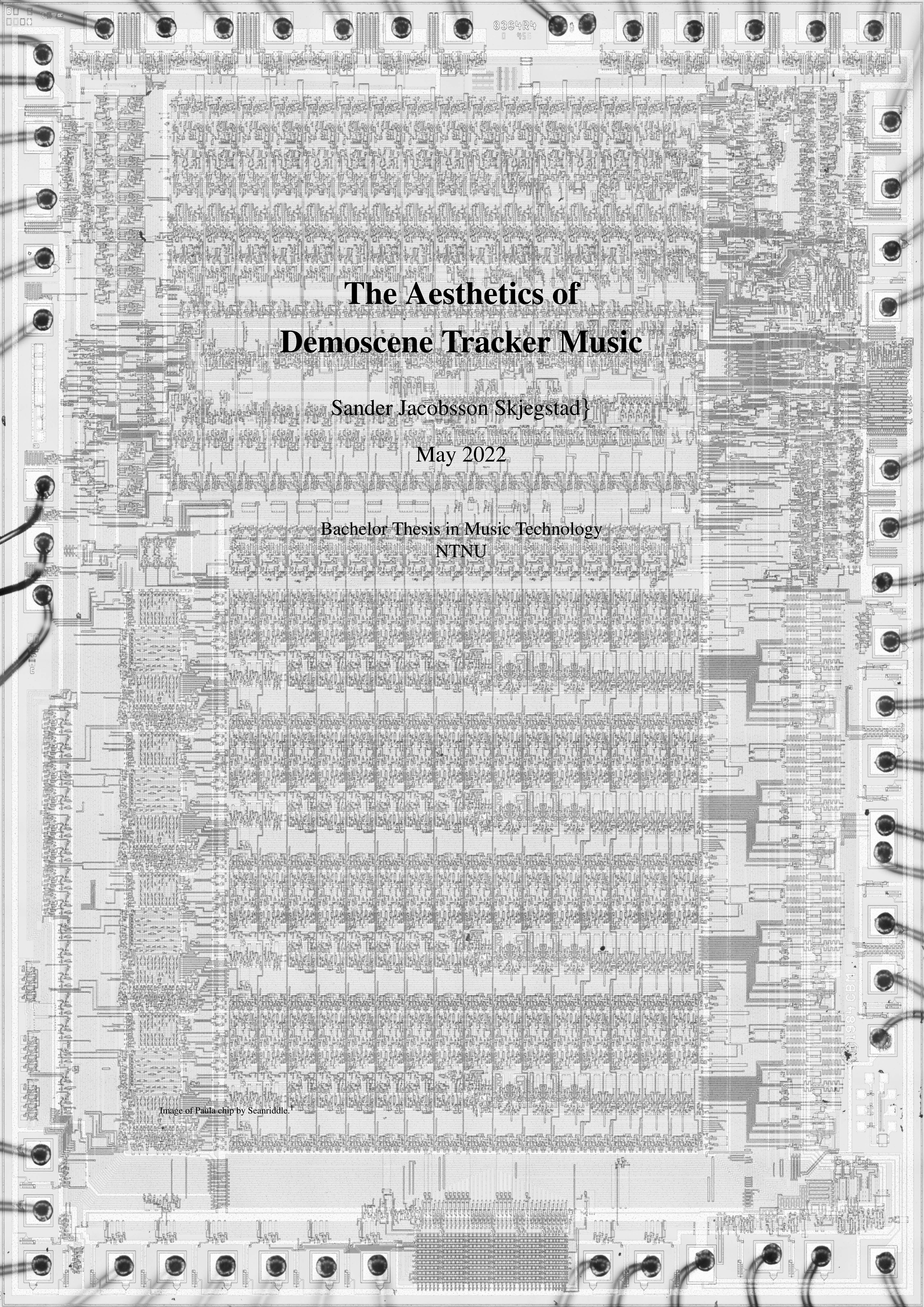


Image of Paula chip by Seanriddle.[1]

1 Abstract

In this project I have worked towards the production of an EP, focused on replicating and understanding the musical aesthetics found within the computer art subculture of the 80s and 90s known as the demoscene. In order to adopt this style and understanding, I have used the same type of software used at the time, a tracker. I have found and utilized the same sample sets available to them at the time. And spoken to still-active members of the scene, to understand the way they worked. The computers at the time had quite limited resources compared to what we have available today. Much of this project went towards learning how to bring as much out as much sonic depth from these limited resources as they did. And understanding how the software and workflow affects the final result.

Contents

2 Introduction

2.1 Background and motivation

The demoscene is an underground computer art subculture that grew out of early home computer software piracy. I want to analyze the style of music found in the scene, and employ this aesthetic to work towards a 5-track EP. Focusing more on common traits, rather than genre, as the musicians in the space all made music in different genres and styles.

My personal motivation to do this project comes from a fascination with the scene. The idea of underground software piracy rings, acting out their group rivalries via creating digital real-time rendered art appeals to me immensely. Because it was largely a northern European phenomenon, with many notable members of the scene being Norwegian, I do feel a certain amount of kinship with demosceners.

2.2 Questions posed

Via this project, I attempt to answer these questions:

- What aesthetic tendencies are common across demoscene/tracker music?
- What did these early computer musicians do to fully capitalize on the potential of the technology?
- How can I replicate these situational variables to aid in producing my own work of the same ilk?
- Is this feasible as a default method of music production in the modern day?

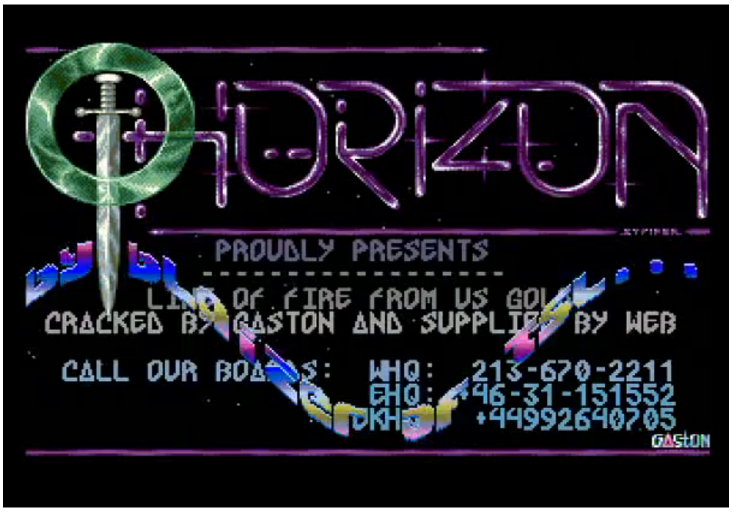
3 Theory

3.1 History of the scene

During the early years of home computing, people started pirating software. Software copy protection was broken by specialized programmers called crackers. Their pirate versions of software are called cracks, or warez (From "ware" as in “software”). These warez were then spread from user to user through various underground channels. BBS, mail, car-boot sales and physical disk swapping.

Due to the highly illegal nature of this, scene members use pseudonyms instead of their real names. Scene members started organizing and collaborating on these projects, which lead to the formation of scene groups. Groups such as Razor 1911, Skidrow, Fairlight and Scoopex.[2]

> By the end of the decade, the practice of cracking the copy protection code and redistributing commercial software had developed into a competitive global culture complete with group rivalries, codes of conduct, and complex communication networks.[3]


HORIZON

PROUDLY PRESENTS

LINE OF FIRE FROM US GOLD

CRACKED BY GASTON AND SUPPLIED BY WEB

CALL OUR BOARDS: WHQ: 213-670-2211

EHQ: +46-31-151552

DKHQ: +44992640705

Figure 1: Intro to Horizon’s crack of the 1989 Amiga game Line of Fire

Scene groups would compete to see who could break the strongest copy-protection, who did it first, and who did it best. On launching their cracked versions of software, they would put an intro. Also known as cracktros, these audiovisual shows served to credit the group members who made the crack. It would also be an opportunity to show off their artistic, musical and programming

[2] Ahoy, *Trackers: The Sound of 16-Bit* (November 19, 2021), 0:49, accessed April 9, 2022, https://youtu.be/roBkg-iPrbw.

[3] Kevin Driscoll and Joshua Diaz, “Endless loop: A brief history of chiptunes,” (Cambridge), https://doi.org/10.3983/twc.2009.096.

skills. Rendered in real time, they would often feature text scrollers containing greetings to, or insults to other scene groups and members.

Groups then started releasing just intros, without an associated crack. These would come to be known as demos. Pure audiovisual demonstrations of musical, artistic, and technical skill. Competition between many of the groups shifted from competing to reverse engineer copy protection best, to making the most impressive demos. This is where the pure demoparty and demogroup comes in. They would organize demoparties, where competitions (or "compos" for short) would be held. Demos would be played on a big screen or projector, and attendees would vote for which entries they liked the best.

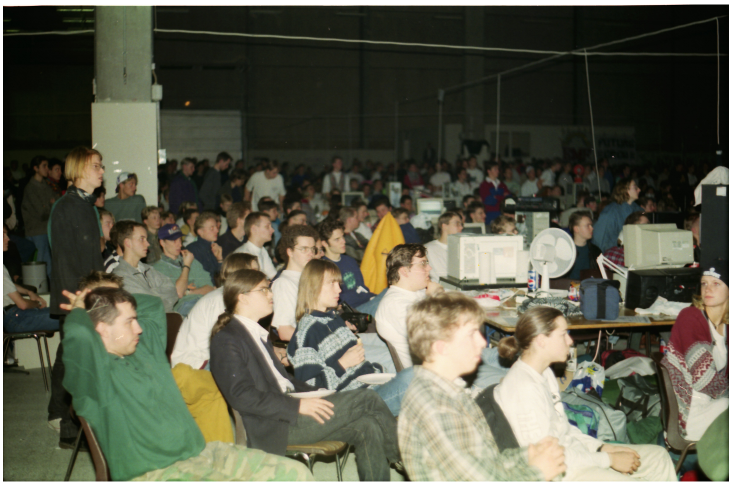


Figure 2: People watching compos at The Party 1994, photo by Thorion

3.1.1 Tracker music and the demoscene

We start in the 16-bit computing era. The main competitors on the market are The Apple Macintosh, Commodore Amiga, Atari ST, and IBM PC/AT. Most of these were fairly lacking in the audio department. The Atari ST had the YM2149 sound chip from Yamaha. While this was far better than the single channel audio of the Macintosh and the 1-bit audio of the PC/AT, it was still of an older generation.[4] Less capable than even the previous generation's Commodore 64 SID-chip.

[4] Johan Kotlinski, "Amiga Music Programs 1986 - 1995," August 20, 2009, accessed May 17, 2022, https://web.archive.org/web/20110708013431/http://goto80.blipp.com/wp-content/themes/goto80/datafoder/kotlinski%20(2009)%20amiga%20music%20programs%2086-95.pdf.

The 1985 Commodore Amiga was particularly significant for musicians because of its Paula chip. A custom chip that could read and play back 4 channels of 8-bit PCM audio at 28kHz sample rate directly from memory, offloading that work from the CPU. Far superior to anything else available at the time. Two of the channels were routed to the left channel, and two were routed to the right channel, allowing for stereo sound.

Early commercial music software for 8 and 16 bit computers was written for people who already knew music. Therefore featuring staves and musical notation as an interface. While this is intuitive to trained musicians, it didn't expose the true capabilities of the sound hardware directly. And to a technically skilled user, that would be more preferable.[5]

Answering that call for his own sake, Karsten Obarski made The Ultimate Soundtracker, to work on a game soundtrack. This was the first ever tracker. It was released as commercial software in 1987, and gave much more direct access to the Paula's capabilities.[6]

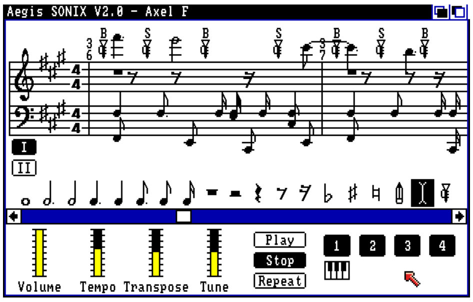
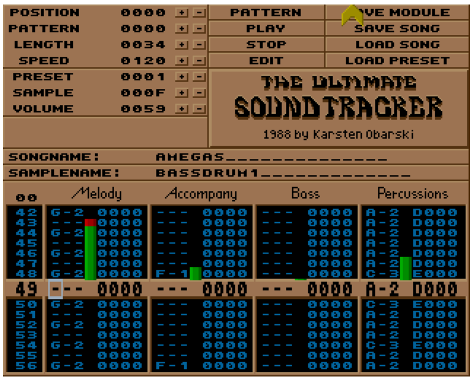


Figure 3: Sonix 2.0 by Aegis



POSITION 0000
PATTERN 0000
LENGTH 0034
SPEED 0120
PRESET 0001
SAMPLE 000F
VOLUME 0059

SONGNAME: AHEGAS_______________
SAMPLENAME: BASSDRUM1_______________

00	Melody	Accompany	Bass	Percussions
42	G-2 0000	--- 0000	--- 0000	A-2 D000
43	--- 0000	--- 0000	--- 0000	A-2 D000
44	G-2 0000	--- 0000	--- 0000	A-2 D000
45	--- 0000	--- 0000	--- 0000	A-2 D000
46	G-2 0000	--- 0000	--- 0000	A-2 D000
47	--- 0000	--- 0000	--- 0000	A-2 D000
48	G-2 0000	F-1 0000	--- 0000	C-3 E000
49	--- 0000	--- 0000	--- 0000	A-2 D000
50	G-2 0000	--- 0000	--- 0000	C-3 E000
51	--- 0000	--- 0000	--- 0000	A-2 D000
52	G-2 0000	--- 0000	--- 0000	A-2 D000
53	--- 0000	--- 0000	--- 0000	A-2 D000
54	G-2 0000	--- 0000	--- 0000	C-3 E000
55	--- 0000	--- 0000	--- 0000	A-2 D000
56	G-2 0000	F-1 0000	--- 0000	A-2 D000

Figure 4: The Ultimate Soundtracker

Although, it had a terrible user experience, members of the scene saw the potential of the system. Among them was The Exterminator from the scene group The Jungle Command. With no care to intellectual property, he reverse engineered The Ultimate Sountracker, and made an unofficial update. Soundtracker II. This would be the first of many unsanctioned upgrades to the software, after he shared the source code. Modified Soundtrackers were now part of the scene, being spread

[5] Ahoy, *Trackers*, 4:20.

[6] Driscoll and Diaz, "Endless loop."

from user to user via the same back channels as the warez.[7]

Significant in this long chain of independent updates was the work done by Unknown from D.O.C. (Doctor Mabuse Orgasm Crackings). They added features such as muting channels, better file handling and the ability to save songs as "modules". This is where the term "modfile" comes from. Now, a modfile is a generic term for any tracker project file. At the time, it only referred to files of the `.mod` file type. A novel filetype that would store all the sample and instrument data along with the note and command data, all in one portable monolithic file. This is now standard for most trackers, but at the time, it opened the door for much easier sharing of songs using novel samples.[8]

3.1.2 The PC era

Trackers continued in the DOS and PC era. Starting with Scream Tracker by Future Crew in 1990, later succeeded by Scream Tracker and Impulse Tracker. Music tracking during the transition to the PC era in many ways experienced a predictably incremental upgrade over the Amiga years. Higher quality sound cards, increased CPU power, more channels, more sounds, more RAM, longer samples.

[7] Ahoy, *Trackers*, 16:00.

[8] Kotlinski, "Amiga Music Programs 1986 - 1995."

3.2 How a tracker works

Trackers are programs for authoring modules, or “modfiles”. They are essentially very complex sample sequencers. A modfile contains the sample and instrument data for the song, aswell as the note and effect data. Almost like how a MIDI file contains the notes and song data, but with sample and instrument data too. This way, it contains everything needed to both work on the song, and play back the song. Unlike a modern DAW which will use a project file which links to separate audio files, and renders the final product to one long audio file. Here, the project file and the finished product are the same thing.

3.2.1 Song structure

In a tracker, you author patterns of note data which are then arranged into the full song. One pattern covers what all the channels play for the duration of the pattern. Almost always 64 lines long by default. Depending on tempo and speed, that usually corresponds 2 or 4 bars of music. Each pattern has a unique ID number, and in the playlist, you sequence them by listing every pattern you want to play in order.

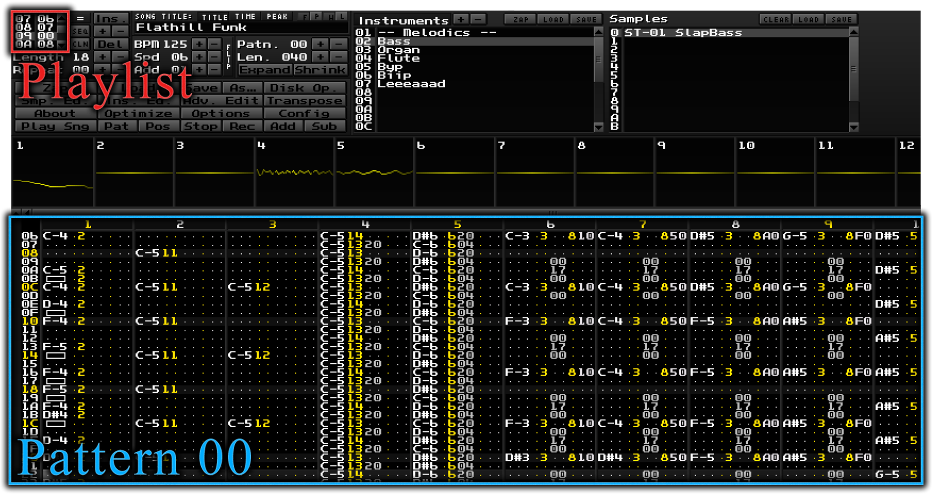


Figure 5: Locations of the pattern view, and playlist in MilkyTracker

When looking at this interface, you will notice that it is written in hexadecimal. Which means instead of counting from 0 to 9, you continue counting from a to `f` after 9. `a` has a value of 10, and `f` has a value of 15. This means that the hexadecimal number `10` is 16 in normal decimal counting. Hexadecimal is a very common sight in trackers. This is because it allows for efficient usage of screen real estate, and neatly divides into 4, the same way music in 4/4 does. The first bar starts on row `00`, row `10` is the start of bar number 2, and row `20` is the start of bar number 3, etc.

3.2.2 Composing

The idea of the tracker distinguishes itself from other types of music software in many ways. Instead of a timeline of musical notation that moves left to right, it uses a spreadsheet-like interface that goes through a list of commands from top to bottom. A single channel is contains a set of command columns, and can only play one sample at any time. In the case of the Amiga, the four channels corresponded to the four channels on the Paula. The first and last channel output to the left speaker, and the middle two channels output to the right speaker.

While every tracker has a slightly different set of command columns, here is a typical example of what one row of one channel may look like:

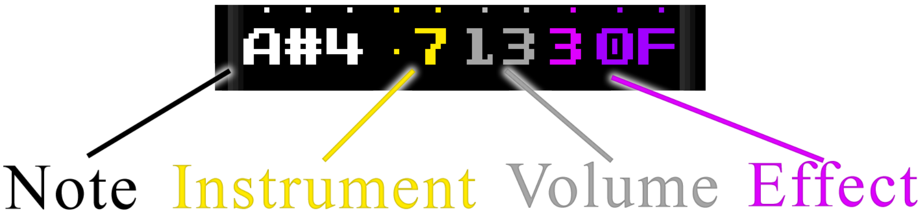


Figure 6: The command columns in MilkyTracker

The first part is note and octave, in this case an A# in octave 4. This note will play until encounters a line containing a stop-note command, or a new note is played.

The second, in yellow, is the index of the instrument that will play the note. In this case, instrument number 7.

The third, in gray, is volume. In this case, the channel volume is set to 13. If it was empty, the channel volume would be set to the default volume of the instrument. Absolute values will set the volume of that channel to the level supplied. Most module formats also have a specification for entering relative changes in volume, such as `u5` to turn the volume up by 5 points over duration of that row.

The fourth, in magenta, is the effect column. First effect selector, and then parameters. What effect corresponds to what number depends on the tracker format you're using. In an `.xm` file, effect `3` is glide-to-note, with the two last numbers controlling the speed of the glide. These effects are not what you may expect. They cannot do things like reverb, phaser or echo. The effects relate more to what the sound hardware is capable of doing with its sample playback, and song settings. Some examples are volume slide, arpeggio, pan slide, pitch slide, vibrato, tremolo, note delay, song tempo, and ticks pr row (aka. speed).

3.2.3 Instruments

All the sounds a tracker produces comes from samples.[9] The way instruments work is different for every tracker format. For `.xm` files, instruments contain samples and information about which keyboard notes they pertain to. They may also contain linear pan and volume envelopes capable of sustain and looping.

Watch the accompanying file milkyWorkflow.mp4 to see what a tracker workflow might look like.

3.3 Notable people, groups and works

Calvin Harris' album I Created Disco was made entirely using OctaMED (A tracker developed independent of the Soundtracker lineage).[10] Video game composer Jesper Kyd started writing music in the demoscene as part of The Silents.[11][12] Some famous users of trackers include The Prodigy, Aphrodite, Bizzy B, DJ Zinc, Aphex

[9] There are exceptions, such as SID-Wizard, trackers that integrate directly with synthesizer-style sound chips, and later trackers that can use software synthesizers

[10] "Interview: Calvin Harris on software, hardware and hit-making," February 20, 2012, accessed May 14, 2022, https://www.musicradar.com/news/tech/interview-calvin-harris-on-software-hardware-and-hit-making-530744.

[11] The Silents, "Global Trash," pouet.net, April 27, 1991, accessed May 2, 2022, https://www.pouet.net/prod.php?which=405.

[12] The Silents, "Jesper Kyd collection," pouet.net, October 1990, accessed May 2, 2022, https://www.pouet.net/prod.php?which=9181.

Twin, Venetian Snares, Darude, Andrew Sega (aka. necros).[13]

The Norwegian demogroup Razor 1911 is, according to the US Department of Justice, the oldest active warez group on the Internet.[14] They formed in 1986. They are still releasing both software cracks containing intros and standalone demos.[15]

3.4 The current state of the demoscene

Many of the largest demoparties transitioned from being demoparties to being LAN-parties. The Gathering, for instance, was originally a demoparty,[16] and still has a small demo-division, but is now mostly about videogames.[17] There are still many pure demoparties happening, however.[18]

4 Method

4.1 Analysis

There are a lot of large online archives of modfiles. Chief among them are The Mod Archive, and the scene.org file archive. Using these resources you can browse and collect files to analyze. Because of the way modfiles work, once you've downloaded one, you can open it in your tracker of choice and it will be like having the project file for the song. The archives are great resources, but I was sent far more modfiles from people's personal collections than I downloaded from archives myself. They would usually come with a recommendation that I look at how the artist achieved some specific effect.

[13] Ahoy, *Trackers*, 25:27.

[14] "Former Leader of Razor 1911, the Oldest Game Software Piracy Ring on the Internet, Sentenced," U.S. Department of Justice, June 6, 2006, accessed May 20, 2022, https://web.archive.org/web/20130415114316/http://www.justice.gov/criminal/cybercrime/press-releases/2003/pitmanSent.htm.

[15] Razor 1911, "Deus Ex: Mankind Devided Cracktro," pouet.net, November 2016, accessed May 12, 2022, https://www.pouet.net/prod.php?which=68539.

[16] "The Gathering 1997," KANDU, accessed May 19, 2022, https://archive.gathering.org/tg97/.

[17] "The Gathering 2018," KANDU, accessed May 19, 2022, https://archive.gathering.org/tg18/.

[18] "Upcoming parties," demoparty.net, accessed May 22, 2022, https://www.demoparty.net/upcoming.

4.2 Collaboration and communication with demosceners

Before the start of this project, I had already made friends with a few tracker musicians and demosceners on the voice chat application Discord. During the project, I frequented a Discord server where most of the demoscene gathers, simply titled “demoscene”, and had various discussions with people there. I also made an account on pouet.net, the largest demoscene-related forum, and started participating there.

One track on the EP, Initialising Keygen, was made in collaboration with an Internet-friend who goes by the pseudonym TB-303. The monolithic nature of the modfiles, and the small file sizes, made sending revisions back and fourth very easy. Because of the standardized file types, we even worked in different software for a while. Me in MilkyTracker, and him in OpenMPT, another open source tracker. Although he did switch to MilkyTracker for his own reasons half way through production.

4.3 Choice of tracker

In choosing a tracker, I needed to consider a few things. I wanted something which was suitably close to what the 80s and 90s demosceners were using, but also something which wouldn’t take me so long to learn that I couldn’t finish the EP. Every one of these programs have their own set of hotkeys and strange workflow quirks that one needs to adjust to, and I was pretty certain I’d only have time to learn one. Here is a list of the ones I most seriously considered using, with reasoning for the final choice.

4.3.1 Renoise

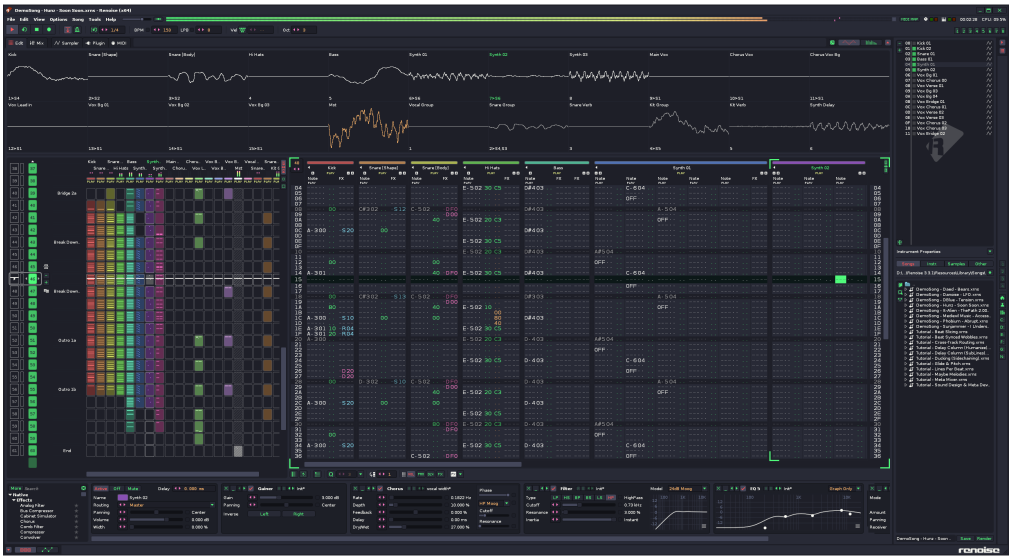
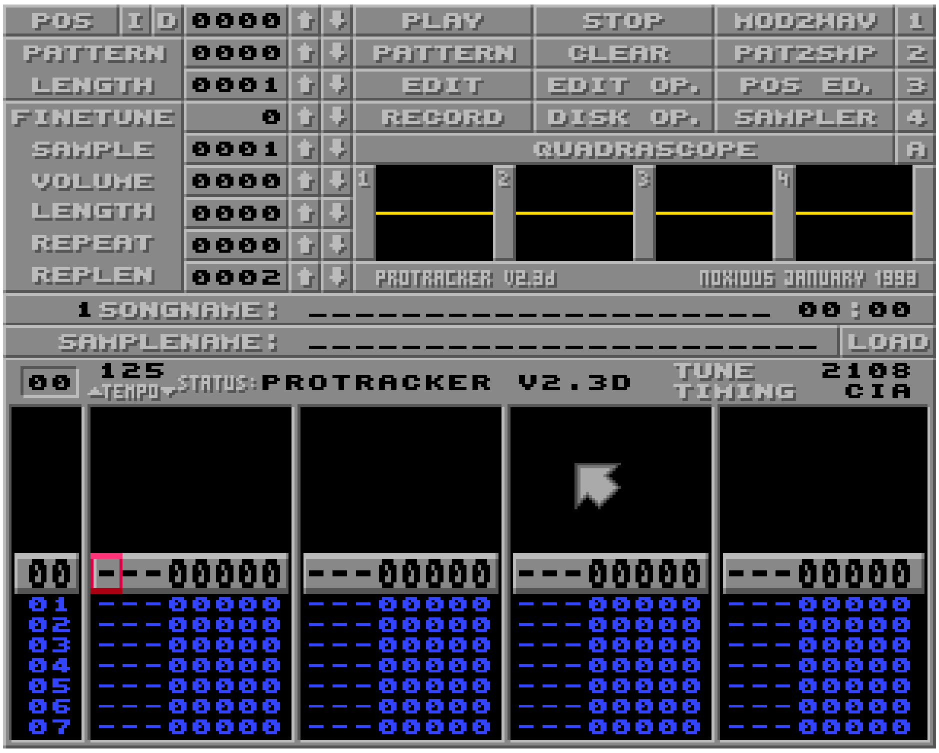


Figure 7: Renoise v3.3.1

Renoise is a powerful modern commercial tracker, that has kept evolving along with modern DAWs. It is fully capable of loading VSTs, doing automation, adding DSP effects like compressors and reverb, routing and sending, you can even do true sidechain compression. It is higly customizable, and comfortable to work in.

Because I was coming from a background of using modern DAWs, this was the first tracker I tried to use. It seemed very approachable and familiar to me as a DAW user. I made one sketch for a track in it. During the production of that sketch, I realized that, while it is a very comfortable environment to work in, the modern conveniences of Renoise make it unsuited for exploring classic scene music. It enables high-quality sinc interpolation and anti aliasing for every instrument by default. There are scripts that go through and disable interpolation for every instrument in a project, but you have to keep running it every time you make a new instrument. Because of the VST functionality, the project files are not truly portable in the same way classic tracker files are. Demoparties won't accept the Renoise file format, `.xrns`, because you need to use Renoise to play them back. I would also have to fight myself to not use the modern conveniences it makes available such as the mixer, DSP effects like delay, and automation.

4.3.2 ProTracker

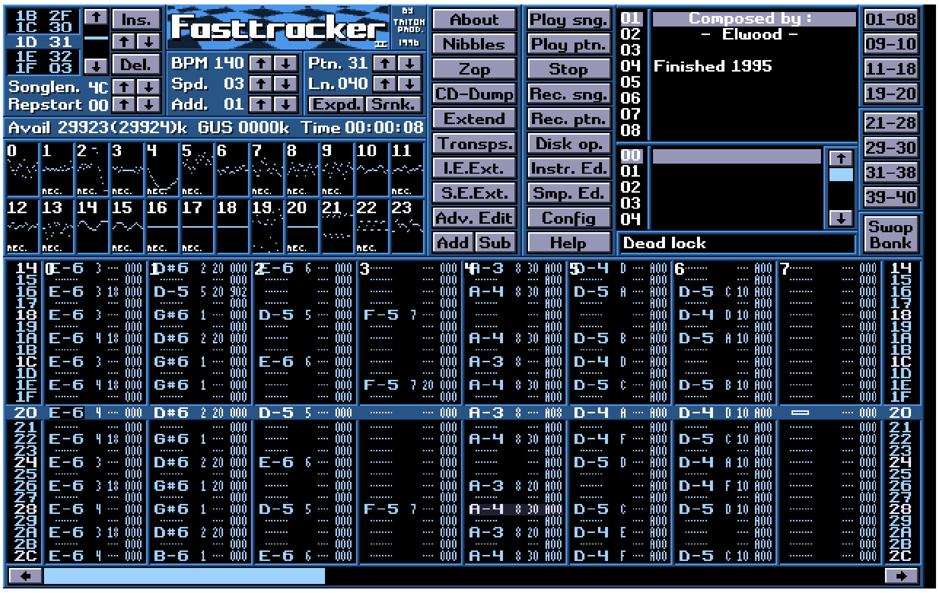


POS 0000
PATTERN 0000
LENGTH 0001
FINETUNE 0
SAMPLE 0001
VOLUME 0000
LENGTH 0000
REPEAT 0000
REPLEN 0002
1 SONGNAME: ____________________ 00:00
SAMPLENAME: ______________________
00 125 TEMPO STATUS: PROTRACKER V2.3D TUNE TIMING 2108 CIA

As part of the original lineage of Soundtracker clones, ProTracker originally ran on the Amiga. It uses `.mod` files. Most modern users of it utilize a clone made by 16-bits.org, which runs on modern systems.

It would be ideal for exploring the Amiga era of tracker music. However, I also wanted to look at 90s tracker music. And this was hard enough to use that it would take a very considerable amount of time to get used to. Too much for me to also learn a tracker that was capable of doing DOS-era music.

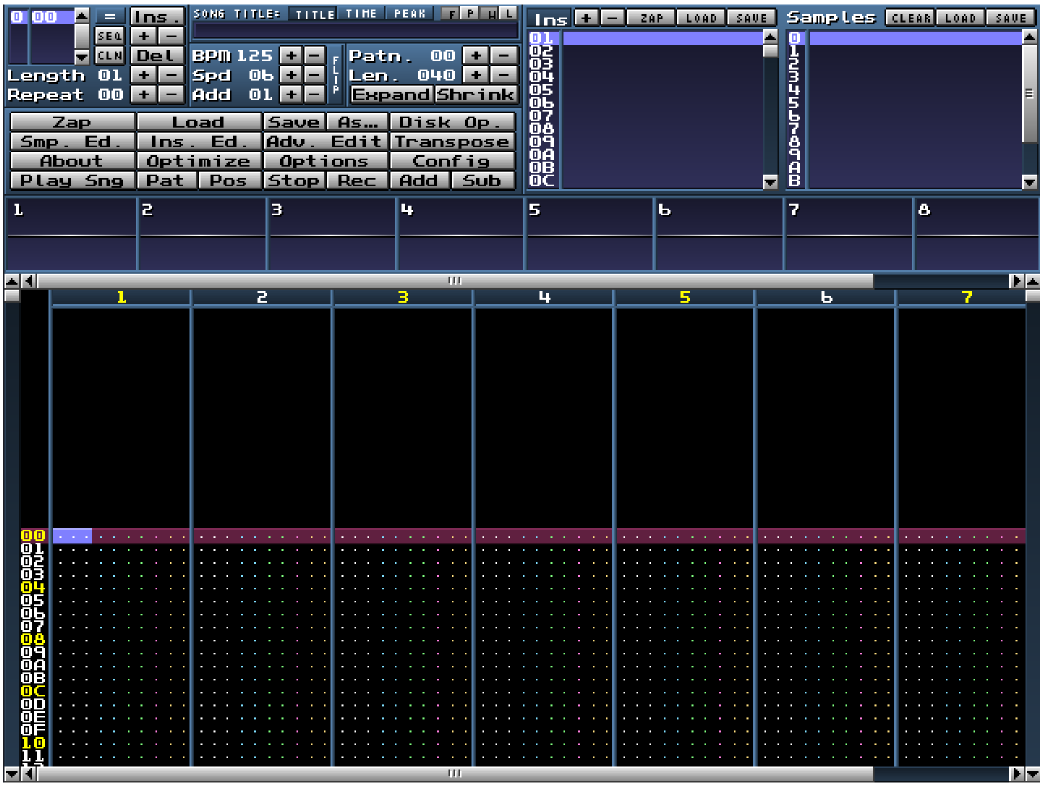
4.3.3 FastTracker II



FastTracker is the originator of the `.xm` module format. Originally running on DOS, this also has a popular clone made by 16-bits.org.

I used that clone for a while, and it is definitely a more comfortable experience than ProTracker. It is able to open, edit and save `.mod` files, although without correct panning. A few modern quality of life features such as the ability to undo is missing, however. And although that is authentic to the original DOS version, there is one very similar tracker that fixes both of those things.

4.3.4 MilkyTracker



MilkyTracker is the tracker I chose in the end. It is an open source and multi-platform tracker that can work perfectly with both `.mod` and `.xm` files. It is created “in the spirit of FastTracker II” and is mostly similar to it.

From the About-page on milkytracker.org:

> It attempts to recreate the module replay and user experience of the popular DOS program FastTracker II [...][19]

And it has an undo buffer, and other modern quality-of-life features. Yet, because it still uses the original formats, it is subject to most of the same limitations as the original software.

[19] “Milkytracker: About,” accessed March 27, 2022, https://milkytracker.org/about/.

5 Findings

5.1 Aesthetic traits, and their origins

5.1.1 Traits of technical origin

The channels of the Amiga's Paula were hard panned. Any song created for playback on the Amiga using the `.mod` format, will have channel 1 and 4 hard panned to the right, and channel 2 and 3 hard panned to the left.

Tiny samples were used to fit next to the cracked software. When fitting the intro for your cracked version of a game or software onto a floppy disk, you often wouldn't have much space for it. Therefore, these intros needed to be quite small. And because the samples in modfiles are stored as raw PCM data, storing long, high quality samples for the intro's music wouldn't work. This is why many intros feature criptune music. Music made with very small, often single-oscillation, samples, looped quickly to create the pitch. This can get a modfile's size down below 15 KB, and fits nicely into the limited space available for an intro.

Rarely any lead vocal melodies. Sampling vocal sections of songs and using that was done, but the limited RAM on these machines meant you couldn't record a full vocal take to go on top of an instrumental. The capacity simply wasn't there. Instead, one would commonly opt for clear lead melodies using instrumental samples or waveform samples.

No Interpolation. Given how low the samplerate many of the samples used, doing decent quality interpolation would put an obvious Nyquist cutoff way down in the perceptible range. And it would shift around as you play back the sample at different pitches. Not interpolating or using sample-and-hold interpolation instead, would fill the frequency range above each sample's Nyquist limit with harmonics and artifacts. This gives a brighter, crisper sound, exemplified by `interpolation-spline.wav` and `interpolation-none.wav`. The Amiga did have a lowpass filter that was used for rough interpolation, but most decent trackers bypassed it, because of how it muffled the sound.[20]

Bitcrushing is not the same as using actually low resolution sound files without interpolation. The sample rate would naturally change with the pitch you're playing

[20] Ahoy, *Trackers*, 19:20.

back the sample at. This would not happen with a modern high resolution sample, interpolated, and then bit crushed. The whole channel / instrument would sound like a single, long, low-res sample.

Quantization. Because trackers play through distinct rows of commands, they lend themselves to quite quantized music. It is possible to achieve a more humanized feel by either using the note delay effect, or selecting an extreme tempo, giving you higher temporal resolution. However, the tracker workflow does not lend itself to that sort of work. Both have been done, but rarely, because of how tedious that workflow is.

Arpeggio used instead of playing full chords. This started in the Commodore 64 years. The C64's SID chip only had 3 channels of audio available, so if you wanted to play a basic triad chord, you would already be out of channels. Therefore, to approximate the sound of a chord, the computer musicians of the era would add features to their personal music suite that allowed them to switch between notes extremely fast. Arpeggios measured in hertz. This today lives on in the form of chord or arpeggio commands in the effect column. In the `.xm` format, it is the first of all the effects.

The sound of the available effect commands. Using the effects made available to you was essential to create an articulated and detailed sound. This means that you will find a lot of glides at the start of notes, and volume commands used to create rhythmic gating.

Manual echoes. Because one channel can only play back a single sample at a time, there is no way to just add a delay effect with feedback. The way tracker musicians got around this to still create an echo effect was by using two versions of the same technique. If you have an arpeggio with an open row between each note, you can put an offset copy the arpeggio in-between those spaces, with a lower volume. This will create an impression of an echo, while only using one channel. Later, in the era of `.xm` files, where channels weren't such a precious resource, you could simply copy the melody to another channel, offset it, and reduce the volume.

The ST-01 sample set. The Ultimate Soundtracker came on two disks. One, named ST-00 contained the program, the other, ST-01, contained a starter sample set. This was a collection of 126 samples sourced from Karsten Obarki's own Roland D50, DX7 and Casio CZ-1. Instruments typical of the 80s and 90s synthesizer-filled music. These samples are ubiquitous in the tracker music world, and have contributed massively to the aesthetic of the music, especially to the Amiga years, having a quite synth-pop character.[21]

The sample sources of records in the 80s and 90s. If you access to a sound card with an input for your Amiga, you could steal samples from your parent's record collection.[22] This would contribute to the collective sample pool, as the modfiles were distributed and so were the samples.

5.1.2 Social origins of traits

I learned from my conversations with demosceners that the competitive environment of 80s and 90s computer fanatics played a big role in shaping their musical output. Scene members who regularly went to demoparties felt driven to make something that that would do better in the next compo. They wanted to win over their rivals and impress their friends. This created an iterative process of feedback and creation, and pushed them to create something which would appeal to the rest of the scene.

6 Result

6.1 Tracklist

The finished and unfinished tracks of the EP are bundled along as `.flac` files, however I highly recommend listening to the original `.xm` and `.mod` files. Much like how people describe vinyl as having a "magic" about it, listening to the computer render the song live, and watching the note commands scroll by is part of the modfile experience. Instructions for doing so can be found in the accompanying `README.txt`.

Below, a list of the included tracks, with a short description of each:

Welcome An ambient intro `.mod`, made using a single sample named "HEAVEN" from the original ST-01 sample set.

[21] Ahoy, *Trackers*, 12:08.

[22] debuglive, *Amiga Samplers: Budget dance music in 1990* (January 31, 2020), accessed April 16, 2022, https://youtu.be/i9MXYZh1jcs.

Flathill Funk A funky house-inspired 16-channel `.xm` module, driven by its bassline, and made using the ST-01 samples. The lead melody received the classic manual echo treatment.

Initialising Keygen A 4-channel `.xm` created in collaboration with TB-303. Meant to mimic the noisier styles cracktro music, this track doesn't have much of an intro. The music in cracktros usually had to cut right to the chase, in case the user clicked their way right through the intro. It has to hook quickly. It is also intended to loop indefinitely, as this music would while the user clicks through the installer, or uses the keygen. And in keeping with the style, it uses tiny samples, and only uses 13.6 KiB of disk space.

Cruising Pangaea An upbeat and highly melodic 10-channel `.xm` module. It makes heavy use of manual ducking to strengthen the kick. By setting the volume to 0 on the same row as an effect command that pulls the volume back up again, it emulates the sound of sidechain compression. This module also contains a lot of chord commands, used for all the chord impacts throughout.

The sketches The sketch named `skippie.xm` uses more liberally sourced samples. Some cheap keyboard and piano sounds grabbed from personal equipment, and some stolen from necros' module "Tangerine Fascination".

`leading.xm` was the result of me testing out some harmonic ideas, and trying to do the same style of musical accents/impacts I had heard in other tracker tunes.

There were many other sketches and attempts left on the cutting room floor, but these two are the ones I will most likely extend into full tracks to complete the EP.

6.2 Participation in Revision 2022

During Easter of 2022, Revision 2022 was held online. The worlds biggest demoparty. Usually held in-person in Germany, they moved online as a consequence of the COVID-19 pandemic. I had tracked a song to submit to one of their competitions, usually referred to as "compos". I submitted Cruising Pangaea, got through pre-selection, and was played first.[23] Once the viewer votes were counted, I placed

[23] Revision Demoparty, *Revision Satellites 2022 - Compo - Tracked Music* (April 22, 2021), accessed April 23, 2022, https://youtu.be/lKGvoL9CkXA.

9th in the tracked music compo.[24] After this experience, I really understand the motivation for doing even better at the next demoparty. I've already ordered tickets to travel and attend physical demoparties in the future.

7 Reflection

7.1 Workflow

I've become much more proficient with trackers over the course of this project. The subjective sense of flow I experience when working in a tracker, with familiar keyboard shortcuts is now as good as it is when I work in a DAW. And better than when I do specifically music-related work in a DAW.

I suspect that the increased sense of flow I experience in a tracker is irrevocably linked with the fact that you can't use effects or VSTs. This is an environment where it is impossible to get distracted doing detailed EQ on your kick, because there is no EQ. This focus and flow has helped me write melodies that I'm more proud of than any other melodies I've composed. I can't say for certain, if this is truly due to the tracker interface, or because I'm taking piano classes and getting better at music.

7.2 Trackers vs modern DAWs

Now to evaluate if trackers are a feasible default method of music production in the modern day. If what you're looking to produce is current modern pop music, then no. That is not how modern pop music is made. With good reason. The workflow is entirely different, and trackers are very uncomfortable environments for recording performances. However, if you are doing mostly instrumental electronic music, then it does have its advantages.

They tend to have a drastically reduced need for window management compared to something like Pro Tools or FL Studio. Everything can be controlled through the keyboard, letting you work very vast and reach a state of flow easier. There is

[24] "results.txt," The International Scene Organization, https://files.scene.org/view/parties/2022/revision22/results.txt.

research supporting this, and that the edit-audition cycle we utilize when composing on a computer is faster in a tracker.[25] The system requirements for doing tracking are very low, letting you easily track on a low-power laptop while traveling. The files are perfectly portable, allowing for low-friction collaboration, even with people who are not using the same software as you. Although, that portability requires using an older format such as `.xm` or `.mod`, which will come with some pretty severe limitations compared to modern DAWs. You lose the ability to do DSP, add VSTs and effects, software synthesizers, efficiently mix a song, etc.

Some of these can be worked around by doing a hybrid workflow, of composing in a tracker, and mixing in a DAW. Or, if you're willing to give up the interoperability between programs, use a modern tracker like Renoise, which has the best of both worlds.

7.3 Reflection on the result

For me, making an EP in half a year is a somewhat ambitious. Making an EP using unfamiliar software and a whole new workflow is another level of ambitious. Half way through, I started to shift focus from creating finished tracks, to doing tests and experiments around the aesthetic traits I wanted to exemplify for the assignment. In the end, I have four complete tracks, and a few loops and experiments.

This means I fell just short of my goal of a complete 5-track EP. I bit over more than I could chew. While that is a disappointment, the music itself, I feel represents some of my better work. The techniques and tricks I found in other tracker musicians' styles and songs have been integrated with my own, and the music reflects this. This short tracklist certainly can't be seen as representative of the style of the diverse field of tracker and demoscene musicians, and it doesn't use every trick in the book, but it doesn't sound out of place in a compo. And in that way, the project was a success.

[25] Chris Nash and Alan Blackwell, "Tracking Virtuosity and Flow in Computer Music," in *Proceedings of the 2011 International Computer Music Conference* (Huddersfield, UK: Michigan Publishing, 2011), http://hdl.handle.net/2027/spo.bbp2372.2011.116.

References

Ahoy. *Trackers: The Sound of 16-Bit.* November 19, 2021. Accessed April 9, 2022. https://youtu.be/roBkg-iPrbw.

debuglive. *Amiga Samplers: Budget dance music in 1990.* January 31, 2020. Accessed April 16, 2022. https://youtu.be/i9MXYZh1jcs.

Demoparty, Revision. *Revision Satellites 2022 - Compo - Tracked Music.* April 22, 2021. Accessed April 23, 2022. https://youtu.be/lKGvoL9CkXA.

Driscoll, Kevin, and Joshua Diaz. "Endless loop: A brief history of chiptunes." (Cambridge). https://doi.org/10.3983/twc.2009.096.

"Former Leader of Razor 1911, the Oldest Game Software Piracy Ring on the Internet, Sentenced." U.S. Department of Justice, June 6, 2006. Accessed May 20, 2022. https://web.archive.org/web/20130415114316/http://www.justice.gov/criminal/cybercrime/press-releases/2003/pitmanSent.htm.

"Interview: Calvin Harris on software, hardware and hit-making," February 20, 2012. Accessed May 14, 2022. https://www.musicradar.com/news/tech/interview-calvin-harris-on-software-hardware-and-hit-making-530744.

Kotlinski, Johan. "Amiga Music Programs 1986 - 1995," August 20, 2009. Accessed May 17, 2022. https://web.archive.org/web/20110708013431/http://goto80.blipp.com/wp-content/themes/goto80/datafoder/kotlinski%20(2009)%20amiga%20music%20programs%2086-95.pdf.

"Milkytracker: About." Accessed March 27, 2022. https://milkytracker.org/about/.

Nash, Chris, and Alan Blackwell. "Tracking Virtuosity and Flow in Computer Music." In *Proceedings of the 2011 International Computer Music Conference.* Huddersfield, UK: Michigan Publishing, 2011. http://hdl.handle.net/2027/spo.bbp2372.2011.116.

Razor 1911. "Deus Ex: Mankind Devided Cracktro." pouet.net, November 2016. Accessed May 12, 2022. https://www.pouet.net/prod.php?which=68539.

"results.txt." The International Scene Organization. https://files.scene.org/view/parties/2022/revision22/results.txt.

Seanriddle. *CBM_8364R4_top_metal.jpg*. February 13, 2017. Accessed May 15, 2022. https://commons.wikimedia.org/wiki/File:CBM_8364R4_top_metal.jpg.

"The Gathering 1997." KANDU. Accessed May 19, 2022. https://archive.gathering.org/tg97/.

"The Gathering 2018." KANDU. Accessed May 19, 2022. https://archive.gathering.org/tg18/.

The Silents. "Global Trash." pouet.net, April 27, 1991. Accessed May 2, 2022. https://www.pouet.net/prod.php?which=405.

———. "Jesper Kyd collection." pouet.net, October 1990. Accessed May 2, 2022. https://www.pouet.net/prod.php?which=9181.

"Upcoming parties." demoparty.net. Accessed May 22, 2022. https://www.demoparty.net/upcoming.

Proudly written in a terminal and compiled with LaTeX.
Word-count through `detex`: 5045

NTNU

Norwegian University of
Science and Technology