THE OXFORD HANDBOOK OF
INTERACTIVE AUDIO

Edited by
KAREN COLLINS, BILL KAPRALOS,
and
HOLLY TESSLER
CONTENTS

List of Common Acronyms Found in the Handbook  ix
List of Software Found in the Handbook  xi
List of Games Found in the Handbook  xiii
List of Contributors  xvii
About the Companion Website  xxvii

Introduction  1
KAREN COLLINS, HOLLY TESSLER, AND BILL KAPRALOS

SECTION 1 INTERACTIVE SOUND IN PRACTICE

1. Spatial Reconfiguration in Interactive Video Art  15
   HOLLY ROGERS

2. Navigating Sound: Locative and Translocational Approaches to
   Interactive Audio  31
   NYE PARRY

3. Defining Sound Toys: Play as Composition  45
   ANDREW DOLPHIN

4. Thinking More Dynamically about Using Sound to Enhance Learning
   from Instructional Technologies  62
   M. J. BISHOP

5. Acoustic Scenography and Interactive Audio: Sound Design for Built
   Environments  81
   JAN PAUL HERZER

SECTION 2 VIDEOGAMES AND VIRTUAL WORLDS

6. The Unanswered Question of Musical Meaning: A Cross-domain
   Approach  95
   TOM LANGHORST
How Can Interactive Music Be Used in Virtual Worlds like World of Warcraft?
Jon Inge Lomeland

Sound and the Videoludic Experience
Guillaume Roux-Girard

Designing a Game for Music: Integrated Design Approaches for Ludic Music and Interactivity
Richard Stevens and Dave Raybould

Worlds of Music: Strategies for Creating Music-based Experiences in Videogames
Melanie Fritsch

SECTION 3 THE PSYCHOLOGY AND EMOTIONAL IMPACT OF INTERACTIVE AUDIO

Embodied Virtual Acoustic Ecologies of Computer Games
Mark Grimshaw and Tom Garner

A Cognitive Approach to the Emotional Function of Game Sound
Inger Ekman

The Sound of Being There: Presence and Interactive Audio in Immersive Virtual Reality
Rolf Nordahl and Niels C. Nilsson

Sonic Interactions in Multimodal Environments: An Overview
Stefania Serafin

Musical Interaction for Health Improvement
Anders-Petter Andersson and Birgitta Cappelen

Engagement, Immersion and Presence: The role of Audio Interactivity in Location-aware Sound Design
Natasa Paterson and Fionnuala Conway

SECTION 4 PERFORMANCE AND INTERACTIVE INSTRUMENTS

Multisensory Musicality in Dance Central
Kiri Miller

Interactivity and Liveness in Electroacoustic Concert Music
Mike Frengel

Skill in Interactive Digital Music Systems
Michael Gurevich

Gesture in the Design of Interactive Sound Models
Marc Ainger and Benjamin Schroeder

Virtual Musicians and Machine Learning
Nick Collins

Musical Behavior and Emergence in Technoetic and Media Arts
Norbert Heuber

SECTION 5 TOOLS AND TECHNIQUES

Flow of Creative Interaction with Digital Music Notations
Chris Nash and Alan F. Blackwell

Blurring Boundaries: Trends and Implications in Audio Production Software Developments
David Bessell

Delivering Interactive Experiences through the Emotional Adaptation of Automatically Composed Music
Maia Hoeberechts, Jeff Shantz, and Michael Katchabaw

A Review of Interactive Sound in Computer Games: Can Sound Affect the Motoric Behavior of a Player?
Niels Böttcher and Stefania Serafin

Interactive Spectral Processing of Musical Audio
Victor Lazzarini
SECTION 6 THE PRACTITIONER’S POINT OF VIEW

28. Let’s Mix it Up: Interviews Exploring the Practical and Technical Challenges of Interactive Mixing in Games
   HELEN MITCHELL
   479

29. Our Interactive Audio Future
   DAMIAN KASTRAUER
   498

30. For the Love of Chiptune
   LEONARD J. PAUL
   507

31. Procedural Audio Theory and Practice
   ANDY FARNELL
   531

32. Live Electronic Preparation: Interactive Timbral Practice
   RAFAL ZAPALA
   541

33. New Tools for Interactive Audio, and What Good they Do
   TIM VAN GELEEN
   557

Index
   571

List of Common Acronyms Found in the Handbook

AI: Artificial Intelligence, referring to machine learning ability.
API: Application Programming Interface, a specification designed to interface between software.
DAW: Digital Audio Workstation, a home computer recording studio.
DLC: DownLoadable Content, content that is commonly an add-on to games or other software that can be downloaded by the user.
DSP: Digital Signal Processing, in referring to sound, is the various effects used to enhance or change a sound wave.
FM: Frequency Modulation: in the context found here, FM is used in regard to an early form of sound synthesis (as opposed to a radio broadcast mechanism).
FPS: First-Person Shooter, a genre of game in which the player is in first-person perspective, commonly holding a gun.
GUI: Graphical User Interface, an image- or icon-based interface.
HCI: Human–Computer Interaction, a branch of computer science that focuses on the interaction between humans and computers (hardware and software).
HRTF: Head-Related Transfer Function describes the location- and distance-dependent filtering of a sound by the listener’s head, shoulders, upper torso, and most notably, the pinnae of each ear.
IASIG: Interactive Audio Special Interest Group, an industry-led organization that creates specifications and standards, and research reports on audio-related topics. http://www.iasig.org
MIDI: Musical Instrument Digital Interface, a music industry specification for interfacing between instruments and software.
MIR: Music Information Retrieval, a branch of computer science that focuses on our ability to search and retrieve music files.
MMO/MMORPG: Massively Multiplayer Online Game/Online Role-Playing Game, online games in which there are multiple simultaneous players over a network.
NIME: New Interfaces for Musical Expression, an annual conference that brings together work on new and emerging musical interfaces and instruments.
"CHIPTUNE" gets its name from the sound chips in computers that were originally used to balance the processing power of generating sound effects and musical tunes from the CPU. Each computer or videogame machine would have its own sound chip to create audio and it gave a unique sound to each system. The sound of chiptune is the sound of nostalgia filtered through the mono speaker of a blurry television from a videogame console made over thirty years ago. However, we are now seeing a generational shift from those who grew up with videogames and computers from the 1980s to those born into the age of the internet who are listening to chiptune music. There is a certain "retro chic" at play, where the first generation has a nostalgia for the original systems, and the new generation is attracted to the old sound as a way to generate something new, similar to the color photography film digital effects that remind us of the film development from years ago, or the recent trend for Polaroid-esque photos. There is an overlap of the old and the new generations that has created a fascinating blend of styles and approaches to the relatively new label applied to this type of music that has become popularly known as chiptune music.

The synthetic nature of chiptunes allows them to be easily manipulated in real time and most composition systems allow for a degree of live control. Some of the first playback systems for chiptunes (such as the Commodore SID chip) even allowed for live instruments to be modified by the chip’s onboard processing. The common format of chiptune creation, being module trackers or "trackers," still finds a home today within modern game-building tools such as Fmod and Unity3D supporting this flexible chiptune format which is explained further in the following sections. A single definition of chiptune is debatable, but one can split the definition of chiptunes into three separate definitions from most restrictive to the most inclusive, relating to hardware, nostalgia, and the chiptune aesthetic.
(Commodore Business Machines 1982) and the Nintendo Entertainment System (Ricoh 2A03 sound chip (Taylor 2004). Modifications such as SD card-to-cartridge adapters are allowed to be added to the system, but only at the hardware level and should not pervert the original commonly viewed limitations of the system. The notion of this definition is to preserve the “pure sound” of the chiptune coming directly from the original system. Since the composition software often has an effect on the composition and the sonic capabilities of the song, there can even be a distinction between using an old-school tracker directly on the system, such as SID DUZZ’IT for the Commodore 64 and a new-school tracker such as GoatTracker, which runs on a modern PC but also allows one to export the file such that it can be played on an actual Commodore 64 (see Figure 30.1). Not only does using purely keyboard input of SID DUZZ’IT change one’s compositional style, but it also allows for the integration of software samples (“digis”), which GoatTracker does not currently support. Another clear example is Nanoloop for the Gameboy Advance (GBA) that allows FM synthesis, which is too computationally restrictive to use in an actual GBA title but which expands the sonic palette of the GBA via software. In its most refined sense then, an expert chiptune artist can tell the difference not only between chiptunes from different systems but also from different composition programs used to produce the songs.

Chiptunes made for videogames on the original systems are often referred to as VGM (for “videogame music”). The dividing line between the “classic videogame” synthesis sound can be seen as lying between the simple waveform channels of the Nintendo Entertainment System (NES) in 1985 and the sampled sound generation of the SNES in 1990. Accordingly, “chiptunes” can be seen as a modern term for songs composed with these technologies, although not necessarily made for exclusively for videogames.

### 30.2 Nostalgia

An informed listener will be able to determine which particular sound chip is being emulated, simply by knowing the original specifications of a chip. For this definition, the core constraints of what made the sound of a particular sound chip unique in comparison to other chips is integral in the form and structure of the composition. The overall idea is to capture the “spirit” or “soil” of the chip for the correct feel of the chip but not necessarily all the grit of being constrained by limitations or errors in the original chip design and physical hardware that made compositions on the original systems problematic. A clear example of such a system bug is the Nintendo vibro bug, where severe clicking and distortion can occur due to a design flaw in the original 2043 chip (see the Famitracker WIKI: http://fami-tracker.com/wiki/index.php?title=Sound_hardware). This flaw forces composers to be sensitive to certain vibrato ranges that should not be used. The drawback of this nostalgia-based definition is how to define a subjective notion of the “core constraints” of a particular sound chip. This nostalgia-based definition covers the work that I completed for the Retro City Rampage Soundtrack, in which all of the composers used the open-source PC tracker OpenMPT with a set of samples to emulate the sound of the original NES. The songs would need to be slightly rearranged to play on an actual NES, but with the use of the conversion utility IT2NSF or by hand rearrangement in Famitracker or another NES-native program, the results could be very similar to their originals.

### 30.3 The Chiptune Aesthetic

The most relaxed definition of chiptune is a subjective one, where the aesthetics of classic sound chips are what establishes a song as being chiptune. This definition usually includes reference to the basic waveforms such as pulse, triangle, sawtooth, and pseudo-random noise with compositional artifacts such as rapid arpeggiation, portamento, and limited polyphony give rise to that “chiptune sound.” This broader definition allows for the addition of reverberation, DSP compression effects, delays, and other techniques not originally available on a particular system, to be used to enhance the musicality of the basic waveforms. An example of such a broad definition is composers that use VST (“Virtual Studio Technology”) plug-ins such as Plogue
Chipsounds as an element in their compositions. This plug-in has the advantage of using sampled drums and other elements to produce a chiptune. With this definition, it can be difficult to define a chiptune song with its closely related electronic genres such as electro, house, or other styles that the song may also reference. In this case, the chiptune sound is likely just another element in a layered mix within a song.

30.4 Genre or Instrument?

With the definitions above it can be seen that the chiptune genre is both a compositional form and an instrument for composition. Chiptune purists tend to enjoy the arcane nature and high level of difficulty of producing chip tunes on original or personally customized hardware, whereas others see the expansion of the chiptune aesthetic into other forms of music as a natural step in an artistic canon. I would argue that the core of chiptune lies in the instrumentation, since basically any style of music can be composed using chiptune methods. If you play a piece of music on a piccolo it doesn’t make it "picocolute!"

Making chiptune music can be seen as an orchestration and arrangement exercise, where each channel is its own instrument, and compositions will often need to be rearranged to suit the limited polyphony and waveforms of the target chiptune system. Part of my own attraction to chiptune is the minimalism required to arrange a song for a very limited number of channels. This minimalism can also help the mix of the song. For example, on the Commodore 64, the bass guitar and kick drum are often on the same channel so there is no mix difficulty of their sonic content conflicting with each other as they play at different times. Similar to other chiptune artists, I am forced to be very spartan and focused in my chiptune compositions by limited voices, which boils the song down to its essential elements (Yabsley 2007).

Within the chiptune genre, there are definite stylistic differences between the old-school sound of songs produced in the 1980s and the new-school sounds produced in recent years. Similar to other musical genres and art movements, chiptune music closely refers to the music styles of the times but it is also self-referential as well. "Old-school" European chiptune songs have their roots in the electronic music of pioneers such as Kraftwerk, but they are also self-referential, due to the demoscene producing a distinct composition style that often referred to clear melodic lead lines and repeated musical structures relating to later synthpop music. This contrasts somewhat with the Japanese style exemplified in the Konami-style, which is related to fusion and prog-rock genres with blazing virtuosic lead lines, dense rhythmic patterns, and often alternating odd-meter time signatures throughout the composition. The cross-pollination of genres continues to this day, with artists drawing from existing genres and creating new sounds for chiptunes that refer to dub-step, moombahton, and other emerging music genres.

30.5 Advances in Chiptune Hardware and Software

The creation of chiptune music is a combination of various elements that include the composer, the composition software, and the hardware. The hardware can be broken down further into the sound chip itself and the central chip used to control the sound chip. In the case of the original Nintendo Gameboy, there is no dedicated sound chip, so the sound chip is considered to be the CPU of the system. With the second generation of chip music, the sound chip can be removed entirely from the original system and put under a new system of control. For instance, the HardSID4U (created by Tel’i Sándor in 2010) is one example of a system where four of the original Commodore 64 SID chips are removed from the control of the original Commodore 64 CPU, and housed in a unit that can respond to a modern PC that sends it messages via a higher-speed USB connection. This addition allows for modulation of the chips at a rate that exceeds the capabilities of the original 1 MHz 502 and also makes it easier to allow for audio signals input into the SID chip to be processed as well. The primary advantage of the HardSID4U is that it removes the sound chip from electrical interference by other electronic components, and this increases the audio fidelity of the signal from the chip itself via the pro-audio sound components of the HardSID4U. This is similar to "pro-sound mods" that are available for the DMG, NES, and other systems that increase the audio fidelity of the system by modifying the audio signal path of the hardware to reduce interference.

A significant increase in the quality of chiptunes has been from the improved software interface between the composer and the sound chip. In the early days of the Commodore 64, game composers such as Rob Hubbard would be required to learn 6502 assembly language programming in order to make music for the SID chip (McSweeney 1993). These days, there are tracker programs, such as GoatTracker, that have become refined to better suit composers. Over time, this development has allowed for increased efficiency in the use of the mouse, cut and paste functions, and a more user-friendly workflow, as usability is refined over time and incorporates feedback from a variety of users. The response time of a composition system is very important, as it can produce a feedback loop nearing real time such as with traditional instruments. This system is in stark contrast to other systems that required the data to be meticulously hand-coded by hand, compiled and reviewed to allow for updates, such as using MML (music macro language) and other macro or scripting languages to make chiptunes. Although this older type of system produces a different type of music, often even the original composers state that they wouldn’t have the patience now to make music in the "old way" (Greening 2010). Sometimes this intensity and amount of focused time required to create chiptunes using the original software and system produced a music that was more minimalist, simply because it required a significant amount of extra time to produce music with a large degree of detail. This minimalism also makes more nuanced chiptunes stand out, such as
Martin Galway spending an entire month coding the guitar solo for the song "Times of Lore" (1988) (Commodore Zone). When the technical challenges were greater, it was more difficult for a composer to be "in the flow" and not struggle against the system in comparison to today's environments. However, even with very limited tools, composers such as Rob Hubbard were able to produce composing environments in which they were able to bring the responsiveness of a system into a workable range. Sometimes, such as with my own composition practice, composers choose to use older software, such as trackers, to produce chiptune music to preserve this compositional environment and limitations to create chiptunes.

Composing a song with assembly language, burning it onto an EPROM (erasable programmable read-only memory) for an NES is very different from making an NES song on the PC with FamiTracker, putting it on a PowerPak via a memory card and playing it on a pro-sound NES system of our "second age" of chiptune. Learning the assembly language of the NES is difficult and time-consuming. As such, the notion of composing a chiptune "from scratch," using just the base system as composers originally using the system would have had access to, is a challenge indeed. Making an instrument from scratch, such as the gAtrai by C-trix, is both a technical and artistic feat. One can see that there is a scale of increasing difficulty between creating chiptunes from scratch and simply triggering a chiptune sample from a sample library. Table 30.1 attempts to outline how combinations of hardware and software produce a difference in the general difficulty level of producing a chiptune.

In a live performance setting, the physical and software interface between the performer and the music becomes even more important to allow for a balance between the amount of real-time control parameters and the complexity of the music produced by the system. Traditional instruments such as the violin require significant manual practice to gain an acceptable amount of control over the inputs for a live performance that would be pleasing to a typical audience. With electronic music it is easy to possibly replay a very complex song by simply pressing a single button. It can be very difficult for the audience to know what is "live" and what is not. With live chip tune music, most performers appear to try to strike a balance of musical spontaneity and serendipity with repeatability and spectacle of the overall performance. As a performer, I find that performances are less enjoyable for me when I am overwhelmed with performance possibilities, and I prefer to use DJ techniques such as crossfading, song selection, and other, similar practices to allow me to be entirely physically and mentally present in the performance, while having enough flexibility to quickly respond and react to the feel of the audience. If the performer is feeling awkward or not having fun, then it is difficult for the audience to feel otherwise.

### 30.6 The Demoscene

In the early days of the Commodore 64, a rising number of people began to pirate games and trade them in meetings or via online bulletin board systems (Paul 2005). Pirate groups would often include introductions during the load screen of the cracked software.

---

**Table 30.1 The Chiptune scale**

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Sound chip</th>
<th>CPU</th>
<th>Composition Software</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most difficult</td>
<td>ATtiny85</td>
<td>ATtiny85</td>
<td>Custom: coded in assembly language</td>
<td>1-bit Symphony by Tristan Perich (Tristan Perich 2009)</td>
</tr>
<tr>
<td>Original Commodore Original C64 MOS 64 (C64) MOS 6881</td>
<td>Original C64-MOS 6510 (6560 series)</td>
<td>New: Goattracker</td>
<td>Original: custom (Rob Hubbard's sound routine and assembly language)</td>
<td></td>
</tr>
<tr>
<td>Original C64 MOS 6850</td>
<td>New: evaluates C64 CPU on PC: HardSID4u</td>
<td>New: Goattracker</td>
<td>New: Goattracker</td>
<td></td>
</tr>
<tr>
<td>New: Hardware emulation—C64D1V (no filters)</td>
<td>New: evaluates C64 CPU by C64D1V</td>
<td>New: OpenMPT</td>
<td>Running Goattracker with emulased SID Retro City Rampage Soundtrack (virt, Freky DNA and Nomin Radd) Sound examples from Plogue Chipsounds Using a chiptune sample library in a song</td>
<td></td>
</tr>
<tr>
<td>New: Samples and synthesis</td>
<td>New: host system</td>
<td>New: host system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easiest</td>
<td>Sampled</td>
<td>New: host system</td>
<td>DAW</td>
<td></td>
</tr>
</tbody>
</table>

To show off their skills they would produce increasingly complex graphics and music during these cracking introductions (or "cracktrios"). Eventually, these groups began to compete with each other on technical and artistic merits with demos that illustrated their skills. Over time, a lot of great music was produced for these competitions. The core of the demoscene started and remains in Europe, but there are small groups throughout the rest of the world. These competitions (or "comps") continue to this day and are a great way for chiptune artists to prove their skills to their peers and create a stronger community. With its origins in pirating, the demoscene is primarily noncommercial, and this aspect has spread to much of the art created by the community, including its music productions. Although demo parties often play compositions live to their audience, it is focused on the notion of real-time computational functions to generate the productions, so human performances were not a focus of these demonstrations or "demos."
30.7 Live Chiptune

The second generation of chiptune is much more focused on live performance than the first generation of chiptune musicians, who were often involved in the demoscene. One of the better-known events for live chiptune began in New York City for the Blip Festival in 2006 and has spread to locations such as Japan and Australia among other countries. The film Reform at the Planets provides a good insight into the chiptune music scene that is centered on New York City. The New York scene has more of a DIY culture than a coder culture and has democratized the sound of chiptunes with the growth of relatively easy-to-use cheap and accessible tools, such as Little Sound DJ (LSD) on the Gameboy. Chiptunes can be seen as a rebellion against the overproduced sounds of pop music and a return to a sound of electronic music by electronic instruments. This scene is in direct contrast to the existing DJ culture of the time, with its focus on the star power of the DJ rather than the composers who made the music. The chiptune community tends to be humble and accessible in sharing how they create their music, and performances tend to include a large network of peers who have an in-depth understanding of how chiptunes are made.

30.8 The Chiptune Genre

Styles and genres can blur, with bands such as Anamanaguchi, who use the NES as part of their sound to be "chip-punk," or the indie rock band The Advantage, who primarily play cover versions of NES songs. In the case of Anamanaguchi, chiptunes can be viewed as just another element or instrument within the band. For The Advantage, when the synth channels of the NES are orchestrated for guitars, bass, and drums one can hear the influences of the styles of the 1980s music that influenced the original compositions, rather than just hearing the chiptune sound.

Emulation of sound chips is always a tricky topic for chiptune artists. However, when one sees the game system as an instrument similar to a violin then it is easy to see why musicians are so particular about the sound. I believe it comes down to subjectivity, but there is something to be said for the nonlinear sound that comes with the actual hardware instead of an emulation that has a real difficulty with analog circuits. When listening to the sound of real SID chips in comparison to an emulator, I find that there is a richness to the sound quality of the original chip that is inspiring to work with, in contrast to the sound of an emulator. Each time a song is played, it is slightly different on the real chip, providing it with a certain mystery and life.

The Commodore 64 SID chip is a great example of this phenomenon, as each chip sounds slightly different, due in part to a design issue with the first version of the 6581 chip. Some chiptune artists prefer the original 6581 chip rather than the subsequent 8580 chip that corrected certain flaws but doesn't have the same loudness and noise that allows for the common method of generating digitized sound via software. The comparison between various revisions of a sound chip on a pattern system doesn't end with the Commodore 64, however, and similar comparisons have been made on the Gameboy and other systems. In the case of the Gameboy, musicians have even gone to great lengths to give an objective argument for their preferences of which Gameboy revision is best (Weixelbaum 2012).

Usually, chiptune music is programmed note-by-note using a tracker instead of being played live. For example, on a Gameboy there are a very limited number of input buttons to interface with the tracker, so it creates a very dense style, similar to trying to put a lot of information within a single text message. To write a novel—or even a few pages—by texting on a phone keypad also breaks up the message, similar to the way tracker music is broken up between patterns of a longer score.

The use of trackers influences the musical composition. Since patterns are often looped or reused in a tracker file, often phrases will not span a pattern and so shorter patterns and melodies tend to dominate over longer solo or melodic lead phrases which are more difficult to compose. The tracker composer will typically loop a single pattern while editing it. Looping a subsection of connected patterns within a song is often difficult with most trackers. It also takes more time to review changes when looping a larger section of the song. For example, if a pattern repeats every four seconds and one element is changed, it just takes four seconds to review the change. However, if a longer solo section of sixteen patterns is being looped and edited, then it requires a minimum of $16 \times 4 = 64$ seconds to hear that change. Conceptually, it is more difficult musically to keep track of the other elements one is attempting to focus on within that sixty-four-second loop, which in turn increases mental complexity and strain during composing, which is less pleasurable in comparison to the nearly instant gratification of hearing a change at most four seconds later.

There are also other user-interface constraints common to the tracker. It is often possible to visually see almost all of a single small pattern on a screen, whereas seeing the details of sixteen patterns becomes problematic. There is simply a cognitive disconnect between seeing the edited notes visually and being able to hear them at one time. This visualization is in direct contrast to traditional musical notation which is very good at visually displaying a long melodic articulated phrase. The tracker composer can watch the notes stream by in "follow mode," but it is often a visual blur with higher tempos. It is possible to drop the tempo during composition and programming and increase the tempo for playback, but often the articulation density at a lower tempo causes strange overmodulated effects when sped up, and the melody can easily become stilted rather than emotive.

It is also very rare for a tracker to allow a level of zoom so that the composer can select a level at which to view the composition besides at the pattern-note level and the pattern-arrangement song level. All that is commonly available to trackers is to change the font size and to hide effects columns to allow the display of more channels simultaneously. This setup is in contrast to most piano-roll style MIDI editors, which allow the composer to easily zoom in and out of a composition on a vertical and horizontal
aspect to visualize long note changes over time or easily see a harmonic relation vertically between notes for chords. Here we can see that there are many qualities of tracker music composition which cause a noticeable result in chiptune compositions.

The tracker file format is uniquely constructed to be a near-optimal music format for games. It has the advantage of requiring very little CPU overhead for playback, the playback and creation tools are robust, the tools are typically open-source, and the files themselves are optimized in their use of sample data. A distinct advantage of working with tracker files is that one can easily change the tempo dynamically, and the quality of the playback of the music doesn't become distorted in the way that a linear sampled sound recording does. The instruments can be switched while the song is playing back, and each track can be muted or adjusted in volume in real time, providing many possibilities for adaptive music playback systems for games. The definite drawback of tracker music is that it requires a significant amount of work to achieve a production quality that is as polished as the music from a typical digital audio workstation (DAW). However, the tracker file format is flexible and can include layers of additional information coded in the MIDI format that can be used to send messages to the game in synchronization with the song such as beat or phrase marker information.

As trackers continue to improve in quality and allow for more sophisticated digital signal processing (DSP) effects, the possibilities for using tracker files in videogames continues to grow. With the growth of the mobile market, tracker files continue to provide a good balance of music production quality, file size, and CPU requirements.

### 30.9 Chiptune Composition

There are many ways to compose chiptunes and the next few sections will investigate a cross-section of various techniques that have been used throughout chiptune history. The simplest form of digital sound generation is 1-bit music which has roots in early computer music history. The basic synthesis capabilities of the NES are then examined to show how the first game console sound chips were originally used and how they are used for composition thirty years later. The Commodore 64 SID chip is then covered to show how its capability of dynamically changing oscillators on its channels, and the possibility for real-time filtering, changes one's chiptune composing style for this system. To conclude, the use of OpenMPT to compose NES-style chiptunes is detailed in reference to the soundtrack for the modern game *Retro City Rampage*.

#### 30.9.1 1-bit Music

One of the attractive elements of chiptune music is its relationship to minimalism. In fact, with PC speaker music or 1-bit music one gets to the fundamental, "atomic" element of sound, in which either the speaker has been "clicked" or not. All one does is organize a series of clicks through time; very minimal indeed. This clicking on and clicking off of the speaker is commonly referred to as "beeper" music.

The history of beeper music dates back to the early days of computers and the origins of computer music itself. Beeper music has seen a small resurgence lately due to the increased availability of trackers for 1-bit music as well as the popularity of the "1-bit Symphony" music release by Tristan Perich that encases an entire hardware music playback system with 8kb for forty minutes of playback. The release is interesting in that each "performance" renders the music each time it is heard and the hardware electronics of the playback system are small enough to be distributed in a regular compact disc case.

In the 1980s, when it was expensive to add a soundchip to an already expensive IBM PC and when costs of the ZX Spectrum were to be kept down, the soundcard was effectively eliminated entirely and the generation of sound was entirely dependent on the CPU for oscillating the series of bits sent to the speaker. Of course, even the most basic sound routines would allow the description of a series of tones over time. The more sophisticated sound routines allow for polyphony and a varied number of effects, such as phase modulation, speech synthesis, volume and envelopes, and different timbres. Continuously supplying the bit-stream for the speaker can be taxing on the CPU and many games would include music only on the title screen.

The sound of 1-bit is determined by the software rather than the hardware. Since the software is changing values so quickly over time, any small change in the manner in which the sound is generated often has a very noticeable effect on the result. Artists such as Mister Beep consider them conceptually almost as different instruments based on their capabilities (Mister Beep, personal communication, 2012). There is an interesting dichotomy between the minimal amount of information required to produce the audio on the hardware and the large amount of digital information required to accurately emulate the sound of the actual system.

#### 30.9.2 The Nintendo Entertainment System (NES)

The next level of complexity for chiptune composers is having access to different waveforms, which can be illustrated by the NES. The NES had four fixed synthesis channels and a 1-bit DPCM sampled sound channel. In terms of compositional elements, the composer has two pulse waves, a triangle wave (fixed volume), noise channel, and DPCM channel, with no effects.

It should be noted that it is possible to affect the volume of the triangle wave (and noise channel) by the information in the DPCM channel. The "frequency" of the noise is the timbral result which makes it sound as though the noise channel is increasing in pitch through sixteen different values, and the type of noise can be switched between two different timbres with one sounding quite buzzy and being infrequently used. The effect of changing the pulse width on the pulse channels is a 2-bit value but since two of the values cannot be perceptually differentiated, there are functionally three different timbres of the pulse channels in practice. Certain NES game cartridges expanded the
audio capabilities of the NES, but these games are rare and the nostalgia surrounding the NES sound typically references the stock NES audio hardware.

This brings up an interesting point in that the actual audio hardware of a cartridge-based system is expandable by the chips contained in the cartridge itself. The most popular of the expansion chips is the VRC6, which allows for two more pulse channels and a sawtooth channel. However, in the overall history of the NES, the majority of the games used the basic soundchip, as adding extra soundchips in the cartridge was an additional cost.

A composer, then, really has access only to the timbre of the note and how loud it plays (sometimes) for a total polyphony of five sounds. There is also the issue of sound effects that can play at any time and steal a channel (or more), so sometimes composers would use just the DPCM channel for sound effects and the rest for music. The problem with the DPCM data is that the sample memory ROM was costly and often limited to less than 16 Kb. Adding more ROM to the cartridge for samples increased the cost of the cartridge and therefore you either had voices of your music dropping out when sound effects played or your samples were quite limited in memory size, implying that the sampling rate of the samples was low, the number of samples was low, or both, in an effort to conserve ROM. The DPCM channel was also unable to perform real-time sample rate conversion and the playback rate was limited to sixteen set values and if you wanted to use it for a melodic instrument, the sampled data would have to be included at different pitches.

With these constraints of the channels being fixed and the volume being locked on the triangle wave, often composers make guitar-type melodies on the two pulse waves with an interval relationship between the two, a baseline on the triangle wave and drums on the noise channel. Since there are few possibilities of effects to modulate the sound, NES compositions tend to focus on melody and less so on rhythmic elements. It was also released two years earlier in Japan than the rest of the world, so much of the sound of the early NES soundtracks have a strong Japanese sound to them. The "Konami Style" is a musical style of the Japanese-based game developer Konami's sound team that composed music for many games during the years that the NES was popular. Chiptune composers from today, such as Jake "virit" Kaufman, often reference the Konami style in their contemporary compositions.

In the early 2000s, it was difficult to compose for the NES and so "music macro languages" (MML) and other similar systems were often used. An early NES tracker was Nerdracker (2003) followed by the release of FamiTracker at the end of 2005, which brought in support for expansion chips and other features. In combination with chiptune artists such as Anamanaguchi utilizing the NES for their songs and the ease of making NES songs, it has become a much more popular platform for composing in the last few years.

### 30.9.3 The Commodore 64 (C64) Sound

In contrast to the NES, the Commodore (C64) has a different set of cultural beginnings and hardware capabilities, which result in a very different sound than the sound of the NES. The Commodore 64 was released worldwide in July 1982, about a year before the NES was released in Japan. The C64 spread quickly to become one of the best-selling computers in history with affordable cost and high value. Although there were many talented composers from around the globe using the C64, the European composers primarily set the style for the computer. United Kingdom composers Rob Hubbard and Martin Galway arguably set the pace for other C64 composers to follow. The C64 style was much more centered on timbral shifts and rhythm than the NES, simply because the waveforms could be switched on each channel and the filter allowed for subtractive synthesis, which was not possible on the NES. Unfortunately the filter design was unstable in the early version of the SID chip (MOS 6581) and therefore some composers opted not to use it at all (Carr 2001). However, once the SID design was updated to the new MOS 8580 chip, the filter bug was fixed, but unfortunately it also had a side-effect of reducing the effectiveness of the way most "digis" (sampled sound) had been done until then. Since there was a lack of channels, fast arpeggiation was also a trademark of the system. Another trademark of the C64 was its software digitized samples that required additional processing time from the CPU. The 12-bit phase resolution on the pulse wave oscillator also allowed for a dynamic phaser effect not available on the NES. Ring modulation was also an option but not overly used as it reduced the number of effective voices to two, due to one voice being consumed to be used as the modulation signal. Overall, these capabilities meant that the C64 was often best suited to a clear lead line on one voice, arpeggiation for backing chord progression timbre on a second channel, and often the bass and drums were left to alternate on the last channel.

One of the common instrument programming approaches was to compose a sound from multiple oscillators in sequence on a single channel—such as using a quick noise burst followed by a triangle wave to simulate a bass drum. Filtering across multiple channels allowed many different sonic possibilities especially when utilizing different filter types. Changing the resonance and the center frequency of the filter allowed for a great degree of control over the filter. Depending on how tricky one gets, almost any real-time effect is possible on the C64, from time stretching to autotune to vocoding as demonstrated by by Pex "Mahoney" Tufvesson in October 2010 in which eleven real-time effects ran on an unmodified C64 (Tufvesson 2010).

#### 30.9.4 NES Chiptunes Using OpenMPT

Much of my own interest in the NES has been driven by being one of the composers and sound designer for the NES-style indie game Retro City Rampage. When I was asked to develop the sound for the game, I looked into various options, and since the game wasn't running on an actual NES, the strongest toolset for making mod music at the time was Open ModPlug Tracker (OpenMPT). In addition, since we wanted composer Jake Kaufman to be involved, we decided that his format of making NES-style tunes using the impulse Tracker format in Open MPT would be a great choice. Since the format was an open source, it was easy to integrate the playback routine into the game code and have it...
quickly running on several different systems. The game uses the Impulse Tracker module as a source but renders it to an audio stream during run-time, making it easy to port the game to various systems and to keep the CPU requirement to a minimum. I wanted the audio playback to occur at real time to generate the noise waveform dynamically but all of the synthesis waveforms are actually just short digitized samples. To port the music to an actual NES tracker such as Famitracker or to use a conversion program such as IT2NES would take some time but the quality of the songs would remain quite similar to their sources.

Recently, I have begun composing in Famitracker, but I find that the interface for OpenMPT is much better aligned to the way my mind works during the compositional process. It is very important for the composer to be able to stay "in the zone" or "in the flow" during the artistic musical process or elements can become confused and not as focused as one might like. Jake Kaufman has switched entirely over to Famitracker for his latest release of FX4 and continues to highlight his talents as a leading chiptune composer. For the release of the soundtrack, the songs were mastered with a minimal amount of equalization but no reverb or other elements were added in the goal of keeping them sounding as NES authentic as possible.

30.9.5 Tracker Tricks

Although the NES does not have any capability for real-time effects in hardware, there are possibilities of simulating certain effects using certain tracker "tricks." Likely, the easiest audio effect to duplicate in the tracker is a delay effect which is simply done by duplicating the notes of a particular channel at a quieter volume delayed in time. For feedback, one just continues to produce time delayed notes with decreasing volume, commonly around 50 percent and eventually reaching silence after three or more echoes. This is most easily accomplished by using a second channel for the echoing notes such as the following which can be seen with the Figure 30.2 and 30.3 examples.

The most obvious difficulty that we can see here (see Figure 30.2) is that if we attempt to collapse the delayed notes into a single channel to use fewer channels then we will lose our extra delayed note at C-4 at position 8 due to the new D#6 note. The easiest way to rectify this situation is to change the delay time interleaved with the tempo of the notes such that they do not occur at exactly the same time. An example follows as #1. Here we can see that simply increasing the delay time by two rows (or one-tenth of a second) we can now represent the echo effect on one channel (see Figure 30.3). If we had built this delay effect in Famitracker and defined the echo effect in the instrument, then the delayed versions of the C-4 note would have been cut off at the first D#6 note.

One of the exciting elements of working with trackers is developing sets of tricks that add technical dexterity to your compositions. Effects include flanging, in which a note is duplicated on the pulse channels but a vibrato is applied to one of the channels. Chorusing can be done very similarly except that the frequency of one note is bent slightly and held slightly out of tune with the other note. Using a very fast vibrato can change the timbre of the pulse and triangle waves to make it sound "thicker." Rapidly changing the pulse width can give a feeling of filtering, as the number of overtones for each duty cycle changes with each pulse width but keeps the pitch of the fundamental the same. To simulate reverb, which is basically a set of quiet random delays, one can
have an initial delayed note for the first reflection and then a sequence of somewhat random decreasing volume modulations which fall to silence. There are also instrument effects which one can apply, such as including a short attack of the note that is an octave higher to simulate a plucked sound similar to a plucked string on a guitar. Depending on the phrasing, one can modulate the bends and note leaps of a lead line on a pulse or triangle wave to sound like an electric guitar. A more complex timbre can be created by triggering sounds on multiple channels, such as the noise channel and DPCM channel being triggered together to give an accent on sampled drum hits with the higher frequencies of the noise channel. The possibilities are endless, just as with any other form of composition, and being able to code your own tracker allows for an even greater level of control which is often prized by chiptune composers.

The composition program used by all composers for Retro City Rampage was the free and open source OpenMPT for the Windows OS. This tracker has advanced GUI capabilities over older DOS-based trackers. Using the impulse tracker (.it) format allowed the soundtrack for Retro City Rampage (which is currently approximately two hours and twenty minutes long) to occupy just 4MB of game data. A typical MP3 compression scheme would normally require well over 100MB of storage.

30.9.5.1 An In-depth View of a Tracker Song

This blazing rock theme opens with a heavy power chord riff on the two pulse wave channels and continues to build with the addition of a bassline on the triangle wave and drums on the noise channel and DPCM 1-bit lo-fi sampled sound channel. Detailed articulations in the volumes and portamentos give a classic Kaufman (virt) feeling right from the outset. A drumroll at the end of the first pattern kicks off the main drum pattern. The riff breaks into a climbing line with an octave to possibly simulate a pinch harmonic or feedback in the guitar line. A dueling solo follows between the two pulse waves, punctuated by interval hits and slow bends that compliment the detailed bass line and tom rolls. At a detailed level, relentless use of alternating duty cycles, volume gating, echoes, and subtle frame delays punctuate virt's mastery of the NES 808 guitar fusion rock form. When the riff returns at 0:32, it plays a background role with a complimentary riff to layer another lead onto the theme. Almost hidden in the riff at the end of one of the patterns is a clever use of octave arpeggios which serve to change the timbre of the pulse wave without adding any confusion regarding the melodic intervals of the song. At 0:48 heavy power-chord break hits on all instruments and weaves back into another riff. When this rhythm riff repeats, the notes are punctuated even higher as it continues to grow with finger-tapping frenetic parallel lead lines. When the main riff returns at 1:04, the lead grows to astronomical seven-string Steve Vai guitar heights and never misses a step. At 1:20, paired power chord slides push even higher with more drum rolls and searing leadlines. By 1:36 we are dropped into a triplet power chord thrash fest. By the time we arrive at 1:52 to recap the main theme, we realize that we are just getting prepped for the satisfying loop to the beginning!

The screen grab of the score in OpenMPT from 0:53 to 0:57 at 120 bpm (Figure 30.5) provides a good indication of the amount of detail that is present in every second of this section of the song. It begins in the pulse wave channel with a deep vibrato using the H8f command from position 78 to 83. The DPCM samples show how the noise channels are often paired to add more high-frequency transients to each hit, which was commonly used by the Konami sound team. The kick drum is instrument number 22 at C-5 and the snare is D-5 in the noise channel. If this were an actual NES, they would need to be defined as separate instrument samples as there is no sample rate conversion in the DPCM channel. The tom drum sample is instrument 33 in the DPCM channel. Once again, on the NES, each tom drum sample with a different playback rate would need to have its own sample. The combinations of commands SD2 and SD3 are used to slightly delay the onset of the note by a number of ticks show rapid triplet patterns (for example, in the pulse 1 channel from positions 96 to 102) which help break up the quantization often found in less accomplished tracker songs (or electronic songs in general for that matter). Each row that appears in this pattern is three ticks (one-sixtieth of a second) so SD2 is a one-sixtieth-second delay and SD3 is a two-sixtieth-second delay. A slide down in the pulse 1 wave is predicted by a strong vibrato from 110 to 122 and falls from positions 123 to 127. While all of this is happening, a small quiet unison riff with the bassline on the triangle channel is occurring on the complimentary pulse wave from 112 to 127, with quick octave trills using the JCo command repeated using J00 to change the timbre to a "plucked effect." Using the octave makes the note stand out and provides more sonic energy to the note similar to when a guitar string is plucked and there is an emphasis on
The upper harmonics at the initial pluck of the note. The repetition of the C-5 in the pulse channel can be seen as an open string where notes are articulated on the same string.

As one can clearly see with a small detailed look at approximately four seconds within a 120-second composition, the amount of detail and technical dexterity demonstrated is staggering, especially when one considers that the entire soundtrack for the game spans over two hours of total music.

### 30.10 Conclusions

We live in an increasingly digital world and the sound of chiptune helps us have a conversation with the machines that construct our surroundings when they used to have their own unique voices. There is a personality to working with older digital systems that has been largely lost in the internet age but chiptune is doing its best to reclaim a personal human to machine connection.

If you are at all interested in chiptunes, then the best thing to do is to listen a lot to existing chiptunes and to start making your own. The appendix provides a good starting point to listening to good examples of chiptunes on many different systems from an old-school perspective of the first time around, when songs were made on the hardware, and the new-school approach of utilizing the system within modern contexts.

I feel that chiptunes are a type of folk-art that anyone can enjoy by downloading free software for their computer or buying an older game console from the pawn shop and experimenting on their own. Chiptunes are a look into our own kinder, gentler past with machines and give a new language for the future that will continue to evolve with each generation.

### A Listener's Guide to Chiptunes

This is not meant to be an exhaustive list but just an overview of the more common chiptune platforms. There are also many chiptunes using the Roland Sound Canvas, the Gravis Ultrasound, Commodore Vic 20, Amstrad CPC, the Atari Lynx, and many more. Chiptunes tend not to be made on older arcade console hardware due to their often unique set of chips and the need to burn custom EPROMs to change the audio data.


---

**Figure 30.4** In-depth song analysis: score from 0:53 to 0:57.
This is not meant to be an exhaustive list but just an overview of the more common chiptune platforms. There are also a lot of chiptunes utilizing the Roland Sound Canvas, the Gravis Ultrasound, Commodore Vic 20, Amstrad CPC, the Atari Lynx, and much more. Chiptunes tend not to be made on older arcade console hardware due to their often unique set of chips and the need to burn custom EPROMs to change the audio data.

<table>
<thead>
<tr>
<th>System</th>
<th>Signature sound</th>
<th>Old-school examples</th>
<th>New-school examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amiga</td>
<td>Aliasing noise (upsampling artifacts), limited polyphony of four hard-panned two left and two right, samples from the ST 01 and higher sample set, eight-bit crunchy sound</td>
<td>Phantasmasoria (Amat 1990), Substance (Moby and ATZ 1991)</td>
<td>Acid Chunks (Manley 2012), Acidized Evening (Suni 2000)</td>
</tr>
<tr>
<td>Atari 2600</td>
<td>Two simple sounds, characteristic noise sound, notes are often very much out of tune</td>
<td>Desert Falcon (Bob Polaro 1988), Pitfall II (David Crane 1982)</td>
<td>Disasteradio—Fax Of Life (Disasteradio 2009), Booty Plus Plus (Trix 2012)</td>
</tr>
<tr>
<td>Atari ST</td>
<td>Simple waves often combined with eight-bit samples</td>
<td>Astrofah: The Angel of Death (Hippel 1989), Craftman (de Man 1993)</td>
<td>Ancient Art of Chiptune (qwer 2008), Dynamite (stu 2003) I’ll be a Pimp in Cabin Green (Randall 2008), Edge of Disgrace (Dane 2008)</td>
</tr>
<tr>
<td>Commodore 64</td>
<td>Noise channel quality, ‘gloppy’ filter, sometimes a very distorted sample</td>
<td>Arkanooid (Galway 1987), Monty on the Run (Hubbard 1985)</td>
<td>Masata Gestralt (Zen Albatross 2011), Love Bit (Toriella 2012)</td>
</tr>
<tr>
<td>Gameboy</td>
<td>Two pulse waves, wave channel (usually triangle) and noise channel</td>
<td>Tetris (Hirokazu Tanaka 1989), Kirby’s Dream Land 2 (Hirokazu Ardo and Tadashi Ikemagi 1998)</td>
<td>Masata Gestralt (Zen Albatross 2011), Love Bit (Toriella 2012)</td>
</tr>
<tr>
<td>Gameboy Advance</td>
<td>Two pulse waves, wave channel, noise channel, and two sampled sound channels (FM sounds with Nanoloop)</td>
<td>Sonic Advance 3 (Sonic Team 2004), Mega Man Zero (Ippo Yamada 2002)</td>
<td>Out-House #12 (Henry Homesweet 2011), Live @ Blip Tokyo (10/23/11) (Je Devens E) en 3 Jours 2011</td>
</tr>
<tr>
<td>MSX</td>
<td>Three pulse waves, noise channel</td>
<td>Metal Gear 2: Solid Snake (Theme of) (Masatoshi Iwakura 1997), Space Channel (Konomi Kukeishi Club 1989)</td>
<td>SYNCtified (Inverse Phase 2012), Lucky Star (Itoaya 2008)</td>
</tr>
</tbody>
</table>

(System) (Continued)

- **NES**
  - Fixed pulse, triangle and noise channels, common intervals between pulse channels, triangle as bass and noise as kick, snare and high hat
  - Super Mario Bros. (Kondo 1985), Shatterhand (Mizutani and Yamanashi 1991)

- **Sega Genesis**
  - Sampled sounds (usually for drums) combined with chimes, filtered bass, wood hits and bell sounds on the FM channels plus sometimes some simple waves
  - Adventures of Batman and Robin (Jesper Kyd 1994), NBA Live 95 (Traz Damji 1994)

- **SNES**
  - Eight channels of compressed ADPCM with reverb
  - Panel de Pon—Sea Stage (Masaya Kuzume 1995), Earthbound (Keichi Suzuki and Hirokazu Tanaka 1994)

- **ZX Spectrum**
  - Typically 1-5 channels of very buzzy sounds
  - And so it is (Yerzmyey 2000), Astro Marine Corps. (Jose Antonio Martin Fello 1989)


(Continued)
Ir is worth revisiting the distinction between traditional and new media. In the former, content such as images or sound is acquired by measurement for subsequent playback, a process of data sampling (Nyquist 1928), and so we refer to such media as being data-driven. At various stages the sampled data may be transformed, or edited. It may, for example, be filtered to change its tone and color, or sliced, and the pieces rearranged in time. Sections may be faded in or out, such that they overlap smoothly. They may be stretched or shortened in duration, or effects added to create false sensations of space and atmosphere. All of these aspects are features of nonlinear editing and are the staple functionality of modern digital software for producing video and music.

New interactive media, of which computer games are an important class, do not predefine the timing, order, or quality of visual and audio events. Their final form is deferred until run-time. Choices made by the player through interaction as well as random elements, shape the flow of the experience, limiting, expanding, and selecting possible paths of play. Even the supremacy of the event itself may be challenged, and a model based on a continuum of experience substituted.

The director of ordinary media obtains almost total control over the experience of the user. This control includes fine details such as the color mix of the projection and the acoustic properties of the theater. Issues of cultural subjectivity aside, crafting this experience is precisely the acknowledged role of the film director, music producer, or other person exercising artistic control in a project having definitive form. Similar to the designer of a rollercoaster ride, the rider is a captive audience, a point passing along a precise and carefully planned trajectory.

Contrasting this user experience with the experiences of a computer game player, we see these vary from guided or semilinear to completely free. The semilinear design is like that of a maze, where the player makes path choices at prescribed junctures. The latter form is analogous to an adventure playground or sandbox, in which the player roams freely, interacting with objects at will. Contrasting some artistic demands for