VIDEO GAMES AUDIO [work in progress]

Karen E. Collins kcollins@liv.ac.uk

Introduction

Last year, when I first told a professor of music at a well-established university that I was interested in researching video games music, he scoffed "but that's so simple!" It was then that I realized not only how little appreciation video game music has received, but also how much misunderstanding there is about games music. Most of us who grew up in front of several generations of games consoles take for granted that games music has come a long way from the blipping and bleeping of the early machines. After all, even the notoriously conservative National Academy of Recording Arts and Sciences has now allowed for Grammy awards for interactive games music.[1] In fact, many film composers and popular music groups are becoming increasingly involved in the production of games music, and soundtracks to video games are finding a considerable audience amongst music fans, as more and more games soundtracks become released as music CDs. In fact, the music industry is discovering video games as a new channel for marketing bands, as artists like Andrew W.K., Good Charlotte, and Trust Company have successfully gained exposure through the medium.[2] Popular games developers Electronic Arts' *NBA Live 2003* has recently become the first videogames soundtrack to achieve platinum sales (one million units).

Like the music, the video games industry has come a long way since the early 1970s. It is now a multi-billion dollar business—bigger than the film industry, more popular than television.[3] Nintendo's Mario character has earned twice as much as all five Star Wars movies combined, and is more recognizable to American children than Mickey Mouse.[4] Despite this overwhelmingly important influence of video games on Western culture, there is still surprisingly little academic research into the subject. Although there have been many studies on the effects of game violence, and increasingly on the culture of video games or arcades, academic institutions have been slow to react to the importance of games, and research into games audio is virtually nonexistent.

There has been an increasing number of popular books detailing the history of video games, particularly the "golden years" (1976—1984) of Atari, Colecovision and Intellivision. The most detailed of these historical accounts is the incredibly detailed and thorough Steven L. Kent's *The Ultimate History of Video Games*. I have intentionally left out coverage of the history of video games in this account, and refer interested readers to this book. However, in the 600–page tome, there is not a single entry for music in the index, and only two short entries for sound.

This research aims to provide a basic foundation for further, more detailed work on the development of games audio. This study is necessarily basic and in some cases overly simplified to the extent that it assumes the reader has little or no knowledge of video games, not to patronise more experienced readers, but to be open to all who wish to gain access into this medium. With a broad readership in mind, a glossary is provided at the end.

Without significant previous academic research into games audio, this study is therefore indebted to the many fans who research, collate and collect information on the systems or games, and reproduce it on the internet, including copies of original manuals, technical specifications on sound chips, and detailed discographies of game composers. In particular, I am indebted to Jörg Weske's "Digital Sound and Music in Computer Games" article. I have tried to supplement the information he has provided with some more detail here. The majority of factual information in these pages can be sourced somewhere on the internet. Although facts have been verified where possible with manufacturers, due to the nature of the internet and the fact that some of these manufacturers no longer exist, there is a possibility of some factual inaccuracy in what follows.

---

[1] Game soundtracks may be submitted for best song, best instrumental composition for a motion picture, television or other media, or best soundtrack album.

[2] http://www.digitalgamedeveloper.com/2003/02_feb/news/dleas2403.htm

[3] Video games sales in the USA and Europe has reached a yearly $17 billion. In Britain, video games sales is 60% more than box office sales, and 80% more than movie rentals. US Americans voted video games as favourite form of entertainment—twice those who voted for television, three times those who voted going out to movies, and six times those who voted for renting movies (Poole: 6).

[4] Wired magazine jan 2003 p. 104, Sheff: jacket cover.

This study is arranged in two parts. The first part follows a basic timeline for the development of games audio technology. I have included discussion of both home computer and games consoles, which work effectively the same, with the exception that the advantage of the console is that the composer may know exactly how a game will sound on anyone's system, while PCs are all configured differently, and therefore mean there are variations between units. The second part analyses games audio based on genre… (to be continued)…

Explanation of Game Genres

There does not yet seem to be any real consensus on game genres: Mark J.P. Wolf's most recent work on the definitions of genres includes some forty-two different genres.[5] The disparity in definitions and divisions is most likely because most games fall into more than one category. I have categorised games based on that provided by the Interactive Digital Software Association (IDSA), the industry's major marketing information source. According to the IDSA, the following are the most popular video games genres in 2000/2001: Child; Family; Sports; Action; Strategy/RPG; 1st Person Shooters; Other Shooters; Fighting; Racing/Driving; Arcade; Action; Simulation; Adventure.

The IDSA offers no definition of these genres, however, and so I have provided some temporary working definitions here. As my goal is not to discuss genre explicitly at this stage, I have made several adjustments to the categories provided by the IDSA to simplify the taxonomy. I have not used "child and family", as these are specific to the market, rather that the type of game, so these fall under whichever type of child/family game they are (many are "platform"). "1st person shooter", "other shooters", "action" and "fighting" are combined as "action/shooter". "Role Playing" has been combined with "adventure" as they often cross over, and "strategy/puzzle" games are dealt with separately. I have combined "simulation" with "racing", as racing is essentially a driving simulation game. I've added platform as a category, which are no longer popular, but very popular in the 1980s (such games are probably now divided between categories as action/adventure). I define the genres as follows:

1. Sports mimic or are similar to existing real or imaginary sports.

2. Action/shooter Games in which one or more players spends a majority of time battling other players or a computer-player. The confrontations may take place between human characters, monsters, tanks, airplanes, etc. These may be hand-to-hand, shooting or other combat/fighting games where the purpose of the game is to fight, or they may be combined with some adventurous task, where the "fun" of the game is in the battle, rather than the quest.

3. Strategy and Puzzle Games in which the primary goal is to solve a series of problems or enigmas.

4. Platform Also known as "run and jump", "games in which the primary objective requires movement through a series of levels, by way of running, climbing, jumping, and other means of locomotion. Characters and settings are seen in side view as opposed to top view, thus creating a graphical sense of "up" and "down" as is implied in "Platform" (http://devscape.org/form_genres.php?ID=11) .

5. Role Playing Games (RPG) & Adventure Games in which players create or take on a character or persona and proceed through a series of connected screens/rooms, often having to solve a puzzle to advance. Some are first-person perspective: we do not see the character on the screen. Text-based adventure games generally do not have sound, so are not discussed here.

6. Racing/SIM Simulate a "real-life" experience, including flying planes, building cities or driving cars.

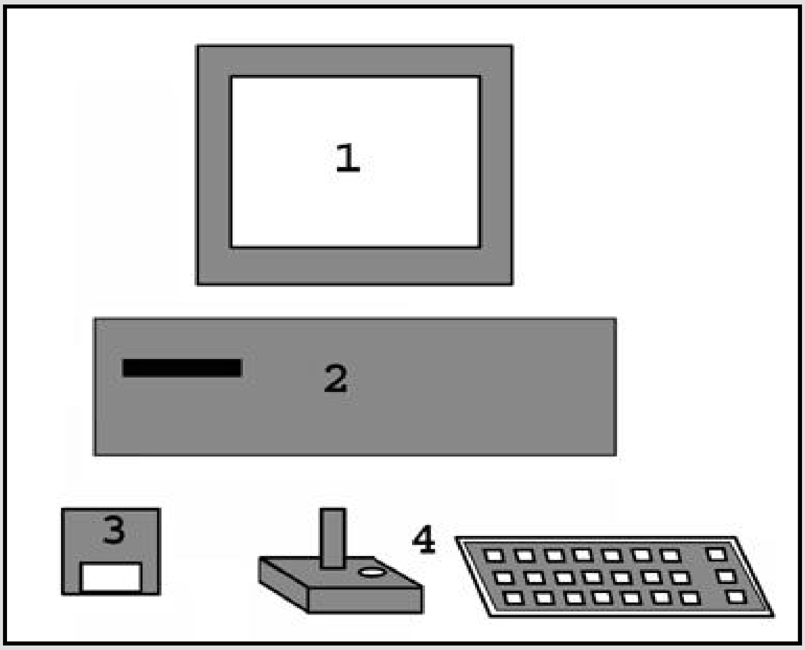
[5] http://www.utexas.edu/utpress/books/wolmed.html

Timeline

1971	First arcade video game, *Computer Space*
1972	Magnavox Odyssey // *Pong* // First international connection to ARPANET // Atari formed
1974	MITS Altair 8800—first PC
1975	BASIC written. Microsoft formed
1976	Coleco Telstar console // Fairchild Channel F: first programmable console
1977	Atari 2600 (VCS) // Apple II
1978	Nintendo formed // *Space Invaders* released // Atari 400 and 800 computers // Odyssey 2
1979	Commodore PET
1980	Mattel Intellivision // *Pacman* (Namco) // DOS (Microsoft)
1981	*Donkey Kong* (Nintendo) // IBM PC
1982	Colecovision (Coleco) // Commodore 64 released
1983	Famicom computer (Nintendo's NES) released Japan // IBM PC-XT
1984	Mattel dumps Intellivision: "video games crash" // Apple Macintosh // Commodore Amiga
1985	NES released USA // *Tetris* // Microsoft Windows
1986	Sega Master System // Atari 7800
1987	VGA // Ad-Lib Soundcard released
1988	Coleco bankrupt // Apple Macintosh II
1989	Gameboy // NEC TurboGrafx 16 // Sega Megadrive // Atari Lynx // World Wide Web // Sound Blaster // SVGA
1990	*Super Mario 3*
1991	SNES // Linux // Atari Panther (32-bit)
1992	Sound Blaster 16
1993	3DO Console (Panasonic) // Atari Jaguar (64-bit) // Intel Pentium
1994	Super Game Boy
1995	Sega Saturn // Sony Playstation // N64
1996	Atari ends
1998	Sega Dreamcast
1999	PowerMac
2000	Playstation 2
2001	X-Box (Microsoft)

Game System Basics

The game console system has the same basic set-up as a pc.



1. Display The output is displayed on either a computer monitor or television screen. Monitors and screens vary in size, colours available and resolution, and there are differences between areas of the world—PAL is the format for Europe, and NTSC is the North American format.

2. Main unit/console The main system or game console contains the Central Processing Unit (CPU) and various supporting integrated circuits, or chips. Most consoles have dedicated chips to handle graphics and sound. The CPU processes all instructions and data, and is driven by an internal clock, the speed of which is measured in Mega Hertz (MHz). The CPU relies on Random Access Memory, or RAM, which is essential for running software; in the case of games, RAM stores temporary information such as what level the gamer is at, or score, etc. Consoles, like PCs, also have an operating system (such as DOS, or Windows) which allows for the interfacing between hardware, and allows game programmers around the world to write games using the same tools.

3. **Carrier/medium** The carrier is the medium that holds the game. It could be a DVD, a CD-ROM, floppy disk, or a cartridge, for instance. Most current systems use DVD or CD-Rom.

4. **Input device/controller** The user control interface allows the player to interact with the video game. For instance, a paddle, a joystick, a mouse, a controller, or a keyboard.

Early Video Games: The 8-Bit Revolution

The earliest video games—those going back to what is allegedly the first, William Higinbotham's tennis game of 1958, and Steve Russell's *Spacewar* (see Kent: 2-14)—had no sound. Likewise, the earliest home consoles—such as the Magnavox Odyssey—also were without audio. Atari's *Pong* (1972), an early electronic table tennis game, was the first video game to feature sound, making the beeping "pong" sound when the ball hit the paddle. As Stephen Kent reports, the pong sound was reported by designer Al Alcorn as an accident:

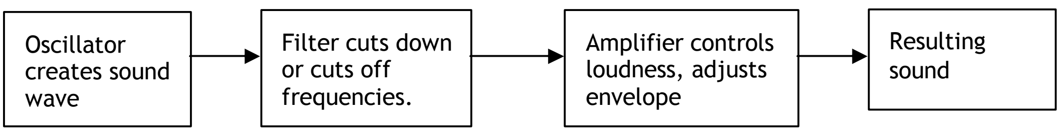
> "The truth is, I was running out of parts on the board. Nolan [Bushnell] wanted the roar of a crowd of thousands—the approving roar of cheering people when you made a point. Ted Dabney told me to make a boo and a hiss when you lost a point, because for every winner there's a loser. I said 'Screw it, I don't know how to make any one of those sounds. I don't have enough parts anyhow'. Since I had the wire wrapped on the scope, I poked around the sync generator to find an appropriate frequency or a tone. So those sounds were done in half a day. They were the sounds that were already in the machine" (Kent: 41-42).

Taito/Midway's *Space Invaders* (1978) was the first major arcade success, and also the first game to include background "music" along with special effects: a marching of alien feet that sped up as the game progressed. Along with the music, the game had six other sounds: a missile shot, explosions, a strike on a saucer, a flying saucer sound, invader hits, and the bonus missile base. The *Space Invaders* sound board (using Norton Operational LM3900 quad amplifiers)[6] was relatively unique: rather than using a sound generator which is amplified through a common amplifier, the SI chip had its own amplifying circuit.[7]

These beginnings of interactive audio—and those of their contemporaries, dealt with below—only provide the vaguest of clues as to what was to come. Sound was slow to develop, and if one recognizes the poor graphics quality of those early games compared with those of today, it's not a major leap of the imagination to realize the sound was equally poor, if not worse. PCs in particular were considered business machines, and audio was not seen to have many business applications, and was therefore not a priority for computer developers. Nevertheless, there were many progressive ideas introduced to the audio of 8-bit machine games.

The majority of 8-bit machines (and early arcade and pinball machines) used what is known as Programmable Sound Generators, or PSG sound, silicon chips designed for applications which generate sound based on a user's specifications. These specifications were usually coded in assembly language, and early sound programmers and musicians needed to understand programming language to engage the chip. This meant that most early composers were in fact programmers working on other aspects of a game, or at best, in-house programmer-musicians who had to work closely with programmers working on other aspects of the game. PSGs offered little control over the timbre of a sound, usually limiting sounds to single waveforms without much ability to manipulate that waveform. Many of these PSGs were created by Texas Instruments or General Instruments, but some companies, such as Atari and Commodore, designed their own sound chips to improve sound quality.

Early PSGs used what is known as analogue synthesis, or subtractive synthesis. Subtractive synthesis starts with a wave form created by an oscillator, and uses a filter to attenuate or subtract specific frequencies and then passes this through an amplifier to control the envelope and amplitude of the final resulting sound:



Many of the PSG sound chips could use a form of sampling either through Pulse Width Modulation (PWM) or Pulse Code Modulation (PCM).

[6] See http://www.du.edu/~etuttle/electron/elect21.htm for technical details.

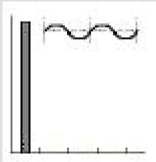
[7] Midway 8080 System Boards http://spaceinvaders.retrogames.com/html/midway_systemboard.html 2001.

With PCM (otherwise known as raw, or AI2 synthesis), essentially, an analogue sound is converted into digital sound, by taking many samples—in other words, measuring the amplitude at regular intervals—of an analogue waveform. The data is stored in binary (1s and 0s), which is then decoded and played back as it was originally recorded. The fidelity of the sound depends upon the sample rate or quantisation—the number of bits representing the amplitude. The method is still used, for instance for DVDs or CDs, where the sample rate is 44,100 times per second (44.1 K), or 16-bit. Most early games could only sample at a maximum of 22,050 samples per second.[8] The down side of this method was the amount of space required to store the samples: as a result, most PCM samples were limited to those sounds with a short envelope, such as percussion. Another solution was what is known as adaptive difference PCM (ADPCM). With the ADPCM method, the difference between two adjacent sample values is quantified, reducing the pitch or raising the pitch slightly, to reduce the amount of data required.

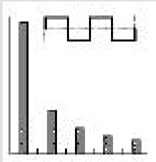
PWM modulation works by changing pulse waves by outputting pulses at a constant volume, while the width and spacing of the pulse gives the effect of different frequencies and volumes, controlled by the modulating signal's amplitude. The PWM method could attain higher volume, and achieve a range of interesting timbres (such as a pseudo-chorus or phasing sound), but the samples were low quality.

[8] See http://mushi.net/hanashi/gmintro.html.

Types of Sound Waves

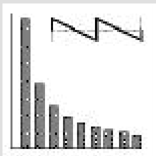


Sine waves have a single frequency—a pure tone, without harmonics. Sine waves are used to create whistle or test tone sounds—the Pong sounds were sine waves.

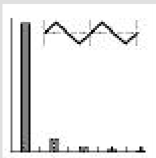


Pulse waves have odd harmonics only. Synthesisers that generate pulse waves allow you to change the duty cycle, changing the number of harmonics. Pulse channels have 4 duty cycle settings: 87.5%, 75%, 50%, and 25%, although the difference between 25% and 75% is indistinguishable to the human ear. As a Pulse Width duty cycle increases from 50%, it sounds increasingly brighter, and as it decreases it becomes thinner, and more “nasal” sounding.

Square waves are created by adding only the odd numbered harmonics (f, 3f, 5f, 7f) with amplitudes equal to 1/n (1, 1/3, 1/5, etc.). A square wave is a pulse wave with a 50% 'duty cycle' or a 50:50 ratio. It has a hollow, woody but bright sound, often used for wind or organ sounds.



Sawtooth waves have both odd and even-numbered harmonics (f, 2f, 3f, 4f), with amplitudes in the ratio of 1/n (1, ½, 1/3, ¼, etc.). The sawtooth wave is often described as rich and brassy sounding, and is often used to mimic brass instruments.



Triangle waves have few harmonics—the odd harmonics (f, 3f, 5f, 7f, etc.) at decreasing intensities (amplitudes in the ratio $1/n^2$: 1, 1/9, 1/25, 1/49, 1/81, etc.). The triangle wave produces a tone similar to the sine wave, but slightly warmer as it has a few harmonics. Its sound resembles that of a recorder or tin whistle.

PERSONAL COMPUTERS

IBM PC XT

Name, Date and Place	IBM 5160 PC/XT Stock Specifications USA 1983
Processor	8088 Intel 4.77MHz
RAM	128k-640K
ROM	128K to 256K
Graphics and Colours	4 color CGA.
Operating System	MS-Dos
Sound	one channel single voice tone generator (bipper). No volume control.

International Business Machines (IBM) introduced their Personal Computer (PC) in April of 1981. Intended largely for business use, the original PC had several advantages over its competitors in the personal computer market. IBM's upgradability and open architecture—meaning the computer used standardized protocols and components—meant that third-party manufacturers could develop products for the PC system, and even build their own PC systems, called clones (e.g. COMPAQ, who was the first to introduce such a system in 1982). The IBM PC XT ("eXtended Technology") was the successor of the IBM PC, and expanded on the original PC design by improving hard disk capability and memory.

The first IBM PCs and clones contained only a tiny speaker that could produce simple tones (typically square waves) of varying frequency but of a fixed volume, designed to indicate errors or other messages, sometimes referred to as a "bipper" or a "beeper". Later, add-on sound cards would be designed as upgrades for early PCs, such as Mindscape's "Music Board" of 1986, a six-voice sound card.

In 1988, Access Software developed a PWM technique for using a normal 3" PC speaker to play relatively realistic digital audio samples, without the addition of a sound card, by vibrating the internal speaker back and forth at a very high pace, to simulate the position between On and Off. Using Realsound, it was possible to create 6-bit digitized samples, which was commonly used for sound effects or for title music in games. More importantly, it could do this using a minimal amount of the processor's power. Realsound was used in a few popular games, such as *Mean Streets* and *Space Racer*, but failed to catch on, as many PCs of the time came with small "tweeter" speakers, rather than the original 3" PC speaker, and there was no need for Realsound technology when soundcards became popular.[9]

Most early games for the PC were coded in BASICA, otherwise known as Advanced BASIC (Beginner's All Purpose Symbolic Instruction Code), which was the official BASIC of IBM, and was included with every PC-DOS diskette set.10 BASIC as a language for game programming was somewhat limited on the old machines, which had trouble managing sound at the same time as executing other tasks. The processors could only maintain static sound, or, in some cases, an uneven dynamic sound, and the separate channels—for those PCs that had them, would occasionally experience delay due to the slow execution of the commands.[11]

The BASICA programming language contained three commands to play sound. The first was BEEP, a simple alert beep which could not really be modified and therefore was not used for music. The SOUND command could be played with to create various sound effects, though it was limited in that it could only modify frequency and duration. The PLAY command would allow for the programming of simple melodies, and allowed for the modification of frequency, duration, rests, note attack types, and tempo, and even made allowances for dotted notes. Chords could be simulated by rapidly arpeggiating notes. Even vibrato could be simulated through rapidly altering the pitch.

---

[9] http://www.oldskool.org/pc/sound/

[10] MS-Dos later came with Gee-Wiz, or GW BASIC.

[11] See also http://www.tu-chemnitz.de/phil/hypertexte/gamesound/pcsound-main.html.

PROGRAMMING SOUND IN BASIC

(adapted from http://www.geocities.com/qbasicstation/tutor/advanced/a_play.html)

The **SOUND** Command

SOUND frequency, duration

Frequency range was from 37 to 32767, although this varied depending on the computer or compiler used. Duration ranged from 0.1 to 65535.

For example, the following creates a sound effect of two short effect notes.
SOUND 600, 3
SOUND 50, 1.5

THE **PLAY** COMMAND

Code	Purpose
Ln	Length, or duration of the note. The variable n does not indicate an actual duration amount but rather a note type; L1 - semibreve, L2 - minim, L4 - crotchet, etc. (L8, L16, L32, L64). By default n = 4. range is 1 - 64.
On	Sets the current octave. Valid values for n are 0 through 6. An octave begins with C and ends with B.
MN ML MS	Stand for Music Normal, Music Legato, and Music Staccato—slows down or speeds up the music. MN - Note duration is 7/8ths of the length indicated by Ln. ML - Note duration is full length of that indicated by Ln. MS - Note duration is 3/4ths of the length indicated by Ln.
Pn	Creates a rest (pause) for the length of note indicated (same as Ln): P1 is a semibreve rest, etc.
Tn	Tempo sets the number of crotchets in a minute. Valid values are from 32 to 255 (default value is T120).
. (period)	When placed after a note, it causes the duration of the note to be 3/2 of the set duration, to created dotted notes.
MB MF	Music Background and Music Foreground. MB places a maximum of 32 notes in the music buffer and plays them while executing other statements. Works very well for games. MF means that while music plays, nothing else is going on in your program (everything stands still).

Letters C, D, E, F, G, A, B are tones, where C is the lowest and B the highest tone.

< and > move down or up by one octave.

(#) or (+) turns the previous note into a sharp, which means that it increases frequency by a semitone. (-) reduces the previous note by a semitone.

Eg. 1. Twinkle Twinkle Little Star: (1st two bars) MusicA$ = “mFt120o2p4l4ddeel2dl4o1ccbbaal2g”

Eg. 2. Beethoven's Symphony No. 5: MusicB$ = "mBt180o2p2p8l8gggl2e-p8l8fffl2d"

THE PC JR AND TANDY 1000

Name, Date and Place	IBM PCJR and TANDY 1000 USA 1984.
Processor	Intel 8088 4.77MHz
RAM	64k, expandable to 1 MB (Tandy 256K)
ROM	64K
Graphics and Colours	16-colour Enhanced CGA (CGA+)
Sound	3-voice TI-SN76496, built-in speaker.

The IBM PC had been largely designed for the business computer market. By 1983, seeing the success of companies such as Commodore and Apple (see below), IBM realized that their business could be expanded into a wider market by making the PC more accessible to the needs of the home user. With this in mind, the IBM PCjr was launched in 1984. To better compete with the other home PCs being marketed at the time, several changes were made to the original PC. In addition to the usual floppy disks which had to be booted by command lines in DOS, the PCjr used cartridge ports that could be loaded up by sticking the cartridge into the port and turning the computer on—much easier for children. Enhanced graphics and sound illustrated the importance of video games in the home computer market.

Although the PCjr had several failings which held it back from great success,[12] these problems were improved upon by its popular clone, Radio Shack's Tandy 1000. Tandy had enjoyed previous success with the TRS-80 model, and customers could upgrade the older model to the 1000 model by replacing only the base unit. The Tandy 1000 retained the enhanced 16-colour graphics and 3-voice sound of the PCjr, making it popular with gamers and game developers in the early 1980s. Eventually, over three hundred games would be released designed for the Tandy 1000/Pcjr.

The Tandy 1000/PCjr's sound enhancement over the original PC was created by the Texas Instruments TI-SN76496 chip, a three-voice tone generator, with an additional fourth noise-generating channel for sound effects. All four voices had independent 4-bit volume control (15 levels plus 0, off). Each of the three tone voices had an independently selected frequency (limited to 110Hz to 111860Hz) and tone envelope. The fourth, the noise channel, contained three pre-selected frequencies producing white or periodic noise, and a fourth option which was to borrow the frequency of the third tone channel.

IBM, in an effort to market the PCjr, hired US American company Sierra to produce a PCjr game that would show off these newly enhanced colours and sound capabilities. Sierra had previously created graphic and text adventures for the PC, such as *Ulysses and the Golden Fleece*, *Mystery House*, and *The Dark Crystal*. Sierra's answer to IBM's assignment was *King's Quest*, the first "3-D" graphic adventure game. But Sierra went one step further, creating an Adventure Game Interpreter, or AGI, which would become the standard for programming Sierra's popular graphic adventure series such as *Space Quest*, *Police Quest*, and *the Leisure Suit Larry* series. The games still had a text interface where users would type in commands, but now had the addition of moving a character about on-screen, using the keyboard, joystick, or mouse (for the Apple). The AGI format was designed for and modeled around the PCjr's sound chip, using all available sound channels. The PCjr, Tandy and Macintosh versions of the songs were composed of four parts; the melody, two accompaniment parts, and one noise (generally for sound effects). The games could also be played on the IBM PC, although the PC's single channel could only play the melody portion (first channel) of the songs.

---

[12] Unfortunately, the PCjr had several failings—the 'wireless' keyboard, and expansion capability of various modules were problematic, and the cartridges held less memory than a floppy disk. http://www.oldskool.org/shrines/pcjr_tandy/

The Apple II

Name, Date and Place	Apple IIc. April 1984. USA
Processor	MOS 65C02 1.02 MHz
RAM	128K
ROM	16K
Graphics and Colours	Various, mono or 16 colours.
Sound	one channel bipper

The Apple II was designed for home use and released in 1977, developing through several different models throughout the following decade. The original Apple II series had similar sound capabilities to that of the early PCs—a one channel bipper for warnings and errors. The graphics, however, were better than those of the PC and attracted many gamers.

As with the PC, third-party sound expansion boards were available for the Apple II, the most popular of which used General Instruments' AY-3-8192 PSG chip (which would also find its way into the Spectrum 128, Amstrad CPC, Atari St and Mattel's Intellivision and is discussed more fully below), which provided three tone channels and one noise generator. Sweet Micro Systems' Mockingboard was the best selling of these sound boards to use the AY-3-8912, and was available in four different packages: The Mockingboard "A", the base card, which added 6-voice music and sound synthesis. The Mockingboard "B" worked with the "A" by adding speech synthesis capabilities via the Votrax SC-01 Speech Synthesizer or the SSI-263 speech synthesizer. The Mockingboard "C" was essentially an "A" and "B" packaged together. The later Mockingboard "D" had the same capabilities as the "C", but attached to the Apple IIc via the serial port. Applied Engineering's Phasor was another popular of Apple II sound board, and followed the Mockingboard, even providing a Mockingboard emulator. The board included four AY-3-8913 sound chips, creating a total of twelve voices and four noise generators. A speech synthesis card (SSI263 AP N8513) and space for upgrading with an additional speech chip were also included.

The Apple IIgs (The Cortland)

Name, Date and Place	Apple IIgs. USA September 1986
Processor	Western Design Centre 65SC816 2.8 MHz
RAM	256K to 8MB
ROM	128K to 1 MB
Graphics and Colours	2 bit color at 640x200, 4 bit color at 320x200
Sound	Ensoniq 5503 Digital Oscillator Chip

The last member of the Apple II line, the Apple IIGS, a 16-bit machine, was also the most powerful in terms of sound. The GS stood for graphics and sound, both of which were greatly enhanced in the IIGS model. Apple originally planned to have a four-voice sound chip for the GS, until Apple marketer Rob Moore suggested using a chip from Ensoniq's Mirage synthesizer. The chip required a separate 64K block of dedicated RAM, upgradable to 128K, although few motherboards supported this.[13]

Far ahead of computers of the time, the Ensoniq 5503 Digital Oscillator Chip (DOC), designed by Robert Yannes (creator of the Commodore 64 SID chip, see below), contained 32 oscillators. These oscillators were paired (for flexibility and better sound quality) to create fifteen generators (two were reserved for timing); in other words, the chip was capable of synthesizing fifteen simultaneous voices. Each oscillator had a 16-bit frequency register and an 8-bit volume register. Apple's operating system, System Software 6.0 released in 1988, included a MIDI

[13] The chip was also used in the Ensoniq keyboards—the Mirage, the ESQ-1 and the SQ-80, there were no synthesizers manufactured without the full 128K.

interface and a sequencer program, SynthLAB.[14] Generally, music scored for the later IIGS games would use PCM samples (for sound effects) and MIDI sequences.

The Commodore 64

Name, Date and Place	Commodore 64 USA 1982
Processor	MOS 6510 (8-bit) running at 1.0 MHZ
RAM	64k
ROM	20K
Graphics and Colours	16 colour 320 x 200 Sprites: up to 8 per line
Sound	SID 6581

Commodore International Limited was founded in Toronto by Jack Tramiel in 1955. The company originally manufactured typewriters and adding machines, and later the first pocket calculator. Commodore moved to Palo Alto California in 1975, purchasing the microprocessor manufacturer MOS Technologies. Commodore enjoyed success in 1977 with its home PET model, but really began to make waves in the computer market when it came out with the VIC-20 in 1980. Originally intended as a game machine, the VIC-20, assembled by Robert Yannes, was based on a MOS 6502 processor and a Video Interface Chip which would allow for advanced graphics. In addition, it had a relatively advanced sound card for its time, with 3 square wave and one noise channel.

After the success of the VIC-20, in 1982 Commodore released its 64K model, which would go on to become the best selling computer of all time, having sold an estimated 22 million units. The C64 was originally conceived of as a game computer, and the graphics and sound remain evidence of this. The C64's microprocessors supported two high-resolution graphic modes, smooth scrolling, bit mapping, character collision resolution and character mapped graphics. The sound chip (called SID, Sound Interface Device) was a three-voice plus noise generator chip, created by Robert Yannes in 1981, who had helped engineer the VIC-20, and would later go on to create the DOC (above) for the IIGS. Each tone on the chip could be selected from a range of waveforms—sawtooth, triangle, variable pulse (square wave), and noise. Each tone could also be subjected to a variety of effects and programmable filters including ring modulation. An independent ADSR amplitude envelope modulator enabled the SID to more accurately imitate other instruments than previous chips. The noise channel could also operate as a simple PWM sampler (by modulating the volume of a voice so fast, a 4-bit sampled sound is created).

THE 8-BIT CONSOLES

8-bit consoles, like PCs, had single PSGs, mostly with three-plus-one channels of audio, and, like PCs, the audio channels had to be downmixed to one mono output. Programmers for 8-bit consoles ran into many of the same difficulties as PC game programmers. There was little room in games for music. At best, a title song or short victory song may have been used. The Intellivision for instance had a cartridge space of 4K, and audio was given maybe ten percent of this space. Atari VCS games were lucky to have any music at all. As the 80s progressed, however, cartridges could hold larger amounts of space, and audio improved.

Magnavox Odyssey II/ Videopac G7000 (EU)

Processor	Intel 4-bit 8048 CPU 1.78 MHz
Sound	One channel sound The voice Chip: SP-0256-019 & SPR-128-003

[14] http://www.callapple.org/apple2/software/tribby/midisynth.html

Ram	64 Bytes CPU/128 bytes EXT
Graphics	280 x 192 16 colors

The original Magnavox Odyssey, which had twelve games, was limited in its capabilities: it was black and white, the sprites were blocks, dots, or straight lines, and there were no sound effects or background graphics, but the Odyssey still managed to sell hundreds of thousands of units. On the success of the Odyssey and competing systems at the time (such as the Fairchild Channel F, Vectrex, Astrocade and the Atari VCS), the Odyssey 2 was built in 1978. The Odyssey 2 came with twenty-four built-in games, but was again limited in capabilities. Compared to the Atari VCS, the Odyssey had lower resolution graphics and had only one audio channel, although it did include a touchpad keyboard for educational games. A later add-on called The Voice of Odyssey 2 was developed to offer two sound channels with speech synthesis, but without the full support of parent company Philips, and problematic hardware issues (such as the fact that the joysticks were not removable, making repairs a mess), the Odyssey 2 failed to generate the kind of interest that Atari had witnessed. The Odyssey 2 did have some success in Europe, where it was marketed as the Videopac G7000, and a later model the G7400, and it reportedly sold over one million units, and was supported by over fifty games.

Atari VCS/ 2600 and 7800 1977, 1986

Cpu	2600: 1.19 MHz 8-bit Motorola 6507 microprocessor 7800: Motorola 6502C 1.79 MHz
Ram	2600: 128 bytes 7800: 4K
Graphics and sound	TIA (Television Interface Adapter). 2 voice, 4-bit volume control.

The Atari VCS (Video Computer System, also later known as the 2600) saw limited success when it was first released, and the first few years of production saw few games created for the console. In 1980, however, Atari licensed the popular arcade game *Space Invaders*, which became a best seller and helped to spur on the sales of the VCS. Eventually, over 25 million VCS systems were sold, and over 120 million cartridges.[15] To keep up with competition, Atari would later release the 5200 and then the 7800. Although the 7800 had advanced graphics capabilities, it retained the poor-quality sound chip of the 2600. Combined with the fact that the 7800 was released two years after it was ready to ship (1986), the console failed dismally to compete with other consoles of the time.

The sound chip in the 2600 was known as the TIA chip, and also handled graphics. The audio portion had two channels, each with a 4-bit volume control, a 4-bit waveform control selector and a 5-bit frequency divider (capable of dividing a frequency of 30KHz by 32 values). This 5-bit frequency controller was incredibly limited, and used a polynomial counter, also known as a Linear-Feedback Shift Register (LFSR). LFSRs are type of binary counter that uses a pseudo-random way of counting, rather than the normal binary incremental/decremental sequences. The number was divided down from the system clock, meaning many pitches were not in tune with others, making it difficult to program melodies. It has been suggested that this lack of programmability meant that the Atari 2600 had more original songs, as copying songs from well-known music was difficult if not impossible.

Intellivision 1980 clones Radio Shack Tandyvision One, the Sears Super Video Arcade, and the INTV Master Component (also called INTV System III and the INTV Super Pro System)

Processor	General Instruments [GI] CP1610 @900kHz
Sound	AY- 3-8914 General Instruments Intellivoice 3303
graphics	GI AY-3-8900-1 160 x 96 pixel in 16 colours, 8 sprites

[15] www.emuunlim.com/doteaters/play3sta1.htm

Mattel’s answer to the Atari VCS was the Intellivision (Intelligent Television), more advanced in sound and graphics. Also important was its modular design, allowing for extensions such as the Entertainment Computer System, a music keyboard and second sound chip, allowing for six simultaneous channels. The original Intellivision used a General Instruments PSG sound chip, the GI AY-3-8914. Part of a series of GI chips popular with game makers (including the AY-3-8910, 8912 and 8913, see above: the chips were used in the Sinclair ZX Spectrum, Amstrad CPC, BBC Micro, Atari ST, Sega Master System and many arcade machines), the 8914 had three tone generator channels and one white noise. Each channel allowed for individual control of frequency, volume and envelope. Pitch was still controlled by the frequency-division method like on the 2600, but the GI chip used a twelve-bit register to set the divisor, allowing for 4096 possibilities instead of only 32 in the ATARI 2600. This mean the Intellivision could more create more recognisable renditions of known songs, such as Bill Goodrich’s use of “Flight of the Bumblebee” (Rimsky-Korsakov) on the game *Buzz Bombers*.[16] An add-on chip was released in 1981, the Intellivoice, using a GI speech synthesis chip called the Orator, containing 16K ROM space for voice data. By the late 1980s, a program was created by Dave Warhol that could convert MIDI files directly into Intellivision music code, but by that time Intellivision had seen its peak success.[17] Counting the Intellivision II, and clones, over three million Intellivision consoles had been sold.

Colecovision 1982

Cpu	Z80A 8-bit 3.58MHz
graphics	TMS9928A 256x192 16colour. 32 sprites
sound	Texas Instruments SN76489 3 channel + noise, mono
Ram	8 K video ram 16K

Coleco (Connecticut Leather Company) created their own console to compete with the Intellivision, once more improving on the available capabilities of their competitors, and allowing for the “porting”, or conversion, of more games from arcades. Colecovision consoles were shipped with the Nintendo arcade hit Donkey Kong, which helped it overtake sales charts. Colecovision used a Texas Instrument SN76489 chip comparable to that of the Intellivision, but the 10-bit frequency divider limited tones to 1024 possible pitches. It had three square wave tone channels and one noise generator with an option of periodic or white noise. It also was possible to play samples on the chip using either PWM or PCM methods.

Nintendo NES/Famicom/Dendy

Processor	6502 8 bit system 1.79Mhz
graphics	256x240 16 colours 64 sprites
RAM	2 kb
Sound	Built in 2A03 (a modified 6502) mono 3-channel + noise + pcm

Undoubtedly the most popular of all 8-bit machines was the Nintendo Entertainment System, known as Famicom in Asian markets and Dendy (or Dandy) in Russia. Nintendo originated as a Japanese playing card manufacturer back in the late 1800s, but the company turned its focus to toys starting in the 1970s. The NES was released in 1983 in Japan, but was not released in North America until 1985.

The NES used a built-in five-channel PSG with one waveform for each channel—two pulse waves, a triangle wave, a noise, and a sample channel. The pulse and triangle channels had an 11-bit frequency control, capable of about eight octaves. The pulse channels had four duty cycle options and a 4-bit amplitude envelope function, and one of

[16] http://www.tu-chemnitz.de/phil/hypertexte/gamesound/history-main.html

[17] See http://www.webcom.com/inty/retrotopia/hifi.shtml

the channels had a frequency sweep function that could create portamento-like effects, and was often used for UFO or laser-gun sound effects. The pulse waves could be detuned, and vibrato effects could be simulated by the programmer.[18] The triangle wave channel was one octave lower than that of the pulse waves, had a four-bit frequency control, and had no volume or envelope control, as it was mainly intended for bass (ibid). The noise channel could use white noise or periodic sounds created by the four-bit frequency control. It also had a lopped noise setting that was used to create metallic sounds (ibid.).

The fifth channel is a PCM channel sampler, also known as the Delta Modulation channel (DMC). There are two basic methods of sampling using this channel. The first is pulse code modulation, discussed above.[19] The second is known as direct memory access (DMA). DMA sampling is only one-bit and works by playing samples from memory, and updating the digital to analogue converter (DAC) by recording only changes in a waveform, then accesses a portion of memory and temporarily halts the CPU to retrieve the sample.

NES sound and music was generally allotted about ten percent (4-40Kb) of the total game size,[20] although some games used in-cartridge sound chips to add extra channels. For instance, the Konami VRC6 and VRC7 chip, used in *Akumajou Densetsu* (Japan only, known as *Castlevania* in USA) and *Lagrange Point*, generated sound using FM synthesis (see below).

Handheld Consoles

Handheld video games became popular along with other video game consoles. Nintendo allegedly released the first handheld console in the mid 1970s, which was known as the "game and watch" system. Early "table top" games—small, one-game arcade ports—were popular in schoolyards in the early 1980s. These were generally without sound, or very limited in sound capabilities. The Atari Lynx (1989) was the first colour hand-held game, and had four-channel 8-bit digital sound. NEC released Turbografx Express a year later, which was basically a portable version of the Turbografx-16 (see below). Both of these models failed to generate any real consumer excitement. Sega's Game Gear was released in 1991, using the SN76496 (3 + 1 channel) chip that was found in the PCjr and Tandy models described above. The Game Gear enjoyed a degree of success, seeing some 250 games released, but certainly the uncontested leader of handhelds is the Nintendo Game Boy.

Nintendo Game Boy

	Game Boy (1989)	Game Boy Colour (1998)	Game Boy Advance (2001)
CPU	8-bit Z80 1.5 MHz	8-bit Z80 2.1 MHz	32-bit RISC-CPU, 16 MHz + 8=bit CISC-CPU 2.1 MHz
Memory	8kb int, 8kb ext	8kb int, 8kb ext	32kb + 96Kb vram int., 256Kb Wram ext.
Sound	4-channel stereo	4-channel stereo	4-channel stereo
Graphics	160 x 144 pixel, 4 shades gray	160 x 144 pixel, 56 sprite mode, 32 000 bitmap	240 x 160 pixels, 511 sprite mode, 32 768 bitmap

Nintendo's Gameboy, which through its various guises has seen over 100 million units sold, received an initial boost in sales when it was released in 1989 with the inclusion of the *Tetris* game. Gameboy received its first update in 1996 when Gameboy Pocket was released—essentially a more streamlined version of the original but no major changes. The second update came with Gameboy Colour two years later. Though this model included improved graphics, the sound remained the same for these three models; a three-plus-one channel stereo sound PSG, the fourth channel offering 4-bit noise, with simple envelope adjustment feature. The Game Boy Advance, released in 2001, also added two 8-bit digital-to-analogue converters (DACs) to the original configuration, and channels 3 and 4

[18] http://www.student.oulu.fi/~vtatila/history_of_game_music.html

[19] The PCM method is used for speech, such as in Tengen's *Gauntlet 2*.

[20] http://www.seanspace.com/iSphere/scores.htm

could also operate as 4-bit DACs to play samples. Not only the hardware, but the software was also improved: the Advance BIOS contained many sound-related functions for converting MIDI to Game Boy data.[21]

EARLY ARCADE

Arcade machines did not differ significantly from home consoles of the contemporary period, although they were often slightly further advanced in graphics and audio. Early arcade machines used many of the same chips discussed above (particularly the AY-8910 series), in other words, they used subtractive synthesis PSGs, often with a DAC to enable the use of samples. The first significant advance came with Atari's Pokey, the Potentiometer and Keyboard Integrated Circuit, responsible for the controlling of the paddle or joystick and keyboard (Pots and Keys) as well as audio. The four-channel Pokey chip was also used in the Atari 5200, the Atari XE and various Atari 8-bit computers (400, 800), and was also used in many Atari coin-operated arcade consoles (coin-ops), such as *Indiana Jones and the Temple of Doom* (1985). Each channel was square-wave and had a three-octave frequency register and control register (setting the distortion and its volume). The Pokey chip had various types of filters, such as high or low-pass, and allowed for adjustments to clock bases and polynomial counters. Each channel also had tone or noise selectors. Later, Atari would include a Texas Instruments 5220 speech chip, which had been used in "Speak 'n' Spell", the popular family electronic game. The 5220 used a form of audio reproduction known as Linear Predictive Coding, which could mimic human speech or be used for sound effects, like the elephants in the original *Star Wars* game. Along with the Pokey, it was also used in games such as *Indiana Jones and the Temple of Doom* (1985).

16-Bit Video Games

FM and Additive Synthesis

YM2149	3 channel + noise	Amstrad CPC, Atari ST, ZX Spectrum, some arcades (YM2149 is virtually the same as the AY-3-8910)
YM2151	8 channel + noise	Taito, Toaplan and Sega System16 games' sound chip. Also found its way into Yamaha DX-series and Korg DS-8 keyboards.
YM2203	3 channel	Taito and Capcom games' sound chip. Very popular arcade sound chip.
YM2610	3 channel	Neogeo and Taito games' sound chip.
YM2413	8 channel + noise	Japanese Mark 3 machines, and also, some European Master Systems. It supports 9 channels, each of which may play any of 15 pre-defined 'instruments', or a user-defined sound. In addition, 3 channels can be set to 'rhythm mode' which allows them to reproduce percussion sounds.
YM2612	6 + 1 digital/ noise	Used in Sega Genesis/Megadrive

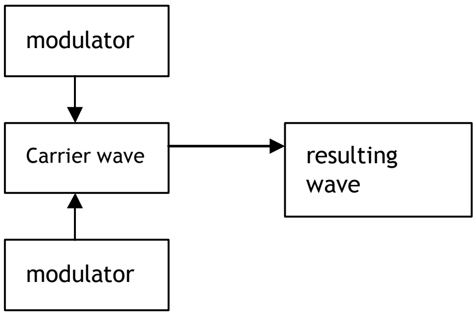
Frequency Modulation (FM) synthesis[22] was developed by John Chowning at Stanford University in the early 1970s, and eventually licensed and improved upon by Yamaha, who would use the method for their computer sound chips, as well as the DX series of keyboards. FM uses a modulating (usually sine) wave signal to change the pitch of another wave (known as the carrier). Each FM sound needs at least two signal generators (oscillators), one of which is the carrier wave and one of which is the modulating wave. Many FM systems used four or six oscillators for each sound (or voice): the Yamaha DX7 used six oscillators per voice. An oscillator could also be fed back upon itself, modulating its original sound.

FM sound chips found their way into many of the early arcade games of the late 1970s and early 1980s, and into most early computer sound cards (see below). Compared with PCM or other PSG methods, they were far more flexible, offering a much wider range of timbres and sounds, with a limited amount of memory required. In light of

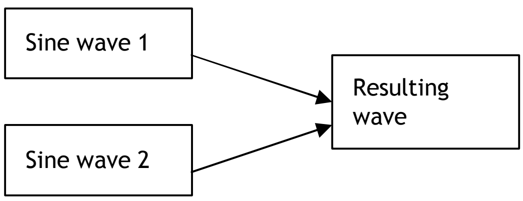
[21] http://belogic.com/gba/

[22] For a more detailed tutorial on FM Synthesis, see Computer Music's guide on http://www.computermusic.co.uk/tutorial/fm/fm1.asp

more recent technology, FM has been criticised as sounding metallic, because it could not produce natural sounding instruments, although it still enjoys popularity amongst some electronic musicians.



FM chips were also generally capable of using additive synthesis.[23] Additive synthesis (sometimes referred to as Fourier synthesis, as it is based on Fourier's theorem),[24] the opposite of subtractive synthesis, refers to the creation of a sound wave by adding together simpler sound waves (typically sine waves). Theoretically, additive synthesis was capable of producing any sound, but the sounds of instruments were still difficult to mimic, due to their complexity. Additive synthesis chips were popular in the Kawai K5 and K5000 series keyboards, though they never gained much general usage. Atari for a time built an additive synthesis chip known as AMY. AMY was designed for use in Atari's 8-bit 65XEM (music computer) model. It was an eight-voice chip with a range of eleven octaves, and had 64 oscillators. For many reasons the machine was never released and AMY left on the scrapheap.



MIDI

Musical Instrument Digital Interface (MIDI) was a protocol defined in 1982 to allow musical devices (synthesizers, keyboards, sequencers, mixing desks, computers) to be compatible in a standardised format. Only commands, rather than actual sounds, are transmitted, meaning file size is very small. A MIDI command might, for instance, tell a synthesizer when to start and stop playing a note, at what volume and what pitch, and what "voice" or sound to use. Initially, some of this information would vary greatly depending on the devices used, but in 1991 a General MIDI standard was agreed upon. This standard laid out a template for 128 instruments and sound effects, so that the same number setting would be the same on any MIDI device: so a command saying "play number 39" would always play a slap bass.

There were however several complaints about the General MIDI standard. First, the selection was limited to 128 instruments, and some of these were taken up with the seemingly ridiculous sound effects such as "bird tweet" (123) and "helicopter" (125), which may have been useful for some game composers, but for musicians were rarely usable. Roland responded by creating the GS MIDI standard, which would allow for 128 variations of each of the 128 available MIDI channels.

---

[23] The difference between FM and additive is that with FM the sine waves were converted into numbers, combined numerically and then converted back into sound, whereas in additive synthesis, the electric current of waves are combined.

[24] Fourier's theorem states that all periodic waveforms are made up of one or more sine waves.

The second major problem was the fact that most MIDI devices sounded differently: a “slap bass” on one person’s sound card might sound very different from a slap bass on another person’s sound card: the timbre, volume, or sound quality might vary. Although for game composers using consoles this was not a problem, as the consoles had the same hardware in every unit, for PC users, this meant widely varying music playback quality. Most game composers would write songs using Roland’s Sound Canvas, and this became a standard by which to compare MIDI cards. Another solution was to return to the idea of SysEx (System Exclusive Data), which had previously been used by some sound devices, and meant that each sound device would have an ID code specific to that device. The playback device could then read this code and know exactly what hardware configuration to use; in other words, what sounds and what effects were on which channel—so number 39, voice 16 might mean slap bass with reverb on a specific card. In order to address compatibility issues, the MIDI device would always default to the standard GM instrument.

ATARI 520 ST (1985)	
CPU	68000 Central Processor running at 8mhz
Graphics	capable of three resolutions, Palette of either 512 or 4096 colours depending on model
RAM	512 K
Sound	YM 2149 Software-Controlled Sound Generator capable of producing 3 voices in mono (On the STFM). The STe had stereo sound.

The Atari ST computer’s sound chip was really a step down from other possibilities that had been opened up by sound developers at Atari. Despite its contemporary 8-channel additive synthesis AMY chip scrapped (see above), the ST went with a Yamaha YM-2149, a relatively cheap chip for the time, with mono output of an 8-octave range on a mere three channels. The chip had a 5-bit envelope generator and a 5-bit DAC for a noise channel, and was also used for controlling the parallel port. What the ST did feature that helped initially keep it ahead of its competitor, the Amiga (see below), was the fact that it had a built-in MIDI port, meaning musicians could plug in their keyboards and sequencers and use software designed for the ST to compose music.

Commodore Amiga 1000	
CPU	Motorola 68000 @ 7.16 mhz (16-bit)
RAM	256Kb expandable to 512Kb internal, Up to 8 MB external
Graphics	Graphic chip capable of three resolutions, palette of 4096 colours
Sound	Paula: 4 channel stereo

The Amiga was initially an offshoot of Atari. It was developed by Jay Miner, who had worked on the Atari 2600 and created his own company, Amiga, which was taken over by Commodore in 1984. The Amiga computer went through many different models—the 500 and 2000 models were especially popular. Despite the many improvements as the models progressed, the sound hardware remained essentially the same.

The sound chip in the Amiga was known as Paula, capable of four-channel digital 8-bit stereo, which could produce fairly complex waveforms over nine octaves. Each channel had its own volume control, but no envelope generator, though each had a low-pass filter capable of absorbing frequencies over 4KHz, eliminating the “whistling” sound created in various sound signal processing. Each channel was capable of modulation, which allowed for tremolo and vibrato effects. The Paula chip was also responsible for tasks like mouse and joystick input, and floppy disk control.

Compositions were created using a software program known as a tracker, and resulting music was stored in what is known as module format, or MOD. Tracker programs worked much like modern MIDI sequencers. A tracker program would store data on the notes, volume setting, effects and instrument (like MIDI), but also record digital samples of the instruments in the actual file, limited only by the size of file (the 880Kb floppy disk).

MOD files had the advantage over MIDI, then, in that music or other sound events would sound as the composer intended, and a wealth of possible sounds was suddenly opened up for the composer to use. MOD files were also easier to program for non-musicians—like many game composers—and made it easy to sequence repetitive loops.[25]

Tracker/MOD format never really caught on because of its size, and the fact that there were nearly twenty different formats, which would allow for a different sampling rate, and different number of tracks. The sound quality of the hardware was generally not as good as MIDI components. Although there were a few game companies that used tracker format (Epic Megagames, for instance), the majority used the better supported MIDI.

The Rise of Sound Cards

Recognising that gamers and musicians wanted decent quality sound from their PCs without having to go out and buy new computers, add-on third-party FM soundcards began to develop in the mid 1980s. Sound cards were designed with the gamer in mind: they generally had a joystick game port which could double as a MIDI port with an adapter. As well, lines in/out for speakers, headphones, home stereos and microphones were often included.

The first popular PC sound card was produced by the small Canadian company Ad Lib Multimedia in 1986. Ad Lib based their card on the Yamaha FM chip, YM3812, which was a later version of the popular YM3526 used in many arcade games. It had nine FM channels, using two operators, or six tone channels and five for percussion—eleven possible instruments at once, in stereo. The use of FM synthesis techniques meant that game developers could now use a wider range of instruments and sounds. To boost sales, the Ad Lib card was packaged with software capable of playing back MIDI files ("Juke Box"), a MIDI sequencer program equipped with 145 pre-set voices ("Visual Composer"), and an FM synthesis program to design sounds or instruments ("Instrument Maker").[26]

Soon after the development of Ad Lib sound cards, a Singaporean company, Creative Technology (now Creative Labs), developed their own sound card, known originally as the Creative Music System, but marketed by Radio Shack as Game Blaster. Game Blaster had twelve channels, but they were PSG and the board was very poor compared with Ad Lib. Game Blaster was abandoned, and Creative developed the infamous Sound Blaster. The Sound Blaster was essentially a copy of the Ad Lib card, using the same FM chip, but added digital audio capabilities for sampling, and a game port.

Sound Blaster quickly became the standard for game sound after a drop in price, and the fact that Creative put out an inexpensive developer's kit for programmers.[27] After the success of Creative, many other countries released Sound Blaster-compatible cards, creating products which were sometimes better, sometimes worse than the Creative card—such as Covox's Sound Master (1989), Roland's LAPC-1 (see below), or the Gravis Ultrasound.

One of the problems with the Sound Blaster cards was the fact that it had to mix the various sound channels into one or two output channels, resulting in a loss of resolution in the sound. If a MOD file wanted to play four or sixteen simultaneous channels, this meant that the sound would come out muddied.[28] The Gravis Ultrasound, using a "GF1" Gravis-specific chip, was designed with this in mind, capable of supporting MOD and MIDI files, with dedicated digital audio outputs for 32 channels. The card worked by loading instrument patches into its RAM, rather than the traditional ROM, which meant whatever samples the user wanted could be uploaded, offering a comparable quality to the Roland MT-32 chips (below). Unfortunately, however, the chip did not support FM sound, and as such was not Sound Blaster-compatible in a Sound Blaster dominated market, and therefore fell into obscurity.

Roland MT-32 and Wavetable Synthesis

Roland had begun making soundcards for PCs in the late 1980s, with the release of their LAPC-1. Remarkable for its time, it had 32 voices, with 256 pre-set yet programmable instruments ready for MIDI use. Roland was given a

[25] http://www.queststudios.com/quest/midi.html

[26] http://www.gamespy.com/legacy/halloffame/adlib_b.shtm

[27] http://www.tu-chemnitz.de/phil/hypertexte/gamesound/pcsound-main.html

[28] http://www.queststudios.com/quest/midi.html

serious boost when the Sierra software company (see above) signed a deal with the company along with Ad Lib. Along with becoming a reseller of the products, Sierra would adopt both Roland's MT-32 soundcard and the Ad Lib as standards for their compositions, beginning with *King's Quest IV*.[29] Sierra would once again show of the capabilities of the hardware components, by bringing on board Grammy and Emmy-nominated composer William Goldstein, and, later Jan Hammer, among others, to compose for the games.

The MT-32 was a MIDI soundcard, capable of 32 simultaneous voices, with 128 pre-set instruments. What made it different, however, was that it used wavetable synthesis. Wavetable synthesis uses pre-set digital samples of instruments, usually combined with basic waveforms of analogue synths. It is therefore much more realistic sounding than FM synthesis, but is much more expensive as it requires the sound card to contain its own RAM or ROM. The Roland MT-32 used a form of wavetable synthesis known as Linear Arithmetic, or LA synthesis. Essentially, what the human ear recognises most about any particular sound is the attack transient. LA based synthesisers used this idea to reduce the amount of space required by the sound by combining the attack transients of a sample with simple subtractive synthesis waveforms. This was known as cross modulation.

Software Developments: LucasArts and iMUSE

Most of the discussion up to this point has revolved around hardware. There were also software developments that used the hardware capabilities to expand on possibilities for games composers. One such development was LucasArts's iMUSE (Interactive Music and Sound Effects), in 1991. iMUSE was originally developed for LucasArt's *Star Wars* game for Nintendo, but found use in popular LucasArts games like *Monkey Island 2*, and *Indiana Jones and the Fate of Atlantis*. The iMuse system sequenced MIDI (now it will play digital) in a way that allowed composers to organize music cues in such a way that it could jump about within a given track, to interact with what was happening in the game. It could, for instance, change volume, tempo, or add or remove instruments or sound effects in response to a given action in a game. In other words, to create true interactive games music.[30]

The 16-bit consoles

The first "16-bit" console was released by the multinational communications corporation NEC in Japan in 1987 as the PC Engine. The PC Engine, or Turbo Grafx 16 as it became known in North America, was not true 16-bit, but rather had two 8-bit processors. Nevertheless, it did have a 16-bit graphics chip, and 6-channel stereo sound with 5-bit sampling, and was popular in Japan when it was released. When it came time for a North American release, however, the Turbo Grafx 16 did not fare so well. Part of the reason for this was the fact that Nintendo had many game developing companies under strict exclusive contracts, restricting them from developing for other systems.[31] The TurboGrafx managed to sell close to one million units, but when true 16-bit systems were released such as the Sega MegaDrive, it was left far behind.

	Sega MegaDrive	**Sega Mega CD**	**Sega 32x**
CPU	CPU: Motorola 68000 at 7.61 MHz . Co-Processor: Z80 @ 4 MHz Controls sound	Motorola 68000 @ 12.5MHz, same co-processor.	Dual Hitachi SH-2 RISC Processors @ 23 MHz each, 40MIPS. Co-Processors: Genesis 68000, and Z80, Genesis 32X VDP
Memory	1 Mb ROM Area 64 Kb RAM Area	6Mb 512Kb (PCM waveform memory) 1Mbit (Boot ROM	512k (4 MBit) additional RAM to original megadrive CD memory
Graphics	64 simultaneous colors of 512 colors. 320 x 224. 2 scrolling playfields, 1 sprite	Same as mega drive	32,768 simultaneous colours on screen, Texture mapping, Hardware scaling and rotation

[29] Much of the information in this section comes from: http://www.queststudios.com/quest/midi.html

[30] to hear examples, see http://imuse.mixnmojo.com

[31] http://www.bhlegend.com/php/show.php3?article=3

	plane, 80 Sprites		
Sound	PSG chip (TI 76489), FM chip (Yamaha YM 2612- 4-operator), 6-channel stereo , 8 Kb RAM , Signal/Noise Ratio: 14dB	PCM Sound: Stereo, 8 Channels. Sampling wavelength: 32MHz max. 16-bit DAC, 8x Internal over-sampling digital filter, PCM and CD sound mixing. 10 channels stereo FM sound, plus 8 channels of Stereo 16-bit sound at 32KHz	Stereo PCM chip, Audio mixing with Mega drive sound, Additional 2 channels (8 Channels total, or 16 with SegaCD)

The first real 16-bit console was the 1989 release by Sega, the Megadrive (the North American name was the Genesis). Sega had previously enjoyed some success in Europe with their 8-bit Master System, but could not compete with the NES. Sega did, however, own a much bigger segment of the coin-op arcade market. With the release of the Megadrive, Sega challenged Nintendo's rule, eventually selling about 50 million units.

The Megadrive produced many games ported from successful Sega arcade games like *Space Harrier, After Burner* and *Ghouls N Ghosts*. The system originally came packaged with the arcade hit *Altered Beast*, but soon took on Nintendo's *Mario* head-to-head with their *Sonic the Hedgehog* character. The Megadrive also had superior sound over the NES: it had one PSG 3+1 chip to handle effects and the occasional music part (a Texas Instruments SN76489AN, used in the Colecovision; see above), as well as a Yamaha FM synthesiser chip, a YM 2612, which had six channels of digitised stereo sound, and one PCM 8-bit sample channel which was capable of a sample rate of 22KHz (the same chip used in the Yamaha DX27 and DX100 keyboards).

There were three models of the Sega Megadrive released, each one becoming more affordable. Released in 1991, the Sega CD, a CD-ROM add on allowed for much better audio quality (adding extra channels and capable of full motion video playback), and an additional Motorola 68000 processor. Despite these advances, the system was priced out of many user's range. The 32X model was released in 1994, with 32-bit power, and high color graphics, but released with bad timing and poor marketing.

Super NES/ Super Famicom

cpu	WDC 65816 16-bit: 2.68 / 3.58 Mhz 16-bit
MEMORY	RAM: 1 Mbit (128 Kbyte) Video RAM: 0.5 Mbit (64 Kbyte) Cartridge Size: 2 Mbit - 48 Mbit
graphics	Max Resolution: 512 x 448 pixels Colors: 32,768 colors Max Colors at Once: 256 Max Sprite Size: 64 x 64 pixels Max Sprites: 128 sprites
SOUND	8-bit Sony SPC700: 8 ADPCM Channels, 64KB Audio RAM

With the Mega Drive leagues ahead of the NES in capabilities, Nintendo realised that they would have to build a 16-bit system to compete. By 1991, they had developed their Super Famicom, or Super Nintendo Entertainment System (SNES). The SNES was superior in graphics and sound capabilities to the Mega Drive, and managed to sell 46 million units over the course of the product's lifetime. The SNES sound module consisted of several components, the most important of which was the Sony SPC-700, which acted as a co-processor with its own memory. The SPC-700 was an 8-bit CPU running at 2 MHz, with an attached 16-bit Sony digital signal processor (DSP), 64 Kb memory and a 16-bit stereo DAC. The DSP was essentially a wavetable synthesiser which supported eight stereo channels at programmable frequency and volume, and effects such as reverb, filters, panning, and envelope generators, and with a pre-set stock of MIDI instruments. The wavetable synthesiser stored samples in a compressed format known as Bit Rate Reduction, at a ratio of 32:9, in order to not overtax the system memory. In addition, there was typically 24Mbit cartridge memory that could be used for sound. For ease of programming, software was developed to convert PC/Mac MIDI files into files executable by the SPC-700. Eventually the SNES sold a reported forty million units.[32]

32 and 64-bit machines

1993 saw the release of the first 32-bit consoles, the Panasonic FZ-1 3DO and the Atari Jaguar. The CD-Rom based 3DO system was never really popular with gamers, in part due to its high price, although it had true-colour graphics, full motion video and 16-bit CD-quality audio. Likewise, the Atari Jaguar also had advanced architecture, but saw little success. This was in part because it was not well supported by games designers, most of whom now had contracts with Sega or Nintendo. There were, however, a few popular titles, most notably Doom, and Tempest 2000. Tempest 2000 was so popular in fact that it is one of the first games to have a separate, commercially sold soundtrack to the game, composed by Imagitec Design. Despite the limited success of the two early 32-bit machines, two other consoles would subsequently be released which would see mass appeal: the Sega Saturn, and the Sony Playstation.

SEGA SATURN 1995

cpu	2x Hitachi SH2 28.6 MHz 32-bit 28 MIPS (each) Hitachi SH1 20 MHz 32-bit 20 MIPS
ram	2MB main, 512K cd ram, 1.5MB video ram, 512KB Audio RAM
graphics	24-bit colour, VDP1 (sprite/geometry), vdp2 (background)
Sound	Motorola 68EC000, Yamaha FH1 DSP, 32 PCM Channels, 8 FM Channels

Sega released their 32-bit Saturn in 1994, a CD-Rom based machine with eight processors. The audio alone had two audio processors, running on the Motorola 68000—the same processor that had been used as the Mega Drive's CPU-known as the Saturn Custom Sound Processor (SCSP), manufactured by Yamaha. The SCSP consisted of a 32-channel PCM sound generator, capable of 44.1 KHz sample rates and a 16-bit DAC. The sound board also had a 32-voice MIDI Yamaha FM synthesiser, whose output could be mixed in stereo using a 16-channel digital mixer and timer. The only drawback to Saturn's sound system was the limited amount of RAM accorded to sound. Because audio samples had to be downloaded raw (decompressed) into the audio memory buffer of 512K, this meant there was a limited amount of space for simultaneous sounds, and so the sample rate was often reduced to conserve memory.

The Saturn saw most popularity in Japan, where it sold close to six million units: the rest of the world only bought about three million. One of the reasons for the ultimate failure of the Saturn to catch on was the competition with the Sony Playstation, a system which was cheaper and easier to program for, therefore seeing the support of more games designers.

SONY PLAYSTATION (1994)

Cpu	MIPS R3000A 33.8688MHz 32-bit
Ram	2MB Main RAM, 32KB CD RAM, 1MB video ram, 512Kb audio ram
Graphics	24-bit colour, 256 x 256 sprite size
Sound	SPU 24 ADPCM channels, 44.1KHz sample rate,

The Sony Playstation began its life as a CD-ROM add-on component for Nintendo's SNES system. Nintendo had

[32] http://www.actsofgord.com/Proclamations/chapter01.html

joined forces with Sony to better compete in the video games market, but the two companies could not agree on the system. Sony decided to press ahead with its own 32-bit system, the Playstation. The PlayStation was enormously successful, selling over 85 million units, most likely due to its affordability and massive library of games available. The Playstation's motherboard featured several graphics chips to handle 3D graphics and texture mapping, and dedicated audio memory and sound chip. The 2x speed CD-ROM drive could also play audio CDs, in fact, there were some games where it was possible to pause the game, and stick in an audio CD to listen to—*Twisted Metal 4*, for instance, a fighting action game. The sound chip was capable of 24-channels of 16-bit digitized sound at CD-quality sampling rate, and allowed for real-time effects like ADSR changes, looping, reverb and pitch modulation. Like the Sega Saturn, the Playstation offered MIDI support for ease of programming the audio, but unlike the Saturn, samples did not have to be compressed, and the typical 4:1 compression rate that was used meant that sound quality could be greatly improved on the Saturn.

Advances in PC Technology

Although by the early 1990s most computers had FM sound cards supporting MIDI, many of these sound cards were cheap, and the FM synthesis made the MIDI music sound disappointing. When CD-ROMS came out, MIDI was pretty much abandoned, and with it the notion of interactive music. Popular early CD-Rom titles like *7th Guest* came out with high resolution graphics and higher quality music. The CD-ROM technology ensured that there was more room for music in games—previously most games had shipped on 3.5" floppy disks, and, perhaps more importantly to the game composers, since the audio was not reliant on a soundcard's synthesis, composers could know how the music would sound on most systems.

The downside of the CD-ROM technology was that most CDs could hold a maximum of 72 minutes of music, and with a game included this meant much less real time for music. As such, various compression technologies were developed, the most important of which was MPEG level 3, known more commonly as MP3. MP3 meant that much less data would be required to store audio, and game companies quickly began incorporating MPEG compression into their games music.

Along with the advances in hardware, several software solutions were produced which enhanced game sound. Beginning with *Windows 95*, the *Microsoft Windows* platform came with DirectX, a series of multimedia application programming interfaces, that improved the speed that sound and graphics cards could communicate. DirectX allows game programmers to access "specialized hardware features without having to write hardware-specific code".[33] In other words, the DirectX interface bridges software and hardware, allowing for higher-quality 3-D graphics, and better control of sound mixing and output. DirectX quickly became standard, and many games now support DirectX.

One part of DirectX, DirectMusic, was a particular advance for games music. DirectMusic overhauls the old MIDI protocols by offering the industry-ratified DLS (Downloadable Sound Levels 1) specifications support for hardware acceleration and MIDI: over one thousand channels, with better timing mechanisms, real-time control.[34] In other words, DirectMusic offers an upgrade to the interactive possibilities previously seen by iMuse and others. DirectMusic opens up MIDI to the possibility of wavetable-like synthesis and sampling, while allowing MIDI to be a triggering mechanism for timing.[35] Wave files (samples), or sound fonts, could be imported into a collection and manipulated in the same way that MIDI controllers manipulate synthesised sources, allowing for much improved interactive music.[36]

3D Audio

Three-dimensional (3D) sound gives the listener the perception that the sounds are emanating from a 360º (three-dimensional) space. The first soundcard to support 3D sound was Diamond Monster Sound, in 1997. Parameters such as room size and acoustic properties could be programmed into the audio, which would initiate filters and effects

---

33 http://www.microsoft.com/windows/directx/productinfo/overview/faq.asp

34 http://www.gamasutra.com/features/sound_and_music/19981106/directmusic_02.htm

35 ibid.

36 http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmusic/html/dm_nmp.asp

to simulate the space.[37] There are commonly several terms used for 3D audio which should in fact be differentiated, including spatialisation, virtualisation and 3D positional audio.

Spatialisation, also known as stereo expansion, uses signal processing to give a listener the effect that the sound is coming from a wider space than that of the real location of speakers. Spacialisation disperses perceived locations of sounds so that a listener cannot determine the location of the sound.

Virtualisation, on the other hand, tricks a listener into thinking there are more speakers present than actually are. Rather than locating a specific sound in a specific location (see below), virtualisation locates a specific channel (e.g. left rear) to a specific location. “However, to use this effect, multichannel audio must be available, and the sounds to be played back on the virtualized rear speakers must be encoded onto those tracks during production. This makes this solution less than ideal for the action portion of games, in which sounds might have to jump from the front speakers to the rear (depending on the player's actions), but cannot due to prior encoding on a specific channel. However, multichannel audio and the virtualization of these channels are very effective for noninteractive game intro scenes”. [38]

3D positional audio, also known as HRTF-based 3D audio (head-related transfer function), uses signal processing to locate a single sound in a specific location in three-dimensional space around the listener, and is common in video games, because it can be interactive. With 3D positional audio, sound objects in a virtual space can maintain their location or path of motion while the gamer moves about.[39]

Nintendo 64

cpu	MIPS R4300i @ 93.75MHz, 64-bit
ram	4MB, Cartridge (32MB), Expansion 4MB RAM
Graphics	32-bit Color
sound	SGI RCP 64 2D Voices, ADPCM

After splitting off from Sony, Nintendo bypassed the 32-bit machines altogether, going straight to a 64-bit release in 1996, the Nintendo 64 (N64). The N64 well surpassed the Playstation in capabilities. The main processor controlled the audio, producing 16-bit stereo sound at a slightly higher sample rate than CD quality—48 MHz. Some games supported surround sound, and this was enhanced with the third-party add-on release of RumbleFx 3D Sound Amplifier. It used ADPCM compression, with the possibility of using 13 simultaneous waveforms in realtime on the hardware, and even more using the cartridge. Filters and effects like chorus, panning and reverb could also be implemented in the internal CPU, or in the software.

The Nintendo 64 created a huge splash initially—selling 300 000 systems in the first 24 hours, but was only released with three games, and sales slowed dramatically after its initial storm.[40] The product’s lack of games availability in comparison with the Sony Playstation, along with the fact that it was cartridge-based in a gamers world then sold on the concept of CD-Rom, eventually meant that the N64 would not see the dramatic kind of sales achieved by the PlayStation, although it did reportedly sell 30 million units.

---

[37] www.tu-chemnitz.de/phil/hypertexte/ gamesound/pcsound-main.html

[38] http://www.gamasutra.com/features/sound_and_music/19980417/directsound3d_disaster_01.htm

[39]This entire section on 3D sound adapted from: http://www.gamasutra.com/features/sound_and_music/19980417/directsound3d_disaster_01.htm

[40] http://www.actsofgord.com/Proclamations/chapter01.html

SEGA DREAMCAST 1998	
CPU	128-bit Hitachi SH-4 RISC @ 200 MHz
Ram	26 MB (16 MB main, 8MB video, 2MB sound).
Graphics	NEC 128-bit Videologic Power VR2 @100 MHZ
Sound	Yamaha AICA. 2 processors: ARM7 32-bit RISC @ 45 MHz, digital signal processor. 64 channels 16-bit at 44.1kHz

The Sega Dreamcast was the first 128-bit console, and used special GD-Roms (GigaByte Disc Roms) capable of holding 1.28 GB. Unlike the Nintendo 64, samples did not have to be decompressed, improving audio capabilities. The dedicated audio processors had their own memory, meaning sound quality was not compromised by other aspects of the game. True 3D audio was supported, in CD-quality 64-channel sound, with effects such as delay, reverb and surround sound. The Dreamcast reportedly sold over six million units until it was discontinued in 2002.

PLAYSTATION 2 (2000)	
CPU	128-bit @295 MHz
	Scratch Pad RAM: 16 KB (Dual port) Main Memory: 32 MB (Direct RDRAM 2ch@800MHz)
Graphics	Graphics synthesiser chip @147 MHz. 3D CG Geometric Transformation 66 million polygons per second
Sound	"Sound Processing Unit" Dolby Digital DTS 5.1 Surround Sound 10-channel DMA, 48-channels of MIDI

The follow-up to the immensely popular Playstation ensured that fans of the original would be suitably impressed with the new machine. With the ability to play DVD movies, and the option of add-ons for modem and hard drive, the Play Station 2 was leagues ahead of competition when it came out. Its games were stored on DVDs capable of holding 5.7 Giga bytes, and fully supported the multi-channel DVD sound standards AC3, DTS and Dolby Digital, offering up to eight separate speaker channels. The sound Processing Unit is capable of 16-bit audio with a maximum sample rate of 48KHz—better than CD audio, and has 48 channels.

NINTENDO GAME CUBE (2001)	
CPU	128-bit 485 MHz MPU: Microprocessor: IBM Power PC "Gekko"
Graphics	Processor: 162 MHz
Memory	40 MB
Sound	ADPCM 64-channel 16-bit Macronix SP, 2MB Ram, Dolby, DTS, AC3 Support, 48 MHz sampling

Nintendo Game Cube Mini DVD discs hold about 1.3 Giga bytes, about the same as a Dreamcast disc, and has an audio capability comparable to that of the Playstation 2, above.

Microsoft XBox (2001)	
CPU	128-bit 733 MHz
Ram	64 MB

Graphics	Graphics Processor: 250 MHz nVidia 3D XGP custom processor
Sound	NVIDIA MCP Audio Processing Unit 256 simultaneous 2D voices (audio streams) 64 simultaneous 3D voices + 256 simultaneous 2D voices Five DSP units in total (3 fixed and 2 programmable) Global Processor – programmable DSP for reverb, chorus, flanger, equalization, 3D cross-talk cancellation, and occlusion/obstruction, Dolby Digital Interactive Content Encoder - programmable DSP for encoding into a Dolby Digital (AC-3) stream Downloadable Sounds Version 2 (DLS2) 5.1 surround

Microsoft's entry into the console business, the XBOX, was built around a Pentium III processor with an 8GB hard drive for music, graphics, and saved-game information. Games are supplied on 5.7GB DVD discs, and the XBox is also capable of playing DVD movies and audio CDs. The XBox features its own audio processor, supporting Direct X 8.0, 256 2D voices and 64 voices using 3D positional audio.

BIBLIOGRAPHY (see also footnotes for hypertext references).

Belinkie, Matthew. 1999. 'Video game music: not just kid stuff' http://www.vgmusic.com/vgpaper.shtml

Crawford, Chris. 1982. *The Art of Computer Game Design*: 'Genres' http://students.haverford.edu/jramos/genres.html

Demaria, Russell and Johnny L. Wilson. 2002. *High Score! The illustrated History of Electronic Games.* Berkeley California: McGraw Hill

Kent, Steven L. 2001. *The Ultimate History of Video Games: The story behind the craze that touched our lives and changed the world.* California: Random House

Krute, Stan. 1983. *Commodore 64 Graphics and Sound Programming*. Pennsylvania: TAB Books.

Leonard, Sean. 2001. 'Scores of Glory, Fantasy and Plumbing: The Concise History and Principal Genres of Video Game Music' http://www.seanspace.com/iSphere/scores.htm

Pidkameny, Eric. 2002. 'Levels of Sound' http://www.vgmusic.com/information/vgpaper2.html

Poole, Steven. 2000. *Trigger Happy: Video Games and the Entertainment Revolution*. New York: Arcade Publishing

Sellers, John. 2001. *Arcade Fever: The Fan's Guide to the Golden age of Video Games* Philadelphia: Running Press

Sheff, David. 1993. *Game Over: How Nintendo Zapped an American Industry, Captured your Dollars, and Enslaved your Children.* New York: Random House

Wolf, Mark J.P., Baer, Ralph (eds). Mark J.P. Wolf *"Genre and the Video Game"* The Medium of the Video Game. Univ of Texas Pr 2001

Glossary

Additive synthesis Additive synthesis (sometimes referred to as Fourier synthesis, as it is based on Fourier's theorem),[41] the opposite of subtractive synthesis, refers to the creation of a sound wave by adding together simpler sound waves (typically sine waves). Theoretically, additive synthesis was capable of producing any sound, but the sounds of instruments were still difficult to mimic, due to their complexity. Additive synthesis chips were popular in the Kawai K5 and K5000 series keyboards, though they never gained much general usage. Atari for a time built an additive synthesis chip known as AMY.

Bit A bit, derived from binary digit, is the smallest unit of information in computer language, a one (1) or zero (0). The number of bits typically indicates how much data a computer's main processor can manipulate simultaneously. It can also be used to describe sound fidelity or graphics. In sound, an 8-bit sound is comparable to AM radio, and 16-bit sound comparable to CD quality.

Byte a unit of computer data made up of eight bits.

Cutscene an intermission or break in a game, often used as a reward for solving a particularly difficult part of a game, and is generally non-interactive: the player sits back and watches.[42]

DAC Digital Analog Conversion - The method of converting digital data to analog data (as in analog sound to digital sound).

Frequency, the technical name for pitch, is a measure of the number of pulses in a given space of time. It is measured in Hertz, or CPS (cycles per second). For example, a note with a frequency of 440 Hz (A), means that in one second, 440 pulses occur.

Frequency Modulation (FM) synthesis[43] was designed by John Chowning at Stanford University in the early 1970s, and eventually licensed and improved upon by Yamaha, who would use the method for their computer sound chips, as well as the DX series of keyboards. FM uses a modulating (usually sine) wave signal (by selecting a modulating index) to change the pitch of another wave (known as the carrier). Each FM sound needs at least two signal generators (oscillators), one of which is a carrier wave and one of which is the modulating wave. Many FM systems used four or six oscillators for each sound (or voice): the Yamaha DX7 used six oscillators per voice. An oscillator could also be fed back upon itself, modulating its original sound.

Generator: An oscillator is capable of either making an independent tone by itself, or of being paired up cooperatively with its neighbour in a pairing known as a 'generator'. To create realistic musical sounds two sorts of generators are needed 1) oscillators to produce the basic waveform and 2) envelope generators to change the waveforms as the notes age.[44]

Interactive Audio, also known as Adaptive Audio: "In video games, music is said to be interactive when it can be directly altered by the player's actions.[45]

LA (Linear Arithmetic) Synthesis: The Roland MT-32 used a form of wavetable synthesis known as Linear Arithmetic, or LA synthesis. Essentially, what the human ear recognises most about any particular sound is the attack transient. LA based synthesisers used this idea to reduce the amount of space required by the sound by combining the attack transients of a sample with simple subtractive synthesis waveforms. This was known as cross modulation.

---

[41] Fourier's theorem states that all periodic waveforms are made up of one or more sine waves.

[42] http://www.vgmusic.com/information/vgpaper2.html#platform

[43] For a more detailed tutorial on FM Synthesis, see Computer Music's guide on http://www.computermusic.co.uk/tutorial/fm/fm1.asp

[44] www.umich.edu/~archive/apple2/misc/textfiles/music.faq.txt

[45] http://www.vgmusic.com/information/vgpaper2.html#platform

Modular: A modular synthesizer consists of modules which can be plugged into one another.

Oscillator: An oscillator is an electric signal which generates a repeating shape, or wave form. Sine waves are the most common form of oscillator.

Pink noise Pink noise is a variant of white noise. Pink noise is white noise that has been filtered to reduce the volume at each octave. This is done to compensate for the increase in the number of frequencies per octave. Each octave is reduced by 6dbs, resulting in a noise sound wave that has equal energy at every octave[46]. It is commonly used for rain sounds in games, sounding a bit like white noise with more bass.

Polynomial Counter, also known as a Linear-Feedback Shift Register (LFSR). LFSRs are type of binary counter that uses a pseudo-random way of counting, rather than the normal binary incremental/decremental sequences. The number was divided down from the system clock, meaning many pitches were not in tune with others, making it difficult to program melodies.

Portamento is the effect of sliding between notes when a voice is playing a melody

Programmable Sound Generator (PSG), silicon chips designed for audio applications which generated sound based on the user's specifications. These specifications were usually coded in assembly language, and early sound programmers and musicians needed to understand programming language to engage the chip.

Pulse Code Modulation (PCM) With PCM (otherwise known as raw, or AI2 synthesis), essentially, an analogue sound is converted into digital sound, by taking many samples—in other words, measuring the amplitude at regular intervals—of an analogue waveform. The data is stored in binary (1s and 0s), which is then decoded and played back as it was originally recorded. The fidelity of the sound depends upon the sample rate or quantisation—the number of bits representing the amplitude. The method is still used, for instance for DVDs or CDs, where the sample rate is 44,100 times per second (44.1 K), or 16-bit. Most early games could only sample at a maximum of 22,050 samples per second.[47] The down side of this method was the amount of space required to store the samples: as a result, most PCM samples were limited to those sounds with a short envelope, such as percussion. Another solution was what is known as adaptive difference PCM (ADPCM). With the ADPCM method, the difference between two adjacent sample values is quantified, reducing the pitch or raising the pitch slightly, to reduce the amount of data required.

Pulse Width Modulation (PWM) PWM modulation works by changing pulse waves by outputting pulses at a constant volume, while the width and spacing of the pulse gives the effect of different frequencies and volumes, controlled by the modulating signal's amplitude. The PWM method could attain higher volume, and achieve a range of interesting timbres (such as a pseudo-chorus or phasing sound), but the samples were low quality.

Subtractive synthesis starts with a wave form created by an oscillator, and uses a filter to attenuate or subtract specific frequencies and then passes this through an amplifier to control the envelope and amplitude of the final resulting sound.

Wavetable synthesis uses pre-set digital samples of instruments, usually combined with basic waveforms of analogue synths. It is therefore much more realistic sounding than FM synthesis, but is much more expensive as it requires the sound card to contain its own RAM or ROM. The Roland MT-32 used a form of wavetable synthesis known as Linear Arithmetic, or LA synthesis.

White noise White noise is a sound that contains every frequency within the range of human hearing (generally from 20 Hz to 20 kHz) in equal amounts. Used for laser sounds, wind, surf.[48]

---

[46] http://whatis.techtarget.com/definition/0,,sid9_gci213526,00.html

[47] See http://mushi.net/hanashi/gmintro.html.

[48] http://whatis.techtarget.com/definition/0,,sid9_gci213526,00.html

Technical assistance

Emulators

An emulator is a software program which allows an operating system to emulate—or simulate—another hardware system's chips. For instance, it allows Windows XP to run programs which sound, look and behave the same way as an Atari VCS, by way of ROMS. For a list of emulators available and information on how to obtain them, see http://www.faqs.org/faqs/emulators-faq/

ROMS

Roms are the name for the software programs that run on emulators. Technically, under the copyright act it is illegal to distribute roms. Although some websites will provide disclaimers or ask that you only download roms that you already own, roms are illegal "whether you have an authentic game or not, or whether you have possession of a … ROM for a limited amount of time, i.e. 24 hours".[49] However, without the availability of some of these roms (especially older arcade games) this project would not have been possible, or, at least, would have taken several years longer and been much more costly. Whether roms should be fall under fair use clauses for academic study is a matter of debate. I asked the IDSA for any suggestions for an alternative way to study video games but received no reply. I therefore used emulators and Roms where I could not obtain originals.

For this research, a combination of original consoles and emulators were used. Original consoles were used for the Atari VCS, Nintendo, Sega Megadrive, PCXT.

The following emulators were used:
Apple IIGS: Kegs 32 http://www.geocities.com/akilgard/kegs32/
Arcade: MAME (Multiple Arcade Machine Emulation).
Atari ST: WinSTon
Atari VCS: Stella, Z26
Colecovision: Virtual Colecovision
Commodore Amiga: WINUAE
Commodore 64: CCCS64
Dreamcast: Dreamer
Gameboy: REW
Game Gear: SegaEmu
NES: Ultra
Nintendo 64: Nemu64
Odyssey 2: 02EM
Playstation: PSEmu
Playstation 2: PSx2
Sega Megadrive: Gens
Super Nintendo: SNEeSe http://www.megagames.com/news/html/emulators/sneesev075.shtml

In addition, the following ressources are useful for extracting music from games, or playing games audio formats:
GAP the Game Audio Player extracts, or "rips" audio files from computer games.
Deliplayer_ plays game audio files of various formats
SIDplay emulates the SID chip in the Commodore 64, plays SID music
VDMSound (emulates DOS, adlib and soundblaster 16)

[49] http://www.nintendo.com/corp/faqs/legal.html