CHAPTER 5

Game Audio Today: Technology, Process, and Aesthetic

What goes into making a game? How do the game's music and sound effects get implemented? Who is responsible for ensuring that the dialogue does not conflict with the sound effects? Who makes the decisions regarding how the game should sound? The roles and processes involved in developing a game can vary greatly from game to game, from platform to platform, and from company to company, but a general sense of the process at a large company will help to answer these questions and to explain some of the audio decisions that must be made that impact the game as a whole. Audio production is not merely a series of compromises dictated by technological and industrial constraints. It is also a series of compromises with a team of people who work collaboratively, which has important implications for its production.

At its widest level, the game industry is comparable in structure to the book publishing, film, or music industries. Much the same as publishers in the book industry, game *publishers* are generally the overseers of the entire development process. Publishers provide some of the financial backing for the project, may provide royalty advances, and are largely concerned with producing a marketable project. Publishers, therefore, may make some important decisions regarding the design or development of a game to ensure that they have a viable product that will sell worldwide; they ultimately have creative control over the product. At present the top publishers include (among others) Electronic Arts, Nintendo, Activision, Sony, Ubisoft, THQ, and Microsoft, showing the clear importance of the console manufacturers on the development industry.[1] Publishers typically command the *developers*, the company/studio/team that designs and produces the game. Developers may be specialized toward producing for a specific platform or genre, or may develop for several markets. Developers are generally split into three divisions: third-party developers, in-house developers, and independents. Third-party developers usually work closely with one publisher in developing a project. In-house developers, also known as studios, are the subsidiaries of publishers. Rather than being contracted by the publisher, they are owned by and directed by the publisher. Independent developers, on the other hand, are not owned by or typically contracted by a publisher. They often rely on self-publishing, through the Internet, festivals, conferences, or word-of-mouth.

Within a development company, there are several important overseers to any project. The *producer* is generally in charge of the entire project's development process. These may be internal to the developer, or, in the case of third-party developers, external to the developer, hired by the publisher. The producer oversees all aspects of a game, including creative, legal, marketing, contracting, and so on. A *lead designer* then typically oversees the concept and design of a game, particularly the production of the design document (see below). The other major players include the lead programmer (overseeing the implementation and programming aspects of the game), the audio lead/audio director (see below), and the art director (overseeing the graphics aspects). Underneath these jobs are the support jobs, consisting of writers, programmers, level designers, sound designers, artists, and so on. The number of support jobs, of course, depends on the size of the company and the size of the game. Some jobs are contracted out to third parties, particularly in the case of audio (see box 5.1, ''Audio Roles'').

THE PROCESS OF TAKING A GAME TO MARKET

As with other aspects described below, the process described here represents an overview of one production model. Production varies greatly from company to company, from platform to platform, and from genre to genre (for more detail on the entire process, see Kerr 2006 and Manninen et al. 2006). Although mock-ups or initial prototypes may be created in the early stages of a game's development, the main pre-production phase involves the creation of a design document. The game design document represents the overall vision of the game, detailing the storyline, dialogue, maps, audio, graphics, animation, and programming. The entire team—including the audio team—works from this document during the production process, although each team may develop a more detailed document for their specific task based on the design document (see below), which describes the game in detail.[2]

Once the publisher has approved the design document, a team will be put together for the production phase of the development, which involves the creation of all of the different elements of the game (graphics, sound, cinematics)

Box 5.1
Roles of the Audio Team

The audio team of a games development company can vary quite significantly, depending on size and budget. Many smaller companies still have one audio person to complete all of the sound effects design, music, voice work, and implementation, while larger companies can have full teams of composers, sound designers, and voice actors all working on one project. Most development companies have in-house sound teams, but music jobs in particular are being increasingly contracted out. Although I have separated the main roles here, it is possible that these roles may be filled by the same person, or even be teams of people.

At the head of the audio team is typically the *sound director*. The sound director is responsible for the overall audio vision and design of a title. He or she oversees the design, defines and drives the creative and technical direction. Sound directors must coordinate schedules, budgets, staff, and technology, and manage outsourced asset creation personnel (casting, dialogue directors, mixers, engineers, and so on). They are responsible for organizing external contract work, as well as creating, editing, and mixing original content. They may also be responsible for the final mix.

Sound designers may also play this same role, creating an audio design document and managing the audio production pipeline. They will work with integrators and audio tool developers to create, integrate, and manage audio assets, and are responsible for sound effects libraries. This effects library may be purchased externally, contracted out, or developed in-house.

Dialogue/voice-over artists are, of course, responsible for providing the dialogue. They may be in-house, but it is more likely that these artists are cast externally for projects and managed by the audio director. The *dialogue director* oversees the dialogue process, often coaching the dialogue artists.

Licensing/contracting directors are responsible for obtaining rights to licensed IP, and contracting out work externally.

Composers are responsible for the music composition of the game. In smaller companies, they are frequently also responsible for the sound design. They may also be in charge of orchestration of their work, although on larger projects there may be teams of orchestrators working together. They are typically responsible for contracting out and overseeing live recordings.

Audio programmers or audio engineers are responsible for audio tools development, and integration of all audio assets in a game. In many cases they also play the role of sound designer. They will be responsible for developing in-house audio tools to work in conjunction with the game's AI, graphics, and physics engines.

and their integration into the game engine. At various stages in the development process, the game will undergo several quality assurance phases, which will test and validate the gameplay, user interface, and market needs. Once the game has been developed, it will undergo a debugging process to check for any problems (''bugs'') in the programming or playback of the game. Once it passes the debugging, the game's documentation and manuals are produced, localization takes place (see below), and the game may be ported to the various platforms for intended release (for instance, a game developed for PlayStation 3 may then be programmed to work on Xbox360). When the game is released, attention turns to marketing, but there is still necessary upkeep or maintenance that may include releasing patches, or upgrades, to fix uncaught bugs and potentially to provide the user with new content.

Within this entire process, the audio team may become involved at any stage in the development of the game. Most often, the sound team does not join the process until very late in the project, when the game and its parameters have already been defined. At this stage, the audio team simply populates the game with sound (Selfon 2006). Other audio teams work more closely with the design and development of the game, to ensure that audio can play a significant role in a game's development. As with film, the ideal, suggested by famed film sound designers Walter Murch and Randy Thom, is that sound should be considered at the earliest (that is, the script/design) stage (Geuens 2000, pp. 197–198).

THE AUDIO PRODUCTION PROCESS

In some ways, the game audio production process resembles that of the film audio process (for discussions of the film sound production process, see Davis 1999, Rona 2000, or Kompanek 2004). There are similar recording techniques for live sounds and *foley*,[3] similar techniques for spotting, and many of the same tools in terms of recording and software are used. Comparing the music process of film and games, *LAIR* (Factor 5, 2007) composer John Debney elaborates,

> The process is similar. There are definitely scenes that one has to compose specific music for. A lot of the game play, i.e. the battles or the big set pieces, essentially has to be scored in some form or fashion. So that's all similar to a film. . . . Aesthetically the biggest difference for me in scoring a video game is that you don't have as much finished product. Much of the time I would be writing to a description of a battle . . . literally just a one or two line description. I would also be writing to maybe twenty seconds of game play that in reality is going to become ten to twenty minutes of game play. That was the biggest difference for me. It was more about writing to a concept or description rather than writing to anything specific. (Cited in ScoreKeeper 2007)

However, there are also significant differences in the processes. Apart from dialogue and some production sound, film audio is generally a post-production activity that takes place after the film has been edited and the visuals *locked* (the final version set). A significant amount of time is spent balancing and mixing sounds in a film's post-production, which is a great distinguishing trait between film and game sound. In games, since (with the exception of the cinematics) timings are variable and the visual sequence is constantly evolving, ''post-production'' as the practice exists in film does not generally exist in game audio production, although there are a few notable exceptions, discussed below.

There are also other differences in the processes, and so it is worth spending some time describing the game audio development process in some detail. However, it should be noted that the production process in video games may differ significantly from what is presented here. Different genres have different types of recording needs: a simple puzzle game is not going to require dialogue, for instance. Moreover, different companies have different budgets and can spend more or less money on team size. Sony's *God of War 2* (SCEA, 2007) music team, for instance, consisted of four composers, three orchestrators, three ensembles (brass, string, and choir), a variety of ethnic soloists, and the development/implementation team (Bajakian 2007). Smaller companies may have one or two people who must perform the equivalent of all of these jobs. The platform can also affect the production process: PlayStation 3 games are going to require many months or even years of work, whereas for a mobile game, audio is typically given about a week for the entire process. I have, therefore, described a semilarge production, with separate composer(s), sound designer(s), orchestrator(s), voice actors, and programmer(s). I have broken the production phase into the three main components (music, sound effects, and dialogue) with a more specific discussion of pre-production, production, and post-production in each category. Further details on the production processes themselves can be found in Alexander Brandon's book, *Audio for Games: Planning, Process, and Production* (2005).

THE PRE-PRODUCTION STAGE

The first stage in the audio workflow is the creation of an audio design document. Supplementary to the game design document, an audio design document details the design and implementation of the game's audio. There may be separate music, dialogue, and sound design documents, but I have incorporated these here. An audio design document is also not necessarily a part of every game, and even when the intentions are there, a game's development does not always follow the plan. Damian Kastbauer, a sound designer working for a major developer, commented that ‘‘while there is sometimes an overarching concept of how things should sound for a given project, such audio design documents are often a myth and the process becomes more about doing what is right for the game [at the time]’’ (pers. corr., March 29, 2007). An audio design document is designed to assist the audio team, as well as the programmers who will need the document to implement the sound into the game. Says Keith Zizza, Audio Director for Impressions Games, ‘‘Designers will want to absorb it, programmers will demand it, and producers, along with just about anyone else who is involved on the project, will want to at least skim it. Whether it’s one page or one hundred, it should be as descriptive as it needs to be for you and your development team. The end result, hopefully, is a harmonious one—working with and enhancing graphics, writing, game design, and the overall gaming experience’’ (Zizza 2000).

An audio team who joins the production process early on may only have storyboards, concept art, crude gameplay, or character sketches from which to develop a design document. Nevertheless, there are many decisions that can be made at this stage, in order that work can begin on the sound early on in the process, and to ensure that audio plays a significant role in the game. First, it is necessary to determine the general game type, in terms of the theme and genre, to determine the style of sound design needed, and the type of music that would be appropriate. One common way of approaching questions of style in music is to create a *temp track*.[4] A temp track places preexisting music temporarily in place of the final composition, defining basic parameters from which the composer can work. Different composers, of course, have different approaches to their musical scores. Koji Kondo (2007), composer of the *Super Mario* and *Zelda* series, for instance, states that he likes to view the entire game as one composition, with each song within the game as part of a larger complete work. In dealing with franchise games or episodic content, it is also important to consider how the sound will relate to previous games, associated films, and so on.

After determining style and mood, the second most pressing issue is to deal with the functionality, or the game-specific behavior of audio—in other words, how the sound interacts with the gameplay. Will music be merely incidental, or will it be a dynamic score? What role will sound design play in the interface? Here the rules for interactivity are defined, since the game design dictates the sound design. *Spotting* is the next major element in the audio’s development. At this stage, it is determined which parts of the game should have ambient sound and music. Specifically, this involves defining cue point entrances, exits, play-ins/play-outs, and game state changes, as well as deciding if game variables (such as player health, surface properties, and so on) will be used to change sound parameters. Kondo (2007) suggests keeping in mind the rhythm of gameplay in determining the structure of sound. For instance, in *Super Mario Bros.*, the rhythm of the hi-hat was used to emphasize the internal rhythm of the game,

Box 5.2
The Music Cue Sheet

Music cue sheets help the composer to organize and design the soundtrack, as well help the programmer to know how to implement the files into the game. Many composers have different approaches to cue sheets: some organize by emotion, some by instrumentation, for instance. Organizing where the game needs music is a great first step to arranging the time needed, orchestration needs, recording sessions, and other important aspects of creating the game's score. The cue sheet below shows a section from Jeff Simmons's music cue sheet from the 3D MMORPG *Earth Eternal* by Iron Realms (2007).

Table B5.2
Sample music cue sheet

Area Cues				
File no.	File name	Action	Time	Notes
1	dungeon_01	Nonlooped	2:07	Slightly dramatic or dark mood
2	Caves_01	Looped	1:37	Scarier, more foreboding than dungeon but less somber than King's Grave
3	Desert_01	Nonlooped	1:57	Egyptian/vaguely Arabic
4	Mystical_01	Nonlooped	1:49	Places meant to be very mystical
5	Gothic_01	Nonlooped	2:00	Darkness filled with metallic elements

and melody was used to bring that rhythm to life. It is also important to consider the controls and how these should influence that sound aesthetic: if buttons are rapidly pushed and the gameplay chaotic, the sound should reflect this physical activity of the player.

For the music, a cue list can be created with preplanned and "to-be-determined" cues (see box 5.2, "Music Cue Sheet"), breaking the script into acts or chapters or segments, defining what is happening dramatically, and creating an *emotion map* for the game, as well as for each individual level (see figure 5.1). Does the level end in a boss? Are there mini-bosses? Do we find a key item? Where are the significant points of tension and release?

As composer Charles Deenen has described, there are six basic audio emotions: happiness, sadness, surprise, disgust, anger, and fear, and each of these can be mapped to major scripted events.[5] Composer Scott B. Morton (2005) writes,

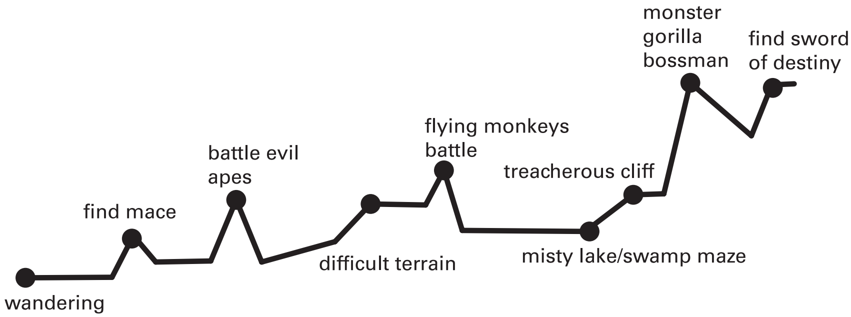

Figure 5.1
Emotion map for game's level tension and release patterns.

> This musical arc is often more important than the literal events themselves because it can infer deeper meanings ... more than simple actions on the screen can.... Composers should start thinking beyond ''What does this level sound like'' to ''What role does this level and its characters play in the grand scheme of the game and the plot? How do I portray that with the music I write? Where do I place the music within the level to bring this across in the most effective manner?'' ... Create a musical climax in your game. Don't use your most intense music until you've reached critical points in the game's dramatic arc. Is the final boss battle more important than the miniboss battle? Show it in the music. A player should be able to subconsciously interpret the importance level of events based on the music that accompanies them.

In terms of sound design, style will also be explored during this preproduction phase, taking into consideration genre, sample rate, resolution, prioritization, and other sound considerations. As with music, spotting the game and creating a list of assets needed is the first step in the sound design of a game (see box 5.3). This involves a reading of the script or design document, looking to spot for objects, actions, environments, pace, tension/release, characters/personalities, and so on. Spotting the game in terms of the emotional needs of sound design is also important: generally, there is a rhythm or movement within a level or game in terms of emotional peaks and valleys. As with music, an emotion map of tension and release points can help the sound designer in deciding which sounds need emphasis, which sounds may interfere with one another, and which sounds are secondary in importance. Asset lists are often based on descriptions of levels, and are broken down into sound types (weapon sounds, ambient sounds, menu sounds, foley, and so on). The asset list can then be used to track recording in order to be able to reproduce sounds at a later stage, if necessary. In addition to sound effects and foley, ambient sounds are a key part of a game's overall feel. This may include music, or ambient dialogue, or it may include outdoor environmental sounds. Creating a mood (of safety, of excitement, and so on) can be used

Box 5.3
Dialogue Cue Sheet

Since dialogue lines have increased to several thousand per game, it is becoming increasingly important to track dialogue sessions. The dialogue cue sheet can track as-recorded files, any DSP effects on the voice, format, file size, and so on.

File name	Character	Location	Dialogue	As recd.	DSP FX	Notes
Sh001	Sharon	Level 3 cave	"Help!"		Reverb cavex03	
Kev001	Kevin	Level 3 cave	"Hang on! I'm coming!"	Hang on!	Reverb cavex03	
Mons001	Monster1	Level 3 cave (just outside)	"Grrrauh!"			Still to add DSP
Kev002	Kevin	Level 3 cave	"What was that?"		Reverb cavex03	

to prepare a player for a particular situation, or to trick the player into thinking an area may be safe when it is not (Kutay 2006). Says Steve Kutay (2006), sound designer at Radius360:

> The psychological impact of ambient sounds can add much to the onscreen imagery, though not physically present in the scenery. For instance a distant, sustained cry of an infant suggests vulnerability or insecurity. A broken fence rattling in the wind of an abandoned city, suggests to the player a previous traumatic event. These are subtle examples used to arouse awareness in the player. More obvious sounds should be used to cue the player of his direct proximity to danger. Dark drones or muffled enemy vocalizations will prepare the player for fierce combat ahead. Fear, anticipation and anxiety are easily evoked by the careful placement of ambient sounds.

It is important, however, that ambient sounds not rely on looping. The player will quickly pick up on any distinctive sound that repeats. One solution that was incorporated into the soundtrack for *Halo* (Bungie, 2001) was the idea of permutation weighting, as described by Marty O'Donnell and Jay Weinland (in Fay, Selfon, and Fay 2004, p. 423):

> We had multiple tracks playing in many cases (such as an outdoor ambience plus a wind track), each with multiple permutations in the ''main'' loop tag, as well as detail sounds that could be randomly triggered and were placed randomly in 3D space. One technique to highlight (which was used extensively in the music tags as well) is permutation weighting. Permutation weighting is the ability to assign different probabilities to each permutation to control how often a given permutation is played. For example, there are six permutations of a ''main'' loop in an outdoor ambience with lengths varying from two to eight seconds and 27 seconds of material. In one of those loops, there is a distinctive owl hoot, which would be repeated every 27 seconds if the material was played back in the same order each time.... Given the randomness of our permutation playback, you actually might hear it on average every 27 seconds but sometimes a little less and sometimes a little more frequently. If that distinctive hoot is still heard too frequently, we can use permutation weighting to assign a high skip fraction to that one permutation that tells the audio engine to skip this permutation x-percentage of the time. We adjust this number until we hear the distinctive hoot at a frequency that seems natural.

With these music and sound design decisions made, a list of required sound assets can be assembled, with separate asset lists for sound design (including, for example, weapon sounds, characters, user interface sounds), music (different modes/levels), dialogue, and additional audio content (for marketing/promotions, cinematics, etc.) (Brandon 2005, p. 21; see box 5. 3, ''Sound Design Asset List''). As Zizza (2000) describes, ''Reference a separate list of all audio content for the game, including those for demos, marketing, your web site, and so on. If you like having everything in one document, and the content list is on the small side, it's probably fine to include the master list here. In any case, it's good idea to include a general outline of content, well before there is enough detail to have an actual, formal list.''

As this stage, it is necessary to determine the technical limitations of the systems being used, including, for instance, how many channels of sound will be used, whether the delivery will be in surround sound format, and what production values the sound will have. Finally, implementation must be considered, including the tools and technology available and required (to be built or bought), such as platform, sound engine, and playback engine. Composer Richard Jacques described that it may be useful to think as the programmer, in creating simple if–then statements, using examples from *Sonic 3D* (Sega, 1996), which used Redbook audio tracks cued dynamically:

Example 1, Level 1, Act 1: Play Green Grove Act 1 music
IF Sonic picks up speed shoes
THEN play ''speed shoes music.''
IF Sonic picks up invincibility icon
THEN play ''invincibility music''

IF Sonic has collected 100 rings
THEN play ‘‘100 rings jingle’’[6]

In this way, the programmer can see exactly how music or other audio elements are to be implemented, and the composer or sound designer can get a feel for what will work in the game. Finally, a schedule and budget typically follows, including time for design, asset creation and production (including recording sessions, and so on), re-dos, implementation, and testing/quality assessment.

THE PRODUCTION STAGE

If the game was not spotted for music cues during the pre-production phase, the spotting of cues will take place from a solid, playable form of the game, if one exists. If a playable does not exist at this stage, composers may create a series of *scratch tracks*, or rough drafts of cues. Composers typically have their own way of creating music at this stage, and the music's creation also depends on the size of the company: smaller companies are unlikely to use orchestras, and so synthesizer tracks can be created, which are more affordable and more easily reworked when changes need to be made. In a large company, once scratch tracks have been approved, temporary melodic assets, or multitrack splits will be delivered to orchestrators, who will orchestrate the songs. Charts are taken to ensembles and pre-records (of synthesized versions and soloists) are delivered to the orchestra. Recording sessions are undertaken and then mixed, mastered, and *sweetened* (altered with layering, effects, and so on).

The production of game sound can typically take place in a number of venues. In the recording studio, stock sound CDs or sound effects libraries are commonly used. Oftentimes these stock library sounds are manipulated and layered to achieve a desired effect. In the studio it is also possible to record custom sounds with the use of various props. Some companies have dedicated foley studios or foley pits, which are designed for in-studio recording, such as footsteps or clothing noise. In addition to in-studio recording, many companies incorporate some field recording—that is, recording outdoors or in various locations other than the studio. Once all of the sound assets have been gathered, typically sounds are manipulated in the studio, treated with various effects, or sweetened to create a more exciting sound and to distinguish the sound from other games that may be using the same effects libraries. This usually involves layering sounds, adjusting various elements of the sound, equalizing and compressing the sound, or using various digital signal processing (DSP) effects (see box 3.1).

One of the most important advances in sound technology on next-generation consoles is more powerful DSP, including the processing of DSP in real time (that is, while the game is actually being played). Being able to add DSP effects onto sounds in real time in the game saves a lot of recording time. For instance, previously, to get the effect of footsteps to change when walking from, say, a stone path into a cave, the effects would have to be pre-recorded onto the footsteps sound file. Each different location would have to have a pre-recorded sample of the sound. Now, DSP filters can be set for locations, and so selected sounds can be processed in real time in that location. Only one set of recorded footsteps is needed to create potentially unlimited sound effects from that one original recording. In other words, with real-time DSP, audio can respond to physics graphics engines to create more realistic-sounding effects in the game during real time, as Sotaro Tojima, sound director for *Metal Gear Solid 4* (Kojima, 2007) describes:

> For example, in the scenario where a bottle falls off a table, hits a metal shovel, and then rolls onto a carpet, conventional sound processing would have the bottle make the same sounds regardless of the environment, or what it collides with. That same scenario on the PlayStation 3 might have the bottle make a metallic tink when it hits the shovel, and then create a muffled rolling sound as it travels across the carpet. If the room had its own sound variables, the bottle's sound might get take on some echo if in a bathroom, or get slightly quieter if in a bedroom. Then you have to factor in on-the-fly surround encoding, which would make the bottle pan from front to back or side to side in your room, depending upon the way it rolled. (Cited in Shah 2006)

The audio team also typically oversees the production of dialogue (voice-over), since this must be mixed with the other audio in the game. In the early days of game dialogue, friends of the audio developer were often drafted to record voice parts, but today professional voice actors are commonly used and the process is much more in-depth. Dialogue can be an important part of a game's overall sound, as Marty O'Donnell and Jay Weinland elaborate in a discussion of *Halo*:

> The dialog in *Halo* was one of the areas that helped to give *Halo* a unique flavor. There are two types of dialog: cinematic dialog that is a traditional linear script and dynamic dialog. As you play through *Halo*, your Marines will dynamically alert you to what is going on around you, warn each other and you of danger, chastise, apologize for allies getting shot or killed by friendly fire, and generally amuse you. This aspect of *Halo's* audio could not have succeeded without the hard work of AI (artificial intelligence) programmer Chris Butcher who gave us many possible AI states in which a character might say something and controlled how much was said to make it all seem natural. . . . The cinematic dialog also works well due to both a great storyline by the Bungie team and excellent writing by Joseph Staten. Scripts for the cut scenes and other interior scenes during gameplay were written and storyboarded so that we could clearly deliver the plot to the player. There is about one hour and 15 minutes of cut scenes in *Halo*, and that puts this part of *Halo's* production on par with a short feature film. (Cited in Fay, Selfon, and Fay 2004, p. 428)

Dialogue events can be separated into several distinct types: ambient dialogue (also known as *walla*, background ambience for environments where there are background people, such as crowds at a baseball stadium, shop, or market); scripted events (which are in-game scenes in which the player can typically walk away, but if he does so, he or she will miss some important information); cinematics; AI cues (also known as *barks*, which are nonverbal lines such as screaming); voice-over narration; and the in-game lines (see Chandler 2005). Together, these dialogue events can represent thousands of lines of script. *Halo 2* (Bungie, 2004), for instance, had more than sixteen thousand lines of dialogue.

Much as it is necessary to organize assets for sound or music, dialogue assets must be carefully cataloged. A script can be broken down into a dialogue sheet by character or scene (see box 5.4, ''Dialogue Cue Sheet''; see also Chandler 2005). Casting must take place for each localized area (if the dialogue is being localized; at times this is not possible and subtitles are used). Generally, studios will cast one actor for several roles. It is therefore important that a cue sheet is carefully organized, so that actors do not end up in the same scene, ''speaking to themselves'' in a game. Studios will record by character, rather than sequential order, so recording sheets for each separate actor must be produced. Details regarding the As-Recorded Script (ARS), which are changes made to the script when recorded, must be described. Dialogue is often recorded in an ADR studio,[7] which is designed for film over-dubbing, but is very suitable for games dialogue work. The *dry files* (raw, unaltered) are delivered and checked for quality. These files are then cut into single asset files for integration and treated to various effects or editing as necessary. Once audio editing is complete, audio is set to video clips of the game, particularly in cinematic sequences. Additional recording sessions and retakes are often part of this process.

An increasingly complex component of game audio is localization. Localization is the process by which a game is adapted for a specific territory or target market. Typically this includes the major European languages (French, German, Spanish, and Italian), and may include Asian markets as well. Localized versions are either completed at the same time as the original version (known as *sim-ship*, or simultaneous shipping, for worldwide release), or *post-gold*, which is a remake of the game with localized assets after the original version is complete. Ensuring a wide market in terms of distribution helps to minimize some of the financial risks involved in producing a game. To some extent, localization is genre-specific, as the Game Localization Network (2005), a company specializing in localization elaborates: ''Shipping a game with a translated manual (a 'doco' version) might be considered sufficient for a [*sic*] arcade racer but it will not engage and assist players of a story driven RPG. An RPG that is targeting a large language market might benefit from a 'full' localization (user interface, in-game text, spoken audio, manual and support documents), a similar game targeting a smaller market might be better shipping as a 'sub-titled' version (user interface, in-game text, manual,

Box 5.4
The Sound Design Asset List

The sound design asset list tracks the assets needed for the game, and can be adapted to also function as an audio report, which can detail where the sound came from, any DSP effects it was treated with, and so on. These asset lists can be as simple as the one presented below, or incredibly detailed including take number, audio channel information, equipment setup notes, and information regarding equipment used to record, along with SMPTE time codes, track numbers, file names, or other identifying information. The cue sheet below shows a section from the sound effects asset list from Jeff Simmons's cue sheet from the 3D MMORPG *Earth Eternal* by Iron Realms (2007). Used by kind permission of Jeff Simmons and Iron Realms.

Hits and Impacts		
Swords		
File no.	File Name	Time
50	sword_flesh_impact01	0:01
51	sword_flesh_impact02	0:01
52	sword_flesh_impact03	0:01
53	sword_leather_impact01	0:01
54	sword_leather_impact02	0:01
55	sword_leather_impact03	0:01
56	sword_wood_impact01	0:01
57	sword_wood_impact02	0:01
58	sword_wood_impact03	0:01
Staffs		
File no.	File Name	Time
59	staff_flesh_impact01	0:01
60	staff_flesh_impact02	0:01
61	staff_flesh_impact03	0:01
62	staff_leather_impact01	0:01
63	staff_leather_impact02	0:01
64	staff_leather_impact03	0:01
65	staff_wood_impact01	0:01
66	staff_wood_impact02	0:01
67	staff_wood_impact03	0:01

and support documents), thus contributing to the bottom line by saving the cost of audio recording.''

Localization is necessary to create a realistic game environment, no matter what the player's language, but it is more than just straight translation: the user interface must be intuitive to that market, the story must be as polished in the translation, and the audio must be as realistic and engaging as the original (see Game Localization Network 2005). Localization may include changes in controller layouts, animation sequences, or frame-rates that may require adjustments in audio. What's more important, there may be significant cultural differences in the various territories that require changes in gameplay. Different music may need to be selected for different target markets.[8] Legal or cultural issues can arise from the different ratings systems in various territories, owing to censors and public opinion. For instance, Germany has very strict laws regarding violence, which for audio may mean toning down the sound effects to reduce the impact of violent sequences. When the narrative of the game is significantly revised for another culture, this is known as *blending* (Thayer and Kolko 2004, p. 12). Blending can involve writing new narratives to target a specific culture, and therefore might also require new graphics and new sound. Examples of blended games include, for instance, *Crash Bandicoot* (Naughty Dog, 1996), in which the main character underwent an appearance change for the Japanese market, and *Half-Life* (Valve, 1998), which had to replace human characters with robots to satisfy different rating standards in violence (Thayer and Kolko 2004, p. 17). Obviously, changing a character from human to robot involves significant changes in audio.

The final stage in the audio production process is the integration of the music, sound effects, and dialogue. Integration of audio into the game involves much more than writing a few lines of code. Describes Marty O'Donnell, ''The implementation side of it is really huge ... it's not just about, 'We need a sound.' It's all about 'Here is this vehicle or here is this weapon, which has many different components and many different ways it needs to act in a 3-D audio environment' '' (cited in Hanson 2007, p. 49). O'Donnell even goes so far as to suggest that implementation is responsible for at least 50 percent of the final audio result. It is, after all, the ways in which audio is integrated into a game that will have an impact on the effectiveness of that audio. Integration typically determines how audio will be cued or triggered in a game, as well as what aspects of the audio may be changed by the game state or game parameters (see Whitmore 2003). For instance, music or ambient tracks may be triggered by location, by game state (such as player health, or enemy health), by time-ins or time-outs, by players, or by various game events.

The music files the composer writes must be able to be integrated into the game in a variety of ways. Integrating the music, for instance, may involve cutting the music into *splits*, *loops*, or *chunks* in order to create a more dynamic score. As John Debney, composer for *LAIR* described (in ScoreKeeper 2007), ''There was almost two hours of music which in the scheme of games is not a lot. They'll take that two hours and cut it up and turn it into hours and hours of music. . . . the mechanics of writing music that they could cut really easily was important. A lot of the action oriented pieces are, by design, pretty rhythmic so that they could cut it very easily.''

In addition to the actual cues being cut for integration, various musical elements may be altered in real time in the game engine, such as DSP effects, tempo, or instrumentation. Middleware tools, such as *ISACT*, *Wwise*, *FMOD*, and others are increasingly being used to decrease production time and costs, and to integrate a more dynamic score into the game. Canadian company Audiokinetic's middleware solution *Wwise*, for instance, allows audio developers to prototype the integration of audio in a game simulation before the game is even finished. Environmental effects can be rendered in real time, and occlusion and obstruction effects can be managed within the software, mixing up to four simultaneous environments per object. Sound prioritization for real-time mixing is also included, as is randomization of various elements for effects such as pitch or volume, to enhance realism. Real-time game parameter controls can also be set, to adjust sound properties based on parameter value changes in a game. So, for instance, if a player's health is running out, music could be sped up, or could increase in volume. Also important, the software allows the user to validate audio profiles for each platform, to manage the performance in terms of CPU and memory usage, in order to adapt and adjust performance before the final integration.

An equally important part of production is the placement of sounds in the three-dimensional space. Use of high-definition (HD) format in television and the adoption of 5.1 have led to a growing consumer demand for surround sound in games. All next-gen consoles are compatible with Dolby Digital's discreet surround 5.1 (see box 4.1). With capabilities of 7.1 in some games systems, surround sound is rapidly becoming the norm, and indeed is expected, particularly as consumers upgrade their home-theater equipment. Spatial positioning of sound is increasingly becoming an integral part of gameplay. Surround sound is used to help create a more realistic and immersive environment in games, as Richard Dekkard of HammerJaw Audio describes in the use of surround in *Auto Assault* (Net Devil, 2006):

> Anything that moves in the game is panned dynamically in the surround field. This includes all vehicles, enemies, explosions, etcetera. There are also stationary sound-emitting objects that are panned dynamically. There are many ''George Orwellian'' propaganda towers all around that you will hear panned interactively. In addition to these interactive elements, we have six levels of ''prebaked'' surround sounds. These include weapon sounds, music, weather, environmental audio, interface sounds, and your own vehicle explosion is in surround sound. We have taken it to every extreme we could. (Dolby.com n.d.)

Surround can be useful for more than just its immersive quality, however; it can be used as an audio cue. A basic example is the use of surround in the gameplay of the Shadow Mario character in *Super Mario Sunshine* for the Nintendo Game Cube (Nintendo, 2002), in which the positioning of audio indicates to the player where the translucent Shadow Mario would pop up. Similar use of surround sound to position enemies was seen in the stealth and first-person shooter games discussed in chapter 4. Another interesting use of surround sound can be heard in *Onimusha 3: Demon Siege* (Capcom, 2004), in which the player is given the option of adjusting his or her ''listening position,'' setting the perspective of the audio as either that of the player or camera. Surround sound, therefore, can play a variety of important roles in games, as Marty O'Donnell and Jay Weinland elaborate for *Halo*:

> Knowing that a Marine ''had your back'' just over your right shoulder brought a sense of security, just as hearing a Hunter's metal armor behind you would bring a sense of impending doom. There were many audio elements in the game that received 3D positioning, such as weapon sound effects, bullet impacts, speech, mechanical objects, particles such as flying dirt or sparks, and outdoor detail ambiences, such as wildlife, rivers, and waterfalls, to name a few. In essence, everything you hear in *Halo* that is not music or an ambient bed is 3D positioned . . . All 3D audio is subject to occlusion, obstruction, Doppler, and HRTF. Anytime a solid piece of geometry gets between the player and a sound source, it is occluded and obstructed. A good example of this is in the hangar bay of the alien ship where there is a dropship with the engine running. As the Master Chief steps behind a column, the sound of the engine is occluded, rolling off both the gain and the high end of the sound . . . HRTF (Head-Related Transfer Function) also adds to the audio experience by filtering sounds, depending on which direction the character's head is facing. You can hear it affect the dialog in the cinematics as well as in the sounds that play during combat. Probably the best way to hear this effect, however, is listening to your own footsteps. As you move the Master Chief around in *Halo*, listen to the sound of his footsteps on various surfaces. Then run through the same areas looking down toward his feet rather than straight ahead and listen to those same footsteps; the difference is stunning. (Cited in Fay, Selfon, and Fay 2004, pp. 424–425)

Sounds are mastered individually. Mastering of separate sound elements (dialogue, effects, etc.) helps to adjust the sounds' dynamic ranges. At present, there is still too little dynamic range within games (see Bridgett 2008). With real-time mastering, unwanted audio information can be removed in favor of a much more realistic, cinematic dynamic range. As Alexander Brandon elaborates, ''another delusion held by all audio folk who gleefully cling to their BBE Sonic Maximizers is that all sound needs not be normalized to überhigh levels, in some cases creating a WAV file that in *Sound Forge* looks like a solid brick . . . proper balance of dynamic and frequency range is vital to avoid butchering the ears of the average listener'' (Brandon 2005, p. 46). Bridgett (2008, p. 129) argues that one reason for this loss of dynamic range was perhaps the fact that ''video games were born in arcades where they had to compete with the sounds of other nearby games consoles'' and as such, as noted earlier, sounds had to be loud to attract attention. He goes on to argue that the cinematization of games is changing their ''cultural positioning'' and thus leading to an aesthetic that is closer to film or classical music, in which sounds demand a certain ''listening etiquette,'' in which they are ''expected to be listened to in isolation and given the audience's undivided attention, they are not expected to compete simultaneously with other [environmental] sounds.''

THE POST-PRODUCTION STAGE

''Post-production'' in games typically involves some degree of mixing. Mixing adjusts the interplay of all audio components in a game to ensure that there is no overlap between frequencies, including deciding which elements should be emphasized and which should be deemphasized in the mix. The mixer must listen for what Charles Deenen calls *believability gaps* in the audio: awkward silences, too much repetition, an unnatural imbalance (in EQ, dynamic range, and so on), or unrealistic dialogue.[9] Integration may also involve dynamically mixing a score, such as in *Grim Fandango* (LucasArts, 1998), which adds or drops instruments at certain key moments or environments. Mixing is, therefore, ideally something that should be considered throughout the whole audio design process, although this differs from developer to developer. As Ed Lima, audio director at Gearbox Software, elaborates: ''I think about the mix throughout the entire design process. I generally try to bake some slight equalization curves or tendencies into families of sounds. For instance, explosions might be bottom-heavy, voice-over might occupy a higher band or the music might be designed with specific instruments and frequency bands in mind'' (in Henein 2007, p. 56).

At present, sounds in games are, in a sense, competing with each other—dialogue, sound effects, ambience, and music all inhabit the same aural space. Particularly since dialogue and the sound effects of, for instance, combat, are often in the mid-range, there is a constant risk of creating what film sound designer Walter Murch calls a ''logjam'' (in Oldenbourg 2005). For instance, if a player's character is in an open field, talking to a nonplaying character, and is supposed to hear a gunshot in the distance, if the composer has chosen to include a lot of mid-range activity in that cue (e.g., snare drums and guitar), the gunshot, the conversation, and/or the music is going to be obscured. Whereas this has always been problematic in film sound and other linear media, in games the added unpredictability of where these events may occur makes the mixing far more difficult a task.

Real-time mixing in games now allows for sounds to be prioritized. As such, if the player must hear the dialogue to know what the next stage in the game is, the music or that explosive sound can drop down in the mix, as Rob Bridgett, sound director for Radical Games, describes:

> Ducking music and FX when dialogue occurs is a very basic way of achieving a more cinematic effect in games, and also of ensuring that essential information is conveyed and is clearly audible. The interactive mixing process can identify a whole slew of prioritised sound effects that need to be heard at designated moments in game play and sometimes dropping a sound entirely is the best option. Ducking importantly allows subtraction of sounds so that you don't just have to make everything louder in order to hear it. . . . Dynamically carving out frequencies from a music track when dialogue is playing for example is a great way of generatively allowing space in the music to exist when a dialogue event occurs. (Bridgett 2006)

As Bridgett (2008, p. 127) argues, previous attempts at mixing in games meant that a group of sounds had to be ''limited and compressed to extremes in order to compete with other sounds and music,'' or, in cases where simple mixing existed, implementation (via text file) was awkward and not ''artistically useful.'' In some ways, Bridgett (2008, p. 131) argues, it was almost beneficial that these older systems could not handle a significant amount of audio information at any one time, which limited the amount of simultaneous sounds:

> With the increased memory and available voices of next generation consoles and more sophisticated audio compression codecs, such as Microsoft's *XNA* and Sony's *ATRAC* allowing for a reasonable quality sample rate at roughly 10:1 compression, the amount of sounds that can now be played simultaneously has increased roughly ten-fold. This means a heavily increased role for in-game run-time mixing in order to prevent a cacophony of sound effects from playing back during run-time. Assigning a finite amount of available voices is one particularly crude way around this, but there emerge problems of really important sounds not being played back because other important sounds are also playing. Mixing in video games, as in cinema, is concerned with sound removal and subtraction rather than pushing volumes up. In mixing, it is often very subtle fader ducking that is required to allow a more important group of sounds to be made prominent above others.

There are still several necessary procedural stages and technical aspects to be explored by game audio designers. In particular, the psychological aspects of mixing can be a challenge, because the most effective mix may not be the most realistic: ''For example, consider any number of movies where the protagonist walks down a busy city street, and the audience hears primarily the interior monologue of the character's thoughts, not the traffic sounds'' (Grigg et al. 2006). It has been proposed that what is required is a ''smart audio engine'' that could

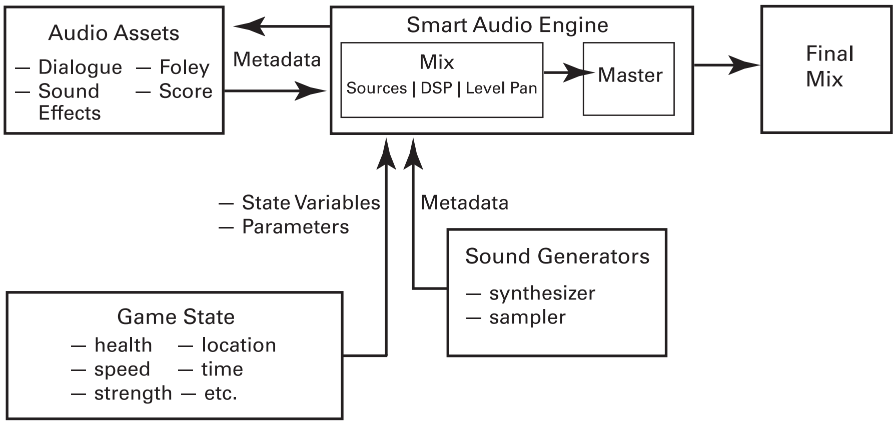

Figure 5.2
High-Level Smart Audio Mix Engine, adapted from Project Bar-B-Q (Bajakian et al. 2006).

take various sound assets, filter them with DSP according to various in-game parameters (game states, for example), perhaps synthesizing the sounds themselves in some cases, mix and pan within the engine in real time, master in real time and then send the sounds out to the player, as in figure 5.2.

There are different models for audio mixing in a game, including a post-production model as outlined by Bridgett (2007). Typically, post-production mixing has occurred in games only in cinematic sequences, in which the visuals can be locked down. Bridgett, however, working with Randy Thom on *Scarface*, created a model of ‘‘snapshot mixing’’ to mix sound, music, and dialogue by breaking the game down into several types of mix, including generic gameplay actions, locations, and specific events. Different mixes were developed for different areas of gameplay.

It is also possible dynamically to reduce or eliminate various frequencies from a music track in order to avoid a logjam with the dialogue. Describes Jay Weinland of the game *Halo* (in Hanson 2007, p. 51):

> One of the things that's pretty cool about our code engine is that we actually have real-time ducking in our game. . . . So that we can control any individual sound, we can control how we duck it under cinematic or other types of dialog, which is our biggest challenge. If the character is giving you some very important information, and all of a sudden three grenades land at your feet and blow up, you don't want that line to get stepped on. We have real-time parameters where we can say, ‘‘we'll duck this sound nine dB over two seconds and let it ramp back up after the dialog is done, over the course of another second and a half.’’ So it allows us real-time control over the volumes in the games, and it helps us to make sure that we can always hear the important dialog and other things that are important.

Such prioritization of sounds, though necessary, can mean that the mixer is making decisions that affect the ways in which the player hears and interprets sound. The composer may believe that the music at a particularly dramatic point in a game should have priority, in order to draw the player emotionally into the scene, while the dialogue may feature a *reveal* (a plot point that will guide the player); while the sound designer might have also spent much time unnecessarily prioritizing the sound of a key dropping off a desk on the right-hand side of the soundscape, for instance. Each of the sound assets may be thought by its creators (or indeed, the player of the game) to require priority. The mixer, then, must make difficult decisions regarding which sounds the listener should hear, potentially altering the intended impact of the audio, and the ways in which the audio information is received by the player.

CONCLUSION

It is important to note that there are specific differences in the production processes between film and games that impact the final sound of the game. For instance, in film sound, ambience is often recorded in the production space: if not at the time of filming, then typically on the same day, in order to ''corroborate, for the spectator, the original inscription of the character's speech'' (Geuens 2000, p. 219). Argues Jean-Pierre Geuens in his book *Film Production Theory*, ''ambiance is the cement which holds the entire aural construction together'' (ibid.). But of course, as with animation and some CGI-heavy live action films, there is no production space in games. There is no ambient background in which actors perform. As such, the sounds are all dubbed in. Dubbing used to be viewed as a last resort by filmmakers, since a good recordist working with a good mixer could get up to 70 percent of the production track into the finished film (ibid., p. 212). Geuens (ibid., p. 216) argues that in dubbing, we ''falsify the space'' since the original context of place is lost, resulting in an unnatural sound. It is ''a procedure to be avoided because newly recorded tracks invariably sound flat and dead in contrast with the location recordings'' (Michael Rabiger, in Geuens 2000, p. 212). In games, however, all sounds must be created in the studio or in field recordings, and then built up to sound real. Without a real reference point for the space that is being created on-screen, these sounds are inevitably in some ways ''less than real,'' and as such, in many ways sound designers compensate by creating a ''more than real'' sound in the studio (a topic I take up in chapter 7).

At the same time, the issue of mixing is much more complicated in games, which must take into account not only real-time changes in gameplay, but changes in the player's positioning of the character. Mixing in film is based on the assumption that the audience is static, an unmoving, passive receiver of the sound. Mixing in games must be based on the assumption that—though the player's actual position may not change—the player's character (and, by extension, therefore, the player) is constantly changing position. Planning to mix a surround-sound game, therefore, requires a set of skills that are becoming increasingly specialized.

As game audio develops, the roles involved are becoming more and more specific and dedicated. Whereas one person used to be responsible for all aspects of audio production and implementation, there are now teams of people with a variety of levels of artistic and technical skills. What needs to be stressed is that game audio is a collaborative process; the programmer cannot implement without the music, and the music, as was shown, depends to a significant extent on how it is implemented. Sound design must take into account the dialogue, and so on. The teamwork involved in creating game audio suggests an important reconstruction (or reduction) of the notion of ''author.'' As shown, sound design, dialogue, and music are as much about integration as they are about composition, and the ways in which the sound is implemented greatly affect the ways in which these sounds are received. To some extent such a relationship exists in film, but it is taken to an extreme in games. Music must adjust to the player's actions, in real time, to other audio in the same scene, and so on.

Moreover, when musical splits or ''chunks'' do not exist as a single linear track, there is no single ''musical text,'' and the author is to some extent the player/listener, whose moves affect the playback of these audio chunks, as is the composer, who is responsible for composing them, and the programmer, who chops them up and places them in a game. Cultural theory has for several decades been dealing with the notion that audiences themselves may construct meaning from texts, and the notion of text itself has become increasingly blurred. In games, these concepts become even more indistinct, as the player becomes an active agent of change, and the text is malleable and impermanent. This is taken even further in online games, for which there is no scripting, and no final version. Localization also further disintegrates the notion of games as a ''text'' or complete ''work.'' Not only is every game different owing to the participatory interactive nature of games, but games may be altered significantly for different markets, raising the question of what is the ''authentic'' version. This reduction of text and authorial power may even have an impact on the future development of popular music in general, as popular songs increasingly are licensed for games, a topic discussed in the next chapter.