A GAME AUDIO TECHNOLOGY OVERVIEW

VENERI Olivier+* *NATKIN Stéphane*+ *Le Prado Cécile* + *EMERIT Marc* *

+ Centre d'Etudes et de Recherche en Informatique (CEDRIC)
Conservatoire National des Arts et Métiers (CNAM) 292 rue Saint-Martin, 75003 Paris, France

* France Télécom
Recherche & Développement
Service TECH/SSTP
2, avenue Pierre Marzin
22307 Lannion Cedex - France

ABSTRACT

This paper intends to give an insight in video game audio production. We try to lay down a general framework of this kind of media production in order to discuss and suggest improvement of the actual tool chain. The introductory part is some kind of big picture of game audio creation practice and the constraints that model it. In a second part we'll describe the state-of-the-art of software tools and sketch a formal description of game activity. This background will lead us to make some assumption about game audio, suggest future research fields and relevance of linking it with computer music research's area.

1. THE BIG PICTURE

1.1. Game audio practices

According to [4] it may be said that we are entering into a third period in the development of sound within games. The first generation of games, supported on cartridges with read-only memory (ROM), like the Nintendo NES, for example, did not allow for storage of recorded sound. The music and sound effects were defined in symbolic fashion (similar to MIDI) and generated by the console synthesizer. This produced a type of characteristic music (Nintendo music) that one may still hear on a Game Boy—fairly exasperating for some people, but it has its fans.

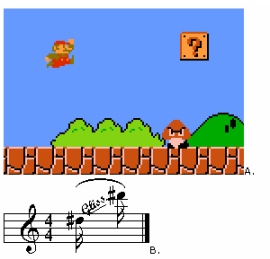

Figure 1 : Regular (small) Mario jumping (taken from [5])

Super Mario Brothers. (*Nintendo Entertainment System*©) © 1985 Nintendo of America Inc.

The revolution came with the arrival of CD-ROM support on consoles and PC's. It then became possible to store high-quality audio. To control the processor-load and memory allocated to the sound, synthesis and transformations are limited to the minimum necessary for managing the interactivity. With this goal, the sonic designer plays with a mix of many channels in real time (48 channels for the PS2), produced by dedicated hardware. The sound design of the majority of present games rests on these principles. The soundtracks of the games *GTA 3*, *Jak and Daxter*, and *Silent Hill 2* are examples of this second period. However, a new approach is beginning to be used, in particular for PC games. New capabilities and the use of powerful sound cards, like the Sound Blaster of Creative Labs, allow the reintroduction of sound treatment and synthesis in real time. As a first step, this involves simulating the acoustics and location of the objects producing the sound. The player, given a helmet or a 5.1 sound system, hears the bullets whistling overhead, and the voice of the monster is much more cavernous in a cave than at the seaside. Furthermore, the appearance of standards for synthesis techniques (using wave tables) allows a musician to control, in a much more precise way, the generation of music on the PC. This gives rise to the possibility of truly interactive, eventually generative, music for games. The soundtrack of a game like *Rez* is an example of the promising aspects of this evolution. In this game sound and some part of the visuals are generated accordingly to the player actions.

Figure 2: Screenshot of the game *Rez*

1.2. An industrial field

To have a clear understanding of what models the tools and the sound design practice, we must review the production constraints.
Generally, game sound designer cut-out sound design in three sound components namely music, voices and sound effects. Their relation ship to the game structure and to the images, are defined during the phase of game design [1]. Unfortunately, the sound is generally integrated into the game only at the end of production, when all the graphics have been constructed. The sound designer must therefore prepare all of the sonic elements without being able to test them before the final stage of production of each game level. In the final stage of the music production workflow, the integration and tuning is made, sometimes by another person, and the final work is submitted to the editorial team for approval.

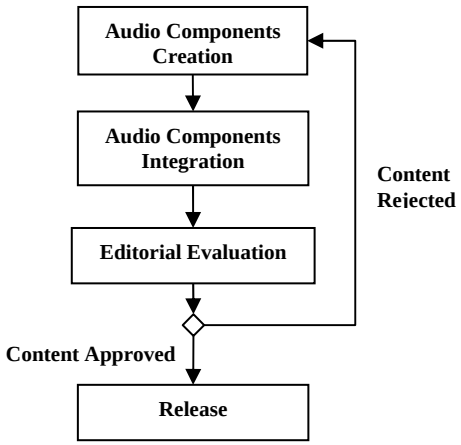

Figure 3 : Game Audio Workflow

In this canvas, the most complex part of sonic design is the construction of interactive music. Most present-day game designers use sound loop with each location in a given level or any defined game state. These loops must then be modified in real time to manage the transitions between scenes or state. The techniques of transition rely either on classic harmonic principles or on sound-mixing techniques [3][1].
We show in the sequel that the problem arises from a certain inadequation of audio design tools and the content creation needs. This gap is a consequence of an historical evolution which come from the 80's music tracker.

2. BACKGROUND

2.1. Architecture

In this section we describe recent game audio engine architecture and describe it precisely.
For the past few years, the software maker's products have been proposed without much success. Microsoft DirectMusic Producer is an example of such tool: it is a very complete and complex real-time sound editor that relies on the Microsoft DirectX sound API and that is based on the principles of classic music composition—the composer proposes melodies, harmonisations and orchestrations. A set of variations, chosen in real time, can depend either on game events or on random parameters. As the sample used can be music or Foley effects, in principle, the same tool can be used for sound design. It seems that this musical metaphor was too far from methods used by game sound designers.
The new generation of tools is much simpler. The Microsoft proposal is called XSACT and the CreativeLabs tool is entitled ISACT. The similarity is not only in the names; the two tools are basically real-time mixers and sequencers, i.e., they allow you to program the ordering and the merging of several sound sources coming from audio samples. They both allow the application of real-time transformations, in particular reverberation and spacialization of the sound, and the creation of random and event-dependent sound mixes. An interface with the game dynamic, through events and transitions, is supplied.
The description of this new tools are summarized by the iXMF file format description[2] This format is proposed by a working group that include major actor of the industry which purpose is to unified interactive audio practice. The key elements of this architecture are:

- The soundtracker manager
- The cues
- The events and scripts
- The variables

2.1.1. Soundtrack Manager

It's a software layer that reads audio content from file and manages its interactive playback with the game system via any number of "players"[2].

[1] Except the partial definition of "interactive music" this paper gives insight of the sound designer's way of thinking and doing interactive music for game.

[2] A player has the same reading functions as a traditional wave player such play, stop, pause…

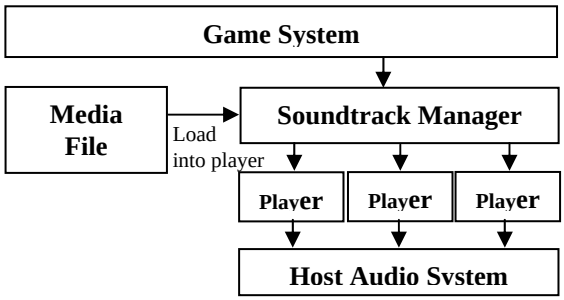

Figure 4 : Soundtrack Manager Description

2.1.2. Cues

Cues are the primary semantic content unit of the soundtrack manager. Each cue is determined by the sound designer and is used to communicate with the game system, i.e. the game request interactive audio services by calling a cue.

2.1.3. Events and Scripts

Script defines the response of a cue to a call of the game system. Scripts are defined by the audio artist in the authoring tool with some kind of pseudo code. They can play or halt a single audio file, or it may be complex, such as controlling the playback of a system of several related files and dynamically manipulating various playback parameters over time.

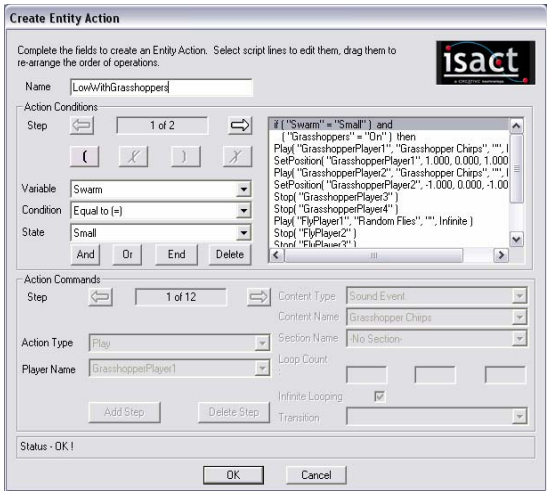

Figure 5 : Defining a cue with its script in ISACT

2.1.4. Variables

To enable communication between all the parts of the entire game/audio system, audio engines use variables. Local variable are available in cue in order to be set, read, modified and tested in conditional branches of script, they can also read and set by the game system. Global variable are maintain by the soundtrack manager and are visible by all the entities of the engine. They can be accessed and modified, like local variables, by the game system or the cues.

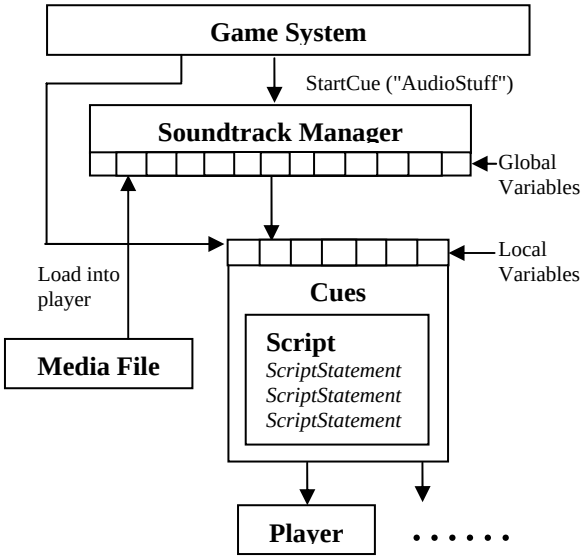

Figure 6 : Cues, Scripts and Variables Description

2.2. Game

In order to discuss audio engine's design the relation between game and audio must be taken from the ground. So we first shape what's behind the term "game" before relating it to music. Defining games is quite a complex question, but we try to fix first some concept about it, thanks to Zimmermann and Salen work [6], to provide a minimal formal frame to our discussion.

2.2.1. Meaningful Play

Zimmermann and Salen analyse the process of making game as the creation of *Meaningful Play*. *Meaningful play* in games emerges from the interaction between players and the system of the game. For example in Pong when you choose to turn the knob, the paddle moves and you, maybe, hit the ball, i.e. you make choices and take action. So we state that, according to Zimmermann and Salen, **"**Meaningful play occurs when the relationships between actions and outcomes in a game are both *discernable* and *integrated* into the larger context of the game" [6].

2.2.2. Rules

Meaningful play append in a formal structure constitute of rules defined by the game designer. General characteristics of game's rules are list by [6]:

- Rules limit player action
- Rules are explicit and unambiguous
- Rules are shared by all players
- Rules are fixed
- Rules are binding
- Rules are repeatable

In addition rules are layered in three levels:

- Constituative Rules
- Operational Rules
- Implicit Rules

We can view constituative rules as the core mathematical rules of a game; they defined the essential game logic. Operational rules usually correspond to the game instructions and implicit rules emerge from the game context and concern thing such as good sportsmanship and other implied rules of game behavior. In the next section we'll see why a clear understanding of these mechanisms is crucial for game audio creation.

3. CONCLUDIN REMARKS ABOUT GAME AUDIO

3.1. Meaningful play enhancement

We think *meaningful play* is the basis of constructing an immersive and engaging game experience because this is what models game media. In order to provide immersion and engagement to the player the entire system which constitutes the game must embed this game kernel. Accordingly to [5] our argumentation about immersion and engagement allows for a richer understanding of the complex communication involved in videogame music. In schematic terms, [5] state that "*immersion is the act of relying on learned behavioral scripts at a level of instinct – being "in the moment" without having to be aware of what it takes to be in the moment – while engagement is the process of learning the scripts and requires an objective awareness of the object supplying the new schema*"[1]. We state that game audio practices must rely on game mechanisms because the soundtrack is a major modality to achieve *meaningful play* i.e. the sonic design must underline the rules that model the game world in order to increase its consistency.

[1] Replace *scripts* by *rules*

3.2. Game music writing's specificities

Behind this formal conclusion about relation between game and his music we need to specify things about writing process.
A game is first of all an imaginary universe which may be neither revealed nor created by following the linear track of a story. The first stage in any game design is that of imagining and describing all of the *object classes* which exist in this universe. Something has the status of object because it can produce or experience an action which has an influence on the progress of the game. We need to think game's music the same way, as object or part of game's world object, because there is no explicit or implicit description of time: the composition must come from a spatial and logical relation between objects.

3.3. The needs

As we've seen the tools for sound designer rely mainly on the sequencer/tracker paradigm, a composer and/or sound design produces sound assets and give them to a sound engineer that fixes them with the media. Such a process seems to be a more convenient one to deal with game audio production because of historical reason from our point of view i.e. game production was seen as a subset of linear media production like TV, cinema....
We think game industry was at a turning point about game audio technology. There's a need on one side for generic tools witch can be use in a wide range of game genre and on the other side for media specific tools to provide composer the ability of writing and connect music's logic to space, drama and emotional state representation.

4. FUTURE WORKS

Our assumption was that it could be possible to design tools that provide a convenient interface for designer and fit with game specific functionality. We argue that music research's area like computational musicology, formalisation of musical structure have common problematic so they must be investigated and linked together.

5. REFERENCES

[1] Gal V., Le Prado C., Natkin S., Merland J.B. and Vega L., "Processes and tools for sound design in computer games", *International Computer Music Conference*, Goetborg, *Suede,* 2002.

[2] Grigg Chris, "Interactive XMF - A Standardized Interchange File Format for Advanced Interactive Audio Content", *MIDI Manufacturers Association*, Los Angeles, CA, USA, 2004.

[3] Harland Kurt, *Composing for Interactive Music*, 2000. www.gamasutra.com

[4] Natkin Stéphane, *Video Games and Interactive Media: A Glimpse at New Media Entertainment*, A.K Peters, Wellesley, USA, 2006.

[5] Whalen Zach, “Play Along: An Approach to Videogame Music”, *the international journal of computer game research*, volume 4, issue 1 November 2004. www.gamestudies.org

[6] Zimmermann E., Salen K., *Rules of Play: Game Design Fundamentals*. The MIT Press, Cambridge, Massachusetts, USA, 2004.