Journal of Gaming & Virtual Worlds
Volume 5 Number 3
 doi: 10.1386/jgvw.5.3.215_1

NIELS BÖTTCHER
Aalborg University

Current problems and future possibilities of procedural audio in computer games

ABSTRACT

Procedural audio is an alternative approach to sound design in computer games, which offers extensive possibilities for dynamic audio. There are many arguments for using this approach to sound design, but as of today procedural audio has not been utilized in many contemporary commercial computer games. This article is based on a series of interviews with audio programmers, sound designers and audio middleware software developers between December 2011 and May 2012. The article discusses and summarizes the most important conclusions from these interviews concerning procedural audio in computer games. In doing so it seeks to answer the question: Why is this approach to sound design not more popular in modern computer games? The lack of well-integrated tools is mentioned as being one of the most important issues. Sound quality, education and poor communication between the industry and academia are other important issues. There are different approaches for designing procedural audio tools and different possibilities for improving the perceived sound quality of the models. In the future, one would not expect to see a full integration of procedural tools directly in the audio middleware, it being more likely to see those tools as plug-ins from third-party companies.

KEYWORDS

procedural audio
game audio
audio middleware
design approach
integration
interviews

INTRODUCTION

Within the last decade computer games and the technology underpinning them has evolved significantly. New motion controllers are regularly put on the market, advanced physics and artificial intelligence (AI) have been introduced to the world of games, and the underlying graphics and hardware technology has also changed dramatically. It has become a standard for most bigger computer games to include physics-based graphics, realistic animations based on motion capture, artificial intelligence and more. But what has happened to the audio side of things?

Generally speaking, most audio in contemporary computer games is produced the same way that it was a decade ago. The sound effects are based on pre-recorded audio and Foley effects (Filmsound.org website) similar to what one knows from film postproduction. Instead of focusing on the production of the sound effects themselves, much work is focused on processing, sorting and mixing the pre-recorded audio. Nonetheless, sound artists produce high-quality recordings of music, voice and sound effects for the games, but most often they make use of standard recording techniques and large databases of existing sound effects.

The use of pre-recorded audio became possible in computer games when the sampling technique used in digital recording enabled the sound designer to convert continuous analogue signals into discrete time-sampled signals (Roads 1996: 9), and mainly when the CD was introduced to the consoles. Before this was possible, the sound designers had to generate or synthesize the sound in real time by using different basic synthesizer components, such as noise, oscillators and filters. Among other places, this technique was used in the early Commodore 64 games and those for other similar consoles from this period of time. At that time, no pre-recorded sound effects were utilized.

Once the sampling technique and CD technology became available to game consoles, the game companies and sound designers started using pre-recorded audio. Within a short period of time, this became the only technique used for producing game audio. The use of pre-recorded audio opened up the world of game audio to less technical sound designers and enhanced many aspects in relation to the sound quality. At the same time, sample-based sound design also introduced new problems for sound designers. Suddenly, much effort had to be spent on designing randomizations and creative compositional approaches, in order to avoid perceived repetitions of sound effects and musical loops (Collins 2008: 66–67).

When looking at the complexity of contemporary computer games, it could potentially be relevant to include additional techniques for producing the game audio. Alternatives to pre-recorded audio are especially interesting and relevant in games with a high frequency of dynamics, such as games using physics, AI, differentiated rendering and similar.

An alternative and not very common approach to game audio is procedural audio. In this article procedural audio should be understood as real-time-synthesized audio, where the timbre and characteristics of the sound are based on variables from a game engine, models or rule-based structures. Andy Farnell (2007) describes procedural audio as a '[n]on-linear, often synthetic sound, created in real time according to a set of programmatic rules and live input'.

Synthesis techniques for producing procedural audio basically include almost any synthesis technique except for sampling. Examples could include

among others things subtractive synthesis, granular synthesis, physical modelling, additive synthesis, frequency modulation and more.

As mentioned above, procedural audio is by far a new concept and was in fact the only possibility for producing sound effects and music in the early video games. With the highly dynamic computer games of today, there are still many reasons for using procedural audio. N. Böttcher and S. Serafin (2013) present various reasons for utilizing procedural audio. Some of the main arguments for using this approach to designing sounds are summarized in the following.

When implementing sound effects in a computer game, one of the most time-consuming tasks is the process of recording or processing multiple similar sounds in order to avoid perceived repetition. One good example of this includes the sound of footsteps in a first-person-shooter computer game. Here the sound designer most often has to implement several different footstep sounds that can be randomized in order to avoid repetitive sounds. In addition to this, the game often has many different surfaces and characters that also have to be simulated in the same way. By adding all of these together, the sound designer often has to produce a very large number of footstep sounds for the entire game, which is a very time consuming process. The same phenomenon is also happening in relation to weapon sounds, collision sounds, sounds related to movement, etc.

By utilizing procedurally driven audio, the sound designer could potentially generate automated or generative randomizations of the sound effects in real time by computational sound models. One of the other strong arguments for utilizing procedural audio is when it comes to the newer motion controllers, such as the Nintendo Wii remote (Nintendo official website), the Microsoft Kinect (Xbox official website), the PlayStation Move (Playstation official website) and similar. All of these devices include continuous 3D sensor data describing the motion of the players. Most often these controllers are not utilized to their full potential, as static sample-based audio is applied to the movement of the controllers by triggering samples, when pre-determined thresholds of the sensor data are met.

Utilizing pre-recorded sound effects requires that the sound samples are stored in the memory of the gaming console. Although the amount of console memory has increased significantly during the last few years, consoles still have a limitation on how many samples can be loaded into memory and streamed simultaneously. By using procedurally driven audio the sound designer could save significant amounts of memory space in the consoles, which is another strong argument for using this technique.

As of today very few contemporary commercial computer games utilize procedural audio. One of the few games often mentioned in the context of procedural content is the life form simulation game *SPORE* (Maxis/EA Games, 2008). Among other things, the music in *SPORE* is generated using procedural techniques. Another example is found in the game *Chicken Doom* (Bulkypix, 2012) developed for the iPad. Mainly sample-based audio is utilized in the game, but in a smaller scene of the game procedural audio is used to generate whoosh sounds connected to the speed of the player's fingers on the screen.

In addition, alternative approaches to sound design related to procedural audio have been implemented in different racing games. As an example of this, the sound designers behind the *Need for Speed* series of games (EA Games) utilized granular techniques when implementing the vehicle engine sounds. In several different academic projects and articles, procedural

audio or physically based sound models have been proposed for computer games (Resounding website; Gatheral.co.uk website; Paul 2008; Lloyd et al. 2011; Dobashi 2003; Farnell 2007; Veneri et al. 2008; Raghuvanshi et al. 2007; Verron et al. 2010). Nevertheless, very few developers tend to ask the question as to why this design approach has been applied to so few commercial games, as presently seems to be the case.

The purpose of this article is to take a deeper look at procedurally generated audio to try to understand why this approach to sound design for computer games has gained so little popularity in the commercial games. However, one of the problems with doing research in game audio is that much of the cutting edge technology is kept within the game companies in order to guard commercial trade secrets. In the hope of discovering some of the knowledge that exists within the game companies, a series of interviews with people involved with game audio were undertaken between December 2011 and May 2012 from a few different perspectives. All of the interviews were executed via Skype, except for one face-to-face interview. The interviews were all semi-structured and lasted 30–50 minutes.

The main motivation for the series of interviews was to get a better understanding of why procedural or real-time generated audio has not been more popular in contemporary commercial computer games till to this point. Furthermore, the aim was to gain insights into the important factors that are worth considering when producing future procedural models or tools. The idea was to talk to a selection of the leading audio programmers, sound designers, and developers of audio middleware, as well as academic researchers who had additional experience from the game industry.

The following persons were interviewed:

Andy Farnell
With a background in computer science and electronic engineering from University College London, he has been working as a sound effect programmer for BBC Radio and is seen by many people as one of the pioneers within procedural audio for computer games. He teaches at many different universities around Europe and has worked as a consultant for various sound-related companies. In 2010 he released the book *Designing Sound* (Andy Farnell 2010) which is one of the few books released on the topic of procedural audio for computer games.

Gordon Durity
Executive audio director at Electronic Arts (EA Sports), Vancouver, and the owner of the company Durity Mediaworks. Additionally, he has been working as instructor at the Arts Institute of Vancouver. He has worked on various AAA games (a term used to describe a large commercial game) creating audio design, implementation research and design, mixing and other topics from companies such as EA, Radical Entertainment, Airwaves and others.

Leonard Paul
Sound designer, composer as well as coder for several computer games such as the *Need for Speed* series (EA Games), *NBA Jam* (EA Games, 2010), *NHL*11 (EA Games, 2011), *Retro City Rampage* (Vblank Entertainment, 2012) and many others. Leonard Paul has worked for companies such as EA Games, Backbone Entertainment, Radical Entertainment, Black Box Games and others.

Amaury La Burthe
Founder of the game audio middleware company AudioGaming (AudioGaming official website), which is primarily focusing on developing procedural audio tools for computer games. Amaury La Burthe has been working as the lead sound designer for the game company Ubisoft Entertainment as well as Sony Computer Science Laboratory. He has a background as a research engineer and holds an M.Sc. in signal processing, acoustics and computer science applied to music from IRCAM, Paris.

Jan Murguc
Audio programmer at Unity Technologies (Unity3d official website). By the time of the interviews he worked as audio programmer for the Danish game company IO Interactive (IO interactive official website), developing among other things the in-house audio tools used by IO Interactive as well as designing and implementing the game-specific audio code for *Hitman Absolution* (IO Interactive/Square Enix, 2012).

Nicolas Tsingos
Currently Nicolas Tsingos is a permanent researcher in the REVES research project at INRIA. He was a member of the technical staff at Bell Laboratories, Lucent Technologies in Murray Hill, USA. He holds a patent on perceptual techniques for audio rendering, a technique that has been used in computer games such as *Test Drive Unlimited* (Atari, 2011) and *Alone in the Dark 5* (Atari, 2008).

Nicolas Fournel
CEO of the Japan-based company Tsugi (Tsugo studio official website). He worked as the principal audio programmer at Sony from 2008 to 2011 developing tools and run-time components for the PlayStation 3. Furthermore, he developed the SPARK tool (Scee Procedural Audio Real-Time Kernel) – an in-house tool from Sony for analysing and automatically creating dynamic models out of static samples. Before that he worked at EA Games, Vancouver, as a software developer. Additionally, he worked in the company Konami (Konami official website) where he was the leading audio programmer on productions such as *Dance Dance Revolution* (Konami, 1998), *Frogger Helmet Chaos* (Konami, 2005) and many other games. In relevance to this article, he maintains the website www.proceduralaudio.com.

Gino Bollaert
Software developer at the Australian based company Firelight Technologies working on the game audio middleware FMOD (FMOD official website).

Benoit Alary
Software developer at the company Audiokinetic, working on the game audio middleware solution Wwise (AudioKinetics official website).

This article discusses and summarizes the most important opinions and conclusions derived from the interviews. The article is structured into the following three main parts:

1. A discussion about some of the possible reasons behind the lack of commercial success for procedural audio in contemporary commercial computer games.

This includes discussions about sound quality, sound design tools, CPU usage, workflow in the companies, communication between industry and academia, sound design education and economic aspects
2. Future perspectives for procedural audio as part of the audio middleware technology
3. Suggestions for improved middleware tools and how to use and enhance procedural sound models in general.

WHY IS PROCEDURAL AUDIO NOT ALREADY A COMMON ELEMENT OF COMPUTER GAMES?

When asked why procedural audio is not being utilized in modern computer games, the game developers that were interviewed tended to agree on two main reasons:

1. The sound quality of the procedural models is not yet good enough
2. There are no reliable tools for producing procedural audio in larger commercial game productions.

When looking at the first point concerning inadequate sound quality, one could ask what that really means in relation to procedural audio, as the term sound quality can be quite subjective. As an example, one of the interviewees answered:

> Purely procedural models … the quality just isn't really empowered … like what can be recorded and played back … I think we are still missing some sort of signature sound out of those procedural models ….
>
> (Benoit Alary)

Many different procedural sound models exist and of course the quality of those models greatly differs. Considering the argument that the sound quality is not yet good enough, it is likely to conclude that people are generally referring to missing residuals and missing complexity in the frequency spectrum. A well-recorded Foley sound effect, which has been processed in Pro Tools (Avid official website) or similar, is by now the standard in computer games. For many people a sound effect with less detail would simply be perceived as a sound effect with poor sound quality.

> CD-quality audio is by now standard, so if your procedural audio work does not achieve this at least, it's hard. Maybe you can do a little less in terms of quality, but it has to be good, it has to be really good.
>
> (Amaury La Burthe)

But does the perception of sound quality in a dynamic interactive computer game differ from the perception of the sound quality of a linear soundtrack that one knows from a typical movie? Farnell argues that perceived realism, which in many cases could influence the perceived sound quality, has to be evaluated in another way when discussing procedural audio:

> You can't quantify impact or what is real … it is not a function of a single listening. It's a function of behaviour over time – so realism becomes a more interesting thing.
>
> (Andy Farnell)

Related to the discussion about sound quality, many procedural models tend to aim for simulating reality or even to implement a hyper-realistic physical model of an object or event (Yoshinori 2003; Lloyd et al. 2011; Raghuvanshi et al. 2007). Often the approach to designing the model is to first of all understand how the sound is produced in reality and then attempt to copy this with a more or less simplified synthesized model. When this is done the game engine and the possible physics could be mapped to specific parameters in the sound object controlling the model when the game is running. The focus is very often on the natural behaviour of the sound more than the aesthetics or hidden narrative of the sound.

But is a very realistic sound model actually perceived as having high quality sound and is naturalistic realism actually requested in computer games in general?

> I don't think the realism aspect of it is so super important … it's more … is it consistent with your game world? … like some old war movie where you feel like you are actually inside the movie with that black and white kind of aesthetic … the aesthetic of that time period to make you feel like you are going back in memory.
>
> (Leonard Paul)

> A lot of people think about procedural audio as physical modelling … it's like something happening in the world and I want to hear the correct sound for that. First in sound design the correct sound is usually boring and not interesting. … it doesn't have to be physically correct … procedural audio is not necessarily based on physical rules … it's just synthesis of sounds in real time … there is no reason why you can't use it to make any kind of sound, any cartoon-like sound, any bleeps and blops.
>
> (Nicolas Fournel)

In general, many of the more critical comments tended to argue that most of the procedural models developed so far tended to focus mainly on simulating reality. Because it is extremely complex and requires a lot of CPU power to calculate the many details of a natural event in real time, the procedural sound models are most often simplified in a way that results in less-detailed frequency spectrums compared to the sound of a natural event that has been sampled. Additionally, the sound models do not carry any chosen aesthetic characteristics.

Another reason for the limited commercial use of procedural audio techniques, as mentioned by those interviewed, seemed to be the lack of adequate tools integrated with the software that is normally used at the game companies and by the sound designers in general:

> The main problem is a lack of very good sound game models. Procedural audio is all about the model … a lack of models and adequate tools … There are very few cases of procedural audio really used in games and that's one of the reasons because there are no good models and there is no easy way to use them … a lot of stuff comes right now from the academics. Those people don't necessarily have a good representation of how things will happen in a production cycle of a game.
>
> (Nicolas Fournel)

> It is not that people are not interested, but they really do need reliable tools before you can convince them to really change their workflow.
>
> (Jan Marguc)

When implementing the audio in computer games, most companies utilize so-called audio middleware solutions. Two of the main middleware solutions often used by AAA game studios are FMOD Studio (Firelight Technologies) and Wwise (Audiokinetic). Some of the larger game companies have developed in-house customized middleware. For instance, this is the case with companies such as Electronic Arts, Sony and IO Interactive.

Audio middleware can be described as a tool that allows connectivity for game engines. By using an audio middleware solution the sound designer can reduce programming time and test the sound design in real time by running it along with the game engine while linking the sound effects to specific game variables such as objects, areas and events. Audio middleware typically includes user-friendly GUI's for controlling and processing samples. The sound designer can quickly add sounds and rules for triggering and playback of those sounds, and then test everything with the game running. Additionally, there is most often a more low-level programmer-oriented part or API of the middleware, where one can basically implement everything that is possible with a programming language – just without any graphical user interface or premade solutions.

As it stands today, audio middleware is mostly used to handle the large amount of pre-recorded samples and for triggering those samples according to different actions and events in the game. Additionally, work is put into making alternative recordings and performing randomizations and processing of the samples in order to avoid repetition. As mentioned earlier, this is typically performed with sounds such as footsteps, weapon sounds, collisions and similar action-related sounds that one hears frequently during a game.

People working with procedural audio are currently often using platforms such as Pure Data (Pure data official website), Max/MSP (Cycling74 official website) or SuperCollider (SuperCollider official website). More low-level code, such as that obtained by using C++ or other programming languages, is also used in some cases to implement procedural audio. The reason for this is that the audio middleware solutions are not designed for working with real-time synthesized audio and no graphical interfaces have been developed to accomplish this.

When implementing procedural models from scratch the sound designer or programmer would most often need basic audio elements such as filters, oscillators, noise generators and similar to run in real time. Most of the audio middleware tools do support the possibility of implementing these basic audio elements, but this approach still requires a great deal of complicated programming to actually implement it and control it.

> We do support like sound waves and filters so there is kind of ways where we've done projects on our own … in theory we have also what is required for that, but it's the GUI … the interface is not really meant for that … so it's not really useful.
>
> (Benoit Alary)

As mentioned above, some of the bigger game companies have their own custom audio engines. In some of these cases, more dynamic sound design

approaches are used, but mainly for processing of sampled sounds and not as real-time synthesis.

> The system we have is very much based on Max/MSP pure data model. It's just completely customised to work specifically for games in the run time as well as authoring tool … it primarily is there to modulate and direct and control audio. Basically what Max does, but with the DSP side of it … without the audio generation side of it.
>
> (Gordon Durity)

When all the basic elements of procedural audio actually already exist in the middleware, can one presume that it might be just a matter of time before we see these functionalities included in the commercial middleware? When talking to some of the software developers at Wwise and FMOD they both agreed that the most urgent need at the moment for game companies in relation to audio is to improve the workflow when managing thousands of assets, as well as integrating the tools that the sound designers are used to working with in their sound design studios.

> The demand is mostly for the workflow … helping with the workflow of managing hundreds of thousands of assets … try to manage more and more bigger amount of sound effects and pre-recorded samples … what they are mainly after is a better workflow to manage all of that and create sort of a bus system with effects … like real-time parameter control.
>
> (Benoit Alary)

Within the commercial audio middleware solutions, a few plug-ins for producing procedural audio already do exist. Wwise has a plug-in called SoundSeed (AudioKinetics official website for SoundSeed) for producing wind and whooshing sounds as well as impact sounds. The company AudioGaming has developed a set of plug-ins working for FMOD. Those plug-ins includes among other things procedural wind, rain, car engines, fire and footsteps. It has not been possible to find any information about the popularity of these plug-ins but, judging from the interviews, the frequency of their use seems to be rather limited among today's sound designers and game companies.

Aside from the issues of sound quality and lack of adequate tools, a common argument held against the use of procedural audio is the increased use of the CPU. Before discussing increased CPU loads, it is relevant to understand how much of the total available CPU time that the audio in a computer game is typically using. When asked how much of the total CPU the sound designers were allowed to allocate for audio on gaming consoles, the interviewees answered between 5–10 per cent of the total available CPU time as an absolute maximum.

When asked if it would be possible to have oscillators, filters and similar elements running in real time and still just use this small amount of the CPU time, some of the comments from the interviewed were:

> I would say yes 100% yes … we should be able to do this now and definitely the next generation of PCs and consoles … I don't see why. The CPU is there … it's a matter of if there is a desire to do it.
>
> (Gordon Durity)

> I think there is no way we can use procedural audio for the whole game. There are many reasons for that … it would take too much CPU. We definitely do not have the models that sound good enough to recreate everything we need in a video game. Some models are ok, like wind, impact … that are the ones we always see and hear … but a lot of other models are just not available where you want them to be for a professional game.
>
> (Nicolas Fournel)

It seems as if it would be possible to generate the procedural audio in real time on the consoles, or at least part of the audio within a complete game. The next step is then to actually build the tools and this step also seems to have a few obstacles. One of the biggest obstacles is how to convince people to change their workflow and routines with sound design:

> It's a question of modellers and artists and computer scientists working on the framework to do this technology, but the obstacles are political. The obstacles are to do with workflow and to do with established technology and its familiar ground.
>
> (Andy Farnell)

> The first big challenge is the mind-set from the creative involved. … the mind-set of the creative leads, creative directors working in games right now, is coming from the model of television aesthetics … apart from the mind-set from the creative side … it is also a mind-set of the technical side to actually provide the audio system and not core data to drive procedural stuff. It requires a lot more actual AI and contextual information being sent to the audio system than just playing a sample. …
>
> (Gordon Durity)

Apart from the change in workflow and mind-set, it also becomes an economic discussion, as it would require a team of audio programmers to develop the tools. Finally, there is the discussion about sound quality and CPU usage, which could potentially evolve into both technical and aesthetic disagreements on priorities between the many different specialists working in a computer game studio.

COMMUNICATION BETWEEN INDUSTRY, ACADEMIA AND DEVELOPERS

Another issue that people from both companies as well as academia mentioned was the limited communication between industry, academia and the developers of audio middleware. The companies in the industry do not share their knowledge and most often they keep the technology and software behind their games secret for many years before something new is developed. On the other hand, many researchers within the field of real-time synthesis for virtual worlds and procedural audio tend to research in fields they believe are of interest to the gaming industry, without actually communicating with the industry and trying to understand the workflows and needs of a sound designer in a game company. This often results in models or tools that might not be useful for the game companies, as they are not designed specifically for games and are unsuited to being integrated into the workflow and software

normally used by the sound designers and game companies. Nicolas Fournel describes the need for communication as the following:

> Better communication at all stages. The communication to create the models so people could share models. Better communication between academics and the industry.
>
> (Nicolas Fournel)

With better lines of communication, academic work within the field could potentially inspire the industry, but most often it is never really integrated into any tools, as the companies do not have the time and money that it takes to actually implement the tools and train the designers to use them.

The problem of poor communication as well as collaboration does not only exist between the game developers and academia. It is apparent that communication between the audio middleware developers and the industry seems to be difficult when the topic is procedural audio:

> We are really trying to bring tools in the way that's very easy to grasp and to use for sound designers – and we really try to adapt to what's practical problems for them … but it's not always easy … we don't feel that much involvement of the people … we have to guess what they would like.
>
> … It is a shame that there are a lot of people that are really enthusiastic about procedural audio … we don't really feel a lot of enforcement from the gaming industry … I think what would be good is to identify people that are really ready to commit just a little time and give practical case where they have problems and where they think maybe some kind of procedural audio solutions could help and start from that. A concrete case from their side.
>
> (Amaury La Burthe)

SHOULD THE SOUND DESIGNERS BE MORE TECHNICALLY EDUCATED?

Some of those interviewed argued that the education of sound designers is also an important aspect that might influence the use of procedural audio. Both the content and methods differs greatly from one educational institution to another, but in many cases the sound design education syllabus might still focus mainly on linear sound design, as one knows from postproduction and film sound. Here the focus is very often held on aesthetic and narrative aspects of the sound design. Some of the interviewees argued that sound design for games would benefit from a more technical knowledge and in some cases a different mind-set. Hereunder especially when discussing procedural audio:

> It's a question of education I think. If nobody understands or if nobody feels any need we will never achieve anything … Most sound designers I have met are more sound engineers … there is no value comparison or anything, it's just that they are more practical people doing sound design and having done sound engineering studies. … It's probably kind of not necessarily knowing exactly what you can do and kind of a not feeling very comfortable with those kinds of technologies.
>
> (Amaury La Burthe)

> The job in my mind is both technical and creative. The creative part needs to be there, but realistically, if you are a senior post production or sound designer, how can you possibly make any sort of design decisions or have any objective, unless you actually know a bit about how technically things works.
>
> (Gordon Durity)

Another group of those interviewed disagreed on this point of view, and argued that the problem was mainly related to a lack of adequate tools and that the sound designers were more technical than most other designers from different fields:

> I think they are actually the most technical artists out there … I just think nobody came up with something to give them to work with.
>
> (Benoit Alary)

> I think we are coming with a new generation. Sound designers are very much into the technical side of things. You see a lot of people who are like more technical sound designers – so basically a mix between creative and technical skills ….
>
> (Nicolas Fournel)

ECONOMIC INFLUENCES

Economic issues were also one of the last things mentioned by a few of those interviewed as being another potential reason for the limited use of procedural audio. If a company chooses to spend time and money on research and the development of procedural audio models or technology, it would naturally expect to gain increased sales or in some other way benefit from using procedural models in the longer term.

When compared to the other possibilities of making money as a result of the sound in a game, it could be hard to argue that procedural audio would have a significant influence.

> It's hard for us to demonstrate that it is gonna help them sell 10% more games.
>
> (Amaury La Burthe)

It would be more common to accumulate sales by using famous artists to do the soundtrack for the game, as is seen with many modern computer games. For example, the *Grand Theft Auto* series (Rockstar Games) has a radio included in the game that has DJ mixes from artists such as Femi Kuti, Roy Ayers, and DJ Premier. The game *Little Big Planet 2* (Media Molecule/Sony Computer Entertainment Europe, 2011) included contributions from artists such as Plaid, Squarepusher and Trentemøller in the sound track.

When game companies are investing in new rendering techniques for graphics or implementing advanced animations, character design or similar, it is very easy for the end-user to perceive the impact of this in the game by watching a trailer, looking at the cover or similar. Smaller details improved by procedural audio will in most cases not even be perceived by the gamer. In the best case scenario, the player might perceive the difference only after several playthroughs of the game.

As an example of the low level of awareness that players have of the audio in a computer game, an experiment was performed at Aalborg University Copenhagen in Denmark (Böttcher 2013). Here a pre-recorded sound was compared to a procedural sound model of a sword-swing in a game designed for the Nintendo Wii remote. As a rather big surprise to the author, only eight out of 40 test participants noticed a difference in the sound when playing the game. On this basis, it might be reasonable to ask whether or not it is really worthwhile for a game company to spend the time on procedural audio.

It is often discussed as an issue that the companies do not have the time to do any research and development in procedural audio because of the budget and limited time for a large game production. In the longer term, procedural audio could potentially end up being that much more interesting, interactive and congruent with the game and controllers that some games could gain an economic benefit from extra sales. Another possible long-term effect from using procedural audio, which in the end could have other economic benefits, is a possible reduction in the hours spent by sound designers on producing variations of sound effects:

> If you have ten hours to prepare your sound effect in Pro-tools, it's going to be better than something I have to do in real time with the limited CPU. But I can now do that a hundred thousand times a second in a hundred thousand different ways, whereas the ten hours that you have been dealt with in making that one sound are, to use an English expression, cast in stone … frozen.
>
> (Andy Farnell)

Potentially, procedural audio tools could additionally gain some unforeseen novel sound design approaches, if sound designers got used to the tools as well as a different mind-set.

AUDIO MIDDLEWARE AS A FRAMEWORK AND NOT AS AN ULTIMATE AUDIO PACKAGE

As the lack of tools for producing procedural audio was frequently mentioned as one of the main reasons behind the limited use of this sound design approach, it was decided to interview software developers from two of the main audio middleware companies about the potential of a future integration of procedural audio tools in their existing software. As there are currently not many tools available with well-designed GUIs one could imagine that the audio middleware companies would implement procedural tools in the future. In these interviews, developers from both of the middleware companies, Wwise and FMOD, mentioned that these tools would have to come from third-party companies:

> Possibly then we will also do certain amount of elementary procedural tools in the sound designer tool, but mostly we are interested in actually using third-party software like using technology that have been crafted for that purpose … so we wanna outsource that as much as possible and focus on a sort of framework that will enable that software to run.
>
> (Gino Bollaert)

Both software developers agreed that the future of audio middleware was more to function as a framework that is compatible with various plug-ins, instead of being one complete solution. In some of the leading audio software applications it has become more of a standard that different software or plug-ins integrate and collaborate. For instance, this is seen with the bigger music software such as Logic Audio (Apple official website) and Cubase (Cubase official website), where both enable various plug-ins to be used with the software. Software such as Ableton Live (Ableton official website) and Max/MSP has been integrated into the so-called Max4Live (Ableton official website for M4L) and similar. The software developers from both Wwise and FMOD agreed that it seems somehow to be the future to integrate different applications instead of developing complete solutions:

> FMOD studio would be more the framework that sort of ties it all together and is as flexible as possible in terms of hooking up with all those partners all those different software companies … I think that's the ultimate aim really … I guess for me what I see as most important I think is the collaboration between different software and this is something that is happening increasingly … even in music software … if you look at Ableton and how they have sort of worked with native instruments and other … Max/MSP, Max4Live and that kind. I think that's the way … that's the way forward.
>
> (Gino Bollaert)

Some third-party software is already used with both of the middleware solutions. As an example, the so-called iZotope (Izotope official website) plug-ins, used with software such as Pro Tools and similar products to perform audio signal processing, are now compatible with both FMOD and Wwise. This means that the sound designers can actually work with the plug-ins they normally use in other situations and embed these effects in the audio middleware:

> People can use their favourite pro-tools plugins inside games … I think it makes sense as well that other kind of tools like sound design tools also get integrated into sound engines … they wanna be using the tools that they are familiar with … so let them use the tools they feel comfortable within instead of trying to come up with a totally new system you from scratch designed specifically for games for example.
>
> (Gino Bollaert)

> Wwise has a plug-in interface, so like everybody can create plug-ins that can be incorporated in Wwise and then the good thing is that you mix it with other effects … the main focus of the software is to manage multiple layers of sounds and then people create plugins to create different input if you want.
>
> (Benoit Alary)

DESIGN APPROACHES FOR PROCEDURAL AUDIO TOOLS

In the last part of the interviews it was discussed how the design approach of procedural audio could be enhanced and also how the tools could be developed in order to increase the likelihood of this sound design approach

gaining more popularity among the sound designers and game companies. As there are many different types of sound designers working, as different from one another as any musician or sound artist, there is not simply one approach to design the tools for producing procedural audio. In the interviews, different opinions about how to think of procedural tools became evident. The main differences were between automated tools, pre-made procedural models and more flexible tools.

As an example of the automated design approach, you have tools such as the SPARK tool from Sony developed by Nicolas Fournel. The idea behind this tool is basically to load in a sample and perform analysis on various characteristics of the sound. When the analysis is performed the tool is then capable of re-synthesizing a procedural interactive model based on the pre-recorded sample. This automated procedural model is now capable of changing textures and many different parameters in real time.

The design approach behind this tool was to maintain the workflow of the sound designer and just provide assistance, with the tools performing automated procedural models or randomizations. When using this tool the sound designer should simply have the focus on crafting aesthetic sounds and the tool creates the procedural models automatically. By using this approach the aesthetic choices are also left completely up to the sound designers and not influenced by the chosen tool:

> … sound designers … they don't want to use a sample that comes from a sound library … because they want to craft their way. … if you are a company and you start selling models … already made models, that means you already made some choices, some aesthetical choices about those models … you are already given something which is very limited. … already you limited the possible range of sounds and that is not necessarily the one the sound designer had in mind.
>
> (Nicolas Fournel)

Another design-approach comes from companies such as AudioGaming that have developed a set of pre-made models simulating wind, engines or similar. Those models are well integrated with the middleware as plug-ins with an easy to understand GUI. One can then load in the procedural model and map variables from the game engine to different parameters in the procedural model such as density, colour and attack for the rain model.

Some of the critics to this approach, such as Nicolas Fournel, are of the opinion that the sound designers are not free to design their own sound aesthetics, but have to use the limited possibilities of changing timbre in the model. One of the great things about this approach is, on the other hand, that it becomes very fast and easy to simulate weather or similar and the sound designer just has to choose between different parameters and map the data from the game engine to the sound model.

Finally, you find people such as Andy Farnell who believes the tools have to be as flexible as possible:

> The way of making this commercially is to provide prefab models and you would use the same model over and over again. Thus the model has to be extremely flexible. … I'm an advocate of tweaking the designer as an intelligent person and giving them flexibility and allowing freedom

> for unvarying to the idea of standards for this … kind of open frameworks and defining an interface for procedural objects.
>
> (Andy Farnell)

Instead of either applying tools that supports only automated models, flexible self-build models or pre-built procedural models, one could instead think of more flexible and multi-purpose tools that incorporate all three approaches:

> I think you have to acknowledge that there are two different kinds of people … there are those people that like to do Reactor (Native instruments official website) sounds and then there are people who like to use them – and I think there is even something in the middle. Maybe we could say that there are people who like to build the low-level models in Reactor, and then there are people who like to take these models and combine them into some nice synths and then there are the people who like to use them. … I think if you want procedural audio to succeed then you have to create tools that enable all these three different people to collaborate.
>
> (Jan Murguc)

As a final and very important point mentioned by one of those audio programmers that were interviewed, procedural audio tools could also be thought of as sound design tools capable of doing other things than just simulating natural events in the games. If the sound designers are equipped with new procedural audio tools, such as real-time semi-modular systems and similar, one could additionally imagine that creative and unexpected sound effects could occur in the future. This would not necessarily have to be natural sounds or events, but possibly just real-time synthesized sounds creating various atmospheres and sound effects in the game:

> It doesn't even have to be natural sounds only … you could even use it to mix in with the music and make some strange rumble sounds or stuff that subconsciously adds to the game. The cool thing about it is of course that it is much more flexible than sampled sound … that you have this whole continuum of parameters that can sweep and adapt to what's happening in the game.
>
> (Jan Murguc)

MIXING PROCEDURAL AUDIO AND SAMPLE-BASED SOUND?

As the sound quality was often mentioned as being one of the biggest obstacles for the use of procedural models in comparison to sample-based sound, it seems obvious to simply combine the two techniques. By doing so one could keep the small details from the crafted sound effects and exploit the interactivity from the procedural models. This could be done by combining the techniques or by using procedural textures on top of more static samples:

> I haven't heard that many examples of procedural sounds like individual sort of procedurally generated sounds sounding up to the level of recorded sounds personally. But I think the marriage of the two could work really beautifully … as soon as you get to those high levels of details maybe they need a bit of help from real textures.
>
> (Gino Bollaert)

> I think at some point it's gonna have to be some kind of re-consideration between sample-based and procedural … maybe we can replace white noise with a sample and then see what we can do with it.
>
> (Benoit Alary)

> I think that's really the key to be able to combine samples and effects and generative audio all together to create a really convincing but emotional convincing result.
>
> (Leonard Paul)

Technically this could be implemented by using semi-procedural techniques such as granular synthesis as seen in Paul (2008, 2011) or by applying a procedural model in parallel with a sample-based technique as seen in Böttcher (2013).

CONCLUSION

For various reasons, procedural audio is not a very common approach to sound design in contemporary commercial computer games. The purpose of this article was to obtain and summarize some of the opinions held by sound designers, audio programmers and software developers in regard to procedural audio for computer games. Those interviewed discussed issues such as poor sound quality, lack of adequate tools integrated with the workflow and little or no economic motivation for investing in research and development in this field. Additionally, procedural sound design offers another approach to design when comparing it to more traditional film school sound design approaches. This could influence some more political discussions and obstacles related to a change in mind-set by the sound designers and audio programmers.

In order to develop more useful tools and models in the future it is recommended that game companies, academia and software developers improve their communication between each other. As procedural models and tools are still in an early stage of development, it is advisable to seek inspiration from the many music production tools that are available today. It is by now more common that different software from different developers is integrated or capable of communicating in one way or another, as seen with modular synthesizers, visual programming languages, sequencers and digital audio workstations, among others. It seems to be the aim of audio middleware companies to function as a framework or platform integrating other applications and plug-ins developed by other third-party companies.

Finally, it is advisable to consider the different types of sound designers when developing tools for procedural audio. As with musicians, there are several different types of sound designers. On one side you find the technical ones, who are interested in investigating synthesizers and new techniques. On the other side you find the aesthetic designers, the good recording artists or the ones who are just very busy and have to produce a large amount of sounds in a short amount of time. It would be advisable to consider these different approaches when creating the procedural tools so that one could have the freedom to choose between pre-made procedural models, automated tools or highly flexible tools.

Cases do exist where sound designers using procedural techniques and sound designers favouring sampling-based techniques are separated into two different groups, almost like two different 'religions'. Instead of choosing

either this or the other approach, it seems beneficial for both parties to combine their techniques and knowledge. This could potentially improve the interaction and variation of the pre-recorded audio, while also enhancing the sound quality of the procedural models.

REFERENCES

Ableton official website, https://www.ableton.com. Accessed 18 August 2013.

Ableton official website for M4L, https://www.ableton.com/en/live/max-for-live/. Accessed 18 August 2013.

Apple official website, http://www.apple.com/logicpro/. Accessed 18 August 2013.

AudioGaming official website, http://www.audiogaming.net. Accessed 18 August 2013.

AudioKinetics official website, https://www.audiokinetic.com/en/products/208-wwise. Accessed 18 August 2013.

AudioKinetics official website for SoundSeed, http://www.audiokinetic.com/en/products/210-soundseed. Accessed 18 August 2013.

Avid official website, http://www.avid.com/DE/products/family/pro-tools. Accessed 18 August 2013.

Böttcher, N. (2013), 'Can interactive procedural audio affect the motorical behavior of players in computer games with motion controllers?', *AES 49th International Conference: Audio for Games*, 6–8 February, London, UK (Paper 2–1).

Böttcher, N. and Serafin, S. (2013), 'A review of interactive sound in computer games: Can sound affect the motorical behaviour of a player?', in Karen Collins, Holly Tessler and Bill Kapralos (eds), *The Oxford Handbook of Interactive Audio*, Oxford University Press, New York, USA, (in print).

Collins, K. (2008), *Game Sound – An Introduction to the History, Theory and Practice of Video Game Music and Sound Design*, Cambridge: The MIT Press.

Cubase official website, http://www.cubase.net/. Accessed 18 August 2013.

Cycling74 official website, http://www.cycling74.com/products/max. Accessed 18 August 2013.

Dobashi, Y., Yamamoto, T. and Nishita, T. (2003), 'Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics', *ACM Transactions on Graphics, Vol. 22, Issue 3, July 2003, Proceedings of ACM SIGGRAPH 2003*, ACM SIGGRAPH, New York, USA, pp. 732–40.

Farnell, A. (2007), 'An introduction to procedural audio and its application in computer games', *Audio Mostly 2007*, 2nd Conference on Interaction with Sound, 27–28 September, Röntgenbau, Ilmenau, Germany.

—— (2010), *Designing Sound*, Cambridge: The MIT Press.

Filmsound.org website, http://filmsound.org/terminology/foley.htm. Accessed 18 August 2013.

FMOD official website, http://www.fmod.com. Accessed 18 August 2013.

Gatheral.co.uk website, http://www.gatheral.co.uk/blogs/udk/. Accessed 18 August 2013.

IO interactive official website, http://www.ioi.dk. Accessed 18 August 2013.

Izotope official website, http://izotope.com/. Accessed 18 August 2013.

Konami official website, http://www.konami.com. Accessed 18 August 2013.

Lloyd, D. B., Raghuvanshi, N. and Govindaraju, N. K. (2011), 'Sound synthesis for impact sounds in video games', *I3D '11 Symposium on Interactive 3D*

Graphics and Games, San Francisco, CA, USA, 18–20 February, ACM, pp. 55–62.
Native instruments official website, http://www.native-instruments.com/#/en/products/producer/reaktor-5/. Accessed 18 August 2013.
Nintendo official website, http://www.nintendo.com/wii. Accessed 18 August 2013.
Paul, L. J. (2008), 'An introduction to granular synthesis in video games', in Karen Collins (ed.), '*From Pac-Man to Pop Music*', London: Ashgate Publishing, pp. 135–152.
—— (2011), 'Granulation of sound in video games', *AES 41st International Conference: Audio for Games*, London, UK, 2–4 February, AES: New Yok, USA.
Playstation official website, http://us.playstation.com/ps3/playstation-move/. Accessed 18 August 2013.
Pure data official website, http://www.puredata.info. Accessed 18 August 2013.
Raghuvanshi, N., Lauterbach, C., Chandak, A., Manocha, D. and Lin, M. C. (2007), 'Real-time sound synthesis and propagation for games', *Communications of the ACM*, 50: 7, pp. 66–73.
Resounding website, http://re-sounding.com/2012/05/28/meltdown/. Accessed 18 August 2013.
Roads, C. (1996), *The Computer Music Tutorial*, Cambridge: The MIT Press.
SuperCollider official website, http://www.supercollider.sourceforge.net. Accessed 18 August 2013.
Tsugo studio official website, http://www.tsugi-studio.com. Accessed 18 August 2013.
Unity3d official website, http://www.unity3d.com. Accessed 18 August 2013.
Veneri, O., Gros, S. and Natkin, S. (2008), 'Procedural audio for games using GAF', *VGN08 Conference Proceedings*, Paris, January.
Verron, C., Aramaki, M., Kronland-Martinet, R. and Pallone, G. (2010), 'A 3D immersive synthesiser for environmental sounds', *IEEE Transactions on Audio, Speech and Language Processing*, 18: 6, pp. 1550–61.
Xbox official website, http://www.xbox.com/en-US/kinect. Accessed 18 August 2013.

SUGGESTED CITATION

Böttcher, N. (2013), 'Current problems and future possibilities of procedural audio in computer games', *Journal of Gaming & Virtual Worlds* 5: 3, pp. 215–234, doi: 10.1386/jgvw.5.3.215_1

CONTRIBUTOR DETAILS

Niels Böttcher (1977) holds a M.Sc. in Medialogy from Aalborg University in Copenhagen at the institute of Architecture, Design and Media Technology.

At the time of publishing this article he was finishing his Ph.D. within the topic of procedural audio for computer games, with a special focus on motion controllers.

Niels has an on-going interest in the relationship between gesture and sound in musical interfaces, controllers, computer games and rehabilitation applications.

In 2002 he founded the record label Jenka Music, which today has more than sixteen international releases, including artists such as Analogik,

Band Ane and Karsten Pflum. Niels has many years of experience as a musician, actor and performer on TV shows, European music venues and festivals with different music and entertainment projects during the last twenty years.

Currently he is working as a lecturer at the Sonic College in Haderslev in Denmark, giving courses in interaction design and sonification as well as dynamic sound design for computer games and related applications.

Contact: Aalborg University Copenhagen, A.C. Meyers avenge 15, 2450 Copenhagen SV, Denmark.
E-mail: ilz@jenkamusic.dk