

Research Article

Procedural Audio in Computer Games Using Motion Controllers: An Evaluation on the Effect and Perception

Niels Böttcher,¹ Héctor P. Martínez,² and Stefania Serafin¹

¹ *Department of Architecture, Design & Media Technology, Aalborg University Copenhagen, A.C. Meyers Vænge 15, 2450 Copenhagen SV, Denmark*

² *Institute of Digital Games, University of Malta, Ferris Street, Msida MSD 1751, Malta*

Correspondence should be addressed to Niels Böttcher; ilz@jenkamusic.dk

Received 5 June 2013; Revised 30 September 2013; Accepted 21 October 2013

Academic Editor: Alexander Pasko

Copyright © 2013 Niels Böttcher et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A study has been conducted into whether the use of procedural audio affects players in computer games using motion controllers. It was investigated whether or not (1) players perceive a difference between detailed and interactive procedural audio and prerecorded audio, (2) the use of procedural audio affects their motor-behavior, and (3) procedural audio affects their perception of control. Three experimental surveys were devised, two consisting of game sessions and the third consisting of watching videos of gameplay. A skiing game controlled by a Nintendo Wii balance board and a sword-fighting game controlled by a Wii remote were implemented with two versions of sound, one sample based and the other procedural based. The procedural models were designed using a perceptual approach and by alternative combinations of well-known synthesis techniques. The experimental results showed that, when being actively involved in playing or purely observing a video recording of a game, the majority of participants did not notice any difference in sound. Additionally, it was not possible to show that the use of procedural audio caused any consistent change in the motor behavior. In the skiing experiment, a portion of players perceived the control of the procedural version as being more sensitive.

1. Introduction

Sound design in computer games has been going through a major developmental phase since the introduction of the early game consoles in the 1980s. Especially for the larger game productions, the quality of sound design is in many ways now at the same level as in a large Hollywood film production. (One good example of this could be the game *Battlefield 3* (EA Games).) There are many reasons for this improvement in quality, including larger budgets and the evolution of consoles and computer hardware. The introduction of the digital sound sampling technique ([1], page 9) in game consoles was also another important reason for the improved sound quality. By utilizing the sound sampling technique, the sound designers were now able to digitally play back prerecorded audio and thereby employ the same kind of well-processed sound effects as one could hear in a film production. Before this point, the implementation of sound

in computer games was based on complex coding and direct control of audio chips [2].

Today most contemporary computer games solely utilize prerecorded audio (if one excludes the use of voice communication typically used in the larger massive multiplayer online role-playing games). The possibilities of controlling and manipulating sample-based audio in computer games in real time are rather limited. With the game audio middleware that is currently available (FMOD (<http://www.fmod.com/>) and Wwise (<http://www.audiokinetic.com/>) are currently the most well-known commercial audio middleware solutions) one can mainly change the playback speed, amplitude, or panning of a sample, as well as applying different effects and filters to the prerecorded audio.

In contemporary music production software, several solutions for manipulating samples at a deeper level are available. In this context, a deeper level refers to examples where sophisticated audio analysis is applied in real

time and used for manipulating the spectral content, length, and so forth of the original sound source. Examples of this could include software such as Melodyne (<http://www.celemony.com/cms/>), Metasynth (<http://www.uisoftware.com/MetaSynth/>), and many other solutions. This type of complex sample manipulation is currently not possible in commercial audio middleware software, most likely because of its significant use of computer processing power.

One of the main concerns in relation to sound effects in computer games is to avoid perceived repetition. As a result, much time and effort is invested in applying randomization and postprocessing to the samples. Often many different samples are produced for the same sound effect, and then different real-time processing is applied to the prerecorded audio.

Along with the increasingly dynamic and open game worlds, as well as the use of physics in the game engines, an almost unlimited amount of variation in the animations and graphics is now being created. In relation to this, prerecorded audio has some clear limitations, as the sound designers constantly have to think about creative approaches for avoiding repetition by applying filters, effects, and randomization of pitch or amplitude and by recording several variations of the same sound effect. All of this is extremely time consuming and could be done dynamically. In the visual components of a game, large parts of the dynamics are performed in real time by the physics engine or by AI algorithms. For this purpose, the graphic artists or animators do not have to create a lot of different prerendered randomizations concerning how particles collide and objects bounce, roll, and so forth.

Another example where sample-based audio has some obvious limitations is in the case of games using 3D motion controllers, such as the Nintendo Wii, Microsoft Kinect, Sony PlayStation Move, and the similar. When using such controllers one has access to real-time 3D motion data, which can be mapped directly to control the graphics in the game and thereby generate real-time motion-controlled variations. By using prerecorded audio, it is very difficult to exploit the detailed 3D continuous motion data such as acceleration, velocity, and rotation, as many details in the sound are predetermined by the prerendered sample. Especially in the example with the motion controllers, the predetermined length of the sample is a problem, one that cannot be solved by applying a filter to the sample.

An alternative approach to sound design is procedural audio. Farnell [3] defines procedural audio as

“nonlinear, often synthetic sound, created in real time according to a set of grammatical rules and live input.”

In this article and in relation to a computer game, procedural audio should be understood as being mainly synthesized sound, which is generated in real time by using directly the data from the game engine. Procedural audio is mostly relevant in games where a large part of the content is nonlinear. In this case, a nonlinear game should be understood as a game without prerendered animations and one that includes many unpredictable choices, movements, and similar. This is different from a movie or a cut scene

in a game, where everything is prerendered and where the narratives are fixed. Nonlinear content could also refer to the use of physics and generative content or the use of motion controllers where one can never predict the motion, speed, direction, length, or the number of game objects and their behavior.

There are many good reasons for utilizing procedural audio. As procedural audio is sound synthesized in real time, based on input from real-time game engine variables, this technique is considerably more flexible and dynamic than purely sample-based sound design. By using procedural audio in parts of a game, one could save a considerable amount of RAM, as less data has to be kept in memory. One could argue that modern game consoles have a large amount of RAM, but here it must be considered that less than 10% of the total CPU and RAM are normally allocated for sound ([4], page 82), ([5], page 9). The rest is normally kept for graphics, animations, AI, and other purposes. On the other hand, procedural audio will increase the usage of the CPU, depending on which algorithms are used, and this is of course something that the developer has to consider.

One of the main arguments still for using procedural audio is that one can avoid repetitive sound effects and create an almost unlimited number of variations of the sound effects directly linked to the actions in the game. For an indepth review and discussion of procedural audio and its possibilities in computer games, the reader is referred to [3, 5, 6].

As procedural audio is actually not a new phenomenon, if one considers the early 1980s game consoles, where the sound was purely synthesized and generated in real time, one can question why this approach to sound design is not being utilized more frequently in contemporary games. Böttcher [5] conducted a series of interviews with various sound designers, audio programmers, and audio middleware software developers, and, among other things, that article discusses some of the reasons for procedural audio not being implemented in commercial computer games today. There appear to be many reasons for this, but some of the most important factors tend to be the lack of tools for implementing procedural audio, as well as a common belief that procedural audio does not have as good sound quality.

Very few available tools exist for producing procedural audio for games. PSAI (<http://www.homeofpsai.com/>) is one such tool that has been developed for producing procedural music out of procedural sample-based segments of music, but to the authors' knowledge this tool has been so far not seen much commercial success.

As an example of a tool utilizing analysis and resynthesis, Sony has an in-house developed system named SPARK (Sony Procedural Audio Realtime Kernel). The idea behind this tool, which was developed by Nicolas Fournel, is that a sound designer can load in a prerecorded sample and automatically resynthesize a procedural sound model based on this sample, where variations of textures and parameters can be set in real time.

When discussing tools for producing procedural audio it is also relevant to mention the French company Audio-Gaming (<http://www.audiogaming.net/>). The people behind

AudioGaming are focusing on developing tools for procedural sound effects. Among other things, the company has developed premade procedural models as plugins for middleware solutions. The sound models include amongst other things a weather simulation, footstep simulations, engine simulations, and more.

Recently the audio tool called Fabric was released by Tazman-Audio (<http://www.tazman-audio.co.uk/>). This audio toolset is developed for the Unity3D game engine and consists of a set of graphical user interfaces and various audio features similar to what one would find in a standard game audio middleware solution. A new feature of this tool is that it also includes a prototype of a modular synthesizer that is capable of synthesizing audio in real time.

Until now very little attention has been given to evaluating the effect of utilizing procedural audio in computer games. Evaluations on the use of music, sound effects, speaker systems, and similar have been performed by Grimshaw et al. [7] and Nacke et al. [8], but mainly in relation to immersion of the players or emotional responses to sound, and not by the use of procedural audio.

As of today, most of the work related to procedural audio has been focusing on the design and implementation of the algorithms. The present paper studies whether the use of procedural audio has a conscious or subconscious effect on the players in computer games using motion controllers. Three main topics are investigated.

- (1) Can one affect the motor behavior of the players by using procedural audio compared to traditional sample-based audio?
- (2) Do the players notice the difference between the more detailed real-time-generated procedural audio and the less interactive sample-based audio, while being actively involved with the game and all its other elements?
- (3) Does the procedural audio have any effect on their perception of how they are controlling the game?

There are many reasons for observing the motor behavior of players. Collins [9, 10] discusses the possible influence of gestural interaction with sound and music in games as a way of increasing empathy and emotion in computer games. Movement and bodily engagement is often closely connected to immersion and engagement in games or applications. The aim of those two articles was not to explain the underlying reasons behind a possible change in gestures or motor behavior but purely to observe if the use of procedural audio could potentially affect the motor behavior of the players.

Within the field of computer music and computer music interfaces, a large amount of work has been undertaken into analyzing gestures and in developing interfaces as well as interactive sound models that support the human gesture. One could refer to conferences such as NIME (<http://www.nime.org/>), ICMC (<http://icmc2013.com.au/>), and SMC (<http://smcnetwork.org/>) where much important and related work has been presented. This not only includes in-depth analyses of musical gestures and mapping strategies, but also real-time sound synthesis for music performances

using synthesis techniques such as physical modeling among other things. This work is in many ways relevant when developing procedural audio for computer games, even though there are some different limitations when working with computer games.

In previous work, Böttcher [11] investigated the effect of procedural sound on motor behavior in an experiment on a sword game. The focus of that article was to understand if the use of procedural audio on self-produced swing sounds in a custom-made sword-fighting game, controlled by a Nintendo Wii remote, could potentially affect the motor behavior of the players. The conclusion presented there indicated that procedural audio could have an influence on the variance of the physical movements and also that further experiments had to be conducted. The present paper aims to validate those results, as well as investigating if the inclusion of collision sounds to the procedural model would enhance the influence on the movements. In addition, the former experiment showed that the majority of the experiment participants (32 out of 40) did not even notice the difference in sound between the procedural and sample-based versions. The present paper aims to further investigate if this is still the case when collision sounds are included in the audio model for the sword-fighting game. The potential effect of procedural audio is also studied in another type of computer game.

The remainder of this paper is composed of five different parts. Section 2 describes similar related work. In Section 3 the method and test protocol used in each of the devised experiments are presented. Section 4 explains the results of the experiments that were conducted. This is followed in Section 5 by a description of the designed and implemented procedural sound models. Section 6 comprises a methodological discussion in relation to future experiments on the effect of sound in computer games, followed by the conclusion in Section 7.

2. Related Work

Two sets of experiments where the role of sound is investigated in relation to the immersion of the player in a first-person shooter game are described in [7, 8]. In particular, those experiments investigate the effects of using sound and/or music on levels of arousal and valence and the bodily reactions of game players (e.g., eye movements and skin conductance). In both sets of experiments it was not possible to show any interaction effect of the sound on the measured physiological data, but only in the responses to a game experience questionnaire.

Shilling et al. [12] measured player physiological responses in a war simulation game. These responses were measured in different situations, such as with sound, without sound, and with different speakers. In this experiment the results indicated that the use of sound and surround-sound speaker systems could influence physiological responses such as heart rate, temperature, and electrodermal response (EDR).

Work has been carried out on evaluating the effect of physically modeled sound effects on the feeling of presence

in naturalistic virtual reality applications [13]. Those results cannot be carried over to a computer game directly. In most cases computer games do not aim for a naturalistic realism but instead aim for what one could refer to as computer game realism [4].

Böttcher and Serafin [14] performed an experiment on an audio-only game using different sound synthesis techniques for generating sword sounds controlled by a Nintendo Wii remote. Without this being the focus of the article, it was noted that the gestures of the players were influenced by the synthesis techniques used in the game. These results cannot be directly carried over to a visual computer game, as it is obvious that an audio-only game will force the player to react to the sounds in a game. When multimodal interaction comes into play, it is most likely that people will react differently and that the visual modality will influence the perception of sound.

The effect of sound on gestures or movement in games with motion controllers has been given very little attention until now. In other research areas such as electronic music instruments, computer music, and systematic musicology, many experiments have been evaluating or analyzing the movements or gestures of musicians [15–17]. Most of this work has been performed in order to understand the needs of musicians in order to better design electronic instruments or in order to assist musicians to improve their skills. Similarly, work has been carried out on applications using sound to enhance the performance for sports [18, 19].

3. Method

In order to assess the effect of procedural audio on players, we devised three experimental surveys, two consisting of a game session and the third consisting of watching gameplay videos. While the game surveys serve to test the effect of procedural audio in a more ecological experiment, the video surveys allow us to investigate the user perception of procedural audio by minimizing differences due to other elements of the gameplay experience. Each of the three experiments are described in the following sections.

3.1. Experiment 1. As a continuation of the experiment described in [11], it was decided to improve the procedural sword sound model by incorporating collision sounds into the procedural model used in the game. A sound model was implemented synthesizing the swoosh sound of a sword as well as the sound of it colliding with the enemy, his shield, and his sword. All sound synthesis was performed in real time, according to the speed and acceleration of the Wii remote.

The main reason for incorporating impact sounds into the procedural model was based on the hypothesis that the player would use the impact sounds in a more functional way. In comparison to the swing sounds of the sword, the collision sounds could provide subtle information about the actions being performed in the game. This could include information about how hard and where one has hit the enemy, and therefore those specific sounds could serve as useful



FIGURE 1: A screen snapshot from the sword game.

information for the player in order to help them become more successful in the game.

The main purpose of the experiment was to investigate if people would perceive a difference in the sound, now that collision sounds had been applied to the model. Another purpose was to investigate if the procedural model that included collision sounds would have a stronger or different effect on the movements of players in comparison to the earlier procedural model that had been tested.

The Sword-Fighting Game. The game was a sword game using a first-person perspective, implemented in the Unity3D game engine (<http://www.unity3d.com>). In the game the player has to defeat a computer-controlled opponent by attacking the opponent with a sword and defending himself (see Figure 1). The player can trigger different attack and defense moves by moving a Nintendo Wii controller in different directions. Further details about the design of the sword game can be found in [11].

For the experiments described here, two variants of the game were created: one using procedural audio on the swing and collision sounds and one using prerecorded sample-based audio with applied randomizations.

Test Protocol of Experiment 1. At the beginning of the experiment, the participants were introduced to the game and its controls. They were then asked to play a test round of the game in order to get familiar with the game and its controls. After becoming familiar with the game, the subjects were asked to play the game twice. In randomized order between the subjects, the experiment participants now played each version of the game.

When they had finished playing, the participants were asked to fill out a questionnaire, which not only contained mainly demographic questions but also included an open-ended question asking if they had noticed any difference between the two versions of the game.

Each individual test lasted 10–15 minutes and a typical game took 1.5–4 minutes to complete. The test participants were all filmed during the experiment with a small camera and the 3D motion data from the Nintendo Wii remote (i.e., the acceleration and velocity in each 3D axis) was logged every 50 milliseconds while they were playing.

3.2. Experiment 2. In the second experiment, the main purpose was to investigate to what extent active involvement in the game affects a participant's ability to perceive the difference in sound. The results of the earlier experiments had surprised the authors, and it was hypothesized that the influence of animations, AI, gameplay, kinematic interaction, and other factors was stronger than the audio feedback—especially in the learning phase of the game. Because of that it was decided to simply compare two different video recordings of the above-described sword-fighting game and to investigate if more subjects would perceive the difference in sound when not being actively involved in the game.

Test Protocol of Experiment 2. Two videos of approximately one-minute recordings of two different playthroughs of the game were compared. The playthroughs were recorded with the intention of playing the two games as similarly as possible. However, because it was two different playthroughs of the game, there were smaller natural differences between the two videos. One version of the game utilized sample-based audio and the other one used procedural audio.

The test participants were asked to observe the videos for any noticeable difference between the two games. They were also told that the two games were played differently, and that this was not the difference they should observe. The test was run on a 17" MacBook Pro where the test participants were placed in front of the screen and asked to look at the videos. The two videos were run in randomized order and the sound was played through a set of Beyer Dynamic DT 770 Pro headphones.

At first the test participants were asked to see one video after another in its entirety, and after this they were asked the following question:

Did you notice any difference between video 1 and video 2? If so, please explain.

In the event that the test participants did not notice any difference in relation to the sound, they were asked to once again look at the two videos and this time to focus on the sound only. After this they were asked to answer exactly the same question as before.

3.3. Experiment 3. In order to investigate if the results of the experiments would be limited to the specific game type or controller, it was decided to perform a similar experiment on a different type of game utilizing an alternative controller. This experiment was carried out on a third-person skiing game controlled by a Nintendo Wii balance board. It is clear that by testing using these two different game types one cannot get a general conclusion that will cover all computer game genres. However, as the two games have different motion controllers, a very different mode of gameplay, and also different perspectives (first person as well as third person), it would be possible to indicate if the results were related to the specific game type and controller.

It is likely that the camera perspective from the player's point of view could have an influence on the player's immersion and thereby also their perception of different modalities in the game. This could also have an influence on how the

players perceive the control in the game and, furthermore, how they behave physically when playing a computer game. This issue was not considered in the described experiments but is an interesting topic for further research.

The primary intention with this experiment was to test if the participants would perceive a difference in the sound between a game using sample-based audio and a game using more detailed interactive procedural audio. Apart from this, it was also the intention to test if the use of procedural audio could potentially cause a change in the subject's perception of control in the game.

The initial idea was additionally to compare the gestures of the players and test if the sound could influence the motor behavior in this case. Very early in the experiment it was clear that the design of the gameplay and the use of the controller did not encourage the players to perform with a great deal of variation or to be especially expressive in the game. In fact, the experiment participants were trying hard to perform in a consistent and stable manner, more so than in a varied manner, when becoming better at playing the game. Because of that it was decided to ignore the motion data logged from the balance board.

The Skiing Game. For this experiment a third-person single-level skiing slalom game controlled by the Nintendo Wii balance board (<http://www.nintendo.co.uk/Wii/Accessories/Accessories-Wii-Nintendo-UK-626430.html>) was implemented. The main purpose of the game was to get to the bottom of a hill as fast as possible, while skiing between as many gates as possible (see Figure 2). One version of the game was designed with procedural audio and a second one was created using sample-based audio. For the sample-based version a recording of the procedural audio was looped.

The game was again implemented in Unity3D and the sound was implemented in Max/MSP. The skiing game was originally developed at the IT University of Copenhagen with the purpose of rehabilitation therapy [20]. The version of the game tested in this experiment was a strongly modified version of the original game, including procedural audio as well as different control mappings and animations.

By leaning backwards or forwards on the balance board, the player was able to control the acceleration of the avatar, and by leaning to either left or right the player would change the angle of the avatar and thereby change the orientation.

On different areas on the skiing lane, different types of snow/ice material were applied. This was visualized by applying different textures with color overlays simulating a different level of icy snow. In total four different types of snow were applied. When the avatar was entering another type of surface this would cause a change in control of the avatar by changing the speed or ease of turning. The difference in color of the snow was kept very subtle, and it was mainly by listening to the difference in sound that the player would notice a difference in the surface.

A video showing an example of the gameplay can be found at <http://www.jenkamusic.dk/niels/PhD/videos.html>.

Test Protocol of Experiment 3. The experiment was performed using a MacBook Pro and the experiment participants were



FIGURE 2: Screen snapshot of the skiing game.

wearing a set of Beyer Dynamic DT 770 Pro headphones. The test participants were placed in front of the laptop running the skiing game. They received a short introduction to the experiment and filled out a short demographic questionnaire. The test participants were then asked to play a test game in order to become familiar with the control of the game. The next phase of the test consisted of two different parts.

Part 1. The experiment participants now had to play two games of the skiing game, version A and version B. One version included the sample-based audio and the other version included the procedural audio; apart from that the two games were completely identical. The order of game A and game B was reversed between each test participant.

After playing game A and game B, the test participants were asked to fill out a short questionnaire that had the following two questions:

Did you notice any difference in the control of the game, between the two versions of the game?

Did you notice any difference in general between the two games?

In a checkbox they could reply either “yes”, “no”, or “not sure,” followed by the possibility to explain any differences that they felt were present.

Part 2. Immediately after having filled out the questionnaire, the experiment participants were asked to play two more games, once again game A and game B. This time the order of the two games was reversed depending on which one they started with in part 1.

The reason for performing these two parts was to see if the test participants potentially would have another focus on the sound and control, once they were more trained with the interface and game play. In total each test lasted 15–20 minutes.

4. Test Results

4.1. Results of Experiment 1. The test was performed on students and staff at Aalborg University in Copenhagen. In total, 17 test participants (14 males and 3 females) took part in the experiment, ranging in age from 21 to 49 years. The mean age of the participants was 27 years (standard deviation = 6.99

years). Only 1 test participant had no prior experience with a Nintendo Wii remote. All test participants reported normal hearing.

In this experiment the purpose was to investigate whether or not there was more variation in the movements. In this case, variations should be understood as either a large variability or many sudden changes in direction, speed, acceleration, and length or similar.

One method of measuring the variability of a data set is to calculate the absolute deviations of data points from their mean value. The amount of deviation from the mean was therefore calculated using the acceleration and velocity in the X, Y, and Z axes. This was performed on all the test subjects and for both versions of the game.

In order to locate sudden changes in the displacement, speed, or acceleration of the logged motion data, it was decided to calculate the amount of relative extrema for all the logged variables. In order to find a relative extremum one must decide upon a threshold value, which describes an extreme difference from the previous value in a data set. This threshold value was based on analysis of the recorded video data and by comparing with the logged data. The threshold value was set to 0.045 with the total range of the acceleration sensor data ranging from 0.0 to 0.99. If a change between a logged value and the previous value exceeded 0.045 a relative extremum would be counted.

In total, 12 different variables were found for each version of the game:

- (i) mean differences in acceleration for each X, Y, and Z axis,
- (ii) mean differences in velocity for each angle (pitch, yaw, and roll),
- (iii) amount of relative extrema in acceleration for each X, Y, and Z axis,
- (iv) amount of relative extrema in velocity for each angle (pitch, yaw, and roll).

In the end, the amount of mean differences as well as the number of relative extrema was compared between the sample-based version and the procedural audio version. As the time spent on each game was different between the games, as well as between the test subjects, it was decided to only look at the first 1.5 minutes of the recorded data. This was the duration of the shortest game recorded in this experiment.

As was to be expected, the extracted features are highly dependent on each user, so it is not possible to analyze the effects of each version by calculating the difference in average values across participants (as used in traditional statistical tests such as Student’s *t*-test). Instead we analyzed the effect of each of the variants by searching for significant correlations between the sound version used (procedural or sample-based) and the movement features.

The correlation coefficients were obtained utilizing the procedure used by Yannakakis et al. [21]:

$$c(z) = \sum_{i=1}^{N_s} \left(\frac{z_i}{N_s} \right), \quad (1)$$

TABLE 1: Correlations and P values between the different versions of the game and the movement features. The P values are obtained from a binomial distribution with expected probability of 0.5.

Feature	Correlation	P value
AccX Rel extremas	-0.13	0.17
AccY Rel extremas	0.00	0.20
AccZ Rel extremas	-0.13	0.17
MotionVeloPitch Rel extremas	0.00	0.23
MotionVeloRoll Rel extremas	0.00	0.23
MotionVeloYaw Rel extremas	0.00	0.23
accX—Am Mean dif	0.13	0.17
accY—Am Mean dif	-0.25	0.12
accZ—Am Mean dif	-0.13	0.17
VeloPitch—Am Mean dif	0.17	0.19
VeloRoll—Am Mean dif	-0.33	0.12
VeloYaw—Am Mean dif	-0.17	0.19

TABLE 2: The postexperiment analysis results of the preceding experiment described in [11]. Here the P values are also obtained from a binomial distribution with expected probability of 0.5.

Feature	Correlation	P value
AccX Rel extremas	-0.23	0.046
AccY Rel extremas	-0.08	0.113
AccZ Rel extremas	-0.33	0.015
MotionVeloPitch Rel extremas	-0.03	0.125
MotionVeloRoll Rel extremas	-0.18	0.069
MotionVeloYaw Rel extremas	0.03	0.125
accX—Am Mean dif	-0.08	0.113
accY—Am Mean dif	-0.13	0.093
accZ—Am Mean dif	-0.08	0.113
VeloPitch—Am Mean dif	-0.08	0.113
VeloRoll—Am Mean dif	-0.13	0.093
VeloYaw—Am Mean dif	-0.18	0.069

The bold data refers to a significant measure was found in the amount of relative extrema for the accelerations in the X and Z axes.

where N_s is the total number of game pairs where the movement features were properly recorded, and $z_i = 1$ if the examined feature is higher in the game with sample sound, and $z_i = -1$ if the examined feature is higher in the game with real-time sound.

No significant correlations were found in any of the movement indexes (see Table 1), which suggests that the use of procedural audio did not generate a consistent change of behavior in the small sample of players analyzed.

In the postexperiment analysis described in [11], a significant measure was found in the amount of relative extrema for the accelerations in the X and Z axes (see Table 2).

As the former experiment included 40 test participants, it is difficult to compare the two experiments. On the other hand it was clear that the addition of the collision sounds to the procedural sword model did not have any additional effect on the motor behavior. One explanation to this could be the

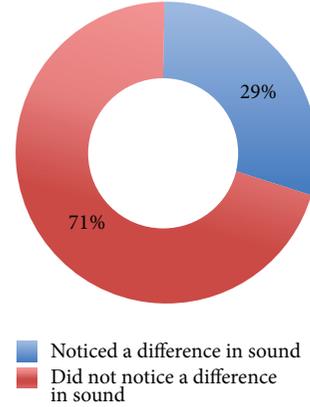


FIGURE 3: Only 5 out of the 17 test participants (29%) noticed a difference in the sound.

fact that the procedural swoosh sounds might have been less audible because of the improved collision sounds.

Besides the recorded sensor data, the test participants were asked to fill out a questionnaire after completion of the experiment. The main important question in this questionnaire was

Did you notice any difference between the two games?

(If you noticed anything please describe what and how?)

The only variable that was different between the two games was the swoosh sound of the sword and its collision with the enemy and his sword. Despite this, most of the test participants did not report any noticeable difference in the sound between the two games. As illustrated in Figure 3, only 5 out of the 17 (29%) test participants noticed a difference in the sound.

In the preceding experiment described in [11], just 8 out of 40 (20%) of the test participants noticed a difference in the sound. As with the preceding experiment, the majority of the present test participants did not seem to notice the difference in sound, even though there was more dynamic sound on both the swing sounds as well as the collision sounds involving the sword. A possible explanation for this could be that the test subjects had their focus on other aspects of the game, especially during the learning phase. Another possible reason for the test participants not noticing the sound could simply be that the difference between the procedural model and the sample-based model was not audible enough for nonaudio experts.

4.2. Results of Experiment 2. All the test participants were either students or staff from Aalborg University in Copenhagen. In total, 33 test subjects participated in the test. Out of the 33 test participants, 24 were males and 9 were females. The mean age was 25.5 years (standard deviation = 7.03 years), with the oldest being 50 years and the youngest being 18 years old. All the test subjects reported normal hearing.

TABLE 3: Numbers of experiment participants that did or did not notice the correct difference in the sound without being told to focus on the sound.

Without instruction—did they notice a difference?	
Number of experiment participants reporting a difference in the sound between the two games	11
Number of experiment participants <i>not</i> reporting a difference in the sound between the two games	22

In order to understand if the test participants had perceived some or more of the implemented differences in the sound between the two versions of the game, an analysis of the comments was performed. As the test participants were not audio experts, they were unable to reply using correct audio terminology, and in many cases more perceptual descriptions of the sound were given.

As examples, the test participants reported things such as “the sound had more whoosh in the end,” “the sound was more intense,” “the metal sounded different,” “the sound effects sounded more dangerous,” “the sound was faster, but not so hard,” “the sound was more intense somehow,” “there were more sound effects in the second one,” “the swing moves a bit around,” “the sword is clinging more,” “the sound was a bit slow . . . felt a bit more natural in the second version somehow,” and “the sound in the second one is taking longer”.

All of the above comments are good examples of feedback that was interpreted as if the test participants had noticed one or more elements of the correct differences in the sound between the two games. All the comments from the test subjects were then clustered into two categories:

- (1) not noticing the correct difference,
- (2) noticing the correct difference.

As Table 3 shows, only 11 out of the 33 experiment participants (33%) noticed the correct difference in the sound. When comparing this to the results of experiment 1, where 29% of the test subjects noticed a difference in the sound, or the preceding test (described in [11]) where 20% of the subjects noticed a difference in sound, it does not seem to make huge difference whether or not the test participants were playing the game or just observing the videos.

After being asked to focus purely on any difference in sound between the videos, the 22 subjects who did not notice the difference in the sound in the first place were asked the same questions again. Fifteen out of those 22 participants were able to describe one or more correct differences in sound after being instructed to focus on the sound specifically.

It was rather surprising that there were still 7 of the experiment participants who were unable to hear the difference in sound between the two versions of the game (see Table 4). This could indicate that the design of the interactive parameters in the procedural model could have been made more extreme.

4.3. *Results of Experiment 3.* The test was performed mainly by second semester (second half of the first year at Danish

TABLE 4: Numbers of experiment participants that did not notice the difference in sound in the first place, that subsequently did or did not notice any difference in sound after being instructed to focus on the sound only.

With instruction—did they notice a difference?	
Number of experiment participants reporting a difference in the sound between the two games	15
Number of experiment participants <i>not</i> reporting a difference in the sound between the two games	7

TABLE 5: Numbers of experiment participants that did or did not perceive any difference in the sound between the two versions of the game after part 1.

Experiment part 1—a difference in sound?	
Number of experiment participants reporting a perceived difference in sound between the two games	7
Number of experiment participants <i>not</i> reporting a perceived difference in sound between the two games	16

TABLE 6: Numbers of experiment participants that did or did not perceive any difference in the sound between the two versions of the game after part 2.

Experiment part 2—a difference in sound?	
Number of experiment participants reporting a perceived difference in sound between the two games	11
Number of experiment participants <i>not</i> reporting a perceived difference in sound between the two games	12

universities.) students from different Information Technology education streams at Aalborg University. A total of 23 test subjects participated in the test. The mean age of the participants was 23.7 years (standard deviation = 5.07 years) with the youngest being 19 years old and the oldest being 35 years old. Only 3 of the test participants were female, the rest being male. Nine out of the 23 participants had prior experience with a Wii balance board.

One of the most interesting findings in this test was the fact that again most people did not notice a difference in the sound between the two games. Based on the knowledge from the previous experiments, where most people did not notice the difference in the sound, an effort had been put into designing the mappings of the procedural audio version in such a way that the difference in sound between the two games was likely to be very clear for an untrained ear.

Tables 5 and 6 show the numbers of test participants who reported a difference in the sound between the two games or reported changes in the game that were related to the sound. As seen in Table 6, more people noticed a difference in the sound in part 2 of the game, which could indicate that the attention of the test subjects was primarily on aspects of the game other than the sound in the first two games. When testing for the effect of sound in computer games it seems relevant to consider the learning aspect of the game before testing on the effect of the sound.

Between the two games there was no difference in how the control was designed, but it was hypothesized that the use

TABLE 7: Numbers of experiment participants reporting a change in control between the two versions of the game after part 1 in the experiment.

Experiment part 1—perceived change in control?	
Number of experiment participants reporting a change in control between the two games	8
Number of experiment participants <i>not</i> noticing a change in control between the two games	15

of procedural audio potentially could affect the perception of control in the game. This could, for instance, manifest itself in the way that the players felt that they had a more nuanced or precise control of the game.

As mentioned earlier, the test subjects were asked if they noticed any difference in the control of the game between the two versions. The results of the responses to this question turned out to be very unclear, as many of the test participants reported that they did not notice any change in the control but additionally noted perceived changes that were strongly connected to the control of the game. Examples of this, among other things, are as follows:

“I found it more easy to control the skier in the second version.” [procedural version]

“The second game felt less sensitive” [sample-based version] and he additionally mentioned, *“The second game felt much easier.”*

“The control responsiveness was better in the second version and the character seemed to move faster.” [sample-based version]

“It was a much more sensitive experience the second time.” [procedural version]

“Maybe a little bit more sensitivity in game B.” [procedural version]

“Second game was a little more stable and easier to control.” [procedural version]

Because of this the answers were grouped by what the test participants answered. If the answer could be related to the control of the game, it was concluded that they felt a difference in the control of the game, even though they replied no to this question in the questionnaire.

As one can read from the sample answers provided above, the majority of the answers related to a more precise or sensitive control of the game. Because of this, it was decided to make another cluster of test participants describing the control as being more precise or sensitive. Tables 7, 8, 9, and 10 show the results of the different clusters in part 1 as well as part 2 of the experiment.

Again there was a noticeable difference between part 1 and part 2 of the experiment. For the first part, most of the test subjects (15 out of 23) did not notice any difference in control. In part 2, the results were surprisingly different. Now 13 out of

TABLE 8: Numbers of experiment participants reporting a change in control between the two versions of the game after part 2 in the experiment.

Experiment part 2—perceived change in control?	
Number of experiment participants reporting a change in control between the two games	13
Number of experiment participants <i>not</i> noticing a change in control between the two games	10

TABLE 9: Numbers of experiment participants that described either the procedural or the sample-based version as more precise/sensitive after part 1.

Experiment part 1—difference in sensitivity?	
Number of experiment participants reporting a more positive or precise/sensitive control of the procedural version	3
Number of experiment participants reporting a more positive or precise/sensitive control of the sample-based version	4

TABLE 10: Numbers of experiment participants that described either the procedural or the sample-based version as more precise/sensitive after part 2.

Experiment part 2—difference in sensitivity?	
Number of experiment participants reporting a more positive or precise/sensitive control of the procedural version	9
Number of experiment participants reporting a more positive or precise/sensitive control of the sample-based version	1

23 test subjects had perceived a change in the control between the two games.

In the first part, 4 persons believed that the sample-based version was more precise/nuanced and 3 persons believed that the procedural version was more precise/nuanced. This could indicate a random factor in the answers or that the test participants were not fully aware of a change in control.

In the second part, the results were remarkably different. This time 9 test subjects replied that the control was more precise, nuanced, or sensitive in the procedural version, whereas just 1 had made the same observation about the sample-based version.

5. Design and Implementation of the Procedural Sound Models

5.1. The Sword Sounds. All of the sound in the sword game was implemented using the Max/MSP (<http://www.cycling74.com/products/max/>) graphical programming environment. The procedural version of the swoosh sound of the swinging sword was implemented by a combination of two granular synthesis modules and a subtractive synthesis module (see Figure 4). The reason for choosing these synthesis techniques was mainly based on prior experiments performed in [14], where among other things modal synthesis, granular synthesis and subtractive synthesis were compared for generating sword sounds.

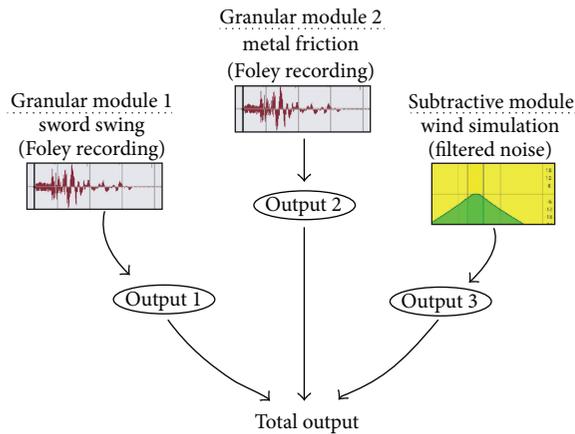


FIGURE 4: Illustration of the design of the swing swoosh sounds of the sword, combining granular synthesis with subtractive synthesis.

Subtractive synthesis is a simple and CPU-friendly synthesis technique, which can be used to generate highly dynamic and responsive aerodynamic sounds. In [14] the subtractive synthesis-based sword sounds turned out to have a good effect on the players' motor behavior. On the other hand, purely subtractive synthesis performed on white noise can, in some cases, suffer from not having as many details as what one finds in prerecorded and well-processed sound effects.

The reason for combining subtractive synthesis with granular synthesis was the fact that granular synthesis contains many rich details because it is a sample-based synthesis technique. Granular synthesis is by far more flexible compared to purely sample playback, among other things because one can change the length of the sample without changing the pitch of the original sample. Similarly one can change the pitch without changing the length of the sample.

For a more detailed description of the implemented sword sounds the reader is referred to [6, 11]. The combination of granular synthesis and subtractive synthesis in various sound models can be found at <http://www.jenkamusic.dk/niels/PhD/videos.html>.

In general, using the two above mentioned synthesis techniques in combination is a simple way of generating highly interactive sound and still maintaining the sound quality that good Foley recordings contain. In some research, granular synthesis has been referred to as dynamic sound instead of procedural sound, as this technique includes small snippets of sample-based audio. In this paper it will be referred to as procedural sound as the technique is mixed with subtractive synthesis.

For the sound model used in this paper, the acceleration of the Wii remote was among other things mapped to the scrubbing point (when playing through the different grains in granular synthesis one often uses the term scrubbing though the grains) of the sample in the granular modules as well as the timbre of the filter in the subtractive synthesis module. A more detailed description of the design and implementation of the swing swoosh sounds can be found in [11].

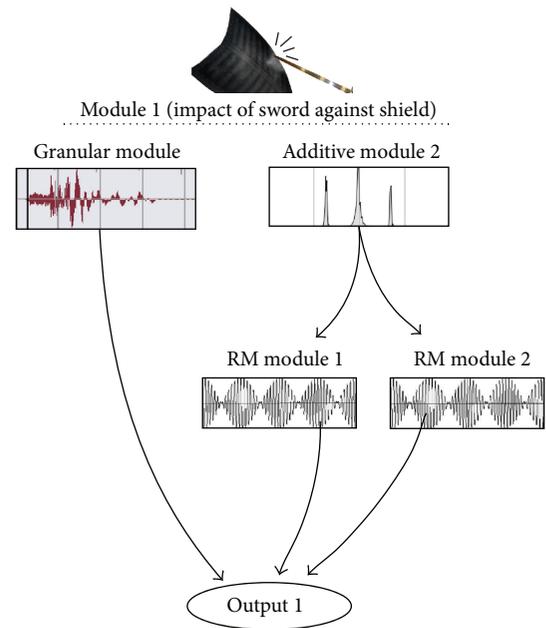


FIGURE 5: Illustration of the impact module 1 combining additive synthesis with granular and ring modulation for the sword against shield collision.

The collision sound of the sword against the shield, as well as that of the sword against the sword of the enemy, was implemented by combining granular synthesis, additive synthesis, ring modulation, and a prerecorded sample. The design approach mainly used a perceptual perspective and the aim was not to simulate correct physical behavior of the objects or situation.

In opposition to the swing sound of the sword the procedural model of the collision sounds was mostly implemented as a proof of concept regarding the use of procedural audio. For future implementation other synthesis techniques might be more appropriate to use. This is especially so because the additive synthesis module is not the most efficient technique to utilize in computer games due to the relatively large use of CPU compared to other techniques. Here resonance filters or wavetables could, for example, also have been utilized successfully to synthesize the collision sounds using less CPU power.

Two different modules were developed in order to simulate the collision sound: module 1 simulating sword against shield collision and module 2 simulating the sword against metal collision.

Module 1 (Sword Against Shield Collision). This module is depicted in Figure 5 and was implemented using a combination of granular synthesis and additive synthesis processed through two ring modulators. The main part of the model was the additive synthesis part, which was implemented with inspiration from a procedural alarm-bell sound model originally designed by Farnell and described in [22].

Two modules, each consisting of five different partial-groups (groups of sine waves), and each of the individual

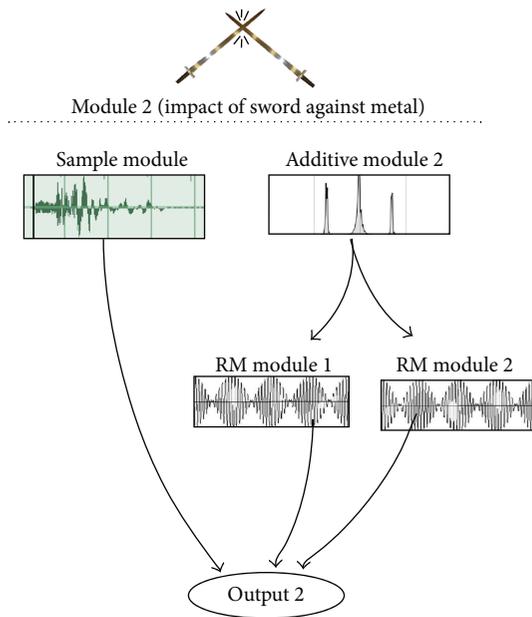


FIGURE 6: Illustration of the impact module 2 combining additive synthesis with sampling and ring modulation for the sword against metal collision.

partial-groups, including 3 oscillators and an envelope (the function describing the amplitude of the signal) were implemented. The frequency of each partial group was mapped to the strength of the acceleration at the point in time when the virtual collision occurred in the game. When the acceleration was harder the pitch of the partial groups was increased accordingly. The pitch of each oscillator was different and the exact frequency was decided based on subjective perceptual decisions. Furthermore, the frequency of the different partials was designed with a bending effect so that the frequency would fall slightly after the initial hit. The harder one would hit, the longer the range of the frequency bend would become. Additionally, the decay (the part of the envelope function describing how the amplitude is decreasing after the initial hit) of the envelope of the different partials was mapped to the strength of the hit, with the decay time being longer when the hits were harder.

In order to simulate detuning as well as the additional partials accumulated when hitting extra hard, two ring modulation modules were added. The modulating frequency of each ring modulator was designed so the effect would only be audible in the case of a strong hit. The acceleration of the Wii remote at the time of the virtual collision in the game was mapped to the frequency of the modulating frequency as well as to the amplitude of the ring modulation. Furthermore, the frequency of the modulating frequency was bent depending on how hard one would hit.

Additionally, a synchronous granular synthesis module (Granular synthesis is often divided into synchronous and asynchronous. Synchronous playback is often used when the aim is to preserve the original sound of the sample and asynchronous playback is often used when simulating random particles or when the intention to manipulate a

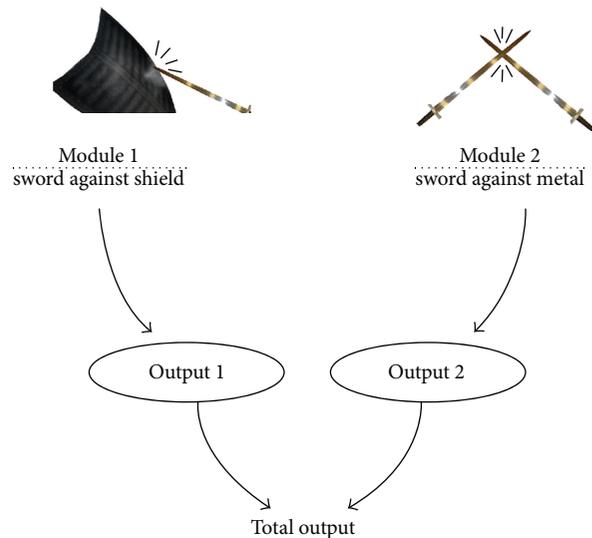


FIGURE 7: Illustration of the design of the collision sounds of the sword, combining the two modules.

sample.) was implemented where the acceleration of the Wii remote was mapped to the scrubbing point of the granular synthesis, as well as the pitch and amplitude of the granular synthesizer. The sample being scrubbed through was a custom-made sound effect simulating a collision between a sword and a shield.

Module 2 (Sword against Metal Collision). The second module, which is depicted in Figure 6, was designed very similar to module 1. Again, additive synthesis and ring modulation were the main part of the module. The frequencies and envelopes of the additive synthesis part were designed with a longer decay and higher frequencies in order to simulate the sound of metal against metal. Also in this module, ring modulation was applied to the additive synthesis to simulate especially overtones when hitting especially hard.

In order to improve the sound quality of the model, a sample of metal against metal sound was added to the model. The acceleration of the Wii remote was mapped to the amplitude of the sample playback, so that the player would only be able to perceive the sound of metal in the case of a strong hit.

For the case of a sword-against-sword collision in the game, only the second module was utilized, but in the case of a sword-against-shield collision, the two modules were combined, as seen in Figure 7.

For a more detailed description of the implementation the reader is referred to <http://www.jenkamusic.dk/niels/PhD/>, where illustrations, videos, and sound examples are provided.

5.2. The Skiing Sound Model. The skiing sound model was designed as a combination of four different modules (see Figure 8):

- (1) a random-noise burst module generating random clicks for simulating the friction between different smaller and bigger snow/ice parts and the skis,

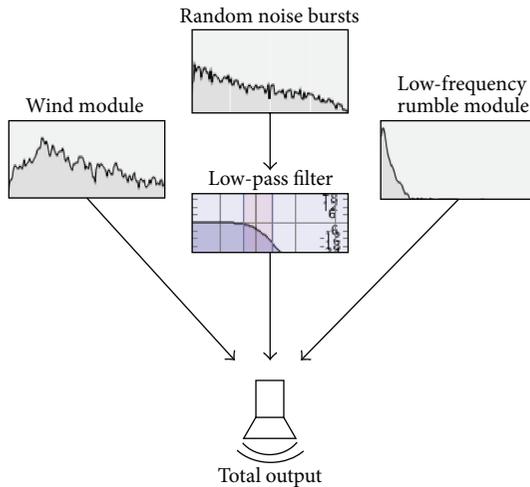


FIGURE 8: Illustration of the four different modules of the procedural skiing sound model.

- (2) a low-pass filter for simulating the speed, turning of the skis as well as entering another type of snow material,
- (3) a module simulating deeper frequent random clicks for making the overall sound more engaging,
- (4) a wind module.

The Random-Noise Burst Module Generating Particle-Like Clicks. This model was developed with great inspiration from the fireplace model presented by Farnell in [22]. In order to generate the random-noise bursts, full-wave rectified (By applying full-wave rectification to a signal, the negative parts of the waveform become positive, and the signal becomes entirely positive.) low-pass filtered noise was used as an amplitude envelope on band-pass filtered noise. The signal controlling the envelope was rectified in order to reduce the difference between the amplitude peaks. As the rectification of the signal also lowered the frequencies of the control signal, the signal was later multiplied in order to have more frequent peaks in the signal—just with a smaller difference between the amplitude of the peaks (see Figure 9 for an illustration of the signal controlling the amplitude of the band-pass filtered noise).

The result of using a control signal as described in Figure 9 was an envelope generating very natural random events similar to the crackling sounds that one could hear at a fireplace.

In order to simulate the sound of different sized snow/ice particles instead of the crackling sounds of a fire, the value of the center frequency of the band-pass filter was set between 150 and 2500 Hz and was controlled by the speed of the avatar as well as being dependent on the material of the snow. Depending on the material of the ice, the Q-factor (The Q-factor describes the bandwidth of the band-pass filter as a nondimensional parameter instead of a static bandwidth, and keeps the proportion of the frequencies passed independent of the filter's center frequency.) of the band-pass filter was

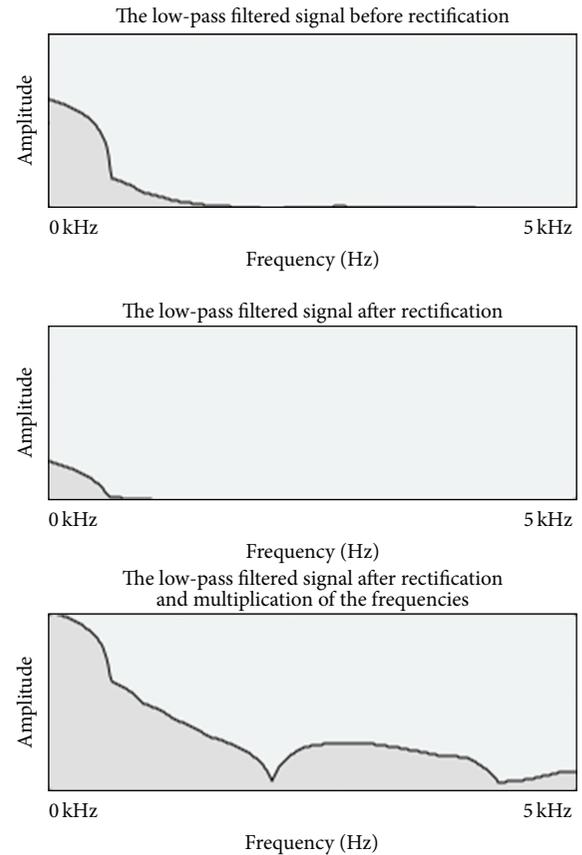


FIGURE 9: Illustration of the process of the rectification and multiplication of the low-pass filtered noise controlling the amplitude of the band-pass filtered noise for generating the natural random-noise bursts.

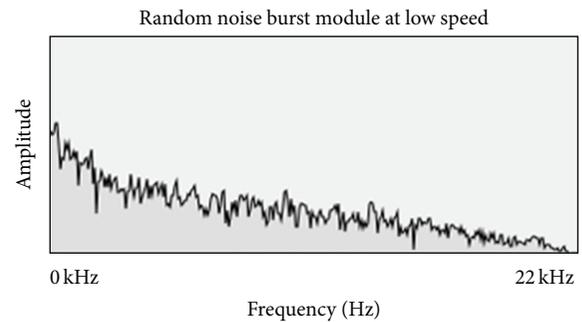


FIGURE 10: Spectrogram showing a snapshot of the random-noise bursts at low speed.

set between 0.15 and 1.65. The value of the cutoff frequency of the low-pass filter controlling the amplitude of the band-pass filtered noise was set between 2 and 15 Hz. This cutoff frequency was also dependent on the speed and the material. The faster the avatar would go, the higher the cutoff frequency of the low-pass filter was set, and therefore the noise bursts would appear more frequently and would also contain higher frequencies.

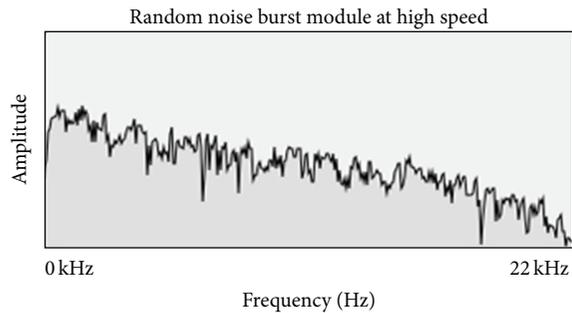


FIGURE 11: Spectrogram showing a snapshot of the random-noise bursts at high speed.

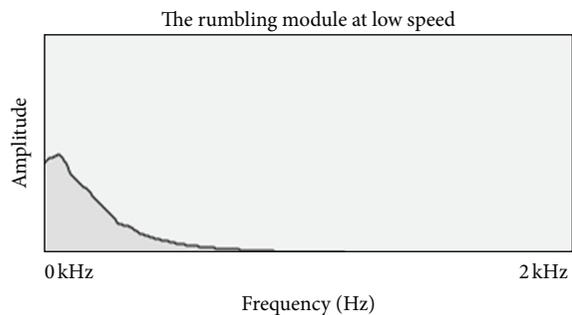


FIGURE 12: Spectrogram showing a snapshot of the rumbling module at low speed.

Figures 10 and 11 illustrate two different states of the random-noise burst module: one at high speed and one at low speed. It is apparent that there is much more high-frequency content in the sound at high speed compared to low speed, and the lower frequencies (e.g., in the first quarter of the spectrogram) are also boosted in amplitude.

Deeper Low-Frequency Rumbling Sounds. In order to give the overall sound a more intense deep low-frequency rumbling sound, as one could find in an action computer game, the deep low-frequency rumbling module was implemented. This module was designed by implementing the rolling tin can model devised by Farnell [22]. The tin can model was modulated by changing the parameters of the different band-pass filters, envelopes, oscillators, and low-pass filters so that the timbre of the rumbling became deeper and no longer sounded like a tin can.

The speed of the avatar was, among other things, mapped to the amplitude as well as the low-pass filters in the model. This was done in such a way that the deep frequent rumbling sounds would only become audible when the avatar reached high speed. Figures 12 and 13 show spectrograms of the sound produced by the rumbling module at low and high speeds. As one can see, there is almost no sound at low speed, but at high speed the amplitude is much higher and more frequencies are

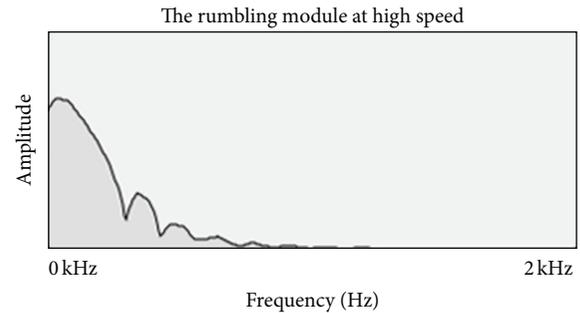


FIGURE 13: Spectrogram showing a snapshot of the rumbling module at high speed.

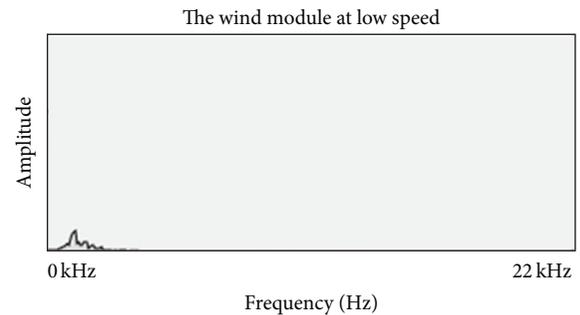


FIGURE 14: Spectrogram showing a snapshot of the wind module at low speed.

present (but still only the low frequencies in order to give the sound a low-frequency rumbling timbre).

Wind Module. A wind model was also implemented. This was a reproduction in Max/MSP of a wind model originally implemented in Pure Data (<http://www.puredata.info/>, an open source visual programming language similar to Max/MSP) by Farnell in [22]. The main components of the model were filtered noise, low frequency oscillators, and resonance filters.

The speed of the avatar was mapped to the speed of the modulation of the resonance filters in the wind model, as well as the amplitude of the wind. The faster the avatar would go, the faster the modulation would become. Additionally, the speed of the avatar was mapped to the amplitude as well as the cutoff frequency of the filters in the model. The faster the player would ski, the louder the wind sound would become and the higher in pitch the filter would go.

Figures 14 and 15 show spectrograms of the sounds from the wind module at low and high speeds. As one can see, the amplitude of the spectrogram is extremely low at low speed, meaning that the player would only be able to hear the wind-resistance sounds in the case of skiing down the hill at high speed.

Different Surfaces. Besides the speed of the avatar, the different surfaces of the snow/ice had an influence on the different parameters in each module. The random-noise-burst module and the low-pass filter module were especially influenced by

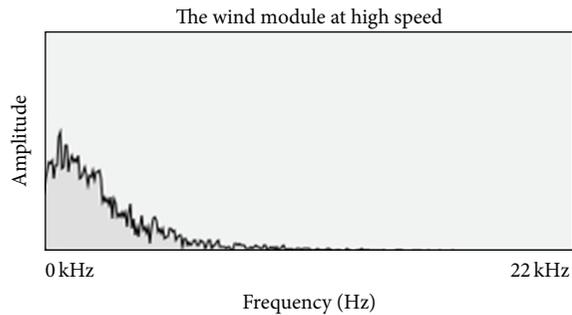


FIGURE 15: Spectrogram showing a snapshot of the wind module at high speed.

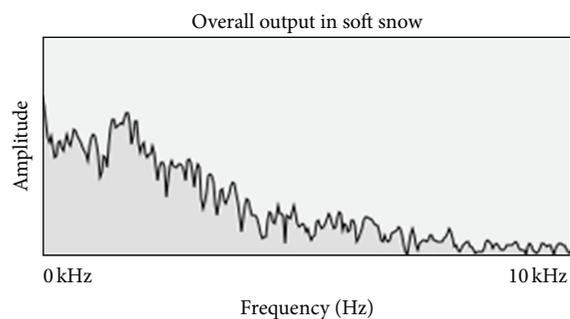


FIGURE 16: Spectrogram showing a snapshot of the overall output in very soft snow.

the changes in surface. Among other things, the frequency of the random-noise bursts was shifted upwards in conjunction with the band-pass filter, letting through more high frequency sounds whenever the surface became icier.

Figures 16 and 17 illustrate the effect of changing the material from a soft snow surface to an icy snow surface while keeping the avatar speed the same. It is apparent that the amplitude of the higher frequencies is raised in the very icy surface. Furthermore, one can also see that spikes in the spectrum are more frequent in the icy version. When the player is skiing with slow speed, this becomes audible as more frequent click sounds are generated while skiing. When the player is skiing fast, the more frequent clicks become more difficult to hear.

Turning. An additional and important feature of the skiing model was the turning of the skis. When turning the direction of the skier, several parameters were substantially affected. The overall amplitude was raised in the initial part of the turn by shifting the amplitude and afterwards applying a ramp to slowly turn the amplitude down again. In a similar way, the cutoff frequency of the low-pass filter was raised, as was the center frequency of the band-pass filter in the random noise burst module. The Q of the band-pass filter was also raised slightly.

Sample-Based Version for the Comparison. A version based on sampled sound was also implemented. A recording of the

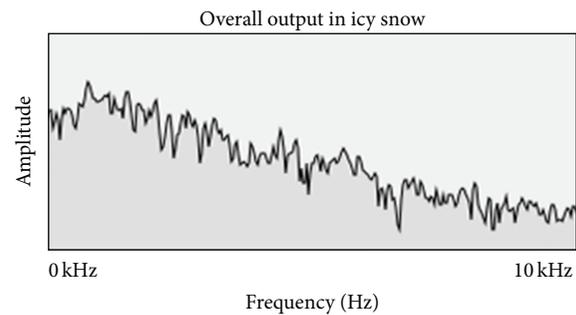


FIGURE 17: Spectrogram showing a snapshot of the overall output in very icy snow.

procedural audio was looped and the speed of the avatar was mapped to the amplitude of the sampled sound as well as the speed of the sample.

The reader is referred to <http://www.jenkamusic.dk/niels/PhD/> for a deeper look at the models, sound, and video examples.

6. Discussion

As shown in the third experiment, the attention of the test subjects seemed to be focused primarily on aspects other than the sound in the learning part of a computer game. In order to improve the initial experiments that were described and for future tests on the effect of sound in computer games, it is advisable to consider the learning curve in a computer game and its controls, particularly for games that utilize motion controllers or balance boards, as they may add another dimension to the learning curve compared to games using more traditional and less complicated controls. Even though the test subjects were given time to learn the controls and get familiar with the game, it took a few games before they were able to observe for changes between the games.

Testing the various aspects and effects of sound in a computer game is in many ways complicated. In order to be perceived as a “real” computer game, the game naturally has to include many unpredictable variables, as it has to be open to a certain degree and should be able to solve scenarios in many different ways. Longitudinal tests, such as those applied by Gelineck and Serafin [23], may be more advisable when testing a computer game, as the experiment participants may tend to focus on modalities other than sound in the initial learning phase of the game.

7. Conclusion

In this paper, the design and implementation of two different computer games utilizing procedural audio was presented. These were a first-person sword-fighting game controlled by the Nintendo Wii remote and a third-person skiing game controlled by the Nintendo Wii balance board. The procedural sound models were all based on alternative combinations and moderations of existing procedural models. Common to both models was the fact that they were not designed with

the intention of simulating correct physical events but were primarily based on perceptual parameters.

One experiment was performed on the sword-fighting game with two main purposes:

- (1) to understand if people changed their motor behavior into more varied movements when using more nuanced and interactive procedural audio, compared to sample-based audio,
- (2) to investigate if people notice the difference in sound when playing the game.

It was not possible to show that the experiment participants increased the variability in their movements or increased the amount of sudden changes in direction, speed, acceleration, length, or the similar, when utilizing the procedural sound model that included collision sounds. Furthermore, the majority of the subjects did not notice a difference in sound between the sample-based sound and the procedural-based sound version of the same game.

In a second experiment it was investigated if more people would notice the difference in sound between the two versions of the game, if they just observed two different recorded videos of the game in action, without playing the game. The second test showed that the majority of the experiment participants would still not notice the difference in sound between the two games. After being told to focus on the sound, most of the experiment subjects were able to describe and perceive the difference between the two games, but still it turned out that 21% of the subjects were unable to hear any difference between the two versions of the game.

The last experiment was performed on the skiing game. The main purpose was to investigate if people would notice a difference in the sound in a completely different type of game that also utilized a different controller. An additional purpose was to investigate if the experiment participants would perceive a change in control between the game utilizing sample-based audio and the game utilizing more nuanced and interactive procedural audio.

As a difference from the first two experiments, this third experiment consisted of 2 runs through each of the 2 games, so the learning phase of the game was incorporated in more detail. The results showed that most of the test subjects did not notice a difference in the sound. This was the case after having played the two versions of the game both for the first and second times.

Regarding the perceived control of the game, most of the test subjects also did not perceive any change in control between the two versions of the game after having played the two versions just once. But, after having played the two different versions twice, 13 out of the 23 test participants reported that they felt a change in the control of the game. Nine out of those 13 test subjects described the change in control for the game utilizing procedural audio as being more precise, sensitive, or similar.

Not much work has so far been presented in the available literature showing the perceptual, physiological, or kinematical effect of using procedural audio in computer games. The present paper has described a series of experiments evaluating different aspects of procedural audio in

two different computer games. It is not possible to draw a general conclusion from the results about the effects of using procedural audio, and in order to do so more experiments have to be carried out. It is hoped that this paper encourages more people to not only design more procedural sound models but also to make an attempt to evaluate how this approach to sound design could potentially influence the players of computer games.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge Alexander Birke for programming the major parts in the sword-fighting game, including among other things the state machine and main game mechanics.

References

- [1] C. Roads, *The Computer Music Tutorial*, MIT Press, Cambridge, Mass, USA, 1996.
- [2] K. Chang, G. Kim, and T. Kim, "Video game console audio: evolution and future trends," in *Proceedings of the Computer Graphics, Imaging and Visualisation: New Advances (CGIV '07)*, pp. 97–102, Bangkok, Thailand, August 2007.
- [3] A. Farnell, "An introduction to procedural audio and its application in computer games," in *Proceedings of the Audio Mostly Conference*, pp. 27–28, Röntgenbau, Ilmenau, Germany, September 2007.
- [4] K. Collins, *Game Sound: An Introduction to the History, Theory and Practice of Video Game Music and Sound Design*, The MIT Press, Cambridge, Mass, USA, 1 edition, 2008.
- [5] N. Böttcher, "Expert interviews about the current problems and future possibilities of procedural audio in computer games," *Journal of Gaming and Virtual Worlds*, vol. 5, no. 3, pp. 215–234, 2013.
- [6] N. Böttcher and S. Serafin, "A review of interactive sound in computer games: can sound affect the motorical behaviour of a player?" in *The Oxford Handbook of Interactive Audio*, H. Tessler and B. Kapralos, Eds., Oxford Press, Oxford, UK, 2013.
- [7] M. Grimshaw, C. A. Lindley, and L. Nacke, "Sound and immersion in the first-person shooter: mixed measurement of the player's sonic experience," in *Proceedings of the Audio Mostly Conference on Interaction with Sound (AM '08)*, pp. 9–15, Piteå, Sweden, 2008.
- [8] L. E. Nacke, M. N. Grimshaw, and C. A. Lindley, "More than a feeling: measurement of sonic user experience and psychophysiology in a first-person shooter game," *Interacting with Computers*, vol. 22, no. 5, pp. 336–343, 2010.
- [9] K. Collins, "Making gamers cry: mirror neurons and embodied interaction with game sound," in *Proceedings of the 6th Audio Mostly Conference on Interaction with Sound (AM '11)*, pp. 39–46, ACM, Coimbra, Portugal, September 2011.
- [10] K. Collins, *Playing with Sound: A Theory of Interacting with Sound and Music in Video Games*, MIT Press, Cambridge, Mass, USA, 2013.

- [11] N. Böttcher, “Can interactive procedural audio affect the motorical behaviour of players in computer games with motion controllers?” in *Proceedings of the 49th Audio Engineering Society International Conference (AES '13)*, London, UK, February, 2013.
- [12] R. Shilling, M. Zyda, and E. C. Wardynski, “Introducing emotion into military simulation and videogame design: America’s army: operations and VIRTE,” in *Proceedings of the GameOn Conference*, pp. 151–154, London, UK, November, 2002.
- [13] R. Nordahl, “Self-induced footstep sounds in virtual reality: latency, recognition, quality and presence,” in *Proceedings of the 8th Annual International Workshop on Presence (PRESENCE '05)*, pp. 353–354, London, UK, 2005.
- [14] N. Böttcher and S. Serafin, “Design and evaluation of physically inspired models of sound effects in computer games,” in *Proceedings of the 35th Audio Engineering Society International Conference: Audio for Games (AES '09)*, no. 13, pp. 11–13, London, UK, February 2009.
- [15] S. Dahl, M. Grossbach, and E. Altenmüller, “Effect of dynamic level in drumming: measurement of striking velocity, force, and sound level,” in *Proceedings of the Forum Acusticum (European Acoustics Association)*, pp. 621–624, Danish Acoustical Society, Aalborg, Denmark, 2011.
- [16] A. Hunt, M. M. Wanderley, and M. Paradis, “The importance of parameter mapping in electronic instrument design,” *Journal of New Music Research*, vol. 32, no. 4, pp. 429–440, 2003.
- [17] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, vol. 21 of *The Computer Music and Digital Audio*, A-R Editions, Middleton, Wis, USA, 2006.
- [18] N. Schaffert, S. Barrass, K. Mattes, and A. O. Effenberg, “Exploring function and aesthetics in sonifications for elite sports,” in *Proceedings of the 2nd International Conference on Music Communication Science (ICoMCS '09)*, pp. 83–86, University of Western Sydney, Sydney, Australia, December 2009.
- [19] M. Takahata, K. Shiraki, Y. Sakane, and Y. Takebayashi, “Sound feedback for powerful karate training,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '04)*, pp. 13–18, Hamamatsu, Japan, June 2004.
- [20] D. Dimovska, P. Jarnfelt, S. Selvig, and G. N. Yannakakis, “Towards procedural level generation for rehabilitation,” in *Proceedings of the Workshop on Procedural Content Generation in Games (PCGames '10)*, pp. 7.1–7.4, ACM, Monterey, Calif, USA, June 2010.
- [21] G. N. Yannakakis, H. P. Martínez, and A. Jhala, “Towards affective camera control in games,” *User Modelling and User-Adapted Interaction*, vol. 20, no. 4, pp. 313–340, 2010.
- [22] A. Farnell, *Designing Sound*, MIT Press, Cambridge, Mass, USA, 2010.
- [23] S. Gelineck and S. Serafin, “Longitudinal evaluation of the integration of digital musical instruments into existing compositional work processes,” *Journal of New Music Research*, vol. 41, no. 3, pp. 259–276, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

