First Person Shooters as Collaborative Multiprocess Instruments

Florent Berthaut
LaBRI - SCRIME
florent@hitmuri.net

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

Hironori Wakama
Kwansei Gakuin University
cuh79618@kwansei.ac.jp

Naoyuki Totani
Kwansei Gakuin University
naoyuki.totani@kwansei.ac.jp

Yuichi Sato
Kwansei Gakuin University
bih62181@kwansei.ac.jp

ABSTRACT

First Person Shooters are among the most played computer video games. They combine navigation, interaction and collaboration in 3D virtual environments using simple input devices, i.e. mouse and keyboard. In this paper, we study the possibilities brought by these games for musical interaction. We present the Couacs, a collaborative multiprocess instrument which relies on interaction techniques used in FPS together with new techniques adding the expressiveness required for musical interaction. In particular, the *Faders For All* game mode allows musicians to perform pattern-based electronic compositions.

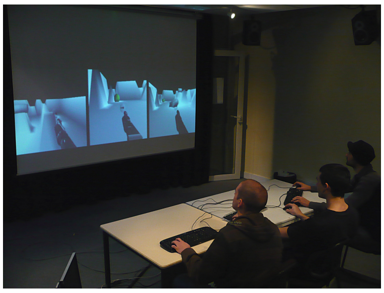

Figure 1: Three musicians playing the Couacs in *Faders For All* mode

Keywords

the couacs, fps, first person shooters, collaborative, 3D interaction, multiprocess instrument

1. INTRODUCTION

In the past years, success of musical video games has grown quickly, with games such as Rock Band, Guitar Hero, Wii Music and so on. Some of these games use or imitate musical instruments, adding evaluation of players performances. These games rely on musical interaction techniques. We believe that, in the same way, musical interaction can benefit from techniques developed for video games using devices such as keyboards, mice, gamepads, joysticks. In fact, players may develop specific skills, as described in [15], and enhance some abilities that musical instruments may build on. An especially interesting genre of video game are First Person Shooters (FPS).

These games feature rich 3D virtual environments that can be used to facilitate control and visualization of multiple sound processes.

In this paper we discuss the possibilities but also the issues brought by First Person Shooters for musical interaction. Then we present the Couacs, a collaborative multiprocess instrument which focuses on interaction between players/sound processes. A picture of three musicians playing the Couacs can be seen on figure 1, and a presentation video is available [1]. In particular, we propose and evaluate a game mode called *Faders for All*.

2. RELATED WORK

Some musical video games rely on performance evaluation. For example in Rock Band [2], the goal is to play as much correct notes as possible according to a score. They are usually played with input devices that imitate musical instruments. Other games such as Rez [3] rely on traditional gaming devices and consist in triggering predefined musical events. In these games, however, musical freedom, as defined by Jordà [14], and expressiveness are limited as explained by Marczak et al. [16]. Interestingly, although games such as *Rock band* are multiplayer, there is almost no interaction between players. Each player focuses on his own actions.

On the other hand, in the past years several laptop orchestras were created, such as the Standford Laptop Orchestra [18]. They make use of common devices used in laptop music [6], i.e. mouse and keyboard. In addition they enable interaction and musical dialog between several musicians.

Much research has been done on using common input devices for musical interaction, for example by Zadel et al. [20] or Fiebrink et al. [8]. Often 2D graphical interfaces are used in conjunction with these devices.

Some new instruments rely on first person navigation in a 3D environment. For example, the musical piece *La ménagerie imaginaire*, built upon the research done by Wozniewski et al. [19], allows musicians to apply effects on their acoustic instruments by navigating in a virtual environment.

Finally Hamilton developed several instruments and systems based on FPS. Maps and legends [9] is an instrument which makes use of navigation and sound spatialization, by superimposing the virtual environment and the concert room. Q3osc[10] is a modification of a game engine which outputs OpenSoundControl messages for every game parameter. Weapons projectiles may then be associated with sounds triggered on collisions with the environment.

Rather than sound spatialization, we are interested in interaction between musicians and control of multiple sound processes in the 3D environment. We want to take advantage of skills developed by FPS players to enable expressive musical interaction.

3. FIRST PERSON SHOOTERS

3.1 Overview

In First Person Shooters, players control 3D avatars and perceive the 3D virtual environment through their eyes with a first person

[1]http://www.vimeo.com/19347468
[2]http://www.rockband.com/international
[3]http://www.sonicteam.com/rez

perspective. The gameplay consists in navigating in the environment, usually using the keyboard, and shooting at other players with different weapons. The mouse is used to aim, change weapon and shoot. When a player gets killed, he starts again in another spot of the environment. Items such as health, shield, weapons, invincibility, or speed can be found in the environment and picked up by the avatars, usually simply by walking through them. In multiplayer FPS, the goal depends on the chosen game mode. For example, in a *Free For All* (FFA) game, the goal is to have more "frags", i.e. kills, than other players at the end of the game. In a *Capture the Flag* game, the goal is to grab the flag from the other team's camp and bring it back to one's camp. Different virtual environments, called *Maps*, are used, such as indoor maps with several rooms, terrains with hills and trees, platforms in space and so on.

3.2 Musical FPS

In this section, we study the possibilities provided by the use of First Person Shooters as musical instruments, but also the issues that it raises.

Interaction

First of all, interaction techniques developed for FPS provide several control dimensions. Navigation in the environment can be used for example as several continuous parameters with the absolute position and rotation. But it can also be interesting as a discrete parameter with movements states such as crouching, jumping, running and so on. Items can be used for discrete modulations of sound parameters, for example it may affect several sound processes at the same time. Weapons and shooting have several parameters such as weapon type or weapon mode. Finally, FPS make use of bimanuality, with one hand handling large movements using the keyboard while the other hand performs more accurate movements to aim and shoot.

On the other hand, input devices used in FPS are common and affordable but they often restrict musical freedom and expressiveness. A mouse button only outputs a 1 bit value, so that one can only control the rhythm of clicks but not their velocity. This prevents players from correctly performing instantaneous excitation gestures as described by Cadoz [5]. On the other hand, graphical actions, such as translations and rotations, provide a good spatial resolution but they can not be done with as much temporal accuracy as mouse clicks, due for example to the latency of graphical rendering. Therefore we have to provide additional degrees of freedom for gestures performed using the mouse and keyboard.

Collaboration

Shoot, touch and other interactions between avatars can be used as musical metaphors for various musical interactions between sound processes. Game modes may be a way of switching from one metaphor to another, and therefore between collaboration modes.

Visualization

As stated by Jordà [13], graphical interfaces allow for efficient interaction with several sound processes, by giving informations on their state and parameters, and by facilitating access to sound processes. FPS allow to visualize sound parameters using different 3D graphical elements. First of all, the environment may represent musical structures. Avatars may be used to display individual or combined sound processes. Projectiles fired by weapons also have several parameters that can be used to control sound parameters. Finally, graphical effects such as shading can be applied to the whole environment to represent effects applied to all sound processes or global mood of a song.

On the other hand, we need to correctly choose how to represent these sound processes in the environment in order to easily identify these processes and visualize their parameters.

Immersion

FPS provide a good sensory immersion [7] compared with other video games. This immersion is especially interesting for the players, as it may improve the implication in the instrument. But it can also be interesting for the audience as it may ease the understanding of players' actions.

Accessibility/Spreading

Compared with other 3D instruments, FPS-based instruments make use of simple and common input devices such as keyboards and mice, instead of six degrees-of-freedom tracking systems. This may facilitate the spreading of such instruments among gamers and laptop musicians, as they usually already have all the needed hardware. Furthermore, communities built around these games may also help improving these instruments by creating new game modes, new maps and organizing Local Area Network (LAN) Parties or Tournaments and even Concerts. This may partly solve the problem of most new instruments which are never played again after the first paper/concert.

Learning

Learning and gaining expertise is an important issue for new instruments. Existing FPS tournaments prove that players can improve their skills. Eventually, some players become virtuoso by mastering all game techniques and improving their accuracy and reactivity.

Game or Instrument

A final question is the balance between gaming and playing music. How can we use some game actions for musical control without disturbing other game actions not connected to sound, and vice-versa ? Will gamers/musicians try to learn how the instrument works and how they can produce specific musical results or will they only play without paying attention to the generated music ? Should these instruments have a goal like a video game or not ?

4. THE COUACS

In this section, we present the Couacs, a collaborative multiprocess instrument based on First Person Shooters. This instrument allows us to experiment the adaptation of interaction and collaboration techniques used in FPS to musical interaction. It uses Irrlicht[4] for graphical rendering, Jack[5] for sound rendering and libxtract [4] for audio features extraction.

4.1 General approach

In the Couacs, each musician controls a 3D avatar associated to one or several sound processes. Actions and characteristics of the avatars modify the sound processes, and in return the aspect of the avatars reflects properties of the sound processes. The Couacs enables the use of several mice and keyboards simultaneously, so that several musicians can play with the same computer in split-screen mode. Each game mode may be a totally different instrument with different sound processes and mappings. For now only the *Free For All* mode, renamed *Faders For All*, has been implemented. In section 4.2, we present and evaluate this game mode regarding the possibilities and issues described in section 3.2.

4.2 Faders For All Game Mode

The first game mode implemented is called Faders For All. In this mode, each avatar is associated to a different sound process, i.e. instrument, composed of a base pattern with several sound samples and several audio effects. When an avatar shoots, it triggers a variation of its associated sound processes. If one avatar shoots another avatar and hits it, the triggered variation and the effects are imposed to the sound process of the player that has been hit. Each time an avatar is hit, the volume of its associated sound process is reduced. It can be recovered by grabbing health items. Therefore, the musical result oscillates between base patterns, mix between sound processes, solo breaks and joint breaks. This mode is aimed at electronic music performances. Pattern-based compositions can be translated into songs defined in files with an XML syntax, containing instruments definitions and patterns. Each instrument is then controlled by a different musician and may interact with other instruments.

4.2.1 Interaction / Expressiveness

As explained in section 3.2, in order to be able to perform expressive instrumental gestures, and especially instantaneous excitation

[4]http://irrlicht.sourceforge.net/
[5]http://jackaudio.org/

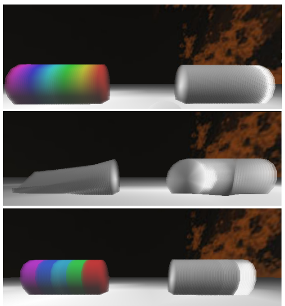

Figure 2: Tunnels (left to right, top to bottom): Continuous Hue, Continuous Density, Continuous Distortion, Continuous Rotation, Discrete Hue (5 values), Discrete Density (5 values).

gestures as defined by Cadoz [5], we need to extend the gestures done using mice and keyboards. Therefore we use the interaction techniques which were developed for an input device called *Piivert* [2], in particular percussion gestures. These gestures were designed to perform instrumental gestures with several parameters, such as velocity, direction and duration. For example, high-level *Flam* gestures are composed of two successive low-level *Hit* gestures, here button clicks, done with different fingers. *Roll* gestures are composed of three *Hit*. Instead of having only two 1-bit gestures on the mouse, we obtain two gestures, i.e. Flam and Roll, both with a 1 bit direction parameter and a duration parameter encoded on at least 7 bits (depending on the accuracy of time measurement). Therefore gesture duration can be used to replace velocity that would be provided by a pressure sensor. With these gestures, one may perform temporally accurate and expressive instantaneous excitation or modulation gestures, of course with some training

In addition to gestures done using the mouse, avatars movements can be used to control sound parameters. But as explained in section 3.2, we believe that using these parameters should not force players to move to fixed positions, i.e. they should be able to control them anywhere in the environment. This is why the Couacs relies on movement states instead of absolute position and rotation of the avatars. We define ten movement states which reflect movements with increasing dynamics, i.e. *Crouch, Stand, Crouch_Walk, Backward, Strafe, Run, Jump, Jump_Back*, *Jump_Strafe*, *Jump_Forward*. This gives us a 10 values discrete parameter that can be used in conjunction with mouse gestures for example to provide an additional parameter to excitation gestures.

Along with the fast discrete gestures performed with the mouse and the keyboard, the Couacs allows for graphical modulations of the sound processes parameters, using 3D graphical tools called *Tunnels* [1]. Players modify avatars parameters, such as color or transparency, and therefore their sound processes by moving through these tools. On the contrary to traditional graphical sliders, *Tunnels* may control one or several parameters with several discrete or continuous scales, as depicted on figure 2.

Usual 3D graphical items, such as portals, health and special abilities, are also used to enrich interactions with the environment and provide other musical possibilities, such as switching from part of a song to another.

4.2.2 Collaboration

Collaboration in the *Faders For All* game mode relies on the shooting metaphor and allows players to modify other players sound processes.

When an avatar shoots another one, it imposes its sound process variation to the other sound process, which means that for a short period, they will play notes simultaneously. At the same time, avatar parameters and their associated audio effects parameters are copied to the avatar that has been shot. Therefore, players try to shoot other players to influence both their pattern and their audio effects. Usually this leads to short musical dialogs, but also transitions between atmospheres since the audio effects tend to propagate among players, e.g. a player that has been shot shoots another player.

4.2.3 Learning / Mappings

Each weapon corresponds to a different set of mappings between input, game and sound parameters. Therefore, weapons with simple one-to-one mappings can be used by beginners while more advanced musicians can use many-to-many mappings as described by Hunt and Kirk [12]. Weapon selection thus modify the *Expertise* needed for the instrument as called by Wanderley et al. [17], along with the *Musical Freedom* described by Birnbaum et al. [3]. As in most FPS, weapons can be selected using the scroll wheel of the mouse. For now, four weapons have been implemented: *Velocity*, *Pitch*, *Repeat*, and *Multi*. When selecting a weapon, only the projectile, which hangs at the end of the weapon, changes as it can be seen on figure 3. Each projectile type represents a different control effect applied to base patterns.

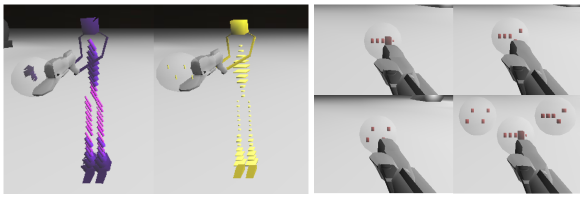

Figure 3: Left: Avatar and projectile with different aspects reflecting the audio effects applied to the sound process. Right: Projectiles for the Velocity Weapon/Effect, Pitch Weapon/Effect, Repeat Weapon/Effect. The *Multi* Weapon combines the three projectiles.

For the *Velocity*, *Pitch* and *Repeat* weapons, one triggers variations, i.e. activations of corresponding control effects, using *Hit* gestures done on the mouse. Effects values are mapped to movement dynamics, i.e. movement state ranging from *Crouch* to *Jump Forward*. In addition, the choice of the finger for hit gestures, i.e. mouse clicks, controls effects spreading.

The *Multi* weapon allows one to control the three control effects almost simultaneously since the movement state sets the projectile type and thus the triggered control effect. When standing or moving backward, this weapon triggers the velocity effect which modifies velocities of pattern notes. When strafing, it triggers the pitch effect which modifies pitches of pattern notes. When moving forward, it triggers the repeat effect which repeats pattern notes. If these movements are done while jumping, the duration of the effect is increased. The *Multi* weapon also mutes the sound process when there is no movement at all. Variations are triggered using *Flam* gestures, with gesture duration controlling the control effect value and gesture direction controlling the spread parameter. The *Multi* weapon requires more expertise since it uses more complex mappings and gestures. On the other hand it offers more musical freedom.

During an informal study, users confirmed that in order to learn how to play in the *Faders For All* mode, one must start with simple patterns, e.g. a single note, and the first weapon in order to understand which sound process they control, how to apply effects with tunnels and how to interact with other players. Then they may switch to other weapons to gain musical control. Finally, one of the users commented that it would be interesting for expert musicians to have an additional weapon allowing them to trigger the notes of patterns themselves.

4.2.4 Visualization

For each sound process, Bark coefficients of the spectrums of all sound samples are added and used to set the shape of the associated avatar by scaling cubes composing its body, from lowest frequencies on its feet to highest frequencies near its head. Loudness is also analysed in real-time and modifies the scale of the avatar. This combination of static and dynamic analysis and visualization allows players to identify other players sound processes and follow their activity.

Graphical parameters of avatars and projectiles can be modified by moving through the *Tunnels*, or by being shot. Color Hue, Rotation, Shape Distortion and Density parameters are mapped to

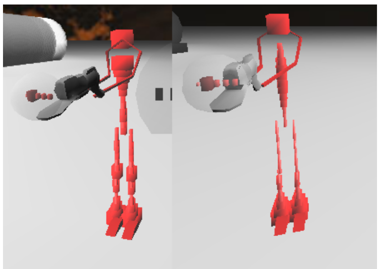

Figure 4: Avatars corresponding to a synth pattern with high and low frequencies (left) and to a drums (kick, snare, hi-hat) pattern (right).

audio effects of sound process. These visual features are combined on models to visualize sound processes parameters, taking inspiration from research done in the information visualization field by Healey [11]. They can be reset to the default values using a *Three-strike Roll* "Right-Left-Right" gesture with the mouse.

When a player is hit, the opacity of his avatar decreases. The opacity is mapped to the sound process volume which also decreases. When the sound process is almost silent, the avatar is almost completely transparent. That gives an advantage to the player. The player then needs to pick up an Health Item to restore the opacity and volume.

4.2.5 Game or Instrument

Goals of the original game mode, i.e. to kill the other players and to avoid getting killed, are preserved. In fact, since sound process volumes are associated to players health players don't want their volume to get reduced. They also try to shoot other players in order to influence their sound processes and therefore the global musical result.

5. CONCLUSION

First person shooters are characterized by highly dynamic gestures, expert interaction techniques, visualization and collaboration possibilities, and strong communities. Digital musical instruments may build on these advantages to provide new expressive interfaces while solving issues peculiar to musical interaction.

The Couacs is a collaborative multiprocess instrument based on FPS. It makes use of gaming interaction techniques and adds techniques such as *Tunnels* and percussion gestures to improve expressiveness of mouse gestures. It allows for the visualization of sound processes parameters and audio perceptual features using 3D avatars, weapons or the environment. In the *Faders For All* mode, a shooting metaphor allows for musical dialog between players.

The first perspective is the evaluation of the *Faders for all* game mode, with both musicians and gamers, in terms of musical control, learning curve, collaboration and visualization.

In order to explore new collaboration possibilities, we are working on other game modes. In particular, in the *Capture The Fader* (originally Capture the Flag) game mode, there are two teams, with base camps on each end of the environment, associated with two synchronized songs and a 3D flag acting as a crossfader on a dj mixer. The following game mode will be the *Rhythm Chase* mode (originally Rabbit Chase), in which one player holds a pattern which the other players complete with occurrences of their sound by shooting him until he drops the completely filled pattern.

6. ACKNOWLEDGMENTS

This work was made possible by a grant from the Centre National de la Recherche Scientifique (CNRS) and the Japan Science and Technology Agency (JST). Florent Berthaut would also like to thank Pascal Guitton for the opportunity and Professor Katayose and his team at Kwansei Gakuin University.

7. REFERENCES

[1] F. Berthaut, M. Desainte-Catherine, and M. Hachet. Interaction with the 3d reactive widgets for musical performance. In *Proceedings of Brazilian Symposium on Computer Music (SBCM09)*, 2009.

[2] F. Berthaut, M. Hachet, and M. Desainte-Catherine. Piivert: Percussion-based interaction for immersive virtual environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, 2010.

[3] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley. Towards a dimension space for musical devices. In *NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression*, pages 192–195, Singapore, 2005. National University of Singapore.

[4] J. Bullock. Libxtract: A lightweight library for audio feature extraction. In *Proceedings of the International Computer Music Conference*, 2007.

[5] C. Cadoz. *Musique, geste, technologie*. Éditions Parenthèses, 1999.

[6] K. Cascone. Laptop music - counterfeiting aura in the age of infinite reproduction. *Parachute, issue 107*, 2002.

[7] L. Ermi and F. Mäyrä. Fundamental components of the gameplay experience: Analysing immersion. In *DiGRA conference Changing views: worlds in play*, 2005.

[8] R. Fiebrink, G. Wang, and P. R. Cook. Don't forget the laptop: using native input capabilities for expressive musical control. In *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pages 164–167, New York, NY, USA, 2007. ACM.

[9] R. Hamilton. Maps and legends: Designing fps-based interfaces for multi-user composition, improvisation and immersive performance. *Computer Music Modeling and Retrieval. Sense of Sounds: 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 27-31, 2007. Revised Papers*, 2008.

[10] R. Hamilton. q3osc: or how i learned to stop worrying and love the game. In *Proceedings of the International Computer Music Association Conference*, 2008.

[11] C. G. Healey. Building a perceptual visualisation architecture, 2000.

[12] A. Hunt and R. Kirk. Mapping strategies for musical performance. *Trends in Gestural Control of Music*, pages 231–258, 2000.

[13] S. Jordà. Interactive music systems for everyone: exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments. In *Proceedings of the Stockholm Music Acoustics Conference (SMAC03)*, 2003.

[14] S. Jordà. *Crafting musical computers for new musics' performance and improvisation*. PhD thesis, Universitat Pompeu Fabra, 2005.

[15] P. Kearney. Cognitive callisthenics: Do fps computer games enhance the player's cognitive abilities? In *Proceeding of the International DiGRA Conference*, 2005.

[16] R. Marczak, M. Robine, M. Desainte-Catherine, A. Allombert, P. Hanna, and G. Kurtag. Enhancing expressive and technical performance in musical video games. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, 2009.

[17] M. Wanderley, N. Orio, and N. Schnell. Towards an analysis of interaction in sound generating systems. In *ISEA2000 Conference Proceedings*, 2000.

[18] G. Wang, N. Bryan, J. Oh, and R. Hamilton. Stanford laptop orchestra(slork). In *Proceedings of the International Computer Music Conference*, pages 505–508, 2009.

[19] M. Wozniewski, Z. Settel, and J. Cooperstock. A spatial interface for audio and music production. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), 2006*, 2006.

[20] M. Zadel and G. Scavone. Different strokes: a prototype software system for laptop performance and improvisation. In *Proceedings of the 2006 conference on New interfaces for musical expression*, NIME '06, pages 168–171, Paris, France, France, 2006.