MUSIC AS ACTIVE INFORMATION RESOURCE FOR PLAYERS IN VIDEO GAMES

Marian Nagorsnick and Alke Martens
Institute of Computer Science, University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, Germany

ABSTRACT

In modern video games, music can come in different shapes: it can be developed on a very high compositional level, with sophisticated sound elements like in professional film music; it can be developed on a very coarse level, underlying special situations (like danger or attack); it can also be automatically generated by sound engines. However, in all these approaches, no active adaptation of musical elements to user activities can be found. Context dependent and adaptive music elements are no longer background or emotional support structures, but transform to information resources, which can be actively used by the user. Thus, they become a kind of human computer interface. In this paper, we focus on music as active information resource in video games and game-based learning systems.

KEYWORDS

Music, information, human computer interaction, auditory interface, video games.

1. INTRODUCTION AND ANALYSIS OF GAMES

Up to date video games and game-based learning systems are dominated by visual output and mostly rely on the visual interface to pass information to the player (see Nagorsnick (2012)). Even if this meets the user's expectations in most cases, it does not satisfy the multimedia potential video games have reached by now. The possibility of playing video games without loss of information with the volume turned off is characteristic for the majority of them. The graphical user interface or GUI – as part of the visual interface – provides the player with useful information, often permanently throughout the entire game. Optionally the player can blend or hide parts of the information. Nowadays, video games use special processors on the graphic cards in the computer, to generate the computation intensive visual information, like the background of a virtual world, the non-player characters' movement etc.

In some video games the player must keep track of information at all times. This becomes even more important if time is a crucial game element, e.g. as part of a strategy. Often video games contain such a large amount of relevant information that the GUI is about to become confusing. In really bad cases, the rendered virtual reality is hidden behind all the information-conveying GUI-elements. The cognitive effort to keep track of all this is comparably high, whereas the bandwidth of cognitive input is comparably low (restricted to vision, excluding auditory feedback) (see for example Mousavi et. al. (1995)). Brewster (1994) points out that "[sound] is a useful complement to visual output", because it can "reduce the amount the user has to perceive through the visual channel". Alty (1995) argues, that the use of music in Human Computer Interface design should be considered. And like studies by Brewster (1994), experiments with music representing hierarchical menus (LePlâtre, 1998) or navigation with music (Jones, 2006) show that music is well suited to convey information to the listener. Music as part of the auditory interface of the video game is mostly only used to enhance the atmosphere experienced by the player. If it would be possible to shift some relevant information from the GUI to the music, three effects might be reached:

1. Take human information processing to another level (audio adding to video), potentially also allowing multitasking, and make the game more interesting as multiple cognitive levels are addressed.
2. Reduction of cognitive load on the pure visual level (GUI elements can be reduced).
3. Relieve the visual system of the player an intensify the game experience.

For this approach, we decided to get rid of automatic music generation using Algorithmic Composition techniques, like e.g. in the algorithmic music generators FractMus and SoundHelix (see http://www.gustavodiazjerez.com and http://www.soundhelix.com, last visited June 2015). In contrast to this, our approach is based on human musical compositions for certain situations, combined with an algorithm which steers an automatic situation adaptation and an automatic overlay of compositional parts, to achieve the "music" intended by the human composer as the result, instead of simple alerts in form of earcons or auditory icons (see e.g. Blattner (1989), Brewster (1994), Gaver (1986)).

Music is a regular part of most video games. Either it is specifically written for the video game or it is chosen from public music. As a first research step, we made an analysis of different computer games and investigated the use of music there. For the first step of the investigation we selected 7 major titles, namely: *Dragon Age 2*, *Mass Effect 2*, *Command & Conquer 3: Tiberium Wars*, *Dawn of War II - Retribution*, *Sim City 4*, *Unreal Tournament 3* and *Tekken 6*. All this titles are commercially successful and representative for their genre. We found that the main purpose of the (background) music in this games is to enhance the atmosphere perceived by the player (e.g. tension). To convey information to the player, the investigated games used pre-recorded voice messages, GUI-elements (for instance text or health bars) or graphical elements as part of the virtual scene. To achieve a broader view we also examined videogames with a more intense use of music. The games are *Flower, Auditorium, Loom, Audiosurf, Guitar Hero 3* and *SingStar Vol. 1*, for which music is a key element of the gameplay. And although those games' music emerges from mere background music towards a more central role it neither carries exclusive information. We found that among all the examined video games and genres only music games (in this case *Guitar Hero 3* and *SingStar Vol. 1*) established music as an active information resource.

The next step in our research was to extract situations where music can be used as information resource. We chose the CRPG genre as a basis for this investigation, because it combines a large amount of different elements that can be found within the game, e.g. the story told and the story turns, emotions, characters, battles, useable skills, quests, riddles and so forth. Sometimes time is a crucial factor too. And on top of it CRPGs tend to make extensive use of music to fit optimally into the atmosphere. Altogether we examined 9 video games of the CRPG genre, each 3 for a sub-genre. The Classic-CRPGs are *Baldur's Gate II: Enhanced Edition*, *Dragon Age: Origins* and *Star Wars: Knights of the Old Republic II*. The Action-CRPGs are *Diablo III*, *Dust: An Elysian Tail* and *Titan Quest*. Hybrid-CRPGs, that combine elements of both Classic- and Action-CRPGs, where picked with *Dark Souls: Prepare to Die Edition*, *Mass Effect* and *The Elder Scrolls V: Skyrim*. Some of those feature some special elements but in general they have much elements in common, which we called CRPG-elements. A detailed explanation of this CRPG-elements can be found in (Nagorsnick, 2014). With this CRPG-elements we established a foundation for our approach to make music an adaptive and active resource of information, since information is created by those elements. How this foundation is used exactly will be covered in the following section.

2. MUSIC AS ACTIVE RESOURCE OF INFORMATION

As noted by Collins (2007), moving smoothly from one musical piece (so called cues) to another assists in the continuity of a game and the illusions of gameplay. The current methods applied to video games use pre-recorded musical pieces that are played in a loop or cross-faded to switch between the pieces (Collins, 2008). The problem is that "adjacent" audio files have to match in harmony and time signature (respectively measure) during crossfade, so they can't cause unintended disharmony or rhythmical stumbling (see e.g. Reese (2012), Collins (2007), Nierhaus (2009), Shan and Chiu (2010)). We decided to take another direction and thus we will introduce Algorithmic Transition as branch of Algorithmic Composition.

The main idea behind is to utilize aesthetical understanding and creativity of human composers in order to skip the boundaries of computable creativity and all difficulties of Algorithmic Composition techniques. To achieve this goal, a human musician composes a certain number of compositional parts. With these parts the final music can be compiled by an algorithm. For this algorithm, we need to define rules, e.g. to define, which parts can be played together (vertical transition) or how parts have to be merged in time (horizontal transition). Next the algorithm has to keep up the compositional coherence during transitions and therefore being able to manipulate the compositional parts the music consist of. This ability of manipulation or alteration of music is also the key to use music as active information resource (Nagorsnick, 2014).

The first concepts are music events, which we consider as auditory events in a musical context, and musical components. Music events can be atomic or molecular. An atomic music event is a single, sounding musical tone or a single rest. They are called atomic because they can’t be divided any further. A molecular music event, like a sounding melody, consists of at least two atomic music events. Furthermore each music event needs three musical components: tone, time and timbre. This means that a music event needs a defined pitch (e.g. *C’’*), a temporal information that determines when and how long the event occurs (e.g. first semibreve in the first bar) and at last a timbre (e.g. pianissimo cello), that makes the music event sound.

The second concepts are musical parameters, which we consider as modifiable qualities of music events. Like music events, we distinguish between atomic and molecular musical parameters. The former can only be applied to atomic music events, whereas the latter can only be applied to molecular music events. One example: The atomic musical parameter pitch can only be applied to a single sounding musical tone (e.g. *a’*). Pitch would make no sense if applied to a sounding melody that combines multiple pitches. In addition a musical parameter can affect two musical components at a max. Would it affect all three musical components it would be a music event per definition.

To be able to make use of it, we need game events – the third concept. We see game events as derivations of the discovered CRPG-elements. They cause state changes within the virtual reality and are the source of the information actively conveyed to the player by music.

To show the applicability of our approach, we chose three CRPG-elements. To be able to investigate music as active information resource we modelled a virtual test environment using the game engine *Unity3D*. The game events we derived from the chosen CRPG-elements were implemented in the test environment, whereas each game event influences the music via rules provided by a pattern language. Those rules determine which musical parameter is to be used for manipulation and how, e.g. (molecular) melody for major-to-minor change. Furthermore they define when and for how long a manipulation is done. This way we wanted to ensure that the compositional coherence is always kept up. Figure 1 illustrates this by taking the length – measured in bars – of musical themes (each triggered by another game event) into account, both for horizontal and vertical transitions.

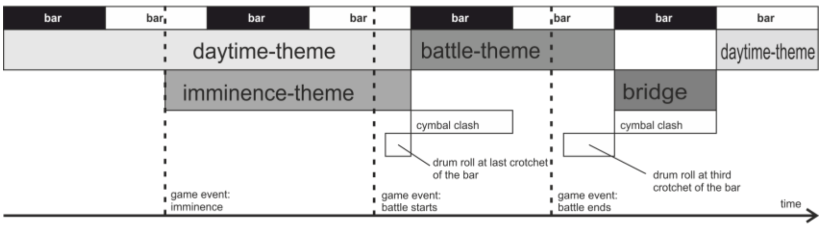

Figure 1. A schematic view for a possible battle scenario.

To be able to make use of the manipulation of music events by musical parameters, we passed music in form of symbolic data to the algorithm, similar to MIDI or MusicXML. To raise our musical themes to the level of music events we used audio samples, representing atomic music events, e.g. a note played pianissimo by a cello. These atomic music events are finally combined to molecular music events conveying information to the player, of course depending on the actual musical themes that in turn depend on game events.

3. CONCLUSION AND OUTLOOK

With our virtual environment we achieved a first successful use of music as an active information resource in video games with the help of Algorithmic Transition. Hence all musical themes are based on a handful of human-composed themes. The rest is successfully created by Algorithmic Transition. The manipulation of music events by means of the musical parameters is working like intended. No music event causes disharmony or other flaws regarding compositional coherence interfered with the experienced gameplay. Even if the audio quality can be improved, our first approach of using Algorithmic Transition to make music an active information resource in video games was successful.

In our current research, we’ll put the formal model on a firmer ground and look at the user’s performance with musical support (in the form of some smaller evaluations and user observations).

REFERENCES

AudioSurf (standard edition). 2008. PC [Game]. Steam: Bellevue.
Auditorium (standard edition). 2009. PC [Game]. Cipher Prime Studios: Philadelphia.
Baldur's Gate II (Enhanced Edition). 2013. PC [Game]. Atari: New York.
Command & Conquer 3: Tiberium Wars (Kane Edition). 2007. PC [Game]. Electronic Arts: Redwood City.
Dark Souls (Prepare to Die Edition). 2012. PC [Game]. Namco Bandai Games: Tokio.
Dawn of War II – Retribution (standard edition). 2011. PC [Game]. THQ: Agoura Hills.
Diablo III (standard edition, with Addon). 2012. PC [Game]. Blizzard Entertainment: Irvine.
Dragon Age: Origins (standard edition, with Addon). 2009. PC [Game]. Electronic Arts: Redwood City.
Dragon Age 2 (standard edition, with all DLCs). 2011. PC [Game]. Electronic Arts: Redwood City.
Dust: An Elysian Tail (standard edition). 2013. PC [Game]. Microsoft Studios: Redmond.
Flower (standard edition). 2009. PlayStation 3 [Game]. Sony Computer Entertainment: Tokio.
Guitar Hero 3 (standard edition). 2007. Xbox 360 [Game]. Activision: Santa Monica.
Loom (standard edtion). 1990. PC [Game]. Lucasfilm Games: Marin County.
Mass Effect (standard edition). 2008. PC [Game]. Electronic Arts: Redwood City.
Mass Effect 2 (standard edition, with all DLCs). 2010. PC [Game]. Electronic Arts: Redwood City.
Sim City 4 (Deluxe Edition). 2010. PC [Game]. Electronic Arts: Redwood City.
SingStar Vol. 1 (standard edition). 2007. PlayStation 3 [Game]. Sony Computer Entertainment: Tokio.
Star Wars: Knights of the Old Republic II (standard edition). 2005. PC [Game]. LucasArts: Marin County.
Tekken 6 (standard edition). 2009. PlayStation 3 [Game]. Namco Bandai Games: Tokio.
The Elder Scrolls V: Skyrim (Legendary Edition). 2013. PC [Game]. Bethesda Softworks: Rockville.
Titan Quest (standard edition, with Addon). 2006. PC [Game]. THQ: Agoura Hills.
Unreal Tournament 3 (Black Edition). 2009. PC [Game]. Midway Games: Chicago.

Alty, J.L., 1995. Can we use music in computer-human communication? In *People and Computers X. Proceedings of HCI 95*, pp. 409-424.

Blattner, M. et al, 1989. Earcons and Icons: Their Structure and Common Design Principles. In *Human-Computer Interaction*, Vol. 4, No. 1, pp. 11-44.

Brewster, S.A. et al, 1994. A detailed investigation into the effectiveness of earcons. In *Auditory display: sonification, audification, and auditory interfaces*, pp. 471-498

Collins, K., 2007. An Introduction to the Participatory and Non-Linear Aspects of Video Games Audio. In *Contemporary Music Review*, Vol. 28, No. 1, pp. 5-15.

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

Gaver, W., 1986. Auditory Icons. In *Human-Computer Interaction*, Vol. 2, No.2, pp.167-177.

Jones, M. et al., 2006. Navigation-by-Music for Pedestrians: an Initial Prototype and Evaluation. In *International Symposium on Intelligent Environments*, pp. 95-101.

LePlâtre, G. et al, 1998. An investigation of using music to provide navigation cues. In *Proceedings of the 1998 international conference on Auditory Display (ICAD'98)*, p. 19.

Mousavi, Seyed Yaghoub; Low, Renae; Sweller, John, *1995.* Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of Educational Psychology,* Vol 87(2), Jun 1995, 319-334. http://dx.doi.org/10.1037/0022-0663.87.2.319

Nagorsnick, M., 2012. *Musikalische Kodierung und Darstellung von Informationen in Videospielen*. Seminar paper. University of Rostock.

Nagorsnick, M., 2014. *Algorithmische Abbildung von Spielereignissen auf Musikereignisse in Videospielen*. Diploma thesis. University of Rostock.

Nierhaus, G., 2009. *Algorithmic Composition – Paradigms of Automated Music Generation*. Springer-Verlag, Wien.

Reese, K. et al, 2012. A framework for interactive generation of music for games. In *Proceedings of CGAMES'2012 USA - 17th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational and Serious Games*. pp. 131-137.

Shan, M.K. & Chiu, S.C., 2010. Algorithmic compositions based on discovered musical patterns. In *Multimedia Tools and Applications*, Vol. 46, No. 1, pp. 1-23.