ENABLING SOCIAL SOFTWARE-BASED MUSICAL CONTENT FOR COMPUTER GAMES AND VIRTUAL WORLDS

Ioannis Deliyannis[1], Ioannis Karydis[2] and Kostas Anagnostou[3]

[1]Department of Audio and Visual Arts, Ionian University, Corfu, GR
yiannis@ionio.gr
[2,3] Department of Informatics, Ionian University, GR
{karydis,kostasan}@ionio.gr

ABSTRACT

The audio characteristics of computer games and virtual worlds have been shown to greatly affect the users' experience in such environments. Social software platforms such as last.fm, youtube.com and facebook.com record their users' musical preferences offering thus a wealth of accessible information for the enrichment of games and virtual worlds. This paper presents a web-based framework designed to retrieve, process, analyse, categorise appropriately and embed personal musical preferences directly into games and virtual worlds. The whole process is demonstrated in action via the development of a classical computer game that features such functionality. Typical issues that arise during application development are discussed, while case based solutions are proposed.

KEYWORDS

Interactive-Adaptive Audio, Personalised Music-Profile, Games, Virtual Worlds, Multimedia.

1. INTRODUCTION

This work introduces the MIRVRES framework designed to incorporate musical content sourced from user-based social software services directly into computer games and virtual worlds. In that respect, personal musical preferences may be used to furnish applications with personalised user-based musical content in a transparent and consistent manner. The framework establishes a method that can be accessed directly from a game or virtual world environment, designed to automate the overall process that integrates querying, retrieval, categorisation and direct access of personalised musical preferences in audio-file format. Successful input of user credentials into the developed system's interface initiates a remote and automated process, which outputs a resulting XML file containing all the external content-access information. The open-source nature of this implementation combined with the remote web-access interface offers to game and virtual world developers a tool to access directly individual musical profiles. This may be used to cover particular interactive music application areas where users interact with the musical content, or adaptive musical needs where user actions within the game or virtual environment trigger dynamic audio adjustment. As the process is completely external to the game environment it does not require the client-system to provide additional audio-processing capabilities, enabling mobile-based applications to utilise it for their purposes.

The rest of this paper is organised as follows. Section 2 introduces related work and examples in the field of adaptive audio for games and displays their developmental complexities, before analysing the functionality of the proposed framework. The development of a real-life case study under which a game is furnished with customised audio is discussed in Section 3. Therein, practical aspects that include the speed of processing, delivery, quality of service and the overall experience are presented while framework-adaptation issues are discussed. The work concludes with the presentation of particular development issues that developers need to take into consideration, when they whish to employ the framework and incorporate similar functionality into their games and virtual world applications.

2. Enabling Adaptive Audio for Games and Virtual Worlds

Although games and virtual worlds often share common characteristics, allowing thus common tools to be used in their development [1], they do not seem to follow recent advances in interactive audio personalisation [1-4] and social software based music information retrieval [5-11]. In game development scenarios and particularly when focusing in the musical forefront, our initial research revealed various approaches [3]. Most games embed a fixed set of composed, edited or carefully selected tracks to the application that may be pre-cached and called on request. Playback may be triggered interactively according to user actions or adaptively, according to certain key points of the underlying scenario. For example, the "Grand Theft Auto"[12] game series follows the above characteristic scenario, as the soundtrack listened to when a player uses a vehicle, may be altered by the user via the appropriate function buttons, simulating change of the selected radio-frequency in real-life. The music-driven game "Audiosurf" [13] is considered a non-characteristic case as user-inputted audio file is used in order to shape the game-play complexity for each level. Many other game development approaches may be referenced including the use of affective models based on the inclusion of mood, emotion, and sentiment [4]. Most game video consoles including Xbox 360, Wii and PS3 allow the use of externally available music from CD media, given that the game supports this functionality [2]. To our best knowledge, proprietary games and virtual worlds featuring dynamic utilisation of network-based music from social software oriented music sources are rare.

It is necessary therefore to introduce a framework that allows direct incorporation of music originating from social software sources, providing each user with a customised soundtrack. Our research revealed various issues that need to be addressed, common to most implementations. For example, although in the case of client-server WWW-based applications use of the proposed framework is highly recommended, as it does not pose any further requirements in the client-side, a negative aspect of the current process is that it may increase application-loading time for the transfer of musical content. This may be addressed via utilisation of caching techniques, similar to those utilised for the development of multimedia applications featuring dynamic-dynamic interaction capabilities [14].

2.1. Introducing the MIRVRES Framework for Games and Virtual Environments

The web-version of the MIRVRES framework is presented in this section, covering the content-access requirements under the current case-study scenario. Figure 1 displays the process schematically. Users authorise the application to search and detect their personalised musical preferences, which may either already be retrieved interactively [5] from services such as YouTube or Facebook, or may be already retrieved from a dedicated system such as Last.fm, a method utilised under the current case-study. These are then matched appropriately using another service; in this case iTunes provides links to the actual audio clip files. Each file is retrieved and analysed by the "Canoris" service in order to detect specific information, in our case Beats Per Minute (BPM). Resulting information is stored in an XML file, which is subsequently accessed by the game or virtual world, enabling direct access to the file and metadata.

An important aspect of the proposed framework is that it may also be implemented locally in non-networked environments, in order to analyse, evaluate and provide a customised experience to a single-user, based on a single music-collection. In order for this functionality to be implemented, step 3 will have to be combined with step 4, both accessing the iTunes library that can provide access to all privately stored musical content.

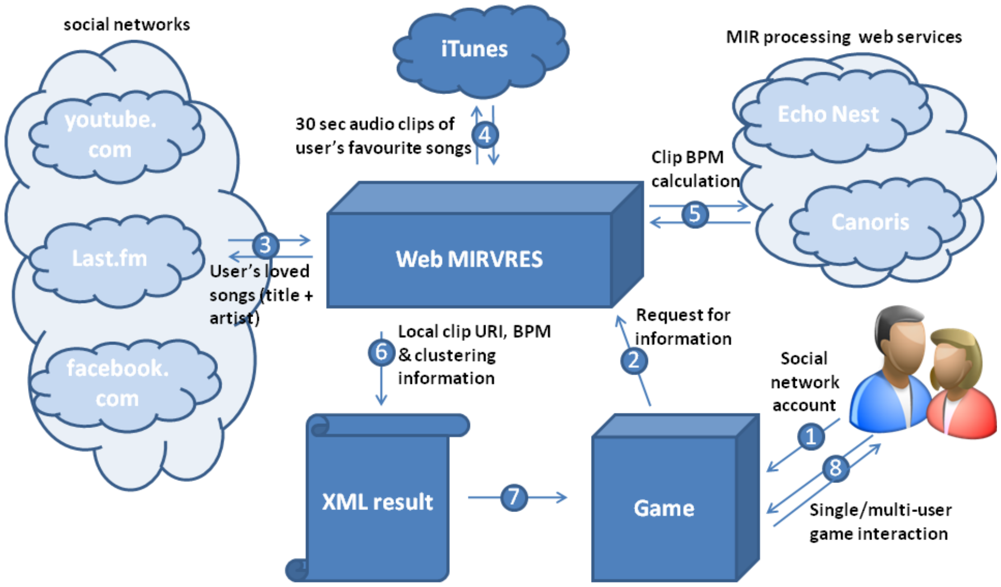
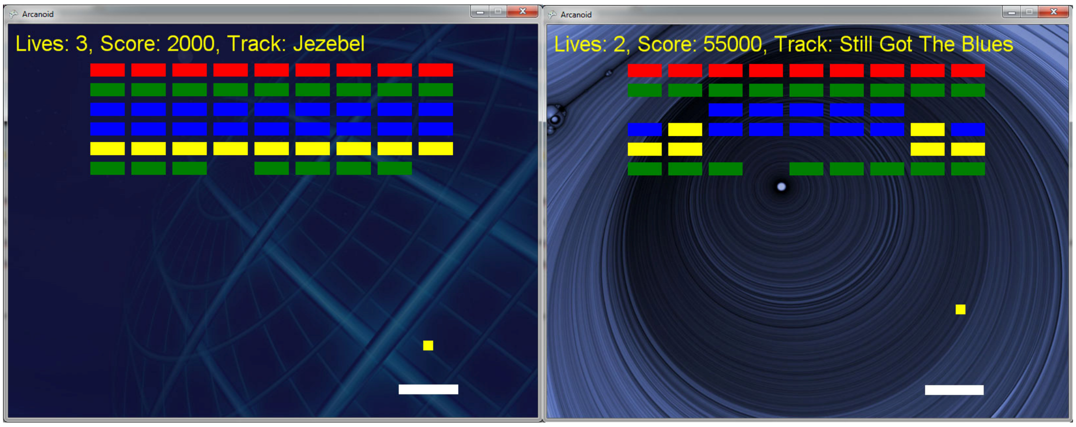

social networks
youtube.com
Last.fm
facebook.com
iTunes
30 sec audio clips of user's favourite songs
4
MIR processing web services
Echo Nest
Canoris
Clip BPM calculation
5
3
User's loved songs (title + artist)
Web MIRVRES
6
Local clip URI, BPM & clustering information
Request for information
2
Social network account
1
8
XML result
7
Game
Single/multi-user game interaction

Figure 1. Schematic of the real-life game-based case study

3. Employing the Framework Under a Real-Life Case Study

In this work a fully functional case study was realised in order to practically demonstrate the functionality of the proposed framework, under a realistic game-scenario. The programming environment selected for experimentation was Microsoft's XNA Game Studio, and the open source game XNA Arkanoid. Under this setting the audio-handling routines were re-engineered in order to act as the testing platform while dynamic audio-adjustment was linked to the game-play. Its functionality may be described as a mapping between each stage's score and the Beats Per Minute (BPM) of each of the selected audio files, calculated by the framework service and stored in an XML document accessible by the game. The game states displayed in Figure 2, exhibit audio track change.



Lives: 3, Score: 2000, Track: Jezebel

Lives: 2, Score: 55000, Track: Still Got The Blues

Figure 2. Two game states where the score is linked to the audio BPM

The XML information derived by the appropriate framework routine for the low-BPM track shown in Figure 2, left, is listed below, followed by the higher-BPM track of Figure 2, right).

```
<Track>
        <track_name>Jezebel</track_name>
        <track_artist_name>Sade</track_artist_name>
        <bpm>75.930725097700005</bpm>
        <uri>http://karydis.ionio.gr/MIRVRES/rj/</uri>
        <filename>8b40bf7a097e41e6af05fd1b981900ab.wma</filename>
</Track>

<Track>
        <track_name>Still Got The Blues</track_name>
        <track_artist_name>Gary Moore</track_artist_name>
        <bpm>81.664299011200001</bpm>
        <uri>http://karydis.ionio.gr/MIRVRES/rj/</uri>
        <filename>e30bd8e12e254c22b953196e90927b7a.wma</filename>
</Track>
```

The BPM-based music selection routine used in the game employs clustering using the *k-means algorithm* [15]. The audio-selection method can be adjusted to suit the intended end-system. For example, virtual world scenario developers that wish to implement personalised background music for use within the environment (musical carpet) may do so by introducing the appropriate entry in the XML file. Under this scenario, the user's actions (interaction) and communication contexts may be used to describe the desired temporal state (relaxed, normal or active) triggering appropriate music track selection.

Another important aspect involves content licensing. The social software website last.fm used to detect user preferences although it records the favourite user-tracks, it does not freely provide the audio streams in high quality. This problem was resolved in the current game via utilisation of the track name field provided by the iTunes API. This service allows an application to download and store thirty-seconds of the original audio file. Limited audio length may not be enough for all applications, thus alternative strategies may be implemented. These include the use of audio tracks stored locally in the user's computer, enabling rapid access and processing, or content streaming from alternative sources such as YouTube.

The framework complete with source code and a ready-to-run example are freely accessible over the WWW from the address: http://karydis.ionio.gr/MIRVRES/. The application receives personalised user preferences from a test-account in Last.fm social software website which may be changed at will by altering the appropriate functional call. In the present game state and for demonstration purposes, the user should access the service via the above web address, and store the resulting XML file in the game folder enabling it to download the appropriate data files. This implementation allows developers to import effortlessly the proposed functionality directly into their game-development environments.

3.1. Framework Adaptation for Games and Virtual Environments

Personalised/adaptive music is an important issue for games [16] and virtual environments [17]. Researchers address the issues that arise from a variety of perspectives ranging from music similarity [18-23], social media [5, 7, 11, 24] and end-system adaptability [4, 9, 25-29], to emotion [4, 30-32], affect [33, 34] and the interactive nature of implementations [2, 35]. Therefore it is imperative to review possible framework adaptation strategies that may address various system and user issues. We categorise these issues under three titles, which are interrelated: Scenario Complexity, User Dimension and Content Availability.

End-system "scenario complexity" may be considered a governing factor as it that affects directly use-scenarios of adaptive audio. Under each new system, developers need to predict, map and address special scenario-based requirements in order to cover special use-cases. A characteristic instance may be the need to replace all the pre-determined audio sequences with new audio streams, a process that introduces varying pre-caching requirements for interactive or adaptive use of the content. This is a typical case where complexities increase. Characteristic complexities that are introduced in varying scenarios are summarised in Table 1, where interactive and adaptive requirements are recorded under different development perspectives.

Table 1. Interactive and Adaptive Music Complexities Introduced During the Development of Game and Virtual World Environments

Music	**Programming Complexities**	**Game / Virtual Worlds Environment/ Network Issues**	**Content-Context Complexities**
Interactive Music	Based on user actions, streams need to be available on request.	Playback length should be appropriate for actions Requires dynamic editing (fade-in/fade-out, BPM synchronisation) to support user-interaction.	Small user-collection may introduce repetition. Large user-collection may introduce selection problems.
Adaptive Music	Based on the scenario, streams may need to be pre-cached in a pre-determined order.	Duration may not be appropriate, may need dynamic editing (fade-in/fade-out, BPM synchronisation).	Music repetition may be permitted depending on game design.

A "user dimension" is another factor that refers to the maximum number of users that may simultaneously use the system or "meet" during their virtual experience. In that respect, a single player game scenario is less complex than the two-player turn-based scenario of the same game, as in the latter case the proposed framework has to be employed twice and the game should be adapted in order to store different player settings and scores. To complicate matters further, implementing the same game to support simultaneous two-player modes increases the dimensionality of the problem, introducing a decidability problem that needs to be addressed during system design. In other words, problems may arise in the case of multi-player games and virtual worlds, where users with different musical preferences are present in a game in the same physical space. This is an issue that may require either the use of the union of their preferences (if any), or the gaming-performance (score) of each user may dictate which users' audio profile will be used, offering an interactive soundtrack selection during game play.

The term "content availability" refers to the volume and quality of information that the user inputs into the system. A new user may have insufficient or a very large number of tracks available in their profile. In the first case a developer may select to utilise the default application audio and display a warning message to the user prompting to add more tracks to their profile. In the case of large availability of tracks, it is suggested a maximum limit of tracks parameter to be set and select only the favourite user-tracks from the web service. In the case of last.fm this is practically easy to be accomplished via the <playcount> parameter found in information stored for each user-track and appropriate sorting of the track list.

Other issues that arise during new system development include the flexibility of the framework in terms of context selection. Clearly under the proposed implementation we have utilised the bits per minute (BPM) attribute that may not always cover the developers' requirements. This is an issue that requires further research, as one may choose various strategies to address it.

Developers may either choose to use musical genre as a filter for track selection, or utilise external music-discovery applications such as last.fm to allow dynamic musical context matching to game content via the similarity search function offered. Copyright issues are also a great concern under interactive music selection as it is not possible to predict beforehand which tracks will be downloaded. This is an issue that has already been examined for interactive application scenarios [36].

4. CONCLUSIONS

Adapting the game to suit player's character and preferences has been recognised as an important factor of user-experience improvement. Videogame players have long had the ability to customise the game-playing experience by interactively switching between elements that include the main game character, the enemies, the difficulty and even the game environment and rules. Although videogame music has the potential to enhance the gaming experience on screen in unique ways [37], it has not yet followed the same adaptation mechanisms observed as in the above-mentioned game factors. This often results in the use of musical scores that are indifferent and not pleasurable due to the high frequency of repetition, a factor that the user is not usually given the option to adjust according to preference.

In this work we have presented a first step towards videogame/virtual world music customisation according to user preferences and life style. Social networks possess a wealth of information about the kind of music the user prefers. We introduced the MIRVRES framework, and a real-life open-source web-based tool that enables developers to furnish their environments with user-based functionality. Leveraging this information to adapt a videogame's musical score can enhance the user experience. In the near future we plan to evaluate the technique and measure enhancement of user experience as well as adapt the algorithm to more complex games and audio attributes that may be automatically extracted in order to offer games and applications with a more adaptive experience.

REFERENCES

[1] D. Trenholme and S. Smith, "Computer game engines for developing first-person virtual environments", *Virtual Reality,* vol. 12, pp. 181-187, 2008.

[2] K. Collins, *From Pac-Man to Pop Music: Interactive Audio in Games and New Meda*. University of Waterloo, Canada: Ashgate, 2008.

[3] K. Collins, *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*: MIT Press, 2008.

[4] M. Eladhari, R. Nieuwdorp, and M. Fridenfalk, "The soundtrack of your mind: mind music - adaptive audio for game characters", presented at the Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology, Hollywood, California, 2006.

[5] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, "Musicbox: Personalized music recommendation based on cubic analysis of social tags", *IEEE Transactions on Audio, Speech & Language Processing,* vol. 18, pp. 407-412, 2010.

[6] J. Green, "Understanding the Score: Film Music Communicating to and Influencing the Audience", *The Journal of Aesthetic Education,* vol. 44, pp. 81-94, 2010.

[7] M. Levy and M. Sandler, "Music information retrieval using social tags and audio," *IEEE Transactions on Multimedia,* vol. 11, pp. 383-395, 2009.

[8] M. Kibby, "Negotiating personal music archives", *Information, Communication & Society,* vol. 12, pp. 428-443, 2009.

[9] A. R. Brown and T. Kerr, "Adaptive music techniques", presented at the Improvise: The Australasian Computer Music Conference Brisbane, 2009.

[10] C. McKay and I. Fujinaga, "Combining features extracted from audio, symbolic and cultural sources", in *International Conference on Music Information Retrieval*, 2008, pp. 597-602.

[11] P. Lamere, "Social tagging and music information retrieval", *New Music Research,* vol. 37, pp. 101-114, 2008.

[12] R. Games. (2011, 26/01/2011). *Grand Theft Auto Game Series*. Available: http://www.rockstargames.com/grandtheftauto/

[13] L. Audiosurf. (2011). *Audiosurf (TM) Ride your Music*. Available: http://www.audio-surf.com/

[14] I. Deliyannis and M. F. Webster, "Multi-Level Modelling and Interactive Multimedia presentation of Scientific Data over various Media", in *IADIS*, Lisbon Portugal, 2002.

[15] J. B. MacQueen, "Some Methods for Classification and Anlysis of Multivariate Observations," in *5th Berkeley Symposioum on Mathematical Statistics and Probability*, 1967, pp. 281-297.

[16] K. Collins, "From Bits to Hits: Video Games Music Changes its Tune", *Film International,* vol. 13, pp. 4-19, 2005.

[17] B. Zhu and T. Liu, "Research on Emotional Vocabulary-Driven Personalized Music Retrieval", presented at the Proceedings of the 3rd international conference on Technologies for E-Learning and Digital Entertainment, Nanjing, China, 2008.

[18] J. J. Aucouturier and F. Pachet, "Music similarity measures: Whats the use?", in *International Symposium on Music Information Retrieval*, 2003, pp. 157-163.

[19] B. Logan, D. P. W. Ellis, and A. Berenzweig, "Toward evaluation techniques for music similarity", presented at the SIGIR 2003: Workshop on the Evaluation of Music Information Retrieval Systems, Toronto, Canada, 2003.

[20] E. Pampalk, "Audio-based music similarity and retrieval: Combining a spectral similarity model with information extracted from fluctuation patterns", in *nternational Symposium on Music Information Retrieval*, 2006.

[21] K. West and P. Lamere, "A model-based approach to constructing music similarity functions", *EURASIP Journal on Advances in Signal Processing,* pp. 1-10, 2007.

[22] M. Slaney, K. Weinberger, and W. White, "Learning a metric for music similarity", in *Information Society for Music Information Retrieval*, 2008, pp. 148-153.

[23] B. McFee, L. Barrington, and G. Lanckriet, "Learning similarity from collaborative filters", presented at the International Society for Music Information Retrieval, Utrecht, Netherlands, 2010.

[24] L. Barrington, D. O'Malley, D. Turnbull, and G. Lanckriet, "User-centered design of a social game to tag music", presented at the Proceedings of the ACM SIGKDD Workshop on Human Computation, Paris, France, 2009.

[25] J. Robertson, A. D. Quincey, T. Stapleford, and G. Wiggins, "Real-Time Music Generation for a Virtual Environment", presented at the ECAI-98 Workshop on AI/Alife and Entertainment, Brighton, England, 1998.

[26] V. L. Hanson, *The user experience: designs and adaptations*. New York City, New York: ACM Press, 2004.

[27] P.-H. Tan, S.-W. Ling, and C.-Y. Ting, "Adaptive digital game-based learning framework", presented at the Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts, Perth, Australia, 2007.

[28] L. V. velsen, T. V. d. geest, R. Klaassen, Micha\, \#203, and l. Steehouder, "User-centered evaluation of adaptive and adaptable systems: A literature review", *Knowl. Eng. Rev.,* vol. 23, pp. 261-281, 2008.

[29] A. Paramythis, S. Weibelzahl, and J. Masthoff, "Layered evaluation of interactive adaptive systems: framework and formative methods", *User Modeling and User-Adapted Interaction,* vol. 20, pp. 383-453, 2010.

[30] L. Leahu, S. Schwenk, and P. Sengers, "Subjective objectivity: negotiating emotional meaning", presented at the Proceedings of the 7th ACM conference on Designing interactive systems, Cape Town, South Africa, 2008.

[31] R. Dillon, N. K. Bee, and S. Rozner, "The Music and Emotion Driven Game Engine: Ideas and Games", presented at the Computer Games and Allied Technology, Singapore, 2009.

[32] B. G. Morton, J. A. Speck, E. M. Schmidt, and Y. E. Kim, "Improving music emotion labeling using human computation", presented at the Proceedings of the ACM SIGKDD Workshop on Human Computation, Washington DC, 2010.

[33] R. W. Picard, "Affective computing: challenges", *Int. J. Hum.-Comput. Stud.,* vol. 59, pp. 55-64, 2003.

[34] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: audio, visual and spontaneous expressions", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 39-58, 2009.

[35] A. Berndt and K. Hartmann, "The Functions of Music in Interactive Media", in *Interactive Storytelling*. vol. 5334, U. Spierling and N. Szilas, Eds., ed: Springer Berlin / Heidelberg, 2008, pp. 126-131.

[36] I. Deliyannis, I. Karydis, and D. Karydi, "iMediaTV: Open and Interactive Access for Live Performances and Installation Art", presented at the 4th International Conference on Information Law (ICIL2011), Thessaloniki, Greece, 2011.

[37] K. Jørgensen, "On the Functional Aspects of Computer Game Audio", presented at the AudioMostly, Piteå, Sweden, 2006.