Musical Mood-Based Mobile Gaming

Alexander HODGE, Karen COLLINS, Kelvin LAM and Peter J. TAILLON
The Games Institute, University of Waterloo

***Abstract*—This paper explores ways of using mood-based audio extraction methods on player-selected music to drive content in mobile video games. Specifically, we describe the methods employed in the development of a game adapting the CLAM C++ Library for the Apple iPod.**

I. INTRODUCTION

User-selected music can be used to drive game elements. This concept was introduced by *Vib-Ribbon* in 1999, in which the music chosen by the player created the structure of the gameplay through altering the game's generation of level mapping. The game scanned the user's CD and made two obstacle courses for each song (one easy and one difficult), so that the game was as varied as the music the player chose. Since *Vib-Ribbon*'s release, there have been a handful of games to use the player's music as a way to alter game content. *Phase*, for instance, was designed by Harmonix for the fifth generation iPod. *Audiosurf,* a puzzle-racing game that uses the player's input music files to design the game maps gained much attention after its release in 2008. These games rely on extracting information from the player's music to procedurally generate gameplay elements. Until recently, such games have primarily relied on beat mapping and tempo. Rosso, Tzanetakis and Gooch describe a method for analyzing player actions and translating it into a beat sequence synchronized to an existing musical track [1]. Arrasvuori and Holm developed a more extensive approach in the creation of two game prototypes where they explore a wide variety of musical properties that could be used to drive gameplay, but do not mention mood [2]. Arrasvuori and Holm describe a list of game parameters that could respond to music, including for instance speed of game, type of objects, location, properties or behaviour of objects, camera angle, rules, scoring, and so on [2]. We will be drawing on their list in developing musical mood-based game prototypes.

In this paper, we propose a method of adding the concept of musical mood to drive gameplay. Here, we describe our approach to adapting open source software library CLAM (C++ Library for Audio and Music) for the iPod, and explain some of the methods used for developing musical mood extraction algorithms. Our engine is still in prototype stage, so we describe here an early game that we have built to employ user-selected song frequency as one parameter of mood.

II. TECHNICAL DESCRIPTION

The Apple iPod makes a logical choice for a music–driven game, since players will already have their own musical selections stored on the device, it is a popular device for both music listening and casual gaming, and there is ample support for game developers. We are using a 4th generation iPod Touch (iOS 4.3.3). We ported portions of the existing CLAM C++ library to the device. CLAM is an audio signal extraction and analysis library with a strong tonal analysis implementation. We have opted to use elements of the CLAM library alongside existing hardware-optimized frameworks. Specifically, Apple's Accelerate framework contains many DSP calculations necessary for audio and mood feature extraction, and so these are favoured over CLAM's existing FFTW (Fastest Fourier Transform in the West)-based analysis.

Our approach to mood extraction is currently focused on two areas: 1) descriptive tonal analysis, both on the instant tonality (chord) and the overall tonality (key) of the piece, and 2) a tempo and time signature analysis, which follows a similar implementation but filters high frequency ranges and performs basic onset detection on the resulting low frequency spectrum. An overview of this algorithm is presented in Fig 1.

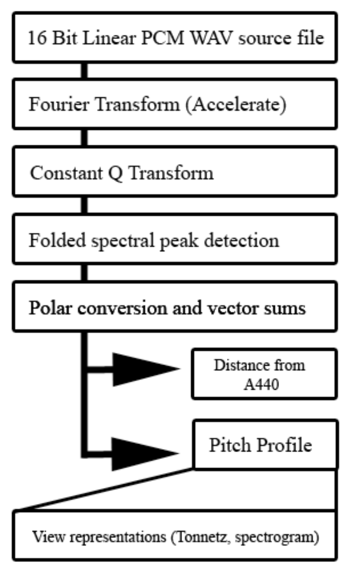

Fig. 1 – An overview of an audio feature extraction algorithm being implemented for research on music-driven games.

The audio file is converted to 16 Bit Linear PCM WAV, an uncompressed digital music format. The Accelerate Framework's vDSP functions are used to perform a Forward Fast Fourier Transform on a split complex vector representation, and the output returns a split real vector. In this form, a frequency can be calculated for each bin by squaring the bin magnitude.

The Constant Q Transform, where Q is the ratio of centre frequency to bandwidth, is implemented following CLAM's approach and that described by Brown and Puckette [3]. The transform range used is the audio band from 55Hz to 8000Hz. This frequency range is chosen because it encompasses most frequencies produced in contemporary popular music and vocals, and because it is less computationally expensive to perform analysis on a narrower band than on the entire audible range. It is common for our chosen range to be subdivided further; the bass range usually existing at 55-300Hz (the fundamental frequencies for most instruments and vocals occur here), the midrange at 300-2400Hz (harmonic frequencies responsible for much of the detail in a musical piece), and the high range at 2400-8000Hz (most brass and orchestral instruments reside in this area of the audio band).

III. Developing Mood Detection: Methods

Musical mood is largely a culturally specific phenomenon. While certain aspects of musical mood have a universal element to them (due primarily to bio-acoustic properties: for instance, low bass is heard as threatening), most of the semiotics of music are due to cultural convention. In the West, this convention has evolved over centuries through a combination of folk music and European art music. We have chosen to use these Western conventions for our purposes of determining mood. In other words, our mood detection may only work on Western musical styles. We developed a database of musical elements (harmonic content, instrumentation, tempo, articulation, timbre, pitch range, and dynamics) that contribute to each mood through a number of different related projects.

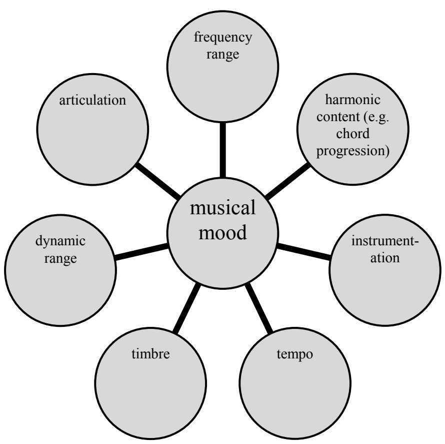

Fig 2. Quantifiable Elements of musical mood

A. *Distributed Classification Games*

Distributed classification is a method of collecting a large number of responses of multiple users, commonly used for meta-data. Users tag media objects with text keywords in a free-association fashion. Tags can then be combined into non-hierarchical groups of associated terminology. In order to engage the audience and increase the amount of tags collected, tagging games can be created [4]. Approaches that encourage players to tag musical data have been developed [5][6]. We developed two distributed classification games and collected data over two years [7].

B. *Sheet Music Analysis*

Photoplay scores are the sheet music used by the pianist that would accompany so-called "silent" films. These short pieces of music were typically arranged according to genre of movie and mood, and were accompanied by keywords that would represent the intended type of scene or mood associated. For example, the 4-bar *Mysterioso* by Julian Rutt was labeled "burglars and creepy business" [8]. We converted the sheet music into usable MIDI files and stored the keywords as metadata associated with the file.

C. *Existing Databases*

We have drawn on existing databases that categorize popular songs according to mood, such as Moodstream, StereoMood, AllMusic and Aupio. In Aupio, for instance, songs are categorized according to mood keyword. StereoMood uses its own musical mood categorization to create mood-based playlists for players, although to our knowledge it uses no extraction techniques, but rather has categorized songs with keywords manually. We obtained as many MIDI files as possible of these songs and stored the keywords as metadata in a database.

D. *Artificial Neural Networks*

Using the MIDI files and affiliated metadata, we used the MIDI Toolbox plug-in for Matlab to develop artificial neural networks to scan and compare tagged MIDI files [9]. A neural network uses software to explore data and search for patterns, such as if there are any melodic or harmonic patterns in all of the 'sad' songs. MIDI is limited in that it cannot represent timbre, spatialization or voice. However, we were able to explore patterns of harmony, rhythm and melodic contour within given mood groups.

IV. Creating The Mood Engine

The end result of this data collection is a large database of musical traits of potential influence on mood. We say potential of course, because individually each of these elements cannot in and of themselves indicate a mood. For instance, while musical mode is often cited as a determinant of mood in Western music (major = happy, minor = sad), this is certainly not always the case, particularly in folk music (most sea shanties are minor mode, for instance). However, when combined with other factors—tempo, dynamic range, and so on—we suggest that musical mood can be anticipated with somewhat reasonable accuracy. Simply put, "slower tempo + minor key + narrow dynamic range" for instance is more likely to contribute to a sad feeling than any of these factors individually.

Our current musical mood engine is in its early stages and is not yet ready for full explication here. While we continue to develop and create a more accurate algorithm to determine musical mood, we have been working on demonstration implementations for iPod.

V. Our First Prototype

In order to test our ability to determine musical mood as a gameplay element, we developed a prototype game, called *Frequency Faller*.

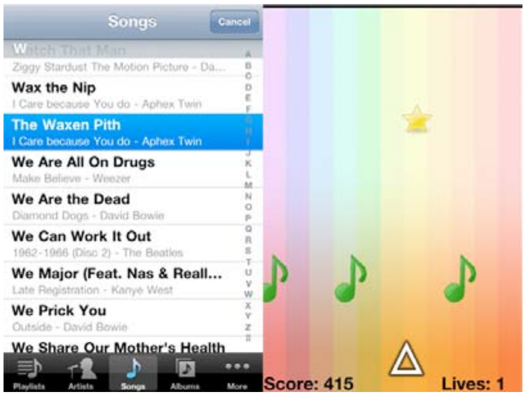

Figure 3. iPod Library selection and screenshot of game play in Frequency Faller

Written in Objective-C, *Frequency Faller* uses the iPod Library Access API released in iOS 4.0 for media selection and conversion, a RemoteIO Audio Unit for playback, and a render callback that passes blocks of sample data to the game's Fourier classes using a bespoke C++ interface. During game play this callback is executed approximately once every 23 milliseconds, and the block of audio is simultaneously passed to the device hardware output and analyzed on separate threads.

The game requires user input from the device's built-in accelerometer to navigate a character (in this instance, a triangle) around falling obstacles (in this instance, musical notes) with the goal of collecting point bonuses (the star in the image above). The positions of the graphical music notes are determined by the frequency content in the audio band of the chosen musical track. The calculated frequency is scaled linearly along the X-axis from 55-8000Hz, and as the player progresses, the naïve algorithm indexes the bin magnitudes and can return the 3 highest bins in this index, resulting in a basic form of polyphonic pitch detection.

Our first iteration of our prototype game currently uses frequency range to drive game play, however the rate of advancement in game play is held constant. Future iterations of this game will account for the other elements of musical mood described in Section III, and will cross-reference these elements to our mood database to determine mood. We anticipate, for instance, that the combination of musical mood with frequency could invert the positive and negative objects falling, alter colors, and so on.

VI. NEXT STEPS AND FUTURE WORK

There are a number of musical elements that contribute to the concept of musical mood, as discussed (harmonic content, instrumentation, tempo, articulation, timbre, pitch range, and dynamics). Our present iteration only takes into consideration frequency, and thus cannot yet call on our database of musical mood, but rather is limited to a quite mechanical extraction of auditory elements.

We anticipate some future difficulties in our mood-based gaming engine for the iPod. The iPod is limited in its processing and memory, and simultaneously analysing music and generating graphics/gameplay will likely introduce some lag. A pre-scan of the player's song will solve this problem, but if the player wishes to change music in the middle of gameplay, this may cause problems. Games that procedurally generate or alter content based on music can be limited. Certain types of music work better than others in these types of games, of course, particularly when it comes to beat mapping—abstract ambient music tends to throw off these types of games, and so the player's experience can vary greatly, depending on their choice of music. Using musical mood and other musical parameters is one way to overcome this problem, but how effective it will be in generating interesting gameplay remains to be seen.

REFERENCES

[1] S. Rosso, G. Tzanetakis and B. Gooch, "Adapting Personal Music for Synesthetic Game Play." *FDG 2010*, June 19-21, Monterey, CA, USA.

[2] J. Arrasvuori, and J. Holm. "Background Music Reactive Games," *MindTrek 2010*, October 6th-October 8th 2010, Tampere, Finland.

[3] J. Brown, M. Puckette, "An efficienct algorithm for the calculation of a Constant Q Transform" June 16 1992 http://www.wellesley.edu/Physics/brown/pubs/effalgV92P2698-P2701.pdf

[3] L. von Ahn and L. Dabbish, "Labeling images with a computer game". *ACM Conference on Human Factors in Computing Systems*, 2004, pp. 319–326.

[4] E. Law, L. von Ahn, R. Dannenberg and M. Crawford, "Tagatune: A Game for Music and Sound Annotation" *Proceedings of the International Conference on Music Information Retrieval* (2007) http://ismir2007.ismir.net/schedule.html .

[5] D. Turnbull, R. Liu, L. Barrington and G. Lackriet, "A Game-Based Approach for Collecting Semantic Annotations of Music" *Proceedings of the International Conference on Music Information Retrieval* (2007) http://ismir2007.ismir.net/schedule.html.

[6] K. Collins, "Generating Meaningful Sound: Quantifying the Affective Attributes of Sound Effects for Real-Time Sound Synthesis." *Audio Engineering Society 35th International Conference: Audio for Games*, London, UK. February 11-13. 2009. London, UK.

[7] E. Rapée. *Motion Picture Moods for Pianists and Organists*. New York: G. Schirmer, 1942.

[8] O. Lartillot, P. Toiviainen, and T. Eerola. A matlab toolbox for music information retrieval. In Preisach, C., Burkhardt, H., Schmidt-Thieme, L., and Decker, R., editors, *Data Analysis, Machine Learning and Applications*, Data Analysis, Machine Learning and Applications: Studies in Classification, Data Analysis, and Knowledge Organization, pp. 261–268, Berlin, Germany. Springer. (2008).