My Mouse – My Music

Marc Conrad & Tim French
University of Bedfordshire, United Kingdom
{marc.conrad;tim.french}@beds.ac.uk

Introduction

Music and the computer have quite a lengthy history of association, as with other humanistic disciplines. Familiar synergistic uses include computer aided stylistic analysis (Bent and Drabkin, 1987; Pople, 2003), computer based or assisted composition and performance (for example pioneered by the CEMAMu: *Centre d'Etudes de Mathématique et Automatique Musicales*, and founded by Xenakis) as well as numerous synergistic applications such as musical score printing and the numerous musical notations prompted by the development of the computer such as 'DARMS' (Digital Alternate Realization of Musical Scores), or inspired by mathematical methods such as the famous work of Allen Forte (i.e. set-theoretic notation) and actualised through the use of the computer as a creative musicological tool.

Indeed, a vast area of intellectual and creative activities that broadly comprises MIDI-based musical synthesis now encompasses both popular and classical musical traditions - enabling the seamless integration of creative, analytic and musical performance activities. As well as serving useful creative purposes, MIDI based technologies have been harnesses for purely *pragmatic* purposes. Indeed, that was our original starting point: to develop a music generative set of algorithms for the purpose of exploring what might best be termed dynamic 'musical passwords' as an alternative to alphanumeric or biometric means of authentication (Conrad et al). Others have similarly harnessed MIDI so as to visualise complex data streams through sonification, including by bomb disposal teams (Franklin and Roberts, 2004).

During the development of our original authentication system centred firmly within the realm of Computer Science, rather than within the field of musical aesthetics or artistic creativity, we came to realise that the system actually exhibited *inherent* creative possibilities. This realisation led to our current project wherein we propose that a set of relatively simplistic music-generative algorithms can in fact be usefully harnessed by those with suitable talents for creative and artistic purposes. Later, we briefly outline one or many such creative possibilities - the use of images as stimuli ("hot-spots" in particular) for the creation of dynamic single or multi-player 'compositions' or rather performances, based on a set of relatively simplistic musical "raw" materials and driven using a ubiquitous device familiar to everyone: namely a computer mouse. One of our precepts was to develop the system in such a way as to facilitate public comment and enhancement - thus the system runs on any standard PC that supports audio and that runs the Java virtual machine environment that was used to build and activate the present system. Our intention now is to present a 'base' sonic events generative system and to invite contributions from those are active in the field of sonification and sonic events within the creative musical community. We are only able to present below a relatively brief outline of the main aspects of the system and to begin to consider its future potential – no doubt future possibilities for deployment and enhancement within various artistic – creative contexts would appear to be numerous, if not infinite.

Brief History of the 'My Mouse - My Music' Project

A sonic event generative program was developed by us earlier (Conrad et al, 2006). The main focus of our work at that time was the development and deployment of a pilot user authentication system based on a user's ability to select and subsequently recognise short musical fragments, generated by the system. These dynamically generated fragments are used as a viable alternative to alphanumeric passwords or indeed user biometric forms of authentication (Conrad et al, 2006). In order to allow for the dynamic creation of musical fragments a design decision was made to create 'sonic events' directly using mathematical functions in a deterministic way (rather than automatic score creation). The deterministic generative mechanism used to construct the events uses the weighted average of two or more functions. Thus, by adjusting the weights from zero upwards to one a continuous transition between the functions is made possible. This in turn leads to the generation of a sequence of closely related sonic events when moving from one function to the next. A mathematical explanation is given below.

The PC Screen as a Coordinate System.

The software package that produces the sonic events has as its core stepwise continuous functions that are themselves combinations of the graphs of cubic polynomials and constants. In the absence of any accepted normative terminology we herein coin these functions *'nice'* functions. Various techniques (as the source code is openly available at http://www.perisic.com/mousemusic we only describe the general principle here) ensure that these *nice* functions do not have too many jumps and exhibit moderate gradients. The nice functions are defined by a 'seed' s and two coordinates a and b. Each triplet (a,b,s) creates a unique, nice function $f_{a,b,s}$: [0..1] -> [0..1]. Two or more functions are used to combine to form a new function by the use of weighted averages. This is used in our application as follows: assume that we choose to use a coordinate system to describe the positions on a the computer screen (Figure 1) where the lower left corner is the point (0,0) and the upper right corner is the point (1,1). Any point on the screen can clearly be identified by coordinates (x,y) with x and y between 0 and 1. In Figure 1 the point (.25,.5) has been highlighted for illustration. For any point (x,y) we then define a new function $g_{x,y,s}(t) = (1-y)((1-x)\, f_{0,0,s}(t) + x\, f_{0,1,s}(t)) + y((1-x)\, f_{1,0,s}(t) + x\, f_{1,1,s}(t))$ as the weighted average of the four functions of the 'corners' of the screen at a point (x,y). Note, that whenever for two functions $g_{x,y,s}$ and $g_{u,v,s}$ the pairs (x,y) and (u,v) are close then the values $g_{x,y,s}(t)$ and $g_{u,v,s}(t)$ are also close for any given t. Thus, for a given seed s and a point (x,y) on the screen we are able to define a 'nice' function having the property that if two points are close the corresponding nice functions are close as well.

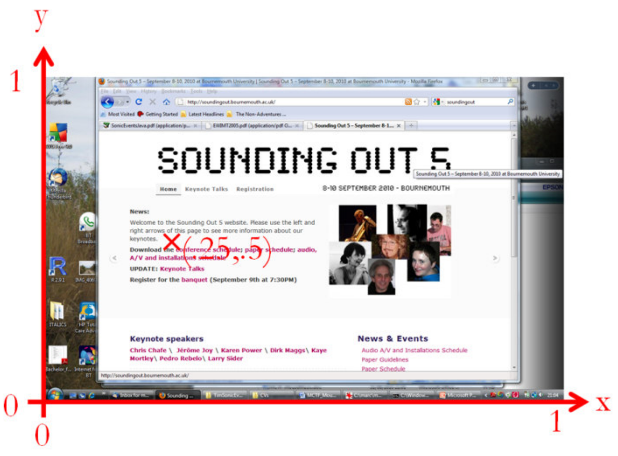

Figure 1 The Computer Screen as a Coordinate System

Translating Functions to Music

The procedure that translates mathematical functions into sonic events, i.e. the 'music' that is audible for a given point has been discussed in detail by Conrad et al (2006). Each sonic event is a short musical fragment that consists of five voices and an additional function that determines the timing of the production of the individual sounds. A voice is defined by three functions, a set of instruments and the pitch range. The functions are mapped to appropriate values through translation and scaling operators so that they provide suitable values for pitch, velocity and the selection of instrument. The [0..1] value domain of the 'pitch' function maps 0 to the lowest pitch in the range and 1 to the highest pitch. The values are then rounded to the nearest semitone of the harmonic chromatic scale. The 'instrument' function is used to decide which of the voice's instruments is to be used to render the sound at any given moment in time. Each voice is related to a collection of predefined instruments and only one instrument is active at any given point in time. The 'velocity' function determines the velocity used as defined in the

MIDI standard and implemented in Java. The velocity has a different meaning for different instruments; as a rule of thumb, a higher velocity creates a more accentuated initial attack characteristic.

Whilst the mathematical functions are purely deterministic, the resulting sounds may vary on different hardware platforms depending and how the Java implementation interacts with it. In particular the application uses the MIDI sequencer as provided by the *getSequencer()* method of the Java class *javax.sound.midi.MidiSystem*[1]. The exact nature of the sequencer used by the system are provided by the system properties.

The Mouse on the Move

The application runs as a background process and constantly re-checks the position of the mouse pointer. If the position has changed the sonic event that is being played is updated. This whole process is hence completely independent from any other software running on the computer. It is up to the user how the system is to be used creatively for the purposes of artistic, media and/or musical expression. For instance on a background image 'hot spots' of a chosen image could usefully be associated with various sonic events by the performer. In this instance (one of many possibilities for creative expression) different images could be used in real-time by performers to inspire radically different combinations of sonic-event performances. It is even possible to envisage the deployment of more than one sonic event player creating complex dynamic soundscapes using sets of suitable stimulus images as creative inspiration. Indeed the creative possibilities would appear to be endless.

Conclusion

From its emergence as a viable research tool in the humanities (actually from the late 1940's) the modern computer has been harnessed by those seeking to generate, analyse, synthesise, hence deepen our understanding of modern as well as more ancient musical forms. Sonic events comprise a kind of hybrid musical space: on the one hand the musical fragments (hence musical vocabulary) are relatively simplistic - even trivial. On the other hand their very simplicity means that listeners can easily recognise events and identify close variants. This was our starting point in relation to purely pragmatic aspects: to develop an alternative musical password system based on the generation of sonic events and their recognition by human subjects. We have since recognised that even such a limited musical vocabulary as we have chosen to use in the pilot system has expressive, aesthetic potential: for instance in the creation of a mouse driven performance, perhaps stimulated by an image or set of image hot-spots. This is not to say we are claiming that the system as presently formulated is in any way an ideal solution. The generative system as presently formulated offers ease of use and a high degree of hardware independence - and has at least some potential to be used for creative musical expression. Clearly, we are a long way here from the world of "high-art" or indeed the artistic and technical complexities of formal musical generative grammars or the complex and subtle sound-world of composers such as Xenakis.

Where we would in fact claim is to seek to locate the current system within a limited musical space that is pragmatic, with some strong future aesthetic possibilities - using a set of musical materials that are intentionally highly constrained. These constraints are a function of the algorithms used as well as the requirements of the Java MIDI run-time environment. Thus, the musical space that the system presently

[1] http://download.oracle.com/javase/7/docs/api/javax/sound/midi/MidiSystem.html

occupies is a hybrid space: originally pragmatic, but also with some definite expressive potential. How best to harness that expressive potential and how to set about tailoring the system in a creative / expressive musical direction without losing its "roots" (pragmatics, use of *nice* functions, hardware/software portability and ease of use) remains very much an open question. We thus warmly invite technical, algorithmic as well as creative suggestions from within the community of interest in the "*My Mouse - My Music*" system. The system, installation and usage instructions are available at http://perisic.com/mousemusic.

References

Bent, I., Drabkin, W. (1987). Analysis. *The New Grove Handboooks in Music.* Antony Rowe Publications, UK.

Conrad, M., French, T., Gibson, M. (2006), A Pragmatic and Musically Pleasing Production System for Sonic Events. In: 10th *International Conference on Information Visualisation (IV06),* London.

Franklin, K., Roberts, J. (2004), A Path based Model for Sonification. *In*: Ebad Banissi et al, 9Ed), *Information Visualization*,. IEEE Computer Societey, 865-870.

Pople, A. (2003), Getting Started with the Tonalities Musical Analysis Software. Available at: http://www.nottingham.ac.uk/shared/shared_music/documents/Getting_Started_with_Tonalities.pdf