2006

Metris: A Game Environment for Music Performance

Mark Havryliv
University of Wollongong, mh675@uowmail.edu.au

Terumi Narushima
University of Wollongong, terumi@uow.edu.au

Publication Details

This book chapter was originally published as: Havryliv, M & Narushima, T, Metris: A Game Environment for Music Performance, in Computer Music Modeling and Retrieval, Lecture Notes in Computer Science 3902/2006, Springer-Verlag, 2006, 101-109.

Metris: A Game Environment for Music Performance

Mark Havryliv and Terumi Narushima

Faculty of Creative Arts, University of Wollongong
Wollongong NSW 2522 Australia
mhavryliv@hotmail.com
tn649@uow.edu.au

Abstract. Metris is a version of the Tetris game that uses a player's musical response to control game performance. The game is driven by two factors: traditional game design and the player's individual sense of music and sound. Metris uses tuning principles to determine relationships between pitch and the timbre of the sounds produced. These relationships are represented as bells synchronised with significant events in the game. Key elements of the game design control a musical environment based on just intonation tuning. This presents a scenario where the game design is enhanced by a user's sense of sound and music. Conventional art music is subverted by responses to simple design elements in a popular game.

1 Introduction

In Metris the person playing the game is also a musical performer. The manipulation of sound events is an integral part of the game strategy. In the design of the sounds, an arbitrary choice has been made to synthesise bell tones based on a Japanese temple bell. The sound design is an extension of work done by one of the authors [1]; game design extends interactive strategies developed by the other author in 'Medium Racing'[1] and 'The Singing Jacket' [2], [3]. The sounds used in Metris have partials that are based on the natural harmonic series. Musical scales used are also determined by the harmonic profile of the bell.

The sound trajectory provokes a musical reaction to the player's performance in the game. In this way, a player is presented with an environment for improvisation that is based on the simple rules of the game.

2 Game Design and Composition Practice

A musical composition is like a game in that the rules and parameters to control the structure of an aesthetic experience are devised prior to its realisation in performance.

[1] Performed at Sonic Connections, University of Wollongong, Australia (2004) http://www.uow.edu.au/crearts/SonicConnections.html

In a musical work, the composer specifies how these rules and parameters should be realised by a performer over time and an ideal performance is a manifestation of the composer's artistic intentions. In a game, however, it is the player who determines its trajectory.

2.1 Game Design and Algorithmic Composition

Computer games are no different from other types of games in catering for a wide range of technical abilities. Metris provides an accessible environment for musical improvisation through the principles and interface of a game. The experience is intended to remain rewarding at all levels of play.

Work has been done by others to explore the audio synthesis and manipulation possibilities of using game control information. This is often realised as adaptations of existing game engines to create sound worlds through which a performer can navigate [4], [5]. Sound is controlled by both interpretation of input events and progression through the world. This process characterises the relationship between the game performance and audio creation as a one-way data stream. Other attempts at generating sound from visually represented algorithms include processes based on Cellular Automata (CA) behaviour [6], [7], [8], [9], [10], [11], [12], [13]. Like most CA visualisations, Tetris is realised as cell behaviour on a grid and as such could be considered a candidate for similar methods of sonification. The timbral and harmonic characteristics of the resultant music, however, are often limited by realisations in MIDI and an arbitrary mapping of algorithmic data to a twelve-tone equal temperament. Moreover, in both cases, mapping is mono-directional and the musical result is not based on any attempt at interpreting an individual's reaction to the relationship between audio and visual.

Metris solves these problems by controlling real time aspects of the sound through the game play itself rather than its visualisation. Tuning principles of just intonation are adapted to determine the timbre of the instrument as well as the pitch material. The player learns to recognise connections between game movements and musical outcomes. Game decisions are based on the player's reaction to both visual and audio.

2.2 Game Actions and Musical Reponses

The game responds musically in a number of ways. While the musical output is a direct result of game play, the organisation of pitch and timbre creates something larger than the memory of individual events. Musical responses to events are crafted so that the sound is affected in different ways depending on the nature of the game event: minor events affect the sound in a subtle, repeatable way (such as microtonal pitch bends when a game block is rotated); major events have a more dramatic effect on the entire sound design (such as a distinct texture modification when a row of blocks is removed).

The player has far greater control over the total sound because of the way different events are interpreted than if game actions were mapped to discrete musical responses. Game design manages individual actions within the context of the entire

game; a cohesive relationship is developed between the nature of game play and the creation of sound. This is a reflection of our use of game design principles to control the player's interaction rather than interactive paradigms traditionally found in contemporary art music.

3 Real Time Bell Synthesis

In order to have control over the tuning and textural composition of a bell at any time during the game, a real time synthesis process was developed in Java. Java was chosen for its portability, but requires a more careful management of resources than a lower level language like C or C++. Real time audio techniques developed for the Pocket Gamelan project [14], [15], where J2ME code manages audio performances on mobile phones, form the core of the audio generation in Metris. For performance and control reasons, no third-party Java audio packages are used.

A bell is defined as a collection of harmonically related sine tone generators with an associated collection of amplitude envelopes. This replicates earlier attempts to create digitally synthesised bells using Csound where an instrument object encapsulates the partial frequencies and amplitude relationships of a particular bell [16]. The frequencies are taken from the five most prominent partials of the harmonic spectrum of a Japanese temple bell, namely the 2^{nd}, 5^{th}, 7^{th}, 9^{th} and 11^{th} harmonics [17]. These harmonics are also used to generate the tuning system for the game. The envelopes must match the sampling grain of the generators to create an accurate synthesis.

Whenever a bell is struck, it is added to the audio output stream as part of a collection of any currently sounding bells. When a bell is read, each oscillator uses a current version of its frequency and amplitude envelope to determine the correct value. The amplitude envelope is stored as a set of floating point values in the same way a *linseg*[2] is constructed in Csound. Because this is applied to the sample 'on the fly' rather than as a pre-computed array, the envelope, along with the frequency of any of the partials, may be changed at any point in the duration of a bell tone. The envelope is then applied to each of the partials at the moment each bell rings.

4 Metris Music

4.1 Tuning

The same harmonics used to generate the timbre of bell sounds were used to determine pitch material for the game. Each of the seven Metris game blocks consists of four squares. In keeping with this, a seven-note just intonation scale was generated by connecting two four-note chords (tetrads) based on harmonics found in the bell timbre.

[2] Following the conventions for *linseg* opcode defined in Csound http://www.csounds.com

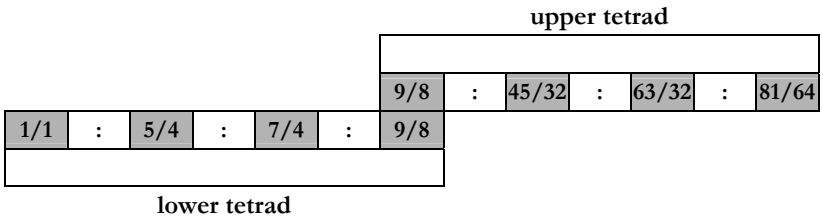


Fig. 1. Construction of a seven-note bitetradic scale generated from harmonics 2:5:7:9. The scale consists of two conjunct tetrads containing the same intervallic structure

Fig. 1 shows the formation of a scale generated from harmonics 2:5:7:9. The fractions represent musical intervals in just intonation by indicating the separation between any two pitches on the natural harmonic series: the numerator is the higher pitch and the denominator is the lower pitch. For example, 5/4 represents an interval of a major third; this occurs between harmonic 5 on top and harmonic 4 below, as can be seen in Fig. 2 which shows the first 16 pitches of the natural harmonic series of C_2. Similarly, 7/4 is a harmonic seventh; 9/8 is a major whole tone, and so on. The unison is represented as a fraction where the two pitches are identical, usually 1/1.

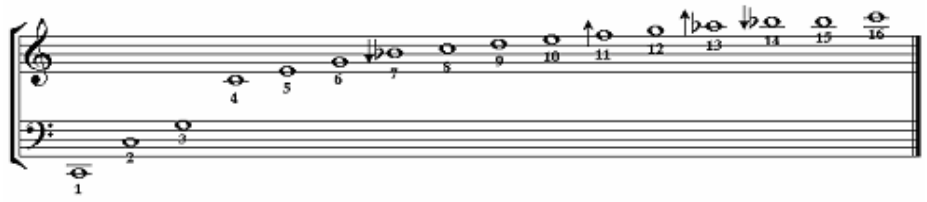


Fig. 2. Natural harmonic series (first sixteen harmonics) on C_2 [18]

The pitches of the two tetrads shown in Fig. 1 are normalised to produce the scale shown in Table 1. The ratios that appear in the second column of the table represent points on the natural harmonic series as described above. The labels that appear in the third column are the historical names given to intervals as found in many musical traditions. The linear factors that appear in the fourth column are calculated by dividing the numerator by the denominator of the ratio. These linear factors are then used to calculate the frequencies of each pitch in relation to the unison of the scale; for example, if Pitch 0 is 440Hz, Pitch 1 will be 495Hz (440Hz x 1.125).

Table 1. Intervals of bitetradic scale with generators 2:5:7:9

Pitch	Ratio	Interval above tonic	Linear factor
0	1/1	unison, perfect prime	1.000000000
1	9/8	major whole tone	1.125000000
2	5/4	major third	1.250000000
3	81/64	Pythagorean major third	1.265625000
4	45/32	diatonic tritone	1.406250000
5	7/4	harmonic seventh	1.750000000
6	63/32	octave-septimal comma	1.968750000

In Metris, each pitch of the seven-note scale thus derived is assigned to one of the game blocks as shown in Fig. 3. In this way, by using a scale generated from the same harmonics found in the overtones of the bell, the instrumental timbre is reinforced by the tuning in which it plays.

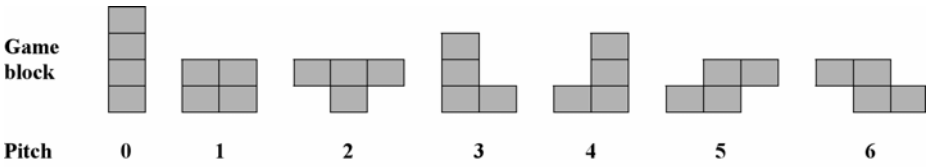


Fig. 3. Pitches 0 to 6 of bitetradic scale allocated to each of the Metris game blocks

The lower and upper tetrads in Fig. 1 are identical chords that consist of the same intervallic structure but transposed so that the highest pitch of the lower tetrad doubles as the lowest pitch of the upper tetrad. This method of constructing a scale is a variation on John Chalmers' tritriadic scales [19], [20], [21]; such a scale is labelled a *bitetradic* scale following the naming system used in Scala, a tuning software program developed by Manuel Op de Coul [22].

4.2 Modulation

The Metris game screen consists of 20 rows divided into four subsections of five rows each. These subsections determine regions of harmonic modulation in the game. If a block lands somewhere on the bottom five rows of the screen, notes from the original bitetradic scale with generators 2:5:7:9 (Table 1) will be played. If a block lands in a different section of the screen, pitches from another harmonic region will be played.

Modulation occurs from the original scale to other just intonation bitetradic scales built from different harmonic generators as shown in Fig. 4. Each step of the modulation introduces new intervals with higher prime-numbered harmonics[3] [23]. The

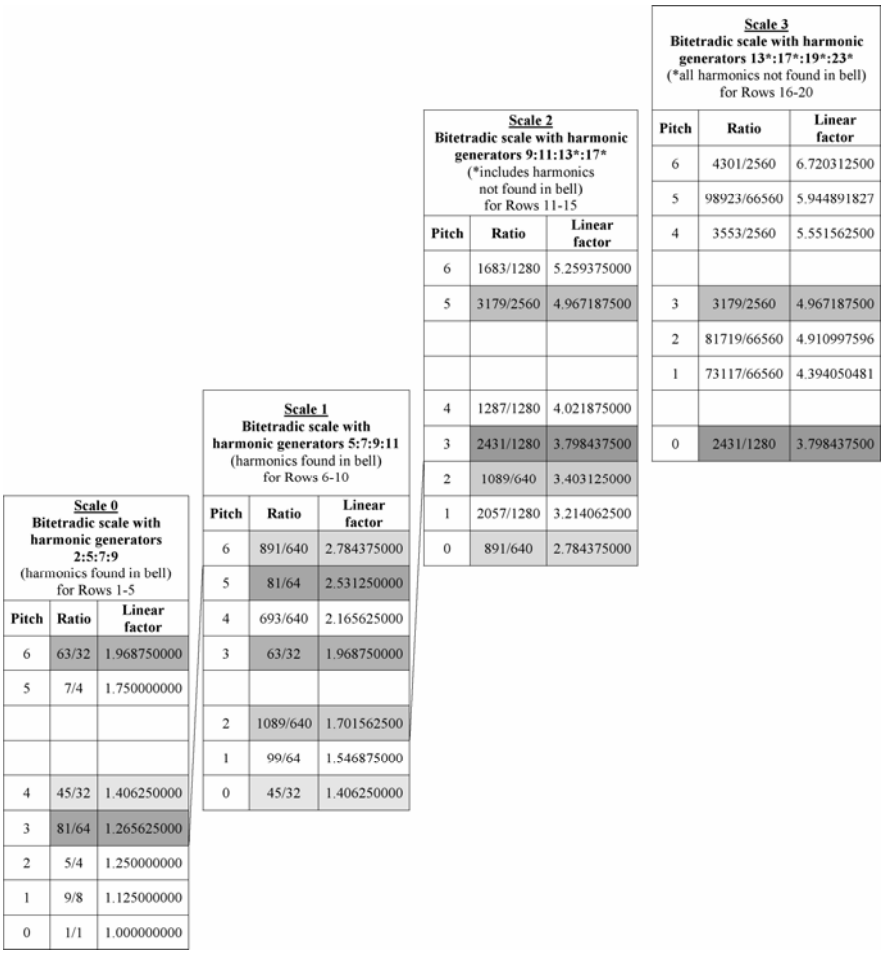
[3] Harry Partch used the term *limits* to describe an audible characteristic of just intonation tunings that is related to specific prime-numbered harmonics. A scale belonging to a particular

original scale (bottom left of Fig. 4) and the first modulation are generated from harmonics that are present in the overtone series of the bell (harmonics 2, 5, 7, 9 and 11). The compatibility between the tuning of these scales and the timbre of the instrument reinforce each other, an idea demonstrated by William Sethares using scales derived from spectral analyses of real sounds [24]. Furthermore, the musical design of Metris includes the possibility of using scales that are not necessarily compatible with the harmonic profile of the bell. This is explored in the second and third modulations: Scale 2 includes harmonic generators that are foreign to the overtone series of the bell (harmonics 13 and 17) and Scale 3 is built from harmonic generators that are entirely unrelated to the harmonic profile of the instrument. When a game block lands in these higher harmonic areas, beating and difference tones are heard as a result of the incompatible tuning and timbre.

Pitches for all the modulations are represented in Fig. 4 as ratios and frequencies relative to the tonic of the original Scale 0. Each new scale in the series of modulations has a number of pitches in common with the previous scale; these pitches (and their octave displacements) are indicated in the diagram as shaded boxes. As the modulations move further away from the tonic, more complex ratios are involved and the intervals formed have no historical precedents. While playing the game, the user has the opportunity to move around different modulation regions depending on where they choose to land the game block.

---

prime limit has a distinctive hue that makes it aurally distinguishable from scales with other limits.



Scale 0
Bitetradic scale with harmonic generators 2:5:7:9
(harmonics found in bell)
for Rows 1-5

Pitch	Ratio	Linear factor
6	63/32	1.968750000
5	7/4	1.750000000
4	45/32	1.406250000
3	81/64	1.265625000
2	5/4	1.250000000
1	9/8	1.125000000
0	1/1	1.000000000

Scale 1
Bitetradic scale with harmonic generators 5:7:9:11
(harmonics found in bell)
for Rows 6-10

Pitch	Ratio	Linear factor
6	891/640	2.784375000
5	81/64	2.531250000
4	693/640	2.165625000
3	63/32	1.968750000
2	1089/640	1.701562500
1	99/64	1.546875000
0	45/32	1.406250000

Scale 2
Bitetradic scale with harmonic generators 9:11:13*:17*
(*includes harmonics not found in bell)
for Rows 11-15

Pitch	Ratio	Linear factor
6	1683/1280	5.259375000
5	3179/2560	4.967187500
4	1287/1280	4.021875000
3	2431/1280	3.798437500
2	1089/640	3.403125000
1	2057/1280	3.214062500
0	891/640	2.784375000

Scale 3
Bitetradic scale with harmonic generators 13*:17*:19*:23*
(*all harmonics not found in bell)
for Rows 16-20

Pitch	Ratio	Linear factor
6	4301/2560	6.720312500
5	98923/66560	5.944891827
4	3553/2560	5.551562500
3	3179/2560	4.967187500
2	81719/66560	4.910997596
1	73117/66560	4.394050481
0	2431/1280	3.798437500

Fig. 4. Modulation regions in Metris: pitches from different scales are heard depending on where a game block lands. Ratios and linear factors are represented in relation to the tonic of Scale 0

5 Game Play and Audio Responses

Tetris has a limited number of options for controlling the performance of the game. A player can rotate the block either clockwise or anti-clockwise to place it in an effective way, and can move the block along the x-axis of the grid. The movement of the block downward is controlled by the speed of the game and is usually related to the level of difficulty, although a player can accelerate the downward progression of the block.

The game has different levels of achievement. At its most basic, the objective is to remove lines of blocks by filling an entire row. More points are awarded if more than one row is removed at a time. This introduces the possibility for daring game play and the soundtrack is designed to reward the player accordingly, as outlined below.

5.1 Rotation

When the player rotates a game block a copy of the fundamental partial of the previously struck bell is created and its frequency is adjusted +/- 3Hz. The direction of the microtonal pitch shift is determined by the direction of the rotation: A 90°clockwise rotation raises the pitch by 3Hz and a counter-clockwise rotation lowers the pitch by 3Hz.

5.2 Row Completion

When the player succeeds in completing an entire row of blocks, a sine tone texture bank (20 random sine tones) with a range proportional to the number of rows removed is played, thus destabilising any remaining tones.

5.3 Controlling the next block

Block creation in Tetris is usually determined by a random algorithm. Metris, however, recognises that a player may intuitively or otherwise wish to control the pitch selection in their playing. This is enabled by using the distance between the last two block placements as a percentage of the total screen size, then using that as an index to select the next block. The only exception to this is when a block lands and it overlaps with the previous block, in which case the same block is created. This was decided because the authors found repeated blocks would illuminate other aspects of the game function.

6 Performance Scenarios

The performance presentation of Metris can take many forms. One possible scenario involves Metris responding to sounds made by an instrumentalist. Metris has also been adapted for a multi-player format called Battle Metris.

6.1 Battle Metris

Battle Metris adds to Metris by allowing one player’s game play to affect another player. The music production system is adapted to allow the sonic realisation of the

conflict between players. The new sound represents a player's fortunes in the game and thus links the players' and audience's perception of the progression of the game.

When a player (A) removes enough rows to add them to the bottom of their opponent's (B) screen, player B's soundtrack is altered. A signal created from the amplitude-modulated sum of the partials (the *AM signal*) is added to the audio mix.

The AM signal is implemented in Battle Metris by multiplying all the partials from each bell. However, the amplitude envelopes of the partials are ignored; a partial is included in the signal multiplication at full amplitude until it is due to end. Removing the effect of the envelopes creates a significant distinction between the Japanese temple bell sounds and the AM signal.

The volume of the existing signal (the sum of the bells) is decreased to accommodate the volume of the AM signal, which is determined by the number of rows added. The partials used to create the AM signal no longer implement their amplitude envelopes; this creates a rich, *oppressive* sound. The oppressive quality refers to a sine tone with a slowly rising frequency multiplied with the AM signal, imitating the effect of the endless glissandi of Risset's implementation of a Shepard tone [25]. In the context of this game, the aural effect of the endless glissandi is similar to that of a motivator like a count-down timer in a typical digital game, in that it puts the player under pressure.

Player B can only decrease the volume of the AM signal in their mix by removing rows. The rich set of partials in the AM signal sets up beating patterns with the output of player A's channel. This is sonically exciting for the audience and player A; experience has shown it increases the level of stress in player B, as their immediate focus shifts to reducing the level of the AM signal in their mix. The oppressiveness of the AM signal complements the addition of rows as a negative effect on player B's performance.

7 Acknowledgments

The prototype of Metris was an extension of a Java version of Tetris by Per Cederberg [26].

References

1. Narushima, T.: Tritriadic Chimes: Bells in Just Intonation (sound installation). MicroFest 2001, Pomona College, Claremont CA (2001) http://www2.hmc.edu/~alves/microfest2001schedule.html
2. Schiemer, G., Havryliv M.: Viral Firmware: What Will I Wear Tomorrow? In: Australasian Computer Music Conference 2004. Victoria University of Wellington (2004)
3. Schiemer, G., Alves, B., Taylor, S.J., Havryliv, M.: Pocket Gamelan: Developing the Instrumentarium for an Extended Harmonic Universe. In: International Computer Music Conference 2003. Singapore University (2003)
4. delire + pix "q3apd" http://selectparks.net/archive/q3apd.htm
5. Todorovic, V. "*tadar.game music" http://www.tadar.net/

6. Brown, A., Sorenson, A.: jMusic: Music Composition in Java http://jmusic.ci.qut.edu.au
7. Chen, H.S.: Generation of Three-Dimensional Cellular Automata. In: Generative Art 2003. Politecnico di Milano University (2003)
8. Elliot, J.: Transmusic: Cellular Automaton Music (2001) http://jmge.net/camusic.htm
9. Millen, D.: An Interactive Cellular Automata Music Application in Cocoa. In: International Computer Music Conference 2004. Miami University (2004)
10. Miranda, E.R.: CAMUS: A Cellular Automata Music Generator (2002) http://website.lineone.net/~edandalex/camus.htm
11. Reiners, P.: Cellular Automata and Music: Using the Java Language for Algorithmic Music Composition (2004) http://www-106.ibm.com/developerworks/java/library/j-camusic/
12. Reiners, P.: Automatous Monk: The Cellular Automata Music Composition Program (2004) http://www.automatous-monk.com
13. Talmudi, A.K.: Evolving Decentralized Musical Instruments Using Genetic Algorithms. In: International Computer Music Conference 2004. Miami University (2004)
14. Schiemer, G., Havryliv M.: Pocket Gamelan: A Pure Data Interface for Mobile Phones. In: New Interfaces for Musical Expression 2005, British Columbia University (2005)
15. Schiemer, G., Havryliv M., Sabir, K.: Pocket Gamelan: A J2ME Environment for Just Intonation. In: International Computer Music Conference 2004. Miami University (2004)
16. Narushima, T.: Composing for Carillon: Exploring the Relationship Between Tuning and Timbre (MMus thesis) (2003)
17. Obata, J., Tesima, T.: Experimental Investigations on the Sound and Vibration of a Japanese Hanging-Bell. In: Japanese Journal of Physics 9 (1933-34) 49-73
18. Doty, D.: The Just Intonation Primer: An Introduction to the Theory and Practice of Just Intonation. 2nd ed. Just Intonation Network, San Francisco (1994) 13
19. Chalmers, J.: Tritriadic Scales with Seven Tones. In: Xenharmonikôn 9 (1986) 3-20
20. Chalmers, J.: Tritriadic Scales with Seven Tones, Part Two: Derived Forms and Structural Properties. In: Xenharmonikôn 10 (1987) 20-30
21. Chalmers, J.: Tritriadic Scales with Seven Tones, Part Three: The M->T and D->M Matrices. In: Xenharmonikôn 12 (1989) 40-68
22. Op de Coul, M.: Scala Home Page (2004) http://www.xs4all.nl/~huygensf/scala
23. Partch, H.: Genesis of A Music: An Account of a Creative Work, its Roots and its Fulfilments. 2nd ed. Da Capo Press, New York (1974)
24. Sethares, W.: Tuning, Timbre, Spectrum, Scale. Springer, London (1998)
25. Shepard, R. N.: Circularity in judgments of relative pitch. In *Journal of the Acoustical Society of America* 36 (1964) 2345-53
26. Cederberg, P.: Online Tetris Game (2005) http://www.percederberg.net/home/java/tetris/tetris.html