Pocket Gamelan: a J2ME environment for just intonation

Gregory Schiemer  
*University of Wollongong, schiemer@uow.edu.au*

Kenny Sabir  
*University of Wollongong*

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

Publication Details  
Pocket Gamelan: a J2ME environment for just intonation

Greg Schiemer, Kenny Sabir, Mark Havryliv
Faculty of Creative Arts, University of Wollongong
schiemer@uow.edu.au
kenny@elephanttrakz.com
mhavryliv@hotmail.com

Abstract
This paper describes on-going exploration of tuning systems through development of mobile instruments appropriate for the audition and performance of music composed in just intonation tunings. The project is a response to a transformation in computer music brought about through the introduction of wireless technologies and is motivated by a desire to enable performance of music based on just intonation using hand-held instruments played by large numbers of non-expert players. Handheld technology offers the promise of new forms of musical interaction between people with development of musical applications focused on new modes of group expression that involve non-expert performance. The project seeks to take advantage of the global availability of this technology yet retain a tuning vocabulary that represents the legacy of many musical epochs and traditions.

1 Introduction
The Pocket Gamelan project seeks to develop a software prototype for a set of mobile musical instruments based on the java programming language using a mobile technology known as Java 2 Micro Edition (J2ME). This technology is used in hand-held appliances such as palm pilots and mobile phones and allows communication between a web server and multiple clients.

The principal motivation for the Pocket Gamelan project has been a desire to develop an extensible interface that will support performance of music in various tuning systems. The diversity of available tuning systems, as reflected in the theoretical writings of Partch (1949) Chalmers (1993) and Wilson (1961, 1967, 1975a, 1975b, 1986), calls for such an extensible interface.

Aims. The aims of the Pocket Gamelan project are:

- to create a prototype network of mobile instruments for performing music free of the tuning constraints associated with conventional music performance interfaces;
- to use this prototype to explore and extend current developments in tuning theory using new performance paradigms.

2 Mobile performance environment
In performance scenarios associated with the Pocket Gamelan project, a musical ensemble consists of large numbers of mobile phone handsets each operated independently by a single user. Each unit functions either as a control device, a sound source or some combination of both. Musical applications for mobile devices are initially developed in a java desktop development environment running under Windows XP.

Program code is first developed on the desktop. It is then loaded into each mobile unit, along with the wave-tables required for audio synthesis, prior to performance. In effect the mobile handset becomes a generic hardware platform for software musical instrument applications which define the musical functionality of the handset. To implement additional performance scenarios this functionality can be redefined by developing new software instruments.

The prototype mobile instrument network will be one in which each mobile unit is easy to play, quick to learn and produces audible tones that are microtonally tunable. Each unit will be battery powered and able to take advantage of new developments in mobile digital computing. Each unit is a hand-held sound source that is played by pressing buttons. Players are free to move each sound source while performing.

The difference between music created using MIDI systems and music created using this technology is the degree of mobility and autonomy that a mobile instrument gives to each player. The extent to which this affects performance of music is limited only by the ways in which performers are allowed to move as part of the performance and the kinds of spaces where a performance is presented.

Whereas desktop computing tends to concentrate the means of producing music in the hands of a single user, mobility offered by this technology introduces new possibilities for musical interaction between members of an ensemble. ‘Gamelan’, in the title, is a musical metaphor for this kind of group interaction.
2.1 Tuning scenarios

Three performance scenarios made possible by this technology were outlined previously against a background of recent composition and musical instrument development in just intonation tuning systems (Schiemer at al. 2003).

Scenario 1. In scenario 1 each unit is used in a continuously variable tuning mode with pitches chosen by each performer.

Such a performance scenario is appropriate for new works realised in the manner of The Great Learning Paragraph 7 (Cardew, 1969) which was composed for a large amateur a capella vocal ensemble, where 'preferred intervals' chosen by each performer play a significant role in the process that shapes this work.

The term 'preferred' describes the intervals chosen by subjects in a listening experiment devised by Benade (1976) to determine whether listeners prefer just musical intervals.

A work that fits this scenario is Mandala 2, created for a set of electronic instruments called the Tupperware Gamelan which I built between 1977 and 1983 (Schiemer, 1999). The Tupperware Gamelan include a consort of 16 mobile instruments called UFOs. The UFO is a purpose-built battery operated hardware tone generator with a built-in loudspeaker and tuning adjustment. In performance each UFO is swung in a circular motion on the end of a cord to produce doppler shifted variants of the fixed pitch.

![Figure 1](image)

Figure 1. The control functions of the mobile hardware oscillator (left) in Mandala 2 will be replaced by software instruments running on mobile handsets; each oscillator is individually controlled; horizontal adjustment of navigation wheel (shown as a circle in the centre) raises or lowers the pitch; vertical adjustment raises or lowers the level.

Each UFO produces a sustained pitch that can be tuned anywhere within the range of one octave and a major sixth. The 16 UFOs are divided into four groups of pitch generators. Each group cover the same pitch range; all four groups cover a range of about five and half octaves.

Mandala 2 was created for an ensemble of 16 players each playing a UFO. Following verbal instructions given in the score each player in turn switches the instrument on, adjusts its pitch and then proceeds to swing the instrument for a prescribed duration. The performance lasts fourteen minutes. It has a strong theatrical element and requires a large space to accommodate the performance gestures associated with swinging instruments. The first performance of Mandala 2 took place in the Leichhardt Festival, Annandale Town Hall, Sydney November 15th 1981.

Like other works created for the Tupperware Gamelan, Mandala 2 was never recorded and can no longer be performed because only a few of the original hardware instruments are functional. A performance realised through software instruments that use J2ME technology will now make this possible.

![Figure 2](image)

Figure 2. The tuning applet for scenario 1 as it appears on the desktop environment shows four instantiations of a sine wave. In performance each oscillator is controlled by a dedicated handheld device as part of a larger consort of devices, each operated by one user.

Scenario 2. In scenario 2 the user activates musical sequences locally on the handset by pressing buttons. Locally triggered note events or sequences of note events are produced where the pitch of each note is pre-tuned. Like the java application code for scenario 1, tuning pre-sets are loaded along with the code for audio synthesis into each mobile unit prior to performance.

Tunings can be pre-set in one of two ways:

Firstly, by selecting intervals from a menu of historical just intervals that are part of Scala tuning analysis editor librarian software (Op de Coul, 1992); this is shown in figure 2a;

Secondly, by selecting a scale created in Scala's SCL format; Scala has a scales archive consisting of several thousand scales.

Just intervals and scales built from them can be auditioned using the Scale player shown in Figure 3. The Many intervals not found in Western music can be heard in the context of the original scales in which they occur. They may also be combined with other intervals to produce new scales.
Figure 3. The Scale Player as it appears on the desktop is a tuning application described in scenario 2 that allows a user to select preset tunings from the lexicon of just intervals found in Scala.

In an extension of scenario 1 intervals read from Scala via the Scale Player will be used to preset the maximum and minimum points in the pitch continuum.

After the tunings have been auditioned using the Scale player, tuning data along with preprogrammed sequences will be loaded into each mobile unit. The players will play preprogrammed sequences of preset pitches by pressing buttons. Sequences consist of melodic or rhythmic motifs.

We plan to use the press-button interface of the mobile phone to select new sequences, start, stop and continue playing and affect the tempo. Press buttons will also be used to invert musical motifs, and perform transposition in just intonation.

Sequencing will be created using standard MIDI software. A live MIDI player is being developed and some work has yet to be done to make this work with the scale player. What the press button interface lacks in individualistic musical expressive power we expect it will make up for by providing a layer of musical control that is higher than simple mechanical activation of single note events.

Scenario 3. In scenario 3, locally pressed buttons activate commands that affect other clients in the network. In the first two scenarios, user operations only affect sound on sound on other handsets the local handset; in the third scenario, user operations affect sound on other handsets just locally. This has yet to be developed.

The most promising musical outcome is the effect of transposition in just intonation where the pitches of a scale are separated by non-equal steps. Transposition in just intonation scales differs from transposition in equal tempered tuning. A motif transposed in equal tempered system will be heard on different degrees of the same scale. In just intonation it is possible that transposition will introduce pitches that were not present in the original scale.

Examples of transposition with just intonation scales can be found in Transposed Hexanies (Schiemer, 2001) where transposition using only the intervals of a six note combination product set scale (Wilson, 1986) has produced 16 and more new pitches that were not present in the original scales.

3 Software Development

We have undertaken some java development for desktop simulation. So far this supports only the first two performance scenarios. Further work has to be done of making the MIDI Player. Some of the J2ME developments has been done using Metrowerks Codewarrior Wireless Studio later switching to Eclipse with Sun Microsystems Wireless Toolkit.

Java classes used in the J2ME environment conform to two important protocols: Mobile Information Device Profile (MIDP, 2002) and Connected Limited Device Configuration (CLDC, 2003). Tuning implemented using the variable sampling increment technique with interpolation requires floating point arithmetic which CLDC1.0 does not support. CLDC1.1 was introduced in March 2003 to address this.

In J2ME devices the miniature J2ME operating system known as the K Virtual Machine – or KVM provides the vital link between java software and physical hardware. Initially we planned to adapt the KVM in order to implement the tuning algorithm in a low-level language.
such as C or assembler. However we aim now to incorporate the benefits of CLDC1.1 which include development entirely in java rather than at the KVM level.

In future we hope to develop the desktop environment as a server that can be used to download musical applications that service multiple clients via a multicast network. Additionally the server may also broadcast MIDI control information that is used to play sequences in any particular scale. It is intended that individual clients will also communicate control information back to the server. Eventually additional performance scenarios will emerge as individual clients will in turn affect the broadcast of MIDI control information.

5 Acknowledgments
This project was funded by an Australian Research Council Discovery Grant 2003-2005.

References
CLDC1.1 (2003) Connected Limited Device Configuration 1.1 Sun Microsystems, Santa Clara
MIDP2.0 (2002) Mobile Information Device Profile 2.0 Sun Microsystems, Santa Clara
Wilson, E. (1975a) “The development of Intonational Systems by Extended Linear Mapping” in Xenharmonikòn 2, 15 pages
Wilson, E. (1975b) “Bosanquet – A Bridge – A Doorway to Dialogue” in Xenharmonikòn 3, 13 pages

Proceedings ICMC 2004