Ethan Hein
Thesis Proposal
March 31, 2013
Advisor: Alex Ruthmann

**Thesis title**

The Drum Loop: a Self-Guided Tutorial System for Programming Dance Rhythms

**Introduction**

Dance music production software has never been more accessible. However, even “beginner-oriented” programs like Apple’s Garageband presume significant musical knowledge. Would-be dance producers who have access to formal music education are ill served by Eurocentric teaching methods and curricula. By and large, those wishing to learn drum programming are largely left to their own devices. This is unfortunate, because learning how to create beats does not only benefit electronic dance musicians. The ability to actively create and alter rhythms and to match their visual notation with the resulting sounds in real time sharpens the rhythmic abilities of any musician.

Most dance musicians must self-teach, and they face some significant obstacles in doing so. Nearly all music production tools are based on the keyboard/piano roll or multitrack tape/DAW paradigms. Beginners struggle to learn these visualization schemes on top of the musical concepts underlying them. A simplified and more intuitive interface would help to prevent frustration and abandonment of musical study. The author believes that a clock face metaphor is a more intuitive visualization scheme for the loops that form the basis of dance music drums. The present project consists of the design of a radial drum machine interface and a series of rhythm programming exercises. User evaluations of the program’s usability and effectiveness will inform the final design.

**Motivation**

Conventional music education focuses primarily on melody and harmony, and tends to neglect rhythm outside of the melodic context. Students of dance music must piece together guidance as best they can from percussion study resources, ad-hoc peer-to-peer learning and trial and error. A systematic and self-guided dance rhythm programming tool would fill this vacuum neatly.

Why take the study of dance music seriously? Music teaching should operate “from within authentic music making contexts” (Martin, 2012). Dance music is accessible enough that even beginners can produce it, thus “folding musical analysis into musical experience” (Marshall, 2010). Drum programming should be part of an overall widening of the curriculum to include non-western and non-classical music. Decanonization is a worthwhile goal for two reasons: 1)
students will remain more engaged when studying music they know and like, and 2) dance music is worthy of serious study in its own right. Digital production tools are not merely musical instruments; they carry with them an entire philosophy of music-making. The digital studio collapses composition, recording and editing into a single act, and expands the definition of the word “musician” beyond traditional performers and composers to include anyone with the patience and the will to learn the software and explore its possibilities (Thibeault, 2011).

Music visualization can significantly aid aural understanding. Visual reinforcement of the aural experience creates a dynamic multisensory feedback system, which improves the learning and retention of rhythmic patterns (Jylhää, Ekman, Erkut & Tahirolğlu, 2011). The best visualization methods take advantage of existing image schemas widely used to conceptualize music. These schemas include such bodily metaphors as containers, cycles, and the notion of center-periphery. Software interfaces and visualizations using body-centric image schemas will be easier to learn and remember (Wilkie, Holland, & Mulholland, 2010).

Drum programming interfaces generally follow either the standard MIDI piano roll format or the time-unit box system. There is some precedent for a radial “clock face” model for music loops. The turntable is one of the most immediate, and it persists in DJ software like Serato. Other clockface representations of repetitive music include Joy Mountford’s Soundscapes instruments (Levin, 2000) and the circular representations of performance data and music theory cited by Benadon (2007). That said, very little software exists that combines a radial loop visualization with an accessible time-unit box system.

Beginner-oriented software must place simplicity above all other considerations. Limiting options need not limit creativity. The more sophisticated music production tools can stifle creativity under the weight of option paralysis. A smaller feature set can be easily mastered, allowing users to quickly move on to musical expression (Magnusson, 2010). Rich inspiration for beginner-friendly user interface design can be found in music games like Guitar Hero, Dance Dance Revolution and the like. Such “rhythm games” must motivate players through many hours of disciplined practice, without any external instructors. Like all the best learning experiences, successful music games use progressive levels of difficulty and frequent rewards for success and persistence to produce a pleasurable feeling of flow. While the games do not enable much creative music-making, they have a well-documented ability to teach players how to listen actively and to think more like musicians (Gower & McDowall, 2012).

Rhythm games have one major limitation as music-making tools: self-expression is fundamentally incompatible with an unambiguous win condition. It seems more promising to create music tools with game-like interfaces, rather than music games per se (Rosenstock, 2010). Software will never be able to be a satisfactory judge of musical quality; users and/or their teachers must decide whether their musical ideas are satisfactory or not.
Goals and Method

The present project aims to create a program that teaches beginner musicians how to program dance beats, and to evaluate this program’s effectiveness. Through a series of interactive exercises that gradually build in complexity, users will gain a hands-on understanding of rock, pop, hip-hop and techno rhythms. Upon completion of the drum programming exercises, users will be better equipped both for study of traditional instruments and for more sophisticated sequencing software like Garageband, Logic or Ableton Live.

The user interface will use a radial time-unit box system rather than the more conventional linear “ice cube tray” model. The radial arrangement makes the symmetries and asymmetries of particular beats significantly more apparent to the eye. Stronger beats fall on the cardinal points, with weaker beats falling at “odd” angles. Syncopated patterns can be identified and understood easily by their differing symmetries from the main groove.

Exercises will take the form of generic hip-hop, rock, techno and Afro-Cuban rhythms, which the user may customize within constraints. The first programming exercises will constrain the user significantly. A drum pattern will be provided, with all parts locked down except for one instrument, likely a cowbell or shaker. The user will be free to place hits on this instrument where they like, with a guarantee that any location in the pattern will have a musically sounding result. Users advance when they are satisfied with their drum pattern. In each subsequent exercise, more instruments within the drum patterns are unlocked. The user becomes able to move the hi-hats, then the kick and snare. All patterns will have a locked kick drum on the first downbeat, along with a locked snare or handclap on the first backbeat. This guarantees that all drum patterns will sound basically musical, keeping the user in a state of pleasurable flow. At any time, users may switch to “free programming mode,” in which they can place drum hits anywhere. Patterns can be saved and exported as audio files.

Two user studies will be conducted, evaluating the program’s effectiveness in conveying musical knowledge and doing so in a pleasurable, flow-inducing way. The studies will primarily focus on beginners, but will include some musicians with diverse training backgrounds as well. The results of the first study will be used to refine the user interface and exercises. The second study will test the hypothesis that the radial presentation and progressive relaxation of constraints enhance users’ learning and retention of rhythmic concepts.

Work Plan

The project will consist of five main phases:

1. Design of the radial drum machine in Max/MSP.
2. Creation of the drum programming exercises.
3. Developing and implementing a user testing framework.
4. Revising the drum machine interfaces and exercises based on the results of user testing.
5. Conducting a second round of user testing, and analyzing the results.
Milestones and Dates

- January - March 2013: Gather sources, design drum machine
- March 15, 2013: Fully functioning drum machine
- May 1, 2013: Final UI design
- June 1, 2013: Create drum programming exercises
- July 2013: First interaction study
- August - September 2013: Refinement of UI and drum programming exercises
- October 2013: Second interaction study
- November 2013: Collate and analyze study data, complete written portion
- December 2013: Defense

References


