The Planetarium as a Musical Instrument

Dale E. Parson
Kutztown University of PA
15200 Kutztown Road
Kutztown, PA 19530-0730
parson@kutztown.edu

Phillip A. Reed
Kutztown University of PA
15200 Kutztown Road
Kutztown, PA 19530-0730
preed@kutztown.edu

ABSTRACT
With the advent of high resolution digital video projection and high quality spatial sound systems in modern planetariums, the planetarium can become the basis for a unique set of virtual musical instrument capabilities that go well beyond packaged multimedia shows. The dome, circular speaker and circular seating arrangements provide means for skilled composers and performers to create a virtual reality in which attendees are immersed in the composite instrument.

This initial foray into designing an audio-visual computer-based instrument for improvisational performance in a planetarium builds on prior, successful work in mapping the rules and state of two-dimensional computer board games to improvised computer music. The unique visual and audio geometries of the planetarium present challenges and opportunities. The game tessellates the dome in mobile, colored hexagons that emulate both atoms and musical scale intervals in an expanding universe. Spatial activity in the game maps to spatial locale and instrument voices in the speakers, in essence creating a virtual orchestra with a string section, percussion section, etc. on the dome. Future work includes distribution of game play via mobile devices to permit attendees to participate in a performance. This environment is open-ended, with great educational and aesthetic potential.

Keywords
aleatory music, algorithmic improvisation, computer game, planetarium

1. INTRODUCTION & RELATED WORK
This project grows out of the intersection of prior work in mapping two-dimensional computer board games to music [9,10] and the availability of a modern planetarium with a high-resolution digital projector, a high-quality surround sound system, seating for 85 people and very good acoustics [4].

The predecessor to the current game-as-instrument is a computer Scrabble® game extended to generate MIDI note and control information based on game rules and the progressing state of play [9]. The key musical aspect of that game is its mapping of statistical regularities in letter distributions in a game to statistical regularities in musical interval distributions in the scales being generated on MIDI channels. For example, in a typical player-configurable scale, common letters such as 'E' or 'A' map to consonant intervals such as the tonic or fifth in a channel's scale, less common letters map to less consonant intervals, and outlying letters such as 'Q' or 'Z' map to chromatic passing tones. Players select game tiles pseudo-randomly then impose lexical structure on them by creating words that map to unique sequences of intervals. Other parameters for mapping placed letters to properties such as meter, tempo and timbre are under more direct control of a performing musician via a graphical user interface.

The inspiration for writing a music-generating game specifically for the planetarium came at a computer music seminar in our facility on September 7, 2011 [7]. Dr. Michael O'Bannon of Atlanta provided excellent improvised video accompaniment for that event. Both the visual and acoustic properties of the facility were so good that the idea of adapting the graphical game-to-music approach to the planetarium presented itself as a natural next step.

Several related projects involving generating music from the rules and states of games have influenced this effort [1,5,13,15]. Two additional influences are work on creating audio productions that are portable among planetariums [2], and an overview of problems for spatial sound from a two-session morning on the subject at ICMC 2011 [3].

In the current project we retain and extend the successful properties of the Scrabble game while replacing the game itself with one designed for play on a planetarium dome with circular seating. We abandoned an initial attempt at a word game because the geometry of a planetarium renders words upside-down and backward for many viewers. We settled on a game metaphor more suited to a planetarium, that of an expanding universe of atoms, where 12 distinct atomic numbers map to 12 distinct intervals in the current scale on a MIDI channel. By regulating the range and distribution of atomic numbers that they inject into the expanding universe, players regulate the degree of consonance / dissonance in the generated music. More complex musical properties emerge from aggregate game state as discussed below. We have had two successful ensemble performances by a faculty member and students to date, one at the Kutztown University planetarium on March 20, 2012, and another at a regional computing conference in southern Pennsylvania on March 30 [6].

2. VISUAL ASPECTS OF THE GAME
Our ensemble performances have consisted of two students playing the game via wireless networking and one of the authors manipulating musical mapping parameters. Game play consists of injection of atoms into the universe and manipulation of statistical properties of their elements. In addition to conventional graphical controls for atom injection and music translation, performers use *live coding* in the Jython dialect of the Python programming language [8] to automate portions of play.

Figure 1 shows a screen shot of an early universe, before expansion, in which players have injected a variety of the 12 element types. An atom appears as a hexagon – we chose this geometry in order to accomplish symmetric tiling of the dome, naming the game *HexAtom*. Each distinct element type maps to a unique color. Multiple atoms occupying a single location appear to nest. Atoms currently being sounded by the music translator appear as filled hexagons, while unsounding atoms are hollow. Consequently, atoms blink on and off in time to their sounding (excitation). Musical tempo drives the rate of atom movement and flashing.

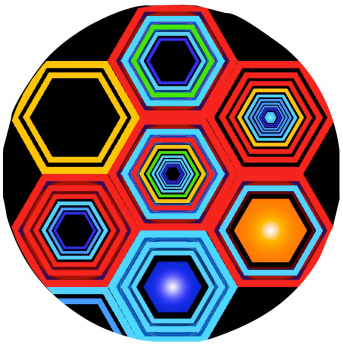

Figure 1. An early universe – The Big Bang

The most basic move by a player is injection of atoms into a location in the universe. Atomic parameters under player control include number of atoms injected at one time, distribution of their element types, location, direction of travel, and speed. There is a (typically) non-zero probability that an atom reaching the edge of the universe will cause its expansion towards a player-defined limit. Atoms can also disappear at the edge, deflect off of the edge or each other, fuse into heavier atomic types, fission into multiple smaller atomic types, and clump into geometric ensembles due to simulated gravity. Associated with each of the 12 element types is a set of probabilities for universe expansion, atom deflection, fusion, fission, and gravitational properties. In addition to injecting atoms, players can manipulate these distinct probabilities in order to guide evolution of the expanding universe. Players can use both manual interaction and live coding scripts to inject atoms, to configure their injection properties, and to configure probabilities of element properties. With evolution of the universe driven by the changes in state of individual atoms interacting with their immediate neighbors, the game is a form of cellular automaton [12].

Figure 2. A young universe in the midst of expansion

Figure 2 is a photograph of the March 20 game, at a stage when the universe was relatively young. Improvisational dynamics vary, and in this performance the players began with only element type 0 (mapped to the tonic) in a universe that was three atoms wide. Our "Big Bang" was a consonant, medium-tempo affair. Figure 1 shows a more harmonically complex Big Bang, with a larger range of element types (as seen by the range of colors) moving and sounding at a faster tempo. By the time our more conservative March 20 universe had progressed to the state of Figure 2, both atomic fusion and player actions had increased the range of element types and corresponding harmonic complexity to a moderate mix, with most generated notes lying within the scale of each MIDI channel being played.

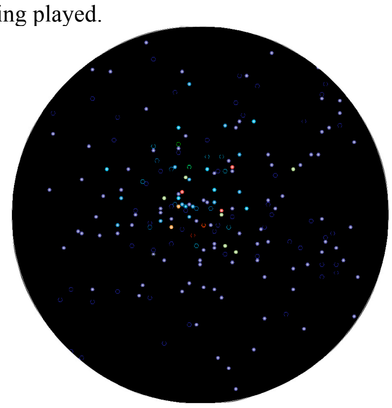

Figure 3. An aging universe evolving towards element 0

Figure 3 is a screen shot of an aging universe. Increased spatial separation of atoms lowers the probability of fusion – atoms must occupy a single location in order to fuse – thus increasing the relative probability of fission. Evolution tends toward a large number of element-0 atoms that typically map to the tonic. It is possible for players to counteract this tendency by increasing probabilities for fusion, decreasing probabilities for fission (in the limit to 0), and injecting a larger proportion of heavier element types into small regions of space. Increasing element probabilities that control gravitational clumping can create opportunities for fusion by keeping atoms closer together.

A typical player activity at the stage of Figure 3 is injection of atoms in unidirectional streams that are maintained by gravity, injection of atoms in geometric forms via live coding scripts using geometric formulae, and emergent geometric properties. The feel of play at this stage is reminiscent of Impressionist painting, where underlying precise geometric forms are smeared and blurred by atomic motion and evolution. One attendee at the March 20 event commented that in a later stage the game took on the appearance of fireflies in a summer night, with small atoms blinking on and off as their notes sounded. During testing one of the authors injected a series of streams of element 0 atoms in the six cardinal directions of the hexagonal layout that used gravity to draw all other atoms into their lines, forming an abstract six-petal flower. We have not yet been able to repeat this configuration.

We typically bring a performance to its end by increasing the elemental probabilities for fission and decreasing those for fusion while increasing the rate of atomic evolution, leading to a large decaying universe that devolves to the tonic and then fades away. In our second public performance we took an alternative approach of shrinking the universe back to a three-atom diameter, increasing element types and tempo to give a rapid restatement of the Big Bang before reversing the probabilities to cause rapid decay and fade out. The game has a wide range of probabilistic behaviors to be explored and mastered.

The overall perspective of the game-as-composition is one of *composition as a superposition of possible states*. A musical score is not a fixed set of musical objects with a probability of 100%. It is a variable set of musical objects with configurable probabilities. Changing probabilities via live coding is a major dimension of performance. Preparation for a performance includes writing small, custom scripts in the extension language that players can set in motion as they perform.

3. MUSICAL ASPECTS OF THE GAME

3.1 Atom to MIDI Mapping

The music mapper is an enhanced version of the Scrabble MIDI mapper. Instead of traversing a Scrabble game board as a maze, the virtual composer traverses the simulated universe of Figures 1 through 3, collecting lists of atoms, their element types and locations. The question of whether element 0 maps to the tonic, element 1 to the fifth, and so on in a MIDI channel's current scale, is left up to the scale currently in use on that channel. A composer defines element-to-interval mappings for each named scale (Aeolian, Mixolydian, Minor Blues, etc.) in a scale definition file, and a performer decides what scale and other parameters to use for each MIDI channel in play, setting these per-channel parameters via a Python command line or a graphical user interface.

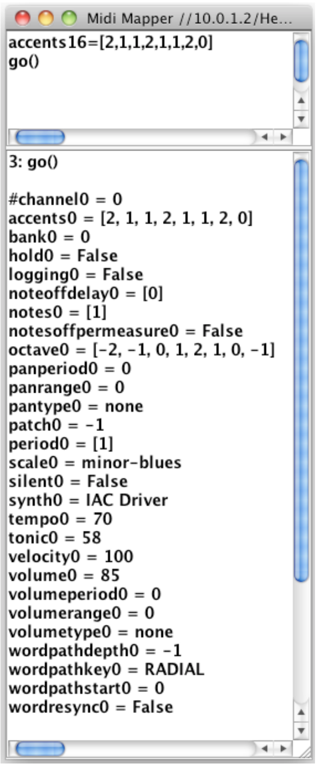

Figure 4. Mapping parameters, one set per MIDI channel

Figure 4 illustrates the per-channel parameters used to map game state to MIDI messages. Several mapping parameters relate directly to the game. **Tempo** relates to the speed of universe expansion, **tonic** sets a channel's tonic note, and **scale** is a named scale for element-to-note interval offset from tonic as previously discussed. A **patch** is the number for the MIDI channel's synthesizer voice, and **accents** is an integer vector that gives the relative strength of each successive note played, with 0 denoting a rest. The accent pattern defines the meter for a MIDI channel. Other parameters affect the number of simultaneous **notes** played, **sustain**, and other musical properties.

The MIDI generator applies these mapping parameters by traversing lists of atoms and their locations received from the game via a network connection. Atomic number maps directly to interval in a **scale**. Many parameters determine how atoms are translated, e.g., the number of individual atoms to translate per sixteenth note (the **notes** parameter) and their relative importance to MIDI velocity and meter (**accents**). Each MIDI channel gets its own set of mapping parameters. As the music mapper generates notes from game data, it feeds note data back to the game in a data feedback loop, causing visual atoms that are sounding to light up with a fill pattern. Filled atoms in Figures 1 through 3 represent atoms being sounded.

Using differing-length accent patterns with shared prefixes generates polyrhythms. Using related but differing scales on different channels generates polychords and harmonic ambiguity. Extraction of melody by a performer from a generated sequence of notes takes the form of mapping only a small, repeating sequence of atoms to MIDI notes. There promise to be new modes of interaction that we have yet to establish for shaping the music mapped from the game.

The availability of live coding via the Python interpreter opens up avenues for new modes of play. A performer can enter Python command lines and script invocations that affect MIDI translation parameters in an expanded top section of Figure 4. Game players have a similar Python coding user interface. Atom injection and probability manipulation of the game can be semi-automated via live coding. Also, all of the mapping parameters of the MIDI mapper are available for semi-automated update on each MIDI channel by performers via performance time live coding. Rather than simple scalar and vector *values* for each channel, live coding provides a means for supplying scalar and vector *functions* with periodic, aperiodic, or chaotic behavior with respect to the values they return to the MIDI mapper. Mastering this live coding aspect of the game / instrument takes practice and attention.

3.2 Spatialization in a Planetarium

We studied a number of approaches to spatialization in non-planetarium environments, including those that often fail [3]. Guidelines from the latter study have to do with *precedence effect* due to varying time-of-arrival of a sound at a listener, *asymmetry* of multi-loudspeaker localization and panning, *directional hearing* and the role of head movement, and interrelationship of spatial hearing with auditory fusion and stream segregation. Attempting complex or subtle spatial audio effects at a planetarium is even more problematic. Concentric circular seating eliminates the possibility of using reference listener body orientation and location. There are no standards for speaker arrangement or relative speaker levels in a planetarium [2], which by its circular nature does not conform to the Surround 5.1 standard.

For the March 20 premiere performance we mapped the spatial properties of the game to the spatial location of the surround sound system. Each of the five speakers around the base of the planetarium dome projected a different set of synthesizer voices representing *sections of a virtual orchestra*, e.g., the string section, the percussion section, the wind section, the pad section, and the reed section. As mobile atoms traversed a section of the dome in the neighborhood of a speaker, they triggered note generation in the MIDI channels feeding the synthesizers driving that speaker. The game's metaphor is thus extended to include the universe-as-virtual-orchestra on the planetarium dome. With pronounced synthesizer voices this approach is effective. We did not attempt to create spatial effects within the concentric audience, but rather to emulate the spatial effects of an orchestra on a planetarium dome, visualized and manipulated as an expanding virtual universe populated by moving sonic atoms.

4. CONCLUSIONS AND FUTURE WORK

Initial ensemble performances in March 2012 have satisfied the performers and earned positive comments from attendees. This software instrument is a testament to the concept of taking collections of events that are individually pseudo-random, but that exhibit collective statistical regularities, and mapping those regularities to statistical regularities in improvised computer music. HexAtom goes further than Scrabble-to-MIDI in exposing underlying probabilities to the performers, and in supporting live coding for performance-time creation of novel statistical functions. HexAtom is a more complicated instrument that we look forward to getting good at playing.

The primary long-term musical goal for this work is creation of a comprehensive stochastic model for composing and performing improvised computer music, where the musical score is a hierarchical organization of probabilistic musical events. There is a tradition of stochastic musical analysis and stochastic musical performance that the present work looks to extend [14,15]. Experience with music-generating games has shown that performance is every bit as work-intensive and concentration-intensive as performing on a more conventional instrument. Rather than focus a human performer's actions and attention on creating individual notes or sounds, however, this style of performance is somewhat more like conducting, in that the primary activity is guiding the activities of musicians, in this case virtual musicians playing on their respective MIDI channels. A large component of composition is planning transitions in scales, tonic, accent patterns, and other configuration parameters of the per-channel MIDI mappers to fit the envisioned piece. Performance includes playing the game and migrating the configuration parameters, more or less according to plan. It also includes manipulating downstream timbre using other hardware and software tools to complement the output of the music generator. In a sense a music-generating game supplies partially prepared musical material that can be refined within the game, for example by selecting a substring of a stochastically generated note sequence as a melody, as well as by tools, such as synthesizers and effects units, that use the game's generated note sequence.

This process-based approach contrasts with Reich's assertion in "Music as a Gradual Process" [11], which states, "The distinctive thing about musical processes is that they determine all the note-to-note details and the overall form simultaneously. One can't improvise in a musical process—the concepts are mutually exclusive." While Reich's assertion is correct for *deterministic* musical processes, it is incorrect for *stochastic* musical processes that place aspects of probabilistic musical event creation under a performer's control. A stochastic musical process can determine the overall form while determining details only to the granularity of a statistical distribution of note-to-note events. Statistical distributions are not random. They shape the note-to-note events in the aggregate, but they do not shape them deterministically at the level of individual events. Giving performers some degree of control over the probabilities, whether implicitly via game rules such as imposing order on pseudo-randomly selected tiles in Scrabble, or explicitly via manipulating probabilities of note fission, fusion, etc. in HexAtom, gives performers a means for improvising within the framework of a stochastic musical process. Thoroughly exploring this approach to composition and performance is the primary goal of the present work.

Exploring additional shapes, games and processes for tessellating a planetarium dome and generating music, and finding additional ways to utilize a planetarium's visual and auditory environment as a musical instrument, is another path for future work. We also plan to migrate game play from laptops to programmable mobile devices distributed among the audience. Each mobile device will act as a controller, feeding commands to the game-playing computer via a wireless network. Luckily, we can explore the stochastic composition path and the planetarium-as-instrument path simultaneously.

5. REFERENCES

[1] Codeorgan website, http://www.codeorgan.com/, January 2012.

[2] Gaston, et. al., "Methods for Sharing Audio Among Planetariums," 2008 Fulldome Summit, Chicago, Illinois, July 3, 2008, http://whiteoakinstitute.org/IPS2008.pdf.

[3] Kendall and Cabrera, "Why Things Don't Work: What you need to know about spatial audio," *Proceedings of International Computer Music Conference 2011*, Huddersfield, UK, August 2011, p. 37-40.

[4] Kutztown University's Grim Planetarium, http://www.kutztown.edu/planetarium/, April 2012.

[5] M. Liuni and D. Morelli, "Playing music: an installation based on Xenakis' musical games," *Proceedings of the Working Conference on Advanced Visual Interfaces*, Venezia, Italy, May, 2006.

[6] PA Computing and Information Science Educators Conference, March 29-30, 2012, http://www.pacise.org/.

[7] Parson, et. al, September 2011 Computer Music Seminar, Grim Planetarium, Kutztown University, https://www.facebook.com/media/set/?set=a.10150793380120117.730472.579770116&type=3&l=7118b6f81e.

[8] Parson, Schwesinger and Steele, "Using Jython to Prototype and Extend Java-based Systems," Proceedings of the 26th Annual Spring Conference of the Pennsylvania Computer and Information Science Educators, Shippensburg University, Shippensburg, PA, April 2011, http://faculty.kutztown.edu/parson/pubs/JythonPACISE2011.pdf .

[9] D. Parson, "Algorithmic Musical Improvisation from 2D Board Games," *Proceedings of International Computer Music Conference 2010*, New York City and Stony Brook, NY, June 2010.

[10] D. Parson, "Chess-based Composition and Improvisation for Non-Musicians," *Proceedings of New Interfaces for Musical Expression 2009*, Pittsburgh, PA, June 4-6, 2009.

[11] Steve Reich, *Writings on Music 1965-2000*, Chapter 1: Early Works. Oxford University Press, 2002.

[12] Joel L. Schiff, *Cellular Automata: A Discrete View of the World*. Hoboken, NJ: John Wiley & Sons, 2008.

[13] S. Smallwood and G. Wang, "Chuck Chuck Rocket," *PLOrk in the Round*, May 2, 2006, http://plork.cs.princeton.edu/listen/green/.

[14] D. Temperley, *Music and Probability*, Cambridge, MA, MIT Press, 2007.

[15] I. Xenakis, Formalized Music: Thought and Mathematics in Composition, Second Edition, Pendragon Press, 2001.