Chess-based Composition and Improvisation for Non-musicians

Dale E. Parson
Kutztown University of Pennsylvania
P. O. Box 730
Kutztown, PA, 19530, USA
parson@kutztown.edu

Abstract

"Music for 32 Chess Pieces" is a software system that supports composing, performing and improvising music by playing a chess game. A game server stores a representation of the state of a game, validates proposed moves by players, updates game state, and extracts a graph of piece-to-piece relationships. It also loads a plugin code module that acts as a composition. A plugin maps pieces and relationships on the board, such as support or attack relationships, to a timed sequence of notes and accents. The server transmits notes in a sequence to an audio renderer process via network datagrams. Two players can perform a composition by playing chess, and a player can improvise by adjusting a plugin's music mapping parameters via a graphical user interface. A composer can create a new composition by writing a new plugin that uses a distinct algorithm for mapping game rules and states to music. A composer can also write a new note-to-sound mapping program in the audio renderer language. This software is available at http://faculty.kutztown.edu/parson/music/ParsonMusic.html.

Keywords: algorithmic composition, chess, ChucK, improvisation, Max/MSP, SuperCollider.

1. Introduction

Performing and improvising music are activities that typically have high entry fees in terms of time and incremental skill development required before producing listenable results. Composition requires creative mastery of music theory within a cultural context. Performance requires automation of mechanical skills demanded by a musical instrument before creative playing and improvisation become possible. The initial path to exploring these skills is too steep for some people.

This paper reports initial results of an effort to introduce non-musicians to musical composition and performance by way of an interactive computer chess game. The resulting system allows two players to create music by playing chess using networked computers. Mapping software within the chess game translates relationships on the board to musical structures. The chess program then transmits these musical structures to a sound generation software process.

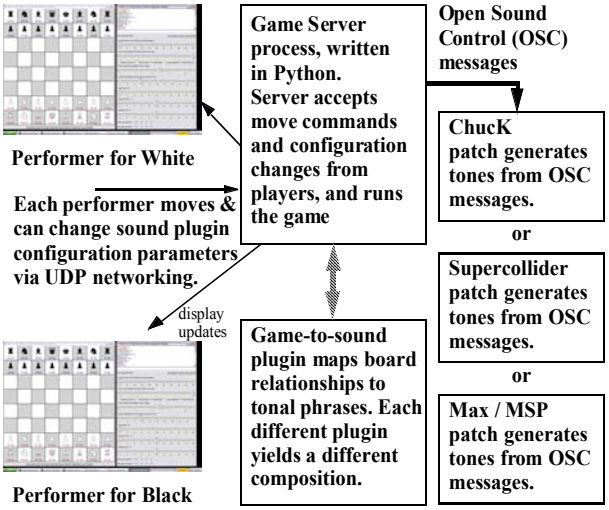

Figure 1. Chess-to-music software components.

This modular software architecture supports composition, improvisation and performance. One form of composition entails writing a *software composition plugin* that maps game structures to musical structures. Another form of composition entails writing an *audio renderer* in a sound generation language. This renderer translates the musical structures mapped by the composition plugin into sound.

Figure 2. Piece-to-piece relationships, shown by arrows.

Provision of a number of *plugin-specific configuration parameters* via a graphical user interface (GUI) makes it possible for players to examine and alter game-to-musical structure mapping at play time. An example plugin maps piece-value differential in an attack or support relationship to sustain duration, maps players' move speed to tempo, and maps depth of look-ahead on the board to phrase length. A different plugin can provide other configuration parameters to players. A related, current project in an undergraduate programming course maps sequences of words in a crossword puzzle game to chords in a MIDI sequence, where configuration parameters support play-time manipulation of tonic and scale for letter-to-pitch mapping, tempo, meter, accents, arpeggiation, and voices (*patches*). In addition to sequences of words, other game state variables such as player identity, score, word length and point value are available for student translation into musical properties. Players learn performance and improvisational constructs both by playing the game and by learning the game-to-music structure mapping via graphical parameter manipulation.

2. Related Work

Using a game's rules and state to compose and generate music at play time is a relatively new area for exploration. Game-based compositions in ChucK include an improvisational ensemble piece based on multiple player slot machines [1] and a piece based on a video game [2]. The chess system differs from these compositions by exposing music mapping configuration parameters to players for manipulation outside the scope of the game.

A notable commercial example is the Lumines™ interactive software game [3] that integrates a block placement puzzle program with visual effects and music that are synchronized to the game state. The game generates music from rules and state in the same spirit as the chess system. The chess system extends the game-as-performance concept of Lumines to allow composers to create new plugins and audio renderers and to allow performers to improvise by manipulating configuration parameters directly, outside the scope of the game.

3. Playing Dynamics and Experiences

The most productive approach to generating listenable music from comes to light when moving into game configurations that create interesting patterns of sound, and then bringing patterns to fruition by manipulating mapping parameters. A sound pattern is intrinsically interesting because of sequences of harmonies and rhythm generated by the game state and audio renderer. The graphical mapping controller makes such a pattern elastic, so that a player can explore and elaborate a good pattern before going on by making a move. Alternation between game playing and parameter manipulation is the main modus operandi for improvisation.

The author has introduced several populations of players with some chess playing ability to this music generation system, including high school students considering computer science as a field of study, musically inclined software engineers, and software inclined composers and musicians. A game distributed across two computers served as an interactive installation at college recruiting fairs and at an electronic music festival [4].

Players typically start out by concentrating on chess playing. Players with some musical background are quick to begin manipulating plugin configuration parameters. At some point a switch in focus by a waiting player to the configuration parameters occurs. Once that switch takes place, even musical novices can begin exploring concepts such as tempo, sustain, harmony, transpositions and phrase structure in a manner that is seamless with playing the game. Most players with experience in chess become actively interested in manipulating the musical structures. There has been no opportunity yet for long term observation of a population of players to determine effect on musical skills. However, non-musician players are clearly learning musical concepts related to both composition and performance. Musical skill improves with practice, as with a conventional musical instrument. A series of plugins with configuration parameters designed as a music tutorial laboratory could be written to lead players through composition and performance concepts beyond those embodied in current parameters.

This body of software is under active, iterative development. The goal is to create an interactive framework for constructing new virtual 2D board games that are oriented towards game state-to-musical structure mapping determined by compositional plugins with graphical configuration parameters. Creation of a new type of musical tutorial laboratory is a long-term goal.

References

[1] S. Smallwood, "On the Floor," *PLOrk: Live at Richardson Auditorium*, http://plork.cs.princeton.edu/listen/richardson/.

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

[3] Lumines™, http://lumines.jp/, January, 2009.

[4] Electro-Music 2008 Festival, August 14-16, 2008, Kingsport, TN, http://event.electro-music.com/.

[5] Open Sound Control (OSC), http://opensoundcontrol.org/, January, 2009.

[6] Cycling '74 Max/MSP, http://www.cycling74.com/, January, 2009.

[7] G. Wang, *The Chuck Audio Programming Language*, Ph.D. dissertation, Princeton University, 2008, http://www.cs.princeton.edu/~gewang/thesis.html.

[8] SuperCollider, http://www.audiosynth.com/, January, 2009.

[9] G. Loy, *Musimathics, Volume 1: The Mathematical Foundations of Music*, MIT Press, 2006.