Patterns: A Graphical Language for Live Coding Music Performance

Roger B. Dannenberg
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
rbd@cs.cmu.edu

Abstract

Patterns is a live-coding performance piece using an experimental visual language. The key idea is that objects generate streams of data and notes according to parameters that can be adjusted on-the-fly. Many objects take other objects or even lists of objects as inputs allowing complex patterns to be composed from simpler ones. The interconnections of objects are indicated by nested circles in an animated graphical display. The composition is created by manipulating graphical structures in real-time to create a program that in turn generates the music. The audience sees the program while listening to the music it generates.

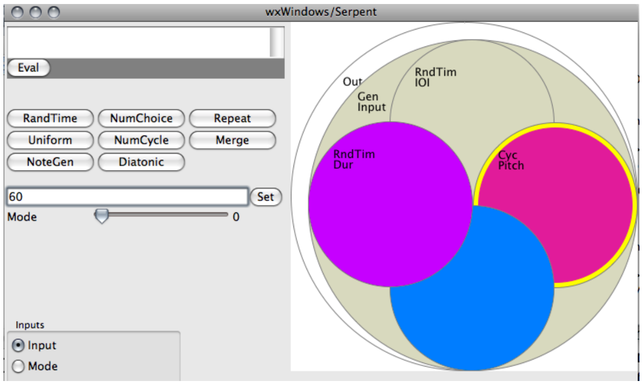
Introduction

Live coding refers to music performance in which sound generation software is created during the performance (Dannenberg 2011). Typically, the software development process is revealed by projecting the developer's computer screen. This gives a new dimension to the listener experience because the audience can see the generative structure of the music as well as hear the resulting sounds.

An obvious drawback to live coding is that software development is usually a slow process. Live coders use a variety of techniques to overcome the problem, including extensive libraries, very high-level languages, memorization, and "tricks" that produce interesting sounds with minimal specification. Patterns is an experimental system intended to provide a very high-level language, a graphical interface that might be appealing to a non-programming audience, and visual language syntax that minimizes programming errors while maximizing expressive power.

The Pattern Language

Figure 1 illustrates the interface. The circles represent objects that generate or process streams of notes and numbers. A typical generator is called Cycle. It generates numbers in a cyclical fashion from a list. A typical processor is named "+". It reads the next numbers from two input streams and outputs their sum. Most generators have several parameters. For example, the Cycle generator has a list of numbers and a "mode" that determines whether to traverse the list forwards, backwards, or back-and-forth. Parameters for the selected generator (shown at the right) appear on the left panel for manipulation. More interestingly, parameters can be determined by new generators. By combining manual manipulation of parameters with automatically generated parameters (and even parameters of parameter generators), deeply nested and interesting musical structures can be specified and manipulated quickly.



60

Mode 0

Figure 1. Graphical interface for the Patterns system.

Conclusions

Patterns is an interesting and viable approach to live coding. It offers the sort of low-level generality one associates with real "coding," yet the graphical interface and high-level stream operators give the performer a fighting chance to make some interesting sounds quickly. On the other hand, the interface is not transparent even to programmers seeing it for the first time, and the flashy visuals lead others to believe this cannot be "real programming." Still, performances have been well received, and there is room to extend both the language and its use in performance to address these concerns.

This work is supported in part by the National Science Foundation under Grant No. 0855958.

Reference

Dannenberg, R. 2011. Live Coding Using a Visual Pattern Composition Language. *Proc. of the 12th Biennial Symposium on Arts and Technology,* Connecticut College: 47-52.