Yuanzhe Li

Music and Computers Final Project Description

For my final project, I decided to pursue glitch music and the art of aesthetics. From some research, I found a algorithmic way of generating interesting sounds from short programs called bytebeat.

Bytebeat is an algorithmic sound generation technique discovered by a Ville-Matias Heikkilä (Viznut) in 2011. He originally used a C program to create an infinite loop, each time incrementing a variable t, and outputting the result of some equation on t as an 8 bit value. Each output value is treated as the amplitude value of a single sample.

For my performance, I created a chuck program with several preset algorithms I liked, either created on my own, or borrowed from others online. These can be chosen and edited in real time, while also keeping the same counter values (not starting over from 0 every time).

I wanted to use these sounds and combine or edit them into an extended performance, exploring the variety of sounds that can be generated. I decided to write a program using ChucK to facilitate this. The chuck program also sends the data over OSC to a short program in Processing (built on Java), to generate the glitchy visualization. The visualization takes 256 bits at a time, quantizes them to 2 bit, and displays them as a greyscale grid.

In the online community, people have analyzed why these algorithms work, and how we can manipulate the different operators to generate patterns, rhythms, and even melodies. The initial analysis by Viznut is here: <http://countercomplex.blogspot.com/2011/10/some-deep-analysis-of-one-line-music.html>.

The soundscapes generated reveal the algorithmic simplicity, and complexity that one line programs are capable of, and is a form that rewards inquisitive exploration over deterministic composition.

A later paper was published by Heikkila describing and analyzing the techniques and theory behind bytebeat:

arXiv:1112.1368v1 [cs.SD] 6 Dec 2011

<http://arxiv.org/abs/1112.1368>