Music 680, Fall 2007: Special Topics in Music - Compositional Algorithms
Class 1: September 10, 2007 - Course introduction

definitions: what is an algorithm? what is algorithmic composition?
- "2. Math. A process, or set of rules, usually one expressed in algebraic notation, now used esp. in computing, machine translation and linguistics." [*Oxford English Dictionary* 2nd ed.]
- "Models of process are natural to musical thinking. As we listen, part of us drinks in the sensual experience of sound, while another part is constantly setting up expectations, and in so doing, constructing hypotheses of musical process." [Curtis Roads, *The Computer Music Tutorial*]
- formalization as the precise (strict) representation of a musical process

perspectives: a relatively open view of algorithmic composition in the syllabus
- continua that might provide perspective:
  - process versus constraint
    - process as procedures which define musical parameters (at any formal scale)
      - isorhythm as a historical example
    - constraint as rules which limit compositional choices
      - species counterpoint as a historical example (though not a formalized one)
  - totalizing vs. limited
    - or, what percentage of the work (by amount, by number of parameters) is algorithmic?
  - stochastic vs. deterministic
    - algorithms which embrace randomness vs. those which provide a fixed result
  - listener-designed vs. composer-designed
    - algorithms designed for perceptibility vs. those intended as creative support
- classification by algorithm type
  - state machines, cellular automata, iterative processes, etc.
- classification by musical style
  - serial, aleatoric, minimal, spectral, etc.

historical instances of algorithic composition
- isorhythm (Machaut as exemplar)
- deployment of numerical ratio (Dufay as exemplar)
- canon and canonic technique (Ockeghem and Bach as exemplars)
  - including inversion, retrograde, mensuration
- the dice game (Mozart)
- variation form

canon in the early twentieth century
- Arnold Schoenberg *Drei Satiren* (1928)
- Ruth Crawford Seeger *String Quartet 1931*

some recent examples of algorithmic composition
- David Franzson *Il Dolce Fare Niente*
- Matthias Spahlinger *128 erfüllte Augenblicke*

the Common Lisp programming environment

- atoms
  - 2
  - pi
- built-in functions
  - (+ 2 3)
- nesting built-in functions
  - (+ 2 (* 3 5))
  - (* (+ 2 3) 5)
  - (+ 2 (* 3 5) 4)
- parameter assignments
  - (defparameter half-pi (/ pi 2))
- procedures
  - (defun plus (x y) (+ x y))
  - (plus 2 3)
  - (defun square (x) (* x x))
  - (square 2)
  - (square (square 2))
  - (square (plus 2 3))
- compound procedures
  - (defun sum-of-squares (x y) (+ (square x) (square y)))
- conditionals
  - (defun absolute-value (x) (cond ((< x 0) (- x)) (t x)))
- recursion
  - (defun fibonacci (x) (cond ((= x 0) 0) ((= x 1) 1) (t (+ (fibonacci (- x 1)) (fibonacci (- x 2))))))
- lists
  - (defparameter x (list 1 2 3 4))
  - (car x)
  - (cdr x)
  - (car (cdr x))
  - (cdr (cdr (cdr (cdr x))))
  - (defparameter y (cons 0 x))
- recursive list construction
  - (defun fibonacci-list (x)
    - (cond ((< x 0) nil) (t (cons (fibonacci x) (fibonacci-list (- x 1))))))
  - (reverse (fibonacci-list 12))
  - ; to create the list in the other order, use a helper function to count up to the original “x” input

using SLIME

- select a code block and type c-x c-e to evaluate
- type ‘a’ to abort the debugger and return to the interpreter

the Common Music algorithmic composition environment (atop Common Lisp)

installing Common Music (on Mac OS X)

- install Aquamacs
  - http://aquamacs.org
- install Common Music
  - http://commonmusic.sourceforge.net