Class 15: Homogeneous Lists: Making and Manipulating Groups of Drawings

Held: Wednesday, February 17, 2010

Summary: Today we explore Scheme’s list data structure and some ways to use lists to make interesting images.

Related Pages:

- EBoard.
- Lab: Making and Manipulating Lists of Drawings.
- Reading: Making and Manipulating Homogeneous Lists.
- Due: Exam 1: Scheme and Image Basics.

Notes:

- Reading for Friday: Anonymous Procedures.
- EC for Rosenfield symposium. Thursday’s convo sounds particularly interesting.
- Sorry that I can’t be in class with you today. I know Nora will do a great job.

Overview:

- Context: What and Why Lists?
- Building Lists.
- Mapping Lists.
- Other List Operations.

Context: What Are Lists?

- Issue: We’d like to make images with lots and lots of similar shapes.
- Problem: How to do so systematically.
- Solution: Scheme’s list data structure and some related procedures.

Building Lists

- (list val1 val2 ... valn)
- (make-list n value)
- (iota n)
Mapping Lists

- Basic operation: \((\text{map } \text{func} \ \text{lst})\). Apply a function to each element of a list.
- For images, we’ll find it easier to do something like
  \((\text{map } \text{drawing-transform} \ \text{list-of-drawings} \ \text{list-of-values})\)

Other List Operations

- \((\text{append } \text{lst1} \ \text{lst2})\) - join two lists
- \((\text{reverse } \text{lst})\) - just what it sounds like
- \((\text{list-take} \ \text{lst} \ n)\) - take the first \(n\) elements of a list
- \((\text{list-drop} \ \text{lst} \ n)\) - drop the first elements of a list