# Computing Drawing: Similarly Different Lines

## Phase 1: Drawing

**Part A**

Amass a collection of nine examples of drawn lines made by artists, designers and architects. Print them on 7" x 8" sheets of paper and label them with bibliographic source information. Then, choose three kinds lines from your collection. For each, articulate three different methods one could use to recreate those lines for a total of nine articulated methods. Focus on the geometric and aesthetic characteristics of the line. At this stage, don't be concerned with media, material or tools. Write in English and avoid ambiguity or a reliance on interpretation of your words.

**Part B**

Choose one kind of line from your selection of three. Alternatively, combine two or more kinds of lines together to form a new hybrid type. Write Python Code to recreate, and then extend, that line. Exact replication of the original line is not important. Allow your interests and the medium to influence the outcome. Begin by drawing in pixels on screen before you transition to controlling a machine that marks paper. Draw your line for multiple durations at multiple scales. Use a fresh piece of paper for each iteration of your line.

**Part C**

Make an 21"x24" architectural drawing with the 24" side oriented vertically. This drawing should contain multiple—perhaps thousands, perhaps three—iterations of your line.

This drawing may elicit representational qualities but will not be overtly keyed to any representational system. It will not be to scale and will not correspond to any subject. Challenge your lines to convey depth and define space. Consider a controlled ambiguity between the 2-D space within the paper and a represented 3-D space perceived in the drawing. Respect the boundary of the page. Move beyond the definition of a drawing as a collection of lines into that of an expressed relationship between lines. How can lines perform collectively? How do they influence each other? As you work, also consider what inherently "architectural" problems your drawing can enlighten. Corners and edges are favorite foundational elements of architecture in this studio, but perhaps other productive "problems" can be articulated and explored.

**Detail Schedule**

TH 2.12    Welcome discussion. Technical logistics. Begin.

M  2.16    Part A Due. Pin up followed by introduction to Python lecture

TH 2.19    Present and discuss progress on Part B. Second Python lecture

M  2.23    Print out and pin up first draft of Part B (lines drawn with pixels). Python to control pen plotter and other machines lecture

TH 2.26    Come to class with progress made on Part B. work/discussion

M  3.02    Part B due. Pin up

TH 3.05    Come to class with progress made on Part C. work/discussion

M 3.09     Come to class with further progress made on Part C including a draft version. work/discussion/critique

TH 3.12    Phase 1 review. Begin Phase 2

**Resources**

| | |
|---|---|
| Alan Gauld | **Learning to Program** (web document). 2007 |
| Marco Frascari | **Eleven Exercises in the Art of Architectural Drawing: Slow Food for the Architect's Imagination.** Routledge. 2011 |
| Deanna Petherbridge | **The Primacy of Drawing: Histories and Theories of Practice.** Yale University Press. 2011 |
| Jesse Reiser and Nanako Umemoto | **Atlas of Novel Tectonics.** Princeton Architectural Press. 2006 |