## Document Purpose

The purpose of this document is to explain why and how programming will be included in the Interaction Design 2: Systems class.

## Why Programming?

There are two goals in including programming in the Systems class:

1. Use the necessarily systematic nature of programming best practices as tangible introduction to systems design principles
2. Provide the students with useful software prototyping skills

## How Will Programming be Included in the Curriculum?

The framework for introducing students to programming will include three separate modules. The final class project will attempt to tie all of these things back together (along with other strands discussed elsewhere).

## The Plan

### Programming Module One: Programming Basics

In this module students will learn programming basic such as variables, statements and loops and begin to understand how these things can be made to work together in simple algorithms.

The emphasis will be less on creating working code from scratch and more on being able to read and modify pre-existing examples.

#### Environment: Processing

processing.org

Processing is java-based and so, perhaps, not the absolutely easiest path into programming, but there are a few reasons we will use this language:

1. designed for producing visual output
2. basis for Arduino environment (see below)

With processing we can introduce some programming basics, and then rapidly apply them to producing screen output, accepting keyboard and mouse input. There are bunches and bunches of tutorials developed for processing we can leverage (http://processing.org/learning/).

##### Lectures/StudiosAssignments (draft - timing, details, relative importance TBD)

1. Installing environment (java version or .js port?)
2. Basic hello world program, output to message window then screen
3. Getting output to respond to keyboard/mouse
4. I will be introducing an important reading in the first class (http://en.wikipedia.org/wiki/Arcadia\_(play)) that deals in part with fractals. I want us to find a way to get the students playing with the fractal examples available for processing, and learning how to change inputs to manipulate the output. I only expect we get to a vague understanding of the particular fractal algorithms, though I want us to design a simple assignment that involves fractals.

### Programming Module Two: Simple Machines

In this module students will combine the coding basics developed above with physical circuits to produce models of simple devices.

#### Environment: Arduino

http://arduino.cc/

Arduino is a physical computing prototyping environment. The development environment is based on Processing (hence plan to introduce programming with Processing). There are 3 main reasons for including Arduino in the class:

1. Provides a way to manifest physically some important systems design concepts like flows and feedback
2. Will be a stand-in for physical devices/hardware that is likely to become an increasingly important part of the interaction designer’s world
3. Playing with this stuff is super fun

**Lectures/StudiosAssignments (draft - timing, details, relative importance TBD)**

1. We'll introduce Arduino by having the students do the basic examples that come with the starter kit (http://www.sparkfun.com/products/10173).
2. Take home assignment to connect a physical input circuit built with Arduino to a Processing program thereby controlling the output drawn to the screen.
3. As part of final class team assignment, use Arduino/Processing to prototype some (simple) aspect of the overall system the team has designed.

### Programming Module Three: Modeling Screen Experiences

In this module students will use standard web technologies and open source frameworks to model user interfaces both in terms of layout and interactivity.

#### Environment: Front-End Prototyping

The goal here will be to get students comfortable creating an environment where they can produce web pages with a standards-based separation between structure, style and interactivity. This should include familiarity with

* a simple IDE such as Panic’s Coda (introduce with trial version, students will purchase if they wish or learn to work with a simple text editor or freeware editor)
* basic HTML
* basic CSS (or introduce a CSS framework?)
* jQuery

**Lectures/StudiosAssignments (draft - timing, details, relative importance TBD)**

1. How to turn a wirefram into a web page
   1. structure with HTML
   2. layout and style with CSS
2. Using jQuery to animate objects, respond to user events
3. As part of final team assignment, prototype some aspect of system team has designed