PARSONS THE NEW SCHOOL FOR DESIGN
Summer 2012 Bootcamp Syllabus

Teachers:

Ramiro Corbetta <corbetta@gmail.com>, Noa Dolberg <noadol@gmail.com>, **Jane Friedhoff <jane.friedhoff@gmail.com>,** Juan Pablo Patiño <patinoart@gmail.com>, Conor Russomanno <conor.russomanno@gmail.com>, Maya Weinstein <luckypapaya@gmail.com>, Vivian Xu <vivian.x85@gmail.com>

Learning Outcomes:

1. Give students a general understanding of what code is, the history of code, how it is currently used, the advantages and limitations of code.
2. Show students how useful code is for an artist (installation art, physical computing, web, etc). Contextualize code in an artistic/interactive setting. Provide resources, books, videos on inspirational artists.
3. Give students resources, both online and in print, to further their coding knowledge. Introduce students to the open source community as both a learning and sharing tool.
4. Introduce students to different open source coding platforms, their uses, strengths and weaknesses. (Many of these platforms will be further covered in weekend Dorkshop workshops.)
5. Understand basic code data types, syntax, and gain a basic feeling for how the 'machine' interprets this information.
6. Understand more complex features of code (functions, loops, arrays) and how these can be used to organize/simplify projects.
7. Strong emphasis on applying human logic (thinking) into code oriented logic (pseudo code). Also emphasis on breaking problems down into manageable parts.

Workload:

Daily readings/videos and assignments will be given the first two weeks. The final week students will work towards a final project. Daily work load will vary depending on assignment and skill level, but 1-3 hours can be expected.

Attendance and Grading:

Pass/Fail course.

Attendance: Only one absence is allowed. The second absence will be reported, and subsequent absences will result in a failure for this section of Bootcamp.

Grading: Grading will be on a pass-fail basis, with a higher concern placed in individual improvement rather than base knowledge.

Plagiarism: Plagiarism will result in an automatic failure. Working in groups and referring to online resources is completely acceptable, but this needs to be explicitly noted on the assignment that is handed in.

Day-By-Day:

Day 1: Mon., 30 Jul.
Opening event in the auditorium for all Bootcamp students

Topics covered:

- How artists use programming in their fine arts, IxD, game design, etc. practices.
- Examples of larger first-year code projects
- The history of computing
- A brief overview of programming languages
- Code and game design
- Resources for coders (references, other languages, inspiration)

Homework: Install Processing and poke around in it. Sign up for Processing forums. Read Code, Chapter 2.

Day 2: Tue., 31 Jul.
Introducing Processing

Topics covered:

- What Processing is, and what an IDE does
- How Processing thinks about 2D space
- Using println() and understanding what the console does
- How to comment code (and why you should)
- How to draw and color basic shapes

Specific code:

- Text to console: println()
- Shapes: rect(), line(), triangle(), ellipse(), point()
- Non-standard shapes: beginShape(), endShape(), vertex(), curveVertex()
- Modes for drawing shapes:
    - rectMode()
    - ellipseMode()
- Thickening and changing line styles: strokeWeight(), strokeCap()
- Using (or removing) color: fill(), noFill(), stroke(), noStroke()

Homework: Complete your section of the Mondrian piece. Read CODE, Chapter 8.

Day 3: Wed., 1 Aug.
Datatypes, Variables, and Math

Topics covered:

- What a datatype is, and why Processing needs to know it
- What a variable is, and why it's useful
- How to declare variables and set them equal to values
- Concatenation
- How to do math in Processing (and how datatypes affect math results)
- How to use variables to simplify and streamline code

Specific code:

- Datatypes: int, float, boolean, String, char
- Math symbols: +, -, /, *, % (briefly)
- Setting a variable equal to a value: =

Homework: Choose an image, photo, painting, etc. and make a version in Processing. Listen to The Information, Chapter 1.

Day 4: Thurs., 2 Aug.
Setup and Draw, Arranging Code, Animation, Scope, Special Variables, Conditionals

Topics covered:

- How to make a Processing sketch, and the difference between setup and draw
- Local and global variables, and how they're different
- How to use background() to simulate animations
- Mouse interactions
- Distance
- How to write conditional statements, and why you'd want to

Specific code:

- Basic sketch outline: setup() { }, draw() { }
- Animation: background()
- Mouse interaction: pmouseX, pmouseY, mouseX, mouseY
- Conditionals:
    - if ( ) { }
    - if ( ) { } else { }
    - if () { } else if { }
- Operators: &&, ||, ==, !=
- Distance: dist()

Homework: Use pseudocode written in class to guide a sketch. Turn in both the sketch itself as well as the pseudocode. Print out or handwrite code, then highlight or otherwise mark (in different ways) the stuff that's syntax-related, the stuff that's a parameter, the stuff that's a command, and the stuff that's a variable. Read Processing,
Development 2.

Day 5: Fri., 3 Aug.
Images, Fonts, Counters, and Nested Logic

Topics Covered:

- Datatypes for fonts and images
- Using (and making) the 'data' folder
- Process for loading in an image
- Process for creating and loading in a font
- Concepts behind how counters work
- Why nested logic can be useful, and how to use it

Specific Code:

- Images: PImage, loadImage(), image()
- Fonts: PFont, loadFont(), textFont(), text()
- Counters: millis()

Homework: Create a simple paint program or photo editor, in which you can click inside an area to change your brush's color, shape, pattern, etc. **Extra credit:** bring in your pseudocode for it. Read CODE, chapter 10. Choose a generative artist and read up on them for Monday.

Day 6: Monday., 3 Aug.
For-Loops, Functions, Randomness

Topics Covered:

- What a for-loop is, and why it's useful
- Structure of a for-loop
- How a for-loop runs, and what i does
- Why you shouldn't panic if you don't understand them immediately
- How to get random values

Specific Code:

- For-loops
- Random: random(max), random(min, max)

Homework: Use randomness within a function to create a sketch that has a generative aspect. You can base the style of your sketch on the artist you chose or your own imagination, but the randomness should be used in a purposeful manner. Utilize a for-loop in the process (e.g., to execute the function multiple times). Read Daniel Shiffman's book, Chapter 7.

Day 7: Tue., 7 Aug.
More Mouse Interactions, Key Interactions

Topics Covered:

- The built-in functions that keep track of what the mouse is doing
- The built in functions that keep track of what's going on with the keyboard
- What ASCII is, and why it's relevant
- How to reference non-letter keys (e.g. spacebar)

Specific Code:
- Mouse functions: mousePressed(), mouseReleased(), mouseDragged(), mouseMoved(), mouseClicked()
- Key functions: keyPressed(), keyReleased()
- Comparing keys:
    - Letters: key
    - Non-letters: keyCode

Homework: Use at least two mouse functions or keyboard functions (you can also mix and match) to create an interactive sketch (try pairing them with words, shapes, colors, etc.). Watch Jer Thorp's TedTalk: http://www.ted.com/talks/jer_thorp_make_data_more_human.html

Day 8: Wed., 8 Aug.
Exporting for Web, Arrays

Topics Covered:
- How to put your sketches on the web
- What arrays are, and how they're useful
- How to combine arrays with for-loops for clean, efficient code
- Zero-numbering systems

Specific Code:
- Arrays and array access

Homework: Create a sketch that mimics several lightbulbs/lightswitches, where clicking on one switch (or pressing a key associated with that switch) 'turns off' the corresponding lightbulb. (Alternate themes that similarly utilize arrays are fine as long as they are cleared first.) Watch Aaron Koblin's TedTalk: http://www.ted.com/talks/aaron_koblin.html

Day 8: Thurs., 9 Aug.
Array Practice, Physics

Topics Covered:
- How to think about speed in a sketch (an amount to increment position by every frame)
- How to mimic gravity in code
- How to do basic collisions

Specific Code:

Homework: Make an array of balls (or images, or whatever) that: (1) either bounce against the walls of the sketch and/or are affected by gravity, and (2) have their x- and y-positions start at wherever the user clicks. Watch Bret Victor: http://www.youtube.com/watch?v=PUv66718DII&feature=BFa&list=PLE3EA8B90F0EC848F

Day 10: Fri., Aug. 10
Mapping, Timers

Topics Covered:

- What mapping is and why it's useful
- How timers work, and how Processing can access your computer's clock

Specific Code:

- Time: second(), minute(), hour(), day(), month(), year()
- Mapping: map(), constrain()

Homework: Use mapping to make a clock (it doesn't have to be circular) or interactive calendar. Come up with a concept for your final project (a *drawing tool*, loosely defined,
or a *game*).

Day 11: Mon., 13 Aug.
Final Project Idea Presentations

Days 14-16: Tues., 14 Aug - Thurs., 16 Aug
Lab Periods, Working on Final Projects

Day 15: Fri., 17 Aug.
Final Critiques, Joyous Dancing, Celebrations, You're Awesome