

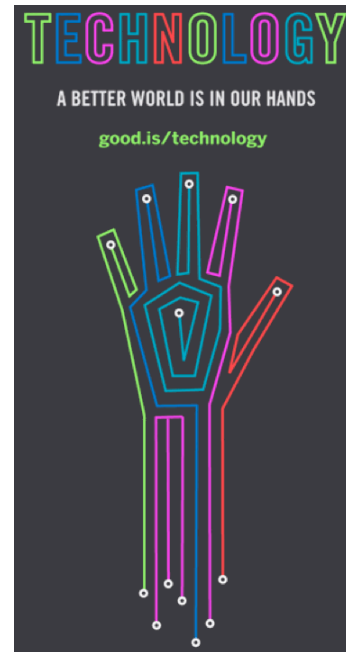


## Extra Credit 1: Tech Drawing

The purpose of this assignment is to give you more practice planning a design and working with Processing. That's all. (We exploit the fact that it is kind of fun to create things with Processing, whether they move or not.)

Notice the illustration of the arm in the technology ad at right. This design technique “fills the space” of an object with colored lines that have a “contact” at each end. The “contact” is really a white-filled circle with a smaller black-filled circle in the center, but it is supposed to look like the place where wires are soldered onto a circuit board.

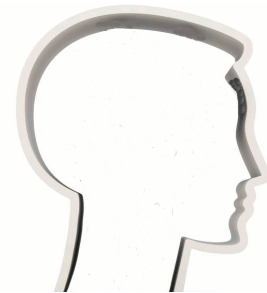
**Assignment Part 1.** Decide on a simple figure that you can use the “tech spacing filling technique” with. Silhouettes are great starting places, but avoid complex forms ... the assignment is not to create a tech version of Rembrandt! If you're stuck, use the adjacent head figure.



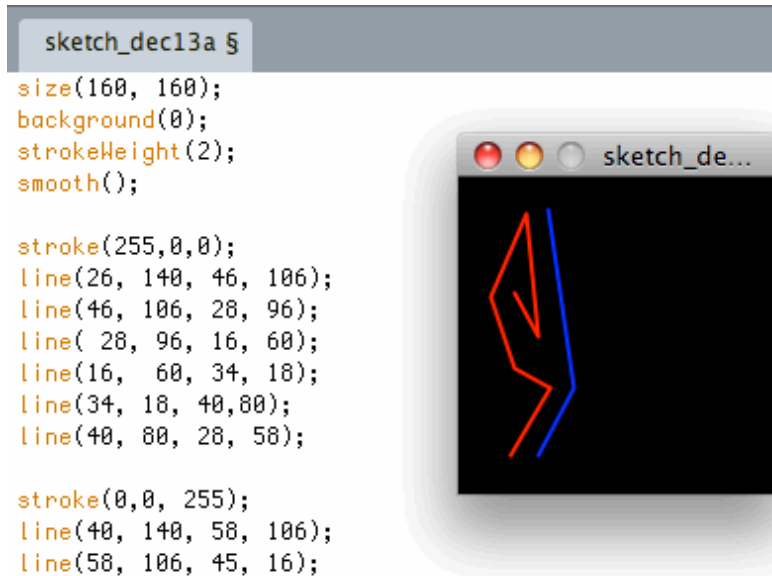
Once you have chosen a figure, plan where to put the lines. (This is most conveniently done using “old technology,” really!) Using graph paper --

<http://www.incompetech.com/graphpaper/plain/>

-- which you should print -- and a pencil, work out where you want the lines to go. Remember that each square can correspond to however many pixels you want it to on the final Processing canvas. (I found it easy to treat each graph paper square as 2x2 pixels; I just multiplied my “square count” by two always.) Also, remember to start counting pixels at the upper left corner, like on the screen.



**Assignment Part 2.** Using your plan, draw the lines of your figure on a black background using the commands of Processing. I found it useful to increase the `strokeWeight(2)` to 2, and specify `smooth()` just to get nice lines. Drawing lines can get sort of tedious, but keep in mind that your drawing isn't for art class – it's for CS Principles. So, approximating the point positions is OK as long as it gives the basic idea (and someone can figure out what the figure is.) Also, notice that because these lines are connected, the second pair of positions in a `line( )` function will be used as the first two in the next one. Here's my solution to the “head” figure, part-way through.



**Assignment Part 3.** To finish the design, put in the contacts. Notice that you know the positions of all of the circles, since they are just the end points on the line. My result – not great, but probably understandable – is shown at right.

### **Turn-In**

Print out – really! – your Processing program and turn it in *with your planning sketch*. Also, submit your .pde file (renamed, of course, to *<your name>6ex.pde* ) to the class drop box.

