

6.914 Homework 2 [updated version]

Assigned 1/15 - due 1/17 by 4pm to mish@mit.edu

0) Send a program of any length that demonstrates the most interesting concept (or combination of concepts) that we've learned so far.

Please write 3 out of the following 5 programs. Make sure to comment your code so it's readable. Copy and paste each of your solutions into your email, and let me know which questions you're answering. Also, if you haven't already, please take the class survey at <http://tinyurl.com/3436zy>.

- 1) Create a program that allows the user to type text to the screen. Enter/return– and backspace, if you can figure out how – should work as usual. Bonus: have the color change depending on the speed of typing, using the *millis()* function (which counts the number of milliseconds since an applet has been started.)
- 2) Using HSB colors, create a drawing program that allows the user to select any of at least 10 colors of different hues and saturations (you can draw the clickable colors to choose from wherever is most convenient) and at least three possible strokeWeights. (*dist* will probably be useful when selecting strokeWeights.) Once the color is chosen, the user should be able to draw in a white drawing area with the selected stroke color and strokeWeight. Also draw the word “clear” – which, when clicked, gives the user to get a blank box to draw in.
- 3) Using nested for loops (one inside another), create a grid of shapes that all gradually move back and forth across the screen, bouncing back toward the other side as soon as any object in the farthest left/right column hits the left/right side of the screen.. Create a rainbow gradient across them. (Shapes in the leftmost column should be most red; shapes in the rightmost column should be most blue/purple; and in between, the colors should be evenly distributed.)
- 4) Using arrays of temporary variables – like two arrays, each containing the last *n* positions of the mouseX/mouseY variables, so that every entry in both arrays is updated each time draw() runs – create a some sort of trail of objects (letters? Shapes? etc) that follow the mouse. (These objects could be filled with colors that are most saturated (HSB!) near the mouse and less saturated farther away.)
- 5) Using switch-case, create a shape that moves continuously around the screen and changes direction according to the user's arrow-key input. (Remember that you'll need keyCode.) If the shape goes off one side of the screen, have it wrap (continuously, so that the same amount is visible at all times – try using modulus) around to the opposite side. If “d” is pressed, double the speed; if “h” is pressed, halve the speed. Feel free to do interesting things with color as well.

If you are having a lot of trouble with one of these, please let me know. This class is not supposed be impossible – if the problem is broken, I'd like to fix it.