MAS 962: *DIGITAL TYPOGRAPHY*

ps#3.

In PS1 we dealt with the joy of text coding, and in PS2 we ventured a bit into placing these codes in an interactive setting. I stress that the letterforms are something we will cover in detail in the latter half of the term, but for now I want to stay in the coding aspect.The key to these exercises is to realize the 'space' of the string, not the particular 'space' of the x-y coordinate system you see. A good example is the work of Brad Rhodes or Chris Kline (this is only a few of those that exemplify this principle). We want to steer far away as possible from meaningless visual effects. When they are meaningful it is fine. In other words, you should ask the question "if the text were to be replaced with color blobs or anything else, would it be any different?" Work that exemplifies this negative point would be Matthew Grenby and Tom White where 'form' is overtaking content. The letters themselves have no meaning. One way to improve Tom's work would have been to make the letter heavier depending upon which letter it were, etc.; in the same vein in Matthew's work one way would have been to give a meaningful transition to each separate letter/symbol instead of a single after-effect like filter.

Not to discourage anyone of course. The work in this course is unlike anything that exists in the world and is something to have pride in.

P.S. Thank you for bearing with me on the technical difficulties we are facing. I hope that my mail about the 'standard platform' should help things.

---

1) Read Chapter 3 in Bringhurst. Do you prefer sans-serif to serif type? Please give specific reasons for your preference.

2) Write a JAVA program with a TextArea widget, a Choice widget, and a Button widget. There should be two items in the Choice widget ("to caps" and "to lowercase"). Depending upon the setting of the Choice widget, pressing the Button should convert all the text to caps or lowercase.

3) In order to focus our concentration on the "space" of the string, but also not lose the sense of the semantics of the string, we will design more of these "String Filters", much in the spirit of an image filter. Add to the JAVA program of (2) a TextField widget for entering floating point numbers between 0 and 1 (inclusive). Add to the possible Choices the following ("Noise", "Blur", and "Enhance"). When the Button is pressed, the value in the TextField should be taken to be the 'intensity' of Noise added, or Blur applied, etc (where 0 is none and 1 is full). Think of solutions that are not only 'form'-based, but 'content'-based also. In order to realize the latter of course you may have to build a dictionary. It does not have to be an all consuming dictionary but just one big enough to demonstrate your ideas.