MAS962

digital
typography

In ps8, the issue raised at the end of the review was, "Why are we creating content that could have been created by an existing digital tool? Why are we not taxing the envelope of expression?" and furthermore, "Couldn't this content have been created *better* with an existing tool?" The same could be said about ps9 of course.

The answer as I see it is quite simple. We are *indeed* taxing the envelope of expression. The main reason being that this content is all subtly different from what you might have created using an existing tool because *you* (hopefully) are in complete control. "Yes," we *are* creating content that could have been created by an existing tool. But you must understand that the work is not exactly the same. The tools and methodologies you create are entirely and completely your *own* process. You are in complete creative control.

What is the advantage? On the surface there is *none*. In many ways you are severely constrained -- no blends, anti-aliasing, etc. The single most important gain is not immediately relevant due to the shortness of a single school term and the limited material that can be covered in these small exercises.

The advantage is scalability. The ability to attack larger problems than your very being. To be able to provide *complete* solutions which you have total responsibility over their effectiveness. PS9 was a good example of this phenomenon. You essentially built media players that can play any sequence of text. Relevance of course is an issue, but nonetheless you can imagine reading your e-mail or whatever text you read from now until forever in the rsvp mode each of you designed.

We are merely working on parts of the puzzle. Methods to creatively process texts, typeface designs that have their own voices, thoughts on fields of texts, and thoughts on streams of text. These are important parts we build for the future. Someday there will be the right context to prove these ideas.

*On another note, Scott Snibbe of Interval mentioned that the advantage of this JAVA exercise system is that "half-baked" interface ideas can be shown. There is no need for final interface detailing (i.e. guaranteeing the proper user experience); all ideas can be shown in their experimental form. And furthermore, the majority of them are *not* merely mock-ups, but actual working engines. The idea is expressed with clarity not as just a good idea, but an actually working idea. This is something that a lot of us going through college never had a chance to prove. All of you have this chance now.*

This is the introductory exercise to the final problem for this course, it is due on dec 5 (friday). On dec 9, the following tuesday we will have an exhibit of the ideas developed in mas962. The problem is essentially the following:

Whenever a word is typed into your applet, this applet will broadcast the word to a central server on acg that is broadcast to all other connected clients. It is the job of your client to accept the word input, and to also listen for new words that have been entered by the server. Essentially you are building a chat client, where the granularity of communication is a "word" (note that letters are not sent one by one, but at the termination of a word (period/space/newline) only then is the word broadcast.

You have a wide variety of vocabulary elements for displaying text, ranging from plain text, to your own typefaces, to talking type, to simple straightforward display. The goal should be one of communication, rather than pure expression.

The server will come up later this week. For now, just think of some ideas. PS10 of MAS964 is a good reference for ideas.