hand coding round robin

part I ~ lecture

I would like to say thank you to all of you at this table. I am really honored to be in this teaching position today.

I was thinking yesterday during Dans class that even though I came into this space with the role of an organizer I pretty much immediately became student of you all.

I said yesterday that I was spreadsheet inside of a cloud, I think what I meant was that the organizer part of me has been feeling like a spreadsheet, something which follows a kind of spatial structure, something which is able to program but is also programmed by the structure itself. But then really I am not feeling wholly just like a spreadsheet, instead that part of me is inside of an entity like a cloud which is porous and but also full like a student and sometimes it rains. I really am so amazed at how many times over this session I was kind of transformed from one into the other and back again,

often I was both at once.

Anyway it is because of you that this constant shifting was even allowed to happen and I really have learned so much from you all so *thank you.*

-

So the schedule for the workshop will be a very baby lecture and then we will all be coding in HTML and CSS on each other computers. I’ll explain more about how that works later but each of us will end up with something that looks like this -> a web page with a bunch of text and colors and formatting.

**1. COMPUTATION AS METAPHOR**

In leading up to earlier versions of this workshop, I had been thinking a lot about the computation and metaphor. More specifically how the metaphor of the desktop was claimed along with the personal computer as “for everyone” but really it was for everyone who was already familiar with the system of the office and in step with yesterday's conversations the office is not only a kind of space but a space that heavily codes the people within it.

I think a lot of the things I was originally considering talking about were kind of imploded by our past three weeks together, so I am going to try and incorporate some of what we have learned from each other throughout.

I am perpetually perplexed that nearly everything I do as a programmer is somehow removed from what my computer is physically doing and instead I have been given sets of metaphors to remember and understand how to do those things within the visual metaphorical interface of the desktop.

Computers require that programming, as we know it today, be a set of extremely abstract processes. So, when I’m coding something, I’m often coding on top of someone else’s code, in someone else’s software, on top of the software that makes up my operating system which is permanently baked into the proprietary silicon chip of my Apple computer. Even the Terminal is a piece of software, a relic of teletypes and later screen-based computing that was text only before there were ever folder icons or double clicks.

**2. ONES AND ZEROES**

Computers at their core consist of a complex series of on and off switches, ones and zeroes. Everest walked us through this when they discussed ascii and Unicode and how a letter is actually a series of bits, and images are a series of complex letters and numbers, which are also just bits. This means that everything we do on our computers today needs to abstract away those complex patterns of numbers in order for us to use them in the way that we do.

These abstractions allow us to do things like writing text in a file, without thinking about *how* or even *if* the computer will be able to do this. It is important to note that since the early days of computing computer time has always been more important than human time. When programmers were writing code for computers the size of rooms the time it took to run the program was really expensive, so people were always writing code that was most efficiently read by the computer itself not by the people who were making the programs. Hence the situation we’re in now where code is written in often illegible jargon and minified into unreadable blobs of text.

**3. DESKTOP**

Going back to this idea of the desktop metaphor. The idea of a file inside a folder on a desktop...and then there's the trashcan. I’m always wondering, when you put something in the trash, does it ever really go away? And what is the file really? And what is the folder?

Melanie unpacked some of these questions for us in their folder poetry class. All files are a series of numbers but the organization and re-making of those files can be a poetic act. A computer scientist friend once told me that folders aren't actually anything. They don't take up \*any\* space on your computer.

Picking up on Kameelah’s concern with analogies and in step with American Artists Black Gooey Universe, what if we didn’t think of computer desktops like the desks we sit at to use them and instead we thought of computer desktops as what they were, screens emitting light, and what happens then when we take the screen on its own terms? How did it come to be? How have metaphors of files and folders limited our imagination for what computing can be? If not for the baked in interface of the desktop what might we put on a screen emitting light?

And in step with Simone Browne if we are to understand that light has historically been intertwined with violence what if we understood computers by for example the sounds they emit not by the visuals they display? What would it be like if we could hear all of the computation happening inside our laptops as we typed out an email? This makes me think about listening as a form of in-betweenness and like Mehrnaz mentioned in Ruha’s class listening as a form of activism

**4. ILLUSION OF POWER**

This is all just to interrogate the fact that there has been a significant shift from physical programming, the act of physically adjusting switches or weaving software into core rope memory to abstracted programming, building software on top of software. Wendy Chun has a really amazing essay on the implications of this shift which I’ve included in our shared are.na channel.

Since folders and files, which if we looked closely at them don’t resemble anything close to a piece of paper inside another folded piece of paper, I’m left wondering does the way that we use our computers also require some kind of illusion of control over our computers. The key word here being illusion.

**5. PROGRAMMERS**

Programmers, through the creation of complex frameworks and algorithmic systems go through so many hurdles trying to get the computer to do what they want it to do. When the computer finally obeys their commands, they are granted with this feeling of immense gratification and power. I’ve felt this feeling of gratification as a programmer after hours and hours of trying to debug something I wrote. It’s a cyclical command and response from you to your computer. My provocation here is not to call for an undoing of this cycle of command and response (although I’m interested in that) but to become aware of what is happening in between. So instead of humans controlling computers and computers controlling humans I am interested in how computers can be between, not over or under, the people that use them.

**6. IN A VACUUM**

If the process of programming a smart device, for instance, is rooted in notions of command and control, and if all computation is metaphor, does this make for dangerous results? Does it perhaps mean that programmers get so caught up in the power dynamic between themselves and their computer that they lose sight of how what they're doing lives in the real world alongside real people? I think the answer is often yes.

This is a video of my grandma writing an email to someone. It makes me wonder what it would be like if programmers had to press the keys of their keyboards so gently that it required them to move extremely slow and therefore with care? What would it mean if programmers had to manually move the switches to on and off positions in order to make a letter appear in a code they were writing?

**8. CAMERONS WORLD OF GEOCITIES**

And so, all of these questions have led me to care deeply about hand coding. Hand coding is the process of coding that bridges the gap between you as the programmer and your browser as the programmed. In other words, it is a way to code which is not dependent on someone else’s software or framework or library, instead it is the process of writing in the inherent language of the browser.

So I’m kind of lying to you here because your browser is software but hand coding in the software of the browser, like making folder poetry in the terminal, is closer to the metal of your computer then doing something like making a Wordpress website which is software on a big server depending on fragile code written by tons and tons of people and then minified into HTML CSS and Javascript. Not that this is bad or good but if we are interested in questioning relationships around intimacy, transparency, encoding and decoding, file systems, surveillance and proprietaryness then doing something like hand coding might be a step towards new questions about our relationships to these things.

So your browser is software which translates HTML, CSS and JavaScript into web pages. Your browser is the performer of the program you are writing.

**Text Slide**

The rest of the class will be a kind of meditation on the last three weeks. think about something you would like to share as code societies comes to a close. it can be in the form of a question or a phrase. it can be something you've been thinking about for a while or something that you're thinking about today. It can be something that you want to ask to the all of us. Or maybe it’s just a question to yourself.

The code you will write in this workshop will be very particular and you will write out each character slowly and by hand, there will be very little copying or pasting. We’ll move as slow as we want to and I will also be writing all the code out with you on the projector.

You will also do this on each other computers. The intention here is to perhaps erode this feeling of ultimate control that you have over your computer's system and to move carefully around someone else's. So, this class will be a little different in that we’re not going to be in discussion with each other necessarily in a verbal way. Instead we’re going to be in conversation through the things we are writing on each other’s computers.

I am in the camp of learning by doing so instead of me going on about HTML structure and CSS syntax and JavaScript we are just going to write code in these languages and then immediately see what they are doing in the browser.

So the round robin part works like this, we’re going to start at our own computers, I’m going to give you a prompt, we’ll all code along with what I’m doing on the screen, you’ll respond to the prompt in your own way, then after a like 3 minutes an alarm will go off and we’ll switch seats. If you're not finished don't worry we'll try to take on kameehlahs way of not trying or needing to finish.

Well do that like 9 times and then come back to our original seats. The intention is to have a bunch of original hand coded pages at the end that we can share with each other and then I’ll compile it all tomorrow as a part of the zine project. All of this can be as anonymous as you want it to be just know that whatever you write will be seen and responded to by someone else at this table.

First let's open the file in the browser. Let's have the browser take one half of the screen. Now let's open the sane file in a text editor like atom or sublime.

So the round robin part works like this, we’re going to start at our own computers, I’m going to give you a prompt, we’ll all code along with what I’m doing on the screen, you’ll respond to the prompt in your own way, then after a like 5 minutes an alarm will go off and we’ll switch seats. Well do that like 9 times and then come back to our original seats. The intention is to have a bunch of original hand coded pages at the end that we can share with each other and then I’ll compile it all tomorrow as a part of the zine project. All of this can be as anonymous as you want it to be just know that whatever you write will be seen and responded to by someone else at this table.