Teaching at the Edges of the Internet

Erin Sparling, sparling@cooper.edu, Cooper Union

June 26, 2019

Contents

1	Historical Context	3
2	Philosophy	4
	2.1 Tenets	4
	2.2 Original Class Structure	4
	2.3 Interactivity	5
	2.4 Iterations	6
	2.5 Current Iteration	7
	2.6 Future Iterations	7
3	Curriculum	8
	3.1 Student Work	9
A	Appendix	
Re	References	

Abstract

The internet (here used as a proxy for a rough collection of protocols, servers, interconnects and activities) is almost unknowably complex. Attempting to teach the internet is a lesson in futility, as learning any topic results in expedient irrelevance. Constructing curriculum, and convincing institutions that the curriculum is worth exploring, is therefore a waste of time at best, and actively against the best interestes of the students at worst.

This document summarizes the experience of constructing Cooper Union's current iteration of TE326 Interactive Design Concepts, including the broad experiences that students bring with them regarding what "the internet" is, and an approach for managing a malleable medium.

1 Historical Context

At The Cooper Union for the Advancement of Art and Science, the Art department has limited exposure to traditional technology curriculum. Additionally, the department tends to eschew techniques-based courses, in favor of studios guided by critical concept explorations. As a result, teaching courses for artists and designers that utilizes technology has always been a challenge.

In 2008, Flash was the dominant form of rich interaction, Adobe and Microsoft were exploring new declarative approaches to layout with MXML and XAML respectively, and ActiveX was being actively pushed. Internet Explorer 6 and 7 were the dominant browsers of the world [1], Google Chrome was just released (as of September of that year), and Windows Vista was on the horizon. The necessity of jQuery was only just beginning to be felt, as a result of Microsoft's insistance that it new better how to implement DOM querying and traversal, and while sites like Flickr had public APIs for at least 4 years [2], publicly available programmatic data access was anything but common. In short, Jeffrey Zeldman and advocates of the open and standards-based web had yet to reach a critical mass, and the future of the standard of HTML was in question.

Concurrently, the first release of iPhone the year prior brought with it the rise of a novel approach to handling different-sized screens in media queries. This CSS standard quickly became the defacto approach to adapting desktop websites to the smaller, touch-friendly fully-capable browsers starting with Safari on iOS, but still took another two years to fully bake into Responsive Design [3]. The existence of powerful mobile browsers launched a renaissance of care for the user: the purpose of a visit to a site from a phone is likely different than that of a desktop; the context of the visitor (in line at a grocery story, on a couch) is dramatically different than that of a user stuck at a computer; the idea of geolocation and context-aware optimization suddenly made sense.

It was in this morass that I was asked to design an introductory web course for the Art Department, and its subsequent two follow-up classes.

2 Philosophy

2.1 Tenets

Early on in the process of developing the class, I settled on a few tenets that would guide me through the subsequent decade of material:

- 1. Give students *tools to enable continuous learning*, and an interest to do so, rather than on a specific set of techniques.
- 2. Keep costs for materials and services to an absolute minumum, *trending* towards free.
- 3. Focus on the *tangible* as quickly as possible.
- 4. Remember that everyone comes at the web from a *different perspective* and level of experience.

Through these guidelines, I developed the first draft of curriculum for the introductory class. Cooper Union gratefully had a local server they were able to provision student accounts on, Cyberduck and TextWrangler were installed on each of the machines, and PHP was enabled.



Figure 1: The first class of TE305 in the Cooper Union Foundation Building

2.2 Original Class Structure

In order to demonstrate knowledge of html and css, it seemed sensible to base the midterm on CSSZenGarden, an example site dating back to 2003 whose sole purpose was to illustrate the flexibility of css on top of well-defined html. As a midterm, the restrictions CSSZenGarden imposed were perfect: use css only, learn a new DOM that someone else authored, treat the existing hundreds of examples as an open book test. For the final, after learning the basics of variables, control flow, loops and functions, the students revisited the midterm, except this time focused on minimizing the html via programmatic optimization, while preserving the css.

Early on, this focus of html, css and back-end programming was a bit contentious to the students. The class description stated "XHTML, CSS and Flash," and some students came expecting exactly those technologies. I learned early on to set expectations about what was actually covered in the semester, regardless of what the class description claimed.

2.3 Interactivity

After students got used to the idiosyncratic nature of interpreting the course catalog, there was only one more major problem with the curriculum: where is the interactivity? After the first semester, I answered this with the embracing of mobile as an acceptable proxy for interactivity, using media queries and the iOS simulator (which wasn't released until half-way through the first semester of the class). While investing in scalable design was a great first step in considering interactivity, refocusing the final assignment as a WordPress theme (leveraging Dreamhost's 1-click install) allowed for a more focused approach to thinking about back-end interactivity through the proscribed templating system, and allowed for limited front-end interactivity via extending existing themes.



Figure 2: Hani Lim teaches WordPress template hierarchy

2.4 Iterations

Despite students having success both in building WordPress templates, understanding the relevant technologies, getting "the bug" for pursuing technology, and relevant internships to support their learning, without real front-end interactivity, I felt like something was missing from the curriculum. The primary challenge was that I had one semester with the students, and needed to go over not just how to make a web page, but also ftp management, the difference between client and server, domain names, folder structure and the like.

As a stepping stone into maintaining the existing curriculum, but drawing focus to front-end interactivity, I re-wrote the final to utilize Tumblr themes. This allowed for the jettisonning of PHP from the curriculum, in favor of Javascript as the programming component, and the use of Mustache instead of WordPress templates.

One of the advantages that WordPress themes had is the fact that the developer is in control of both the installation of the blog, as well as their customization of the theme on top of it. With Tumblr, the underlying mechanism was updated too frequently to let the students' work have any life of its own. This, in conjunction with the increased popularity of jQuery, lead to the abandonment of Themes-based finals, and the adoption of front-end interactivity as the goal.

Subsequently, when I created the first recent instantiation of Interactive Design Concepts TE326, I focused on the students' work leaving the confines of their own creation, and participating in the broader API-driven data ecosystem of the internet. By utilizing technologies like mysql, twilio, twitter's v1 search api, and chrome extensions, students created data visualizations, personal experience trackers, gps-tracking utilities and miners of public data.

To combat the ever-changing APIs that students work was based on, or the access model and authentication schemes required by them, I created a series of proxy services that wrapped popular APIs, added turn-key caching and a simplified authentication method. By removing the requirements for auth and caching, it dramatically sped up the ability to learn one or many of the APIs, which then lent itself to bringing these into the introductory class.

2.5 Current Iteration

In its current iteration, the TE305.1 Introductory class covers HTML, CSS, JavaScript and a set of APIs. The material covered represents the total of the previous introductory class, as well as about $\frac{2}{3}$ of what was covered in the previous Advanced class. This natural evolution of the content, and compression of more and more advanced topics into the introductory class, have helped hook students early in the class, provide with better, more clearly defined learning objectives, and kept pace with the evolution of the medium itself.

As a result of this compression, more and more advanced topics can be covered in TE326.

 $WordPress \rightarrow Tumblr Theme \rightarrow jQuery$ $Google Groups \rightarrow Slack$ $FTP \rightarrow Github Pages \rightarrow Glitch$ $Twitter Search API \rightarrow Twitter Proxy \rightarrow Glitch Proxies$ $Two - dimensional static pages \rightarrow Three - dimensional interactivity$

2.6 Future Iterations

As the semesters go on, trends emerge regarding what students are interested in, but can't be easily accomplished. Common reasons for a topic being out of reach are the depth of background knoweldge or lack of readily available simplified tooling (e.g. machine learning), service limitations (iTunes Connect for publishing native iOS apps), or readily available hardware (stand-alone VR headsets, arduino boards).

Some of these limitations can be solved with the appropriate planning, for example our work over 2018 to establish an Apple Developer for Education account, which allowed us to provide access to students while waiving the \$99 standard fee that Apple charges. Others, like the availability of stand-alone VR headsets, are only partially solved with the existence of affordable hardware: the technical difficulty of going from two dimensions to three, while considering multiple device types in viewing the experience, takes careful planning and sample code creation.





Figure 3: Topics graph for IDC Spring, 2018

- 1. Boundary Conditions of an Arbitrary Medium Discussing ways of approaching an unknown medium, and appreciating its edges and restrictions the point, rather than a limitation.
- 2. The Request How many layers are there between when a website is requested and served, and how can each of these layers be intercepted and reinterpreted?
- 3. **The Cloud** What does the cloud look like? How is data stored, retrieved, and constructed?
- 4. The Algorithm Are we more susceptible to digital addiction, or is that the point?
- 5. The User Experience If the adage of "If you're not paying, you are the product" is true, what does it mean to design for people when their experience isn't the point?

Interactive Design Concepts is divided up into three distinct phases of work: five lectures designed to introduce technical knowledge in a concept (with an assignment due the following week), one 5-week midterm assignment and one 5-week final assignment. Each of these lectures and assignments built upon the key themes explored in the class, as illustrated in Figure 3. [4]

The five introductory lectures revolved around one particular topic, keeping in mind the broad concerns of the semester, and resulted in creative works in their own right. Rather than delve into each of these assignments, the following is a write-up on the final presentations from the students.

3.1 Student Work

The malleability of the medium of the internet is both its greatest strenght and biggest drawback. APIs come and go, making data services unreliable at best. Browser features are prototyped, launched, exploited and then deprecated. Server setups evolve from provisioned to virtual to being billed at increments of 100ms. Along the way, student work adapts, but the replay of it at a future moment can be challenging. Given these restrictions, what follows is a description of student work from TE326, Interactive Design Concepts Spring 2018 at Cooper Union. Links and screenshots are provided when possible.

John Gamble — A Roulette Thing

In the lecture on dark user experience, John latched onto the idea of video games that are openly dismissive of those that play them. Through the lens of a history of family gambling addiction, he brought these ideas to a simplified online casino experience, where the games were hostile to you in the most realistic of ways: your experience doesn't matter.

Each of the games allow you to win or lose virtual money. In contrast with the addictive "roulette-esque" mechanics of modern free-to-play experiences, the games neither celebrate your win, nor chastize your loss. Much like the impartial blackjack dealer, they don't acknowledge that you're doing well or poor. If you lose all of your money, that's it. There is no continue, restart or micro-transaction to re-up. And to play up the anticipation that comes with large bets, the bigger the wager the more drawn-out the play (e.g. the roulette wheel spins even longer than it normally would).

Alex Tomlinson — Hear to There

The simplicity of the UI hides significant complexity of Alex's project. Hear to There is an audio landscape exploration experience. By selecting a start and end-point, the application creates a playlist of geo-tagged field recordings along the path. This project is totally open to user submissions, highlights the challenges of exploring anonymous contributions, and location-specific double-edged sword of public vs private space.

Elvin Hu

Playing off of the minimalist aesthetic inspired by fitness applications, Elvin created a dating app that relied only on voice communication. He chose to invest in a voice-only experience for meeting people because of the lack of anonymity that real conversation reveals, and the lack of an equivalent abuse experience that photo-derived experiences have of uploading misleading or inaccurate photos.

A number of design elements in the application played off of the goal of furthering conversation. For example, photos of the user start as blurred, and are only brought into focus for a potential partner through further conversation.

Jessica Martinez — Sustain My Ability

In a true cross-discipline project, Jessica partnered with students from the electrical engineering and architecture departments to prototype a closedloop urban food system. A rarity for the class, she pursued using a hardware breakout board to feed data from their water control system into the website for real-time visualizations. The long-term development goal of the project was to connect as many sensors as possible from the sealed room to the site, to enable managing of everything from moisture content, current cricket count, temperature settings and more.

Gunnar Kortenbach — Gunnar

Following on the heels of the Kardashians, Gunnar pursued the lofty idea of leveraging the "your data has value" mantra by building a system to be the direct broker (and beneficiary) of that data. Equal parts "in-app purchase as art" and "a social network for one," Gunnar's app took advantage of users' expectations on celebrity access, micro-transactions and privacy, and turned exploiting those norms into the point. "For \$20 a month, you can track me," he says, assigning a dollar value to his gps tracking feed.

A secondary theme throughout Gunnar's project was the idea of scarcity. If enough people purchase individually personalized artifacts from him, the price needs to go up accordingly. Services that had a pre-defined SLA can be re-defined to accommodate surging, or previously affordable access to real-time data can become unthrottled... for a price.

Richard Yee - VR Browser

Rich's project combines a number of threads he pursued over the semester: collaborative multiplayer games, virtual reality visualizations and re-interpreting web data. VR Browser takes any website, analyses a screenshot of it and then re-contextualizes it as a 3D multiplayer environment, compatible with VR googles. Multi-user streaming audio completes the experience of browsing with friends, in 3D.

Anna Burholt & Tandis Shoushtary — Automated Rotoscoping

Anna and Tandis collaborated on one of the most ambitious projects of the semester. Walking the fine line as to whether or not the tool or craft should be visible in the construction of art, they explored a method of using Google Image Search to create auto-rotoscoped representations of work. The path that Anna and Tandis followed had them pursue first a fully-automated way of sourcing rotoscoped representations of an arbitrary frame of video, then to creating a tool that could batch manage like-image selection, and finally to a custom implementation for audio pieces, creating new video based on historical image archives from relevant works.

Diyi Liu —Smasshin

Dividug in to the idea of multiplayer gaming, referencing Tamagachi and Pokemon breeding mini-games to create an "Attack the Egg" game. With a shared experience across many simultaneous players, the egg required collaboration in order to be defeated. Future iterations are planned both for an AR experience, and simultaneously to call to attention that AR multiplayer experiences are inherently antisocial, with a room full of poeple together, but separated by their phones.

Min Joon So

Min Joon set out to explore the most ambitious uses of AR, and ended up hitting the walls of the cutting edge: all of the methods of pursuing AR on his version of Android didn't work as ARCore was not yet readily available. Ultimately though, his mocks were around 3D video playback in an AR space, simulating real-world performances.

Liushifeng Chen — Anonymous GPS Chat

In the spirit of Chat Roulette and Omeagle, Shifeng created a chat application for anonymous communication. His twist on this established genre was via a top-down map view, with each chat suspended in local space, which could only be explored via a radar-esque view.

For his final presentation, Shifeng gave his demonstration by seeding the room with location-tagged messages, and asking the class to explore the room to investigate the features of his project.

Appendix

Assignments for IDC TE326, Spring 2018 [4]

- 1. Create a malicious chrome extension
- 2. Build a http server to mock an object that should be connected to the internet, but currently is't
- 3. Write an RFC for objects that should not be connected to the internet, but were
- 4. Build an api that presents personal data in a way other students could consume
- 5. Document services and their apis that you use on a weekly basis
- 6. Propose a way to undermine yourself, based on data that you already expose
- 7. Create a dark user experience
- 8. Pecha-Kutcha on the above 13 topics, in a direction you'd like to explore
- 9. Build an MVP

References

- [1] N. McAlone, "These world maps show how the most popular web browsers have battled for dominance since 2008," 2015.
- [2] J.-S. B. N. P. P. L. Christophe Prieur (LIAFA), Dominique Cardon, "The Stength of Weak cooperation: A Case Study on Flickr," 2008.
- [3] E. Marcotte, "Responsive Web Design," 2010.
- [4] E. Sparling, "Thread: Assignments for "Destroy the Internet"," 2018.