A Somewhat Intentionally Vague Plan For The Mission Bit LinkedIn Fall 2014 Introduction To Web Programming Class Run By Glen Chiacchieri, With Notes About How Each Class Went

[This is a running journal of the class.]

Monday, September 22

Introduction, discussion, introduction to mac, setting up laptops.

Notes about how the class went

I had sort of high hopes and low expectations for this class. My dream was to get the students in an engaged discussion where they talk about their mental model for programming and we start getting to know each other through that conversation.

The reality, of course, was not that rosy. First, there was a lot of confusion over how to get into LinkedIn's building. I had to sign in twice and then it was unclear who I was supposed to be talking to in order to get more students into the room or how to get mentors in. The room we had was also not ideal, this huge break room thing. We'll be moving into a classroom next class. Eventually this was all worked out.

For the first 15-30 minutes there were only 3 students in the class. The mentors actually outnumbered them. I initially did a lot of chatting with the mentors, but then tried to talk more with the students. It was probably quite weird and uncomfortable to be in this office with a bunch of strange people.

I started by introducing the course, getting most of the logistic details out of the way, when the class met, who I was, who the mentors were and how they weren't always going to be there, that they were going to be getting some laptops, and that this was going to be a programming class starting with Scratch.

I mentioned that Scratch was made by MIT and that harvard's introduction to computer science class used scratch to introduce programming. This was my sneaky way of legitimizing scratch, trying to make it seem like it wasn't just for kids.

I told them that the class was going to start with Scratch but could move on to other things if they were bored. I tried to make it clear that it was based on what they wanted. I also told them it was going to be project-based. I probably didn't explain what that meant well enough.

After that pretty brief introduction, I explained that I worked on Scratch and showed a bunch of the stuff I've made, which some people seemed amused by. I asked the mentors if they had anything to show and a couple of them showed some stuff. I told the students explicitly that programming was used for a lot of different stuff and that I wanted to show them some of that breadth.

Then the most awkward part of the class, when I asked if anyone had programmed before. Only one had and I asked about her experience. She said she tried codeacademy but gave up because it was hard. I told her that I had cried when I tried to learn javascript. No one else had any programming experience, though a student that came later was taking a Java class. This was sort of a sneaky way of showing them that they weren't alone in not knowing how to program, that they were on a level playing field with everyone else.

I tried asking some more questions about which high schools they were from, what year they were, how they heard about this class. The students were understandably shy and I had to sort of pull the answers out of them. I pretty quickly gave up on this and focused on getting them set up with their mission bit macs.

Not all of the students had used Macs much, so I went pretty slowly in showing them how to create and log in to a new account on the mac, then how to connect to the linkedin wifi. Then I walked them through creating an account on Scratch and told them to just look around the Scratch website and play with projects. I used this time to go to the bathroom. I wanted to give them some time to get used to the computer and feel comfortable navigating around the scratch website.

The rest of the class was just them playing around on Scratch, finding interesting projects and playing games. The idea, again, was to get a sense of what kind of stuff you can make. After a while, I showed them that you could "see inside" any project and see the code for the project. I said I'd explain how the editor interface works later, but that's how they got to it. I explained that they could edit any project, but their changes would only be saved if they remixed the project. I'm sure they're still confused about that, but that's okay. I also told them about the "explore" page so that they had another way of looking around.

I tried to wrap up the class by asking them if they had found any interesting projects and wanted to share them on the big screen my laptop was connected to. That failed so I gave up on it.

Today felt very much like a prep day, but I was pretty happy with it overall. The students seemed to enjoy playing around in the Scratch site. They actually seemed genuinely interested.

My Goals:

- try to be clear about logistics of class
- give them some ideas for what programming can be used for (focusing on breadth)
- feel comfortable with each other, me, the mentors, the computers, and Scratch
- get them to discuss what they knew about programming and the social conventions

around it (startups,)
- start giving them the idea that they have agency in the class

Wednesday, September 24

Introduction to Scratch, Scratch tutorial, Find and copy a Scratch project

---

Before this class, I had a great time researching icebreaker games to get groups to know each other. While I don't think I'll use them in this particular session, I really like how ridiculous they are. In general, I think it's really important for people to get to know each other because a) making friends is awesome and b) it's actually pretty hard to help someone learn when you don't know them. While mentoring last semester, I noticed that it took me about a month of once a week mentoring before I felt like I knew the students at all, and then I was personally much more comfortable.

Notes about how the class went

This class was wonderful chaos.

I started off with a short lecture. I expected the lecture to start with putting Scratch in context of other programming languages, wanting to explain that it wasn't just for kids and that if you knew Scratch, the concepts would apply to almost all other programming languages. I did not explain that well, so I may start the next class off with that, especially since a couple students asked about other languages. Something I keep telling the students that I think they're just starting to get is that the class is extremely flexible in what we do, and they actually have some choice. I should tell them that more explicitly in the next class.

The next part of the lecture was a short, and probably confusing introduction to the scratch editor interface. I showed them how to create a new project, how to share a project, and then told them about the different parts of the interface. I don't think I did a very good job, perhaps because I was tired, but also because the interface is pretty hard to explain, especially how which scripts you see depends on which sprite you have selected (and what is a sprite?). Then there's backdrop, costumes, sounds, and the huge palette of commands. At least it's not JavaScript. I did explain somewhat clearly that a sprite can change its appearance completely and still do the same behavior, but I'm not sure how well they got that.

In retrospect, I probably would have walked them through some of the commands more and tried to make some more interesting things happen in view of the whole class, so they had a few more ideas of what things did without having to try everything out themselves.

After the lecture (which I expected to be longer), I asked them to complete the main Scratch project tutorial so that they could become familiar with the basic operation of Scratch. People did that pretty quickly.

While they were doing that, I made a little project of my own on the screen that people could watch if they wanted. I just made the default scratch cat go in a sine wave across the screen. This was probably too nerdy a project to show, but I did it since it was interesting to me. I would really like to find and create more artistic projects, but it's not really my wheelhouse.

After the tutorial, I didn't give any instructions, but people organically started making their own things, which made me really happy. Because everyone was following their own path, it was fairly chaotic, but having several mentors running around was really helpful. Some people were understandably less comfortable with the unstructured time, but actually everyone ended up finding some interesting things to try and it worked remarkably well! I'll try to keep it more structured in the future, but I'm really happy with how it went when it wasn't.

This unstructured time took up the whole class so we didn't end up doing the "copy a scratch project" thing, which I'm perfectly happy with. Maybe next time we'll do it?

Logistical things about this class: a lot of the students can't make it at the start of class (and some also get there a half hour early), so it's tough to lecture and keep them all together if that's what I wanted to do. I asked them at the end of the class if anyone wanted to change the time of the class, and some people said earlier, some later, but no one could say decidedly, so I kept it at 4pm. In the future, I'll just start things that everyone needs to be there for later, when everyone shows up.

My Goals:

- Get everyone to laugh together.
- Put Scratch in context of other programming languages.
- Give a quick tour through the Scratch editor interface to orient them.
- Have students start making Scratch projects and getting comfortable with the interface.

Monday, September 29

My Goals:

- Bring the class together a little more and have them start talking to each other.
- Show them that they have autonomy in what we work on.

Notes about how the class went

This class was pretty terrible. There were small logistical issues with getting students in the door, and then for the first 30-45 minutes only two students were there until 3 more showed up. I also didn't sleep much, so wasn't terribly articulate.

Since only Priscilla was there, I asked her what she wanted to do, and she said Scratch was confusing and wanted to learn how to make web pages because they were less confusing (and I think she was interested in it). I said sure and tried to start teaching her html with no plan. Other students leaked in while I was explaining, and I ended up lecturing the whole class about how HTML and CSS worked. As I should have known, it was a disaster. I had an incredibly hard time organizing my own thoughts, and of course HTML and CSS are nearly impossible to explain because you can't see anything, the box model is really confusing, you're writing words to produce pictures that would be easier to draw by hand, etc etc.

It's possible that introducing them to HTML/CSS in a confusing way was actually harmful. That by showing them that code was extremely confusing that when they encounter it next they'll just assume it's going to be very hard. And hey, that's probably true. But also, I want to give them a positive experience with code, to show them that they're not dumb and that they can totally figure it out, to have enough of an "aha" moment to want to continue.

There is a huge gap between Scratch and the real world. I get this definite feeling that because Scratch isn't "real", e.g. can't be used by anywhere but on the Scratch website, that no one is particularly interested in it. That's so sad because it's the only tool that is even remotely designed for human beings. Next class, I'm going to try to make the connection between Scratch and other programming languages very explicit, as well as try to show the difference between HTML/CSS and programming.

I was perhaps a little too eager to give autonomy to the students. At the end of the class I bumbled about trying to tell them that they could actually choose what to work on. Most of them don't really know what they want. Some of them have inklings (Priscilla asking about HTML, a boy asking about Ruby and Python). I wish I could personally sit with each of them and see what they're interested in and what would be good for them. But because this is a class, I can't really do that, which seems silly and actually kind of sad that I can't help them in a better way. After we talked a bit, I told the students that we'd continue on Scratch and I'll do better in the next class explaining what it means to program in Scratch

Wednesday, October 1

Scratch lecture, work on Scratch projects

My goals:

- Reign in the chaos from last class.
- Make it clearer how Scratch relates to other programming languages.

- Get students to work on Scratch projects again.

How the class went

Today was wonderful. Students and mentors trickled in at odd times so I put instructions on the board to a) continue on their Scratch project from last time b) do some of the starter projects on Scratch c) find an interesting project and try to duplicate it by building it yourself.

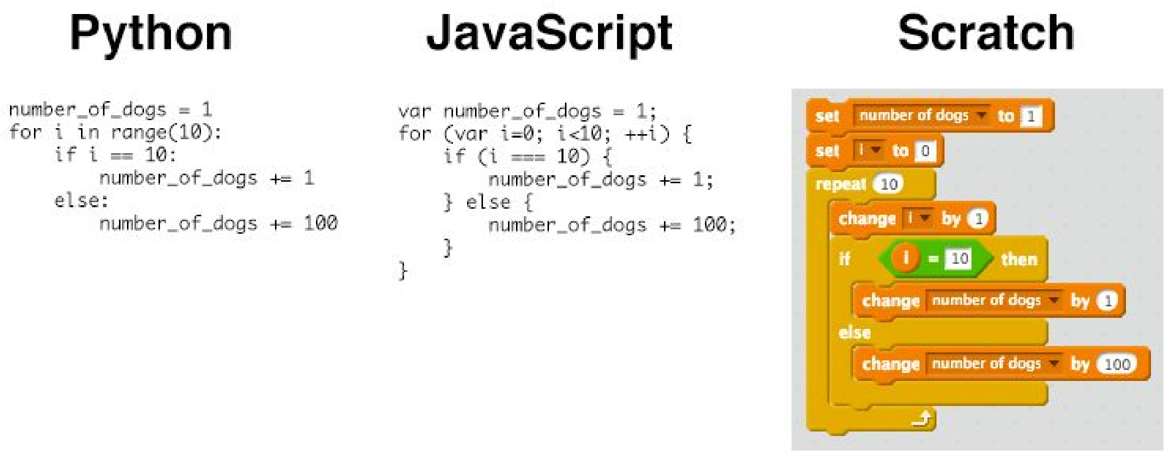
We had enough mentors so that each student had one mentor, and I think that was what made the difference. Students were a little directionless, but once a mentor sat down with them it sounded like everyone started to work on things. I sat down with Amalia and Priscilla and worked with them sort of simultaneously. Much of the work I did was in suggesting or challenging them to do things, and they spent a bit of time figuring out how those things could be done.

With Priscilla, she was sort of interested in how the thing I showed on the first day (the sprite traveling in a sine wave) worked, and so we worked through that. She had already known about sine in math class and could sort of make some leaps. One thing I tried to show her is that when something wasn't working as expected, what I liked to do was make things visible. We also discovered the color effect block in Scratch and had fun playing with that. Somehow we got distracted and started watching Vi Hart videos and I was totally okay with that.

I noticed that some other mentors were helping the middle school teacher taking my class as a student learn Java because she was taking another class where they were learning Java. I don't know what the other students were working on.

I ended the class with a 2 minute lecture on what Scratch represented. I told them programming was a way of thinking and that way of thinking was common across "real" languages and Scratch.

I showed them the following to images to get my point across:



Python

```
number_of_dogs = 1
for i in range(10):
    if i == 10:
        number_of_dogs += 1
    else:
        number_of_dogs += 100
```

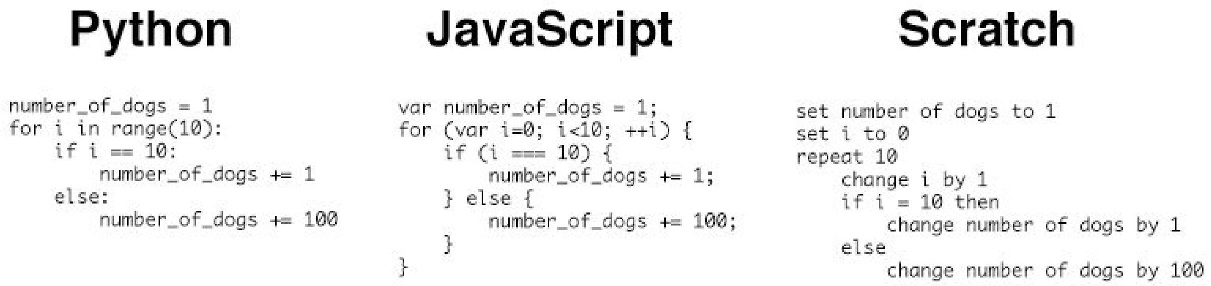
JavaScript

```
var number_of_dogs = 1;
for (var i=0; i<10; ++i) {
    if (i === 10) {
        number_of_dogs += 1;
    } else {
        number_of_dogs += 100;
    }
}
```

Scratch

```
set number of dogs to 1
set i to 0
repeat 10
    change i by 1
    if i = 10 then
        change number of dogs by 1
    else
        change number of dogs by 100
```

and without blocks but with the same words:



Python

```
number_of_dogs = 1
for i in range(10):
    if i == 10:
        number_of_dogs += 1
    else:
        number_of_dogs += 100
```

JavaScript

```
var number_of_dogs = 1;
for (var i=0; i<10; ++i) {
    if (i === 10) {
        number_of_dogs += 1;
    } else {
        number_of_dogs += 100;
    }
}
```

Scratch

```
set number of dogs to 1
set i to 0
repeat 10
    change i by 1
    if i = 10 then
        change number of dogs by 1
    else
        change number of dogs by 100
```

I have no idea if the students understood this at all (the concept of a programming language is still super fuzzy in their heads), but the mentors all told me they liked it.

I still feel weird about the tension between wanting to help everyone work on what they were interested in and considerations of the rest of the class. Today felt good because it seemed like everyone *was* working on what they wanted to work on.

Monday, October 6

continue working individually with Scratch, find and copy a scratch project if they can't think of what to do

My goals:

- Continue with Scratch. Hopefully by this point they've gotten over the initial strangeness of the interface and can start confronting the problems of converting their thoughts into programs.

Next week I think we'll try to pool class knowledge and create a project together.

How the class went:

Today's class went pretty well! Students continue to show up late which is alright. Every student was working on their own Scratch project.

I started by hanging out with Amalia and Priscilla. I sat next to Amalia, expecting to mostly let her alone while I worked on my own Scratch project (a simulation of a pot of water boiling to understand why putting a spoon over the pot prevents the water from boiling over). We looked at Minecraft sculptures a bit and I encouraged her to find a project she liked and copy it. I also playfully challenged Priscilla to (from memory) recreate the Scratch project we made last week, of a dragon flying in a sine wave. Amalia taught me how to draw the reflection from a bubble for my boiling water project. Priscilla nailed the challenge and I suggested a few other things she could do from there (like have the ghost she made float in a sine wave follow the mouse). Priscilla also had the great idea to play some music because the room was pretty quiet, so she became the unofficial DJ for the day. I'm happy she did this because it means she's comfortable and that the class probably doesn't feel very "schooly".

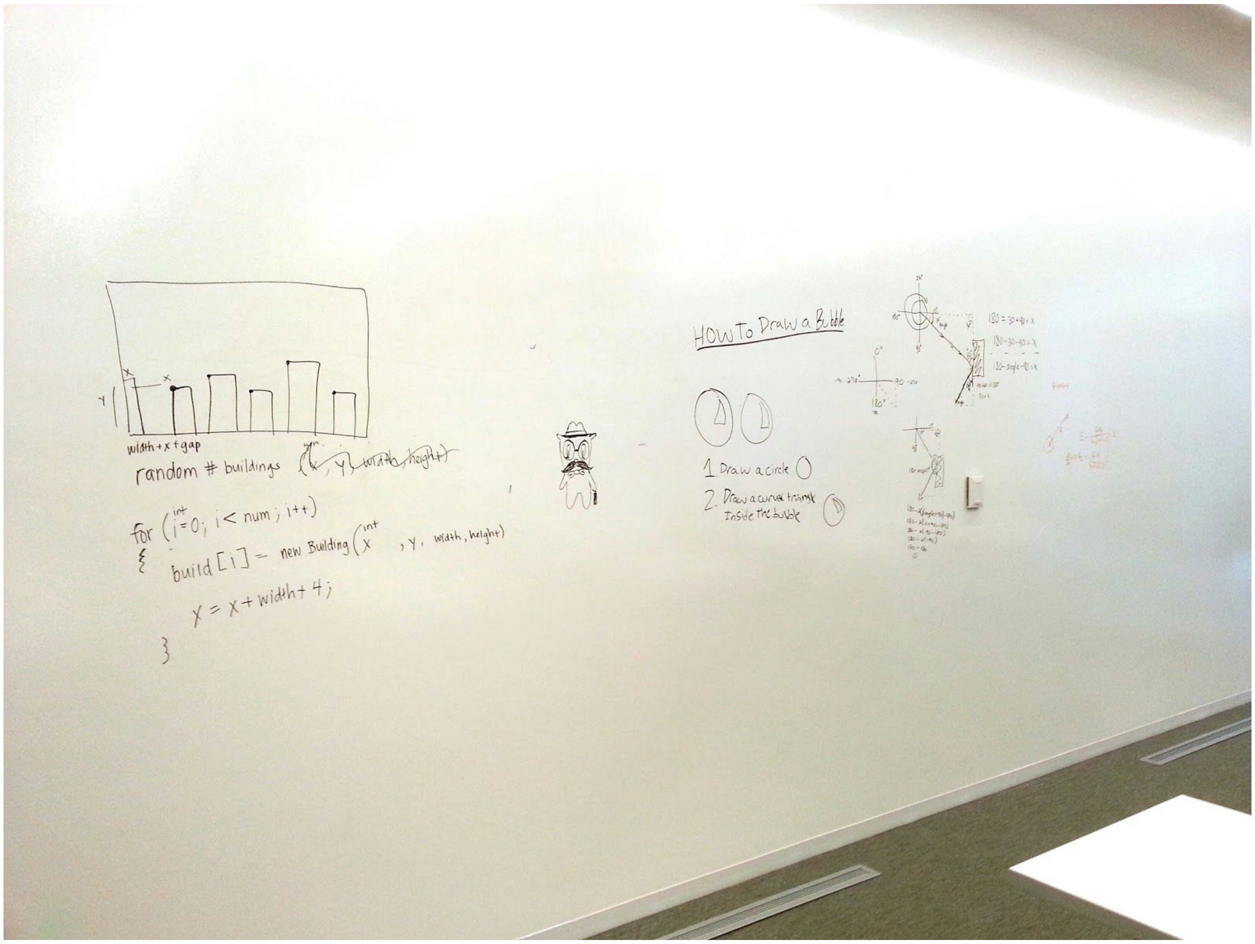
Anyway, some newer students showed up so I spent some time with them getting their laptops set up and getting them working on some things. *Unfortunately*, Scratch was under maintenance so we couldn't use that. I told the students to not turn off their laptops but just shut the lid so that when they come Wednesday they'll be able to save their projects. Hopefully that works. Anyway, with the new students I showed them a few of the dumb projects I've done to show them some of the breadth programming can be used for, then got them started on the code.org starter tutorial which I had never even looked at before. It was some angry birds thing and didn't look horrendous, but it didn't look great. One of the students has been taking a class where she’s learning JavaScript and the other had never been to this class and I don’t believe had ever programmed before. The former exclaimed when the code.org tutorial told her that the block language “repeat 5 times” was equivalent to a JavaScript ‘for’ loop: “Oh, it’s a loop!” I don’t know if she meant the “repeat” block or “for” loop.

I got asked by a mentor to help with a math problem in Christian’s pong game, so I left the two newer students and went to the whiteboard to figure out the law of reflection. I wrote a bunch of stuff down with Christian, trying to figure out how to make the ball bounce off the paddle in the right direction, but was totally confused. Being confused was both frustrating and liberating. Frustrating because I wanted to get the answer right and also it’s a really simple problem that I should know. Liberating because I didn’t *have* to know the answer and I could figure it out with Christian and we could get a look at how real learning works. The class ended right at the point where we figured out that we were really confused, which is unfortunate, but oh well.

One thing of note in the encounter above: I got to use and show off a small learning heuristic. After some calculations and some thinking, we came out with a formula that we thought might work. I suggested that we use the base case of a 0 degree reflection (e.g. if angle is 0, new angle should be 180) and see if our formula produces that correctly. It didn’t, so we both knew it wasn’t right. Also, while I was genuinely interested in the problem and mostly driving the approach, Christian was keeping right up with me from what I could tell. He was concerned about how we could even use any formula we came up with, and I told him that first we should figure out that our math is correct (which is usually the hard part), and then we could figure out how to turn that into code. After I finish these notes, I’m going to go solve that problem.

At the end of the class, I checked in with Priscilla and she had started trying to recreate a maze game she liked where a ghost chased you. From the glances I took at other projects, I think Amalia was working on a musical instrument project, and a mentor was helping Jessica (the middle school teacher) wrap her head around Java for another class she’s taking.

Here’s what the whiteboard wall looked like after the class, for reference:



```
for (int i = 0; i < num; i++)
{
    build[i] = new Building(int x , y, width, height)
    x = x + width + 4;
}
```

On the right is a little diagram a mentor did for a project he was working on out of his own interest. I think it’s a great sign that the mentors are making projects for themselves (while still helping out when neeed).

Wednesday, October 8

continue working individually with Scratch, find and copy a scratch project if they can’t think of what to do

My goals:

- let students continue working on their own projects in Scratch, with one-on-one help from the mentors

How the class went:

The students made it clear today that they didn’t want to do Scratch. Priscilla asked when she entered the class “are we still working on Scratch?” This was her polite way of saying that she wasn't interested in it, and a few other people piped up saying the same (two or three didn't really care what we did). I tried asking some questions about why they didn't want to use Scratch, and I got ambiguous answers, but my guess is that it's basically just not real thing that anyone talks about, and no amount of convincing on my part would work. They know what webpages are, have heard of Ruby, and want to know more about these things. Since more people want to know about webpages, we're going to learn that. This is totally fair and I'm happy to oblige them, but I'm honestly really dreading trying to teach them HTML and CSS (let alone JavaScript) because everything about them is confusing and hard to use. They were just not designed for humans to use, and I feel like I'm going to have to do a lot of lecturing (though I'm going to try to prevent that). That said, it's interesting that a "starter language" isn't acceptable to the students. It obviously makes sense that they want to do real things, and I wonder about a language like John Maloney's GP that might provide a smooth pathway from a Scratch-like system to a more powerful system.

As for the rest of the class time, I worked primarily with Amalia because I feel like I neglected her a little in the last class and wanted to make her feel more comfortable. I helped her make a project in Scratch with an octopus that followed your mouse. The main interesting thing about this encounter was that I broke down steps of what it means to follow something by asking her what exactly she was doing when she, for example, followed her cat around. I guided her quite a bit in how to break down this problem because she doesn't yet know exactly what things need to be broken down in order to make them into a program. But using the procedure she came up with, she was able to turn that into code pretty well.

I then challenged her to make bubbles come out of the octopus' mouth as it followed the mouse around. I mentioned the clone function in Scratch and she seemed to instantly get how that could be used. I tried a similar "break the problem down" approach of figuring out what it means for a bubble to come out of the octopus as it moves and she picked that up pretty well.

I found it interesting the individual steps she took to make things happen. For example, once I showed her that the "go to x:(mouse x) y: (mouse y)" was equivalent of "go to [mouse cursor]", she continued to use the former one in other situations where she could have used the latter. Perhaps I was a bit overeager in showing her that, but the fact that she kept using it made me think it was alright.

Somehow we started talking about really old computers and how they used to be the size of a room (she knew this and even knew where the word "debug" came from) and I told her a little about how weird of an idea it was to give one of those giant computers to a single person (the PC revolution). She seemed to really get this idea and could see how strange it was that this machine the size of a single room should be given to a single person. I also told her about Alan Kay's prediction about hardware and showed her the model and she was really impressed by how "smart" (her words) Alan was. She was also impressed by the idea of a textbook that had things you could play with in it (like Bret's explorable explanation). She actually independently said she wished textbooks were more like games, which I thought was pretty interesting.

As for the rest of the students, I noticed that one mentor was working with Christian on his pong game. I think Priscilla continued with her ghost mouse following project, but the boy she came in with told me (after I asked) that he didn't know what to work on. He was also one of the students. Jessica seemed a bit in her own world, possibly doing homework for her Java class.

Monday, October 13

introduction to HTML (not css), set up your own site on neocities

My goals:

- don't fuck up completely while trying to teach HTML
- get everyone to start getting to know each other so they can help each other learn when things start getting more complicated
- excite everyone by getting them set up with a real live site on neocities

How the class went:

It was Columbus day so only Jessica came :(She didn't seem to want to do anything, so we didn't do anything.

Wednesday, October 15

make a website!

My goals:

- get everyone playing with a website that they've made

How the class went:

Only Amalia and Jessica showed up. Jessica didn't seem to want to do anything (and actually mentioned she was getting paid to be there?), but Amalia did. I told her we were doing websites and asked her to download Textmate (which has a bunch of annoying installation things and is shareware?). A note: the built-in mac text editor does not handle editing raw HTML, so another text editor needs to be downloaded.

Amalia signed up for neocities.org and got the domain http://stupidduck.neocities.org/. For some reason she wanted to have a duck bouncing around the screen and since no one was there I figured that was a simple project we could get done in the class. I sat down with her and with me mostly driving I walked her through how to think about the program, including how to detect when the duck would hit the walls and what “bouncing” really means.

I actually didn’t know how I was going to do this exactly since I haven’t worked with animations much, but first I got her to type out the html of a base page, and showed her how to get an image on the page. Then I started working on the javascript to make the duck bounce. I got it to the point where it was moving across the screen, then got her more involved with the thinking I was going through. I started by showing her what the animation looked like by running once a second, then once every 500ms, then 250ms (at her request), then got down to 50 where it looked smooth. She seemed to get that.

Then we concentrated on the first bounce off the bottom of the screen. I told her that first we had to detect when the duck got to the edge of the screen and walked her through the idea of an “if” statement detecting when the duck’s y position was greater than the window’s height. We noticed that the duck only bounced when its top hit the bottom of the screen, so I walked her through thinking about the duck’s y position plus its height. Once we detected it (by doing a console.log), then we thought about what bouncing actually is. I basically had to give her the idea that bouncing off was just reversing the y speed. She sort of got it, because she picked up on the idea that we had to do it for the other bounces as well.

After we got the first wall working, the duck wouldn’t bounce off the top wall so we thought about that and she seemed to get it. Then we got to the right wall and she had a little trouble, but we figured it out, then she came up with the answer to the left wall quickly and by herself. The “John Holt” effect of her getting slightly faster at each of these was interesting.

And finally we had a duck bouncing around the screen. I emailed her the code and she put it on her neocities site. Woo!

Some personal notes: I really enjoyed talking with Amalia and she seems to really enjoy coming to the class. Along with the other mentor that came today, we’re constantly joking throughout all the work described above and occasionally we’ll wander off topic and talk about something else. We talked about neopets quite a bit today. Amalia is also quick to bring up that she hates math. She seems to be really interested in drawing and animation. In most classes she’ll draw something on the whiteboard wall of the linkedin classroom. Today, she also showed an amazing video made by someone she met at a disney animation camp called “robot elephant” or something? She also told me she had experiment with lots of stop motion and even tore up some pieces of paper to make a stop motion animation. I mentioned that there was a good book by Disney animators that talked about common animation techniques and I told her a little bit about the only one I knew of, squash and stretch. I told her I’d try to remember to bring in the book from my work’s library next week.

Monday, October 20

My goals:

- ?
- html?
- ?

How the class went:

I went into this class without really knowing what we were going to do or who was going to turn up, but it turned out that all the mentors and 6 students showed up so that was good! I asked about last week and a couple people said they had some exams and PSATs which is why they didn't come.

A new student, Wilson, also came to the class with Priscilla and we got him up and running with a computer. He said he came because the other class at Mission High School was really boring. Luckily today we all started together on HTML. I pulled up my slides from my previous web class and gave like a 15 minute lecture on HTML and what it was about, but as I noticed an interesting mood shift in the class. We started class by everyone joking around together and googling pictures of dogs (don't ask) and Amalia showing off the website we made last week. Everyone was in good spirits and it was fun. But after I lectured for a little while and we got to work, everyone was really quiet and it was much less fun. I tried to walk around and stir people up a little, but mostly they just seemed to be quietly working. Maybe that's okay, but I took it as a sign that it was boring.

I also quickly realized while I was lecturing that everything I was saying about HTML was going straight over everyone's heads, so I rushed my introduction a little in order to get everyone to start doing it for themselves. I got everybody to sign up for a neocities account and they used its online editor to mess around with HTML. On the projector I started making my own HTML page while the students did their own thing. I realized lots of them needed some hands-on help, so I tried to get in there and help them without being too "teachery". I noticed that the mentors were a little hesitant to sit by their sides. I think at this fragile stage of beginning, having someone right next to you if you need it is crucial.

At one point in the class I told everyone not to be afraid of googling, that that's how I learned and will be really useful as they continue learning.

One boy was getting really hung up on the syntax and was clearly extremely confused about how html worked. For one, he saw something about the <abbr> tag and just couldn't make sense of it. Why was there text that was written in the html that wasn't showing up on the page? I showed him it was just for the hovering thing, but he still didn't really get it. Also, everyone got totally hung up on the fact that <title> wasn't really the title on the web page. Jesus. And of course the default neocities web page comes with some basic CSS, and the relationship between CSS and HTML is just totally baffling to the few students that tried to mess around with it.

At one point, someone found the <marquee> tag, which apparently still works in modern browsers, and everyone wanted to do that, so I wrote it on the board. That was really cool.

Come to think of it, pretty much everyone was confused by HTML's syntax, with its brackets, equals, quotes, and things nested inside each other. Wilson tried to paste an html image tag into a css background-image rule, and that's actually a really smart idea! But of course it doesn't work because that would make sense.

HTML is a disgrace. To think that it came AFTER all the WYSIWYG work of hypercard and word processors, without offering ANY dynamic capability (at first) is just mindblowing and tragic.

Unfortunately, I forgot the Disney book I promised I was going to bring Amalia. I could tell she was a little disappointed about that, but I'll bring it soon. I actually felt like I was neglecting Amalia a little in this class. I picked up on some cues that she wanted a little attention (looking around shyly) and hoped the other mentors would help her while I helped 4 other students simultaneously, but none did. I checked in with her near the end of class, but I probably should have checked in earlier or explicitly asked a mentor to go help her. It also didn't help that she was the only one sitting at one of the rows of desks.

My coworker Toby suggested we arrange the desks in a circle so everyone could see everyone else and thought that kind of environment was helpful for collaboration. It's something I might experiment with at some point, but we seem to be doing okay right now.

Wednesday, October 22

My goals:

- continue with html
- help the mentor Jake ready to lead Monday's class

<u>How the class went</u>:

Pretty good! A handful of students came to this class, and I tried to flit between several of them. To be honest, I don't remember much of this class because I left halfway through to go to London. In the brief time I was there, it seemed like people were just getting into the complexities of html and quickly seeing the need for CSS. I sort of like the idea of starting with html, and then understanding why CSS is needed. At some point, I expect the same thing will happen with JavaScript.

Monday, October 27

My goals:

- none!

How the class went:

I was in London for this class, so I let Jake lead it.

Jake told me that Wayne ran this class and said it went well. I get the impression that Wayne and the mentors mostly just did everything for the students, including make this weird form submission thing with Kristian for his hilarious site KristianMingles.com.

Wednesday, October 29

My goals:

- continue with html and css
- have everyone start to be somewhat self-sufficient in their projects

How the class went:

Today, Kristian, Amalia, and Jessica showed up. I mostly sat with Kristian and helped him make some progress on his fake christian dating website. He wants to copy ChristianMingles.com almost exactly, only change a bunch of stuff so the site is entirely about dating him. I think this is hilarious and fully support his effort. Plus, copying another site is probably one of the better ways to learn html/css.

At first it was slow going. He still didn't really understand the differences between the html and css syntaxes. I showed him how to put a background image on the site using the actual background image from christianmingles. I also showed him how to use margin: auto; to create a container box in the middle of the page.

At this point, I tried helping Amalia figure out a project, but she couldn't really think of anything she wanted to do. The blog about her cat is okay, but she's clearly not interested in it. Without someone to guide her, I don't think. I felt a little bad about going back to working with Kristian, but I think he just needs a little help getting started and will go far once he gets going. Not to say Amalia doesn't have the same, but it's easier to help someone with a specific direction and get them to a more self-sufficient state.

Anyway, once I went back to working with Kristian, he figured out a little how to change the form he already had on his version of the site to look right. I showed him how to use the inspector on the other site to see which properties they had chosen in their css, and he picked it all up surprisingly quickly. For example, he started getting the css syntax really quickly, and seemed to understand the hierarchical nature of html pretty well. Something I love about copying another website is that it doesn't take much for your site to start looking like theirs and I think that's pretty exciting. I didn't really know if it was exciting for him to see his site coming together so quickly, but I was excited for him. I told him it wouldn't actually take that much longer for the site to be complete, and he guessed it would take a week or two to copy everything, which I think is right. Awesome. After a few more little things, we started using namecheap to look up domain names, and Amalia checked out a few, too. Unfortunately, Kristian's first choice was taken, but hopefully he can find one.

I told him that since he doesn't have a credit card (he's 15) I would pay for a year of his domain name if he could choose one. It's only like $10 and I see a lot of potential in him. I think he's at the exact right moment to learn to code and is really interested in doing it. He's quick at picking things up and is willing to work really hard on things. I think he can get a lot out of programming and I would like to give him a push in that direction. I would also be happy to continue working with him after the class is over if he wants to keep going. I actually think it would be fun to have a little hacking circle in my lab, but that would require quite a personal bond with my students.

(As I was writing this, I found out that Neocities was open source, so I created a github pull request to change the default text editor theme from the ugly dark one they have now (which is hard to read and imposing for beginners) to something lighter. I hope I can make more pull requests to neocities to make some improvements that would help my students (save button feedback, view page link, and remembering the default theme, for example).)

Oh, and one final thing I don't know if I've mentioned: I've been sitting at the same desks as the students, right next to them. I couldn't have done this at the beginning of the class (because it would have been weird and intimidating probably), but at some point without thinking I just asked if a student minded if I sat next to them. It feels much more "natural" and less teachery to do this, especially since I'm not sitting next to them in order to look over their shoulders and correct mistakes. I'm just sitting there because I like being with them and will often work on my own thing for a while before they ask me for help.

Monday, November 3

My goals:

- see if I can get Amalia interested in a project
- continue helping Kristian with his hilarious site

How the class went:

Amalia, Priscilla, Kristian, Jessica, and Bill showed up to this class.

I had previously promised Amalia that I would bring in the Disney book about animation from my office's library—The Illusion of Life—and I finally did! I showed it to Amalia and she looked through it for a while but didn't seem terribly interested.

We hung out a little and talked about fun things for a while and gently transitioned into working when the conversation ran out.

Amalia doesn't really have anything to work on, so over the weekend I thought it to ask her about making a sort of portfolio or little showcase of her drawings, but when I asked her about it, she demurred. "My sister", she said, "she should have a portfolio site." I tried to convince her and showed her the horrible comics I used to make, partly as a way of showing her that she has a skill and should be proud of it, but that didn't seem to help.

Bill showed up today and I sat with him for a while because he asked me about the <meta> tag. I walked him through some examples I came up with on the fly for explaining what a <div> was, too. He seemed to be mostly concerned with browsing other sites' html and was hesitant to try things out on his own website. He is definitely really confused about html and especially about how things are styled. Most of my examples were showing a little bit about the totally fucked box model that html has, but I was largely unsuccessful, I think. He also doesn't totally get the syntax and types some incorrect things that I sometimes try to very gently correct. I don't generally feel great correcting anyone after I've read John Holt, but at the beginning it seems to be okay to be a little more overt, as long as I back off once they start getting it.

For example, I worked for just a little bit with Kristian on his awesome Kristian dating site. I'm really impressed with how quickly he's understanding things. He still makes mistakes at first, but if I don't step in he corrects them soon after. Amazing. I sat with him for just a little to get him moving. He demonstrated that he understood what the four numbers in margin: 1px 2px 3px 4px; were immediately after I told him. He did something lots of students seem shy of doing, which is to try out different combinations of things to see if they work in a fairly controlled way, seeing the effects of each change one at a time instead of making a bunch of changes and then seeing what happens. I think he's going to do just fine with minimal guidance from me and I would guess he'll continue doing HTML if he finds another interesting project he could do.

Priscilla seems to be in a good place. She's learning well on her own and copying some sites that she likes the look of. Her HTML and CSS was really well-organized and she seems to be getting the syntax pretty well. She complained about how she wrote all these words (HTML/CSS) and all that came out was this kind of crappy website, which is a good point, in my opinion. She's demonstrated some skill with photoshop, and of course making this layout in photoshop would be trivially easy, but not so with html/css. I sat with her for a little and showed her how to do some basic layout things, like margin: 0 auto; and using floats for layout.

After I was done helping Priscilla, I wandered over to Bill and said, hey, what's up? Safari froze on him and I think I "caught" him looking at a thing he was doing for school. I was really confused if that's what happened so I asked him explicitly and playfully, "Oh, did I 'catch' you doing something else?" He asked, "should I lie?" and this amused me a lot, but I told him that I didn't care what he was doing, that if he wanted to do his Java homework he could do that and we would try to help him. I left it at that, but now I wish I had made it clearer that he could do anything he wanted to in the class and that was fine with me, that I'm not at all an authoritarian and there is nothing riding on this class. But of course showing and not telling is always better. [Update: after reflecting on this incident for a day, I wish my response could have been more emotionally warm and supporting. Instead of a carefree "I don't care if you do something else!" I could have told him that we could help him do anything he wanted, or asked him if there was something else he wanted to work on that we could help with.]

It strikes me now that maybe I should start asking more about the students' personal lives. I don't really know much about how their classes are going or what they're into.

Wednesday, November 5

No one showed up today except Jessica an hour late. I'm about 95% cool with that. I'm happy that they would not show up if they don't want to be there. I'm a little sad that my time was "wasted", but it's really not a big deal.

Monday, November 10

Today I showed up when Amalia showed up and we started the class by hanging out in the lounge area around our classroom. The entire floor of this building isn't being used yet so no one else was there. After a bit we made our way into the classroom and I encouraged Amalia to write back to the person who wrote back to her on the classroom whiteboard. We lazed around a little bit and talked and it was fun!

A little later, Aidan (a mentor) and Christian came in. We joked around a little. I showed them this stupid photoshopped image I made of a whale shooting rocket exhaust out of its blowhole:

They seemed to like that.

I gave a little bit of help to Christian at the start to try to give him some forward momentum, then hung out with Amalia more. I mentioned to her, "You know, I was thinking. We could make a tool to help us make stop motion videos using the webcam." She was interested in this and we brainstormed a little bit. I got the feeling that leaving her alone a little bit has made her a little impatient to start making things. We did a prototype stop motion test in photobooth to see if using the webcam made sense, and it did! It was actually really fun to have your own separate things that you were responsible for animating but we could still work collaboratively.

Anyway, Amalia made a paper crane while I hacked together the basics of what we would need for this tool. It only took about 15 minutes, but Amalia mentioned to me "You know you're the one doing all the coding, right?" I asked sincerely/sarcastically "Oh really?" I think she could tell that I was genuinely interested in making this tool, and I suspect she wouldn't get much out of coding it herself, so I think it's okay that I'll be doing the coding.

She said two interesting things during this collaboration. One was, when I proposed a feature for our editor, that I should "start small". She seemed to think that starting with a big idea probably wasn't the best, which is a very "code-minded" thing to say, and I sort of wonder where she picked it up. Maybe it's from her other creative projects.

The other thing she said was "This could be a site where you find and share images you made."

To me this was interesting because I'm not sure she or anyone her age has seen a tool that's custom-made for only one person and one use-case that you make yourself. All of the tools and websites she's seen have probably been designed for large numbers of people by some other people. They're not tools you make for yourself.

What I think could happen out of this collaboration is that with a pretty small amount of work I (someone that knows what they're doing), can make a tool that will help us do something creative. I don't know if she'll ever want to make tools herself, but at least she'll know it's possible. I especially want to stress the artistic side of this, that coding is a tool to make art for her and it's within her grasp.

Wednesday, November 12

Amalia, Kristian, and Jessica showed up today. My friend and colleague Robert also visited class today. I spent some time with Amalia figuring out what kind of thing we should build for stop motion animation. She sketched some stuff on the whiteboard and after we agreed on some things, I started coding while she browsed the internet. We talked throughout the class, and eventually I got something working well enough to make a test animation. Amalia and I made one and it was fun!

Amalia insinuated both that she came up with the ideas and I did the work. That seems sort of okay in that she still feels like it's her project to some extent. She also insinuated that she could not see herself doing the things I was doing. That makes me sad. I think that means I should include her more in future building projects. I wonder if this is a case where if I set my expectations high she would meet them. It's so hard to tell.

Dan worked with Kristian today and helped him understand html tables a little bit, but from what I overheard my have overtaught a bit, been a little too didactic. I noticed the same thing with Aidan on Monday, that there was a lot of telling Kristian what to do instead of letting him do and giving pointers and responding to questions when asked.

Monday, November 17

Looks like it's just Amalia, Kristian, and Jessica from now on, as they're the only ones showing up.

Today, again, a lot of hanging out.

When no one else was there, I asked Amalia what things she had Tuesday, and she sort of embarrassingly said she went to therapy. I was a little unprepared for that answer, asked mostly in innocence as it was, but after a bit of an awkward silence, I asked her in a very down-to-earth manner (like we were talking about anything else) if she found it helpful, that I've heard mixed things from my friends that went to therapy. She told me she's been seeing her therapist for a year and now is just able to start talking about some of the "deep dark stuff".

I also showed her a bunch of xkcd comics today in a whim because I thought she would enjoy them. I skipped past a bunch of the early obscure ones. I could tell that us looking at comics was distracting Kristian from his site, and I don't know—that seems like a good thing sometimes.

I showed Amalia and Kristian an early prototype of my bubble sort explanation and they were politely interested, bordering on genuinely interested? I had assumed that they would find anything I was working on tremendously boring, but not so. They were surprisingly willing to hear me talk about my work.

Wednesday, November 19

Today, I worked mostly with Kristian on Kristian Mingle. It's slowly coming together. We were all tired today and said so, but Dan, the mentor today, encouraged us to get to work (including me) which I'm glad he did.

One of the mentors that hasn't been to the class in a long time (he is definitely a brogrammer) showed up and after I got him up to speed with what we were doing, he sat with Kristian and proceeded to confuse him horribly. I was sitting next to Amalia and talking with Dan about some of my more radical educational beliefs when I heard this guy asking "is this the editor you use?". I was instantly really suspicious and after I heard a little more I went over there. The mentor asked, "is this [neocities] the website you've told them to use? can't you run a little python server on their machines or…" I cut him off right there and told him that doing that was a incredibly confusing and that lots of magic commands were confusing. He asked if they could even just use a file on their local computers and I told him that the distinction between having a website on your computer as opposed to the web was also really confusing, and that the model of working on neocities was really simple and people got it. He clearly didn't understand any of this. Luckily, he had to leave for a meeting at 4:30. I sort of muttered to myself "what was that guy's deal?" after he left and Kristian said, "He was very… serious."

After this, Dan took it upon himself to work with Amalia and generate a project for her. I noticed in the last several classes that she was actually really eager to work on *something*, but I was sort of unwilling to *make* her do anything, only suggesting projects. She wasn't excited about anything I was suggesting, so I left her a bit. I see now that she really wanted someone to tell her what to do, and Dan (an eager teacher), was only too happy to oblige.

I’m not sure how I feel about this. I’ve been probably *too* non-coercive because I’m very new at understanding when to be demanding and so have been very undemanding. But I think it might be inauthentic of me to help Amalia on a project that I’d be bored with because she’s not interested in it. I also really want to nudge her beyond passivity of being told to do something then loyally doing it. She seems eager to please, and I don’t feel comfortable that that impulse is bound up with pleasing authority. But I don’t know. Maybe it’s too early in her development for me to be coaxing her this way.

I’m afraid she’s taken my lack of technical attention to mean that I’ve put her into a role of artist or non-coder, or possibly even “girl that doesn’t code”, which terrifies me. I showed her an xkcd today about every tic-tac-toe move. She was quick to say, after not understanding it “I get it, it’s just for elites”, meaning nerdy folk and that it was beyond her understanding. I’m afraid I unknowingly solidified this distinction in her head when on Monday I skipped past a bunch of xkcd comics I knew she wouldn’t get or I didn’t think were funny. She heard this loud and clear: these comics are not for you, they are for “us”.

The same thing happened when I coded the stop motion animation tool. She saw my ability to code that tool in under an hour as an inherent property of me and not something she could see in herself. “Maybe you can do that, but I don’t know anything about html and css,” she said.

I think Monday I’ll have some real talk with her. I may end up apologizing. Making her feel like she can’t be part of the programming community is the worst outcome I could hope for and is something I can’t abide.

Finally, Kristian and Amalia and I are getting along really well. It really feels like a genuine relationship and that pleases me a lot (although I wish they were better friends with each other. I saw them hanging out in LinkedIn’s lobby before I got there, each looking at their phone). Kristian seems to be getting much more comfortable and was showing me things he discovered about html and image editing in preview, and we had a good time laughing at stuff, especially the headshot he and his sister (who is “good at photoshop”) made, of just his disembodied head floating on a white background. I got up on the desk at the end of class and wrote “yao ming was here” really high on the linkedin whiteboard wall. Both my climbing on the desk and what I wrote amused them, and they were worried (much more than I was) about ruining linkedin’s desks.

Somehow, an unspoken assumption has come up: that me and the students will all leave the classroom together. They struck around with me till 6:10 even though we weren’t really doing anything for a while before that. We left together.

[long gap where I didn’t feel like analyzing every class because it just felt like

hanging out]

Wednesday, December 10

Today was the last day of classes! Amalia and Jessica came and we just hung out and didn't do much, typical of the last day of any class, I think.

To fill in the gap of the last couple weeks, the main thing that has happened is that it's been really fun hanging out with Amalia and Kristian. I bought Kristian a year of the kristian-mingle.com domain and it's hosted on neocities.org which is a really nice feature. I'm a big fan of neocities and I think I might contribute code to it to make it a better resource for people learning.

Dan taught Amalia some javascript explicitly last week and this week I taught her about dictionaries ("objects") in javascript because I thought she might respond to it, but nope, not really! She told me last class that she really wants to learn how to draw anime and I meant to bring in Scott McCloud's Understanding Comics but I forgot.

Kristian emailed me after class and asked me to let him know when the hacker lab thing is starting. Yay!

To wrap up, I really like the club-like atmosphere the class turned into. It felt really good—much more relaxing than a typical "class" for both the students and me. It felt more like I was setting up a culture of programming, me and the mentors coming to hang out and program our own things and helping out students when they asked or we were bored. I hope to continue this club-like atmosphere at the hacker lab next semester.