Shane Cota
05/20/17

Final Report, Everett Middle School

Summary of Curriculum

The majority of the curriculum was based on App Inventor. I've been informed that the class has already worked through most of the curriculum prior to my involvement. In class programming was done by following the tutorials on the App Inventor website. Much of their work was essentially copy and paste. It was explained to me that at the 7th grade level, the course curriculum was selected with a mind to expose the kids to programming instead of a imbue a deep understanding of programming. The programs taught are "Oh My Spikes", "Pizza Party", "Map It", and "Photo Booth". These covered sprites, clock timers, activity starters, list pickers, data storage, event listeners, and hardware components like locations sensors and cameras.

Occasionally Dan had the kids turn from App Inventor to JavaScript via Code Studio. The curriculum Dan chose involved very small modules where the student was tasked with manipulating an image (say a shape) in some simple way. Maybe they had to change where the shape appeared on the screen, or change its color or size. This was almost always accomplished in just a few lines of code. Modules were presented as milestones that a student was required to work through before moving on to the next lesson. Naturally, the further one progressed the harder the modules became. I have to say, this seems pretty deviously clever on Code Studio's part. The kids seemed to eager achieve these milestones, despite sometimes vocalizing a dislike of the content. Overall, there just seemed to be a higher level of participation than with App Inventor. There must be some kind of Pavlovian satisfaction in accomplishing these milestones. Furthermore, effective modularization sustained their attention.

During my last few days, Dan had an opportunity to borrow some Spheros toys. These are toy robots whose hardware is housed in a sphere that races around on the ground. You can simply play with them like a remote controlled car, or you can program them with drag and drop code in much the same fashion as App Inventor. Naturally the kids loved these.

Finally, I was able to lead the class for a day doing PicoCTF. PicoCTF is an InfoSec "video game" designed with the high-school and junior high-school student in mind. However, it can be quite challenging. Nevertheless, the kids really seemed to love it. The kids evidently express a lot of excitement at the prospect of hacking!

The effectiveness of the curriculum seemed like something of a mixed bag. There are a lot of different factors including size of the classroom, the various levels of competency, even the physical layout of the room and the energy level of the students on a given day. On average, participation seemed to hover at about half of the students. I don't have any hard data to back that up of course! This is just an off-the-cuff observation. As already stated, the kids seemed to be more generally engaged with Code Studio than with App Inventor.

Learning Outcomes from Teaching

Most of my interactions came from helping students with very basic UI problems. However, the debugging process always gave me an opportunity to dig a little deeper than the tutorial. Where error messages where concerned, I tried to help hone their critical thinking skills by prompting them to read the message and make an educated guess as to where the bug would be in the program. This is more or less the extent to which my own technical skills were utilized.

I personally got a lot out of interacting with people who are very different from me. I effectively tried to become "friends" with them first and once I got them to be comfortable with me, we could move on to programming questions. This approach might not work for everyone, but I think it's all too easy for a new mentor to go into the class with the expectation that their purpose is to tutor and as a result try to force that type of interaction. In that case, I believe there is a danger for the mentor to fall into a pattern of behavior where Teaching (with a capital T) becomes a kind of crutch because the mentor doesn't know how to interact with the students outside of that very narrow context. Don't get me wrong, there's nothing wrong with this type of interaction, but its only one way to to benefit a child's school experience and not necessarily the best.

Rapports with kids can be delicate things. If every time you work them and the only thing you talk about is programming and you grill them on what some piece of code means, then that's all you're going to have with them. Leave behind the engineering mindset. You don't always have to measure the success of a situation by whether an immediate problem was solved or not. Take your time with the kids, get to know them, allow them to trust you. There's plenty of time to develop this. When you do, you'll be in a much better position to help them later on, as opposed to trying to rush in with all the right answers. If this course was offering to help us develop our soft skills then this is it. Part of soft skills is learning to be malleable and to react to a whatever the moment happens to be calling for. In a class of 30 7th graders, that's always going to be different on a daily basis.

Learning Outcomes from the Paper and Presentations

I found my paper to be somewhat of a challenge because it was not a data driven piece. It was almost an account of a personal journey. Summarizing the paper was easy enough but I wanted to offer an interpretation of sorts too. There is an urgency to this issue that I am already aware of, and I hope I conveyed that somewhat by rehashing the examples of gender bias that Dr. Tanner elucidates. I would consider myself successful in that regard, by whether I managed to convey the glaring unfairness those examples demonstrate.

The real benefit of reading the paper was in the presentation that followed. This was a good experience being in front of people in a safe, non-hostile setting. Nothing secret to this. It's just good practice, nothing more nothing less. For my part, I definitely got a confidence boost, as I received a few compliments on my presentation style. So cheers to that!

As for the other presentations. To be honest, the process of listening was a little boring. That's not to say I didn't occasionally learn a thing or two. I did. I also recognize that the benefit of the presentations is for the speaker, not so much the listener. However, I think that can change and toward that end I will outline a few recommendations.

Suggestions to Dan

Do more Code Studio. I think this was great for keeping the kids engaged in the work. I think App Inventor is fine, and I think copying and pasting code is fine. That said, the longer and more involved that application is, the more opportunity there is for the student to lose interest. Of the applications we did with App Inventor, I think "Oh My Spikes" was just too long and ultimately didn't produce anything the kids were very excited about. And in my last circle with you guys, where the kids were asked what app they liked best, I don't think anyone mentioned "Oh My Spikes".

Do Pico CTF! You really don't have to be a cyber-security expert to do it. And the kids love it because "hacking" is cool! Also there are a ton of online walk-throughs that detail solutions. I hope I demonstrated there is a hunger for it.

Suggestions to Ilmi

Keep the class 3 units. It's worth it. There are a lot of people interested in taking this, and 3 units will lock them in. I recommend requiring at least two presentations. It's a valuable experience and frankly it's not so much work that we shouldn't be expected to do another. I mentioned that being in the audience is boring. You might consider adding as part of our curriculum a kind of soft requirement for each student to ask a question of a presenter at some point during the set of presentations. There's a case to be made that developing one's listening skills and coming up with a question on the fly is of similar value and importance to the other soft skills we are gaining experience with. For example, effective networking depends a lot on ones ability to have a conversation with a stranger. How better to do that than to really listen to what they have to say and ask interesting and engaging questions?

Everett needs more volunteers! We can be of greater benefit to the communities in the city if we go where we are needed. Some schools have students that are smarter than we are at the college level apparently. It doesn't seem all that necessary to send us there.

As far as helping to develop curriculum goes, I really admired the way Code Studio managed to engage its participants. App Inventor is alright. I think that copying and pasting from tutorials is fine for 7th graders, but I think some of the apps have a tendency to be far too long at which point, copy and paste becomes very boring. For 7th graders, small, modularized programs that can sustain their attention seem to be the way to go! Keep in mind that a busy, crowded school like Everett is going to have all kinds of different skill levels in a given class.

Also, what incentive is there to code one's own game when there are hundreds of online games that are free and way cooler than anything we can create at this level. Engineers seem to think that creating a game is an easy way to engage early students because everybody likes games! But what is the point of creating a game that you're never going to play with? Robots and hacking are much cooler. Really, if you want to engage students and hook them, I think you've got to step away from programming as it "should" be taught to letting these kids explore their curiosity.

I really enjoyed this program, it's done a lot for my confidence level, my resume and even helped to make my last semester at college much more fun than it would have been otherwise. I really hope that you'll offer this to other students again. It's a good program. Again, I highly recommend requiring two presentations per student. Thank you and good luck!

Journal Entries

Shane Cota,

Everett Middle School,
with Dan Remer,
@ 10 am - 11:45

04/18:

We should be done with Pizza Party by the end of the week at latest. Kids are starting to add information to the fusion table. Dan himself is surprised and pleased. Evidently this is his first time using this kind of application.

I get to spend a few minutes debugging Newman today. Haha, nobody likes debugging, but as I've remarked elsewhere in this journal, good debugging skills are one of the most important things are programmer can develop. Although I don't say it to the kids, it is also a good opportunity to go beyond just following slides and get into really understanding what the program is doing.

For instance, most kids resort to a brute-force approach and look for their error slide by slide. Instead, I encourage them to figure out what their runtime error is telling them and be selective in which slides they cross-reference with their code. In Newman's case he is having a connectivity issue with the fusion table. Together we eliminate slides having to do with list pickers and the like. I get the impression that he struggles to understand. I explain in a very rudimentary way what a database is and that the fusion table is basically a database that everyone in class contributes to by a connection through the internet. Again, I'm not sure if he understood.

04/13:

Still working on Pizza Party. Most of the kids are working on a portion of the code to do with list picker blocsk. This portion of the tutorial allows them to hardcode their favorite drink and pizza choices (recall the purpose of the app is to allow them to order food). This helps drive home the point of what the purpose of a list picker is, since the list in question is one that has personal meaning for them.

Bill Marsland is in class today. I haven't seen him in some time. He is interested in pico ctf and potentially bringing it to the students.

04/11:

We've moved on from MapIt and are doing another app inventor project called pizza party. This project is teaching kids the basics of fusion tables (which are also new for Dan and I). A fusion table is a Google Cloud service that aggregates and visualizes data tables. Since we've just begun the kids are mostly focused on building the UI. They are getting quicker at this part of the application and most of their questions have to do with corner case situations. As a matter of fact, in our "scrum" the question Dan decided to field to the kids was on a scale of 1-5, how easy is building a UI? Most of the kids answered 1 (the easiest) but a few of them answered otherwise. The non-one answers were surprisingly thoughtful and this indicates that the kids are in fact absorbing the lessons.

In keeping with my new commitment to branch out and meet and help different students, I spent a little time withy Bryant today. Bryant is a smart kid who displays a lack of interest. I've been working on him slowly so he gets comfortable with me and I think it's working. That is, sometimes I just talk to him and ask him how he's doing in general, as well as on the project. He seems to enjoy the rapport I am building and this may open the door for him to accept my help on this project.

04/06:

We finished up MapIt. Not as much to report today since William and I ended up in the weeds for most of the class trying to debug his program. William is super smart and finishes his work, but he always has bugs. This tends to discourage him a little bit, but I emphasize that the work we are doing

now is a very important skill to learn if you go into the software industry. So much of our work is debugging! He appears to absorb this point.

William eventually he finds his bug. Since I was working with another student in his moment of triumph he excitedly ran over to me declaring “I got it! I got it!” I give him a big high-five. :-)

04/04:

It’s my first day back since they kids have had their spring break. The energy is noticeable manageable. Maybe the week off did them some good. Not only do we finish our circle-whip quickly (in record time!) but everyone seems remarkable focused on the work they are doing. Again, maybe it was the week off. Maybe it’s the relative simplicity of the app, MapIt and getting near to finished with that. To be fair, many students are already done and they are rewarded with free time. In the end, I suspect it has much to do with Dan’s implementing of various milestones for the project in question as well as their school year. He’s constantly checking progress and it seems the kids are starting to pay attention to that.

I help anyone who still needs to finish up. Most of these students are either still at the beginning because they’ve not been focused, or they are near the end. I check on a few of my regulars. Yes, I have regulars or rather, students who trust me well enough that they will either flag me down for help or they accept and answer my questions when I approach them. Of those students, William continues to impress me. His programs never work but he shows discipline, interest and tenacity when it comes to debugging. For students like William, I help them work through this process by identifying just what seems to be broken and consider which piece of the program would probably address that. Furthermore, I have been introducing them to the awesome power of Google with varied success.

I made some remarks in last weeks meeting with Ilmi that part of the challenge at Everett is discipline. Since I do visit a handful of students with regularity, and while I do enjoy them, I see an opportunity to branch out and work with some of the kids who are apparently less interested and whom I haven’t yet gotten a chance to meet. My goal will be to bring back some information as to what it is about the curriculum that is not working for them.

03/27 & 03/30:

SFUSD spring break

03/09 - 03/23: Note to my Professors

I’ve missed a few journal entries, something Ilmi has already called me out for in class. Of those five, two were during SFSU spring break where I allowed myself a relaxed attitude. Technically, it was my break but I went to Everett anyway because I was free and I like mentoring.

With respect to their curriculum, the main thing to know is after dabbling briefly with javascript and UI formatting issues, we switched back to app-inventor to develop MapIt, an application based off

the high-level brainstorming session we did back as far back as 02/23. It's a relatively simple app, but allows the kids to interact with components they are not as familiar with such as location sensors and databases.

With respect to their attitudes, things remain more or less the same. As ever, their energy seems to ebb and flow from focus to frenzy based off school-yard influence I never get to see. Happily, more of the kids are becoming easier around me. This is a good place to make an observation about the general psychology of a 7th grader. I don't know much about developmental psychology but it seems me this is the age when a kid becomes very self-aware and guarded, while also trying very hard to fit into norms established by their peers. In other words, they are tough to get to open up to you. I've already said as much. What's funny to me, is they will let their guard down around me sometimes. In a burst of spontaneity they try to play with me by doing something extremely silly and as soon as they realize what they've done, evidenced by their sudden surprised facial expressions, they put that guard back up and close down.

Sorry about the lapse in journaling, I will maintain the normal schedule from here on out.

03/07:

The kids are working with javascript now. They are busy completing quizzes designed to get them comfortable debugging simple problems such as the placement of a shape on a canvas. Lots of formatting problems right now…

I'm making my rounds when little Amaran approaches me for help. Amaran is a small kid. If you had to guess, you'd think he was a 6th grader. He is also very sweet and friendly. These things make him stand out from his peers.

He also stands out because he seems to enjoy coding. This became apparent as he described the problem he was working on as best he could (English is his second language). He is working on a picture of a rainbow and he really wants it to look the way he wants it to. I'm not even sure how to fix the problem, but I have an idea. At this moment he is working on formatting a circle. I explain which fields in the call to circle() evidently do what. I'm not sure Amaran understands my explanation so I prompt some hands on experience instead, guiding him to try different things. He shows aptitude and quickly decides to try his own solutions without my guidance. However, together we are able to get Amaran's circle in the placement that he prefers.

Working with Amaran puts me in proximity to Bryan. Bryan is always in need of a little encouragement. He'd rather be outside playing. Still, he gives his opinion on Javascript which is one of dislike. Apparently he prefers App-Inventor. I refrain from reminding Bryan that he used to complain about App-Inventor too. The point is he is forming distinctive opinions about some of the nuances computer science has to offer rather than the more typical blanket statement that programming is boring. In this case it's the merits of one language/IDE over another.

But the work must be done, right? I prompt him to venture some answers to his quiz. He slumps in his chair. I know that slump. I still slump like that. It happens when you just don't feel like

doing the work in front of you! So instead of going into detailed explanations of how things are working I tell him that sometimes to get started, I just start hacking. Put in some numbers in those fields (he's playing with the placement of a rectangle) and see what happens. Maybe he'll notice a pattern. It doesn't take long, but we are able to work out a solution.

Funny thing is, when I turn my back on him to help another student, I look over at what he's doing again. It looks like he's playing games and I call him out. He says, "naw I'm working on the next quiz". I take a closer look. Indeed he is. I apologize and offer more help, but I'm pleased to notice that once you get him started, however much he likes to protest he can stay on track and make a real attempt at the work.

03/02:

Class cancelled for parent teacher conference and filming.

02/28:

Classes cancelled for today due to parent teacher conferences.

02/23:

Today I had multiple separate conversations with kids asking about me. When do you come to our class? How long will you come to our class? Do you go to college? When do you go to college? Where else do you teach? Oh. And, "Hey you! What's your name?" That was from D'angelo. A bright, funny, energetic if often misbehaved boy. I nearly chastised him for getting my attention with a blunt "hey you", but when I saw the sincerity of his question I softened. I am pleased at least some of the kids seem to be warming up to me.

We've moved on from the Oh My Spikes app and are starting from the ground up with a project that should last them for the rest of the year. They get to design their own application. This will necessarily entail a little more lecture time on Dan's part, which is always a dicey prospect with this group. He sticks to the "dangling carrot" approach: if they can cover the material the class will be rewarded with free time on the computer.

We watch a short video as a class, which advertises an app designed by junior high school students, and which won a competition put on by Verizon. The app helps its user navigate their route by avoiding areas where crime has been known to happen. This information is updated in real-time as it becomes available. An app like this is of special interest to the street-smart junior high school student.

After watching the video, we turn to a workbook which prompts the students with some questions pertaining to application design… the workbook is titled "Design Project". These include: who is the target audience? What problem does the app address? How does the app address the

problem? There are many more but Dan would like to slow-roll things, as he doesn't want the kids getting too squirrelly with too much written work.

We finish with a brainstorming session. Dan wants the kids to write down ten ideas on which an app can be based. It's a brainstorm so anything goes. The kids comply and many of the apps are good ideas, sensible ideas. I encourage them to get a little crazy though. It's a brainstorm after all. We can scale things back later on. I've decided to work along with the students in their workbook. I offer some crazy ideas of my own. Most of them are silly.

Bryan, one of my favorite students is having trouble. Dan has remarked before that he often works better when directed but left to his own devices, then he will get lost on some tangent. He's a fun kid, and I'm helping him and others when I finally turn to him and chide him for not working on something so easy! To my surprise he picks up his pencil and begins working in earnest. Haha. We brainstorm a few ideas together. The details are trivial. It was fun to see him respond to me that way.

02/21:

The class is full and it takes the students a bit longer to calm down today. It takes them a full two minutes to get quiet enough to even begin the circle discussion. Dan has apparently started keeping track of that too now. In general his strategy in guiding the kids and keeping them on task seems to be about bringing self-awareness and reflection to whatever it is they are doing. I admire that because it must be awfully tempting to lose one's cool sometimes. Nevertheless, it also takes a full twenty minutes to even get the kids fully ready to work front of their terminals. It is also the final day for the Oh My Spikes app. Hereafter, if a student has not completed their work then they will need to schedule time outside of class to finish. Despite the energy, the circle reveals that many of the students have made a good deal of progress since the last time I visited. Still, there remains a great deal of work to be done and many kids will need to schedule time to complete their app outside of class. This doesn't seem to bother them however, and some even appear eager to find another time to return to get the work done.

I spend some more time with William, who needed help debugging last time. Try as we might, we don't find the bug. But the app loads and runs and I praise him for finishing such a large project with so much code on time. He seems content with his work. Other than William, I make several rounds to new students. Since many of them are in the debugging stage, I get to show them a neat feature of app inventor that allows them to collapse their procedures and stack them neatly in an orderly way with just a few right clicks. This seems to come as a surprise to those I show. Evidently they were not familiar with this functionality. I take care to show them how to do it again, since it really comes in handy with large programs that have code haphazardly strewn about the canvas.

A particularly vivid moment occurred today. I was helping a boy with this collapsing feature. Nestled between several children, this boy and I somehow managed to focus in on the work in front of us whilst what seemed like the rest of the class were busy apparently boiling over in a feverish pitch of

pubescent energy. It suddenly became so loud. The thing to do might have been to stop and call for quiet, but there is no way I could contain that energy nor would I bother to try. Instead, we somehow rode it out, like a raft in rapid water and I daresay, we were successful at it too.

I've said this before, but I don't want to paint the picture that these kids are bad, or that Dan is doing a bad job. For what it's worth, I see that there are many challenges that Dan has to deal with in the classroom, and I don't think he can address them all at once. As for the kids, I think they are awesome. They are kids! I also happen to enjoy moments like I've just described. That is, finding success and something to celebrate in a chaotic environment.

When it seems appropriate, I like to interview the kids on how they like coding, among other things. So far this has been merely conversational, but I think I would like to go a little deeper. I've started to keep a written journal, where I scribble notes during class. I already have some funny observations and quotes recorded from the kids. In any case, this will give me a chance to know the students a little better and perhaps offer something of value to the program.

02/16:

There are some absent students in class today. The energy in the classroom is much lower as a result. The kids manage to complete their circle discussion in less than 40 seconds. A new record! It is the last day for the "Oh My Spikes" app, and after typing club, Dan takes a moment, as he always does, to explain new parts of the app they are working on. He never spends more than a minute or two addressing the class in this way, but it often does not go smoothly. I've seen more than one paper airplane take flight while Dan is lecturing. Sometimes he holds up for minutes at a time, attempting to quell little uprisings of noise around the room. However, today he breezes through the lecture with the students being relatively quiet. I muse how unfortunate it is the class can't be just 5 students smaller than it is. Oh well.

I feel more self-assured in what I am doing after my time on Tuesday. Recall that Dan had assured me that I am a huge help by simply being present with even one kid on a given day. Spending close time with several is even better! Very well. As always, I visit with several students in a day. This is part of a personal campaign in which I'm trying to accrue a small amount of facetime with kids every visit and in this way, slowly allow them to build up some trust in me. That said, if someone needs help, I'm glad to give them some more deliberate attention.

I make sure to sit with Bryan for a few minutes. Bryan is a plucky character with a good sense of humor. He is not afraid to joke with me immediately. I ask him where he is in his work. He's about mid-way through, which is where the rest of the class is too. He drags his feet getting started. "What do you think of app inventor?", I ask. He shrugs, "It's alright." I ask another question: "What do you like to do?" And he answers, again with a shrug: "play games n stuff." Despite his luke-warm interest, Bryan is able to make steady progress while following the tutorial.

I also decide to sit with William, who is a quiet boy. As mentioned last time, the kids commit at least few minutes to a typing tutorial in which they can practice their speed and accuracy with typing.

Presumably, the program logs their progress, because as it happens William is much further ahead in his typing skills than the rest of the students. The tutorial challenges him with english essays rather than sequences of random characters.

William is also far ahead in his progress with the app. When I ask him where he is, he explains that he is done. Great! I ask him to load the app and show me. He's already done so. The app doesn't work; the bird doesn't die when it's supposed to. By the way, Oh My Spikes is a game wherein the user taps the screen to control an image of a bird flying around. One tries to score by touching the edge of the screen while also avoid dying by not touching spikes. So in William's case, a bird will touch a spike but it won't die. I help William debug the program who is already checking the final two procedures that he wrote. They both look good, so I ask William to decide on another obvious place to look for a bug given that his bird won't die when it touches a spike. We ponder for a moment together, looking over the tutorial then jumping back to his program when something appears promising.

But going further, I inform William that I'm having trouble reading his code. The blocks are overlapping each other. I take a moment to explain the importance of readability in code especially when working with others. From there we go about organizing things neatly, again I explain, for my benefit, since William seems to have a grasp on where everything is even though it looks like a mess to me. Unfortunately the bell rings and class is dismissed before we could clean up his code and debug the program.

02/14:

I arrive a few minutes after 9:45. Dan explains where we are in the lesson plan. We are still working on the "Oh My Spikes" game, and are nearing the finish line. We are setting up the clock timers so the bird is able to fall and to flap its wings. I am ahead of this point so that is good.

The kids come in with their usual energy. I have not explained this process before, so I will do so now. First, Dan sets up a circle of chairs prior to the bell ringing. Then when the bell rings, he greets the students one by one as they enter the classroom. The students are very familiar with this process and take one of the chairs immediately. It's first come first served; there is no assigned seating for this portion of class. Once all of the students are seated, Dan leads a brief circle discussion, in which he or a student is designated the "process checker" which is a kind of team lead. This person comes up with a question to ask the circle, such that each student takes a turn giving a quick answer to the question as the circle is traversed. Sometimes the question is directly related to the assignment. Sometimes it is not. A student may opt out of answering, and many do. Once the circle has been traversed, a timer which has been running is checked. Dan likes to keep this process under a minute although their record for traversing the circle this way is under a 56 seconds, I think. Also, once the circle is traversed the process checker gets to select five students who may go to a terminal and begin typing club, which of course is a program designed to teach typing. They spend no more than a few minutes doing typing club. Otherwise, Dan selects from the group of students present those he wants

to keep committed to the circle just a few minutes longer, before breaking for work at the computer terminal. These selections are usually based on tardiness to class, lack of participation or being disruptive.

The circle might seem like a strange ritual, but the kids come in with a LOT of energy, and Dan has found that by doing this community activity, the group tends to calm down a bit and focus on the task at hand.

The physical layout of the classroom is challenging. I ask Dan about this later, and he agrees with me, explaining that he was moved to this room only recently. The class is set up lengthwise, like a long rectangle. It's difficult to be present everywhere at once, and the kids take advantage of this. Furthermore, to get access to power outlets the terminals can only face the walls, and it's harder to get facetime this way. Whatever, it is what it is, and I only remark on it now because it just occurred to me.

Like I said, the kids spend no more than a few minutes doing typing club. Not all of them log in right away. As mentioned it's hard for Dan to be present everywhere. Some kids go to youtube or some other distraction. Melanie for example gets her hair braided daily by a close friend (Dan acknowledges these two need to be split up, apparently they were very sharp before but have since lost interest in working for one reason or another). I haven't quite figured out how to approach this group. Melanie is quite clever and knows just what to say to divert an adult from focusing on her, unless they are committed to discipling her. Speaking frankly, I am not and Dan and Bill agree being an authoritarian is not something I should try. Dan also assures me that my presence, focus and questions to some students is having the desired effect. It's not necessarily apparent to me, because I don't get a chance to see what things look like when I am not there! But it is apparent to him.

With that in mind, I am selective about who I spend my time with at this stage, although I at least try to greet as many students as I can. Breaking the ice is a slow process. I think they just need to see me be a consistently present adult, before they will begin to trust me. Apparently they see a lot of adults walk in and out the door.

In any case, the coding begins. Although the lesson plan Dan anticipates the students are near the finish line, most of them are quite a bit behind. As an incentive to stay on task, once certain milestones are reached one gets a few minutes of free time to be on the computer. Dan also makes it a point to tally progress and this further incentivizes the kids to stay somewhat on task.

I make my rounds and when the kids appear to need help, I tell them to ask their question of me. I'm getting more comfortable with app inventor so my ability to be of use in this way is progressing. These kinds of questions have mostly to do with how to set up calls to procedures and simple things like this.

02/09:

Today I get to know more of the kids. I think this is going to be a general theme for awhile. Most of them don't really trust me just yet, but hopefully, when they see that I am there consistently they will

begin to open up. With some that is already happening. With the others, I am patient and simply make an effort to engage them. Often I am rebuked with indifference but if I can get a name and some conversation then I think that's a small success. As for coding, I am realizing I can be of some help with style. Since app inventor is drag and drop, even finished products can look like a hot mess as far as readability is concerned. Over the course of the semester I would like to impress upon the students the need for style and readability. The Oh My Spikes app is nearing completion for many students too. So there is an interesting and fun opportunity to help them debug their programs, which also plays into the need for readable code. Debugging also affords an opportunity to help them understand what they've done a little more deeply. A student named Amaran, got his app to load but it wouldn't run the way it was supposed to. I get to ask him if it won't run, which procedure should he look to, to debug. Alas, we ran out of time, but this was my first chance to ask deeper questions. Of course, there will be more opportunities.

02/07:

Since Prof Yoon will read likely read these ahead of our meeting on Thursday, and one of my appointments at Everett is also on Thursday, my two journal entries should lead with a Thursday entry and finish with a Tuesday entry, which is today. :) --- Good guess!

As was the case last time, Dan splits the room into folks who prefer to have a little extra help and those who feel like they can do it themselves or help each other. He asks me to focus on the kids who are more independent. Nevertheless, the kids feel especially rowdy today. Maybe it's the rain. I'm not comfortable commanding the kids… I don't think they would respond to the new guy telling them what to do. Instead, I attempt to calm them down and focus them on their work by offering help, and asking questions about their code.

They are not all rowdy. Some of the kids are really sharp and on top of it. William has done a lot of work, but his code isn't organized in way that's readable. I encourage him to clean it up at some point, explaining that software engineers must make their work readable for others to be able to follow. He doesn't seem convinced but maybe this is something I can help him improve on over the semester.

I approach some other boys, who seem to have a good grasp of what's going on. They often ask each other how to do something. One boy explains that he is bored with the app-inventor curriculum because he doesn't get to do what he wants. It is essentially copy and paste. He told me that before, the coding projects gave them much more leeway. Apparently, they were coding their own games before using scratch.

After the kids leave, I ask Dan about what role I should be assuming as far as keeping kids on task goes. He assures me that bringing a calming presence just by being next to them and asking pertinent questions is really effective and helpful.

02/02:

The kids started a new project on Monday, a game based off a popular “angry bird” game. Yesterday, they spent their time declaring and initializing global variables, and today they went to set up some procedures. Since they are in the beginning stages of development, I will spend my weekend starting and finishing this project.

01/31:

Met the kids today. They are a rowdy bunch but fun. We are using app inventor. I spent most of my time meeting the kids and watching them finish-up a project.