Monday, November 13, 2017 (Day 1)

Today, I finished up soldering my braingame. I was nearly finished and only had to solder my LEDs and the battery pack, but when I put the batteries in, I discovered that I had put my diodes in the wrong way. So, the braingame wasn't working. Mr. Dubick helped me remove the solder from my diodes so that I could pull them out and switch their direction. I quickly switched them around and soldered them, and then everything worked. I also discovered today that the some soldering irons are easier to use than others.

Tuesday, November 14 (Day 2)

I was able to meet with Mrs. Coble and Mr. Dubick today to discuss what my final project will be. We concluded that my idea for a wearable skating vest that measures your rate of rotation, height in the air, and maybe a sensor for if your air position is tight enough. To do the first two things, I researched and found that a gyrometer measures the rotation of an object around its axis and an accelerometer measures movement (or micro vibrations) on the x, y, and z plane. By using both of these sensors, I can achieve what I'd like the vest to do. I have to do more research on how to use both of these devices, which ones would be best for my project, the respective prices, and how to code for them. I also have to figure out how I will display my statistics and how I would do that. I have a lot of research and work to do ahead.

Wednesday, November 15 (Day 3)

I worked on the design specs of my project, and I determined the purpose of the project, which is to be able to measure and display/access these statistics. I will do this through the use of both an accelerometer (to measure the height of a jump) and a gyrometer (to measure the rotation of the skater). The gyrometer will measure around the y axis, and I have found some possible gyrometers online. Since I figured out that I only need to measure around the y axis, I was trying to find a gyrometer that only senses one axis. This would allow me to cut down on cost and find a higher quality accelerometer. If I cannot find a single axis gyrometer, can I program the 33 axis gyrometer to only pay attention to the z axis? Yesterday, I was more concerned about finding an appropriate gyrometer, but now, I worry about the accelerometer. How will I read the information that it records? I am still quite confused about the usage and coding of both of these aspects, and I worry about time constraints for ordering my materials, especially since I will be missing days 6, 7, and 8.

Thursday, November 16 (Day 4)

I finally figured out what my gyroscope and accelerometer I will be using today: it is the Flora 9-DOF. It is an accelerometer, gyrometer, and a magnetometer all together! Later in the project, I may also add the Flora Wearable Bluefruit LE Module to transmit the information recorded on the Flora 9 via Bluetooth. However, this would be done at a later time and only if my initial goals are met. I spent most of class researching how to program for another type of accelerometer and

gyroscope, but then at the end of class, Mr. Dubick recommended a different type of sensor. So, some of what I did in class will not be applicable. However, much of what I discovered would still probably be useful when I code. For example, I learned that you can set a minimum degree per second so that it will automatically discard the information if the rotation is too slow. So, this already narrows down your data to maximize efficiency when viewing this data. Essentially, today I learned the basics of coding for accelerometers and gyroscopes.

Friday, November 17 (Day 5)

I finalized my design specs and inputted it into my Digital Portfolio. I also realized that I needed to add the battery pack to my materials because the vest, obviously, cannot be plugged into a wall while it is in use. I had difficulty determining the voltage for the Flora 9-DOF, but I eventually found that it is 3.3V but has a converter. Apparently, it has a converter that allows it to accept up to 5V because many people, like myself, use Arduino, which is 5V.

Tuesday, November 28 (Day 9)

Since I was absent for a few days, I only filled out my materials on the materials spreadsheet. I made my document for the code of my accelerometer/gyro, and I've started to set up the code for it. I calculated that for a single rotation, skaters approximately rotate at 1800 degrees/second on lower jumps.

Wednesday, November 29 (Day 10)

I edited and improved my other, previous daily engineering entries. I also found statistics on the average airtime of a skater and realized that my maximum rotations per second should at least be 2037 degrees per second if I want to measure up to a triple lutz. A double axel rotates at approximately 2000 degrees per second. So, with this new data, I disproved what I did yesterday and found more exact rates. Yesterday, I only estimated, knowing that skaters doing doubles are in the air for approximately .4 seconds. However, today, I realized that I might want to measure more difficult jumps, causing me to need a higher range.

Thursday, November 30 (Day 11)

Today, I went through the code more thoroughly and learned about the different terms included in the coding. I did research on terms like "gyroZeroVoltage", "gyro drifting", and "gyroSensitivity". I still struggle with understanding gyroSensitivity, since I don't really understand the unit (mV/ degrees/second). However, I did find a site that explained it, and I will go through it more in depth tonight or tomorrow in class. For some of these terms in the code, I don't really understand them, even though there is an explanation of each line of code. I have to look through it more. Now that I know about gyro drifting, I worry about how much that will affect my data in the future. Though there are multiple ways to fix gyro drifting, I also don't

really understand these methods. I will continue to do research on all of these terms and any other terms I come in contact with.

Friday, December 1 (Day 12)

I started to use the Adafruit Circuit Playground Classic as practice today. Most of the class was spent troubleshooting why it wouldn't sync with the computer. Though we already checked that it was connected to the correct port, it wasn't connected to the correct board. It had to be connected to the Adafruit Classic Board, but it was originally set on the Arduino Uno Board (we always use that board in class).

Monday, December 4 (Day 13)

Today, I experimented with the Adafruit Circuit Playground and got familiar with the values of each axis. Though I could have probably inferenced this from the name, it only measures acceleration. Essentially, if I bring the accelerometer down, then the Z values will drop and then even out again. This is because it, once it reaches the bottom, is no longer in movement. The Z values tend to stay around 9, meaning 90 degrees. Flipping the Adafruit Circuit Playground upside down (so that all of the sensors are on the bottom) will output a value of -9. When you tilt it right to left, the X values get farther from 0 as you increase incline either way. The Y axis functions similarly, just with forward and backward. At the end of class, I found instructions on Adafruit to code for a bike glove. Though I don't actually want to code a bike glove, there are certain code elements that seems like it could help me with coding for my vest, like naming a specific range something.

Tuesday, December 5 (Day 14)

I took the code given by the guy who made the Glove and used it to identify different spin positions, camel and layback. I determined where the accelerometer / flora would rest (the skater's right shoulder, sewn to the inside of the vest) and then experimented with the values. A camel spin is where your body is forward, so it would be within the range of -5 on the Y axis. The layback spin is like spinning in a half-backbend, so its values would also be on the Y axis. It would be greater than 5 on this axis. Much of the class, my code didn't identify the movement pattern as "Camel", and I couldn't figure out why. However, my layback coding was successful.

Wednesday, December 6 (Day 15)

Now that I had my identifiers of Camel and Layback, I had the lights blink different colors when each one is performed. So, when a camel is detected, blue lights flash. When a layback is detected, the lights flash pink. To do this, I had to code them so that all of the lights turned on and blinked off, using delay to determine the amount of time the light is on and off for. I, however, could not figure out how to set a variable for the numbered neopixels. I ended up turning each light on by using CircuitPlayground.setPixelColor(0, NEXT_COLOR); and changing the number values for every neopixel. This was really inefficient, but I really struggled

in trying to figure out how to do it any other way. Thankfully, switching all of the lights off was much simpler (CircuitPlayground.clearPixels();)I also experimented with light colors today after finding a diagram on the different codes for different colors on neopixels. These different colors were determined by #define -ing what NEXT_COLOR was and then using the term NEXT_COLOR in my code.

Thursday, December 7 (Day 16)

The gyro finally arrived, and initially, I had no idea what I was supposed to do with it. I later discovered that, using alligator clips, you can attach it to the Adafruit FLORA. However, I had a lot of trouble finding code for the combination of the FLORA and the FLORA 9DOF (9 degrees of freedom). Most of the tutorials on the FLORA 9DOF used another arduino accelerometer called the LSM9DS0 connected to an Arduino Uno. Although FLORA can simply be used as an Arduino board like the LSM9D0, there are no tutorials on it. I even stayed after school trying to figure out what I should do and how to code, but there was nothing.

Friday, December 8 (Day 17)

Since nothing was uploading to the FLORA, I tried to just blink an LED. However, it didn't work. It kept giving me the error "Error compiling to board "FLORA"." I eventually realized that I had to download a bunch of library boards onto my laptop. My laptop, with its standard Arduino settings, was not compatible with the Adafruit FLORA. Although Adafruit is a popular brand, several extensions must be added to make Arduino compatible with it. I added the "Adafruit AVR Boards" onto my Arduino, and I also ended up doing this with the school computer, as well. The instructions for doing downloading these boards are on my workflow!

Monday, December 11 (Day 18)

I got so much done today: I fixed the error with the board, figured out what I was confused about, and finally got the gyroscope working.The error that said "Error compiling to board "FLORA"" continued today, but only on my laptop. For some reason, the school computers seemed to work. I found out that the difference was that the school computers already had a library called "Adafruit Unified Sensor." So, I went online and added this library to my laptop. I also added two more libraries today: the Adafruit LSMDS0 Library and the Adafruit Sensor Master Library. Today, I realized that the LSMDS0 is the FLORA 9DOF, but just its serial number and identifier in code. This whole time I had thought that the LSMDS0 was another accelerometer / magnetometer / gyroscope interface, but it turns out that it was the one I already had. The FLORA 9DOF is just the standard name, not used in code. So, now that I realized that, I could now use the multitude of online codes. I ended up using the examples in the Adafruit LSMDS0 Library examples, which are the "sensorapi" and "lsm9doftest." The "sensorapi" has a more comprehensible layout of outputs, meaning that the data it collects is easier to read. One problem I encountered was the fact that the alligator clips I used kept on slipping off of the gyro,

causing it to lose connection. Tonight, I will sew the FLORA and the LSMDS0 to a piece of scrap fabric with conductive thread so that the connection is not lost so easily. In the future, I need to figure out what I will do with this data. Can I create a model of the skater? How would I do that? What purpose would that serve to the skater, if they can just see a video of themselves and automatically know their errors?

Tuesday, December 12 (Day 19)

Today, I sewed my Flora and 9DOF onto a piece of fabric so that the flimsy alligator clips don't disrupt the connection. I just got a scrap of fabric from the art department. The code uploaded to my Flora, but as soon as I tried to open the serial port, it gave me the error "Board at /dev/cu.usbmodem[number] is not available." So, I tried to do it on the school computers. My school computer gave me the same error, so I moved to the computer that we downloaded the libraries onto (the one beside me). This one didn't really work either: though I was able to open the serial port, I wasn't able to view any of the incoming data. The title in the serial monitor showed that the program was running, but I wasn't getting any of the incoming data from the gyro. Also, when it does run (maybe 50% of the time), when I start to move it, the values sometimes just stop loading up. I was thinking that it might be my connection and I may have to sew the conductive thread again.

Tuesday, December 13 (Day 20)

I continued troubleshooting my gyro today, but I realized that the error messages that are coming up aren't because of my code. Either something is wrong with the hardware or the computer isn't connecting properly to the gyro. Every time I tried to view the serial monitor/plotter, it would say "Board at /dev/cu.usbmodem is not available." So, I tried to connect the gyro and Flora with just alligator clips to see if that would fix anything. That didn't work. Mr. Dubick suggested restarting my computer. This did help and I could open the serial monitor. However, once the serial monitor was open, I didn't see any values from the gyro. I just got the title of the code and a blank screen underneath. Finally, I looked at how I sewed the conductive thread. It turns out that some of the excess threads were touching each other, shorting out the gyro. I trimmed the thread and everything worked perfectly well.

Thursday, December 14 (Day 21)

I bought the Flora Bluefruit and started setting everything up! For the hardware, my Flora had solder on the 3.3V and GND. I removed all of the solder. I haven't attached the Bluefruit to the Flora yet, though. For now, I'm still using alligator clips. I also couldn't get the Bluefruit to factory reset, but it seemed to connect to the iOS app. I spent a lot of the class trying to figure out why the LSM9DS0 didn't connect to the Bluefruit, but it turns out I was looking at the wrong setting. I was looking at the gyro stabilizer instead of the plotter, which shows the number values and puts it into the graph (similar to the serial plotter on Arduino). I still have to figure out why it

isn't sending any information to my phone. Furthermore, I realized that the Bluefruit is only discoverable if it isn't connected to my laptop (and subsequently Arduino) because all of the information is going straight to the computer. If I plug it into, for example, just the wall, then it is discoverable.

Wednesday, January 3 (Day 22)

I realized that the Bluefruit app can only discover the data if it is in the CSV format, so I have to figure out how to format the serial.print to make it show up in the CSV format in the serial monitor. When it shows up correctly in the monitor, then it should go through to the app. I eliminated the accelerometer, thermometer, and magnetometer functions from my code so that I could isolate the variables and try to figure out which color represents each variable. Generally, today I just experimented with different aspects of the code to figure out how to put the values into CSV format. On the Bluefruit Connect LE app, we updated to version 7.7, but this caused the Bluefruit to not be discoverable, meaning that the app couldn't find it at all.

Thursday, January 4 (Day 23)

To troubleshoot my connectivity with the Bluefruit app, I tried to connect a neopixel and change the color using the app. In this process, I realized that I was missing key aspects of bluetooth coding in my normal gyroscope code. Furthermore, I stumbled upon more examples that showed me how to utilize this code and maybe how to implement it into my own gyroscope code. Initially, I was going to use the neopixel as connected to a normal Raspberry Pi, but then I realized that the coding for Flora and normal Raspberry Pis are very different. I opted to just connect the neopixel to the Flora anyway, just for practice for coding my gyroscope later on. I also used much of the class time finding multiple types of code for the neopixel and trying to decide which one to actually use. Today, I also practiced using the soldering iron when assembling/fixing the neopixel wiring.

Friday, January 5 (Day 24)

I think I short circuited my neopixel, so I had to solder a new one. This ended up being a lot more difficult that I originally thought it would be, so I spent a lot of time on this. Then, I tried using the Bluetooth feather. I actually got the Feather to communicate with my phone and computer. I used the "bleuart_cmdmode" to send messages between my phone and my laptop, as seen in my documentation video. For the Feather, much of it worked initially. I think this may have been because the example code within the "Adafruit_BluefruitLE_nRF51" examples were designed for the Adafruit Feather and not the Bluefruit Flora (which I have been using). I didn't have to alter the code at all and simply had to upload the programs to the Feather. However, all of the example programs were to communicate from the computer > bluetooth module > iPhone app. I don't know how to code a sensor > iPhone app, and there are no example codes for something like this. So, I don't think I will be able to get the bluetooth to work for this project. I

could finish this project and get the bluetooth to work as my final project for Engineering Design 2.

Monday, January 8 (Day 25)

I worked on my digital portfolio: I started my slide presentation, my final reflection, and edited my daily entries. I started sewing my vest, but I kept on having to restart because I positioned the Flora or the 9DOF incorrectly. Then, when I had already sewn one connection, I decided to try the LiPoly battery. The battery didn't work! I plugged it into the Feather just to check, and it worked. So, the problem is not with the battery but probably with the Flora's port. I may just use the Feather. Anyway, it will be easier to connect to bluetooth later if I use the Feather.

Tuesday, January 9 (Day 26)

I wanted to sew my vest, but I left my needles at home. I tried using Grace's needles, but the eyes were too small. The conductive thread is very thick, so smaller sized eyes do not work. I also continued working on my slideshow, labelling each slide to show what I was doing in each picture.