**Information & Intelligence**

**Simon Says**

Deveshwar, Gerald

****

Code written by: Deveshwar

Hardware wired/designed by: Gerald

**Information and Intelligence: Simon Says**

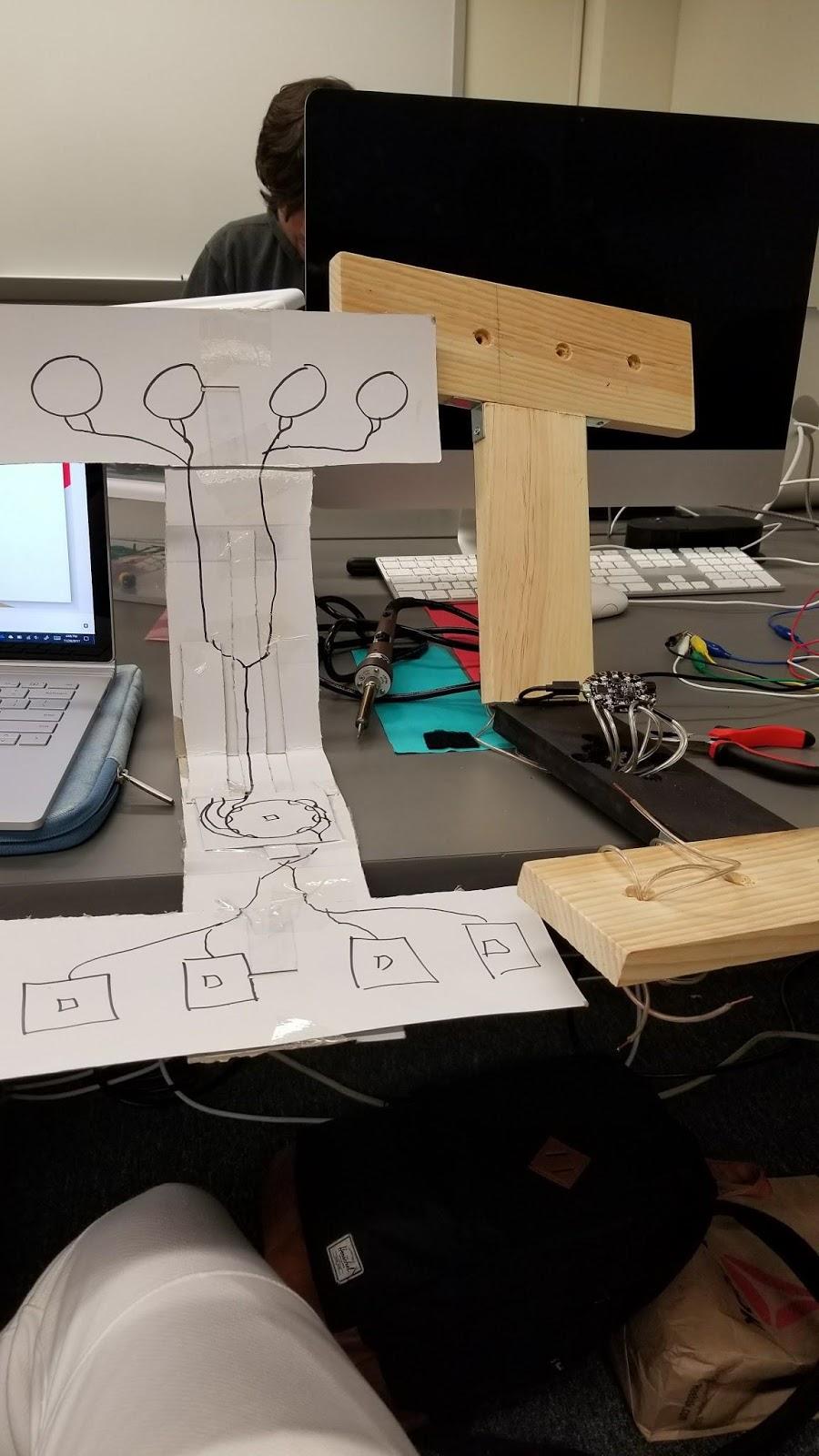
The total of our project relies heavily on the concept of output and input. On the Adafruit Circuit Playground board itself, there are 8 capacitive touch pads. After some research we found that these pads can actually be turned into voltage outputs and that’s where our project began. Instead of using the tiny LED lights on the board itself, we came up with the idea of scaling things up a bit. We brought external LED’s into the equation and with the output voltage of the touchpads, we were able to turn them on and use them for the game. With the 8 capacitive touch pads, we kept 4 of them inputs and 4 of them outputs. With that settled, we connected the LED’s onto the 4 output pads. Now all that’s left is the writing the codes for the game and instead of playing with the touchpads and the Adafruit Circuit Playground tiny LEDS, were using just the touch pads.

As a default, the Adafruit Circuit Playground board’s touchpads are set to input, and they read how much voltage is being put into them. This can be useful when you want something to happen when a certain amount of voltage is running through the touch pads. Instead of using readCap, we used digitalRead, this is different from readCap because readCaps reads an amount of electricity, digitalRead is used to test if a circuit is completed. We used the input touch pads to work as buttons. Eventually we connected actual buttons onto the pads. These buttons, when pressed would complete a circuit that tells the Circuit Playground that it's being pressed.

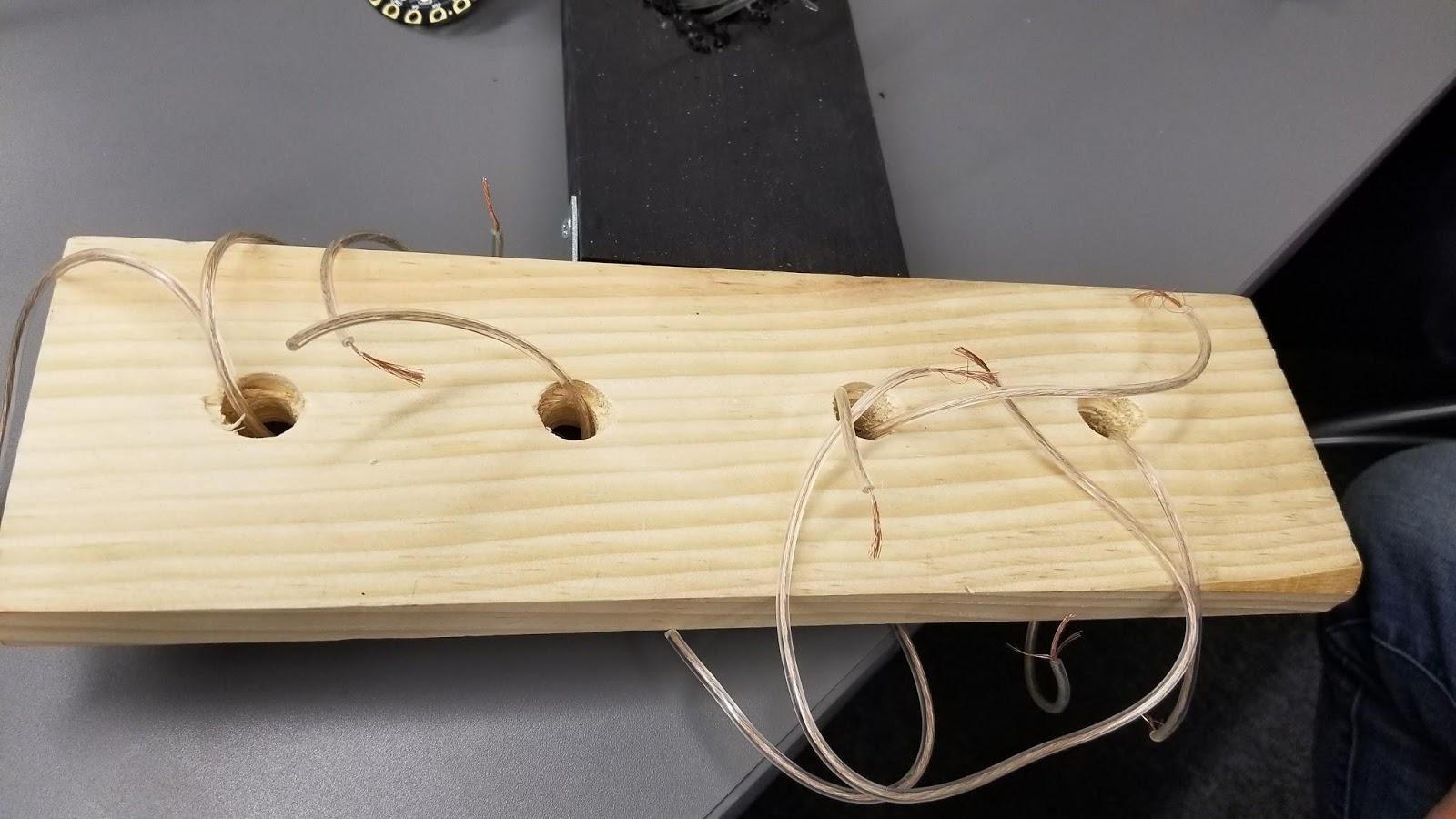
Apart from manipulating the regular Adafruit Circuit Playground, we also worked with the Adafruit Board language itself. We used a special code to control the board and make the capacitive touch pads give off an output voltage. Normally those pads would be set to default as inputs that read voltage which is also as important. Although what we did was we used “pinMode” and “digitalRead”. With these functions we were able to manipulate the touch pads and set its “pinMode” to equal OUTPUT. That made the pads into an output instead of an input. It is now an output but is it giving off any power? That’s where “digitalWrite” comes in and we turned the touch pad’s voltage on by setting it’s “digitalWrite” to (desired pad#, HIGH). To summarize the “digitalWrite” code, it takes two parameters, the number identity of the pad, and “LOW” or “HIGH” with “LOW” being off and “HIGH” being on.

It is truly hard to create a game without intelligence. For our game, Simon Says, we coded a lot of intelligence into the Adafruit Circuit Playground board. We used certain number of if statements connected to the buttons. We used a code that came up with random numbers every time and each number would correlate to a touchpad that was giving off power to an LED light if the board decided it should. We used if statements to work with the buttons. If a button was pressed the board would decide if it was the correct input (as in if it matched Simons color) and depending on the players input, it would either give a point or let the player no it's the wrong answer. With the random numbers generated for Simon’s turn, we had to store them somewhere in order to let the board decide later if the player’s input matches Simon’s color/number. We decided to use an array to store Simon's choices. The next big issues we came across was finding a way to allow the user to input more than 1 input and how to check if it was right before moving onto the next stage. It took us a while, but we finally got it to work with a nested loop. The last issue we ran across before time caught up to us was mapping, the leds/numbers Simon has to choose doesn't match up with the numbers for the capacitive touches. The leds go from 0-4, whiles the capacitive touches go from 3,2,0,1.

As far as physical challenges go, there were many obstacles that were in our way that prevented us from moving on forward. Figuring out how to set the pads as output was the first main issue we had. Turning the pads into outputs was the foundation of our entire project. Without that, our project would have been limited to a much smaller size. Through just plain searching things up on google, we found the codes mentioned earlier and we were able to make this as big as we wanted. Another challenge was making the actual prototype. As a first prototype, we put our idea onto paper. Once the blueprint was finished, the next step was to build a 3d model. With cardboard and some tape, we were able to make this.



After a while with some tools and hard work, we came up with the second prototype and hopefully not the last.



The challenge came with working with the wood. With some tips from Home Depot workers, we were able to drill holes into the wood making actual hollow places for the buttons, lights, and board. Instead of alligator clips which were messy and came off easily, we decided to use speaker wires. When we went to present, we ran into a problem with our 3rd prototype and had to revert back to our first prototype for the demonstration. In the 3rd prototype there was a short circuit with the speaker wires causing one of the buttons to be read as always being pressed. There is still things we want to add to our Arduino like more lights and buttons, and different stages and game modes, but we liked the way our Arduino says turned out.