

Week 15:

Date: 12/2/2022

Hours: 14

Description of design efforts:

Piezoelectric Cleanup:

This week I spent a lot of time cleaning up the hardware on the table and preparing our mechanical structure for our projector and camera system. Any excess wires on the piezoelectric sensors below the table were trimmed to the optimal lengths, where we can reach the PCB connectors without any hanging wires. I also added new, better piezoelectric sensors to handle bounce detection. While the crystals are the same size as the old ones, the larger metal plate it is attached to seems to have increased signal strength. The new piezoelectric sensors were connected to the table using super glue, instead of painter's tape like our old sensors. The old sensors have been left on the table, though, with their wires neatly folded, in case of broken sensors. The bottom of the table is shown in Figure 1, below.



Figure 1. Bottom-of-table wiring.

This caused my code to be overly sensitive, and I had to increase the hardcoded threshold for a “bounce”. Though, the increased strength is a good thing – it reduces the chance for incorrectness (which was already nearly 0). The only problematic situation occurs with the net attached, removing our physical barrier between the two sides of the table. Here, we *can* accidentally interpret a ball hit *on* the net holder on one side of the table as a bounce on the wrong side. However, there is almost no chance the ball will ever hit the net holder, or even in the front few inches of the table.

The increased signal strength in the piezoelectric sensors also required me to increase my “bounce timeout”, where we do not sample the ADC for a few milliseconds directly after a bounce occurs. Originally, this was set to 10 ms, where up to 5 bounces could be detected now on a very strong hit. Now, with this set to 100 ms, we never detect multiple bounces from our testing. Though, this does result in the possibility that a quickly bouncing ball may not trigger every bounce. This could happen, for example, if a player accidentally traps a ball in motion below their paddle, causing it to bounce multiple times. Under this circumstance, we may miss *some* bounces, but the ball should bounce long enough that at least a second bounce will be caught. Then, extra bounces will be ignored by the laptop, as a score has already been counted.

Mechanical Structure:

I began to construct our mechanical support system to hold our camera and projector, as we have our aluminum t-slots cut out to our needed lengths, and had some connectors delivered. The structure is not complete, as we are awaiting more connectors (and the possibility of added aluminum bars for extra support). But, even with our small number of connections already made, our structure is incredibly strong and static. A demonstration of this strength is shown in Figure 2, below.



Figure 2. Support structure strength demonstration.

Game Testing:

As we have our working version of basic ping pong, with automatic scoring, we began to test that logic, as well as our projector display menus. The best way for us to find bugs in our game, we have found, is by playing and acting like children. Someone reasonable wouldn't try to decrease their score below 0, but that is something we must ensure cannot happen. At this point, there were very minor bugs to fix, such as the exit button not actually exiting the game.

With our "vanilla" game working properly, we began to integrate the parts of our "dropshot" game to work properly. We have ball tracking, bounce detection, projecting, and scoring logic, but needed to consolidate them to work together. All the logic needed currently exists but is not implemented. That is, besides our projector displaying the shapes for the ball to "break" in dropshot. Then, we need logic to determine which shape should be "broken", which I have written but not yet tested.

Next Week:

Next week, we have little left to complete, "content" wise; a few missing pieces of our code will be filled in to ensure our game mode works. Beyond that, we will continue to build our support structure and test it, continue game testing, and begin preparing for our demonstrations.