Week 7:

Date: 10/7/2022

Hours: 10

Description of design efforts:

Microcontroller Programming:

Code on the microcontroller this week was rather simple; I just finalized many parts of the code that had already been written. This includes refactoring the “left” and “right” side of the table to “red” and “blue”, adding error reporting to packet sending, and adding safety checks for if the code stops running. Beyond minor tweaks we run into in the future and modifying any calibration logic or any work for microphone array trilateration, the microcontroller code is complete.

Game Logic Programming:

Most of my focus this week has been on programming the game logic on the laptop. This code will be in C++ to better incorporate with the OpenCV library. I have never touched C++ code in my life. Though, I did not find it too bad beyond some syntactical formats for object creation, as I am very confident with C programming, and I have significant experience in other OOP languages (Java and Python).

I spent a large chunk of time planning how the game logic code would look. We decided on using 2 threads – one to handle the main game logic and communication (UART connection to the microcontroller), and one to read and process camera data. These were designed with synchronization in mind – we have chosen to have as little variables as possible that need to be accessed by both threads. As of our current knowledge, this is made up of:

1. The current X-coordinate of the ball
2. The current Y-coordinate of the ball
3. A recent, previous X-coordinate of the ball
4. A recent, previous Y-coordinate of the ball
5. Variables for calibration

As the game logic thread is responsible for printing information to the screen, it will need to know the ball position, which it will read from the camera thread. It may be relevant for the game to know where the ball previously was, to determine the direction it is moving in, and to help it figure out if the ball has left the playing area (and thus, a point should be awarded). Calibration is another issue that must be figured out. For the piezoelectric sensors, this will be fully handled on the microcontroller (if even needed). The projector calibration will have to be calibrated manually, as that can only be changed through positioning and mechanical settings on the projector itself. However, the camera calibration will require more work. We need the position reads from the camera to be “in tune” with the projector. That is, if the camera reads that the ball is at (200, 300), we should be able to turn on blue light on the projector at the equivalent of (200, 300), such that it illuminates the ball. This system will require communication between these 2 threads and will be a difficult task to resolve.

Next Week:

Next week, I intend to pursue completion of the following tasks:

1. Prepare for and complete our midterm presentation
2. Continue working on the game logic code – ensure proper integration between threads with no deadlock
3. Create basic game logic and scoring system