

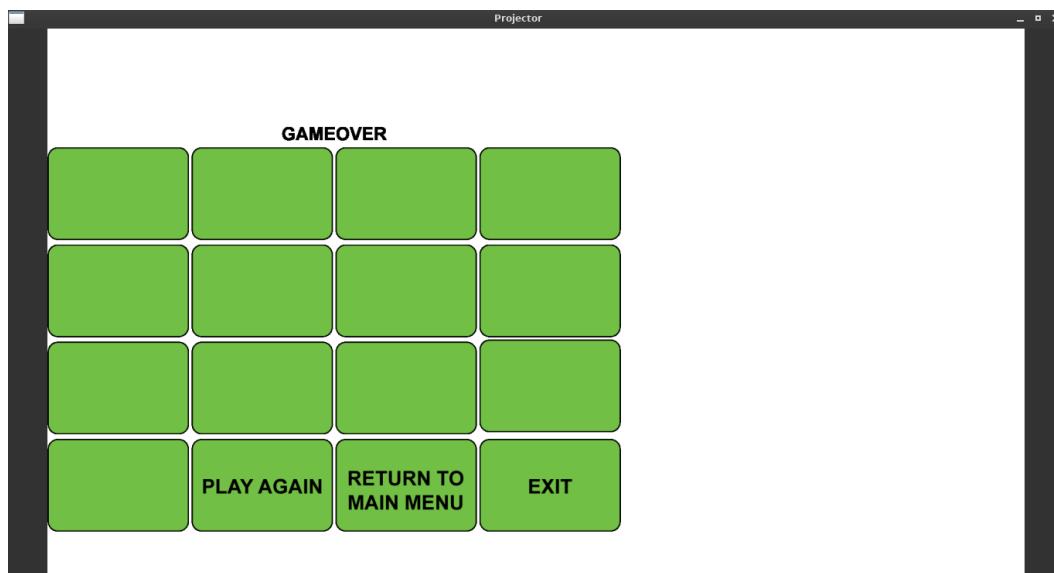
Week 11

This Week:

Display Graphics:

This week I ran and debugged the projector functions that display TIFFs to an OpenCV mat. The biggest change was tweaking the coordinates that read from the RGB buffer. Additionally, on the project laptop, I included tiffio.h into the build requirements. This change is seen in our SConstruct file.

After initial debugging, and a bit of procrastinating, I added TIFF scaling to the render function. The function call now takes a *scale* parameter. This parameter is a float, and describes the size of the final rendered image. For example, a scale of .5 would produce an image that is half as large as the original.



TIFF Rendered and Displayed using an OpenCV Mat

Contour Detection:

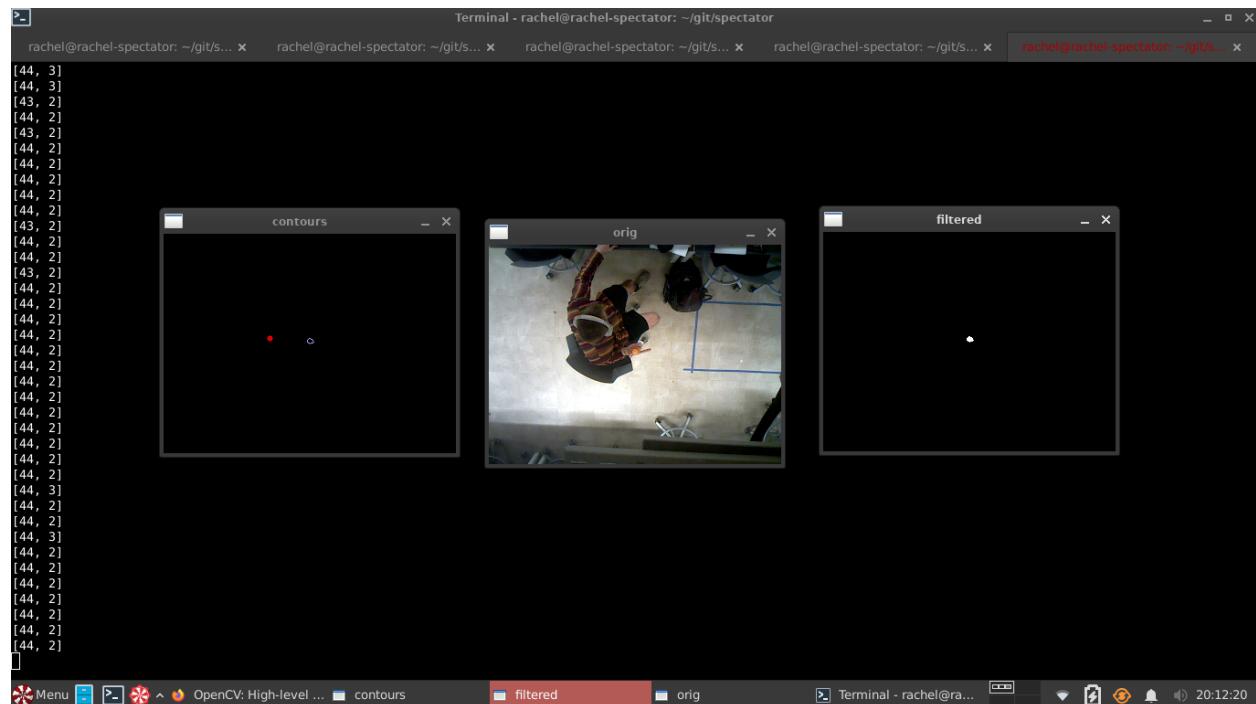
A goal of our contour detection system is to end up with *scaled* table coordinates. To do this, we must specify the corners of the table. This can be done with a number of strategies. A few that we have considered are:

- Depth Camera → edge detection
- RGB Camera → corner detection
- Manual Setting

To start off we will use manual positioning of the table corners. To do such, I created a little UI loop that draws a small point to the screen. This point is adjustable through the keyboard. Once the placement of the point is confirmed, the ball detection algorithm kicks, and uses the point as a reference for ball coordinates.



Point Positioning Screen

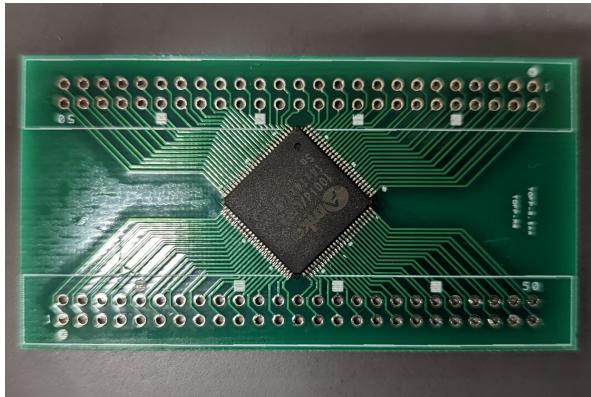


Adjusted Ball Coords seen to Left

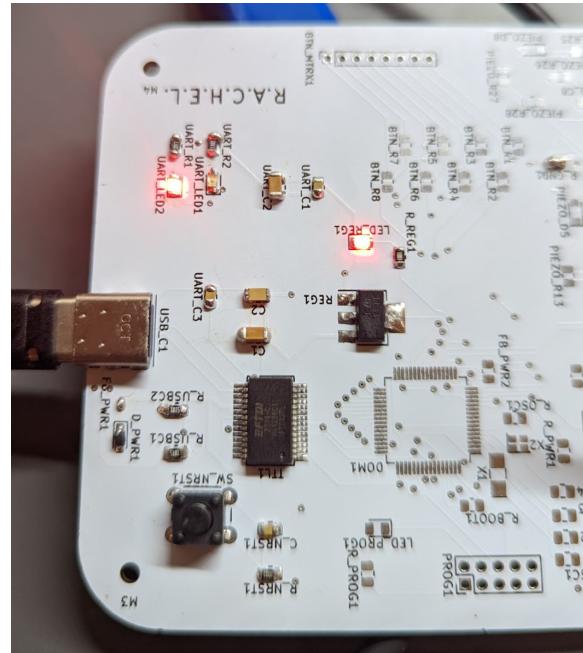
Hardware:

This week we started assembling our PCB. The first thing we soldered was the power system. I soldered the voltage regulator. Additionally I practiced drag soldering using the available resources in the lab

Once I was confident in my ability to do so, I soldered the FT232RL to our board. James assembled the rest of the UART circuit, which involved a few caps, and the LEDs and resistors to visualize TX and RX lines. After a bit of debugging we were able to light the RX LED by sending a USB message.



Test Drag Solder



Power LED (Right) Rx LED (Left)

Next Week:

Table Detection:

Next week I will extend the point positioning algorithm to 2 corners (assuming that the table is a perfect rectangle) and use the manually positioned corners to get a normalized coordinate for the ball. The top left corner will have a value of (0,0) and the bottom right corner will have a value of (2,1). Coordinates of the ball will be interpolated between these two values. I will also create sidedness EQs to determine if the ball is inside the bounds of the table.

Farther Forward:

Given the time and resources there are a few things that can be done to make this process more seamless and accurate. The different strategies listed above can be used to automatically gather the corner positions. Additionally, extra linear algebra can be conducted to avoid making the lofty assumption that the camera views the table as a perfect rectangle. This will require us to place all four corners, and somehow place the ball on this imperfect surface. I will look into this algorithm, if it is necessary.

Hardware:

PCB assembly is full steam ahead. The next step is to solder on the MCU and test UART messages. Then after that we will solder the piezo circuits and terminals.

When we get the packaging comes back from the machine shop we will assemble the two PCB and attach the product to the table.

Gamer Logic:

The logic for the menu system is still not started. We have the ability to load TIFFS to show menu controls, we now need to implement the state machine to show these graphics to the table. This menu system also needs to initialize the correct objects.