Eye tracking direct participant comparison

By Zelda Ferris

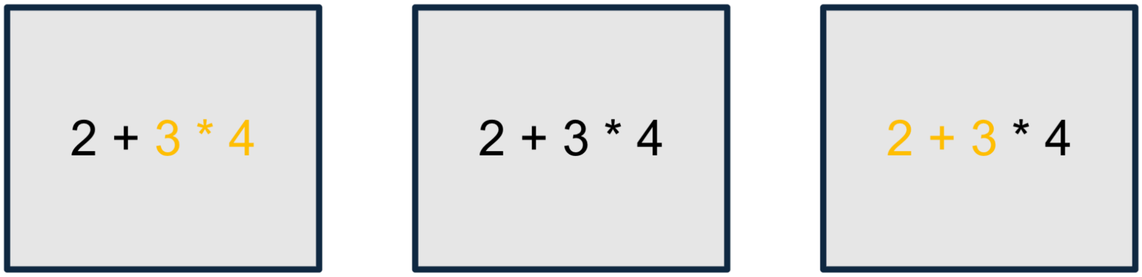
Github repository: https://github.com/zelda-f/eyetracking_vis

Overview video: https://vimeo.com/1080669072/0155ef2e6b?share=copy

Contents:

Phase Zero: The study
Phase One: Rendering dots
Phase Two: Coloring based on AOI
Phase Three: Trying to simplify and make more sense of everything
Phase Four: Add stimuli, filter data
Phase Five: Putting it all together…
Phase Six: Design break
Phase Seven: Density contour (singular)
Phase eight: One to many!
Phase Nine: Backend work
Phase Ten: The final product

C vs. N vs. I

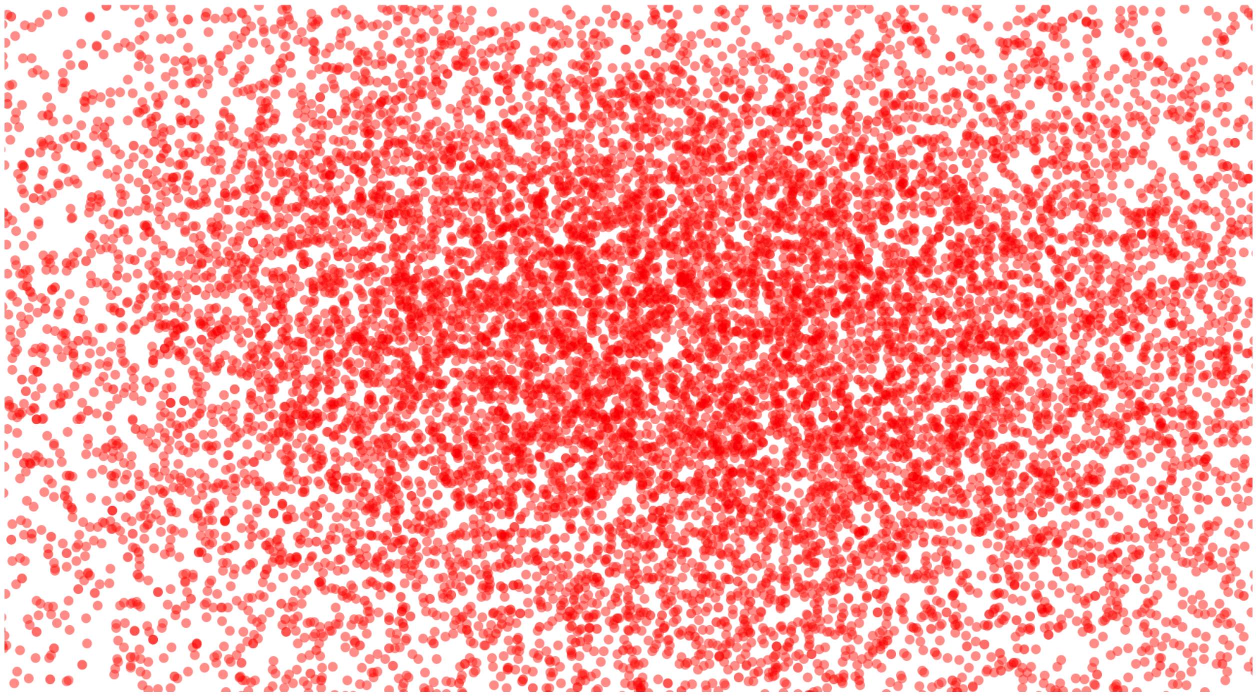


Phase Zero: The study

First I will briefly introduce you to the BCSER grant series of studies. The BCSER grant looks at understanding math reasoning performance with the presence of two perceptual cues, color and spacing (color shown above), while also considering individual differences in executive function. Executive function (EF) is a set of cognitive measures that allow us to complete explicit tasks (like solving math problems). In this study we considered two EF measures – working memory (WM) and inhibition. Additionally, (to make matters more complicated) participants' eye gaze was tracked with an online gaze software WebGazer. And that is the data that we will be using for this visualization project.

I hope to create a meaningful and interactive visualization that makes the most of the data dense research project. I would like to incorporate the EF measures in some capacity, so highlight how individual differences not only effects performance, but also their gaze while completing arithmetic problem solving tasks.

Above is examples of three distinct stimuli, the left image has a *congruent* color cue, the middle image has no perceptual cue (and is what is rendered on the final visualizations of this project), and the right image has an *incongruent* color cue. Concurrency is related to which step of the order of operations is highlighted. If the higher order operator is highlighted (HOO; ie. multiplication and division) then the cue is congruent. If the lower order operator is highlighted (LOO; ie addition and subtraction), then the cue is incongruent.

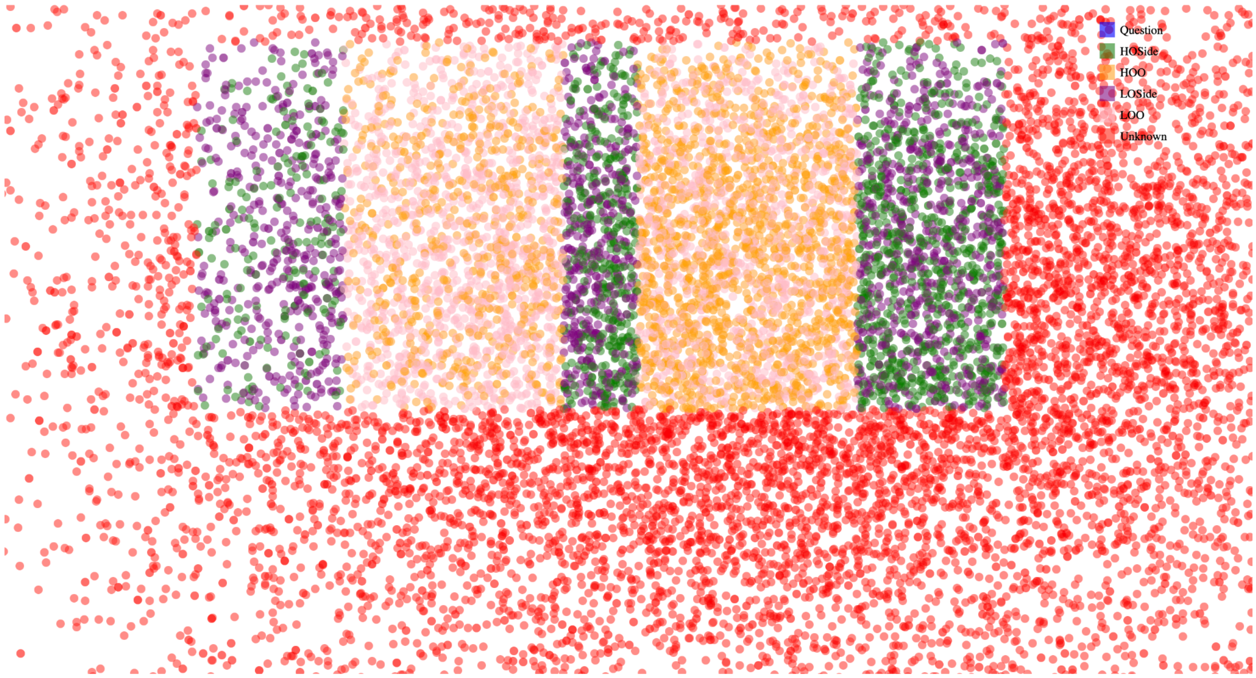


Phase One: Rendering dots

Fortunately I had all the data and it was extremely clean. Each participant has a CSV named after their WPI ID with all of their gaze data coupled with the problem level data. This allowed for a simple starting place: rendering dots on the screen. All I had to do was draw small transparent circles in d3 with their coordinates being the X and Y values provided from WebGazer. It actually was as simple as that. Obviously this visualization tells us absolutely nothing except that their gaze is, as expected, concentrated around the middle of the screen. Additionally this was a participant chosen at random, so it does not give us any idea about the patterns of eye gaze across participants.

The characteristics of this phase were:

- Just getting the data on the screen
- Using assignment one and two to scaffold the simple rendering of dots one after the other onto a screen.



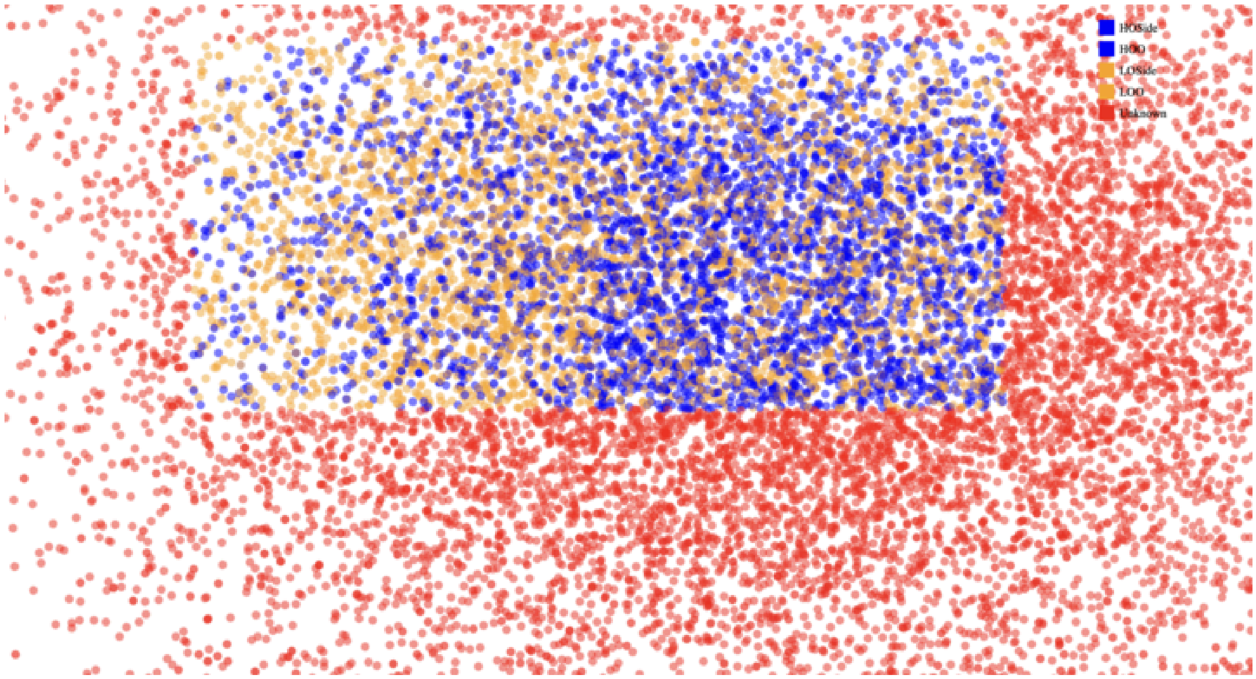
Phase Two: Coloring based on AOI

So we needed more meaning, and like I said this data was SO CLEAN. Clean enough to eat off of if it were a physical object. So clean and preprocessed in fact that there were already areas of interest (AOIs) defined and listed in the CSV I was working with. The defined AOIs are as follows: HOO operator side (the side of the equation with the HOO on it), LOO side, HOO (the actual operator symbol), LOO. Depending on the problem, the HOO side switches, meaning that the colors for HOO and LOO AOIs will be mixed. Still, you can see a bit more information than before. For example you can see that to the left of the pink and orange block there is just less density in general. Additionally, I added a very difficult to read legend in the corner.

These updates required very little technical updates from the code in the just red version. Now if a row has an AOI, then there is added logic to make the color of the dot match the color specified in a custom color map to the AOI. Like I said, the logic for the color was simple enough, and similar to assignment two. I used GPT to generate the code to make the illegible-legend.

Characteristics of this phase were:

- Higher order operator/ Lower order operator symbol are the pink and orange
- Purple and green are the HOO/LOO side.
- Confusing because the LOO and HOO sides are sharing the same space bc this is all the data from one participant.



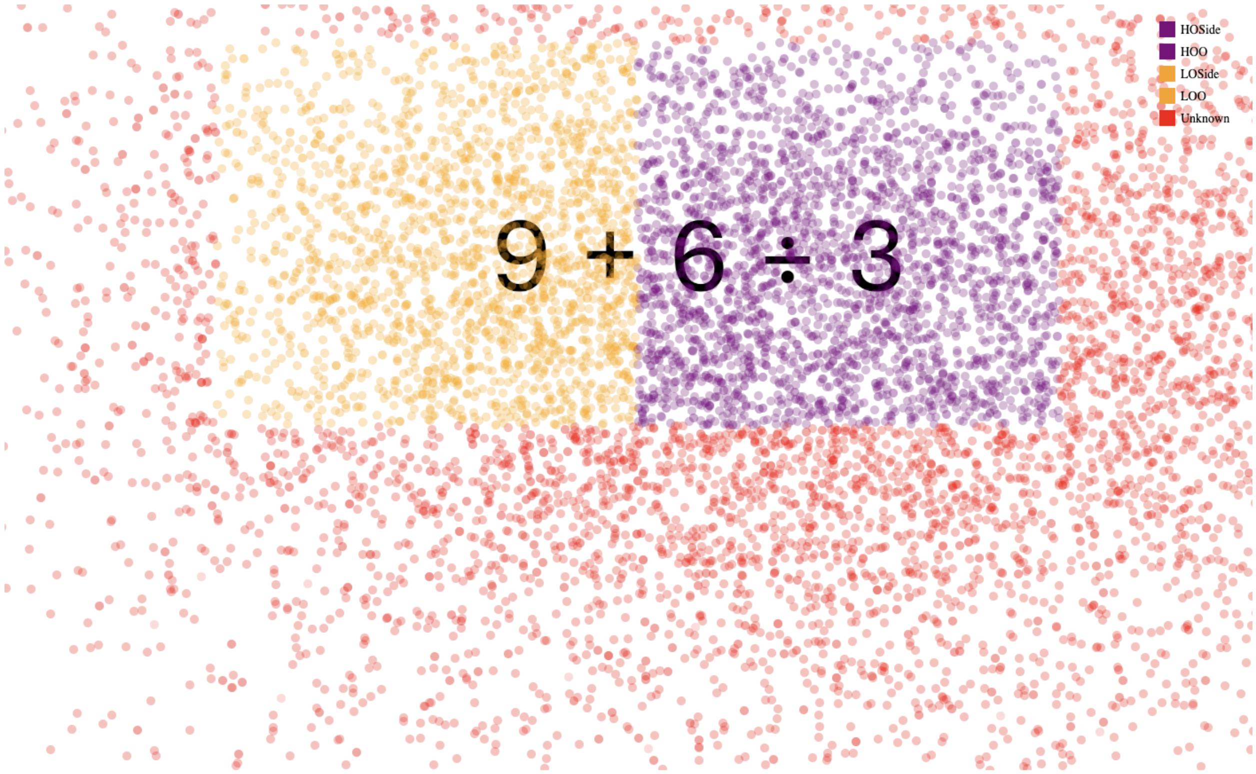
Phase Three: Trying to simplify and make more sense of everything

So phase two was maybe not the most clear or informative with all those mixed colors. So for phase three I tried to simplify things to gain more information from the visualization alone. I pared down the previous version by simply changing the colors to match HOO and HOO side (both blue) and LOO and LOO side (both orange). This visualization in my opinion, is the most descriptive of the dots. YOu can see that there is greater density of blue on the right side, meaning that when the HOO is on the right, this participant is spending more time looking at that HOO side than when it is on the left. (This makes sense on a cognitive level and is in our hypothesis, that because we read left to right we have a harder time solving these problems when the HOO is on the right.)

Still there is not much we get out of this view specifically what were these participants looking at? Why do we care about that blue and orange block in the middle? What do the AOIs even look like? To answer those questions we have to move on to the next!

Characteristics of this phase were:

- Getting rid of the specific AOI for the HOO and LOO sides. From here we can see that there is a higher density of blue dots on the left in general for this participant. That indicates that they are spending more time on problems when the HOO is on the right.
- Still not informative of what this person was gazing at.



Phase Four: Add stimuli, filter data

I got sick and tired of the mixed up color. It was, overall, not helpful and just confusing. So I was quickly able to eliminate half the data by filtering on HOO (Only Right side HOO is shown above). So notably the orange and purple sides of the AOI block are unmixed and have a clean line down the middle to indicate that the HOO is on the right, and LOO on the left. Filtering the data was again, pretty straightforward and provided a path to further filtering down the road.

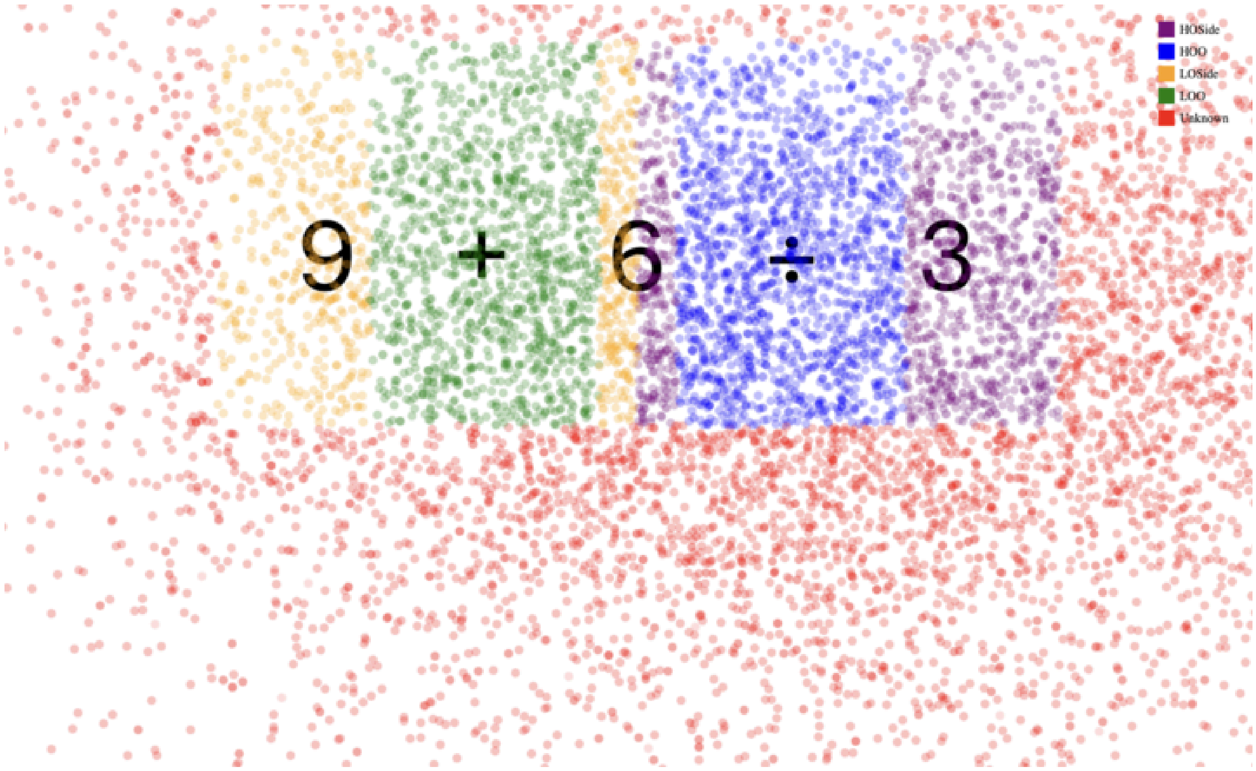
The other great and lasting addition to this phase was the stimuli! While the spacing needed some work, it's a clear step in the right direction. As someone who helped code up this experiment I did not need stimuli to understand the previous visualizations, but I clearly have expert blindness on the topic.

Adding the stimuli was more challenging than I had anticipated. The numbers and symbols are PNGs which are a little tricky to work with in d3. I did my best for this go round by brute force guessing where they land on the plane. Always good to leave room for improvement.

Overall I think this visualization is two steps forward one step back. When comparing it to the previous phase we lose the information that this participant was lingering longer on HOO on the right problems. In contrast we gain the all important information about the stimuli that they are looking at.

Characteristics of this phase were:

- Filtering the data to only include instances where the HOO is on the right makes it a little more readable
- The greatest add was the stimuli to give an idea of what it is like… unfortunately the spacing is still off in this version



Phase Five: Putting it all together…

The final step of the dot plotting of the gaze data was to combine all my ideas into one visualization. While there is a distinct lack of information in this vis (and maybe an even distinct lack of meaning), there are some great tidbits to highlight.

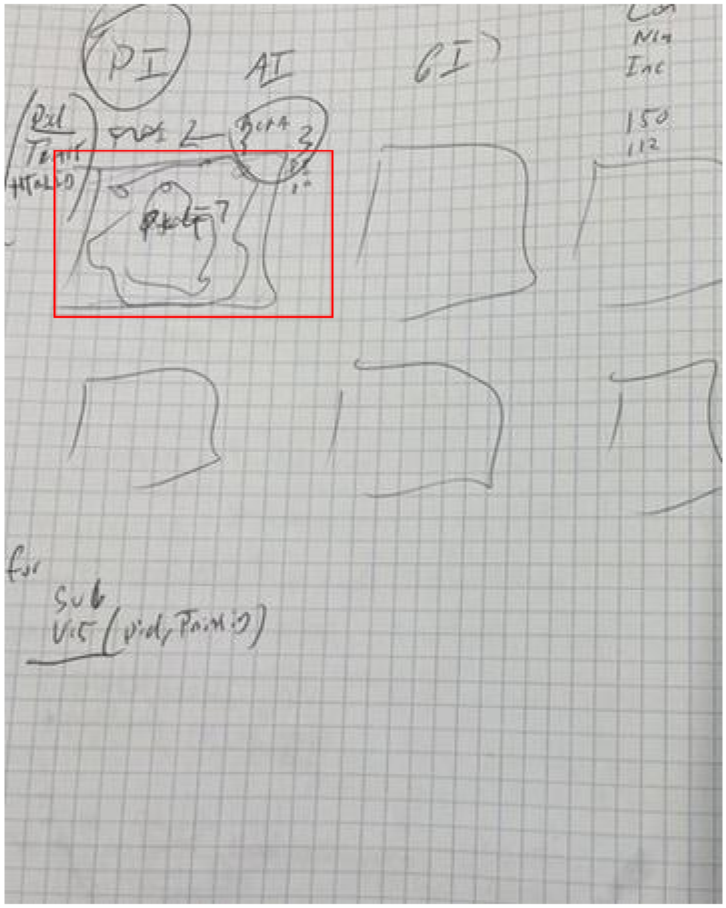
I added back in the distinction between HOO side and HOO symbol. Five color schema makes more sense when we can see the stimuli.

Speaking of stimuli I got my spacing worked out. You can see how the operators fall in the middle of their AOI region, and the six splits HOO/LOO sides.

Finally I kept in the right-side filter. I think overall we can see more clearly without explanation when the filter is present. And as I said before, the filter logic stays consistent throughout the project.

Characteristics of this phase were:

- Added back in distinct colors for the HOO side and symbol
- Fixed the spacing – looks great!
- Kept the right side filter of the data so the color do not mix
- Still only seeing one participant
- Still only seeing the gaze data as transparent dots… not super informative still



Phase Six: Design break

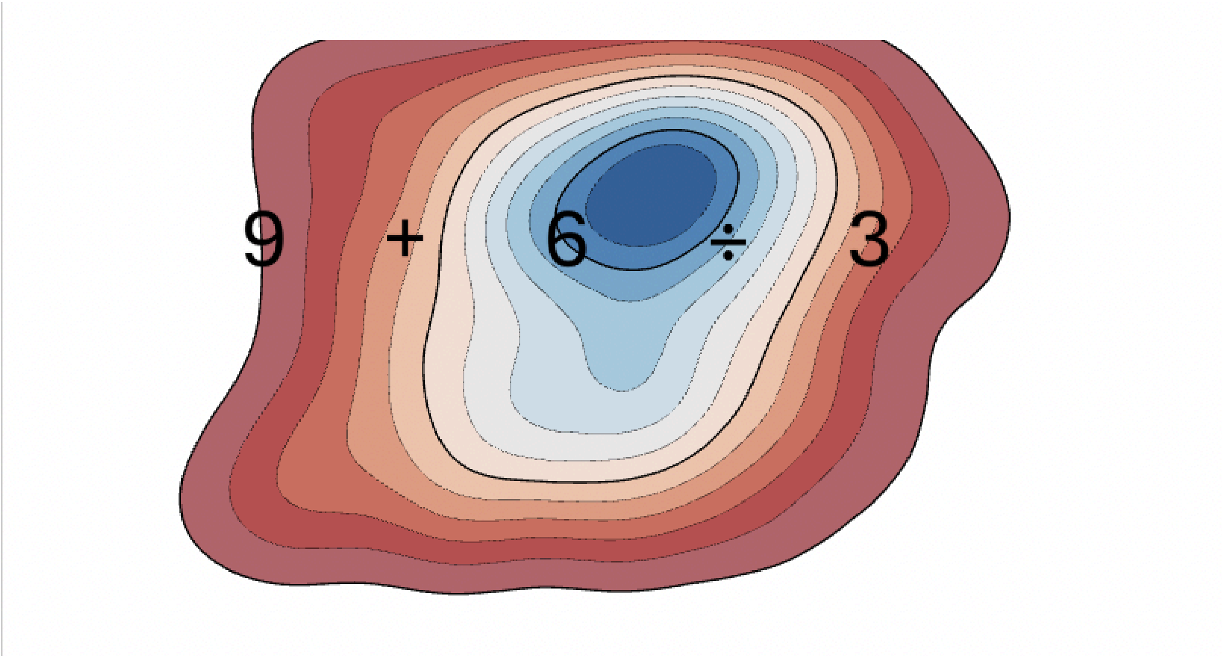
About two weeks into this project I touched base with Lane to see what he thought would be a good final product. We (he) came up with this interactive interface that allows users to filter based on the conditions of the problem, and the EF of the participant. – yes, we are circling back to the cognitive science stuff. Lane wanted me to have a window that contained all the participants pata side by side. He said “ditch the dots, lets go with a density contour map.” I'm paraphrasing of course. He initially wanted the user to select a single participant from a dropdown and/or be able to filter based on the conditions. There would be drop down menus for each condition, and for the EF measures, and a search bar for the WPI ID.

The sketch shows a general layout of the windows and a pane that shows the density contour and stimuli highlighted in red at the top left. From here I get to coding!

Characteristics of this phase were:

- I checked in with Lane, and we decided it would be ideal to create an interface where we could see all the participants side by side
- Each participant would have a pane which shows their gaze data as a density contour map.
- The stimuli would be on top of each pane to give the needed context.

- Finally, there will be filters for each of the different stimuli conditions (HOO location, color and spacing), and for the EF measures (inhibition, and working memory)
- The pane circled in red is a mini mock up of what each pane will look like.



Phase Seven: Density contour (singular)

Transitioning from dots to lines and colored in density maps was more challenging at first that the dots were ever. Lane was a huge help in getting something on the screen that resembled contours, then I was on my own for rendering the color schema. I used GPT to debug, but overall once we had the initial code from the d3 documentation, it was not too bad to get everything rendered so that we could see color and area contours.

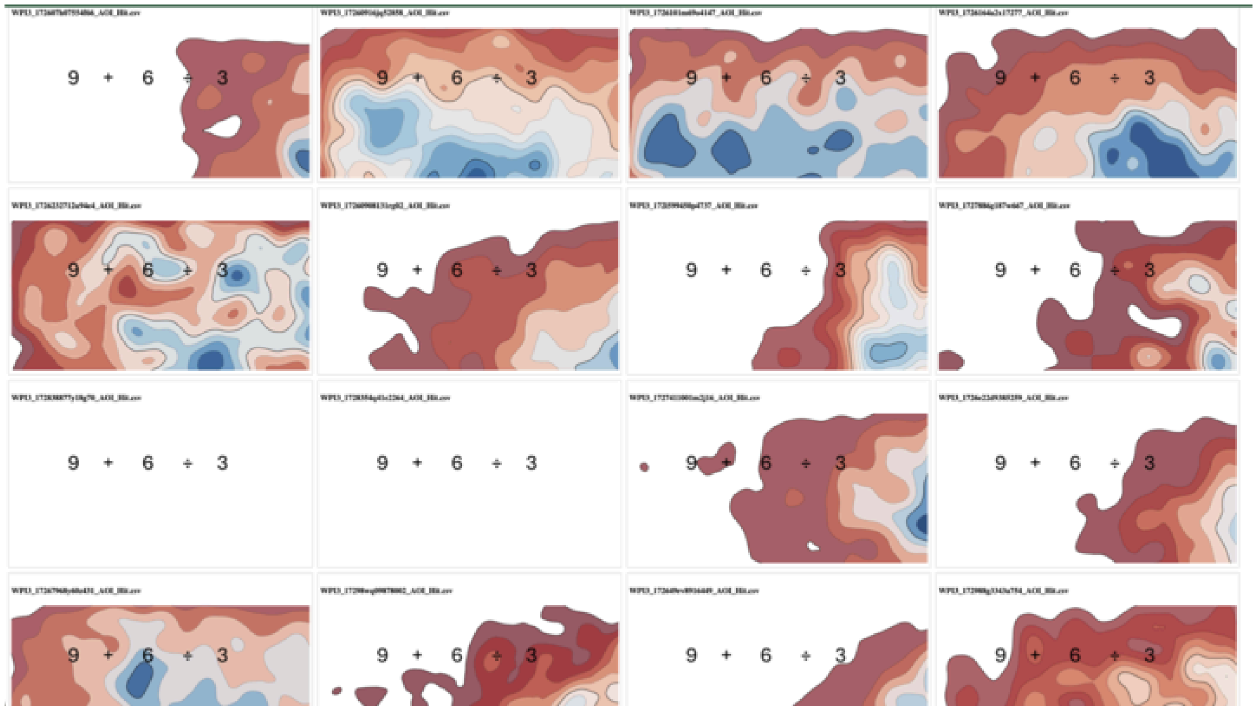
This visualization provides a great deal of information about our singular participant. First, as expected the participant is primarily looking at the HOO (division) and its first operand (6), there are very few data points on the outskirts which are represented in the white area outside the red. The perimeter of the visualization is a good sanity check in this case. In eye tracking analysis we call it the eye-mind assumption. That is, the mind will think about what it is looking at for the most part. We know this participant answered the question (we removed exceptions to this rule), so presumably they were thinking about it, hence we expect their gaze to be on the order of operations problem presented.

Before developing the interface with all 150 participants, I wanted to generate one to use the code as a starting point before looping through many. What Lane pointed out was cool about this process is the code to create the contours is similar to our first assignment with creating lines – just with more behind the scenes math!

Characteristics of this phase were:

- Now we know we can work with the data and manipulate it with relative ease – so lets make the visualization meaningful

- Lane helped by giving me a good idea of what the heat map should look like and finding resources on line to start the code (d3 documentation, chat GPT)
- Before rendering all the participants, I rendered just the one so I had the base code I could loop through.
- The red represents low density eye gaze and the blue is high density eye gaze. You can see in the example above that there is a fixation on the space around the HOO (in this case ➗)



Phase eight: One to many!

To render more than one visualization per screen I once again had to turn to GPT. I had not used containers in HTML before, nor had I used a node.js server. The containers made it easy to add as many visualizations on a single webpage. You can dynamically change the number that appear in a row by zooming in or out of the page. I had to use the node.js server because all 150 of my data files were in a separate folder than my html and .js files. With the help of GPT that was also easy to navigate and simple enough to read and create a for loop in my code. All 150 participants are represented at this point!

The greatest issue at this stage was scale. Because I changed the canvas size when I wanted to fit more than one participant in a row, it messed up the scale– hence why many of the density contours are off to the side. Additionally, I removed the HOO filter for this test run so the maps are a bit less smooth that the one from the single run.

Overall this was a huge win! Getting more than one vis on the page meant that the end was in sight. I had some backend work to finish up and a quick polish and we would be good to go!

Characteristics of this phase were:

- Added the containers in the index html (used chat GPT to help)
- Created a node.js server for to read all the files from a folder (used GPT to help)
- All 150 participants have a pane!
- There is an issue with the scale. Some windows were smaller than others in the experiment or they just did not fit the size of the containers.
- Data was not filtered by HOO side
- A huge step in the right direction, the end is in sight!

ace_data_levels

	age	handedness	bid	BOXED.rcs.overall	FLANKER.rcs.overall	BRT.rt_mean.correct	BACKW	inhibition	wm
1	18	RIGHT	wpi31726232712u94e4.session0	0.954907161803714	2.00215616818112	296.833333333333	6	Average	Low
2	19	RIGHT	wpi3172626625ps4717.session0	0.997474747474747	1.68158068584469	318.8	6	Low	Low
3	19	LEFT	wpi31726e22d9385259.session0	1.22299992473847	2.02055697092155	277.769230769231	8	Average	Average
4	20	RIGHT	wpi317d26k247126090.session0	1.10365398956003	1.84270656740621	314.5	8	Average	Average
5	18	RIGHT	wpi31728q4153194o63.session0	1.02874145353254	2.1913190054783	282.793103448276	9	Average	High
6	21	RIGHT	wpi317267968y60z431.session0	1.00737397751541	1.76852033798389	300.066666666667	7	Average	Average
7	20	RIGHT	wpi31726101m69o4147.session0	1.2496638562412	2.2766078542971	280.6	7	Average	Average
9	21	RIGHT	wpi3pf1726083661580.session0	1.4167172637118	2.32516544446432	391.1	7	Average	Average
10	36	RIGHT	wpi3j172v6090848366.session0	1.30508626083847	2.03466465712133	263.833333333333	7	Average	Average
11	23	RIGHT	wpi317260908131rg02.session0	1.0966326925558	2.29399889888053	296.709677419355	6	Average	Low
12	19	RIGHT	wpi3173047339k17v58.session0	1.25464045098309	2.36707938820102	272.035714285714	8	High	Average
13	21	RIGHT	wpi317r260913860u89.session0	1.18951246509937	2.25375626043406	334.586206896552	7	Average	Average
14	22	RIGHT	wpi3172z609y1394122.session0	1.30434782608696	2.10526315789474	280.5	6	Average	Low
15	19	RIGHT	wpi31726lg255892316.session0	1.06107550613302	2.58243941199841	243.259259259259	7	High	Average

Phase Nine: Backend work

Getting all the filters to work and adding the logic to match participants based on their EF data was the next step in the process. Adding in all the stimuli condition filters was easy. I already had the logic for the HOO side, and adding logic on the perceptual cue conditions was just a matter of adding to a contains statement.

Filtering for the executive function was more challenging. I had to do some data management and add the rightmost columns to the EF data frame above. I picked the thresholds of high and low, I took the mean and standard deviation and placed participants who were one SD about the mean or below it in the high and low categories respectively. I then had to match on participant ID to filter out the non-matching conditions.

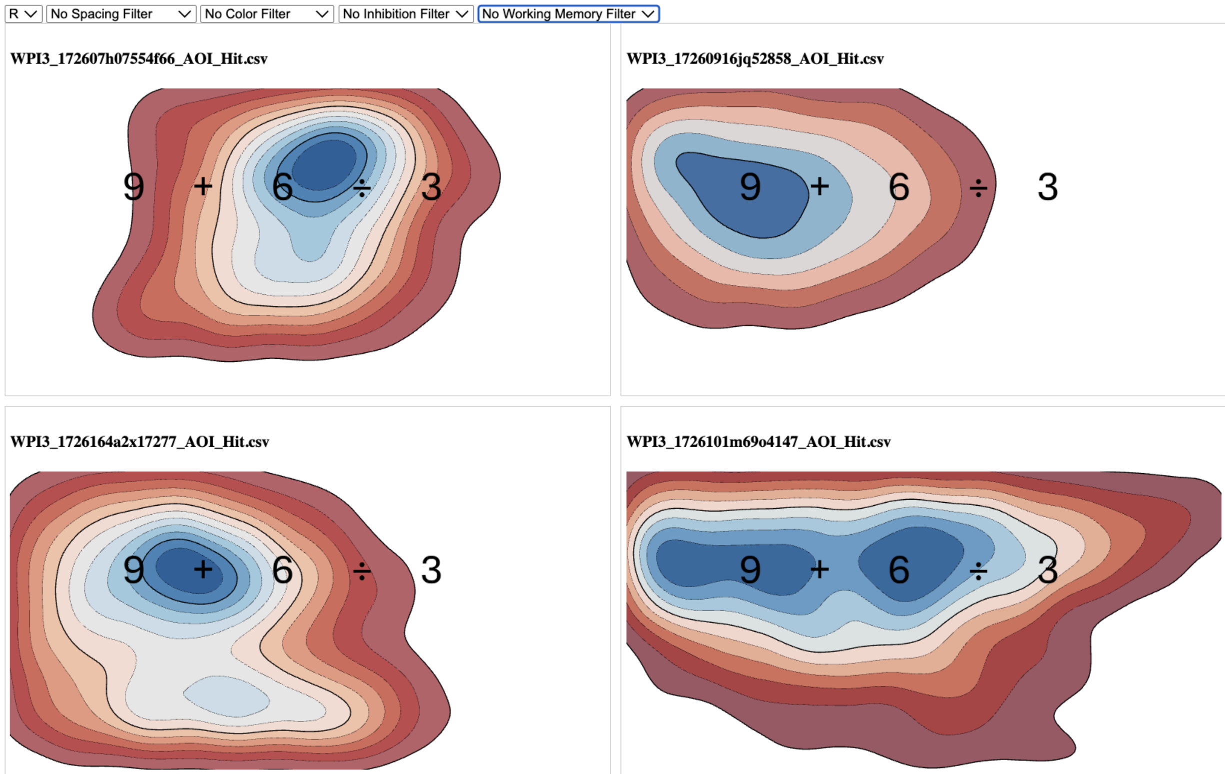
Finally, I needed to add the logic for the conditions. I created 4 variables that when changed they updated and ran a filter function. The filter function did different things depending on the dropdown changed. In the event that the HOO was changed it re-rendered the stimuli as well as the data. If the stimuli filters were changed it just updated the data. If the EF filter was changed then it cross references the gaze data to the EF data and keeps those that match the new filter.

The biggest back end change I made was structural changes to the code so that I was not reading the same CSVs over and over. While I am sure there are more efficiencies i could add, the rendering process is far shorter than it was before.

Characteristics of this phase were:

- Filtering the conditions was easy, just required adding additional terms to the already established filter logic from before
- Filtering based on EF had more steps:
    - Get the EF data which is not attached to the gaze data
    - Find the average of the EF measures and the SD

    - Add the two rightmost columns for working memory and inhibition (in red)
    - Reading the csv and cross referencing it in the rest of the logic
- Making sure the logic for filtering was present with variables that could be preset
- Making the read and filter process more efficient so the render takes less time



Phase Ten: The final product

Now back to the front end fun! While I am not a front end girl at heart, there is something satisfying about working with js and getting to see your code in real space. I quickly scaled the data properly so it didn't look like complete nonsense. The biggest step I needed to take at the very end was adding the physical drop downs on the interface. I had to learn how to create buttons in HTML – a new skill. Super straightforward as it turns out. Once the drop downs were up and running I was off to the races!

I stress tested all the dropdowns, making sure the correct participants were rendered when needed, and that nothing crashed or stopped working. Then I told literally everyone I knew about it :)

I feel very proud of this work and think there is potential to expand it more if the spirit (or my advisor or Lane) moves me.

- Properly scale the data so it fits in the images
- Add the drop downs in the html
- Render the data and fail test the filters
- celebrate!