The N.E.R.D Spark Data Tracker

A data-driven approach to getting notes in speakers

Gabriel Koo

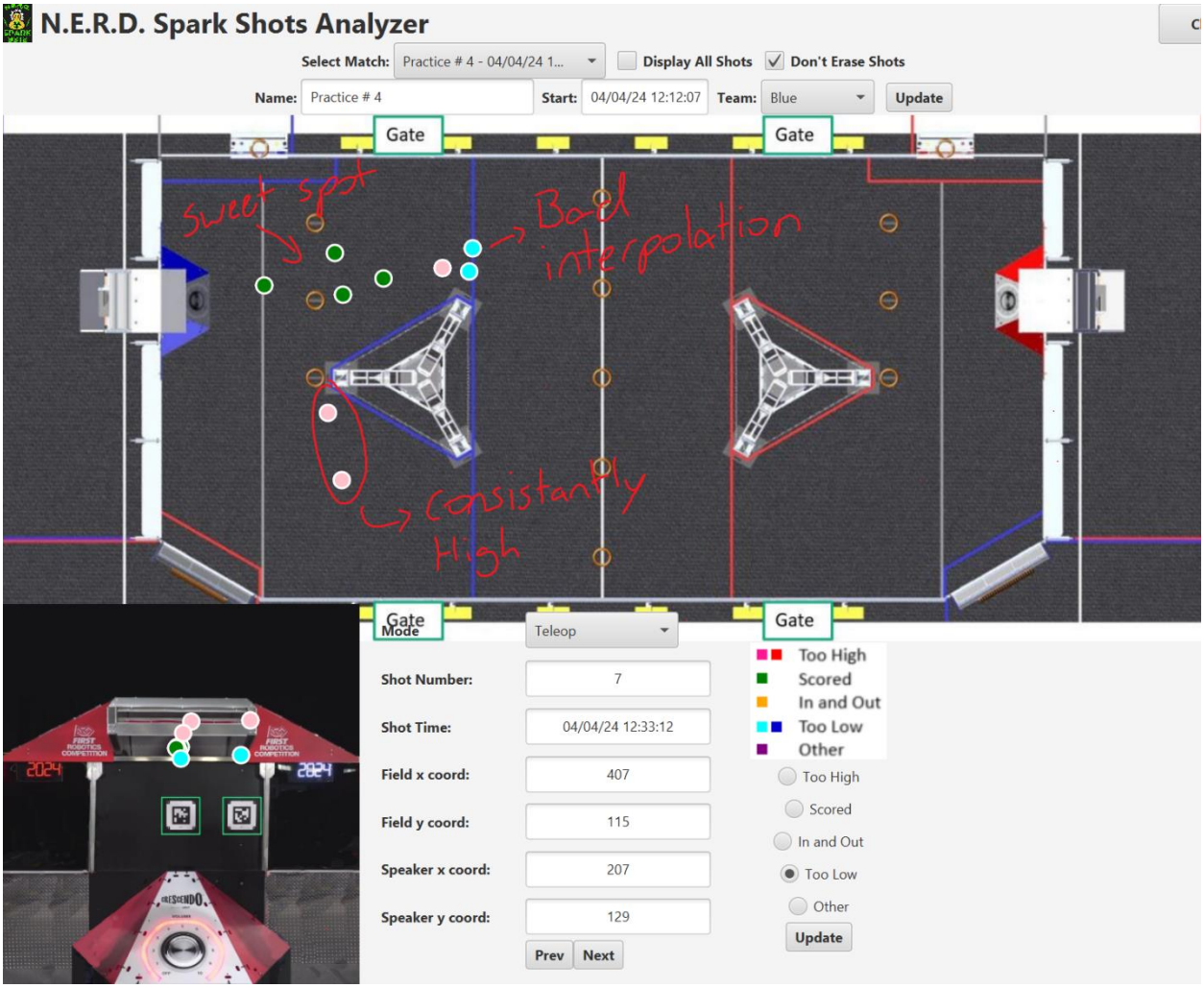
4/10/24

1 TABLE OF CONTENTS

2 INTRODUCTION

One of the many challenges our team faced has been tuning our auto-aim system to new fields in real-time. The build and design team have put together a robot capable of shooting ridiculously long shots, and the programming team has devised a way to lock onto the speaker. And yet, due to the inevitable "shock induced adjustments" that occur during regular matches, it became hard to keep track of the calibration tweaks necessary to keep our shots consistent.

We had a concept for a data-tracking system where records of where shots are taken from are mapped with results of those shots, giving a heat map of our robot's accuracy. With that information, we can look for patterns where shots were either consistently scoring goals or going high/low. We then provide this feedback to the programming team to adjust the robot's calibration constants, and to the drivers to assist them in decisions on where they should be shooting from on the field.



Our first Practice match, adjusting to the new field

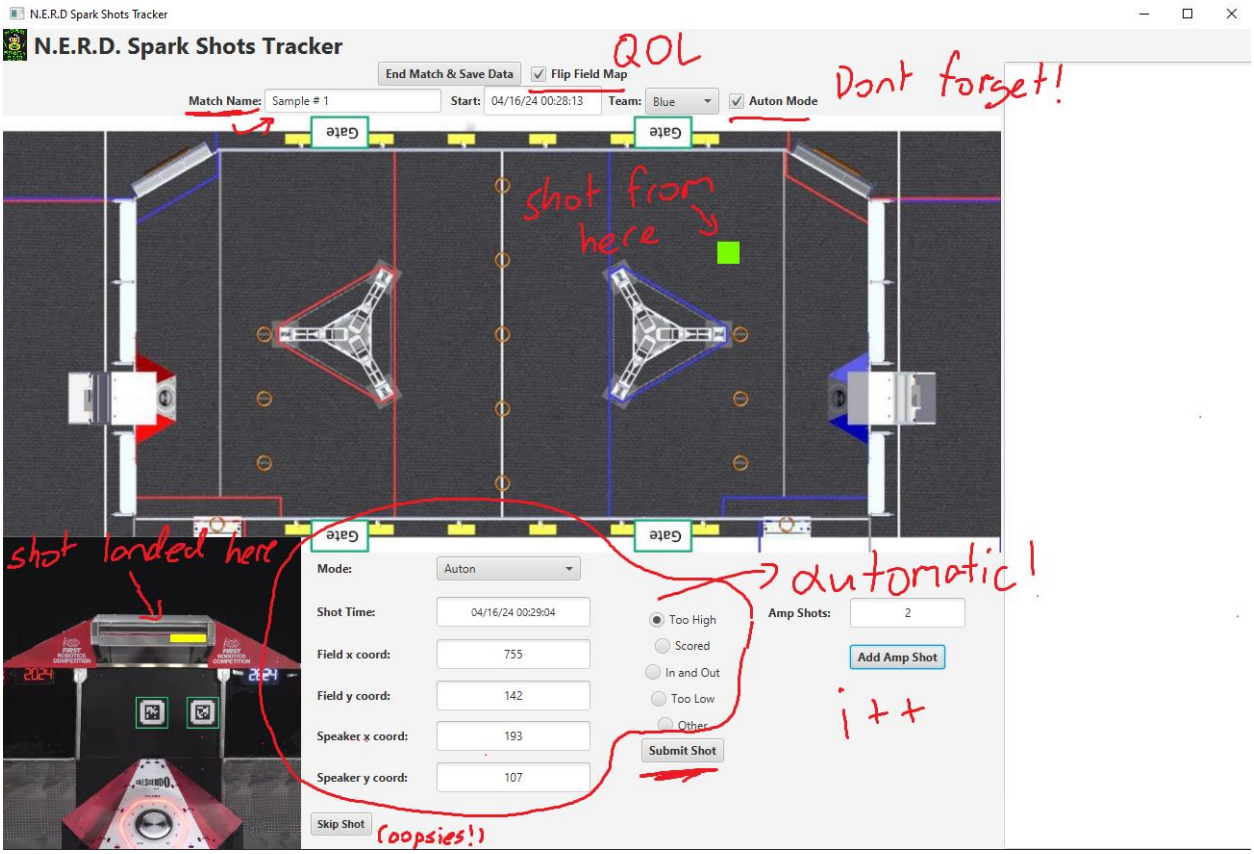
Having figured out the necessity for such a team during Week 1 Quals, we immediately put together a team to create a working system by our next competition: Week 5.

We chose to write a Java application for this project. Our data science team members have a variety of Apple MacBooks and Windows laptops, so it made sense to us to have a single codebase that could work for everyone. The Java promise “Write Once Run Anywhere” turned out to be a nightmare of smoke and mirrors (thanks to JavaFX), but we still managed to get it working eventually.

Figure 1: Data Science team in the Data Closet

3 DATA ENTRY

We called our data entry component the “Shots Tracker.” In addition to capturing some basic information about the match itself (such as which round, start time, etc.), we also added some quality-of-life features like flipping the field map to make it easier for our data team to record the match no matter which side of the field they were sitting on.

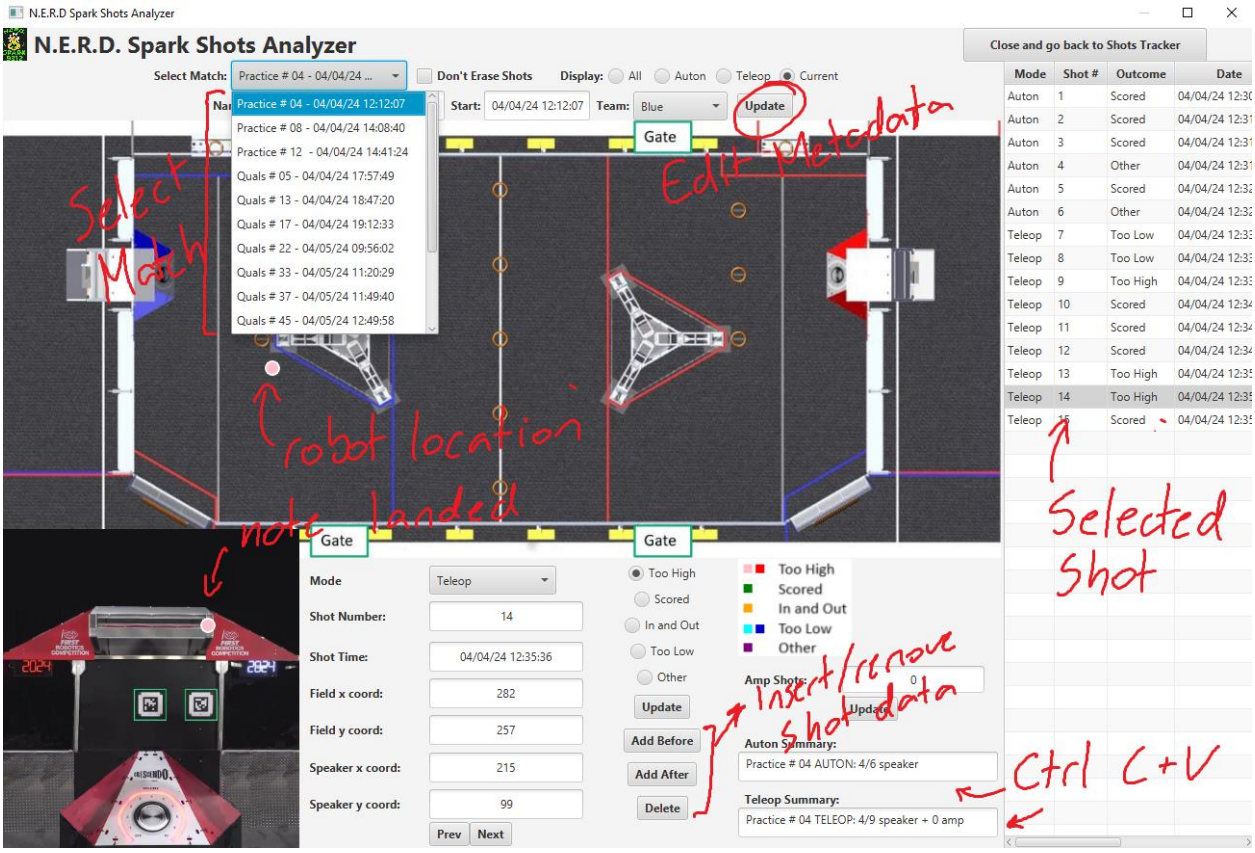


We tried to capture as much data as possible so that we could understand exactly what was working or not working at any given time. Things like distinguishing between Auton mode from TeleOp shots were important so that we could direct relevant information to the correct team members. However, we also needed to avoid making the app too slow to use. We figured out that, by using an intuitive point-and-click system, we could track where the note was shot from on the field and where it ended up at the speaker, and then the shot time, field x/y coordinates, and the shot outcome will be automatically populated (and can be manually adjusted if needed).

We also felt like a simple hit or miss result was not enough to make the correct changes to the robot, so we aimed to discriminate missed shots based on where they went. Too high and too low were obviously important to discern, but we also found it useful to track when they missed horizontally. There were also cases when the shots were on target, but went in and out of the goal due to the shooter’s velocity. We even added a “Skip Shot” functionality to allow team members to put in a blank placeholder for ambiguous shots, so that they can be revisited later during data analysis. Finally, we provided a simple counter field to track the amp shots as well, just for our own record-keeping.

4 DATA ANALYSIS

For quick data analysis, we created a companion application called the “Shots Analyzer” to visualize and update data collected from previously recorded matches. After each match, we used this to consolidate the data we each collected into a final report before showing the team. This is done shot-by-shot the navigation buttons under the shot info, and corrected using the add/update/delete shot buttons.

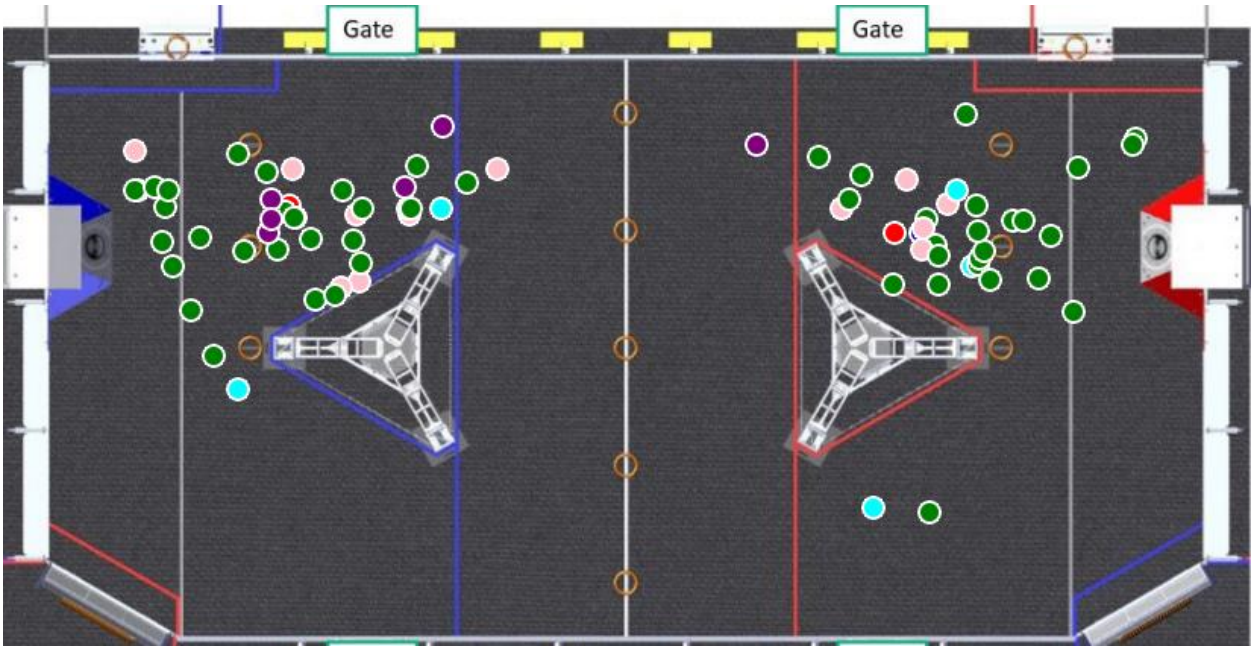


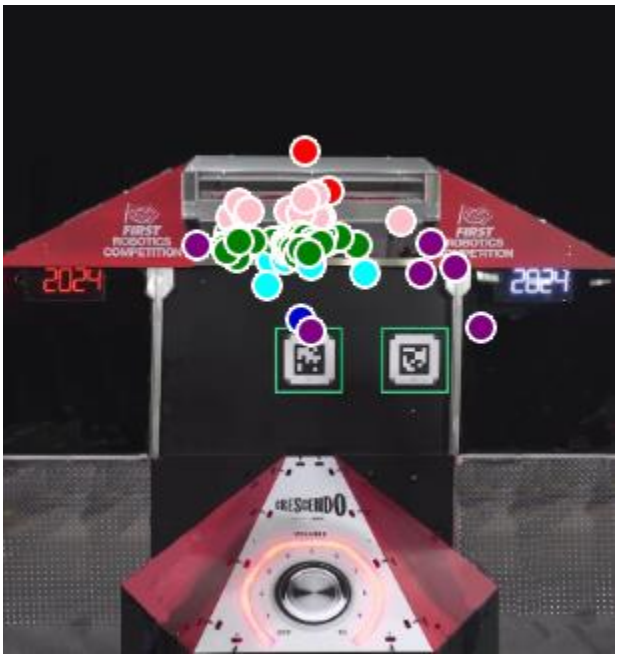
In addition to maintenance, the primary function of Shots Analyzer is to provide a convenient, digestible UI for visualizing shot selection and accuracy. Shots are directly displayed by match or by shot, while the “Don’t Erase Shots” option allows overlaying any combination of shots and matches to make the data as useful as possible. However, between matches, simply selecting between the Auton and TeleOp shots was usually enough, so we added buttons to quickly do this on the fly.

When selected, shots are displayed as both the positions on the field from where the shot was taken, as well as positions on the speaker where we estimated the shot went. Each shot is color-coded to indicate the shot outcome (i.e. Scored, Too High, Too Low, In-and-Out, and Other), and each shot circle also has a ring around it to indicate if it was made during Auton (white ring) or TeleOp (neon green ring). Since it was difficult to differentiate between particular shots due to clustering clustered, we also added tooltips that would indicate the shot’s number when hovered over. Finally, we included an automatically-generated performance summary (split by Auton/TeleOp), which we would use to directly compare performance at a high-level between rounds. Along with the shot reports, this was part of the final report shared with the team.

5 RESULTS

With the benefit of a full competition's worth of data, we can go back and see what went right and what went wrong. Unfortunately, while using the Shots Analyzer at this point is possible, it is not as helpful as we'd like due to the mixed configurations and robot wear-and-tear. However, generalized trends in shot accuracy by distance can be nicely visualized, such as our tendency to do better at closer-range.

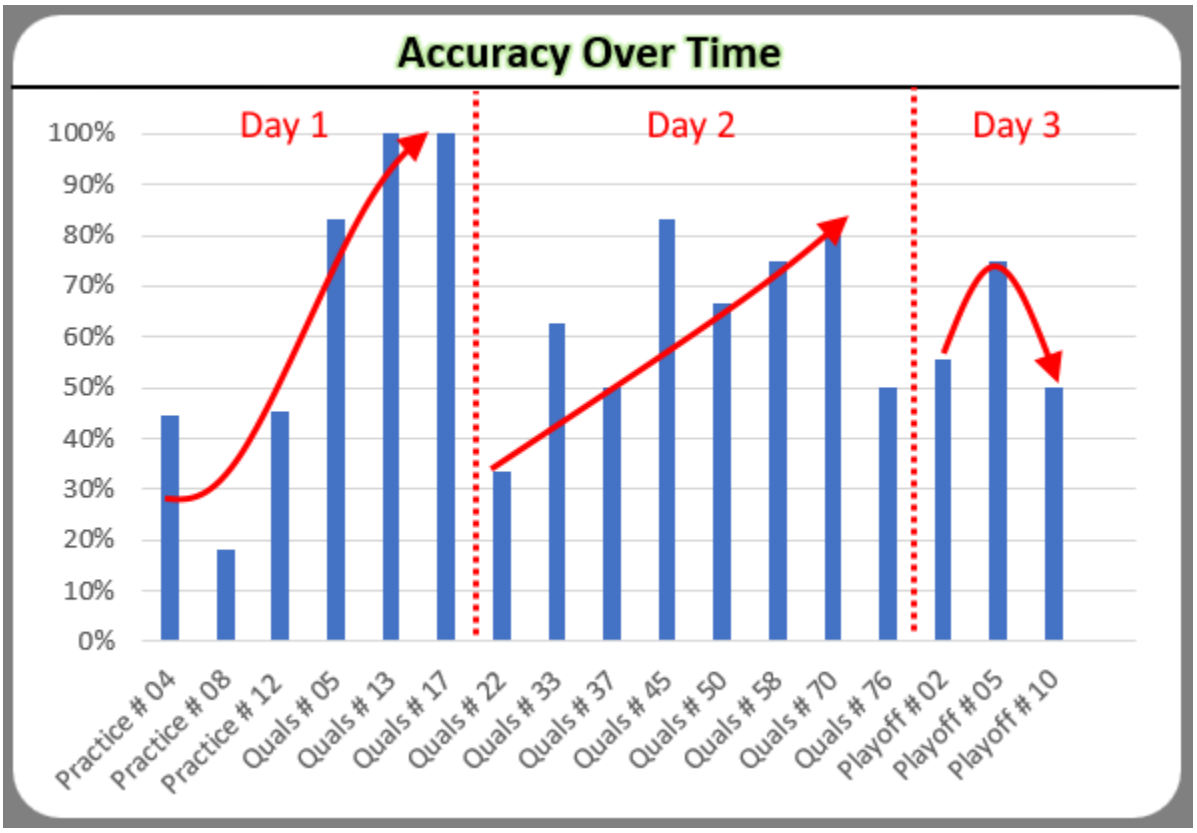




A composite of all our Michigan State Championship matches

Thankfully, we've thought about this beforehand, and we made sure to store the data in simple, self-documenting JSON. This was both our way of handling the notoriously poor internet connection found at FRC events, and to keep the data refactorable as we modified the data tracked between matches.

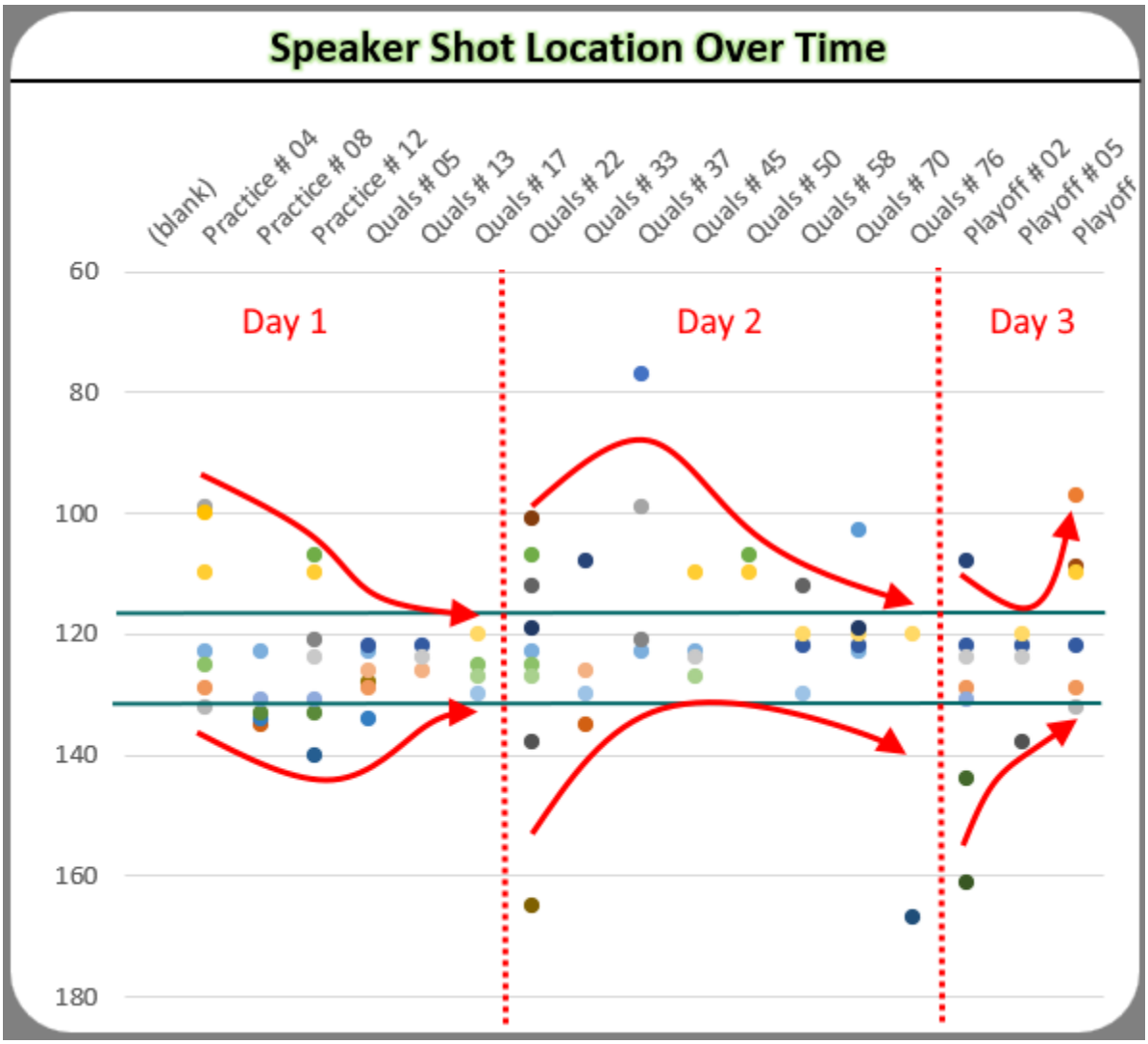
We opted to import this data into Excel, and see what it looked like graphed-out. The beauty of data is in how it is able to speak for itself when used correctly.



Shot accuracy over the course of the Michigan State Championship competition

The configuration tweaks definitely seem to be effective in creating upward-trends, however an odd pattern emerges in the data: we start every day at a lower point than the previous day, by a lot. While I could go on about my theories of nocturnal sabotaging ninjas, knowing that such a devastating pattern exists is an important first-step to diagnosing the problem.

But standalone accuracy doesn't tell the whole story; we also want to know *where* we've been missing. To capture this, I plotted the shots' landing y-coordinates over time. While this does miss some of the data's nuances (i.e. missing left/right), it's an effective in showing our team's primary trends. Note that Y-axis is measured in pixels, and the Teal lines represent what would have gone in the speaker.



Shot landing location over the course of the Michigan State Championship competition

The same pattern appears as we saw in the previous graph, now with more insight into the issue: it appears we start each match with a much larger shot spread. Again, this alone is not enough to definitively identify a culprit, but it is a start.

We can also see that, after losing our 100% accuracy from day 1, we seem to be tampering with the calibration again to get it back. This ultimately leads to further issues, and our overcompensation in the playoffs could have been our downfall.

6 CONCLUSION

As a relatively new team, Nerd Spark has a relatively open, flexible structure that actively encourages innovation. The Data Science team owes its existence to this openness, and is the reason that our team was able to successfully put together and set up the N.E.R.D Spark Data Tracker for active use, all between our Week 1 and Week 5 Qualifiers.

We've also been lucky enough to have a consistent stream of feedback and ideas from our strategy, programming, and drive teams, which has allowed us to dynamically build the application as new requirements emerged.

Looking forward to Worlds and onwards, we've learned a lot about our own team and the way we work. We've shown the power of making decisions based on numbers, instead of debating over how different people feel about our performance. By leveraging quantitative trends, we will continue to incorporate data into our design and strategy, in addition to whatever new applications we can come up with.

The one thing we lacked this season was time. Next year, our plan is to start sooner and faster, building on top of our current app to get where we need to be earlier. Now that we know what we're doing, we'll be sure to make the most of this.