Ctrl-Z Electronic Scouting System

2015-2016 Season

Jayasurya Sridharan

Scouting Captain

Contents

Introduction

The Importance of Scouting

Scouting in FRC is the act of evaluating other teams' performance through collecting data. Scouting can take many forms—it can be as simple as jotting down a few observations on teams as matches go on, or a fully electronic data collection and analysis system. However, regardless of how complex a team's scouting system is, it has been said time and time again that scouting is one of the most important things a team can do at competition. Unfortunately, scouting is often overlooked by many FRC teams because they do not see the need to spend such time and effort collecting data that they may never use. On the other hand, some teams aspire to scout but are unable to due to lack of resources. I hope this paper can serve as a guide and as an inspiration to those trying to create their own system.

Scouting is a very beautiful thing, as it gives teams the ability to look into their opponents' strengths and weaknesses, allowing them and their alliance members to construct a strategy—even before alliance selections. Often, even a couple more wins can give teams a towering advantage in the rankings. When it comes to alliance selections, scouting allows teams to find teams that will work best with their strategy and brings to light under-ranked teams. With the right strategy, even a team without the best robot can attain a notable advantage over their competitors.

History of the Ctrl-Z Scouting Systems

Before getting to the implementation of this year's system, let's take a look at how the Ctrl-Z scouting systems have evolved over the years.

2012/2013

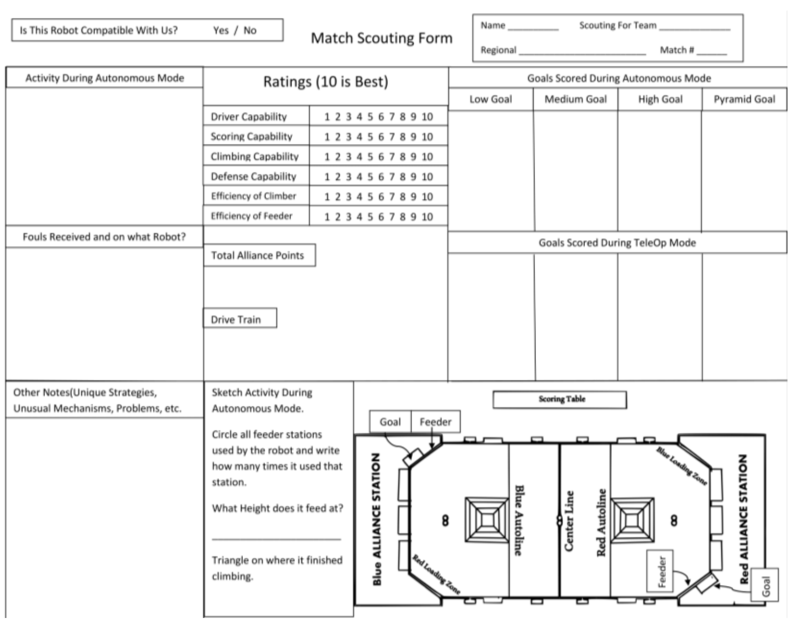


Figure 1.2.1 – A Ctrl-Z scouting sheet in the 2012-2013 season, Ultimate Ascent

2013 was the first year Ctrl-Z had a focused effort on scouting. In this first year, resources were limited so we resorted to what the majority of new teams do – paper scouting *(Figure 1.2.1)*. Logistically, however, it was a lot of work. We printed over 500 copies of the scouting sheet – one per robot for each match. Six scouters would each scout one robot per match, and when the match was completed, the six scouting sheets were passed onto the scouting captain who keyed all the data into a spreadsheet. On Friday night, we sorted all 500+ sheets by team and sifted through the sheets, while being guided by numbers produced by the spreadsheet. By the end of the competition, we had two 4-inch binders full of paper!

2013/2014

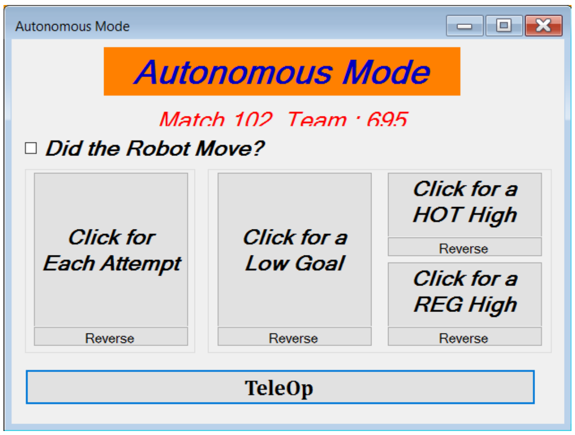


Figure 1.2.2 - One screen from the Ctrl-Z scouting application built for the 2013-2014 season, Aerial Assist.

In 2014, Ctrl-Z transitioned to an electronic scouting system. During competition, 6 laptops would be running a VB.NET application *(Figure 1.2.2)* at all times, while 6 others were at an outlet charging. When laptops began to run out of charge, they would be replaces by a laptop in the charging station. As users supplied the application with data, it was written to a local Microsoft Access database. At the end of the day, all 12 databases were merged together into a master database. We then replaced the local fragments on all of the laptops with the new master database. Since the data was in database format, it could, unlike a spreadsheet, be easily queried with SQL to produce a table of useful aggregated data. This system worked exceptionally well for collecting data, but we were not able to effectively use the data because we preprogrammed all of the queries. Due to this, it was not easy to get a specific set of data on the spot without spending a considerable amount of time writing a new query. This made discussions about the data tedious and cumbersome.

2014/2015

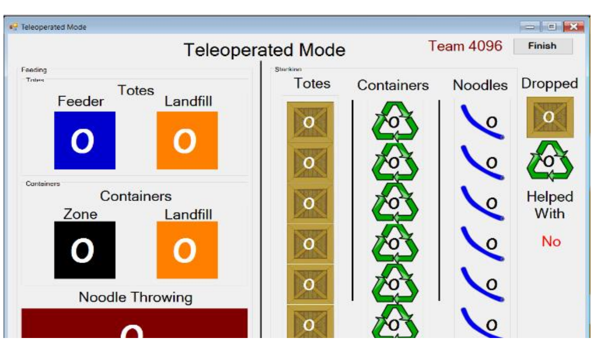


Figure 1.2.3.1 - One screen from the Ctrl-Z scouting application built for the 2014-2015 season, Recycle Rush.

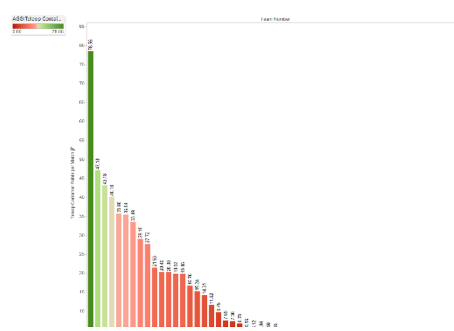


Figure 1.2.3.2 - An example of a graph produced by Tableau from the data provided by the application.

Due to the difficulties faced during Aerial Assist with analyzing the data in different ways on the spot, we moved to Tableau to do our analysis *(Figure 1.2.3.2)*. Tableau is a program that allows for quick and easy creations of sometimes complex visualizations. It became much easier to look at the data and create new reports, which allowed us to come to conclusions quicker and easier. The data collection remained essentially the same *(Figure 1.2.3.1)*.

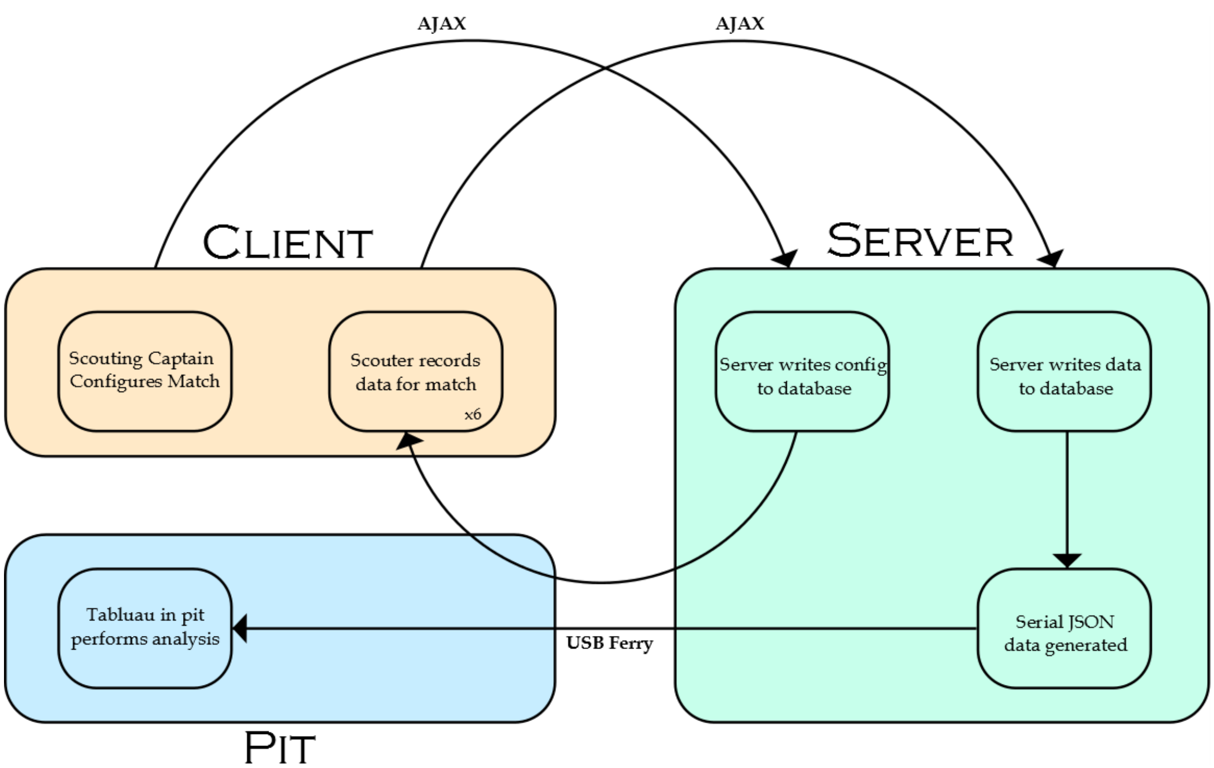
Implementation of the 2015/2016 System

Overview

One of the goals of scouting, as mentioned in Introduction, is the ability to strategize effectively with alliance members before each match – qualification or elimination. In the past, since the data fragments on each laptop were only compiled at the end of the day, the data could not be used during qualification matches to strategize. This put us at a significant disadvantage, as we were missing an integral function of scouting. The solution was to network the computers together and send data to a central database, from which data can be sent to the pit for strategizing.

Technically, the system is essentially a website (written using HTML, CSS, and JavaScript) served over a local wired network, with one laptop acting as a server. The server hosts the backend logic, written in PHP, which responds to requests sent by the client through AJAX. The server maintains the MySQL database, which holds all the data collected during the competition.

Before the match begins, the order of the defenses is sent to the server through a configuration page. Then, data collection for that match is opened up and data can be collected. At the end of the match, that data is sent to the server and translated to JSON, which is then sent to the pit.



Data Collection

The application

The application was designed to allow the user to quickly and efficiently collect data on their robot, while not holding their focus from watching the match for too long. The application features four screens, each representing a type of action their robot can do.

Feed

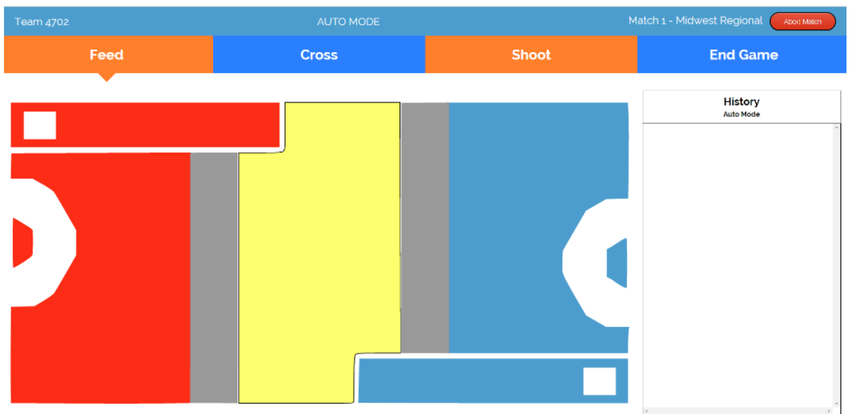


Figure 2.2.2.1 – The screen a user would see when recording a Feed event.

The "Feed" page is very simple. When a robot picks up a ball, the user clicks the area from which the robot picked it up. The white squares in the Secret Passages indicate the human player dropping the ball into the robot.

Cross

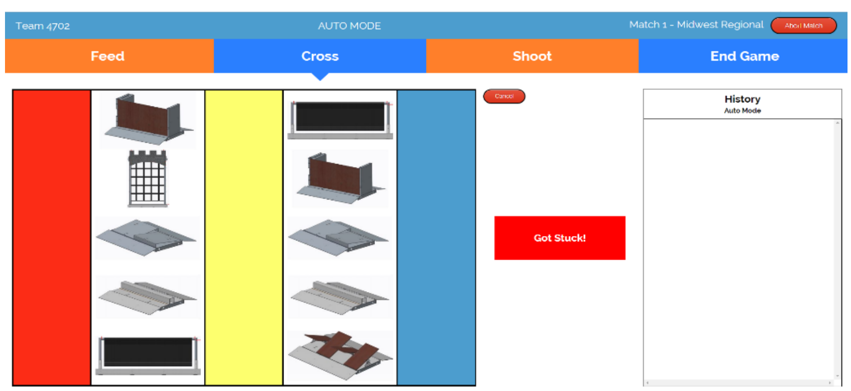


Figure 2.2.2.2 – The screen a user would see when recording a Cross event.

The "Cross" page requires three clicks of the user. The user first clicks where the robot started, then the defense it attempted, and lastly, where the robot ended. If it got stuck in the defense for more than ten seconds, the user is to click "Got Stuck!" With these three clicks, we know whether it made it through successfully, tried and backed out, or got stuck entirely. Since each match is different in the configuration of defenses, each match must be configured before data collection can start. That data is sent to the server and then propagated to the rest of the clients.

Shoot

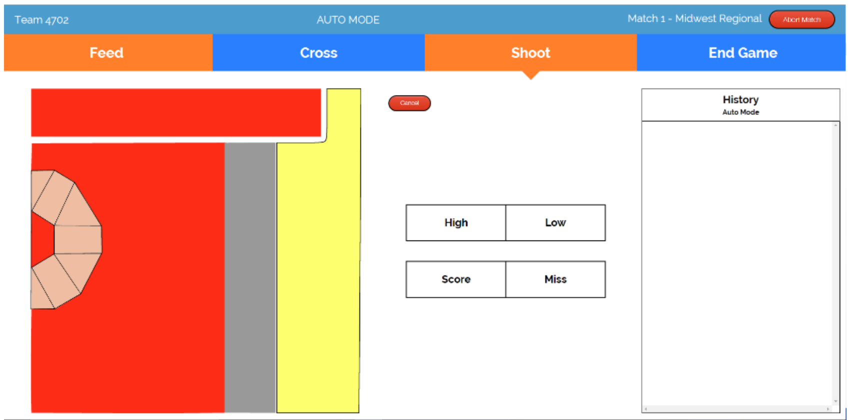


Figure 2.2.2.3 – The screen a user would see when recording a Shoot event.

The "Shoot" page also requires 3 clicks of the user. The user first clicks where the robot was positioned at the time of the shot, then whether it was intended to be a high or low goal, and finally, whether or not the shot was successful. The three small data points, over the course of the tournament, allow us to generate a heat map of where this robot is most comfortable shooting from.

End Game

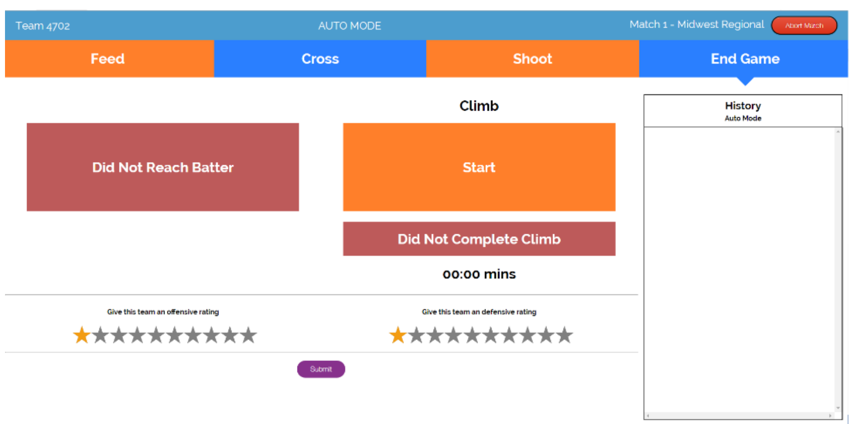


Figure 2.2.2.4 – The screen a user would see when recording the End Game actions

The "End Game" page is where the user will end up at the end of the game. It holds the data for whether or not the robot reached the batter and whether or not it climbed – and if so, in what amount of time. Once the match is over, the user is to give the robot a subjective offensive and defensive rating. We found that these subjective ratings were not very useful, as we have a wide variety of members scouting – rookies to longtime veterans. As a result, the data is skewed depending on who is scouting, as well as how long it has been since the competition started – people tend to rate more harshly after seeing the best robots. After filling out this last screen, the user submits and moves on to the next match.

The network

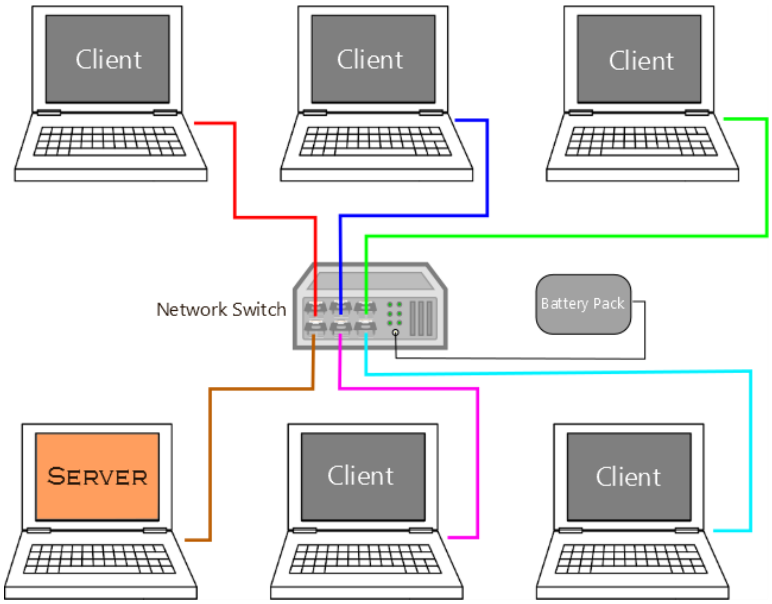


Figure 2.2.1 – A diagram of the way laptops were connected together in the stands.

In order to send data immediately to a central database, we implemented a classic server-client system with all of the laptops. Because competition rules restricted us from creating ad-hoc Wi-Fi networks, the laptops had to be wired together through a network switch *(Figure 2.2.1)*. The clients would simply request the Server's IP address to receive the scouting application. At the end of each match, data would then be sent via AJAX back to the server, which would write the data to a database.

Networking Challenges

These small issues took us quite some time to figure out, so for those trying to implement a similar system, here are some tips:

1. Each laptop needs a static IP. Since there is no DHCP server available to allocate IP addresses to the laptops, you need to set it manually. Only then will the server know which IP to respond to.
2. The server must have an exception in the firewall for incoming connections on port 80. Only then can the server send responses to the client through the port.
3. The Apache server configuration must be changed for it to be accessed over the network switch. Make the following changes to your `httpd.conf`
    a. Make the server listen to all addresses it receives on port 80 by changing the Listen statement to `Listen *:80`
    b. Change the `ServerName` from `localhost:80` to `<your-static-ip>:80`
    c. Within the default `<Directory>` directive, comment out `Require all denied` and add `Order deny,allow` as well as `Allow from all`
    d. Within the `<Directory>` directive of the path being served, change `Require local` to `Require all granted`

Data Transfer

As matches go on, the central database on the server accumulates data, but that is of no use unless the strategy team in the pit can use it to gain insight on their opponents. In the initial planning of the system, we contemplated various methods of data transfer, but all involved uploading data to a remote server using a USB tethered data connection from a team member's phone. We deemed this to be risky, as there is no guarantee for cellular service or Wi-Fi at the venue. For this reason, we decided to transfer data via a "USB Ferry." Every few matches, and especially before our own matches, data would be serialized into a file in the JSON format. A team member would then walk to the pit and import the data into a database there. After refreshing, the Tableau workbook would hold all the new data points.

Data Analysis

This year, like last year, we used Tableau for our data analysis and visualization. Tableau offers a wide variety of visualization options, from bar graphs and scatter plots to plotting on maps and rose plots. It allows the user to customize the visualization on its color, size, labels, detail or shape based on any data point you choose. It also allows the user to quickly create visualizations that fit the discussion's needs, without the need to spend lots of time writing SQL. All in all, Tableau is a very versatile, dynamic software that has a vast potential. However, in my experience using this software, Tableau also has a steep learning curve. It can be confusing at first, especially to one without experience in the Business Intelligence domain, and it requires a lot of time and effort to learn. Once you have an idea on how it all works, though, the rest is incredibly simple. For anyone learning to use Tableau, I would recommend learning about Level of Detail calculations, as they really come in handy.

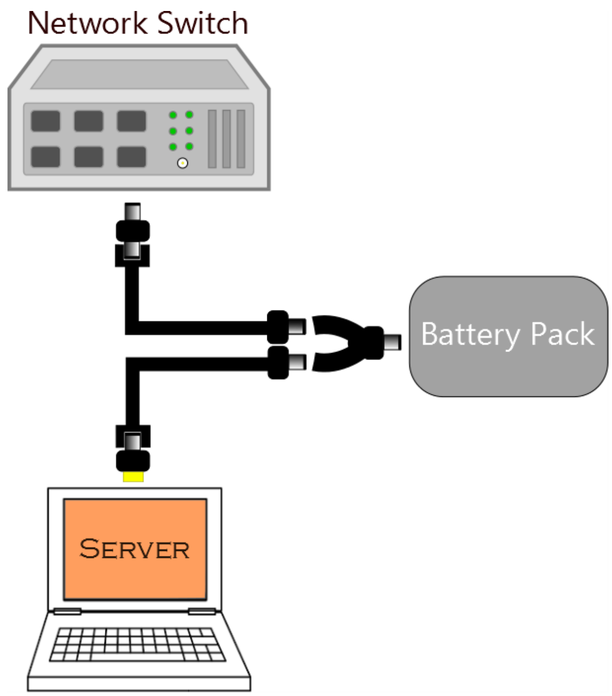
This software may sound too good to be true, but it's not. Many teams believe that its price of $999 is well above their means, but FIRST has actually included a free license to Tableau in the Kit of Parts for the past few years.

Logistics and Costs

Logistics

The biggest logistical problem for our team in implementing this scouting system was getting laptops and powering them. Buying 10 laptops was not practical for our budget, so we turned to our team members. We asked for as many ultrabooks as we could get because they would have a long battery life and we wouldn't need to worry about them losing power. Many ultrabooks also have a touchscreen, making it easier for the scouters to input data. Unfortunately, only a few members owned one, so we ended up with many regular notebooks as well.

Because of the low battery life of regular notebooks, we had to make sure that we always had at least 6 laptops with enough power to be scouted on. One of those could never be replaced, as it would act as the server and hold the central database. We also needed to power the network switch, which connected all of the laptops together. In order to provide constant power to both the server laptop as well as the network switch, we purchased a large 50,000 mAh battery and connected them together as per the diagram below *(Figure 2.5.1)*.



However, the battery pack only had enough power for the server and switch. Because not all of the laptops we had were ultrabooks, we needed to cycle them out with charged ones as they lost power. To do so, we had to keep the laptops not being used plugged in on the walkway above stadium seating. We had to keep a team member or parent stationed with the laptops to watch them. Logistically it isn't incredibly convenient, but we did not see another easy way.

The management of scouters at the competition is another logistical challenge to take into account. Scouting requires focus and patience, and after a while, people will get tired – which often leads to inaccurate data. Our team generally has anyone who is not in the pit, driving, or preparing for a presentation in the stands scouting. The idea is that as people get tired, someone else will take over their spot, ideally preserving the integrity of the data. While this works well for our team of about 30 members, smaller teams may face difficulty managing scouters in this fashion.

Implementing such a system also require time and labor. To put it into context, I am a relatively well-experienced web developer, and it took me about 100 hours to make the system. For teams without members knowledgeable in web development, it may take much longer.

Costs

1. 10 laptops, preferably small power efficient ultrabook: $219.00 * 10 = **$2190.00** (Acer Cloudbook AO1-431-C7F9 for base price)

   Having ten ultrabooks is an ideal situation. As our team did this year, we asked for students to lend us their laptops for a day. We first asked people who had ultrabooks, and when we did not have enough to run the system for an entire day, we asked those with regular notebooks. Although the system only requires seven laptops (six in the stands, one in the pit), it is always good to have backups in case they run out of power or software/hardware malfunctions.

2. 10 USB to Ethernet adapters: $9.99 * 10 = **$99.90** (AmazonBasics Ethernet LAN Network Adapter for base price)

   Since most ultrabooks do not have an Ethernet port, they will need to utilize a USB Ethernet adapter to connect to the network.

3. Ethernet Cables: $6.29 * 10 = **$62.90** (AmazonBasics 10ft RJ45 Cat-6 Ethernet Patch Cable for base price)

These are the cables that will connect the individual laptops to the network switch. Although only 6 Ethernet cables are required, it is good to keep spares in case of malfunctioning cables.

4. Gigabit Ethernet Switch: **$19.95** (TP-Link TL-SG1008D for base price)

   The Ethernet switch is the heart of the network. It is what connects the clients to the server, which hosts the application as well as the central database.

5. Battery Pack for the Server and Network Switch: **$135.99** (MAXOAK 50000mAh Power Bank for base price)

   Since the server holds the central database, it cannot be switched out with another laptop, and therefore must last the whole day. We used this battery pack to keep the server powered.

6. DC Barrel Plug Splitter: **$7.99** (PHC DC-3Z DC Barrel Plug Splitter for base price)

   Since both the server and network switch need to be powered by the power bank, we need to split the output into two.

7. 2 Male to Female DC Power Extension Cable: **$8.99** (Valley Enterprises Male to Female DC Power Extension Cable for base price)

   These are the wires that extend from the battery pack to the network switch and server.

8. Male to Male DC Power Coupler: **$6.99** (Valley DC CCTV Power Coupler Male to Male for base price)

   Since both the power extension and the network switch have a female output, we need a male to male coupler to connect them.

9. Tableau: **$0.00** (Tableau Desktop for FRC)

   All FIRST teams get a free license to the incredibly versatile Tableau, which we need to analyze all the data we collect.

Total: $2531.91

Although this looks like a huge price tag, the laptops can most likely be borrowed from the team, and the rest is a one-time investment, as the materials can be used for years.

A Look into the Data

A scouting system implemented in this way offers many uses throughout the competition, and can be very useful in a wide variety of situations. The following screenshots and explanations are simply meant to provide a practical context for building such a scouting system – to show how such data could be analyzed, interpreted, and be put to use. They do not consist of all the graphs we generated this season, but just a few important ones we used frequently.

Match Strategy

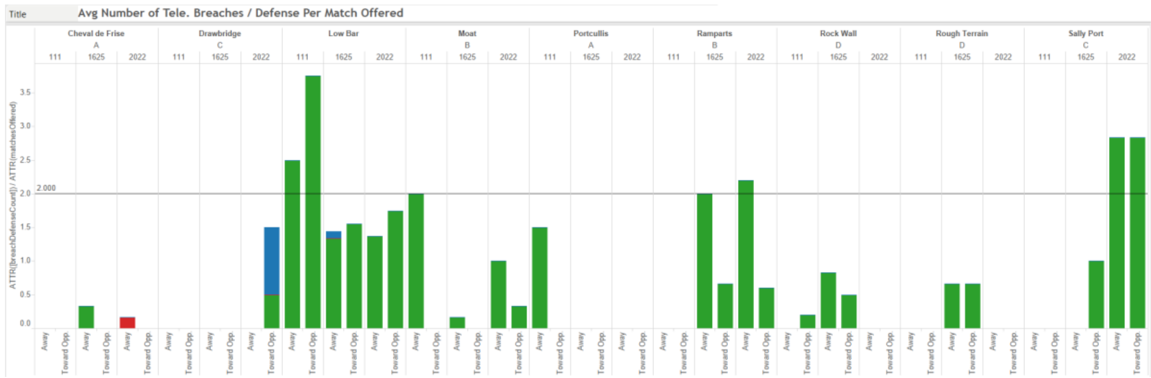


Figure 3.1.1 – A graph showing the average number of crosses each team performs per match where said defense was offered, grouped by defense and direction. The title in the image is a misnomer – "Breaches" should be "Crosses"

From the very beginning, scouting data can help with match to match strategy. For example, the graph above *(Figure 3.1.1)* shows how many times each team crosses a defense on average per match, split by the outcome (Success, Backed Out or Got Stuck). Data like this can help teams select their defenses in their matches. For the average team, an alliance of 111, 1625 and 2022 would be very hard to beat, but with a little bit of analysis, we can see that 111 really like to use the Low Bar, and if one could defend that defense, it would slow them down. Not only that, but either the Cheval de Frise gives all three teams some trouble, or they don't like to use it. Putting that on their side might prevent this super alliance from getting a breach.

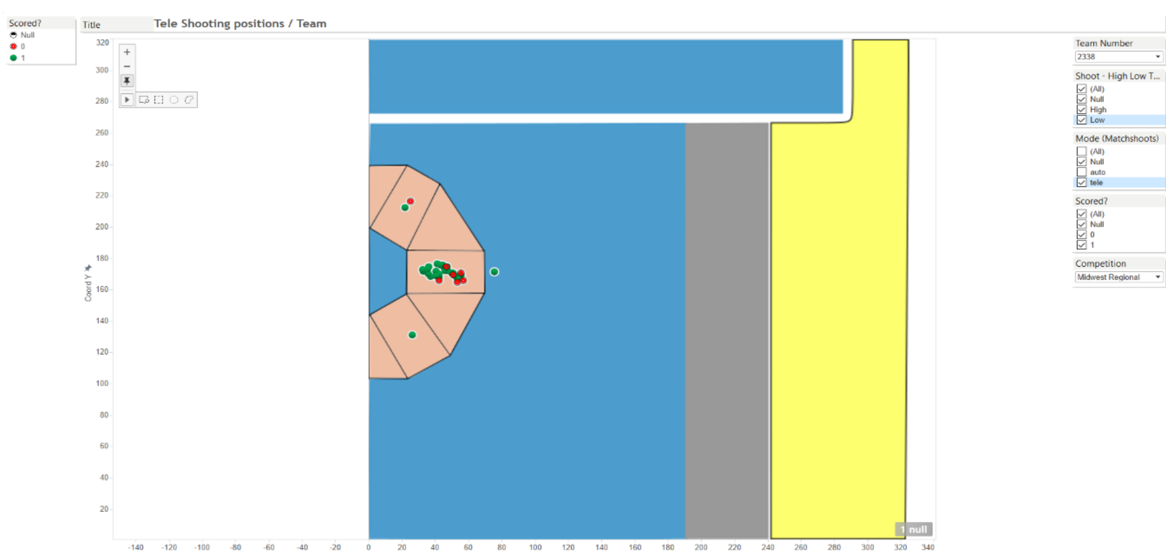


Figure 3.1.2 – A graph showing plots the position of each shot a robot has taken during the competition, as well as the outcome of the attempt.

The data also gives one the ability to strategize each one of your matches. If a team was going against team 2338, they could see on their Shooting Positions graph *(Figure 3.1.2)* that 2338 shoots primarily from one area – the center of the batter. They can strategize to include defending in that position, which might severely impair the ability of the opponents to score points.

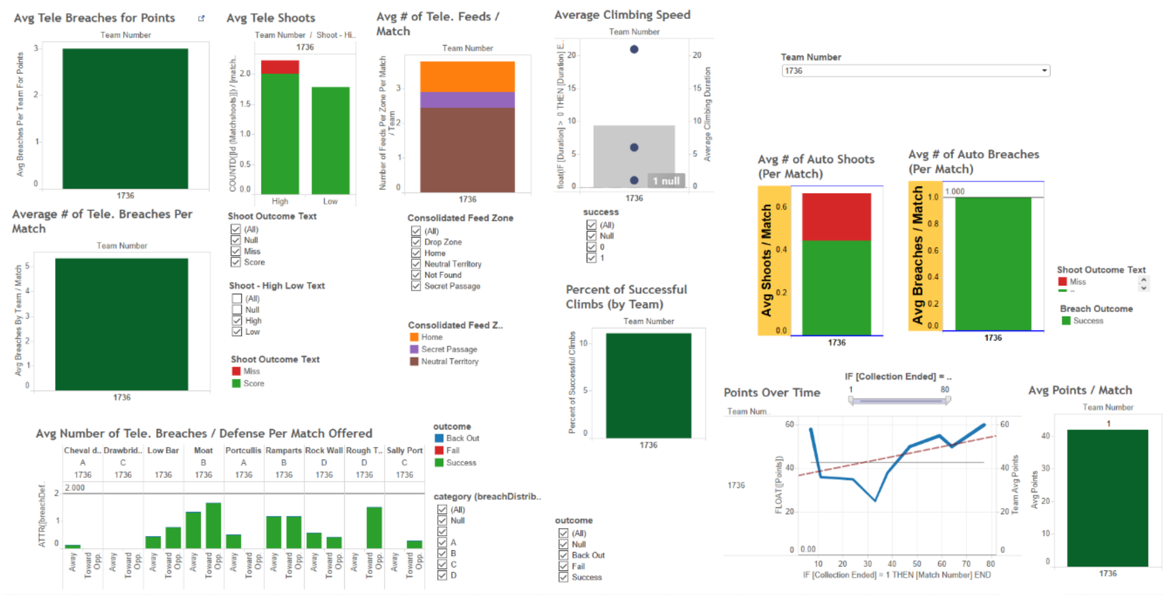


Figure 3.1.3 - The "Team Dashboard" which shows an overview of everything we know about a team's performance.

Sometimes, however, a team doesn't have a weak point that is easy to take advantage of. All the drive team can do is know what the other teams will likely do. At competition, a common demand is, "Give me all the data you've got on Team ____." Instead of sifting through many different graphs and applying filters to each one, we can simply create a team dashboard which holds all the graphs for a given team, as shown in the above image *(Figure 3.1.3)*. We can use this data in a rush when we just need to know what to look out for during a match.

Alliance Selections

The traditional use of scouting is creating the "pick-list" for their alliance selections, but often, teams don't know where to begin when it comes to creating one. Concrete data gives the user the ability to start with whatever has the biggest impact on their strategy.

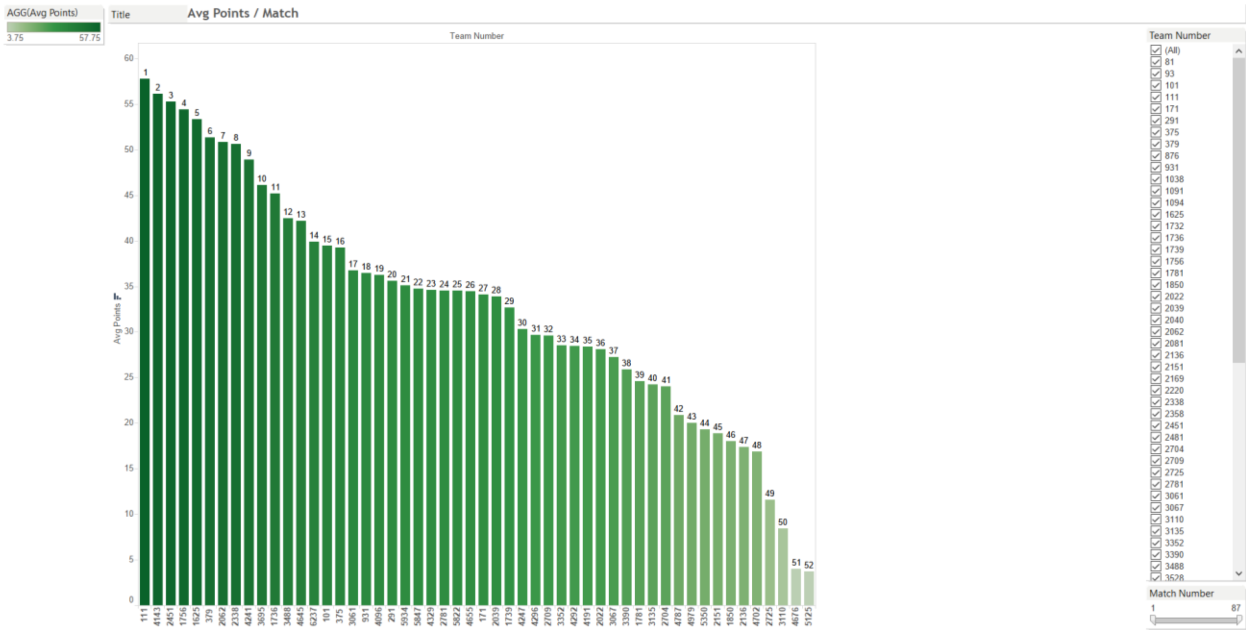


Figure 3.2.1 – A graph showing the average number of points a team scores per match, if they were alone on the field.

For example, teams looking for well-rounded robots can start with a graph for the average number of points scored per match by team *(Figure 3.2.1)*, and narrow it down by adding more criteria from there. With this graph, we can see that Team 111, 4143, and 2451 are the 3 best well-rounded scoring robots.

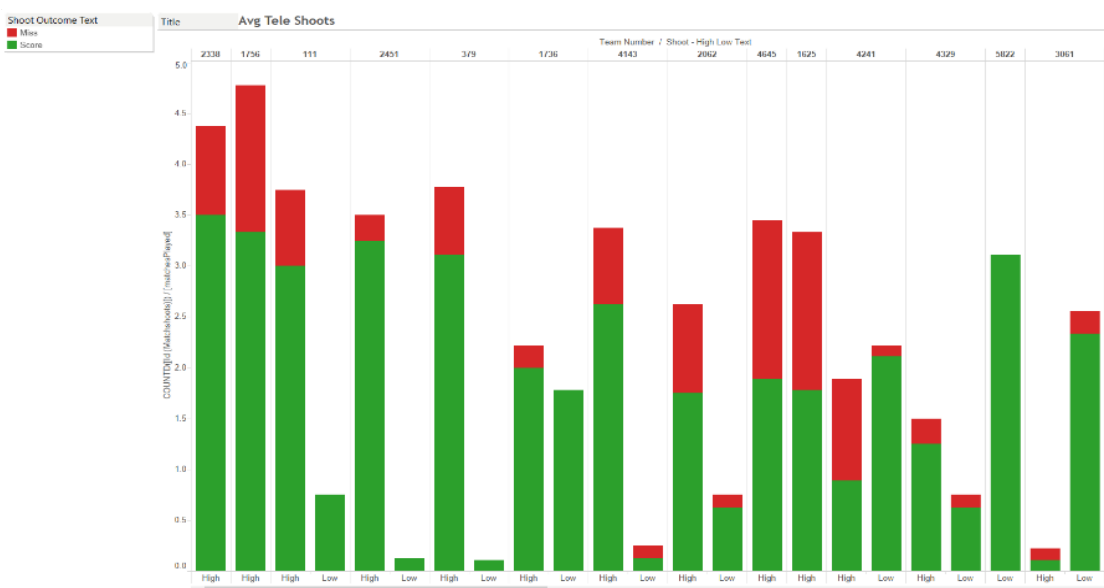


Figure 3.2.2 – A graph showing the average number of shoot attempts a team takes in the teleop mode, and how many of them make it into the goal.

The graph above *(Figure 3.2.2)* shows the average number of shots in the teleoperated mode, split up by the outcome and the height of the goal, sorted by the average number of points made on teleoperated shots per match. In the graph, we can see that team 379 has a relatively good shooter robot. Although this graph does not show their performance in other areas, a team looking for a robot that can just focus on shooting can keep 379 in mind. For that team, it is quite convenient that team 379 is ranked #18 by the Midwest Regional because team 379 will likely be overlooked by other teams – giving them the best shot to pick them.

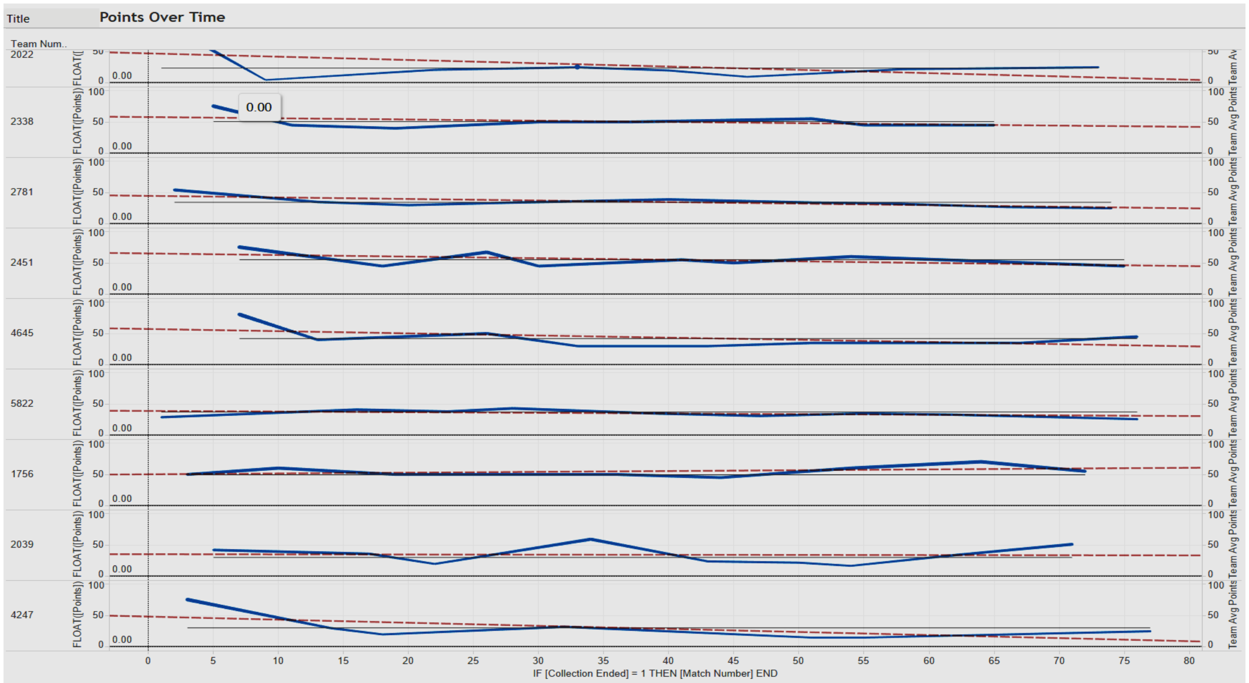


Figure 3.2.3 – A graph showing how many points teams score per match over time.

A versatile scouting system can not only tell you what teams you want on your alliance but also the teams you should be careful about. The graph above *(Figure 3.2.3)* shows the points scored by this team over time. There are two lines – the red is a linear regression, and the gray is the average. The graph is sorted by the slope of the regression line. The teams shown have the most negative slopes, which means over the course of the tournament, these teams' performance has decreased. We can be wary of choosing these teams before doing proper pit scouting to make sure their robot still works.

Future Goals

No matter how well the system works, it can be improved. In the future, we hope to make it easier logistically. We would like to find a way to keep the laptops powered without having to run up and down stairs to switch them out. The stands were pretty disorganized this year, with wires sprawled everywhere. It was very hard to set up initially. In future years, it would be very nice to have better organization in the stands to allow for better maintenance. We would also like to review the graphs we created before and the graphs we created during competition and make a plan on what kinds of graphs we as a team tend to use, so as to be more prepared. To further extend this guide on implementing a dynamic scouting system, I hope to write another paper detailing the way Tableau works and how it can be used in this context.

Conclusions

The mission of FIRST is to prepare students for work in the real world by developing skills they would use there. In that manner, a scouting system does a lot of that. By building a scouting system, students gain life skills of critical thinking, strategic planning, and risk analysis. It is another mission of FIRST to inspire students to appreciate technology. It might just be me, but it's pretty hard to not appreciate how much clicking a few buttons every time a robot performs an action can help a team come to the right conclusions.

This year, Ctrl-Z has created a scouting system which fulfills all of the basic functions of scouting, while also being easy to use, fast and efficient at aggregating data, and dynamic enough to allow the team to generate new reports as needed. We collected 51,615 data points in this season alone! Given that, there are a few shortcomings. It requires a bit of logistical planning, full stack web developers, and a good amount of money – which many teams may not be able to afford. Personally, I share the same view as many others – that scouting is not only useful for the team for winning matches, but also an integral part of the competition experience. Although not everyone is able to implement a system similar to this one, everyone should make some sort of effort to scout teams – it's pretty amazing what one can do with the right data.

Team 4096 would love to help other teams plan and execute their own scouting systems, whether it is like this one or not. Please feel free to contact me with any questions as scouting@team4096.org.

All application source code and Tableau Workbooks for analysis can be found at https://github.com/CtrlZ-FRC4096/FRC-Scout-2016.You can read this paper as well as any others by Ctrl-Z FRC Team 4096 at our website www.team4096.org.