Paly Robotics Team #8 Scouting Documentation

Authors: Aditya Ramanathan and Zachary Brandt

Table of Contents

I. Introduction and Structure

Importance of Scouting

Scouting is essential at every competition. Our team's goal is to rank first at regionals or be the most appealing first pick at our regional. When we place high in the rankings, it is important to be able to choose alliance partners that will compliment us during the elimination matches. That's where scouting comes in. It shows us the data of each robot, so we can effectively determine what the best teams to select would be.

Competition Structure

At competitions, we have groups of six scouters for an hour at a time. One of the six members of the scouting group will be the lead scouter. The primary purpose of the lead scouter is to ensure that all of their members complete the scouting tasks in the app. We use 6 iPads (one for each robot on the field) to collect the data. If scouters make a mistake while scouting a match, we do not upload the data to our database because it is better to have no data than false data. In addition, we can also scout the match later because most matches are recorded and can be viewed on thebluealliance.com

Who Scouts?

All freshmen and sophomores who attend competitions must scout. The more scouters we have the less time each scouter has to scout for. There are exceptions to these rules. Anybody on our drive team does not have to scout nor does the pit manager or competition manager.

Lead scouters are usually the students who are involved with Strategy and Scouting.

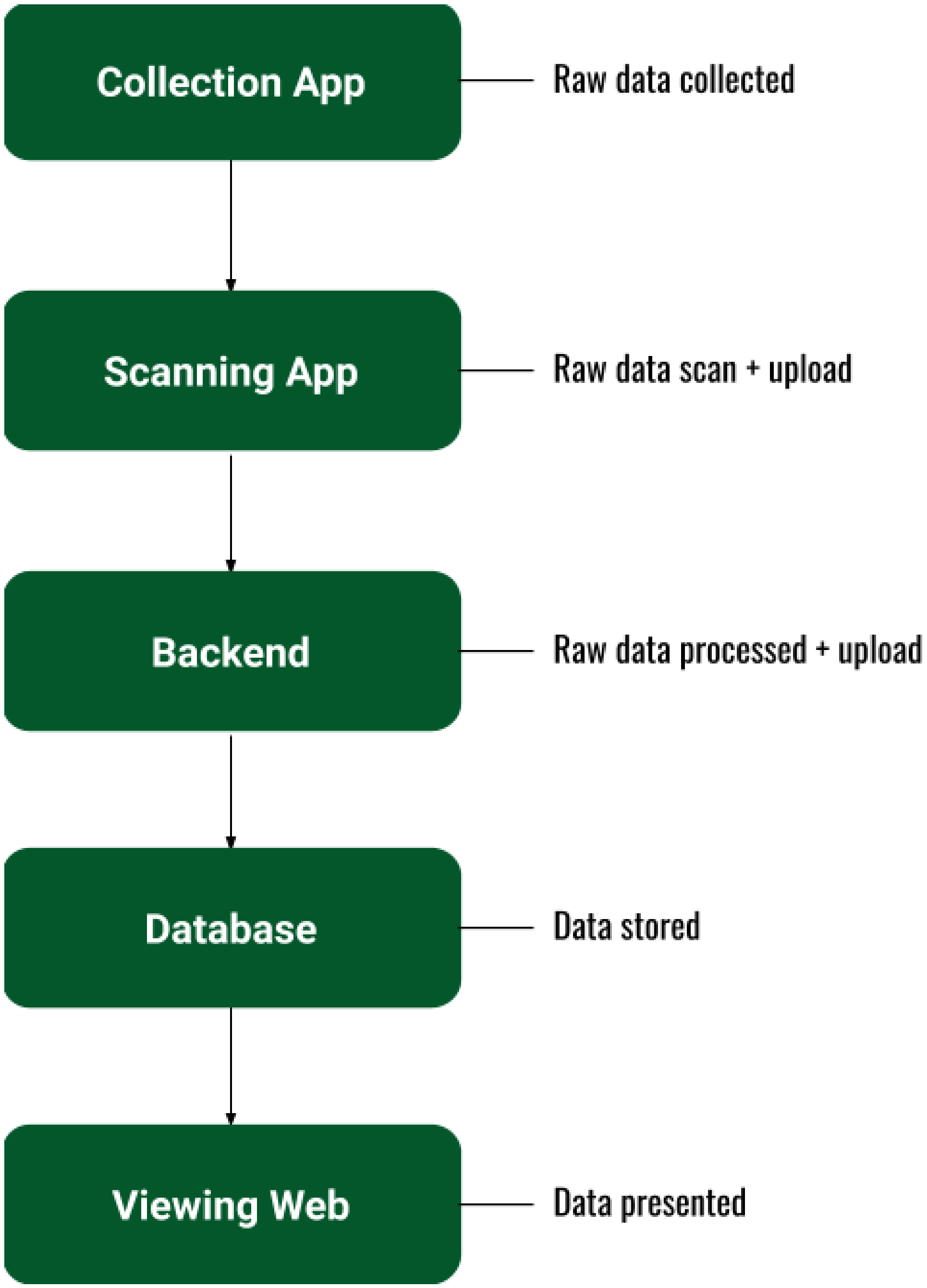
Scouting Training

All scouters must be familiar with the game and have learned how to use the collection app. To achieve these prerequisites, scouters must complete the game rules test and complete basic scouting practice.

Roughly 2 weeks before our first competition we make all scouters take the game rules test which is a simple Google form with questions. Here is a link to our game rules test from infinite recharge V1. The goal of the game rules test is to ensure that scouters are familiar with the gameplay. The test is broken down into three categories, the first contains some straightforward questions concerning point values, these questions ensure that the scouter knows the value of performing individual tasks. The second category evaluates the scouter's knowledge of key rules. The third category requires more creativity and challenges the scouter to write responses to questions like, “Explain a way for a robot to score exactly 19 points by itself in auto. Give a detailed explanation of the path it takes, shooting positions, etc”.

After taking the game rules test, scouts are eligible for scouting practice which is a quick 10 minute activity for scouters to get familiar with the collection app. After scouters learn how to use the app they practice by watching the previous week’s competitions. If we have week 1 competitions scouters can practice by watching week 0 competitions. Once they have finished scouting practice they are ready to attend and scout at competitions. If scouters do not finish this practice, they are not able to attend the competition.

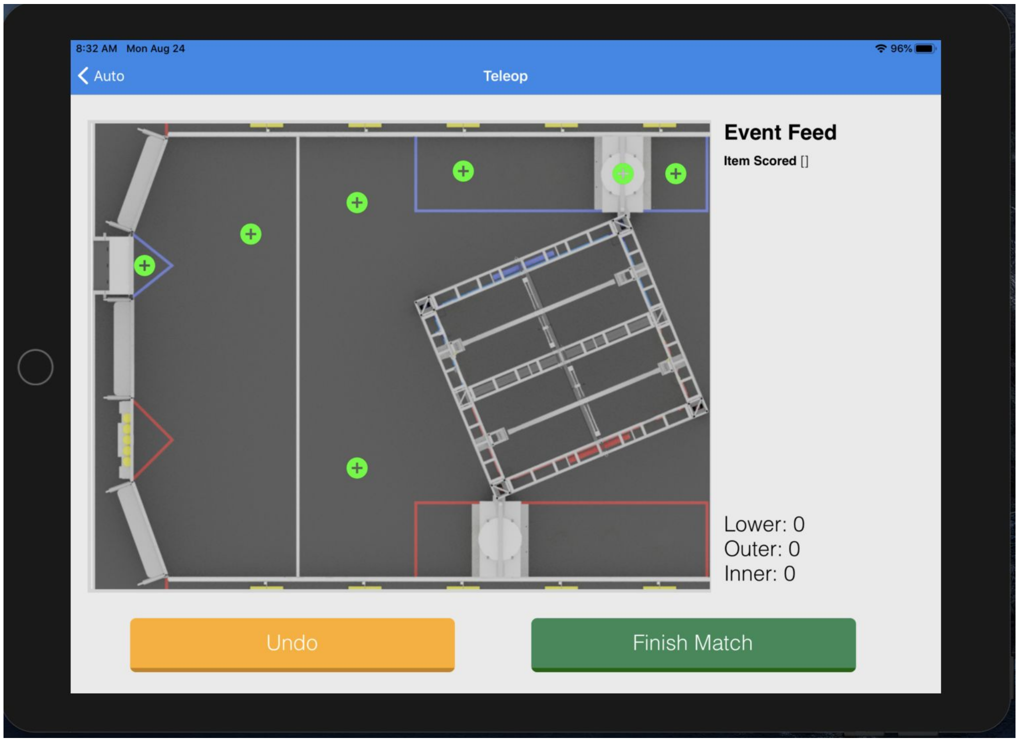
II. Implementation



Collection App

In essence, the collection app is the app where we collect the data. We installed this app on each of our six iPads. The collection app has 6 screens which are:

1. **Match List Screen** - Scouters select the team they are going to scout. We use The Blue Alliance API to get the teams that are playing.
2. **Pre Match Screen** - Scouters enter their name so we know who was scouting the team and they select the starting position of the robot. They also have the option to toggle the field orientation if they don't like the default view on the iPad.
3. **Auto Screen** - In this screen we collect the number of powercells that were scored in each port during auto. We decided to include both inner and outer because after scouting at the Utah regional we realized that it is easy to tell whether power cells went into the outer or inner ports.
4. **Teleop Screen** - In the teleop screen we not only collect which ports the robots score in but also where they scored from. We have 5 different positions that scouters can select, as well as another category if a robot shoots from somewhere else. In addition to collecting data about scoring, we also collect control panel data, which is as simple as completed rotational control/positional control.
5. **Post Match Screen** - The post match screen is where we collect data that is collected after a match. We have a general notes section and a hang notes section. We also collect data on whether or not the robot died or crossed the initiation line. We also collect hang status. Finally, we collect miscellaneous ratings which include driver rating, defense rating, shooting rating, and powercell intake rating.
6. **QR Code Screen** - The app compiles all the data into a QR code and displays it on the screen, it also shows all of the data that you have collected if you want to look back at it.

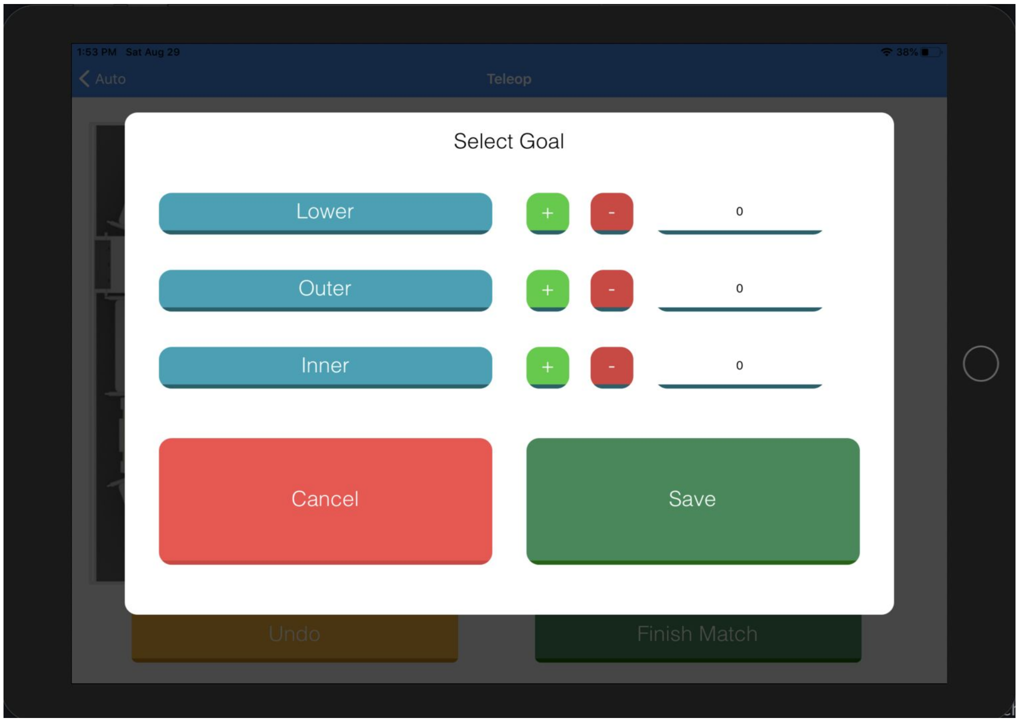


Event Feed

Item Scored []

Lower: 0
Outer: 0
Inner: 0

This is the teleop screen. The green buttons are where you can select the position of where the robot is shooting from, when you click one of the buttons a modal open.



Lower 0
Outer 0
Inner 0

This is the modal where scouters can select how many power cells were scored in each position.

Some scouting systems don't show the game field in their collection app even though, showing the game field gives scouters a better experience while scouting. The scouting experience with the game field visible is superior because it makes it easier to scout. Without the game field, scouters would be pressing random buttons without any context and makes it harder to use the app which leads to frustration with scouting.

Scanner App

The scanner app is the most simple component of our scouting system. It scans QR codes and shows what team you have scanned. Finally, you can press the upload data button and the data gets sent to our backend.

Backend

Modules

The backend for our scouting app was written using JavaScript and uses Node.js runtime environment.

The backend is broken down into multiple modules, each performing their own role in processing raw data from the scanner app and uploading processed data to the cloud database. The server module handles requests made to the backend which are made through tunnels to localhost and calls the other two utilities and firebase interactor modules. The utilities module processes raw data into JSON, adds, and updates the team's statistics, and interacts with The Blue Alliance API. The firebase interactor accesses the cloud database in order to initialize or upload data.

Cloud Database

Our cloud data is stored on our Firebase realtime database. Individual unprocessed match data is stored on QR codes generated from the collection app, and the backend processes this raw data and pushes it to the cloud database. The cloud database contains data on each team's performance and matches across multiple events. We can make a request to the backend to pull data from the cloud database. To upload data to the database, requests are made to the backend with individual match data.

The Blue Alliance API

We get event data and match schedules from the Blue Alliance's API. After making a request to initialize a specific event (multiple events can be stored in the database) the database organizes itself by event, team, and match as well as having empty stats data.

Logging

When requests made to the backend are successfully handled and executed, the backend gives a visual terminal output that shows what the backend is doing. Each instance of initializing an event is logged in the terminal, as well as each instance of match data uploaded is logged in the terminal.

Requests

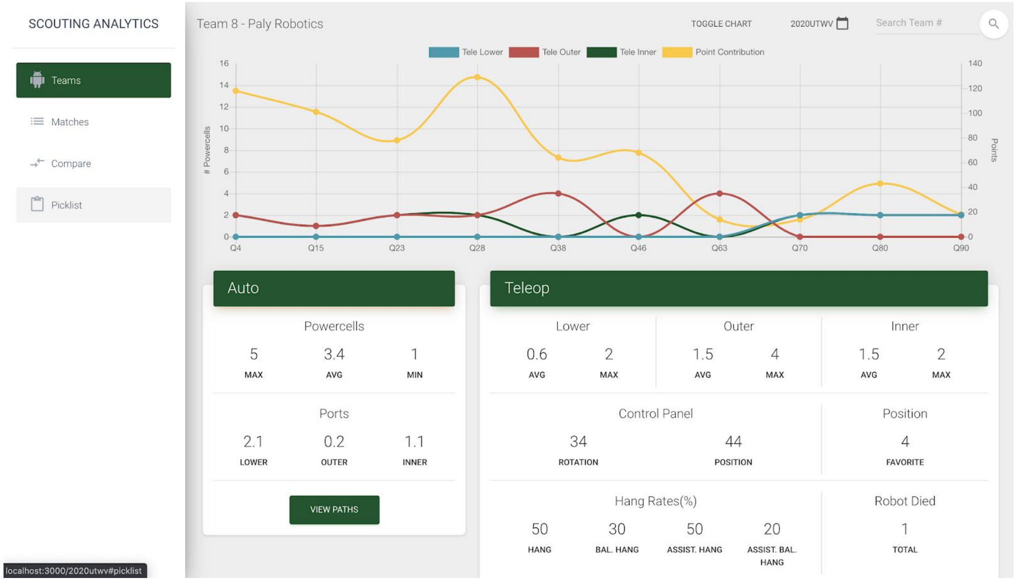
Three requests can be made to the backend, an initialize event request, an upload match data request, and a get data request. The initialize event request is a post request which pushes event and empty team match data as well as empty team stats data to the cloud database and contains the event key (from The Blue Alliance) in the request parameters. The upload match data is a post request which pushes individual match data to the database by event and team and contains raw scouting data in the body. The get data request is a get request which pulls the entirety of a specific event stored in the cloud database and returns it in the response and contains a specific event key.

Viewing Website

The viewing website is where we are able to view all the data that we have collected from the matches. We divided this website into 4 parts.

1. **Teams** - We are able to view data on specific teams. In this section, we are able to view a graph of tele lower, outer, and inner, as well as point contribution (see match data descriptions). The graph is important because it shows us whether teams have been doing better or worse as the competition moves forward. We can also view stats from auto and teleop. Finally, we can view miscellaneous data such as ratings and notes.
2. **Matches** - On the matches page we are able to view stats from individual matches. This includes viewing each robot's point contribution and scoring for the match. Each team's contribution to the match can be clearly seen.

3. **Compare** - In the compare screen, we can view important data. We are able to sort teams by point contribution, tele lower, tele outer, tele inner, and driver ratings. The sort feature is helpful because we can view the top teams without having to scroll and look for teams. This saves us time when we want to create our picklist.
4. **Picklist** - The picklist page in our viewing website allows us to create a picklist. We are able to view data that quickly gives us a good sense of the team's performance on this page as well. Some of the data that is displayed here includes tele lower, outer, and inner averages and stats about hangs.



Auto

Powercells

5	3.4	1
MAX	AVG	MIN

Ports

2.1	0.2	1.1
LOWER	OUTER	INNER

Teleop

Lower		Outer		Inner	
0.6	2	1.5	4	1.5	2
AVG	MAX	AVG	MAX	AVG	MAX

Control Panel		Position
34	44	4
ROTATION	POSITION	FAVORITE

Hang Rates(%)				Robot Died
50	30	50	20	1
HANG	BAL. HANG	ASSIST. HANG	ASSIST. BAL. HANG	TOTAL

III. Data Collected

Match Data Description

`match/PowercellIntakeRating:` An integer anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`match/defenseRating:` An integer anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`match/driverRating:` An integer anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`match/shootingRating:` An integer anywhere from 0 to 5. 0 Where is N/A and 5 is the highest rating

`match/event:` Event key string obtained from The Blue Alliance

`match/alliance:` A string either “blue” or “red”

`match/assistedBalancedHang:` A number either 0 or 100. 0 being no assisted balanced hang was performed during the match and 100 meaning there was

`match/assistedHang:` A number either 0 or 100. 0 being no assisted hang was performed during the match and 100 meaning there was

`match/soloBalancedHang:` A number either 0 or 100. 0 being no solo balanced hang was performed during the match and 100 meaning there was

`match/soloHang:` A number either 0 or 100. 0 being no assisted balanced hang was performed during the match and 100 meaning there was

`match/autoEvents:` A dictionary containing events that occurred in auto

`match/teleEvents:` A dictionary containing events that occurred in teleop

`match/autoLower:` The number of power cells scored in the lower port during auto

`match/autoOuter:` The number of power cells scored in the outer port during auto

`match/autoInner:` The number of power cells scored in the inner port during auto `match/autoPowercellPoints:` The number of points scored in auto through scoring

`match/telePowercellPoints:` The number of points scored in tele through scoring

`match/autoPowercellTotal:` The total number of power cells scored in auto

`match/telePowercellTotal:` The total number of power cells scored in tele

`match/notes:` A string. The general notes of a match collected

`match/hangNotes:` A string. The hang notes collected during the match

`match/initiationLine:` A number either 0 or 1. 0 being they did not cross the initiation line and 1 meaning they did.

`match/match:` A string beginning with a capital Q and the match number

`match/name:` A string containing the scouters name

`match/pointContribution:` The number of points the team scored in the match

`match/rotationControl:` A string either true or false

`match/positionControl:` A string either true or false

`match/robotDied:` A number either 0 or 100. 0 being the robot did not die. 100 meaning the robot died

`match/startingPosition:` A number between 1 and 5 based on where the robots start

`match/team:` The team's team number

`match/tele1:` A dictionary which records the lower, outer, and inner goals that were made from position 1

`match/tele2:` A dictionary which records the lower, outer, and inner goals that were made from position 2

`match/tele3:` A dictionary which records the lower, outer, and inner goals that were made from position 3

`match/tele4:` A dictionary which records the lower, outer, and inner goals that were made from position 4 `match/tele5`: A dictionary which records the lower, outer, and inner goals that were made from position 5

`match/tele6`: A dictionary which records the lower, outer, and inner goals that were made from position 6

`match/teleLower`: A number which is the amount of power cells scored in the lower port during teleop

`match/teleOuter`: A number which is the amount of power cells scored in the outer port during teleop

`match/teleInner`: A number which is the amount of power cells scored in the inner port during teleop

`match/telePosition`: The position that is the most scored from for the team in that specific match

`match/telePosition1Total`: The amount of power cells scored from position 1

`match/telePosition2Total`: The amount of power cells scored from position 2

`match/telePosition3Total`: The amount of power cells scored from position 3

`match/telePosition4Total`: The amount of power cells scored from position 4

`match/telePosition5Total`: The amount of power cells scored from position 5

`match/telePosition6Total`: The amount of power cells scored from position 6

Stats Data Description

`avg/assistedBalancedHangAverage`: A number between 0 and 100. The average of `match/assistedBalancedHang` values for all matches of a specific team

`avg/assistedHangAverage`: A number between 0 and 100. The average of `match/assistedHang` values for all matches of a specific team

`avg/autoInnerAverage`: The average amount of power cells scored in the inner port during auto

`avg/autoLowerAverage`: The average amount of power cells scored in the lower port during auto `avg/autoOuterAverage`: The average amount of power cells scored in the outer port during auto

`avg/autoPowercellPointsAverage`: The average amount of points scored in auto through scoring

`avg/defenseRatingAverage`: The average integer rating anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`avg/driverRatingAverage`: The average integer rating anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`avg/pointContributionAverage`: The average number of points the team scored in the match

`avg/powercellIntakeRatingAverage`: The average integer rating anywhere from 0 to 5. Where 0 is N/A and 5 is the highest rating

`avg/robotDiedAverage`: The likelihood of a robot dying across all matches between 0 or 100.

`avg/shootingRatingAverage`: The average integer rating anywhere from 0 to 5. 0 Where is N/A and 5 is the highest rating

`avg/soloBalancedHangAverage`: A number between 0 and 100. The average of `match/soloBalancedHang` values for all matches of a specific team

`avg/soloHangAverage`: A number between 0 and 100. The average of `match/soloHang` values for all matches of a specific team

`avg/teleInnerAverage`: The average amount of power cells scored in the inner port during teleop

`avg/teleLowerAverage`: The average amount of power cells scored in the lower port during teleop

`avg/teleOuterAverage`: The average amount of power cells scored in the outer port during teleop

`avg/telePositionAverage`: The position that is the most scored from for the team in that specific match `avg/telePowercellPointsAverage`: The average amount of points scored in teleop through scoring

`max/assistedBalancedHangMax`: A number either 0 and 100. The highest value of all `match/assistedBalancedHang` for all matches of a specific team

`max/autoInnerMax`: A number, the highest value of all `match/autoInner` for all matches of a specific team

`max/autoLowerMax`: A number, the highest value of all `match/autoLower` for all matches of a specific team

`max/autoOuterMax`: A number, the highest value of all `match/autoLower` for all matches of a specific team

`max/autoPowercellPointsMax`: A number, the highest value of all `match/autoPowercell` for all matches of a specific team

`max/defenseRatingMax`: A number between 0 and 5, the highest value of all `match/defenseRating` for all matches of a specific team

`max/driverRatingMax`: A number between 0 and 5, the highest value of all `match/driverRating` for all matches of a specific team

`max/pointContributionMax`: A number, the highest value of all `match/pointContribution` for all matches of a specific team

`max/powercellIntakeRatingMax`: A number between 0 and 5, the highest value of all `match/powercellIntakeRating` for all matches of a specific team

`max/shootingRatingMax`: A number between 0 and 5, the highest value of all `match/shootingRating` for all matches of a specific team

`max/soloBalancedHangMax`: A number between 0 and 100, the highest value of all `match/soloBalancedHang` for all matches of a specific team

`max/teleInnerMax`: A number, the highest value of all `match/teleInner` for all matches of a specific team

`max/teleLowerMax`: A number, the highest value of all `match/teleLower` for all matches of a specific team `max/teleOuterMax:` A number, the highest value of all `match/teleInner` for all matches of a specific team

`max/telePowercellPointsMax:` A number, the highest value of all `match/telePowercell` for all matches of a specific team

`min/assistedBalancedHangMin:` A number either 0 and 100. The lowest value of all `match/assistedBalancedHang` for all matches of a specific team

`min/autoInnerMin:` A number, the lowest value of all `match/autoInner` for all matches of a specific team

`min/autoLowerMin:` A number, the lowest value of all `match/autoLower` for all matches of a specific team

`min/autoOuterMin:` A number, the lowest value of all `match/autoOuter` for all matches of a specific team

`min/autoPowercellPointsMin:` A number, the lowest value of all `match/autoPowercellPoints` for all matches of a specific team

`min/defenseRatingMin:` A number between 0 and 5, the lowest value of all `match/defenseRating` for all matches of a specific team

`min/driverRatingMin:` A number between 0 and 5, the lowest value of all `match/driverRating` for all matches of a specific team

`min/pointContributionMin:` A number, the lowest value of all `match/pointContribution` for all matches of a specific team

`min/powercellIntakeRatingMin:` A number between 0 and 5, the lowest value of all `match/powercellIntakeRating` for all matches of a specific team

`min/shootingRatingMin:` A number between 0 and 5, the lowest value of all `match/shootingRating` for all matches of a specific team

`min/soloBalancedHangMin:` A number between 0 and 100, the lowest value of all `match/soloBalancedHang` for all matches of a specific team

`min/teleInnerMin:` A number, the lowest value of all `match/teleInner` for all matches of a specific team `min/teleLowerMin:` A number, the lowest value of all `match/teleLower` for all matches of a specific team

`min/teleOuterMin:` A number, the lowest value of all `match/teleOuter` for all matches of a specific team

`min/telePowercellPointsMin:` A number, the lowest value of all `match/telePowercell` for all matches of a specific team

`total/assistedBalancedHangTotal:` The total amount of assisted balanced hangs performed in the event

`total/assistedHangTotal:` The total amount of assisted balanced hangs performed in the event

`total/autoPowercellPointsTotal:` The total amount of points scored in auto by the team in the event

`total/positionControlTotal:` The total amount of position controls performed by the team in the event

`total/robotDiedTotal:` The total amount of times that the team's robot died in the event

`total/rotationControlTotal:` The total amount of rotation controls performed by team in the event

`total/soloBalancedHangTotal:` The total amount of balanced hangs performed solely by the team in the event

`total/soloHangTotal:` The total amount of hangs performed solely by the team in the event

`total/telePosition1MatchTotal:` The total amount of times the team's robot scored in teleop from position 1

`total/telePosition2MatchTotal:` The total amount of times the team's robot scored in teleop from position 2

`total/telePosition3MatchTotal:` The total amount of times the team's robot scored in teleop from position 3

`total/telePosition4MatchTotal:` The total amount of times the team's robot scored in teleop from position 4 `total/telePosition5MatchTotal:` The total amount of times the team’s robot scored in teleop from position 5

`total/telePosition6MatchTotal:` The total amount of times the team’s robot scored in teleop from position 6

`total/telePowercellPointsTotal:` The total amount of points in teleop scored by the team in the event

IV. Creating a Picklist

Before selecting an alliance, we create a list for first picks and second picks by looking at our scouting data. Usually, our first pick would be a robot that could score well and efficiently. Our second pick is normally a defense robot that will slow down other robots from scoring. After we finished creating a picklist we discussed possible situations that can happen during alliance selection, however you can never predict it completely.