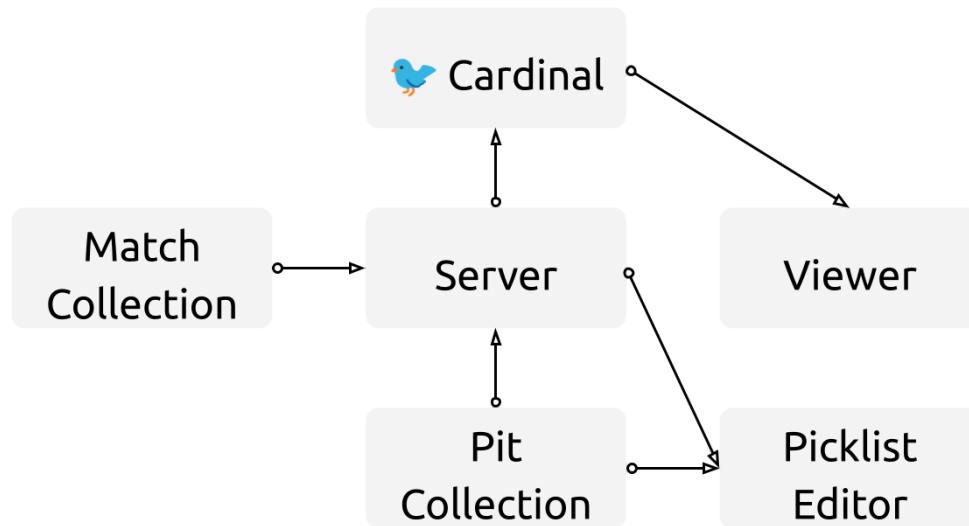


CITRUS CIRCUITS 2021 SCOUTING SYSTEM



OVERVIEW

This whitepaper reviews the different parts of Team 1678 Citrus Circuits' entirely electronic-based scouting system in the 2021 season. Since 2021 in-person season competitions were cancelled due to COVID-19, we used the 2021 build/competition season training new and experienced members. The majority of our improvements took place during the offseason for our two offseason competitions.

We use an electronic scouting system that requires a collection of 20 tablets, 3 Android phones, and one computer at competition. Our developer team is about 25 students, split into two equal groups: Front End (responsible for the apps and spreadsheet that interface with the scouts and strategy team) and Back End (responsible for the server and web server to process, store, and send data). Each end has one designated student lead. Our subteam meets for about 8 hours a week during offseason and will increase to about 21 hours a week during build and competition season.

The aim of this document is to briefly describe the functionality of the system, rather than its inner workings. We highly encourage you to peruse our [GitHub Repositories](#) and email us with any questions at: softwarescouting@citruscircuits.org



PIT COLLECTION

The Pit Collection app is used by pit scouts at competition to record game-specific pit data about each robot. The app is split into two modes: Objective and Subjective. Objective Pit Collection is used by our Match Strategist to record objective, straight-forward data (see below) while Subjective Pit Collection collects specialized, more subjective pit data. For example, during the 2020 season, the 1678 robot had a “buddy” climb that required a strap to be “cheesecaked” onto another robot and Subjective Scout had to assess whether or not the strap could be installed.

List of Features:

- **List page** - Lists teams in order of the teams_list.json that is downloaded manually onto the pit collection app.
- **Collection page** - opened by clicking a team's cell in the list page. Once data has been inputted and saved, the cell in the list will turn green. If there are photos but not data it will be.

Objective data collected: Can fit through the trench, Has ground intake, Drivetrain type (Tank, Mecanum, Swerve, Other), Drivetrain motor type (Minicim, Cim, Neo, Falcon), Number of drivetrain motors, Pictures (full robot, shooter, drivetrain, etc.)

Subjective data collected: Climber strap installation difficulty (1-10) with additional notes



Version: 2.0.2

OBJECTIVE PIT COLLECTION

SUBJECTIVE PIT COLLECTION (SEALS)

Version: 2.0.2, Objective Collection

253

841

852

971

9971

973

9973

9323

1662

1671

2073

3859

3970

4255

5104

Version: 2.0.2, Objective Collection

253

CAN NOT CROSS TRENCH

CAN NOT GROUND INTAKE

Drivetrain

Drivetrain Motor Type

of Motors

SAVE

Version: 2.0.2, Subjective Collection

253

Climber Strap Installation Difficulty

☐ 1

☒ 2

☐ 3

☐ 4

☐ 5

☐ 6

☐ 7

☐ 8

☐ 9

☐ 10

Climber installation notes

SAVE

MATCH COLLECTION

Just like Pit Collection, Match Collection is split into two modes: Objective and Subjective. Objective Mode (aka the Scout) is used by 18 students at once, with 3 watching each robot. As the name implies, all of the data collected in Objective mode is objective—scouts are trained to follow the robot and use the app with consistency, rather than being trained to make subjective decisions on whether a robot is 'good' or 'bad.' Subjective mode (aka the Super Scout) is used by two students at once, with each scouting one of the two alliances. Super scouts are members of our strategy subteam and are trained to make more subjective, informed decisions on robots' performance.

Objective Mode Features:

- **Match set-up page** - input alliance color, type of assignment of robot (automatic or override), scout name, scout ID (assignment number, 1-18), match number
- **Collection page** - start timer, switch to teleop, rotation/position control, low goal count, high goal count, start/stop incapacitated, start/stop climb attempt, undo/redo action
- **Edit page** - change any information inputted wrong in the set-up page, such as scout name, team scouted, etc.
- **QR display page** - Displays QR with compressed data to be scanned and downloaded by the server computer

Subjective Mode Features:

- **Match set-up page** - input alliance color, assignment (automatic or override), super scout name, match number
- **Collection page** - rank each team on the alliance compared to one another (3 is best and 1 is worst) in field awareness and quickness.
- **QR display page** - Displays QR with data to be scanned and downloaded by the server computer

Data Collected: [GitHub Schema Repository](#)





 	Match Number 31
Automatic Assignment	Team One 115
Backup 1	
Scout ID: 1	Proceed
Version: 2.0.2	

Start Climb Attempt	Redo
Start Incap	Undo
Low Goal - 5	High Goal - 0
Rotation	Position
TELEOP - 114	115 Proceed

Match Number 31
Team One 115
Backup 1
Proceed


Proceed

 	Match Number 32
Automatic Assignment	Team One 5285
	Team Two 1661
Backup 1	Team Three 5818
Version: 2.0.2	Proceed

5285			
Quickness	- 1 +	Field Awareness	- 3 +
1661			
Quickness	- 2 +	Field Awareness	- 2 +
5818			
Quickness	- 3 +	Field Awareness	- 1 +
Proceed			

Match Number 32
Team One 5285
Team Two 1661
Team Three 5818
Backup 1
Proceed


Proceed



VIEWER

The Viewer is an Android visualization app written in Kotlin that displays the data processed by Server and gathered by our Collection apps. The Viewer connects to Cardinal, which returns JSON-formatted data. This app is used by our drive coach, match strategist, and strategy mentors/students

List of Features:

- **Match Schedule**
- **Our Match Schedule** - filtered for only 1678 robots
- **Rankings**
- **First Pickability** - list of robots ordered by their attributes and data that makes them a potential first pick (weighted and calculated by the server)
- **Second Pickability** - list of robots ordered by their attributes and data that makes them a potential second pick (weighted and calculated by the server)
- **Team List** - list of teams in numerical order
- **Manual refresh** - updates data when clicked; displays timestamp of when new data was last pulled
- **Match Details** - opens when a match cell is pressed (such as in Match Schedule). Displays alliance data, such as RPs and scores, as well as team specific data
- **Team Details** - opens when a team cell is pressed (such as in Pickability list or Match Details) and displays all the data on the team. Team in Match Data is not displayed in this screen at the moment
- **Other:**
 - **Predicted and actual data** - a match's predicted data will be replaced with its actual data once a match has been played. On the Match Details screen, team data will be replaced with team in match data.
 - **Photos in team details** - only displayed if photos are downloaded locally on the device
 - **Navigate Drawer** - navigate between activities from anywhere in the app



Version 1.3.18

Pred. climb RP	0.0	Pred. climb RP	0.0
Pred. score	0.0	Pred. score	80000
Pred. 3rd stage RP	0.0	Pred. 3rd stage RP	0.0

Team: 1678 6657B 1671 7419 841 1678C

Auto						
Auto Balls Low	0	0	0	0	0	0
Auto Balls High	0	0	0	0	0	0
Tele						
Tele Balls Low	0	0	0	0	0	0
Tele Balls High	0	0	0	0	0	0
Other						
Incap (s)	0	0	0	0	0	0
Climb (s)	0	0	0	0	0	0

Version 1.3.18

1678

Citrus Circuits

PICTURE

Current RPs	0
Current Rank	0
Current Avg RPs	0.0
Pred. RPs	0.0
Pred. Rank	0

Auto	
Auto Low	0.0
Auto High	0.0
# Auto Lines	0
Auto Max Balls Low	0
Auto Max Balls High	0

Tele	
Tele Low	0.0
Tele High	0.0
Driver Ability	0.0
Driver Field Aware.	0.0
Driver Quickness	0.0
Can cross trench?	?
Tele Max Balls Low	0
Tele Max Balls High	0
Max Incap	0
Avg Incap Time	0.0

Version 1.3.18

Match Schedule

Our Match Schedule

Rankings

First Pickability

Second Pickability

Team List

REFRESH

Last Updated: 11/22/21 11:17:58 AM

Version 1.3.18

1	7419	841	1678C	52.0
Actual	1678	6657B	1671	149.0

2	6657	1678D	1678B	100.0
Actual	5817	5940B	2073	73.0

3	971	1662	973B	180.0
Actual	3859	4255B	3970	114.0

4	5104	5940	971B	145.0
Actual	1323	852	4255	118.0

5	973	841	1678	234.0
Actual	971	5817	5940B	125.0

6	4255B	1662	5940	51.0
Actual	1678D	5104	1323	80.0

7	2073	1678C	971B	161.0
Actual	1671	4255	6657B	49.0

Version 1.3.18

Rankings

TO PRED. RANKINGS

	Current Avg RPs	Current RPs	Pred. RPs	Pred. Rank
1	1678	3.00	36	24.0
2	1678B	2.42	29	23.0
3	1678C	2.08	25	22.0
4	1678D	2.08	25	16.0
5	1678E	2.00	24	25.0
6	1678F	1.92	23	13.0
7	1678G	1.83	22	30.0
8	1678H	1.58	19	22.0
9	1678I	1.50	18	23.0
10	1678J	1.42	17	8.0
11	1678K	1.42	17	24.0
12	1678L	1.33	16	16.0
13	1678M	1.25	15	8.0

Version 1.3.18

Team List

Team Number	Team Name
1323	Madtown Robotics
1662	Raptor Force Engineering
1671	Buchanan Bird Brains
1678	Citrus Circuits
1678B	Citrus Banana Bots
1678C	Jack in the Lime
1678D	Citrus Tide
2073	EagleForce
3859	Wolfpack Robotics
3970	Duncan Dynamics
4255	RoboDores
4255B	RoboDores



PICKLIST EDITOR

The Picklist editor spreadsheet is used during our picklist meeting to view data on teams and then rank them in our first pick or second pick lists.

List of Features:

- Display team's data in their row based on a CSV export
- Automatically shuffle the order of teams and their data when their rank is edited
- View team photos
- Compare up to four teams' data with a bar graph

Rank	Order	Pick Ability	Driver Ability	Climbs	Tele: Balls High	Tele: Balls Low	Auto: Balls High	Auto: Balls Low	Time Insec	Climber Installation Time	Drivetrain	Drivetrain Motor Type	Drivetrain Motors	Climb Attempts	Climb Time	Solo Level Climbs	Park
1	1678	1	1000.0	-1009.0	7	999.0	1000.0	1000.0	1000.0	0	0 tank	falcon	4	1000	999	1000	0
2	9036	2	58.9	-1.7	7	9.2	0.0	3.4	0.0	6	1 tank	falcon	4	9	16	3	7
3	9084	3	2.1	1.2	0	0.0	0.0	0.0	0.0	2	0 tank	falcon	4	10	0	0	6
4	9004	4	88.5	-1.7	2	8.4	0.0	0.8	0.0	0	2 tank	neo	6	5	17	1	7
5	9089	5	56.0	-0.4	7	0.0	0.0	0.0	0.0	0	1 tank	minicim	6	10	24	1	4
6	9078	6	71.9	0.5	0	0.6	0.0	0.0	0.0	1	1 tank	falcon	4	10	0	0	5
7	9120	7	70.5	0.5	3	3.9	0.0	2.0	0.0	0	1 tank	falcon	4	9	24	0	5
8	9229	8	38.9	0.4	7	0.7	0.2	1.3	0.0	4	1 tank	cim	4	11	21	0	9
9	9485	9	93.4	0.1	6	4.5	0.0	2.9	0.0	0	1 tank	falcon	4	11	18	0	1
10	9099	10	38.2	-0.6	1	5.8	0.0	2.0	0.0	6	1 tank	neo	4	9	14	0	6
11	9142	11	7.0	0.9	0	0.0	0.0	0.0	0.0	14	0 tank	neo	6	10	0	0	0
12	9277	12	30.7	0.9	0	3.5	0.0	2.4	0.0	13	2 tank	neo	2	10	0	0	7
13	9189	13	25.1	-0.3	8	8.8	0.0	4.6	0.0	5	3 tank	neo	4	9	21	3	6
14	9147	14	163.3	0.5	10	0.0	3.4	0.0	1.4	15	2 mecanum	neo	4	10	22	2	9
15	9275	15	128.4	-0.4	4	0.0	2.0	0.0	1.0	7	1 tank	neo	6	11	14	0	3
16	9242	16	145.2	-0.2	2	0.0	0.0	0.0	0.0	7	1 tank	falcon	4	9	25	0	9
17	9218	17	200.3	0.9	4	0.0	3.2	0.0	0.6	12	1 tank	neo	4	10	21	2	8
18	9544	18	200.6	-1.2	4	3.8	0.1	1.3	0.0	9	1 tank	neo	4	9	19	0	1
19	9548	19	74.1	1.3	0	0.7	0.0	0.0	0.0	18	1 swerve	cim	8	10	0	0	6
20	9339	20	103.4	-0.6	0	12.5	0.0	1.9	0.0	0	1 tank	neo	4	11	0	0	5
21	9269	21	3.3	-1.1	8	9.8	0.0	2.4	0.0	0	3 tank	neo	4	10	17	0	9
22	9283	22	130.3	-0.7	0	0.0	11.3	0.0	2.5	0	2 tank	falcon	4	11	0	0	5
23	9482	23	104.4	1.6	0	0.1	0.0	0.0	0.0	16	1 tank	falcon	4	9	0	0	3
24	9462	24	106.9	0.1	0	0.1	0.0	1.5	0.0	6	1 tank	neo	4	11	0	0	1



CITRUS

SERVER

The server compiles and processes all of the data collected in the system. Data is gathered by downloading JSON files from Pit Collection, scanning QRs from Match Collection, and pulling data from The Blue Alliance (TBA). Calculations are performed on the raw data and all the data is sent to our cloud database to be pulled by Cardinal and displayed in the Viewer. The server also contains unit tests and scripts for logging errors, starting the server, downloading QRs, and sending app APKs to the tablets.

Main Calculations:

Team in Match Data - By taking a weighted average of the values reported by three scouts per robot, we calculate approximately how many balls each team scored in each of their matches, split by auto/teleop and low/outer/inner goal. We also find whether the team achieved position or rotation control and calculate the amount of time it took the robot to climb, and the amount of time it spent incap. Using data from TBA, we find if each robot crossed the auto line, if it successfully climbed, and if the climb was level.

Team Data - This includes averages of all the Team in Match (TIM) data, such as the average number of balls a team scores in the outer goal in auto. We also calculate the standard deviations of each team's TIM data (how widely spread the values are for each data point across all their matches), and total counts of the number of climb attempts and climb successes. New to the 2021 offseason, we also store the maximum number of balls scored in each category in a single match, and the maximum time a team spent incap in a single match. Finally, we calculate a team's subjective driver ability score using the ranked lists from subjective match scouts.

Inner Goals Regression - Using score breakdowns from TBA and high goal data collected by our scouts, we use a regression to calculate the approximate percent of high goals scored



by each team that made it into the inner goal. More detailed information can be found in our 2020 whitepaper.

Predictions - There are two types of predicted data, predicted match data and predicted team data. Predicted match data includes the predicted score and RPs for each alliance. Predicted team is calculated based on predicted match values and includes a team's predicted RPs and rank. We also pull the actual scores, RPs, and rankings from TBA when they are available.

Pickability - Pickability is a value used to presort our picklist before human strategists make edits. It is a weighted sum of the data points our strategists have determined to be most important in an alliance partner. We calculate these both for the first choice and second choice in the draft as an alliance captain for each team, with first pickability placing more emphasis on ball scoring and control panel, and second pickability mainly focusing on driver ability. Both pickability calculations strongly value climb success rate.

CARDINAL

Cardinal is a Django REST API that was designed to fix a system architecture issue (Django REST framework is a commonly used tool for building web APIs with python). The initial idea for communication between Server and Viewer was that both would talk to our MongoDB cloud cluster individually to send data from one to another. However, Viewer was unable to access the data from the cloud MongoDB database, so it was proposed that an API be used in place of the direct connection to the database.

Cardinal is primarily a communication tool that coordinates the flow of our data, but it also provides features like automatically generated realistic test data for Viewer by adding the URL parameter “?test”. This allows Viewer to simulate competitions to fully test its features. In addition to competition data, Cardinal stores and distributes the “teams list” and “match schedule” files to the viewer. Cardinal allows for systems testing, benchmarking, and coherence between the database and the Viewer.



The production version of Cardinal is hosted on a mentor-provided server to allow 100% uptime and fast speeds for data transfer.

List of Features:

- **Distributes static JSON Viewer files**
- **Host updated data collections from MongoDB Cloud Database**
- **Automatically generate test data**
- **Collection name and schema verification** - check internal schema and collection names to validate up-to-date schema

Cardinal 1.0.0 OAS3

</cardinal/api/openapi-schema>

Cardinal

cardinal ^

GET	/cardinal/api/hello/	▼
GET	/cardinal/api/version/	▼
GET	/cardinal/api/collection/{collection_name}/	▼
GET	/cardinal/api/supported-collections/	▼
GET	/cardinal/api/generate/{data_structure_type}/	▼
GET	/cardinal/api/match-schedule/{comp_code}/	▼
GET	/cardinal/api/teams-list/{comp_code}/	▼
GET	/cardinal/api/log-file/	▼



CONCLUSION

Going forward, Software Scouting plans to continue creating a consistent and reliable system. In the apps, we hope to add JSON schemas, unit and UI tests, and use more abstraction and inheritance. Specifically for the Viewer, we also aim to add trend graphs for teams throughout their matches, comparison graphs between teams, update using changes rather than pulling all of the data each time, and to have photos pulled from Cardinal (rather than manually downloaded from Pit Collection). The Server plans to look into the accuracy of our predicted data by comparing it to the actual values, increase test coverage, and find a way to judge the accuracy of our scouts.

If you are interested in learning more about how we train our software members, please see [our Training Guide](#).

If you have any questions about our system or making your own, feel free to email us at the address below or find us at competition! softwarescouting@citruscircuits.org

