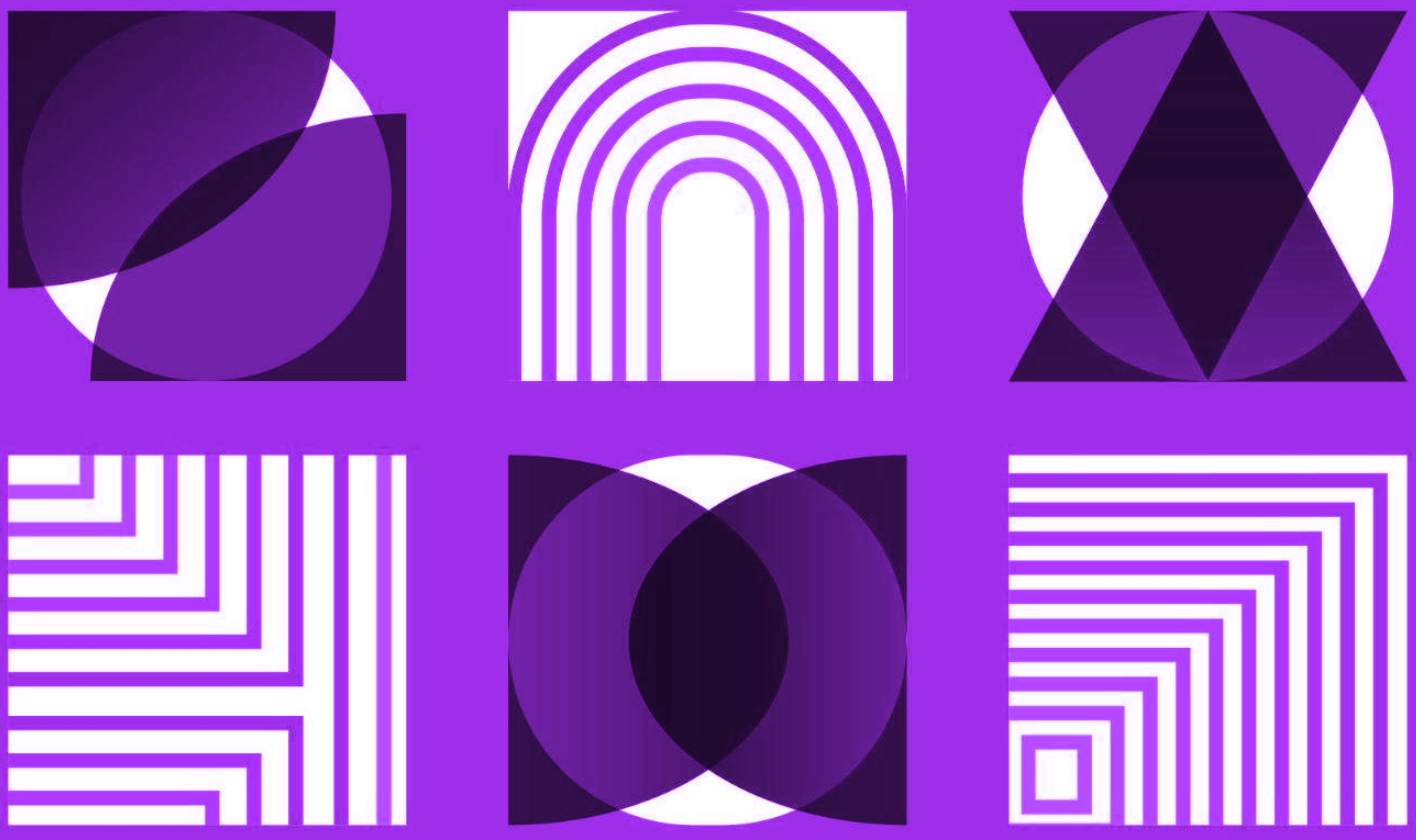
2024

THE PURPLE STANDARD

Release for Feedback



Contents

Version: Jan 10, 2024

What is The Purple Standard?

The Purple Standard (TPS) is a unified, community-driven standard for FRC scouting data. After working with 140+ teams who scouted collaboratively with us on The Purple Warehouse (TPW) last year, we realized that many of you still couldn't enjoy the benefits of collaborative scouting because of different scouting interfaces that collect data using incompatible formats. We were able to work with several teams to manually write reformatting scripts, converting their data to our app's format so that they could contribute to and access TPW's pool of shared scouting data. From our experiences this past season working with many different incompatible scouting data formats, we want to propose a better solution.

The goal of TPS is to provide a flexible and consistent data format for scouting to allow any team to easily share scouting data regardless of which interface is used. We hope this will strengthen the FRC scouting ecosystem by allowing more teams to share and receive scouting data which can be corroborated to provide more accurate analysis and insights while requiring less reliance (and pressure) on individual teams. In 2023, we saw that collaborative scouting through TPW significantly helped smaller teams and rookies who weren't able to scout as many matches individually but could still access a fuller picture of the overall event's scouting data by working together.

How Does Collaborative Scouting Work?

The Purple Warehouse is our scouting app and analyzer which allows any team to scout through the TPW interface and then access + analyze both their own data and the data of other teams who use TPW. To prevent freeloading, teams get increasing access to data from other teams as their own contributions increase. We were also able to implement an accuracy score for each scouting entry, enabling better strategy and analysis tools. With TPS, we want to bring the magic of collaborative scouting to every scouting interface, making it easy for you to share and receive scouting data through the TPW shared database.

TPW Data Format

To encourage maximal interoperability with TPW's existing collaborative scouting tools and analysis features, we've published the data format that we currently use in TPW (we may add some additional properties throughout the season as we improve our app). We highly recommend that teams build off of this format, adding more properties based on the data they want to collect. You can read more about how to customize TPS to your own needs in the Data Framework section.

Release for Feedback

We are releasing this first draft of The Purple Standard to get the feedback of the community in the next few weeks so that we can finalize the standard, leaving enough time for app developers and teams to fully adopt TPS. Please share your thoughts and propose any changes/additions for TPS either by replying to our Chief Delphi thread, sending suggestions on our Discord server, or opening issues/PRs on the TPS Github repo so that we can ensure TPS supports the needs of every team!

TPS for App Developers

If you are building a scouting app, please reach out so we can learn more about how to best help you with TPW/TPS integrations! In the coming weeks, we will be releasing more information about our APIs which will help you utilize features of TPS and TPW in your own apps.

Data Framework

To allow teams to use custom data properties and to provide a pathway for compatibility in future years, we are releasing a framework for scouting data which can be applied to any game and any attribute of robots.

Design Philosophy

The TPS framework has 6 interfaces (data categories) which can include any data you could ever want to scout. If there is some data that your team wants to scout that is not yet included in the TPS GitHub repo, you can add a property to the appropriate interface by opening a pull request. TPW and other scouting apps will be able to recognize the new property if they choose to add support for it (or they can simply discard it if the property is not used in their app).

With this framework, TPS has complete customizability of its properties, allowing future seasons to remain TPS-compatible and empowering every team who adopts TPS to recognize a shared set of properties to analyze.

Data Categories

- `Abilities`
  - An ability represents a singular action or state that a robot can take or be in during the match
- `Counters`
  - A counter represents a repeated action that a robot can take during the match
- `Data`
  - This is the default interface and can be used for any type of data that does not match any other interface
- `Ratings`
  - A rating represents a subjective evaluation of actions that a robot can take during the match
- `Timers`
  - A timer represents the duration of time spent on an action that a robot can take during the match
- `Metadata`
  - This is the metadata interface and can be used for any type of data that does not directly relate to actions performed by a robot during a match

References & Sample Data

On the TPS GitHub repo, we have a quick reference containing summary tables of all interfaces and properties, with links to view additional information such as example values and design recommendations. We also have a folder of sample data, including examples based on properties that TPW currently plans to collect.

How to Contribute

The TPS interfaces and properties are listed in our Github repo, and you can open a pull request to suggest changes to an existing property or propose a new one (you can reference an existing property to see how to consistently format your new property).

Security Considerations

Because TPS enables arbitrary input fields to be shared and displayed, often on web-based scouting/analysis apps, there is a heightened risk for XSS attacks (learn more here, here, and here). For this reason, we highly recommend that apps take measures to sanitize TPS data if they plan on displaying data from other teams on a web interface. In particular, web-based apps should replace HTML tags (< and >) with equivalent character entities (< and >) when displaying data. If you have any further questions about security, we are happy to provide additional guidance and suggestions to minimize risk for your particular implementations.

Attributions

- Cover Page Design from TheGoodocs
- This document was written by Omkar Govil-Nair and Kabir Ramzan on the TPW Team, a part of Harker Robotics team 1072.