Basic Information

Team Number

1444

Team Name

Lightning

Programming Language

What programming language do you use?

Java/Kotlin

Public Code

Is your code public?

Yes

What is your team's GitHub account?

https://github.com/frc1444

Vision

What camera do you use?

PS3 Eye

What do you like about your camera?

It's very cheap and is a reliable camera to use for vision processing

What do you dislike about your camera?

It has the ability to adjust the zoom via a slider, so if that gets accidentally touched it screws up vision

How are you planning to do vision next year?

We don't know yet

Path Planning

How do you design your paths?

We don't do path planning, we just our own version of something similar to the command framework. This makes our robot less smooth, but with because we use swerve the time difference is minimal and we can still get some pretty complex movements in (moving and rotating at the same time)

How do you integrate sensors with your paths?

Since we don't use paths, integrating sensors was done at a higher level of abstraction (just gyros, and wheel encoders)

This year, we used absolute positioning making it very easy to reset our position while in autonomous. However, we decided against using our vision to reset absolute position in autonomous because if the vision was even a little off, it would throw off the actual driving of the autonomous. This mostly only happened while rotating. We're still thinking about the best way to integrate vision data into autonomous more accurately.

Training

How do new programmers get trained?

Our team has had difficulty training new programmers this year and in the past. This year we had our new programmers program the old robot.

GitHub

How do you control access to the team GitHub?

Usually we only have one or two people working on the robot code, so the github is available to both programmers and they have to work out when to commit to master and when to put their code on their own branches.

How do you handle merge issues and multiple people working on the same file with GitHub?

Merge the code into master every once in a while and deal with merge conflicts then, ideally when both programmers are next to each other.

How does your team make ReadME.md documents?

This year I made lots of different documents with lots of random documentation. Usually the main README file has instructions on how to deploy and any quirks for the robot code for that current year.

Other Sensors

What other types of sensors do you use?

This year we used three sensors to detect a ball in front of it.

How do these sensors help your robot?

This helped us stop jamming and allowed us to get balls ready to shoot as quick as possible

Documentation

How do you document your code?

Comments when necessary, javadocs on classes and methods and sometimes we'll put a package-info.java in.

Creating from Scratch vs Inheritance

How does your team balance inheriting WPILib functions with writing custom functions?

This year we did all custom stuff. If we have the time, we'll likely keep our custom code up to date, if not, we'll probably end up using lots of WPILib functions.

What are some examples of custom functions that your team has made?

Our robo-sim library that we used for the first time this year abstracts most of WPILib away from the main robot code and allows a graphical simulation for autonomous and teleop practice.

Joystick Layout

Who determines the layout of the joystick for your team?

Usually the main programmer with input from the driver

How do you test the joystick layout?

Usually in the simulation or on a test robot with fake debug outputs.

PID Tuning

When you get the robot, what is the first thing your programming team does with it?

Get the basics driving

How does your team determine if motors should have encoders or not?
If something needs to go at a certain velocity or stop in the middle, it probably needs encoders

When you PID tune a motor for position control, what is your procedure?
P up until oscillates, then depending on application, add I or D.

When you PID tune a motor for velocity control, what is your procedure?
Add some F, then add P to compensate