Basic Information

Team Number
4681

Team Name
Murphy's Law

Programming Language

What programming language do you use?
Java

Public Code

Is your code public?
Yes

What is your team's GitHub account?
amhsrobotics4681

Vision

What camera do you use?
Limelight 2

What do you like about your camera?
We just got it and started experimenting. It truly is plug and play, and we were able to flawlessly develop our aiming system. Autonomous will be greatly improved.

What do you dislike about your camera?
Figuring out contrast, hues, saturations, lighting, etc. is not included in the instruction manual. We test in a room with windows and lights/windows are giving false positives. When we erase those, the target becomes limited and the robot is shaky while aligning.

How would you compare this camera to other cameras you've used in the past?
We tried to use a PiCam on an rPi4, but we couldn't get the networking to work.

If you could do vision differently, what would you change?
We aren't experienced enough with vision to know how to improve it.

How are you planning to do vision next year?
Same thing: Limelight 2. Same code, same everything.

Path Planning

How do you design your paths?
Through trial and error, we have a robot that drives for 'x' amount of seconds through a periodic timer in the autonomous. If we need to turn, we just figured out how to use a gyro this year.

What forms of path planning have you done in the past and if you have changed them, why?
In the past, autonomous was just "drive forward for 5 seconds and don't bump the robot next to you." Once younger (and therefore, more creative) members cycled through the programming, we tried new things. In 2018, we drove forward, turned (!), and dropped a cube on the small

scale. This year, we turned (!) and drove forward to collect balls (!). We were not able to test turning back around and aiming with our new vision system because of corona.

How do you integrate sensors with your paths?
A gyro on the roboRIO. We turn the robot until we reach a certain angle, then drive.

If you could do path planning differently, what would you change?
Probably learn how to do actual path planning, but that requires a summer workshop on encoders that the team is not ready for.

Training

How do new programmers get trained?
This is the first year we've actually had a sort of training. In the autumn, I spent a Saturday going through libraries and what they do on the robot. The next Saturday, we pulled out an old tank drive robot and I told them, "Program it." And with a bit of a walkthrough, they got the hang of it. I trust them to be able to help modify the code in competition. In fact, one caught an error in my autonomous code! Good on them.
When it comes to kick-off, I wanted to show the youngsters how to get the NI stuff and update WPILib, but our coach forced them onto hardware. That's fine, I'll try to stick around as a mentor next year.

Do new programmers have to work outside of practice?
No. Robotics is not a class; therefore, we have no homework.

What is your general training order? (ex. Classes, functions, data types, reading documentation)
They had prior programming experience, but it was basic. They're smart though. Anyways...
1) Robot modes (auto, teleop, etc.)
2) WPILIB Libraries and New Robot Project
3) ... then using joysticks to control robots
4) Different Classes for Different Mechanisms
5) Control Loops with Switches and Timers
6) Autonomous Code

What do you do if there is not enough work for all the programmers?
In our case, there's not enough programmers for all the work.

On average, how many programmers do you have?
2017: 1 main, 1 side, 2 training*
2018: 1 main, 1 training*
2019: 2 main, 2 side*
2020: 2 main*, 3 training
(asterisk denotes where I was)

On average (weighted mean), 2 programmers and one helping.

GitHub

How do you control access to the team GitHub?
We don't encourage remote access (except for lead programmers, because those in power set the rules).

How do you delegate using GitHub?
Everyone can view code. Again, only leads can change stuff.

How do you handle merge issues and multiple people working on the same file with GitHub?
With only at school access (or two programmers from home on different mechanisms) we rarely have merge issues.

If your team uses private repositories: what are the advantages of this/why did you start doing it?
N/A

How does your team make ReadME.md documents?
On GitHub.com, switching between preview, markdown, and a help document.

Other Sensors
What other types of sensors do you use?
Limit switches, LIDAR, infrared, a gyro, cameras, a Limelight, and an encoder on a gearbox.

How do these sensors help your robot?
To detect when we're touching something to run pneumatics or a motor or to stop an elevator, to get distance to the wall, to get reflective tape on the ground, to turn, to see, to align, and to move something precise relative distances.

Of those sensors, which are you planning to use again in the future (if any)?
All of them

How do you learn what new sensors to try and how to use them?
When we have the budget and the time, we buy them. For example, I wasn't ready to learn how to use encoders, but hardware depended on it, so we figured it out. Our head electrician was hesitant to create advanced autonomous, but I vetoed that (because who's the programmer?) and our new autonomous was successful. Almost a six ball auto! We learn from the internet, from other teams' experimenting and all that.

Off Season
What do you do in the offseason to prepare for build season?
Practice bot, and dismantling previous year's structures as an intro to tools.

How does programming interact with mechanical for off season activities?
Not at all (yet, because there will be changes next year).

Documentation
How do you document your code?
GitHub commit comments and comments in the code, and a journal for preseason. (:-D) I know, it's bad.

Have you documented differently in the past? What do you like better now vs then?
Prior, there was barely any handing down of knowledge. We borrowed code from past years and hoped it would work. (:-D) I know, it's bad.

Build Season

What do your programmers do at the start of build season?
Update everything. VSCode, WPILib, NI stuff, Phoenix stuff now (because we're using Talons), etc. Then we help design. Then once the design is done, we build a code framework. Then we help build the robot until a mechanism needs to be tested. Then it's plug and play.

How useful are the tasks that they do at the start of build season? (from 1 - 10)
10

How much time does programming get to program the robot (without mechanical intervention)?
We can program whenever we want. We can only test the code in small 10 minute snippets, and if something breaks, they expect an instant fix.

How do you divide up the time programming gets on the robot between different mechanisms, tuning, and autonomous?
In that exact order: mechanisms until "stop build," tuning between "stop build" and competition, and autonomous at competition. Pretty much what happens.

During programming's time on the robot, how does your team handle mechanical failures and imperfections?
We play the blame game. It's not helpful.

How do you make the schedule for programming?
Schedule? What schedule? Why?

How does your team use gearbox ratios with encoder counts?
N/A

How does your team define code standards?
Just look at what other people have done and follow established conventions.

Creating from Scratch vs Inheritance
How does your team balance inheriting WPILib functions with writing custom functions?
We don't write custom. We don't have the luxury of time to do that.

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

Interesting WPILib Functions
Are there any WPILib functions that are unusual and make your life a lot cooler or easier?
N/A

What class do you use for joystick control?
Joystick

What class do you use for automating actions?
We're not that advanced (N/A).

Joystick Layout
Who determines the layout of the joystick for your team?

Collaboration among the drive team (so it's less programmer focused).

How do you manage changes to the joystick layout?
We change it in the code because we have a public constants class for our use, and then we hold driver training.

How do you test the joystick layout?
We upload the code to the robot and hold driver training.

PID Tuning

When you get the robot, what is the first thing your programming team does with it?
Test our code. We write it while the robot is being built so we don't fall behind. That's why design in Week 1 is important. It establishes the basic motors of the robot.

How does your team determine if motors should have encoders or not?
First year using encoders, so we only used it in one place: turret angling, which requires precision.

When you PID tune a motor for position control, what is your procedure?
I don't know. I didn't code this feature.

When you PID tune a motor for velocity control, what is your procedure?
N/A