

Basic Information

Team Number

6738

Team Name

Excalibur

Programming Language

What programming language do you use?

Java, thinking of moving to Kotlin for next year

Public Code

Is your code public?

Yes

What is your team's GitHub account?

ExcaliburFRC

Vision

What camera do you use?

Limelight 2

What do you like about your camera?

That it's pretty much plug-n-play...

How would you compare this camera to other cameras you've used in the past?

It works without problems:

In 2019 we used an RPi with a webcam, and in the competition we weren't able to network it correctly, leading to us not having vision that season (luckily for us, there was no auto period in 2019...)

How are you planning to do vision next year?

Limelight 2

Path Planning

What library(ies) do you use to make your paths?

We would have used the wpilib trajectory lib (if we had time to work on path following), but for next year we will probably use pathweaver.

What forms of path planning have you done in the past and if you have changed them, why?

We didn't do pathfinding, we plan to next year.

If you could do path planning differently, what would you change?

Actually do it.

Training

How do new programmers get trained?

A training seminar of the team.

Do new programmers have to work outside of practice?

While learning the language, there are some "home assignments", but little to no work is done at home for the robot.

What is your general training order? (ex. Classes, functions, data types, reading documentation)

Basic syntax of the language

Functions

Classes

Inheritance/Polymorphism

Focus on FRC Programming

What do you do if there is not enough work for all the programmers?

Split the work, if programmers are idle they help other subteams.

On average, how many programmers do you have?

In 2019 we were only two programmers, now we have around 7 , most of us don't do much though. It will change for next year.

GitHub

How do you control access to the team GitHub?

Only one uploads to github

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

Only one person uploads to github, will probably change for next year.

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

To prevent mid-season code from being public.

Other Sensors

What other types of sensors do you use?

Encoder, ultrasonic, IR distance, limitswitch, gyro

How do these sensors help your robot?

Gyro, drivetrain encoders -> control drive motion

Other encoders -> control position/velocity of the mechanism

Other sensors -> check if a ball is in the mechanism.

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

All of them, a few more.

How do you learn what new sensors to try and how to use them?

Internet/ChiefDelphi search about the sensor, trying to decipher the user manual of the sensor (if there is one)

Off Season

What do you do in the offseason to prepare for build season?

Try out new sensors/motors/advanced programming that we want to use in the following season.

How does programming interact with mechanical for off season activities?

Mechanical builds a robot, we program it.

Documentation

How do you document your code?

Javadoc, line comments on complex parts.

Have you documented differently in the past? What do you like better now vs then?

We used to not document, now someone that less programs can understand how to use the code..

Build Season

What do your programmers do at the start of build season?

After the whole team decides on the properties of the robot, we program the subsystem classes and some commands

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

8

How much time does programming get to program the robot (without mechanical intervention)?

In 2020 we barely got two days (we were week 1) , because of immense mechanical problems

How do you divide up the time programming gets on the robot between different mechanisms, tuning, and autonomous?

Start with making sure all mechanisms work, then tune them, then work on auto.

During programming's time on the robot, how does your team handle mechanical failures and imperfections?

If we can make a workaround in the code we try to do it, otherwise mechanical takes and fixes the robot.

How do you make the schedule for programming?

Start and see where we get.

How does your team use gearbox ratios with encoder counts?

If we use an internal NEO encoder, there isn't much to do, if not we use a CTRE Mag encoder gearbox stage.

How does your team define code standards?

By the basic standard of the language, with some adjustments the sub team agrees on.

Creating from Scratch vs Inheritance

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

If we know of a WPILib (or other lib) function that works well and does what we want, we use it. Otherwise, we write it ourselves.

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

Balancing two flywheels, voltage compensation.

Interesting WPILib Functions

What class do you use for joystick control?

The WPILib Joystick class.

If we want to bind a command to a button we use a JoystickButton object, otherwise we use joystick.getRawAxis/Button.

Will probably change if we move to Kotlin.

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

The drivers and programming.

How do you manage changes to the joystick layout?

We have a joystick binds data sheet.

PID Tuning**How does your team determine if motors should have encoders or not?**

If we need to control the velocity or position of a mechanism, it needs an encoder. If we don't need extreme position control we use a CTRE Mag Encoder, otherwise we use the internal NEO encoder (if we use a neo for that mechanism)

When you PID tune a motor for position control, what is your procedure?

If it's a mechanism that opposes gravity, tune arbitrary feed forward.

Calculate a starting kP by finding the error/change ratio.

Then balance kP/ki/kD/AFF by the results of the first two stages.