

Programming Struggles and Challenges

Overview: This document will describe all the programming challenges, struggles, and issues that we encountered during this Tobor programming season. It will not only cover the difficulties I faced but also outline the solutions that were found to overcome them.

Overall Biggest Challenge: I recently had a new experience where I had to learn and solve problems independently as the only programmer for Tobor. While I could still take input from the team, I had to approach problem-solving from a different standpoint. Although I enjoy figuring things out, being the only returning programmer presented a challenge as I had to solve problems alone. If I were to identify my biggest struggle, it would be the new experience of programming independently this year.

1. SDK issues

- 1.1. Android Studio does not allow me to create a Java class under team code.
- 1.2. Android Studio does not pre-write my package name along with the head class for the project.

Issue 1 resolved: *opening the SDK was done incorrectly, try opening the SDK by opening "FtcRobotController-master," this project should have the green android mascot on the file, this shows that the file is good to open. If any further errors are held with the SDK try redownloading it and reopening it.*

2. Organizing the code

2.1. Using both LinearOpMode and OpMode in one Java class

Struggle 2.1 resolved: *this was a struggle at first, but I later resolved this struggle by extending LinearOpMode and creating an interface with multiple required methods that I can call within the runOpMode method. This way I am using LinearOpMode to make the code layout look like both LinearOpMode and OpMode.*

2.2. Can not call dcMotor, telemetry, hardwareMap, or any other hardware without extending a class.

Issue 2.2 resolved: *It took me a little while to figure this one out, but finally, I found out that if I wanted to use hardware Map in a class that had no extension, I could create a method that has a 'HardwareMap' type inside, like this: public static void mapMotors (HardwareMap mapHardware, String frontLeftMotor, String frontRightMotor, String backLeftMotor, String backRightMotor). This way when I call the hardware map from the Java class that does extend, I can call it like so: mapMotors (hardwareMap, "fr", "fl", "br", "bl");.*

- 2.3. Uploading the entire SDK team code to Git Hub cannot be done, because the file is too big.

Issue 2.3 resolved: Uploading big files such as the edited SDK team code to your GitHub repository cannot be done by simply dragging the files. It must be done from Android Studio itself. To resolve this problem, I looked up how to upload code from Android Studio to Git Hub. Before using Android Studio to upload code to GitHub, git must be installed on your computer, and then you can push your code to your GitHub repository. When managing code in GitHub, I found it is best to use Android Studio and Git to push, pull, clone, comment, manage version control, and more! I have not found out all about these helpful tools, but I will investigate learning more about GitHub and Git.

- 2.4. Allowing other GitHub users to edit code.

Struggle 2.4 resolved: this was a bit of a struggle at first, but I found out that to do this you have to go to permissions and select what account you wish to allow to edit the files.

- 2.5. Managing branches

Struggle 2.5: At first this seemed to be a struggle for me, but later I realized the problem was with the GitHub account I was using. I was using my personal GitHub account when the Tobor account was the one that created the repository. To rename or edit the default branch I had to be using the Tobor account, the account that created the repository. With my account, I can still edit and delete sub-branches, but just not the default branch. I need to use the team account to do that.

3. Test robot issues

- 3.1. Non-responding encoders

Issue 3.1 resolved: *this issue was rather ridiculous, I kept trying to run the test robot encoders but later I realized that the encoders were not plugged in.*

- 3.2. The test robot slowly moves leftward when told to move forward, it moves forward for a little while but slowly moves leftward after a while.

Issue 3.2 resolved: *mechanical fixes will not be performed since the test bot is not what the team is using for this season's competition, however possible programming fixes could be obtained. For example, if the robot is slowly moving left, maybe put more power on the left wheels to make the robot turn right again.*

4. Mecanum Drive

- 4.1. Mecanum Drive is not working properly at all.

Issue 4.1 resolved: *definitions in the code were not set correctly, I told the code to set the front left wheel to the front right power and set the back left wheel to the back right power. I changed that and the robot worked a lot better.*

- 4.2. Mecanum Drive program strafing left when it should strafe right, cannot find any error with the code.

Issue 4.2 resolved: *To fix this issue I finally found where in my code I was accidentally calling the wrong motors. What I had going wrong was I had a variable of a wheelset to the wrong wheel somewhere in the code. If someone looks over all their code, the code is good, but everything is not working correctly still then take it back to the issue. If no error is showing on the driver's station, but the robot is not doing as expected it could be incorrectly configured. An incorrect configuration was also my issue in this case. The wheel direction should also be checked, but in my case, that was not the problem, the problem was I was calling the wrong motors.*

5. Working From Home

- 5.1. Setting up the Tobor Google email so that I can work with it from home. **Not yet resolved.**
- 5.2. Defining distance.

Challenge 5.2 planned: *I planned out this challenge what I will do is tell the robot how many rotations a specific number of inches is. Afterward, I will take all that information to see how many rotations are in an inch to the robot by doing a bit of simple math. Once this is done, I should be able to measure out all the dimensions of the entire Center Stage game field and tell the FTC robot to move accordingly while I am at home without even setting eyes on the fiscal robot, yet having confidence that the code will run efficiently. This concept is extremely helpful for my team as everyone needs the robot. This will make it easier for others on team Tobor to use the robot while I code.*

6. Team Prop and Pixel Detection

- 6.1. What are the best objects to scan when using AI?

Challenge Question 6.1: I found out from doing research that when using tensor flow AI, the best objects to use for a team prop are complicated. The more complicated the shape the better. If you can add complicated colors too that will be even better. This is because the AI needs something unique apart from the rest of the field to recognize what it is.

6.2. The AI is saying that the edge of the laptop is a pixel.

Issue 6.2: It seems that the AI technology is not meeting my accuracy expectations. This may be because the pixels are created with basic colors and shapes, limiting their capabilities. However, there is a solution to this problem - using team props can boost your move points and allow for more control over the game.

6.3. After training the team's team prop it detects good, but not as good as I hoped.

Issue 6.3: I have decided to switch to using a color/distance sensor instead of AI for detecting the team prop, as you believe that AI has its limitations. I plan to use a distance sensor to determine whether the robot sees the team prop and knows how to get it working efficiently. I believe that if the team can get points that way, then it's worth going for it. Another reason I decided to switch is that FIRST only gives a team a limited amount of training time, and I don't want to waste that time on training repeatedly if the AI has trouble detecting the team prop due to color issues. Last year, the AI didn't work well either, so this year I was hoping to get it working better, but it didn't turn out as perfect as I wanted. Therefore, I have decided to switch to the distance solution, which I believe will be more reliable than using a color sensor due to color variations and lighting conditions.

7. Robot Not Moving Straight!

7.1. TeleOp not moving straight.

Issue 7.1: This doesn't need high attention due to joysticks being able to straighten out the movements of the robot. So, I can fix this later, however it isn't that urgent.

7.2. Autonomous not moving straight.

Issue 7.2: This is more of an issue. If the robot cannot go straight in autonomous, the code can not work reliably. I fixed this by using a setVelocity command instead of using a setPower command. What this is doing is setting the speed of each motor instead of sending power or raw voltage. Sending the power can be unreliable, however, using speed is more consistent. I also will be using ticks with encoders and odometers.