D.S. Robot Designers

FTC 17191
Portfolio

Index

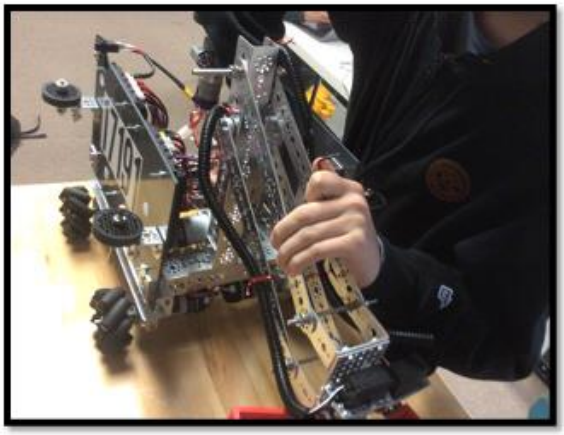
#17191

Meet the team

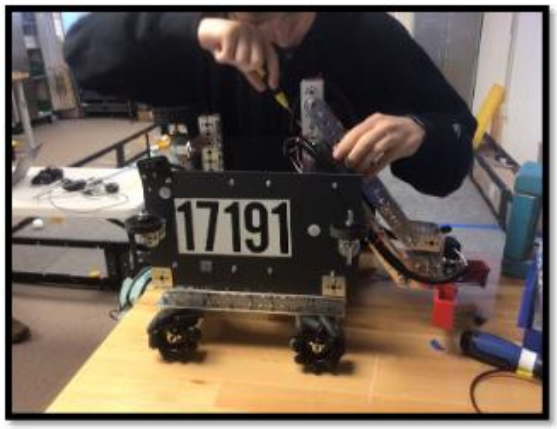
Collier, 2nd year FTC, builder

Luis, 1st year FTC, builder

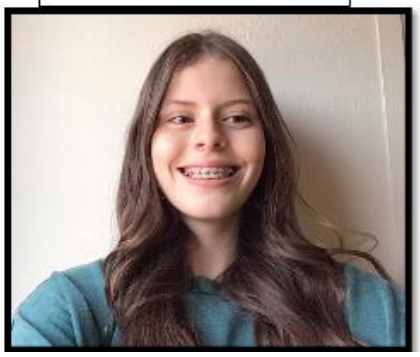
Tate, 2nd year FTC, programmer, builder



Dakota, 2nd year FTC, builder



Brianna, 2nd year FTC, driver



Miley, 2nd year FTC, notebook

Team Mission Statement

As a team our main goal is to inspire and have a successful and enjoyable robotics experience for all of our members. To achieve this goal, all of us strive to make this experience the best we have had in FTC. Our program is doing their best to inspire others. Gracious professionalism is also one of our key missions as a team. We strive to be the best people we can be in everything we do, especially in our robotics program.

Outreach

This year for outreach, our team started a coin drive for a local fundraiser called Snack Shak. It is a charity in which our team raises money over a span of a week to get toiletries for kids in need of them. Our team raised three-hundred and fifteen dollars over the week from sixth through eighth grade and the money was donated.

Later during our season we got together with an area FTC team. Our team showed them things that we had done on our robot and showed them things they may need to improve. The team had also spoken to them about using blender as CAD Software.

Sponsoring and Fundraising

This year for our fundraiser we sold braided breads and cookie dough to our community to raise money for our robot and our program. As a whole, our program was successful and raised about $3,000. As a team we raised about $1,000 which is a great achievement. Our goal was to each sell about 10 to 15 items.

Our program also depends on sponsorships in the form of financial and services. This year our program received sponsorships in the amount of $6,200 of which our team has access when needed. Additionally some businesses offer services such as helping us with CAD, recognizing sponsors in the local news paper, and ordering unique parts and supplies.

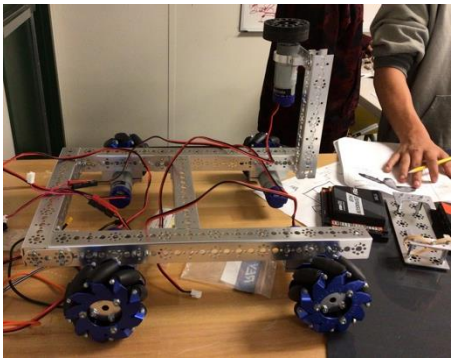
Sustainability Plan

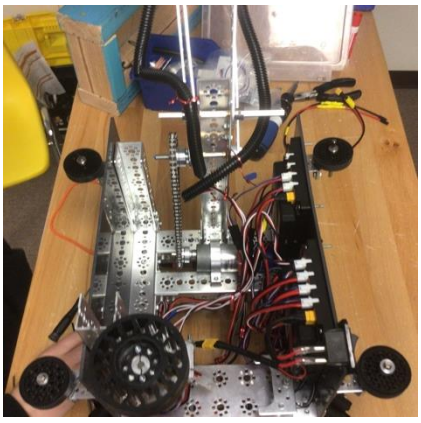
Our team realizes a sustainability plan is important for the growth of our team and overall program. Our team has developed a plan to recruit new members through an after-school program where our team members work with interested students to build and program simple mechanisms and robots using LEGO EV3. Through this plan, prospective students are able to see

if they have an interest in robotics. Our team will teach the students and give them an opportunity to learn new things.

Engineering

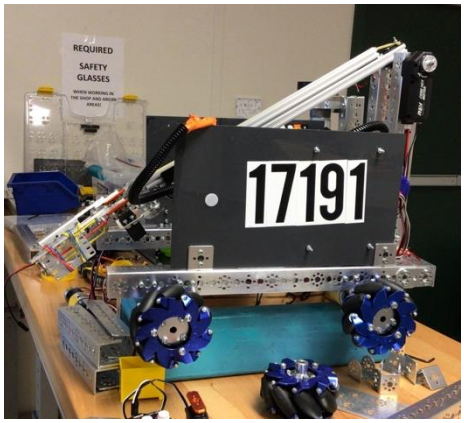
The original prototype was not planned out and was oversized very frustrating to build on. Also it was not very sturdy and flexed too much the frame also made us get stuck on the Barriers trying to enter the garage.

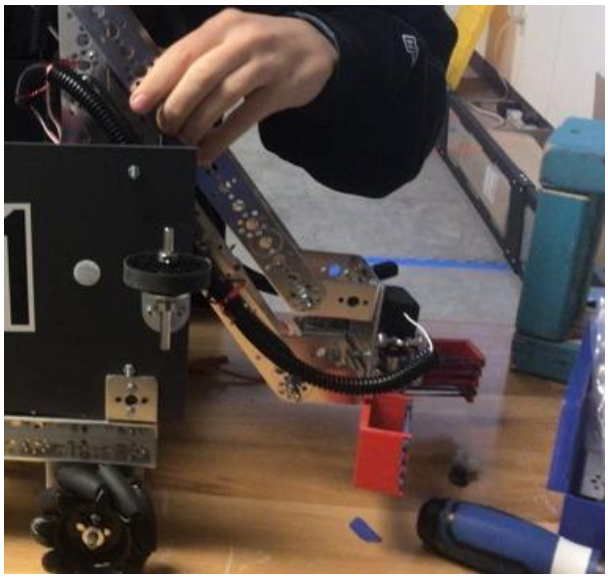




We had to disassemble the old frame and downsize our new frame to be able to squeeze through the side in hopes of not getting stuck again. With downsizing the frame it also got very sturdy and had little to no flex.

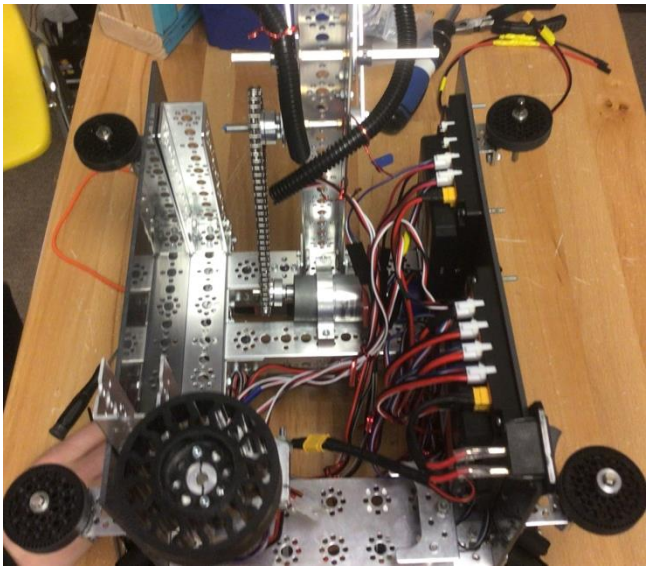
The original mechanism to get the scoring objects was a liner slide but with it being flimsy we would encounter many problems. We made a mistake of believing in it and it could not move up or down. The whole thing was a mistake it was another reason to disassemble the robot.





The new mechanism is a molding around the object. It uses rubber bands to rap around the object to get the best hold on it. The difference is it moves up and down. It worked when we needed it its small and was a smart move.

The side wheels help not to scrape the side of the wall to make sure we don't get penalty's. The concept was made to keep the wheels from getting damage its also protection from scraping penalty's.



Drive Control Strategy

Our team has gone to three league meets and we have worked together to drive and communicate as a team. We have had struggles through all of it but in our opinion the best thing is communicating and making sure we listen to each other. We like to go talk to the other teams and make sure we are on the same page and that we can work together to score more points.

Our team decided to use two remotes to control the robot. One of them controls the arm and claw to pick items up and score them, and spins the carousel. The other remote is the one that drives

the robot around. Our strategy was to make our robot small enough to go between the wall and the PVC pipes, so that instead of getting stuck on the PVC pipes. Our team made the claw where we could pick up the blocks, balls, and our team shipping element. Our strategy if for the robot to go between the wall and the PVC pipes to score on the shared hub.

End Game Strategy

As a team our strategy of endgame was to get the ducks on the carousel and to be able to squeeze very quickly through the barriers and park in the garage. To get as may points as possible in the last 30 seconds of the match. We could also go and continue scoring in our 3 layered hub. The most worth it would be to get as many ducks as possible then 8-5 seconds left drive into the garage to get those extra points. The two drivers have to be communicating together or they will both be confused communication is key for endgame they need to work together quickly and efficiently.

Programming

Over the year we have worked on 2 major things.

1. TeleOp
2. FTC Layer

TeleOp

We created a TeleOp with FTC Layer, that uses its abstractions, inside a Robot class. The Robot class holds objects that are connected to the hardware. So, we have an object for the arm and a object that controls the drive train.

We then made an object out of the Robot Class, which we then used basic if statements and boolean flip flops to control the robot. A lot of the code was made easier due to FTC Layer's abstractions over the SDK.

FTC Layer

FTC Layer is a Android library we have developed. It is abstraction layer over the bare FTC SDK. Kind of like OpenGL and SDL2. SDL2 removes all the unnecessary bits that are in OpenGL and makes graphics programming very easy. FTC Layer takes inspiration from SDL2 and how it abstracts and makes programming faster.

FTC Layer also takes inspiration from FTC Lib. FTC Lib, aims to make programming easier and less time consuming. FTC Layer isn't necessarily designed to making programming more beginner friendly, it is designed to make it less time consuming and simpler for more advanced users. However, when you aim to make programming less time consuming, it generally makes programming easier.

For example: Finding an algorithm for mecanum wheels was a big challenge last year, All (except Luis) of our rookie year. The entire mecanum code with FTC Layer takes 2 lines and look like the following

```
Mecanum train = new Mecanum(hardwareMap, "id", "id", "id", "id");

train.powerDrive(gamepad1.left_stick_x, gamepad1.left_stick_y, gamepad1.right_stick_x);
```

We also made extensions over common SDK classes, like GamepadEx, or MotorEx. We even have a class in their that takes a motor, and treats it like a servo. A lot of times we have issues with encoder ranges, and that's a good solution.

FTC Layer uses Jitpack to distribute new versions of FTC Layer, and make FTC Layer compatible with gradle.

We also set up a website for documenting FTC Layer with Cloud Flare Pages. The documentation site is a Docasaurus theme with FTC Layer branding, and we are going to contribute more documentation soon,. We have added a rule to FTC Layer that all classes must have documentation (that is classes that are public, utilities used in the backend don't have to be documented on the website, however it is good practice to leave a markdown file with functions and what they do.)

The collaboration to the FTC Layer project, is a great example of gracious professionalism.
It has the professional environment which projects prosper in, but the collaboration that makes a community and projects so great. We've had people who have contributed ideas, and a lot of the documentation site was set up by a contributor.

Challenges

Problem:

Our first iteration of our chassis was too big and too slow.

Solution:

We Redesigned our chassis to Smaller and faster.

Problem:

Using git sub modules for distributing FTC Layer is easy to mess up and not native to gradle

Solution:

Use Jitpack to distribute FTC Layer with gradle dependencies.