Engineering Portfolio

N.E.R.D. BOTS! TEAM #14353

FIRST TECH CHALLENGE

Fair Play is my...

ULTIMATE GOAL

PRESENTED BY Qualcomm

FIRST in Michigan

FIRST TECH CHALLENGE

FTC 14353

Introduction

We are the N.E.R.D. Bots (Nuclear Explosive Rubber Duckies), an FTC robotics team from East Middle School, from Canton, MI. Our team number is 14353. The N.E.R.D. Bots started in the 2018 Rover Ruckus season and this is our 3rd season.

We are very passionate about building, programming, and contributing to the community. Each week we hold meetings for leadership, where we organize fundraisers and outreach events, build, where we CAD and build our robot, and programming, where we program our robot.

This year due to the COVID-19 pandemic, we have been having online calls instead of meeting in person. Even though we haven't been meeting in person, we were still able to hold a fundraiser and a virtual outreach event. Also, we were able to CAD and assemble our robot, despite not being able to build in person for most of the season. We shared parts and assembled and tested them at home. We gave the programming team hardware modules to test and calibrate their code at home as well. Once all the prototyping testing was done, build met in small groups to complete the robot build, and then the competition robot was passed amongst the programming team to incorporate the snippets of code into a unified and tested complete code set.

We are incredibly proud to present to the judges our robot, "Quarantine", for this season and show the judges the work we have done in the past 9 months.

Robot Capabilities Summary

Mechanical:

- Fast ground ring intake
- Accurate high goal shooter
- Opposable thumb wobble goal arm
- Custom omni directional drive train with field centric controls
- Custom odometry pods

Sensors:

- Encoders for drive and shooting motors
- Encoders on odometry
- Built in IMU gyroscope
- Web camera for Vision

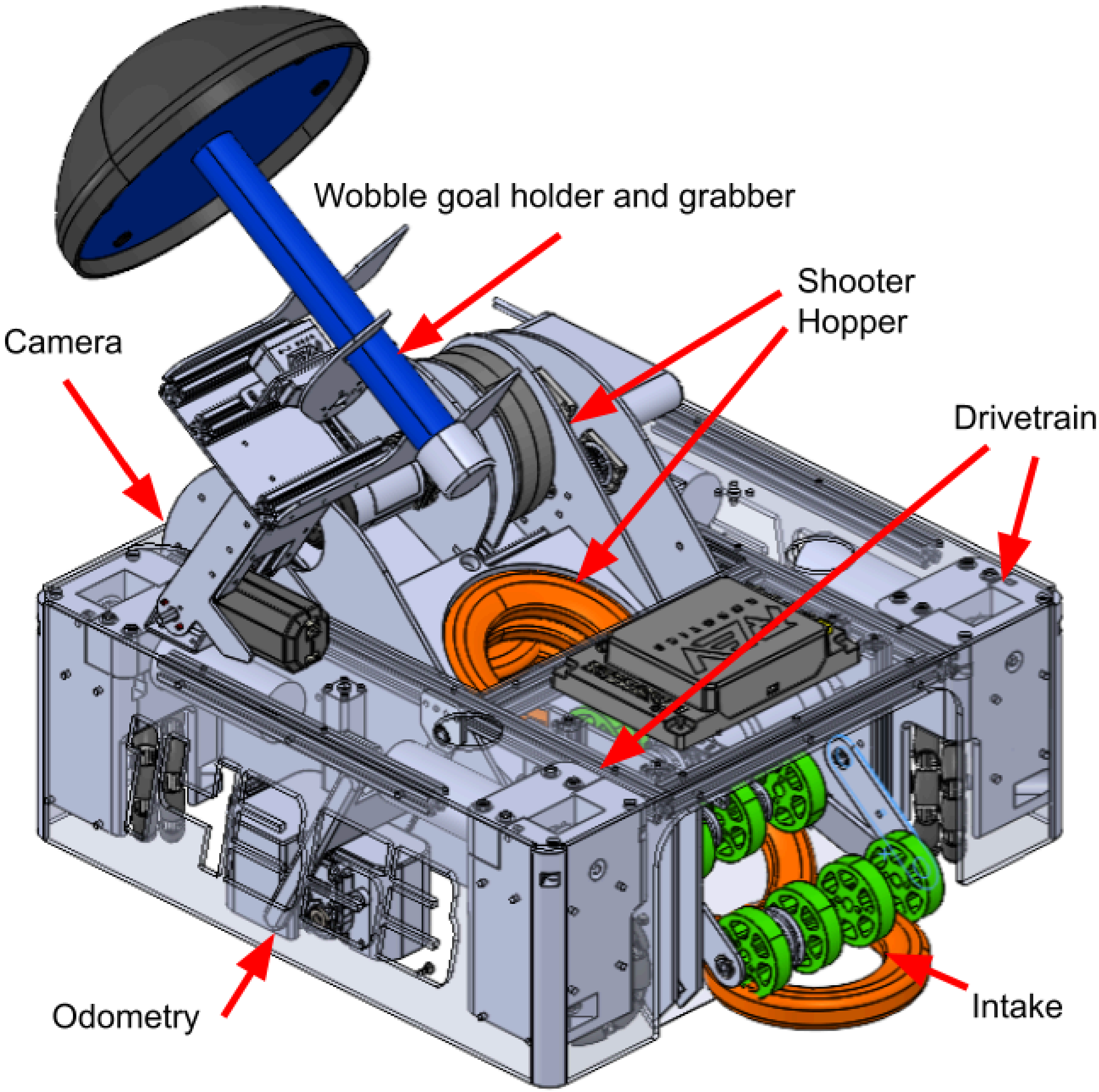
Autonomous:

- Vision detecting ring randomization
- Deliver both wobble goals
- Shoots pre-loaded rings into high goal
- Intakes additional rings on the field and shoots into the high goal
- Parks on the line

Programming:

- Pure Pursuit
- Custom velocity following PID
- Field centric driving and odometry
- Custom PIDF algorithm
- Custom OpenCV and TensorFlow models

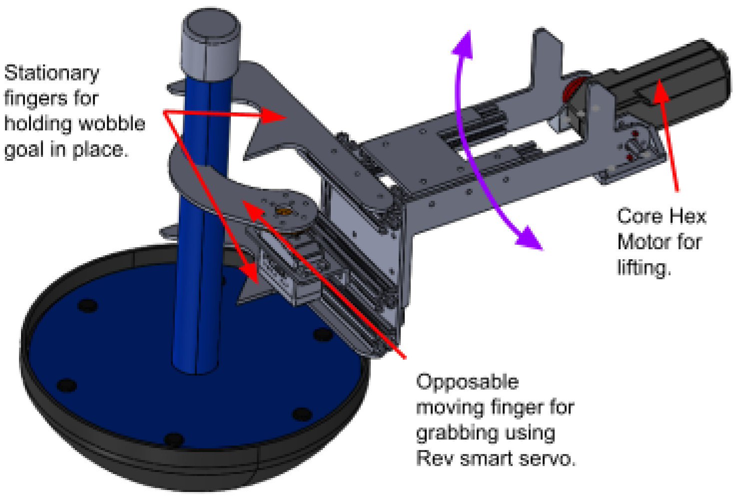
Build and Design - Current Robot Design



Our robot consists of **FIVE** main parts. There is the wobble goal holder and grabber; the intake and conveyor; the shooter, indexer, and hopper; the drivetrain; and the odometry pods. All of these parts are connected to a central REV control hub and expansion hub. We use a REV battery to power our robot.

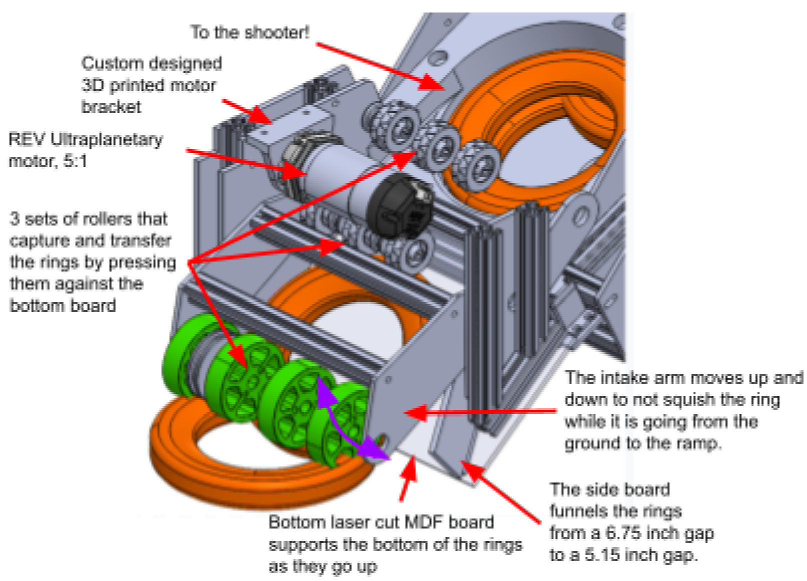
Wobble Goal Arm Design

Our wobble goal holder and grabber uses a REV core hex motor and servo to grab and lift the wobble goal. We used CAD to make different designs and we used a laser cutter to cut the CAD design on a piece of MDF board. Then we used aluminum extrusions to connect the pieces. To pick up the wobble goal, the opposable finger pinches the wobble goal between two stationary arms with gaps in them so that the wobble goal stays inside until released.



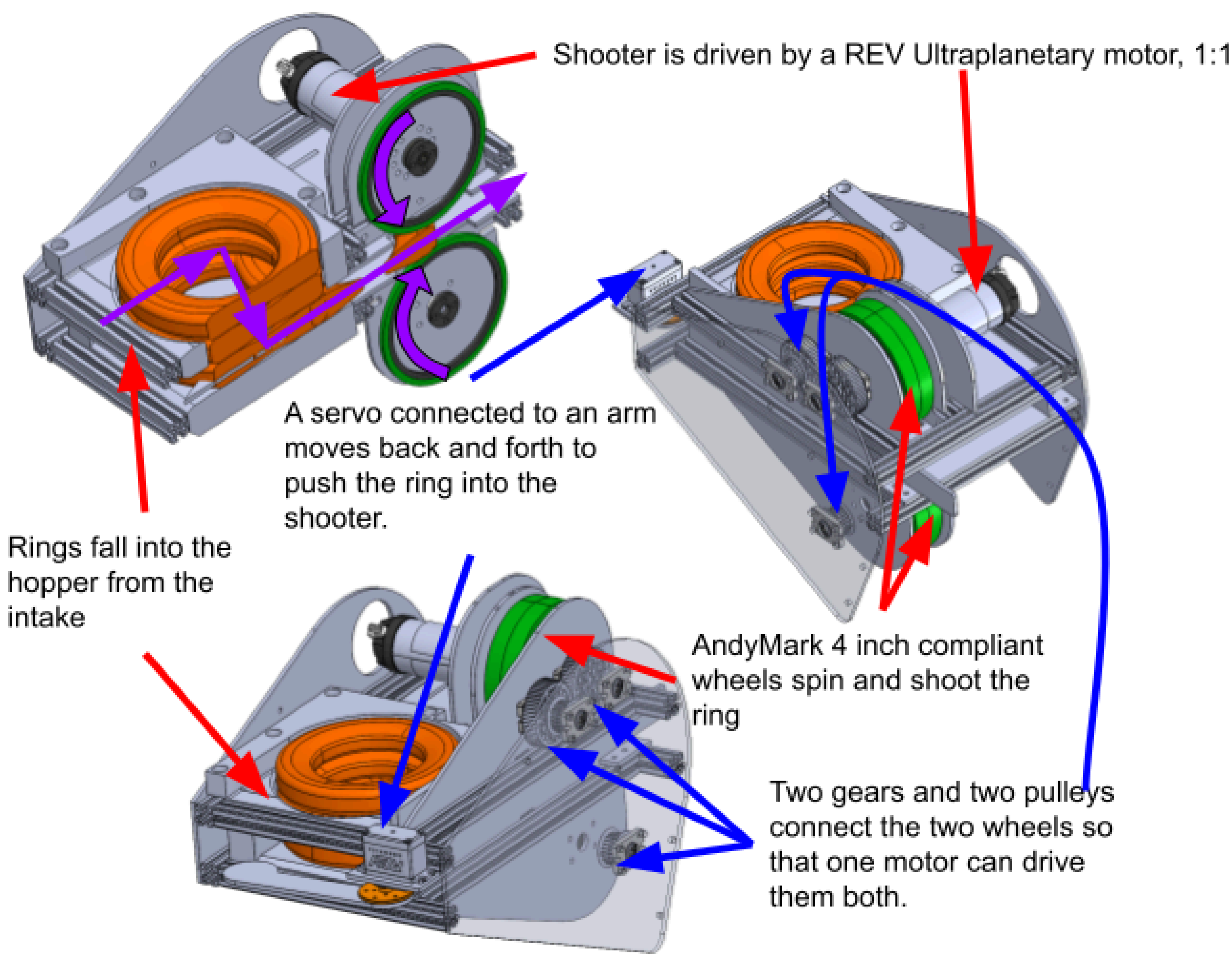
Intake and Conveyor Design

Our intake and transfer mechanism uses 3 sets of rollers to pick up a ring and push it up an MDF ramp. The first set of rollers is on a hinge that allows it to bob up and down to effectively capture the ring without bending and squeezing it against the ground and our ramp too much. We had a lot of trouble testing with the amount of compression that we are exerting on the ring.



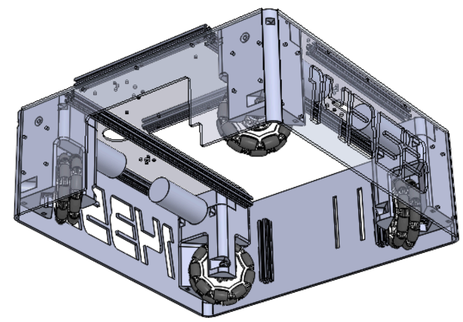
Hopper, Indexing and Shooter Design

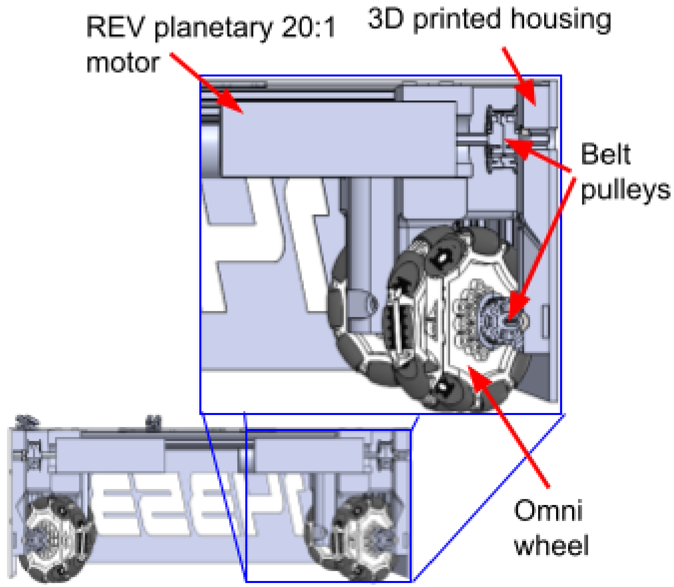
Our shooter, indexer, and hopper are three different mechanisms integrated into one assembly. The shooter uses a REV 1:1 motor and AndyMark stealth wheels to launch the rings out of a “barrel” made out of laser-cut MDF. We made the barrel to be like the barrel of a gun and aim the ring at a specific place consistently. Our shooter uses two gears and a timing belt to drive both of our shooting wheels with one motor. It took us a while to make the belt and gear spacing correct to get the right belt pulley distance and gear overlap. Our hopper is 3D printed. It holds 3 rings and has holes for rings to enter from the intake and exit to the shooter. The intake picks up rings and drops it in the hole for the hopper. For our indexer, we have a servo moving an arm that pushes the bottom ring out of a hole in the side of the hopper and into the shooter.



Omni-Directional Drive Train Design

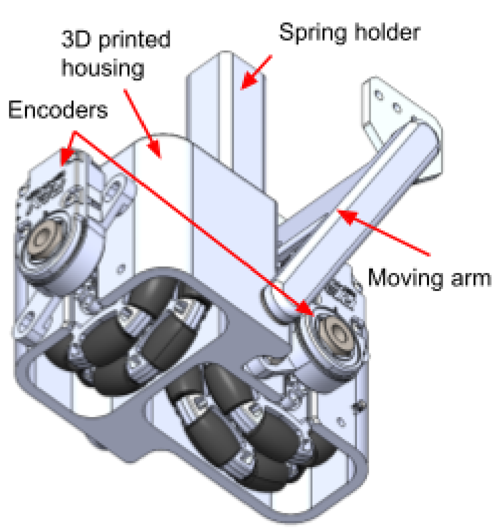
Our drivetrain is what we call a “diamond drive”. The drivetrain has 4 Omni wheels positioned at 45-degree angles. The movement of the Omni wheels combined can allow us to move forward and backward, left and right, and also turn. Last year, our pods had wheels attached to a motor with a chain. This caused motors to stick into the robot and take away a lot of space. We spent the entire summer redesigning the drive pods using a 3D printer. In our pods, our motors are connected to our wheels with a belt that is twisted 45 degrees. This allows us to tuck our pods to the sides of the robot and save a lot of space to build other things like intake and odometry. Getting the spacing between the motor pulley and the wheel pulley right so that the belt had the proper tension took a long time and a lot of trials. We finally got it perfect after the fifth prototype. The pods are simply screwed into an MDF board that holds the robot together. The large MDF board is where all of our individual assemblies connect together.

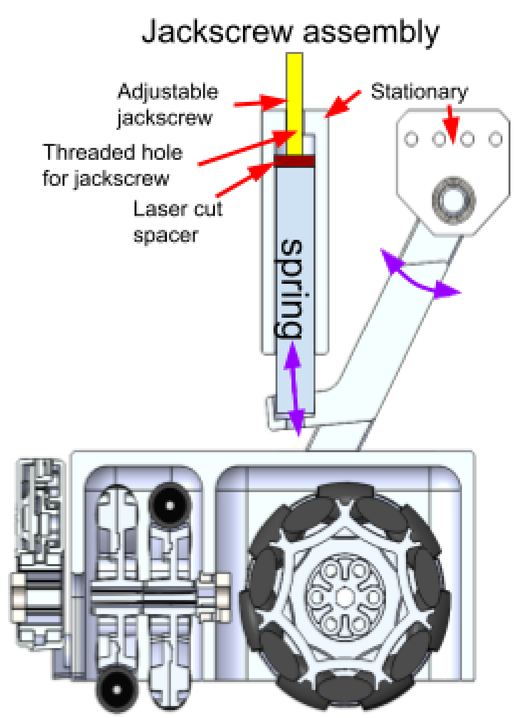




Odometry Design

This year, we needed to have more accuracy and consistency during the autonomous period. In order to do this, we had to supply our programming team with odometry. The advantage of using odometry instead of just using the motor encoders is that we don’t have to worry about the wheels slipping. It allows for a much more efficient autonomous program. Our odometry is spring-loaded. The spring pushes on the pod from a stationary holder to ensure that the wheels always maintain contact with the ground. A jackscrew can be adjusted to change the compression of the pod.





Programming

Auton -

Unique parts of our auton consists of path following using a custom Pure Pursuit algorithm, vision detection using both TensorFlow and OpenCV, and a custom velocity following algorithm to provide precise movement control on our drive train. Our robot can grab and place both wobble goals into the correct position using vision, shooting rings into the goals with time left, and then parking on the launch line.

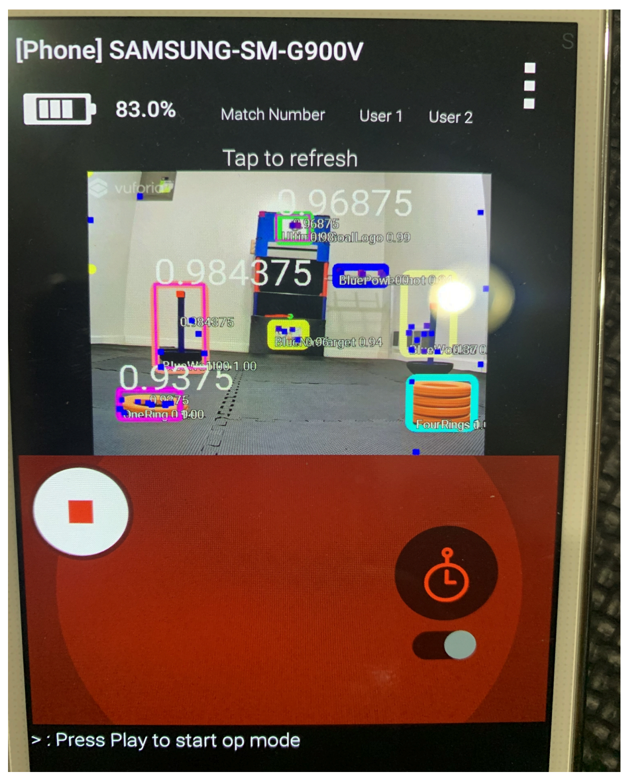
[NERDBOTS - AUTON]

Vision - TensorFlow and OpenCV

Vision will be used for detecting the number of rings and will also be used to find the angle for the robot to turn and shoot the powershot targets. There are two softwares used for vision. One is TensorFlow and the other is OpenCV.

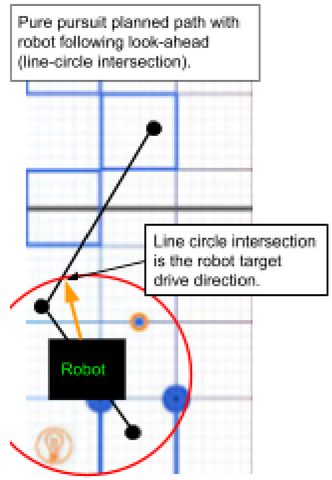
The process used for OpenCV is that the robot draws rectangular boxes. These boxes will cover the area where the rings are. Then the robot will see how much orange or red is being detected inside the box. Based on these numbers, the robot will see how many rings are in the stack. It will also detect where the powershot targets are, and where the high goal is.. We got some of these ideas from Wizard.Exe tutorials

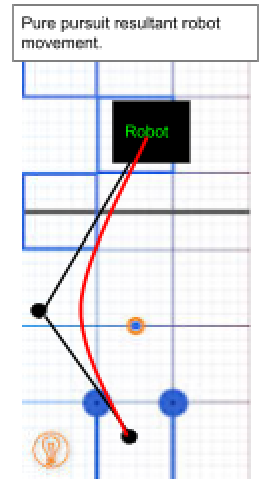
Tensorflow is a machine learning software. It takes many images from the user with the desired objects labeled, and then trains itself to detect these objects when the program is run. We took a video surrounding the desired objects, and then labeled every 10-20 frames. Tensorflow will take the frames that were labeled and train itself, as mentioned earlier, and when stopped, the trained model can be exported and then used in programs. We were able to train our own Tensorflow models to detect Wobble Goals, PowerShots, and the High Goal and then export it to use in our code, rather than the default model in the SDK, which could only detect rings. Using these models, we were able to find the distance to the desired object as well as the angle to the desired object. And then use that to align ourselves to the high goal.

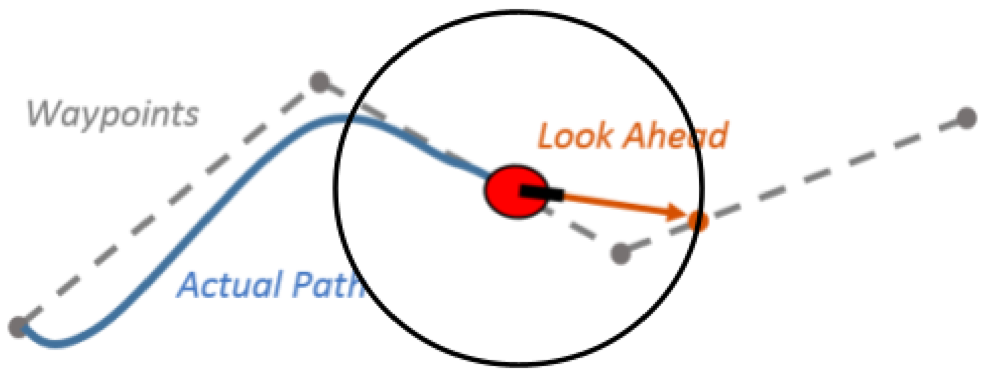


Path Following using Pure Pursuit - Custom Algorithm

Pure pursuit is a path tracking algorithm and is part of our motion profiling system. Pure pursuit allows us to create a path on the field for the robot to follow and allows the robot to smoothly follow it. The path is created with (x, y) points on the field. These points are read into an array, and a line is calculated for each set of points. The robot follows the path by calculating a look ahead point on the path and driving towards that point. The look ahead is calculated through a line/circle intersection, where the look ahead distance is the circle radius. We decided to use pure pursuit because it enables smooth, continuous motion over a defined path and saves time.







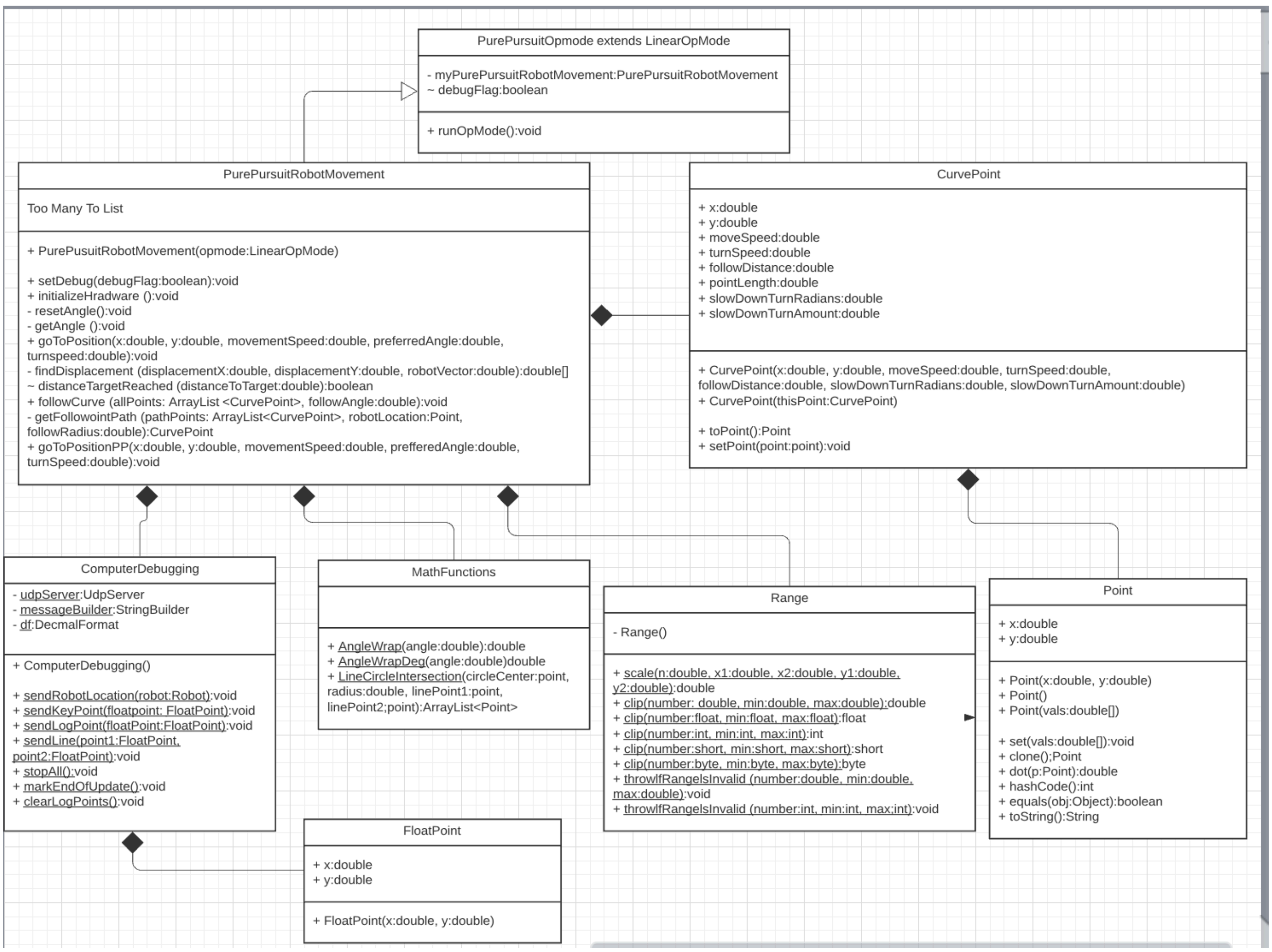
In order to do pure pursuit, other controls are needed, and these include: field centric driving, field centric odometry, PID (Proportional Integral Derivative), and we also used feed forward.

Path points are planned in relation to the field, so field centric driving will enable driving to the points, and facing a certain way it goes. Field centric driving is where we command the robot to drive in a certain direction in relation to the field. Field centric driving allows the robot to face a specified direction while moving across the field (think about x,y, and z speed/power). For pure pursuit, we will command the robot to go to different (x, y) coordinates on the field. In order to drive to the next commanded coordinate, we need to implement field centric driving.

Field centric odometry also plays a large role in pure pursuit. We use odometry and odometry encoders to track where the robot is/has gone on the field. Doing so enables the path planning and motion profiling that is crucial to making pure pursuit work. Right now, we have the odometry track (x,y) motion where it accounts for the robot turning.

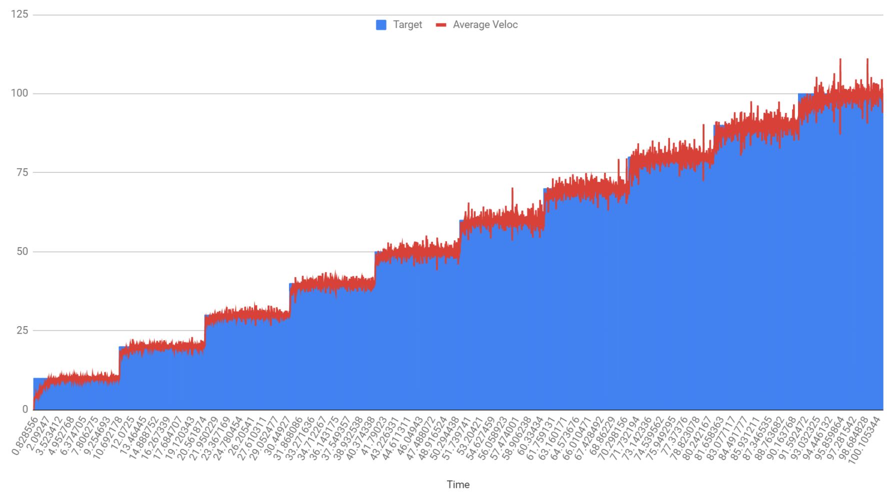
Even though pure pursuit is a tough concept, our team decided to try it out. We first started investigating it and looking into it in January of 2020. By the end of March, we had it working a little bit because the algorithm was working. We still had many issues to fix, but it was starting to work. In September-October, we merged the pure pursuit code and velocity following code, which was the next step to making pure pursuit usable. It took several tests to successfully merge the code, but we finally got it working. Once we merged the pure pursuit and velocity following code, the pure pursuit wasn’t very accurate and it needed to be tuned. There was also an orbiting issue we needed to fix.

Eventually, we got it working completely with both pure pursuit and velocity following. We learned a lot from the GlutenFree(11115) and Wizards.Exe(9794) tutorials. Now our robot can move faster and more accurately than ever before. Using advanced pure pursuit technology, our robot is able to complete tasks in auton quicker and more accurately.



Velocity Following - Custom Algorithm

We use velocity following to feed the information we get from Pure Pursuit to our drivetrain. It uses a separate PID on each of our wheels. The PIDs receive the target velocity from Pure Pursuit, and also the motor velocities from another function in our code. That function basically sees how many motor ticks have passed over one loop, and translates that velocity into inches per second, which is the unit that we use. We also have a running average on the output of that function to smooth out the values. The PIDs then increase (or decrease) the motor power until the motor velocities are equal to the target velocities. The final function of velocity following is a custom ramping algorithm, which we use to speed up and slow down accurately and without wheel slip.



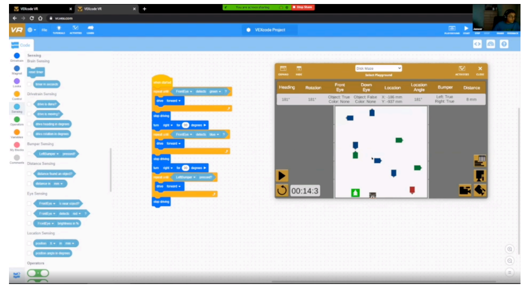
One test with ramping enabled. The red line is the average velocity of all the motors, and the blue is the target velocity.

Teleop

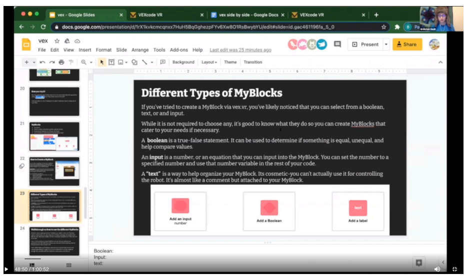
Our TeleOp uses a mixture of many algorithms and techniques. We use field-centric driving to make driving intuitive and easy, so our drivers can focus more on navigating the field instead of the robot's orientation. It basically makes sure that, relative to the field, forward is always forward, right always right, and so on and so forth. We do this by essentially seeing how much the robot has turned using the gyro and offsetting the heading commands. For our wobble goal arm, we use a time-based system to lower and raise the arm, as well as code to move the servo. The intake is simply one motor which we run whenever we want to pick something up. Finally, our shooter has two parts, an indexer, and the motor which actually flings out the rings. The indexer is a servo that moves in and out to push rings into the motor after it reaches enough speed, and the motor is just run at full power whenever we shoot. We also use vision to aim the shooter at the goal, which we went over earlier.

Leadership

FIRST Outreach - NERDbots VEX-VR Programming Booster Pack



On January 2nd, over a Zoom call, we taught about 25 students in elementary school how to program a virtual robot on a website called vr.vex. The kids learned to use a block coding language based on scratch to guide their virtual robot through a maze, detect colors and objects with sensors, and create complex programs using loops, variables, and custom blocks. We chose a programming session to teach kids as outreach. Since we were virtual, teaching them programming was the easiest and best option.

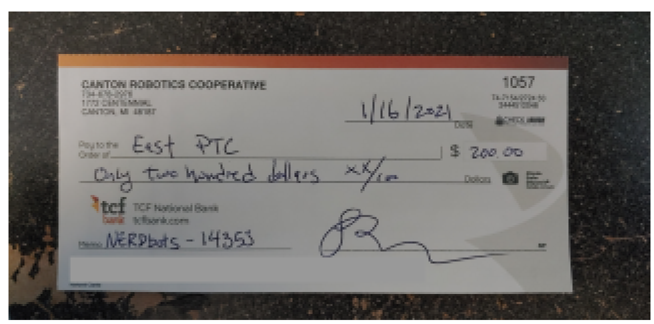


We taught them programming because it is useful for knowing if they join robotics competitions like FLL and FTC. We chose to use a simple program that was block programming so that it was easier to understand. There was a virtual bot so it was easier to see what they did.

[Vex-VR Booster Pack video]

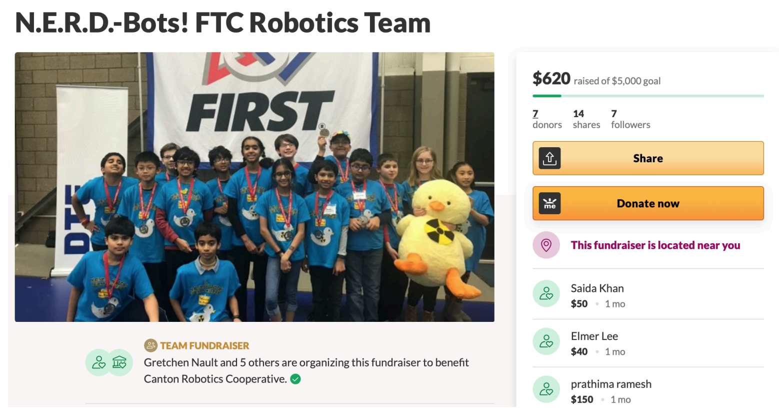
Community Outreach - Donating to the East Middle School PTO

After 3 weeks of our Leaf Raking Fundraiser, and earning $560, we decided to donate 35% to our school's PTO, which is $200. Due to COVID-19, our school's PTO had a difficult time fundraising. It was only natural that we help our school and our community in their time of need. We got in contact with the East PTO's (our school) treasurer and asked her how we could donate. She said that we could donate through Paypal or send a check to the East PTC bank account. To the right is the check we sent.



Fundraising - GoFundMe Page

Just like last year, we are using a GoFundMe page for funding our robotics team. We chose to use a GoFundMe page because it's easy to access and everyone can donate, no matter where you are. On our GoFundMe page, we have a blurb about our team and our achievements, along with an explanation of what FTC is. Last year, we got over $1400 from our fundraiser. We currently have our GoFundMe page up, and so far we've collected about $600!



This year, we did a leaf-raking fundraiser to raise money for our school and team. We gave 35% of our earnings to our school and kept the rest to use for parts or materials. We used the GoFundMe page to collect funds from people who had us rake their leaves.

Our GoFundMe page

Fundraising - Leaf Raking

We decided this year to raise money for our team by raking leaves in the fall. We made sure that everyone wore masks and stayed a good distance from each other. We started by creating flyers to put in people's mailboxes. Then we made a spreadsheet where people would put the times they were available. We would then assign a day and time for everyone to meet. We would rake the leaves and leave a suggested donation amount adding up, per bag. We gave people the link to our gofundme page and told them they could donate with that. We raked 9 yards and filled 53 bags of leaves. We raised a total of $560, and we gave 35% of our earnings to our middle school's PTO.

Fundraising - Corporate Sponsorships

This season, our team, the NERDbots gave out sponsor packets to different companies that we could reach out to through this pandemic. We gave them out to Ford, Aisin, Flex'n'Gate, and Conceptual Innovations. In total, we got $1,800 from just sponsors!

★Golden Ducky Sponsors: $500

FLEX N GATE

CONCEPTUAL INNOVATIONS
ENGINEERING FROM CONCEPT TO INNOVATION

For a Better Tomorrow
AISIN GROUP

☆Silver Ducky Sponsor: $300

Ford

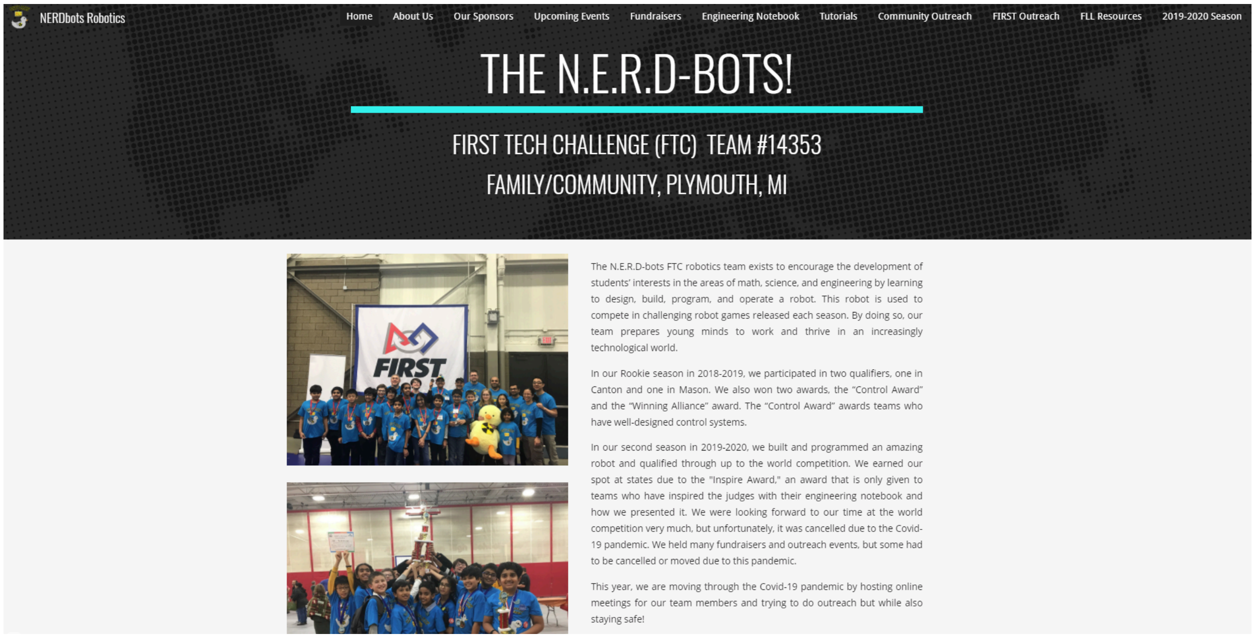
Here is our **Sponsorship Packet**.

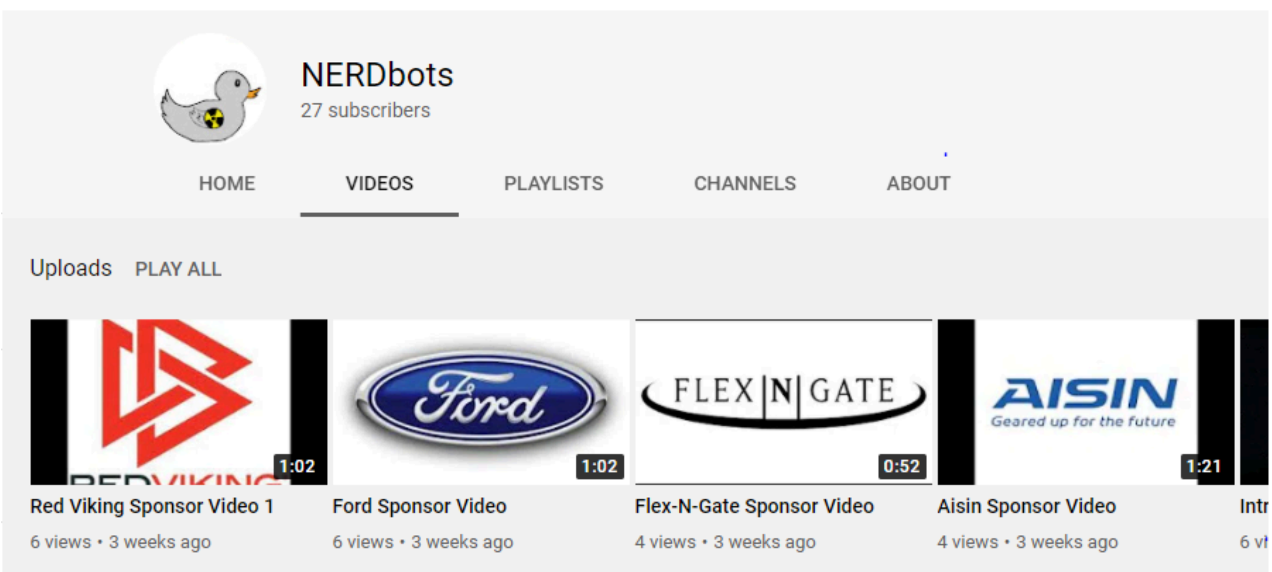
Digital Outreach

Website: **https://www.nerdbots.org/**

Youtube: **NERDbots**

In our website, we have a lot of information that informs the community of what we are doing. We have many shared designs and programs that can help other FTC teams. We also held a booster pack event to kickstart the FLL teams and help them with more advanced topics. We plan to hold more in the future. On our Website, we have engineering notebooks like this one, that inform the community about our outreach, what we create, and who we are. We have details about us and our team. From our first season all the way to now. Our youtube page has a lot of helpful tutorials about tips for FTC teams like how to fundraise, how to hold outreach events, and what FTC is. Our youtube page also has our previous matches, sponsor videos, and more.


NERDbots Robotics
Home
About Us
Our Sponsors
Upcoming Events
Fundraisers
Engineering Notebook
Tutorials
Community Outreach
FIRST Outreach
FLL Resources
2019-2020 Season
THE N.E.R.D-BOTS!
FIRST TECH CHALLENGE (FTC) TEAM #14353
FAMILY/COMMUNITY, PLYMOUTH, MI
The N.E.R.D-bots FTC robotics team exists to encourage the development of students' interests in the areas of math, science, and engineering by learning to design, build, program, and operate a robot. This robot is used to compete in challenging robot games released each season. By doing so, our team prepares young minds to work and thrive in an increasingly technological world.
In our Rookie season in 2018-2019, we participated in two qualifiers, one in Canton and one in Mason. We also won two awards, the "Control Award" and the "Winning Alliance" award. The "Control Award" awards teams who have well-designed control systems.
In our second season in 2019-2020, we built and programmed an amazing robot and qualified through up to the world competition. We earned our spot at states due to the "Inspire Award," an award that is only given to teams who have inspired the judges with their engineering notebook and how we presented it. We were looking forward to our time at the world competition very much, but unfortunately, it was cancelled due to the Covid-19 pandemic. We held many fundraisers and outreach events, but some had to be cancelled or moved due to this pandemic.
This year, we are moving through the Covid-19 pandemic by hosting online meetings for our team members and trying to do outreach but while also staying safe!


NERDbots
27 subscribers
HOME
VIDEOS
PLAYLISTS
CHANNELS
ABOUT
Uploads
PLAY ALL
1:02
Red Viking Sponsor Video 1
6 views • 3 weeks ago
Ford
1:02
Ford Sponsor Video
6 views • 3 weeks ago
FLEX|N|GATE
0:52
Flex-N-Gate Sponsor Video
4 views • 3 weeks ago
AISIN
Geared up for the future
1:21
Aisin Sponsor Video
4 views • 3 weeks ago

Sustainability Plan

Recruiting

In the past, we held outreach events at our local library and school to help get kids interested. Our team grew from 6 to 15 members because of our recruitment in the summer of 2019.

This year due to Covid, we couldn't hold outreach events but we did get some new team members through our internship program which provides an opportunity for younger kids to learn about CAD, programming and more about FIRST.

Click here to see one of our recruiting events here.

Training

Typically, over the summer, we train our team members by doing in-person tutorials in Cad and Programming. Due to Covid-19, this year was only online and at home. For example, our build group participated in a catapult design exercise where we made

catapults with limited dimensions and only one rubber band. We had to try and use the least material and have the smallest weight. We would be Covid safe and test everything at our houses. For programming, teammates were given projects so that they can learn Java, and continue advancing their skills. For leadership, teammates brainstormed fundraising and outreach ideas to plan for the upcoming season.

Fundraising

In order to purchase parts, compete, and sustain our team, we need to fundraise. In previous years, we held can drives and fundraisers at restaurants in addition to having a gofundme page and getting money from our corporate sponsors. We also held robot demos at our fundraisers at local restaurants. This year, due to COVID-19, we weren't able to get cans and go to restaurants. So, we held a leaf raking fundraiser and raised $560. Once everything is back to normal, we hope to resume holding outreach and fundraiser events at public spaces.

Budget

Starting Balance	$1,956.72
In-Season Costs	
Robot Parts	$4,835.93
Raw Materials	$1,921.43
Competition Fees	$275.00
T-Shirts	$504.00
Incoming Funds	
Corporate Sponsorship	$1,800.00
Membership Fees	$4,875.00
General Fundraising	$620.00
Ending balance	$1,715.36