# 13190 Roblivion Engineering Portfolio
## Adrian C. Wilcox High School


**Powered By:**

# Engineering Portfolio

## Table of Contents

**Journal Reference:**
[https://docs.google.com/document/d/1sSCjNCHx5khVf4wi5Rt6prWqcqOe1KUgdgEGfOJnH2g/edit#heading=h.1j52h5ip6ntg](https://docs.google.com/document/d/1sSCjNCHx5khVf4wi5Rt6prWqcqOe1KUgdgEGfOJnH2g/edit#heading=h.1j52h5ip6ntg)

## About the Team

We are Team 13190 Roblivion based in Santa Clara, California. We did not compete in the 2020 season due to the pandemic, so when we decided to participate in this year's season, we found ourselves with an abundance of new members who had never competed in FIRST before. Last year, 13190 was disbanded, and so the team is composed of 'rookies' this year. With completely new members, the team has been transformed from all levels, essentially as a completely new team. Therefore, we have had to adjust to setbacks such as a lack of collaboration between committees. Every year, before the release of the season's challenge and kick-off, we work at our school's club fair in order to publicize and gain new members. We hold the opportunity for new members to join the team directly or join the club, which we mentor and the members are free to explore without pure commitment.

## Strategic Plan - S.W.O.T Analysis

**Strengths:**

- This is the strongest 13190 team in years.
- Large team size
- Solid leadership and delegation skills
- Opportunity to learn under our sister team 8872

**Weaknesses:**

- Lack of consistent attendance
- Inexperience in new members
- Time it takes to teach new members
- Disorganized and small room
- Few members work on software and business
- Lack of collaboration between the software and hardware divisions

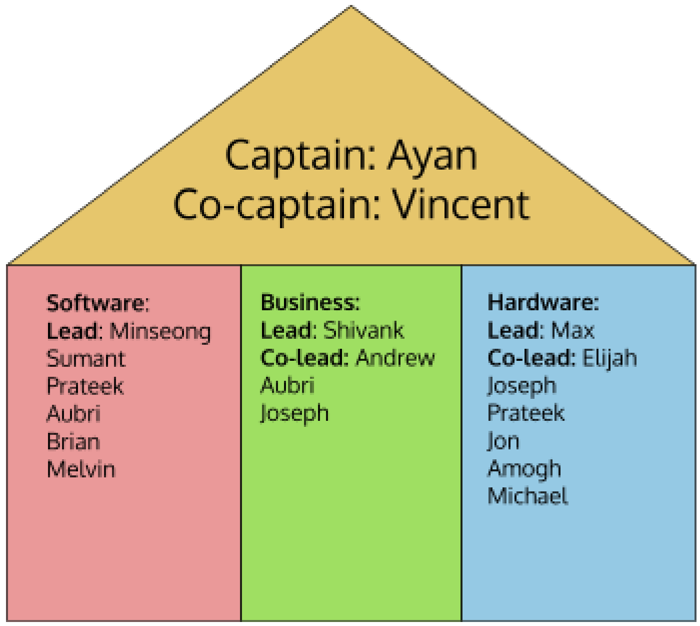
**Opportunities:**

- Parade of Champions
- Cruise: autonomous vehicles
- Meeting different teams and communicating with them
- Attending school events
- Hosting an in-house tournament
- Making connections with Vijay Kumar and other industry leaders

**Threats:**

- The rise in COVID cases resulting in even less consistent attendance
- Tighter pandemic mandates = difficult time to meet/lowering the possibility of communication
- The inexperience of some members who may not be able to step up when they are needed

## Team Organization




Team 13190 is divided into three divisions: hardware, software, and business.

### Hardware:

The hardware division works on prototyping the robot and engineering the physical design; they make blueprints in Fusion 360 and share their ideas in Blender. The hardware team are the ones who assemble the robot and test it for potential physical issues.

### Software:

The software division programs the robot in Java. The software team programs the Autonomous and TeleOP phases of the competition. They work closely with hardware to ensure that the code is accurate, precise and relevant to the bot being built.

### Business:

Business members get to plan outreach events, work on budgeting, fundraising, and attracting sponsors for the team. The business committee also works on documentation, writing and formatting the engineering portfolio and notebook.

These departments are managed by the Captain and Co-Captain, who ensure the team is on schedule and delegate tasks to team members.

## Team Goals/Plan

| Strategy | Actions | Group Responsible | Completion |
|---|---|---|---|
| Learning from engineering leaders in robotics. | We learned about robotics from Dr. Vijay Kumar, who is the Dean of Engineering at the University of Pennsylvania. | Leads, Software, Business, Hardware | January 2022 |
| Developing the software and hardware skills of new members. | Ayan, the team captain, taught members about Blender and Fusion 360, which are the two pieces of software used to model prototypes. Minseong, the software lead, taught new software members about Java, which is the programming language used to code the robot. | Hardware, Software, Leads | November 2021 |
| Have outreach presentations for local elementary schools in the Santa Clara area. | Team 13190 had outreach presentations directed at elementary schools in the Santa Clara area, mentoring elementary school students about STEM and robotics. | Leads, Hardware, Software, Business | December 2021 |
| Mentoring other local teams | Team 13190 mentored FTC design practices to many other rookie teams, including 19568 Legot and others. | Leads, Hardware, Software, Business | February 2022 |
| Outreach event: The Parade of Champions in Santa Clara | Team 13190 built a robot for the Parade of Champions, an annual event in the city of Santa Clara. The parade helped promote robotics in the community. | Leads, Hardware, Software, Business | October 2021 |

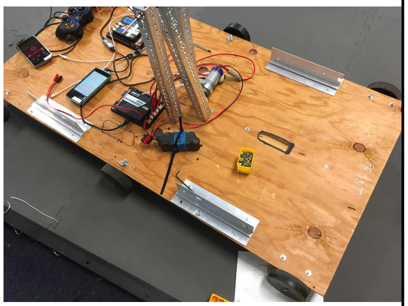
## Team Strategy

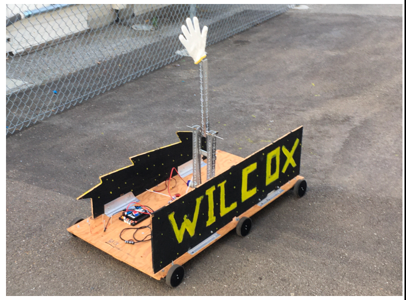
| | |
|---|---|
| **Autonomous - 30 seconds** | Path 1:<br>Deliver Duck<br>Park in warehouse<br>Path 2:<br>Deliver Duck<br>Park in alliance specific shipping unit<br>Path 3: |

| | |
|---|---|
| | Park in warehouse<br>Path 4:<br>Deliver preloaded freight onto corresponding level of alliance specific shipping hub<br>Deliver duck from carousel<br>Park in warehouse |
| **Tele-op - 90 seconds** | Strategy 1:<br>Deliver Freight to the top level of the shipping hub as consistently and efficiently as possible<br>Strategy 2:<br>Deliver Freight to the shared shipping hub and try to keep it unbalanced towards our side |
| **Endgame - 30 seconds** | Strategy 1:<br>Continue delivering freight to the top level of the shipping hub and park with 5 seconds left<br>Strategy 2:<br>Attempt to deliver ducks using the carousel and park in the warehouse |

## Outreach and learning experiences

**1.) Parade of Champions:** Annual parade in the city of Santa Clara every October.
On October 9, we attended the annual Parade of Champions, along with our sister team 8872, in a parade in Downtown Santa Clara that celebrates frontline workers and recognizes local businesses. Over 150 groups participated in the parade itself, including various sport clubs, bands, and cheer teams. All participating groups came from local schools and community organizations. Many people came to watch the parade. The city worked with the Parade of Champions Organization to make the parade a success. For this event, we designed and built a robot for the parade to promote robotics in the city. The team added a waving hand to make it distinct from the rest of the crowd. The arm utilized a chain mechanism to move back and forth, allowing for the hand to wave toward the crowd. The robot used a large piece of plywood for its base, and six high traction plastic wheels, which were useful, especially on the asphalt of the road. Four motors were used, while two supporting wheels sat in between. The software team reviewed code while the hardware team worked together while working out the last kinks of the robot. Right before the parade, while testing the robot, the wooden waving hand snapped due to extreme pressure, so we had to use zip ties instead to attach it to the arm. The robot ran fine for most of the parade. A few feet from the finish line of the parade, the chain mechanism supporting the waving hand broke, and one tire fell out shortly later. However, the robot was still able to continue the parade, although at a slower pace with a stationary hand. We learned the importance of double checking, rather than shipping out an untested product. We also learned that products should be tested more. Our team also acquired new skills, including more knowledge on the engineering process, how robots should be made, and the basics of motor control in Java, as many members on the team had little to no experience at this time.

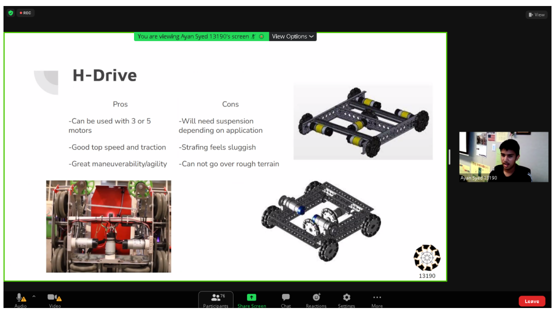

Hardware setup

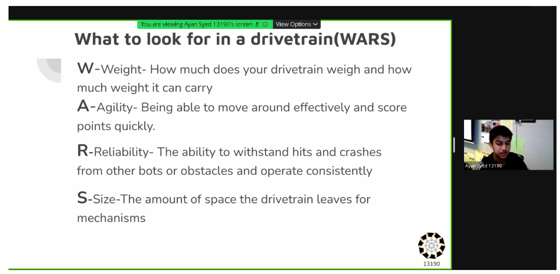


The parade robot.


The robot in action at the parade.

**2.) Virtual Presentations - Drivetrains**

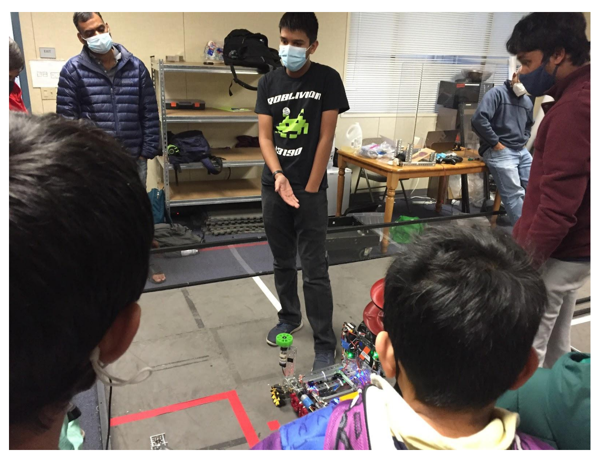


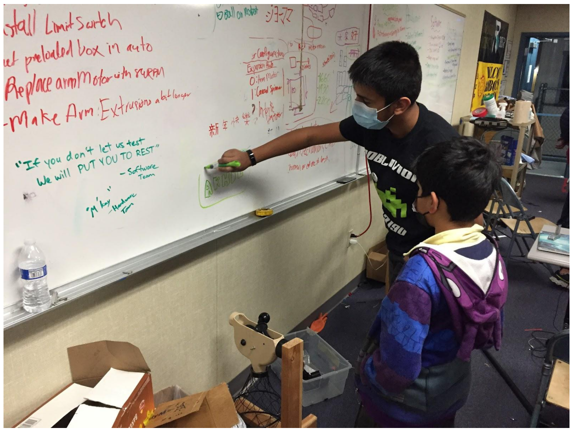




This year, we had presentations about drivetrains, directed at newer FTC teams. 19568 Terra Transport attended the meeting as well. In these presentations, we introduced our team, and lectured about the basics of drivetrains. We described the pros and cons of the different drivetrains, and which ones we personally believe are optimal for the challenge. We compared Tank Drivetrains to Holonomic drivetrains and went into depth about the specific use cases of each type. Not only did this help newer teams learn about the options available to them in terms of drivetrains but also allowed us to expand our preexisting knowledge on the subject.

**3.) Mentoring other FTC teams**


Mentoring Fusion 360


Explaining how our robot works

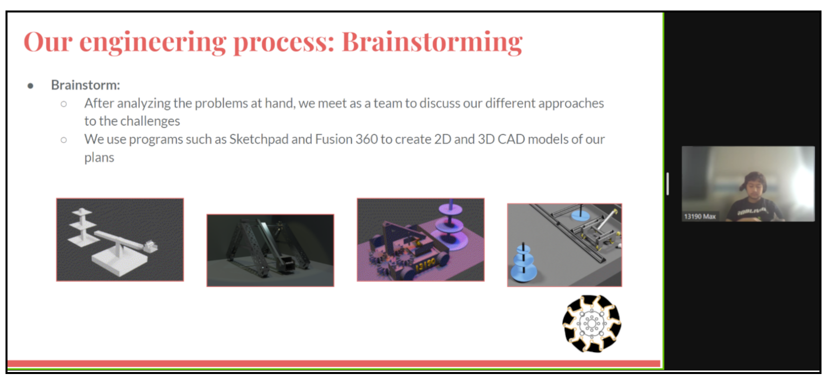

Describing the engineering process


Answering 19568 Legot's questions

Team 13190 mentored the basics of FTC hardware to many other rookie teams, including 19568 Legot, 13217 Terra Transport, and others. We described how the challenge worked, and the engineering process, including how we developed prototypes in Blender and Fusion 360. The team also gave presentations about drivetrains to other rookie FTC teams. With the experience that we learned from trial and error, we taught about the common errors that we made with hardware and software, and how one can avoid them. We also hosted Blender workshops, and how Blender can be used during the prototyping process in robotics. We showcased our robot, and the other teams utilized our practice field to test out their robot. We gave feedback to the other teams, and how they can improve their designs. There is a common saying that goes, "Teaching is the best way of learning". Our presentation to these younger students was in hopes for them to expand their horizons to robotics in hopes for them to gain an interest in this technology as we have.

**4.) Virtual Presentations - Santa Clara Unified Elementary Schools**

We hosted presentations about STEM and the engineering process for elementary students in our district. We introduced our team and explained the foundations of the engineering process. The presentations were used to define what the engineering process is, using real life examples.  We believe that educating students about the engineering process (specifically robotics) would be extremely relevant for the future as technology is on the path of being fully automated soon.  The presentations were done with the goal of inspiring and teaching the next generation of coders, scientists, and researchers about the engineering process. We also taught about our team, in hope of recruiting new members.



### 5.) Dr. Vijay Kumar - Dean of Engineering at the GRASP lab at the University of Pennsylvania

Over Zoom, Team 13190 and 8872 had the opportunity to hear and learn about Dr. Vijay Kumar's incredible robotics accomplishments. He is a board member of one of our sponsors, O2 Micro. At the University of Pennsylvania, he worked on aerial robots, improving their computing performance while trying to keep them at a smaller form factor. He described robotics as an interdisciplinary field, with required expertise in computer science, electrical engineering, and mechanical engineering. With the advice of Dr. Kumar, our team learned more about the field of robotics and the industry as a whole.

### 6.) CRUISE AV

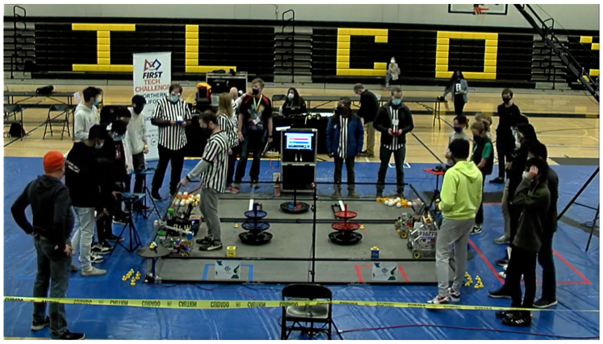
The exterior of Cruise AV's data collection vehicle.

Team 13190 inside "The Origin"

Exterior of "The Origin," a self-driving autonomous vehicle with no steering wheel.

Our team visited an autonomous vehicle company, known as Cruise AV. The team got the opportunity to visit their office and test garage in San Francisco. We got an exclusive look at the test garage, the place where vehicles are tested and prototyped. At Cruise AV, our team learned how the skills acquired while completing the engineering challenge can be applied to real scenarios. Our team saw cutting edge technology, including "The Origin", a vehicle that is fully autonomous without needing a steering wheel. To train the computer vision algorithms for the autonomous vehicles, street data was collected through normal passenger cars, and hard drives containing terabytes of data were collected every day of driving! From all of this data, the vehicles are able to read road signs and adapt to the traffic on the road. With the recent developments of self-driving trucks in Texas, autonomous cars may soon become prevalent on the roads and highways. However, the technology must be tested in making sure that it is safe, reliable, robust, and resilient for the rapid nature of traffic. The visit at Cruise AV emphasized the importance of testing, rather than simply shipping out a product. The products at Cruise AV went through multiple revisions, and most of the vehicles are under the prototyping stage. Like the vehicles at Cruise AV, our team utilized the use of autonomous technology in our robot, with the use of an attached webcam. On the software side of the vehicle, TensorFlow was used for computer vision, detecting road signs and traffic lights. With our robot, ducks and freight need to be detected during the autonomous phase using computer vision. The team also learned about reliability, especially when building and designing our robot. In a busy and dangerous city like San Francisco, reliability is key to ensuring both passenger and pedestrian safety.

**7.) In-house Tournament**

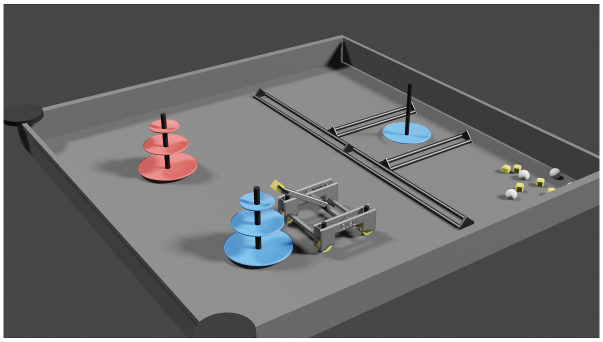
| | |
|---|---|
|  The tournament view. |  The background view. |

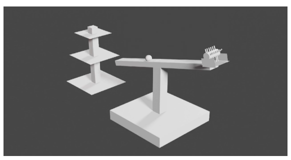
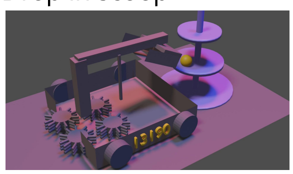
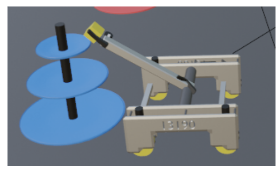
On Saturday, January 15th, we hosted a tournament at Wilcox, along with our sister team 8872. This was a great opportunity to experience the robots of different teams, we learned about possible design choices that can be implemented in our robot. We also learned about the time format and setting of the tournament, which will be extremely useful, due to how our team consists of mostly new members. 8872 mainly worked on the execution of the tournament, while 13190 set it up.  A day earlier prior, we prepared for the tournament. We knew that the main tournament would take place in the Main Gym, while the school's cafeteria functioned as a pit, in which teams work together to test and work on their robot. To prepare for the tournament, we moved the practice field from our robotics room to the cafeteria. While at the cafeteria, we organized the tables for the respective teams.  Later, we set up extension cords and power strips in the cafeteria, so the other teams have electricity to charge and power up their robots and electronics in the pit-stop. Two separate sound systems were set up in the Main Gym and the Cafeteria. The official FTC inspectors and engineers came in to set up the official field and the computer systems. A camera system was set up in the Main Gym, with the purpose of livestreaming the tournament onto YouTube.

**8.) Kevin Yamada**


Team member Michael's dad, Kevin, a material scientist, taught about the properties of different metals, alloys and plastics, and how they are applied. This was especially important for the team, due to the influx of new hardware members who have little past experience about the usage of materials. He taught how bronze and copper work poorly for the gears, due to how they are both soft metals. We also learned that steel was an alloy, or a homogeneous mixture consisting of iron and carbon. We learned how to classify the different types of plastics, which can be applied during the 3D printing process. We learned that ABS is one of the most durable plastics, but it releases toxic fumes. We also learned that PLA has a lower melting point that produces more accurate 3D prints and is widely used among hobbyists. PETG is more flexible than other thermoplastics such as PLA & ABS. We learned that TPU is a more flexible plastic resin used for 3d printing.

## Engineering Process

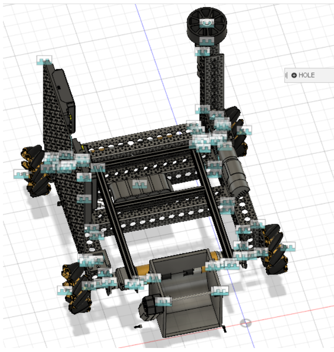
| | |
|---|---|
|  | Ayan alongside the hardware team made the first prototypes in Blender of the robot shortly after the official announcement. There were several designs proposed by members of the team such as ones that utilized a claw design, an intake slider, or a scoop. |

| Intake slider | Drop in scoop | Claw | Scoop |
|---|---|---|---|
| <br>**Advantages:** The seesaw can slide through it and it is efficient when there is no need to move the robot.<br><br>**Disadvantages:** The freight on the seesaw might get stuck or fall out from it | <br>**Advantages**: Both claw and scoop that can carry freight one more at a time.<br><br>**Disadvantages**: The freight might get stuck in the intake due to how precisely it takes to get the freight in. | <br>**Advantages:** The claw can carry on freight quickly and deliver it to the shipping hub faster.<br><br>**Disadvantages:** The claw can lose control of holding freight when the robot bumps frequently. | <br>**Advantages:** The scoop can carry the freight close to the shipping hub and has a lot clearance to go over the barriers<br><br>**Disadvantages:** The arm on the scoop would need a lot of torque to spin. |

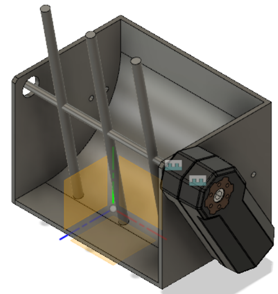
Out of the four prototype designs, we eliminated the claw mechanism, due to its unreliability, considering the precise amount of pressure that must be applied on the claw. This makes it unreasonable, considering that most team members are beginners, and are unable to control the claw precisely. In addition to this, items from a claw could fall out when the robot travels around the field.
The scoop design was eliminated, because of the amount of torque needed. Due to the short time given in the tournament rounds, the tossing scoop would require time to align itself with the shipping hub. Also, the angles of the arm would have to be adjusted and calculated, making it require large amounts of testing and adjusting. With limited time in both the construction process and the actual tournament, this design had to go.
The intake slide design was also eliminated, due to how it would not work properly with the cube freight. The intake slide design works by staying at a flat level, and after the freight is lifted onto the cubes, a motor changes the slope of the slider, causing the freight to fall towards the shipping hub. On the sides of the robot, there would be barriers, so the freight would not fall. However, this design was eliminated, due to the difficulties that may be encountered during the construction process, especially with building and designing the slide.

## Intake Prototyping

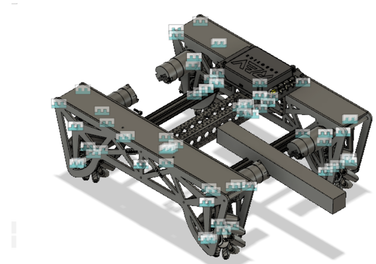
The team eventually settled with the drop in brush design, rather than a claw design, as it was far more easy to implement. However, from the initial prototype in Blender, we decided to do some modifications to the scoop. Rather than using a separate intake mechanism, as seen in the Blender render, we decided to make the scoop both an intake and drop in mechanism toward the shipping hub. Both the hardware and software teams agreed with this design, and started work.



**The final blueprint design of the robot made in Fusion 360.**

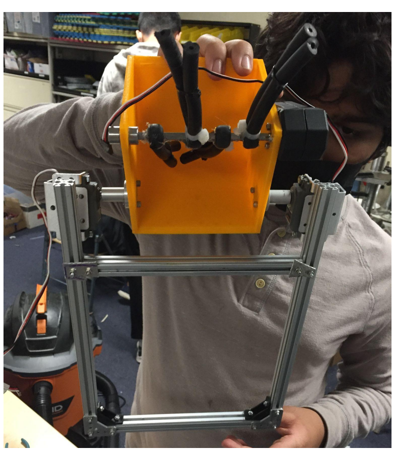


**The brush style intake mechanism.**

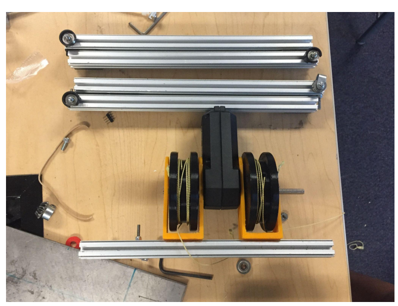


**A prototype for the base before the design change.**

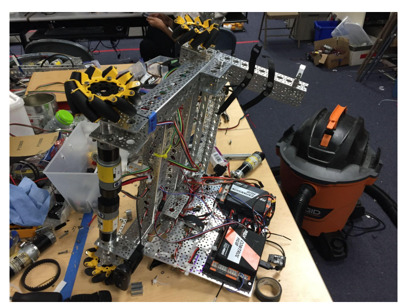
Hardware lead Max argued for the importance of a versatile and robust robot even before construction. Learning from the issues at the Parade of Champions, Max emphasized on making the robot remain strong and tightly put together, especially in the face of adversity. Max also called for a resilient and reliable robot, and he emphasized on prototype testing, throughout the steps of the engineering process. He and Ayan led the design of the robot, and created the blueprint prototype in Fusion 360. The drivetrain and basic chassis were designed first, with the intake mechanism designed later after seeing how the both moved around the field Max oversaw the assembly of the robot, and he encountered some difficulties, especially with the screws, as he needed to make sure that they remained tight during the competition. The arm mechanism also had some issues during our testing phase, as the REV sliders did not move. To combat this problem, the screws were loosened in the sliders, which ameliorated this issue. The intake mechanism was built using surgical tubing, which when rotated, acted like a brush. The arm then rotates around reaching until the back of the bot. The team periodically conducted maintenance to test the arm mechanism to make certain that it is working properly.



**Intake mechanism.**

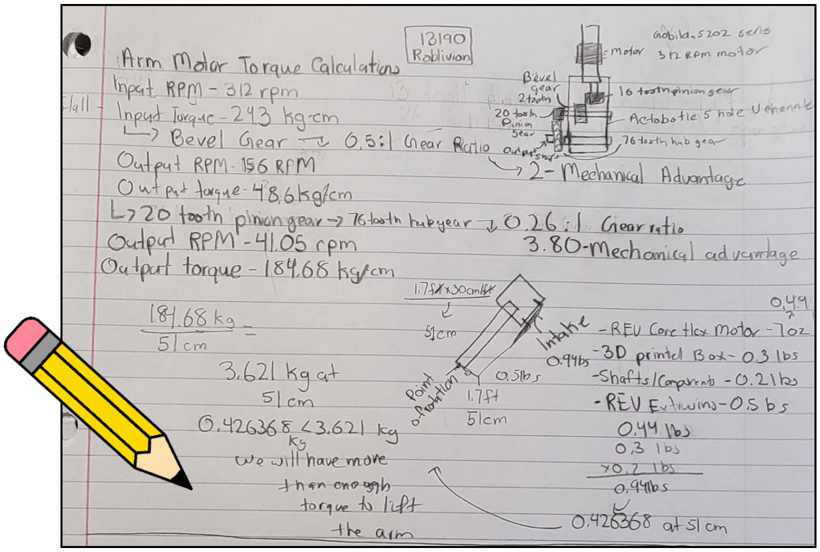


**The arm mechanism's REV sliders prototype**



**The body is under construction.**

After we had gotten the rev slide working, we had found out that the pulley system that would go with them was not working well and would slip. We tried to fix this by placing the pulley closer to the side of the rev slides, but it still wouldn't work properly and kept on slipping. So, we had to change the movement of the strings on the rev slides so that it would work most of the time, but it would have to be precise. After careful evaluation of our options, we decided to change the arm design and instead have the base near the back of the bot and have it stretch out across the length of the robot. The idea was simpler than the previous one and we wondered why we didn't think of it before. When we thought we were ready for our competition, the bevel gear that was rotating the arm sheared off. We worked tirelessly to fix this problem from occurring again. Ayan came up with the idea of doing a series of gear reductions to increase the torque output of the motor.




We had to figure out the necessary torque by using mass and torque calculations. Through trial and error, we learned that we would have to calculate the power and torque needed to move the arm prior to investing time and resources into building a prototype. None of our hardware members had taken a physics class or had experience calculating torque and revolution. Despite this, we were able to calculate the power and speed after a series of gear reductions. 2 days after the incident, we were able to implement the gear reductions successfully.

This year, having control over the bot was essential. On our robot we used multiple sensors as well as encoders on all of our drive motors. We made use of a camera on the front of the robot to determine the events going on the field as well as the level on which to score the preloaded freight in autonomous mode. We also use a REV bore encoder to find the arms angle which helped us achieve reliable arm movement in autonomous as well as driver aid.The arm utilizes a gearbox that converts speed into torque, so the motor does not get sheared off. We mounted a gear at the base, so the axle does not slip.

## Software

Tele-Op

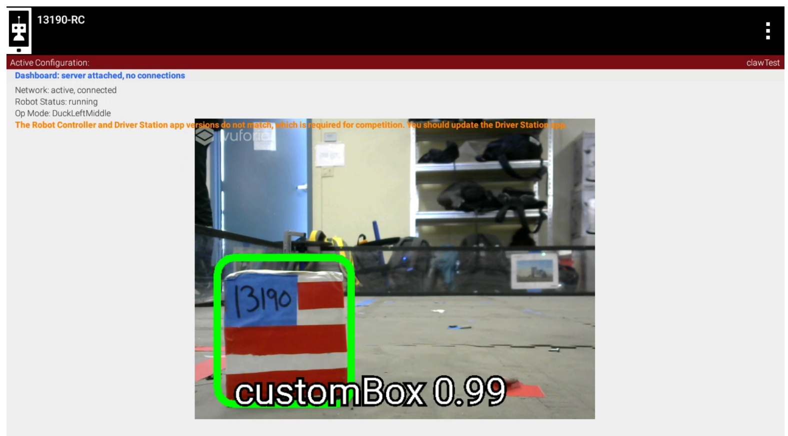
In order to implement a reliable Tele-op script, the workload was divided into 2 pieces: the driving portion and the intake portion. Once the programming of each portion was completed, we tested each program and revised our program step by step so the robot would move as intended. Different people who have different interests and talents in programming worked together to create a complete TeleOp code. After this, the code was reviewed, analyzed, and tested on the physical robot. From there, the team meticulously combed through the code, specifically hunting for bugs, defects, and details that could be optimized further. The robot utilizes a dropping scoop mechanism and uses surgical tubing for the intake. On our robot we used multiple sensors as well as encoders on all of our drive motors.

In order to reduce the mistakes made by human errors and to increase the efficiency, we developed an algorithm that positions the intake and arm in a pre-programmed position so drivers can easily place freight in the top level as quickly as possible. We achieved this through the use of an encoder that finds the position of the arm and an algorithm that repositions the arm in case the arm exceeds the target position. When delivering ducks during endgame, the compliant wheel used to spin the carousel spins linearly. This allows drivers to score more ducks in a shorter amount of time and not have to worry about the ducks falling off if the carousel spins too fast.

Autonomous

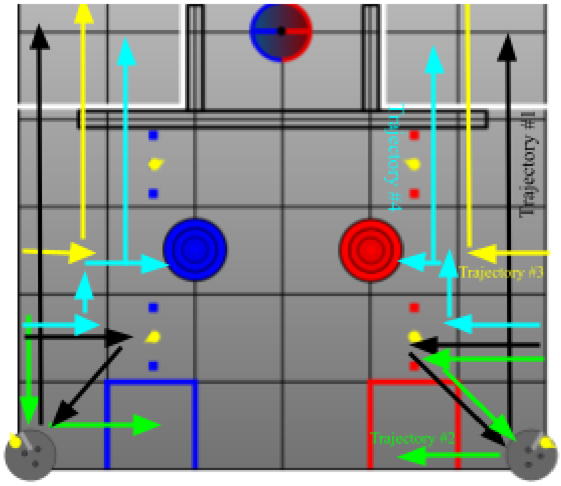
At the beginning of the season, we saw that a significant amount of points can be scored in the autonomous period of the match. We then decided that we needed good and effective autonomous scripts. At first, we planned to only complete the duck delivery and park in the warehouse. However, as our software skills improved, we decided to implement three other autonomous scripts that we avidly use depending on the strategy. We make use of GitHub to share and collaborate among the members of the team, especially through the sharing of code and the utilization of external libraries and algorithms, such as RoadRunner. However, most of the code was developed in house with our own algorithms.

In order for the autonomous section of programming to work, we first did odometry, testing out the possible ways that the robot can travel. We used a sorting algorithm to determine the most efficient path that the robot could travel on the field. However, due to time constraints, we had to use a heuristic method, (thus at the cost of efficiency) from the duck carousel to parking. Using TensorFlow and computer vision, the ducks and freight can be detected physically; the robot then utilized an attached Logitech webcam to look at the images on the sides of the field. Through testing and modifying the code, we had to make sure that the robot stayed within the parking lines. The team made sure that the robot stayed within the lines of parking. It also took trial and error when timing the spin of the duck carousel and comparing it to the motion of the wheel. Overall, making the autonomous section work was difficult, but even more rewarding. We fully utilized the encoders to use position tracking, which would determine the coordinates of the robot. This helped us easily move our robot to the desired location. We decided to use spline paths due to the fact that it adds fluidity and speed to the movements of the robot. We realized that this would help us save time, and therefore chose this method to use while running the autonomous program.

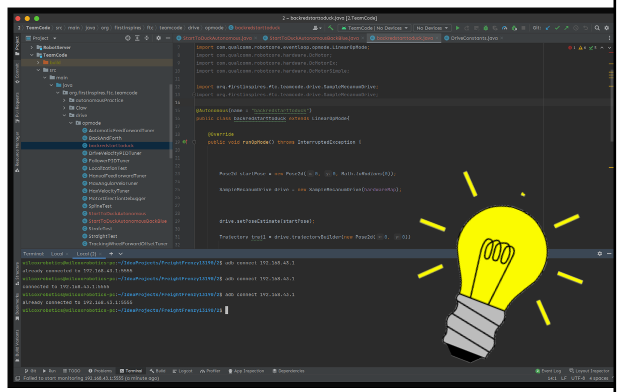



Sensors used: We use a variety of encoders and sensors to receive precise readings regarding location, occurrences on the field, and information about the internal mechanisms of the robot. First, each drive motor was equipped with an encoder which each contributed to specify the precise location of the robot during the autonomous stage. This allowed us to get accurate and reliable readings which would contribute to our consistent scores. Next, Our bot makes use of a camera to find the team shipping element location and use it to deliver the preloaded freight onto the corresponding level of the alliance shipping hub. We use a program called Tensorflow in conjunction with vuforia to calculate this information. Also, our bot uses a gyroscope built into the control hub to turn effectively. This is used alongside roadrunner to achieve phenomenal autonomous movements. Last, in order to move the arm to correct position consistently and reliably, the bot has an encoder connected directly to the axle of the arm. We cross check the reading by using the encoder on the motor powering the rotation of the arm. Both these encoders help us retrieve accurate data which contributes to driver aid and arm movements during the autonomous period of the match.

We used the RoadRunner library. RoadRunner is a library designed to create autonomous robot movement. This allowed us to plan trajectories for the robot to follow autonomously. An example of a trajectory is moving from the starting position to the duck carousel while rotating the direction the robot is facing. RoadRunner allowed us to do all of this in one movement. RoadRunner allowed for precise movement, allowing us to score the duck carousel, as mentioned above, and the parking stages.




Autonomous Diagram



↑The code for the duck carousel in IntelliJ.

## Overall Challenges and Lessons Learned

Communicating with the hardware team and adjusting the settings of the robot was the most challenging part of the project. The robot would not move properly or sometimes break because of the lack of our communication, but we managed to learn the importance of communication and teamwork and were able to implement code that fits with the robot hardware. One issue we ran into was the motors being reversed and the correct coordinates not being defined properly, resulting in the robot being unable to drive. To resolve this issue, the motors had to be manually checked one by one and aligned with the code. Collaborating with the hardware team was another challenge, as both teams worked at different rates. Due to how quickly the hardware team changed designs with the scoop mechanism, the whole software program had to be redeveloped repeatedly. Thanks to the prior robotics experience of several on the software team, they were able to keep up with the hardware team's modifications. In addition, when writing the autonomous programs, we realized that it requires a lot of trial and error, as the drive constants the robot is set to run on were slightly off, and had to be changed based on our observations when testing. This trial and error process was frustrating at first, but we quickly realized how important it was, as just a little adjustment would mean smooth sailing for the rest of our autonomous paths.

Team 13190 has been riddled with adversity, with certain members having to mentor the younger ones, members juggling several commitments at once, and several redesigns of the robot. Throughout the months-long process of competing, we have learned several important lessons. One is the importance of testing and collaboration; it has been a great boon to the team as a whole. Most importantly we learned how to practice gracious professionalism; while competition could get heated, it is important to remember to stay courteous towards the opponents. The point of the competition is to learn about technology and robotics, and to have fun. It is the team members, each of whom with their own talents and quirks, that brings the robot to life; it is the spirit of engineering that is the most important part of this competition. We thank all our sponsors, volunteers, coaches, and members for making this competition truly more than just robots and into memories that would resonate with us.

## Software Used

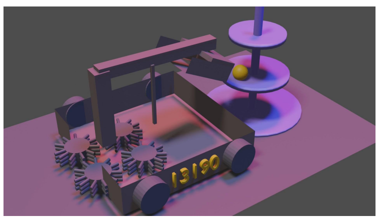
**CAD** - or Computer Aided Design is the process in which one uses software to model physical designs.

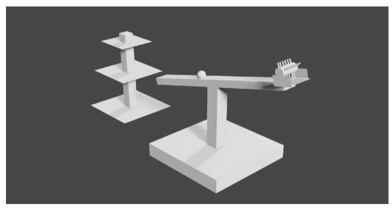
“Through CAD, our hardware team can plan and share ideas to the rest of the team. We ran simulations and tested to see if parts would fit and work the way we want them to on our bot. Instead of wasting time and money on a prototype which does not work, we can preemptively diagnose and solve problems virtually.” - Ayan, team captain

The team uses two different CAD software for two distinct use cases. The team uses Blender for concept prototypes and designs, earlier in the engineering process, while Fusion 360 is used later in the engineering process to create detailed blueprints after a general consensus of the basic design.

**Blender**, a free and open source 3D modeling program, was also used to model prototypes and create animations of the robot. This was especially useful when designing multiple concept designs for the robot, and making simulations to test their functionality. The team utilized Blender to make different prototype designs, before choosing a final design to work on in Fusion 360. Blender is used to test out different chassis and intake designs before making a more detailed and accurate model.

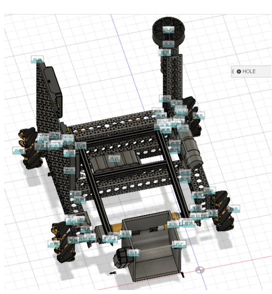
-Used to make concept designs and prototypes.

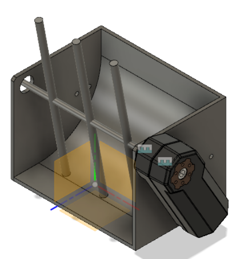


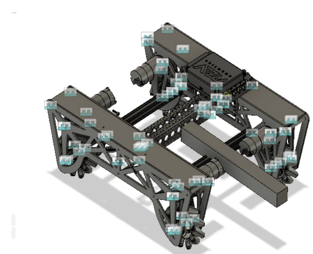



**Fusion 360 -** Team 13190 learned about CAD, using Fusion 360. Fusion 360 was especially useful for making blueprints, a framework for physical construction. As a 3D printer is available in our room, Fusion 360 allowed for the precise measurements to 3D print items for the robot. Fusion 360 is used for more detailed blueprints, after a basic design of the robot as a Blender prototype is chosen to be built. It also allowed team members to make detailed designs and prototypes, without having to build physical models. Fusion 360 allows for creation of detailed and precise models, which can be used as a direct model for what we would physically build.

-Used to make detailed and precise blueprints.







**IntelliJ**, an IDE for the Java programming language, was used to code the software for the robot, due to how it has more features than Android Studio, including Github integration. The IntelliJ IDE is also efficient and lightweight, while still having a large amount of features. The team uses the Github repository integration feature in IntelliJ, for greater collaboration among the different software members on the team.  The IntelliJ software also has built in tutorials and a more user-friendly UI, making it optimal for the newer members of the team, who have never used Java. IntelliJ also allows new members to improve their fluency in Java programming and code, rather than simply dragging and dropping blocks.

-Used to code the robot in Java for both the Autonomous and Tele-OP phases