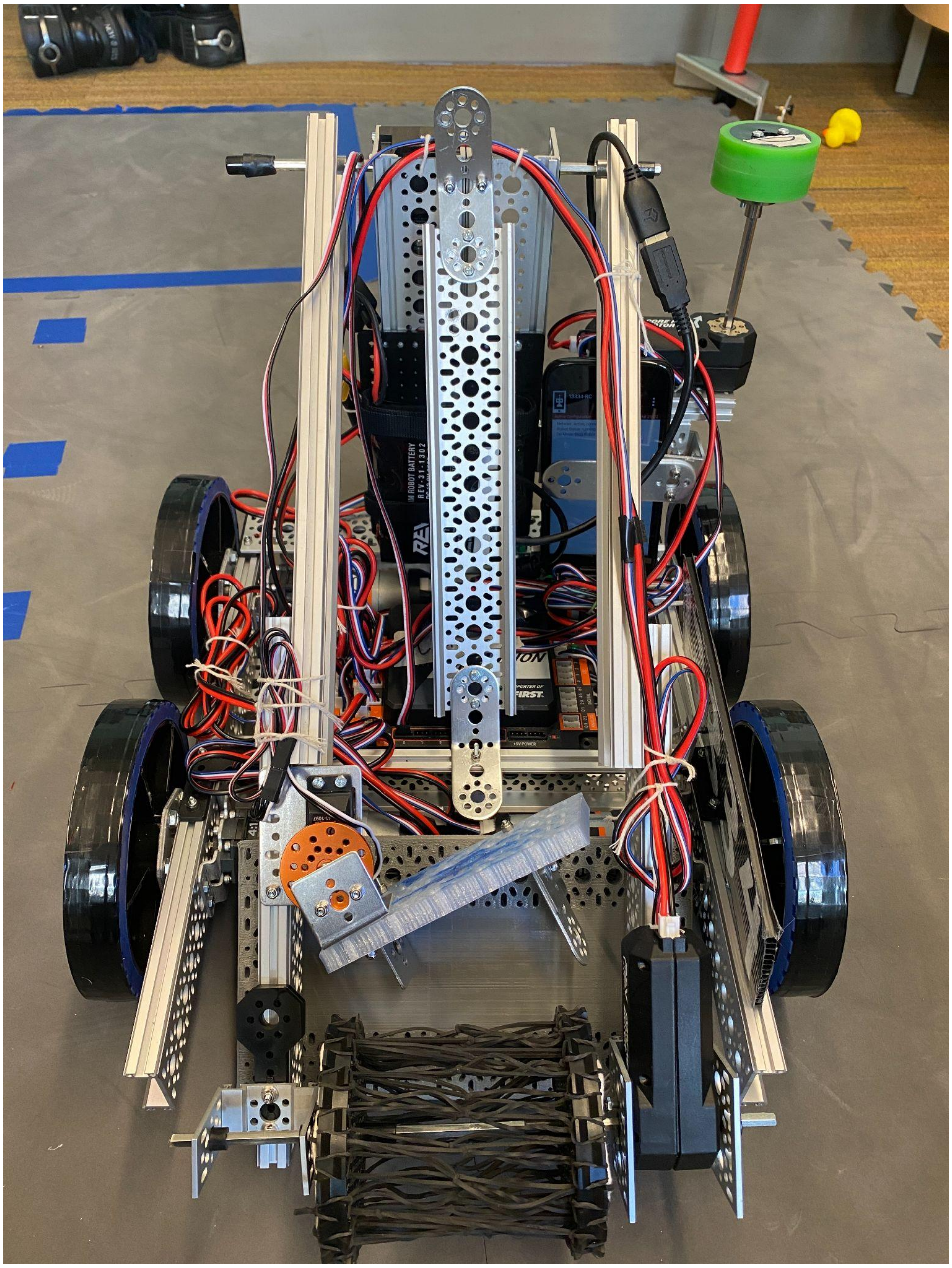
# Engineering Portfolio

2021 - 2022



SPENCE LIGHT SABERS

13334

# Meet the Spence Light Sabers!

Jannah

Sophie

Alyssa D.

Ushrat

Syd

Shay

Alyssa S.

Gia

Lidia

Ari

The Spence Light Sabers team was founded in 2018 and first competed in the *FIRST* Tech Challenge Rover Ruckus Season. This season our team is composed of 10 students from 10th to 12th grade who all attend Spence, an all girls independent school in New York City. Although most members of our team did not have prior robotics experience, we are now a closely knit team brought together by our common love for robotics. We have had teammates from 8th grade through 12th grade of different cultural and ethnic identities.

Although we did not compete last year because of our school's COVID-19 restrictions, we used remote learning as an opportunity to educate ourselves on Java, CAD, and other components of the FTC robotics process.

Our team's aim is to inspire and encourage women and gender non-conforming students to pursue STEM, a largely male-dominated field.

# Team Plan

## Our Plan: Sustainability, Self-Taught Skills, & Inspiring Young Women in STEM

Our Team Plan is to create a sustainable team, teach ourselves many new skills each season, and inspire young women to pursue STEM.

In 2018, one of our team's initial concerns was its sustainability due to a lack of interest in STEM at our school. With only four members, two of whom would soon graduate, we worried about our team's future. We took initiative in the offseason following our first competition, and devised new recruitment strategies. We grew our team by hosting interest meetings open to all Upper School students, presenting our team to the Computer Science I and II classes at Spence, and actively recruiting students from our middle school's *FIRST* Lego League team (The Spence School educates K-12). We have also presented to 8th graders on Upper School Visiting Day, where 8th graders have the opportunity to visit and learn about Upper School clubs. We particularly emphasize that students do NOT need any prior STEM or robotics experience to join. This early exposure to our team ensures freshman recruitment in the following year. Since membership is no longer a concern, we have now focused our efforts on building foundational team skills. We have created guidelines, references, and video tutorials geared towards teaching new members how to research, write a notebook entry, design, code, and engineer. As our team grows, we will ensure everyone begins with the same basic skill set and has the resources to develop their own additional specialized skill set in disciplines like coding or engineering.

As a student-run team, we strive to teach ourselves many new skills each season: CAD, Java, and the engineering design process. This season, we met one of our long term goals of switching from Tetrix to REV parts, which has allowed us more flexibility in our design and building process. Currently, we are in the process of teaching new members Java programming through classes we hold twice a week that we call "Collab Coding Classes." In 2018, when only two out of the four members on our team had previous coding knowledge, we were limited to using block code. However, we have since expanded our knowledge of Java syntax which allows us more freedom and autonomy in our coding.

Finally, our team's aim is to inspire and encourage women and non-gender conforming students to pursue STEM, a largely male-dominated field. Our solution is to spread interest in STEM and *FIRST* among members of our school's community. We have recently achieved this goal by working with admission to advertise our robotics team to incoming students, mentoring our own school's FLL team, and by bringing in a Spence alum who currently works in robotics to inspire our generation of Spence girls in STEM.

## Team Goals and Direction

*Note: Goals we achieved this season are* **bolded**

| Goal | How to Achieve | Challenges |
|---|---|---|
| Find a Mentor | Find alumni who graduated from Spence and work in the STEM field<br><br>Look at mentors provided by *FIRST* | Since our meeting schedule is inconsistent (especially with changing COVID regulations), it would be difficult to have a mentor present for all of our meetings<br><br>We are unable to find someone at school who has the time to mentor our team |
| **Expand Outreach** | Research different ways to get involved in our local community in NYC relating to STEM (i.e. helping organizations that teach girls how to code)<br><br>Collaborating with other schools to better our and their teams<br><br>Reaching out to Spence's Lower School to promote STEM to a younger demographic of women | Each member has a vastly different and busy schedule, therefore it is hard to collaborate with more communities<br><br>Our communication with other schools is limited<br><br>The Lower School runs on a different schedule, making it difficult to meet with younger students |
| Split into Sub-Teams | Have every team member be specialized in a certain field, i.e. | Not everyone has mastered the basics (since we have so many |

| | | |
|---|---|---|
| | engineering, coding, etc. | new members), so we will need more time before all members can develop in a specific area of expertise |
| **Switch to REV** | Research REV parts (i.e. names, how to use, etc.)<br><br>Purchase basic parts for future use and the parts we need for our robot design | Since we have grown accustomed to using Tetrix parts, it will be difficult to adjust to using REV parts<br><br>Buying a lot of new parts is expensive |
| **Learn Java** | Learn what object-oriented programming is and therefore how to conceptually code in Java<br><br>Watch tutorials on Java and look at the Java dictionary for FTC to learn the syntax | Many of the people on the team do not have any coding experience, so jumping straight into Java would be difficult<br><br>It takes time to learn a new language and teach it to the whole team |
| **Grow the team** | Host more recruitment meetings (i.e. hosting interest meetings, presenting to the computer science classes at Spence, etc.) | There is not that much interest in STEM at our school<br><br>Our team is high commitment, so some members join and then drop out of the club |

## Robot Design Process

Prior to this year, we did not have a clear, reliable process for designing our robot. Since we were inexperienced, we mostly relied on trial and error and improvised our robot design as we built. Our team's growth this year encouraged us to find a routine way to design our robot so that everyone's ideas could be heard and employed to create a well-designed, efficient robot.

As our team is fully student-run, we had to structure our current design process on our own. Our process is heavily based on the engineering design process and our personal needs.

First, we identify the challenges of the FTC season competition. We make realistic and achievable objectives for the gameplay given our level of experience and

limited time. As we devise our game strategy, we task each member with individually designing a sturdy foundation for our robot. We present our ideas the following meeting and choose a base to work off of and then start planning our intake, outtake, and attachments. We continue to assign weekly, independent assignments where members brainstorm, illustrate, and detail designs for specific objectives (i.e. designing an arm to place a shipping element onto the shipping hub) and then individually present ideas to the team; these show-and-tell style meetings enable everyone to be up-to-date on our design and practice their public speaking and presentation skills.

Eventually, after innovating and proposing multiple designs, we decide on a single idea. We discuss the pros and cons of each design, and we are often tied between a few designs. For example, one arm design might be sleek but heavy while another arm might be lighter but is less sturdy/reliable. We do our best to combine the designs. Finally, we take our separate part designs and figure out the best way to orient them on our foundation. Our end goal is to have an efficient, functional, and structurally sound design.

## Team Management

Our team meets twice a week during the school day. We also meet everyday afterschool for a few hours. As every team member has a different schedule, members commit to attending at least one of the two school day meetings and are encouraged to join after school meetings when available. We also do asynchronous work outside of our meeting times (see Robot Design Process above).

After finalizing our design, we use our after school meeting times to build the robot. Whoever attends a meeting builds what they can and writes a notebook entry to update the rest of the team. The school day meetings are used to problem-solve issues from our after school build meetings.

Once the building process is over, we begin to code. This year, since we transitioned from block code to Java, we dedicated time during our after school meetings to self-teach ourselves Java. Those who attend these longer meetings also test the code and practice driving the robot. In order to teach ourselves an entire

<u>Hardware Classes</u>

*Template you should follow: HardwarePushbot*

1. Change file name and match class & constructor name to said name
2. Declare your OpMode Members
    - “public DcMotor leftDrive = null;”
    - public [CLASS NAME] [name] = null;
3. Initialize OpMode Members
    - “leftDrive  = hwMap.get(DcMotor.class, "left_drive");”
    - [name] = hwMap.get([CLASS NAME].class, "[NAME ON PHONE]");
4. Set Modes/Powers/Positions etc. (depends on what you are coding, I’m using a DcMotor class here as an example)
    - “leftDrive.setPower(0);”
    - [name].setPower(0);
    - “leftDrive.setMode(DcMotor.RunMode.RUN_WITHOUT_ENCODER);”
    - [name].setMode([CLASS NAME].RunMode.RUN_WITHOUT_ENCODER);
5. Delete what you don’t need and adjust the comments

programming language, we held meetings in between qualifiers and championships called “Collab Coding Classes.” We learned Java terminology and syntax as well as how to create hardware, teleop, and autonomous classes. We gave ourselves assignments to write our own programs based on what we learned in that class. Our entire lesson plan for each of the classes can be seen in our Engineering Notebook. Examples from our lesson plan can be seen above.

We spend the weeks leading up to the competition writing presentations, award applications, etc. During these weeks, we designate specific jobs to members: practicing driving, creating a presentation, designing a poster board for our team, etc.

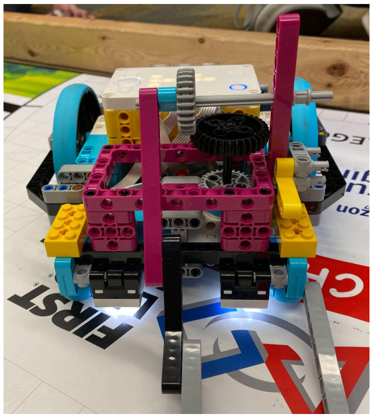
Ultimately, we start the season with individual assignments, transition to more rigorous meetings where everyone builds and codes, and then return to assigning specific tasks to teammates as we approach the competition.

## Outreach Improvement

After our qualifying competition, we started improving our in-school outreach efforts to inspire young women towards STEM—one of our main goals. After competing, we presented our robot and our accomplishments to the Upper School during Gathering via a slideshow presentation and a fun video we put together. Our presentation and achievements were received with much interest from the school. Many more students who were not previously interested in robotics are now considering joining the team next year.

Following our Gathering presentation and video, our school’s admissions office reached out to our team. We worked with our school’s club, Red Door—a student club that works with the administration to host admissions events and give tours to prospective students—as one of three clubs/teams chosen to represent Spence and market our team to prospective, accepted students.

We spent a few of our after school meetings working with and mentoring our middle school’s FLL team. We taught them new ideas and ways to use robot parts so they could implement what we taught them into their own missions. An example of this is when we taught them how to use bevel gears. They needed to make an attachment that moved up and down, but their motors were attached in a way where



they would only be able to move the attachment side to side. We showed them how they could attach 2 gears at a 90 degree angle to adjust the axis of rotation so they would not have to compromise their foundation for a single attachment. Their implementation of the gears can be seen on the left.

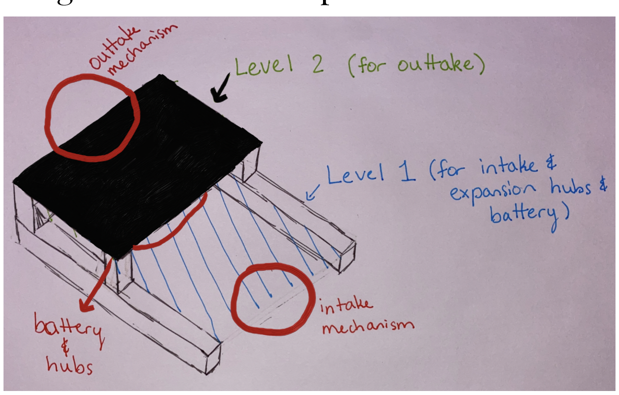
Finally, we reached out to a Spence alum and a female roboticist who work at VIAM Robotics to come and speak to our Robotics Team and STEM-inclined Upper School students. They talked to us not only about what they do, but about how being a woman in STEM has affected their experiences. They were both as impressed by our self-run team as we were by their work, and encouraged us to keep pursuing our passions in STEM. They invited us to tour VIAM and to keep in contact for future outreach opportunities.

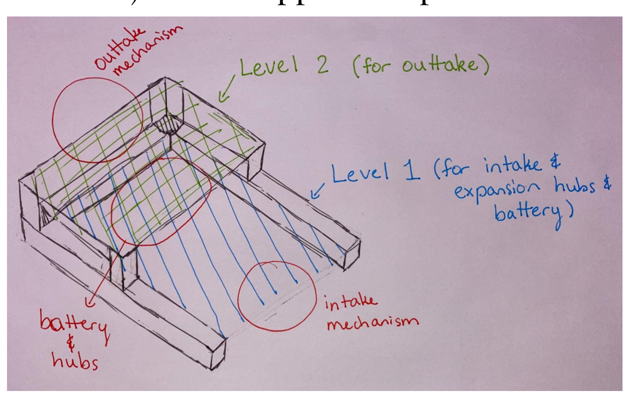
Through our collective outreach efforts, we have learned to market our team, to share our knowledge with younger teams, and learned how professional engineers work through engineering obstacles and how we can apply that knowledge to our problem-solving methodology.

# Designing, Building, & Coding the Robot

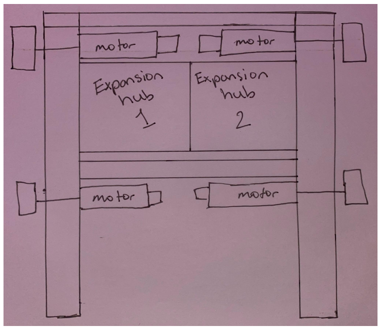
Note: highlighted text represents our final design decisions

The foundation could be 1 or 2 levels depending on the mechanisms attached. The Dual Level design as seen below (left image: how it would actually look; right image: Level 2 is transparent so that Level 1 is visible) would support a separate intake



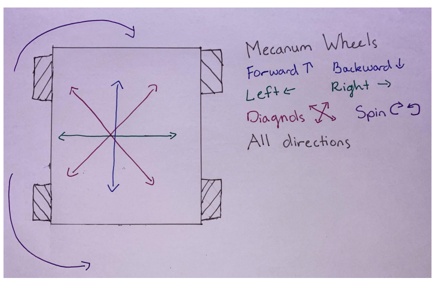




and outake. The shipping elements would be transported from the lower level to the upper level of the foundation, where the robot's arm would push the freight onto the shipping hub. We also had to leave space for the battery and the expansion hubs, which we decided to place directly beneath the upper level and towards the back of the lower



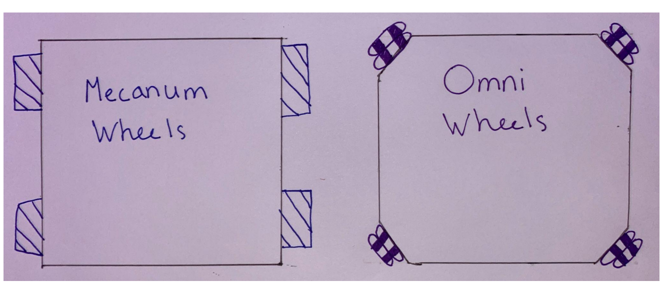

level. Level 2 was meant to be smaller than Level 1 to leave room for the intake mechanism. If the intake and outtake were combined, we would use a Single Level design (as seen on the left). The Single Level foundation resembles the Dual Level design, with the exception of Level 2. An upper level would not be necessary if the shipping elements entered and exited the robot from the same place. Our final robot design combined our intake and outtake into one attachment, so we decided to use the simple yet strong Single Level foundation design.

After we finished designing the base of our robot, we had to decide on which drive and subsequently which wheels to use.  In the past, we have used both a



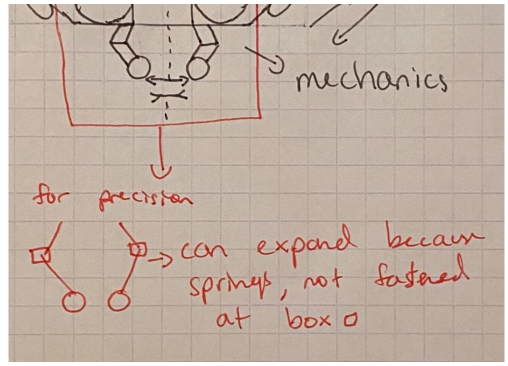

Non-Holonomic and a Holonomic drive (as seen on the left using Mecanum wheels). The Non-Holonomic drive is simpler and easier to control, but the Holonomic drive offers more mobility. Although the Holonomic drive takes practice to perfect, at first, we determined that it is a faster and more efficient option; we decided to implement a Holonomic drive for our robot.

We debated whether we should use Mecanum wheels (left image below) or Omni wheels (right image below) for the Holonomic drive. Omni wheels are lighter but have less traction. Mecanum wheels are heavier, but they provide friction so that the robot does not slide. We wanted to leave the front of the robot open for the intake mechanism and determined that the configuration of the Omni wheels would leave less room for the intake. Since we wanted more traction for ease of

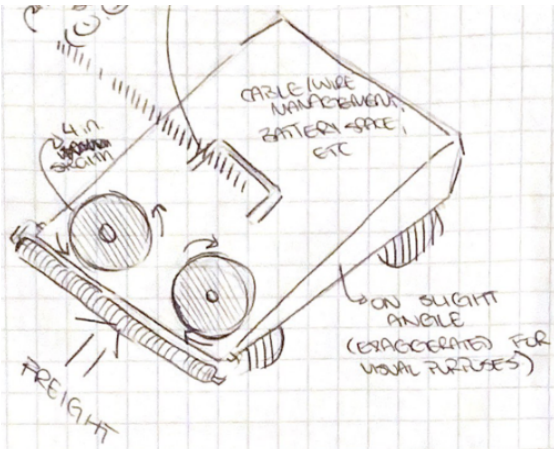



driving, control, and accuracy, and space for the intake, we decided to use the Mecanum wheels. (See "Changes After Qualifiers" for the updated wheel system)

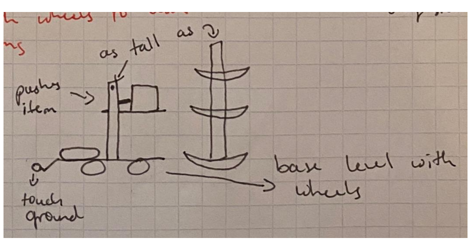
## Designing the Intake/Outtake

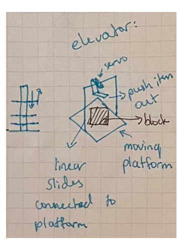


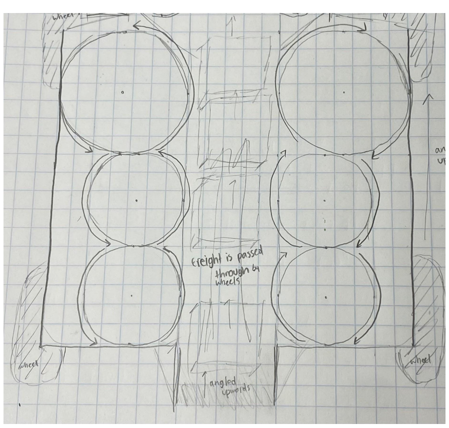

We originally planned to keep the intake and outtake separate. For our intake, we considered implementing a "Stealth Ramp" or "Precision Pinchers." The Pinchers (seen on the right) use stealth wheels that turn clockwise to bring freight inside of the robot; the Pinchers would have a spring-like tension so that they could easily expand to let freight enter and bounce back after the freight passes through. The Stealth Ramp is a ramp lined with stealth wheels (seen on the left). The issue with the stealth wheels being directly attached to the sides of the ramp, however, is that their fixed position would prevent the robot from being able to intake different-sized freights. The Precision Pinchers allow for different freight sizes while the Stealth Ramp does not. We decided against both the pinchers and the stealth ramp as they would have required additional and overly-complicated mechanisms.

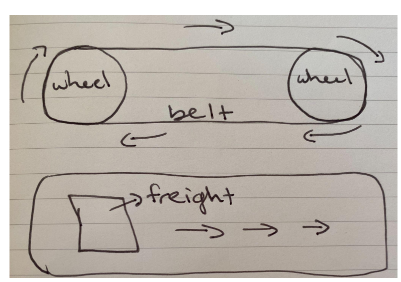



For the outtake, we considered using an Elevator, Treads/Stealth Wheels, or the "Slanted Forklift." In the elevator design, the freight would enter the robot, and an elevator (seen on the left) would carry the freight to the desired shipping hub level's height. A servo attached to the top of the elevator (above the freight) would control a rod that pushes the freight out and onto the hub. The elevator would move up and down using linear slides or a pulley system. We have learned from past experiences that elevators tend to be slow and overly complicated, so we decided against them for our robot. Our second idea was to use treads on the sides of the robot or stealth wheels (on the left). We determined, however, that



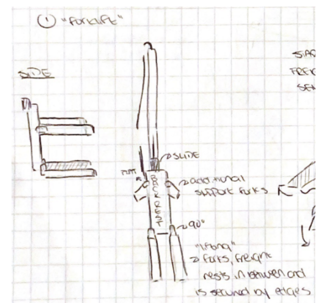



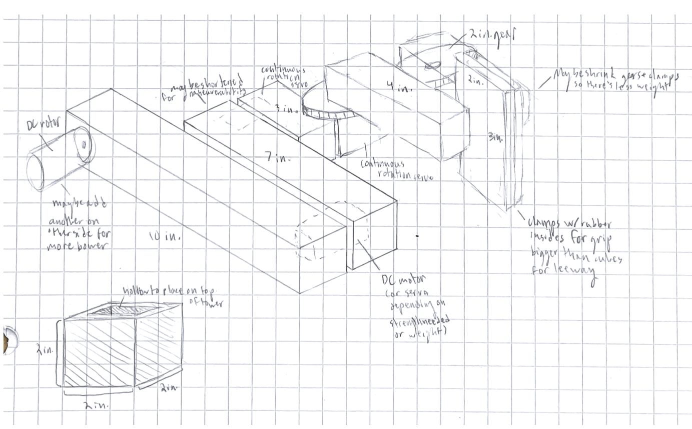
this idea would not accommodate different freight sizes. Therefore, we considered using treads as a conveyor belt (seen above on the right) so the width would not matter. This design would not have been an efficient method for adjusting the height of the conveyor belt to each shipping hub level. Our final idea positioned intake and outake on the same side. We designed the outtake to be on a slant facing towards the intake; the freight would travel up the slanted mechanism through the use of pulleys, linear


slides, or a chain and sprocket and be released at the top of the hub (seen on the left). Our concerns with this design was its structural integrity and its grip on the freight. To combat this we considered using a forklift (seen on the right). The forklift platform

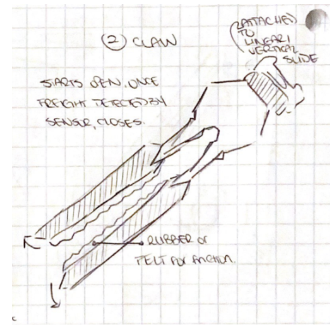


would allow the freight to slide onto it with ease while remaining stable. There would also be safety guards on the sides. This idea is similar to the elevator idea, but it is at a diagonal as opposed to being vertical. A small servo attached to the forklift would push the freight out at the top. Building the forklift on an angle proved inefficient as it required custom parts.

As none of the separate intake and outtake designs appealed to us, we decided to use a joint intake and outtake, which would significantly decrease the time it takes to grab the freight and place it onto the shipping hub. Our designs were an "Arm/Claw," the "Minecart," or the "Spintake." The Arm was a collapsible arm (shown below on the left) that used motors to increase its height incrementally. A claw attached to the top of The Arm would grab the freight, serving as the intake. The Arm would become an outtake when it extended to place the freight onto the hub. Since the arm would have to be positioned on the side of the robot in order to reach the ground, it would cause an uneven weight distribution. We worried that the arm would be

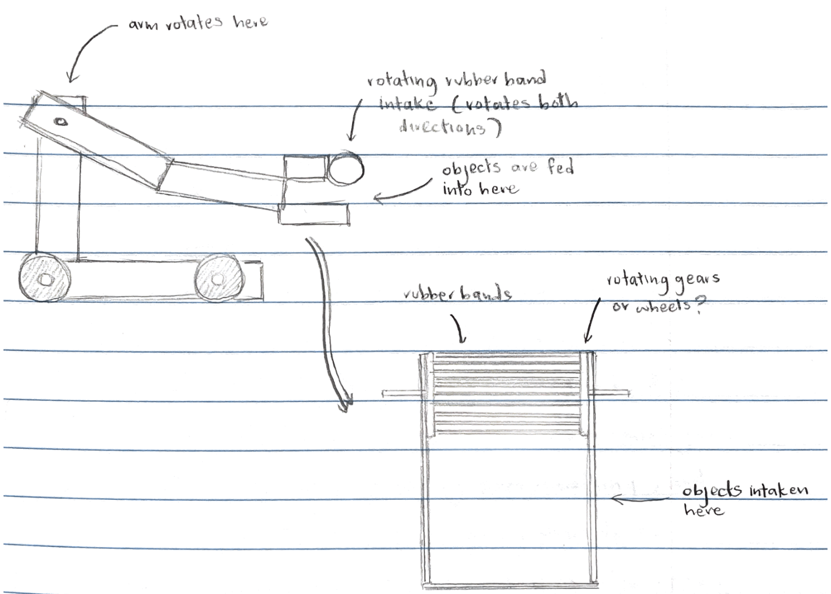



flimsy, as it was heavy and did not have a lot of support. It also used a surplus of parts and would take an excessive amount of time to extend. Our final concern was that the freight could fall out of the claw, especially as the claw would have to account for both cubic and spherical-shaped freight. To tackle this concern, we decided to make the claw similar to an alligator clip (shown below on the right). However, this design would potentially damage the shipping elements. Taking inspiration from our stealth wheel and tread conveyor belt design, we designed a bucket that transported cargo from one side of the robot to the other (shown below on the left). Similar to our previous tread outtake idea, however, adjusting the height of the ramp would be difficult and time-consuming. Additionally, we were unsure of how to get the cargo into the bucket because it had limited entry space for the freight. Having a separate intake to put the freight into the bucket would defeat the purpose of combining the intake and outtake, so we decided not to use the “Minecart” idea.





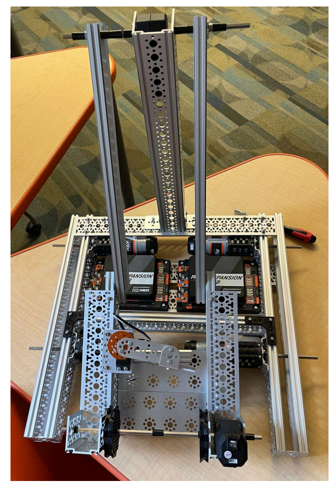
Our final and most creative and innovative idea was the “Spintake” (shown below on the left). This design combined the intake and outtake, thus simplifying our design and creating a more efficient mechanism. It is made using channels that are in the center back of the robot that are attached to an arm. Since the arm is attached to the channels on both of its sides, it is sturdy and stable. The arm is angled halfway through so that it can touch the ground and intake the freight. The end of the attachment is a small platform with walls on its side to keep the shipping elements from falling out.



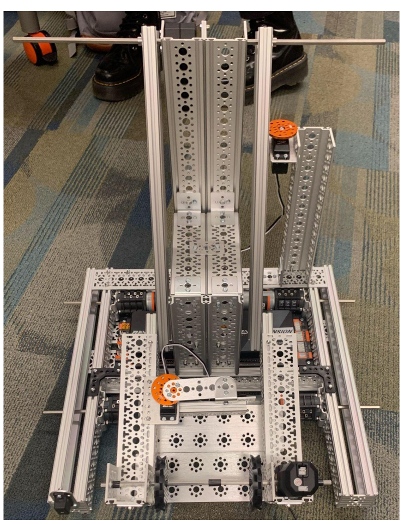

A motor is connected to two wheels where rubber bands are spanned across them with some tension. We decided to use rubber bands since the elasticity of the material provides enough tension and friction to grip the freight. After testing the spintake, however, the rubber bands either snapped or remained too stretched out. To overcome this engineering obstacle, we creatively used rubber hair ties, which don't break easily, maintain their elasticity, and consistently pick up various freights.

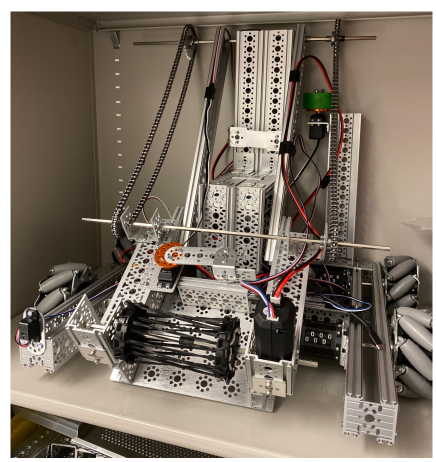
This rubber-banded, cylindrical mechanism spins clockwise so the freight is pushed into the bucket. The arm lifts to the height of the desired shipping hub level and the "Spintake" spins counter-clockwise to push the shipping element out of the bucket. The "Spintake" combines the intake and outtake; the double function of this one part (the spintake) is efficient and innovative, as it decreases the time we spend handling the freight. The elasticity of the rubber bands also accounts for the different freight sizes, since they can bend and fit to the shape of the cargo. Naturally, we implemented the "Spintake" for our final robot design as it is the most effective solution.

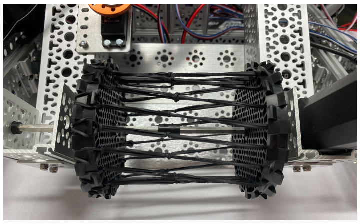
## Challenges when Building the Robot

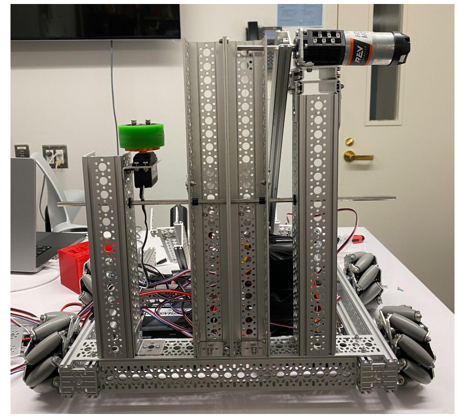


As we began building, we were forced to make small modifications to our design to adapt to the challenges we encountered. While assembling the arm, which was supported by a single u-channel (shown on the right), we realized that the arm's weight caused a forward pull on the robot when the arm was lifted. To solve this problem, we added a second channel and a platform angled at 90° in front of the two channels (shown below on the left). We also attached a chain and sprocket (shown below on the right) to both sides of the arm to further support the motor in lifting the basket. A servo and channel attached to the basket behind the "Spintake" (shown in all images) prevents the freight from falling backwards when trying to exit the bucket. The structure of the rubber bands themselves also posed a challenge; we decided to use rubber hair ties instead of normal rubber bands because they were less prone to snapping, held tension, and were easier to attach to the wheels. The spintake was also too small for all of the freight, so we had to replace



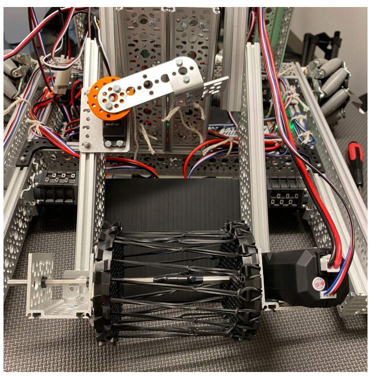


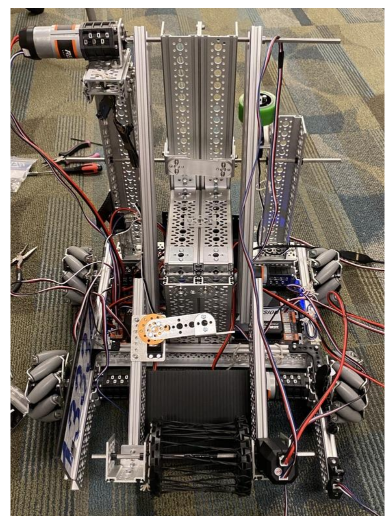


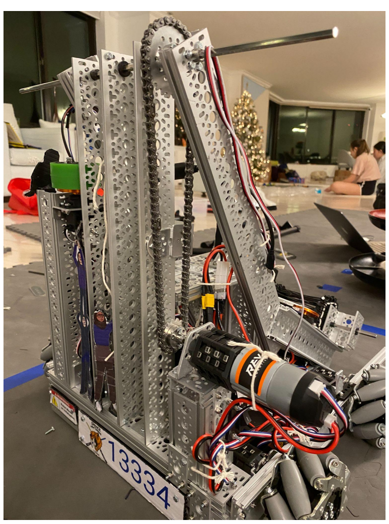


the pulley wheels with larger ones (seen on the left). Our final challenge was finding the perfect angle for our arm, as some angles were too steep for the freight to enter the robot consistently and other angles would not reach the ground. We found an optimal angle once we changed the lengths of the channels.

When we began to code our robot, we discovered issues with our arm design. The arm weighed too much for the core hex motor, so we switched to a DC motor (seen above on the right). We also added supports in the back with a shaft, which gave the robot a more industrial look. The arm was still too heavy after these changes, so we had to slim down the design (seen on the left). We omitted as much metal as possible and replaced heavier metal pieces with black plastic called twinwall. We also made the design neater by removing the chain and sprocket, as we no longer needed it. After making all of these modifications, we realized that our robot was too big and resized it to fit the 18 x 18 inch dimensions (seen above on the right). The robot is more stable and sturdy than before and looks much simpler and aesthetic.



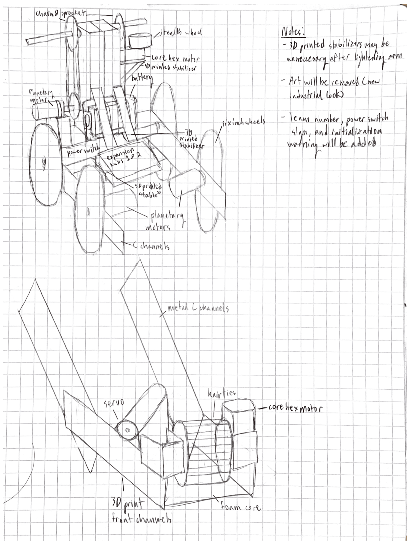




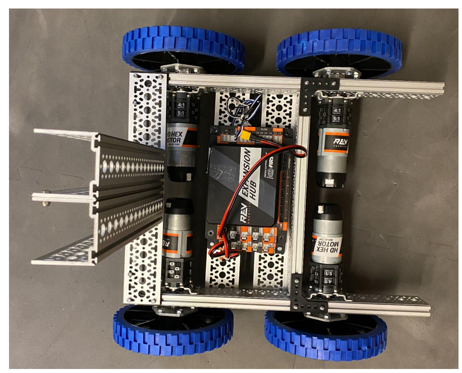

When testing our motor that moves the arm to the respective heights of the shipping hub, it quickly lost power and overheated. Having the motor control the arm at the top was inefficient and ultimately led to an unreliable mechanism. We did some research and decided to re-implement the chain and sprockets (seen on the left) in an entirely different way. In order to relieve the weight of the heavy load on the motor when intaking freight of different weights, we added a chain and sprocket to the top of the arm and shifted the motor down. We also used a sprocket with less teeth for the motor and more teeth for the shaft controlling the arm to increase the power and torque of the arm.

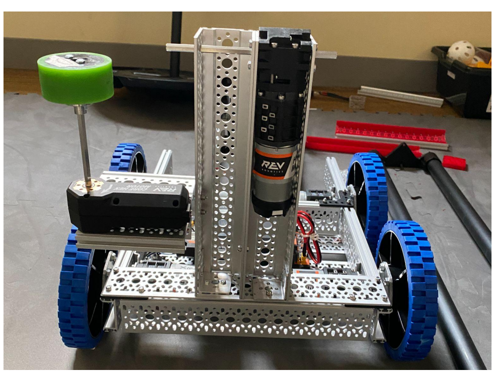
## Design Improvements After Qualifiers

After Qualifiers, we realized there were many opportunities to improve our robot design (our new design plan can be seen below on the left) in order to solve

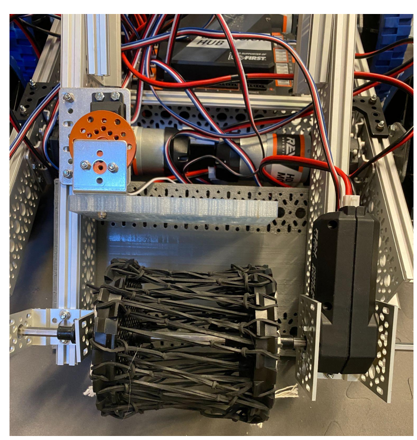



problems we noticed throughout our matches. We experienced difficulty driving over the barriers, the carousel attachment was too slow, and the robot itself was slow. In order to gain enough traction to clear the barriers, we decided to use either treads or larger wheels. As treads would add to the overall weight of the robot, we decided to use taller, skinner wheels that leave plenty of space between the ground and the robot (seen below on the right). Since many points can be scored during the endgame by turning the carousel, we wanted to have a much faster mechanism. To achieve this, we replaced the servo we originally used with a core hex motor (seen below on the left). Considering the fact that our entire robot is slow when





driving we assumed the issue could be any of the following: old batteries, too many motors, robot is too heavy, gear ratios are wrong, and/or issue with code. We tackled these potential problems one at a time. We lessened the weight of our robot by decreasing the overall dimensions and got rid of excess metal parts. Moreover, we designed custom parts in CAD to 3D print and replace the metal and twinwall pieces that were adding to the robot's weight. Designing our own pieces allowed for full customization when it came to specific parts like the base of the arm basket and the servo pusher (seen on the



right). The 3D printed filament is also much stronger than the twin wall we previously used, which allows for more stability and robustness. We also decreased the height of the arm and placed its motor within one of the u-channels (seen above on the left); previously, when the arm was raised at its highest point, our robot would lean forward since most of the weight was concentrated in the front. To problem-solve, we reduced our arm's height to increase the robot's stability, reduce the weight, and lower its center of gravity. The angle of the arm was changed from bending upwards to bending downwards to allow for a flatter platform when delivering freight as well as a more compact arm to reduce the length of our robot.

We also adjusted all the gear ratios for the motors attached to the driving wheels from a 60:1 to 20:1 ratio to maximize the speed. Finally, we decided not to use encoders in the driving motors' code since with this game, speed while driving is more important than precision. Later on, we decided that our arm would still need some support so we attached a single c-channel at an angle, between the foundation and the arm, to provide this. When our robot was trying to turn, the wheels would get caught on the field mats, which we hypothesized was due to excess friction. To counteract this issue, we taped over the wheels with black tape to significantly reduce any friction interfering with the wheels.