METAL MOOSE

1391

1391

ENGINEERING NOTEBOOK

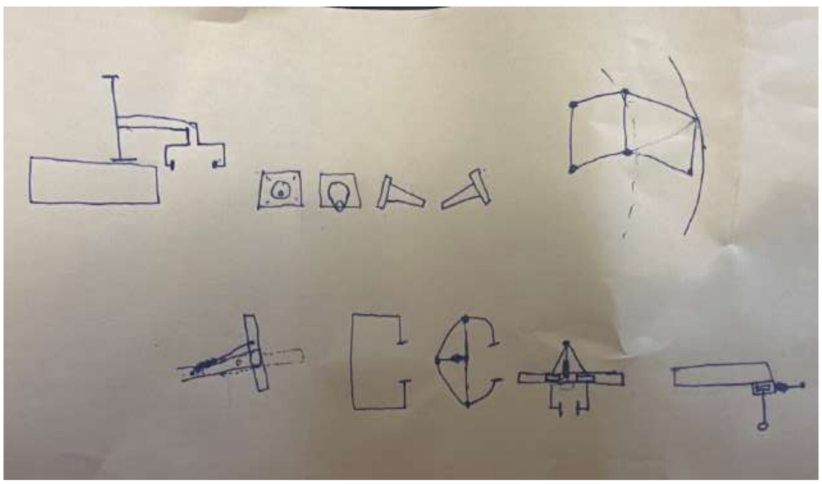
Table of Contents

Kickoff

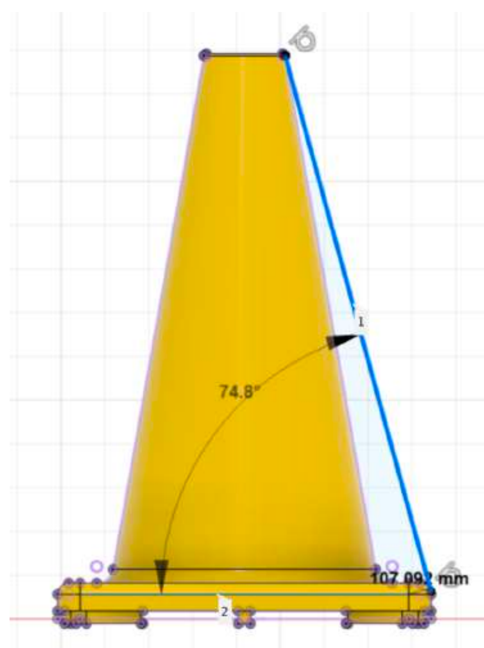
We spent day 1 watching kickoff and then splitting up into small groups. The main goal for these groups was to analyze the game rules, guidelines, field layout and scoring. After 2 hours we met up again as a team, discussed our findings of the game manual and developed strategic scoring imperatives.

Game Piece Analysis

One of the first challenges we tackled was exploring the geometry of the game pieces to inform our first thoughts about manipulating these elements.

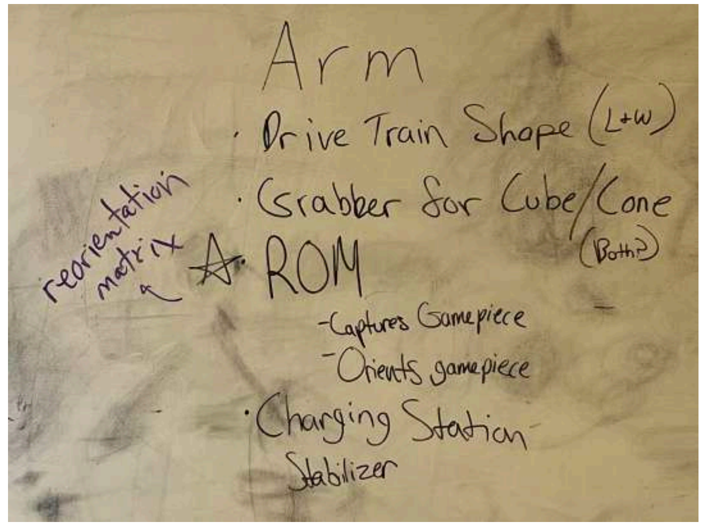


Our findings: This year presented a Interesting challenge. The overall game seemed pretty simple but when it came to taking and placing both game pieces it became a bigger challenge. The cubes and cones did have one common characteristic though; both had a square face of approximately the same dimensions.

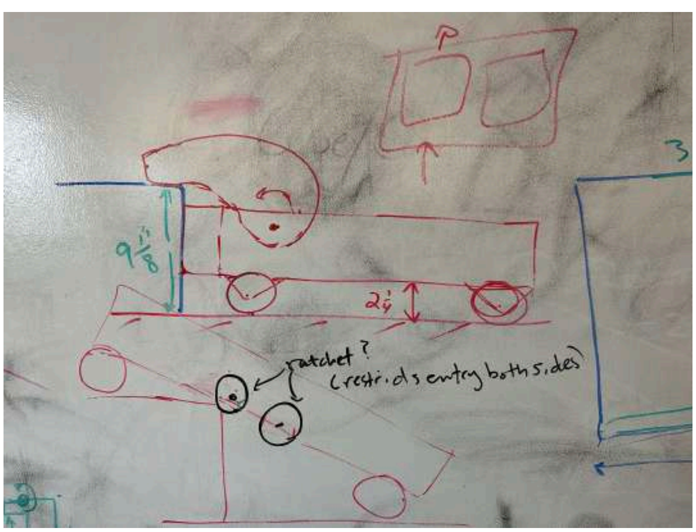


Robot Requirements

Before building anything we started off with a discussion about what the robot needed to complete every task in this year's game challenge. We came up with a couple of key mechanical components for the robot; Drive, Intake / Grabber, Game piece transfer mechanism (reorientation matrix), Arm, and ability to mount the Charging station.

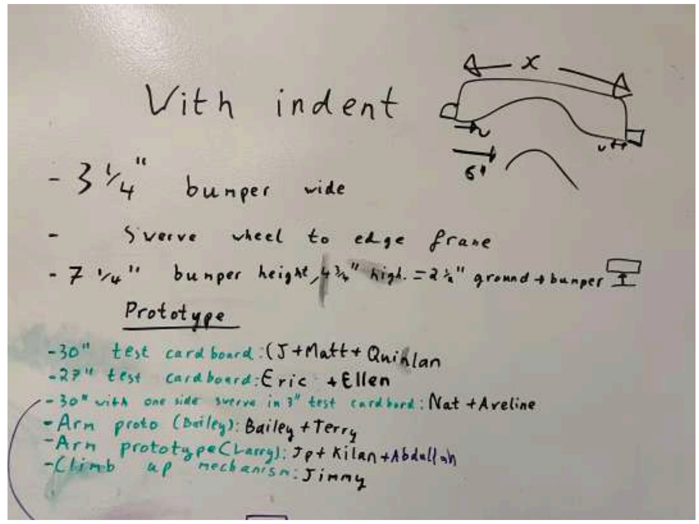


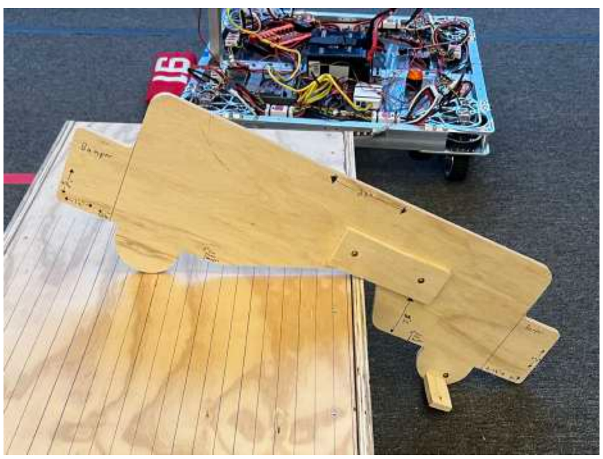
Climb Up Side Prototyping



We decided to start with the most obvious requirement for this year's robot, the ability to mount the charge station effectively, without taking up too much space. This would allow us to facilitate a three robot end game. This came down to two things; robot dimensions or another system to take up even less space. We started by exploring a system to climb up the side of the charge station because we felt it would determine our drive base dimensions.

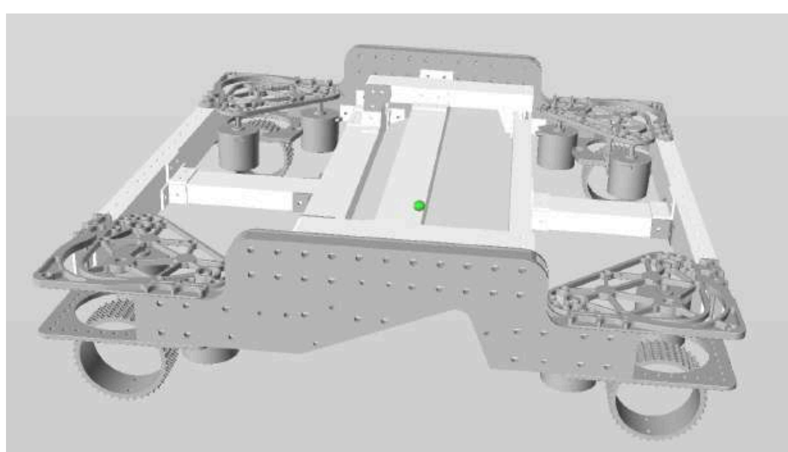
We had already stuck to the decision of using swerve drive based on preliminary game analysis, so we thought a cutout in the robots frame could allow it to be pulled up the charge station with an arm. Everyone was assigned a robot dimension and they attempted to prototype this cutout.





Once the prototyping was over we came up with a pretty solid idea of what the drive base had to look like to allow for this side climb. We made 2D models to show this. Now, we had to figure out how it would actually stay on the charge station. To do this, we explored using a wedge on the edge of the cutout so that when it drove up it would get stuck just above the ground.

Another build branch was simultaneously working on the requirements for the arm to help pull the robot over the charge station with this cutout. To do this they first figured out certain constraints using the test drive base as a model.



PROS	CONS
- Took up less than half the robots size when charging. - Not rely on team for successful charge.	- New drive base geometry + additional arm. - Heavier. - Gap needed in robot frame and bumper, weak point.

RESULT: The idea was there, but we had already spent the first 2 weeks on this idea and needed to focus on more pressing matters. We decided to bench this idea for due to its lack of urgency, and switched to more important mechanical aspects of the robot.

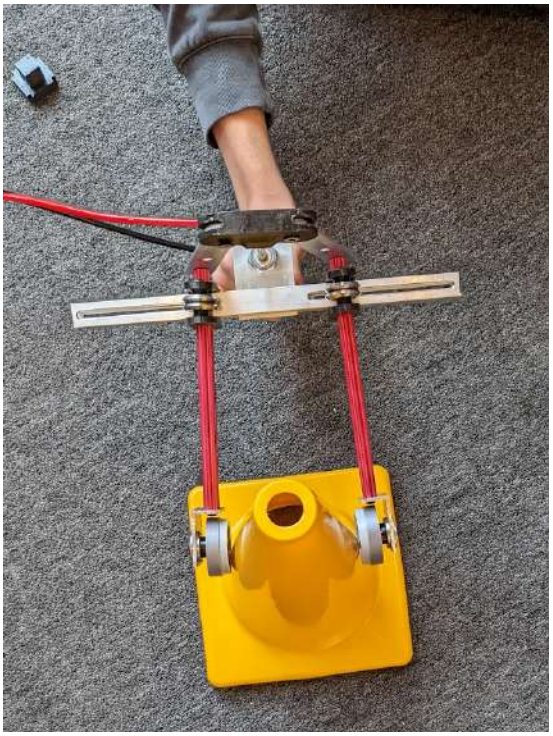
Grabber Prototyping

V1 grabber:

Requirements:

- Grab both cone + cube
- Lightweight
- Simple (more robust and fixable)
- Could align game piece?

Design: The initial design was a 3D printed piece with a piston attached to a C-channel track. This idea seemed to work but a more solid prototype with less friction was needed to decide.

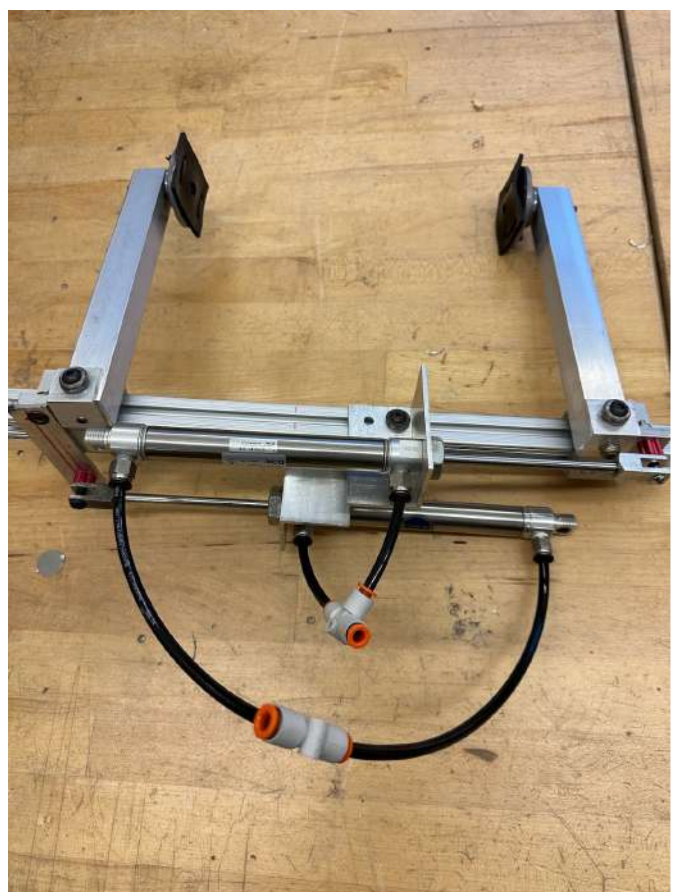


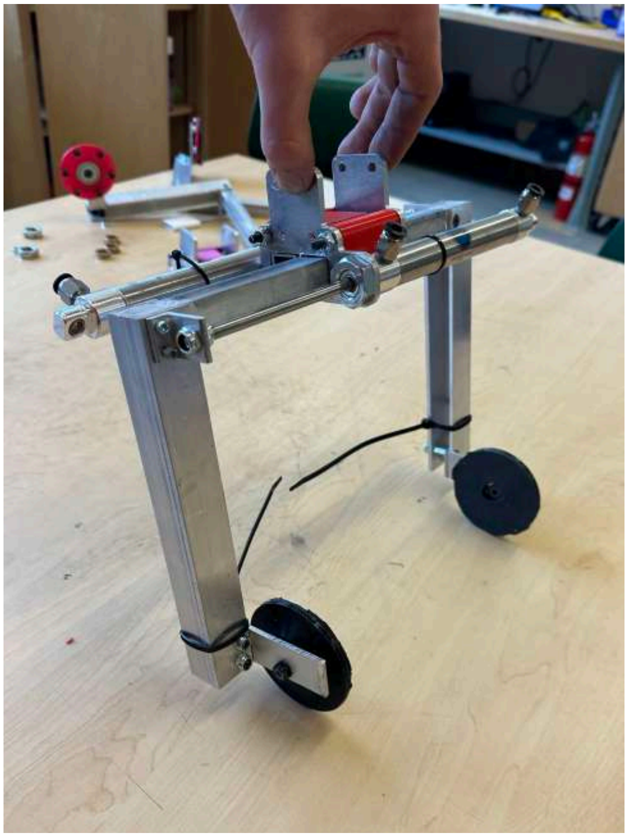
V2 grabber:

C channel with cutouts, piston opens and closes two linkages. On the edge of these pieces of shaft, a bearing attached to a wheel, allowing for the game piece to move freely. (Since in the game piece analysis we saw the cone was naturally bottom heavy we could use gravity to our advantage) This was tested but still had some minor flaws.

V3 Grabber

The c channel was switched to 80-20, allowing the grabber to open and close more smoothly. Also each arm now had its own piston for more strength in the grab. This was an improvement versus the V1 losing strength because the piston was being connected to a linkage, causing some of the force to be lost. This grabber was tested and seemed very promising. It was grabbing great but was a little too heavy.





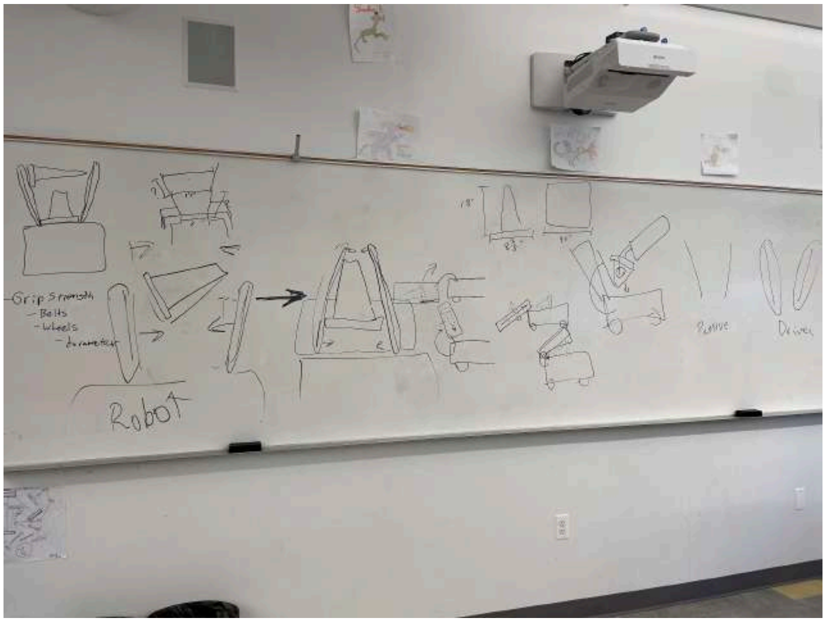
V4 Grabber

The 80-20 was replaced by a lightweight piece of box tube. To keep this efficiently sliding, the box was held by a 3D printed piece with clearance for smooth sliding. The paddles where also changed from wheels to Sorbothane pads. This was because the wheels took up too much space and did not have as much gripping power.

Conclusion:

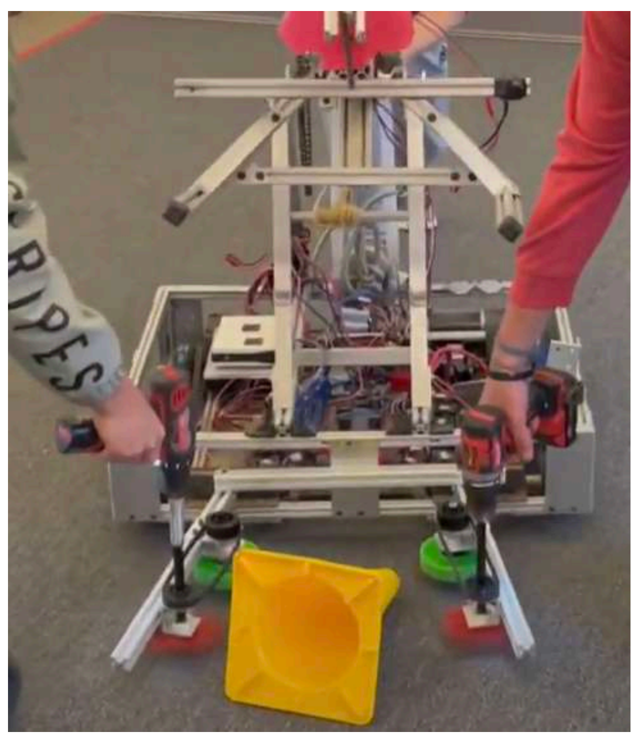
Over the development of the gripper, there were a few main factors we wished to optimize, such as weight, robustness, grip strength, and consistency. Weight was a major constraint as we didn't want something heavy at the end of a long arm. We had many iterations, and it was the part whose development began the earliest and still ended up on the final robot. In the beginning, we were just trying to execute some concept of a linearly actuating gripper with passively rotating wheels on the end, but as we worked more and more on it, the design became more and more intentional until it became the polished final version we competed with during our first competition.

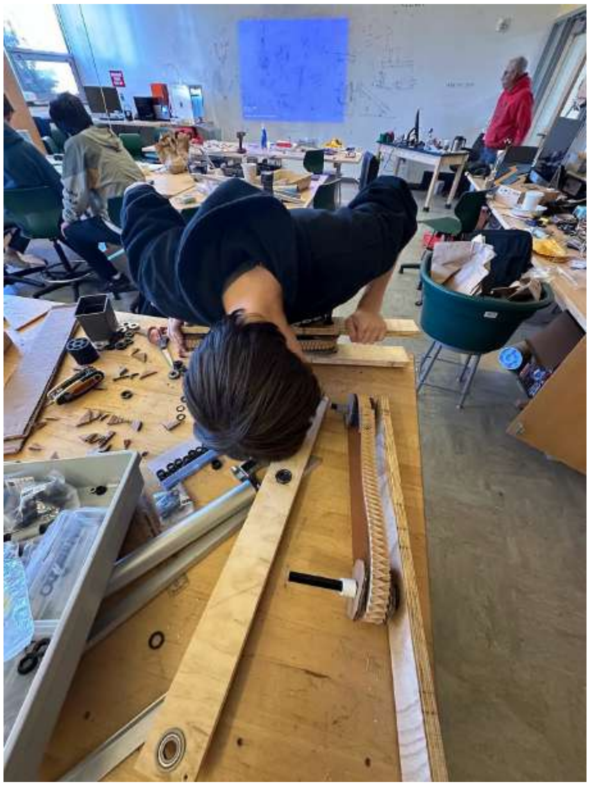
Intake Prototyping



The intake was the biggest challenge in developing this robot. The game pieces were very distinct from one another so we started with conceptual ideas to how we might approach the game piece collection. To do this, we split up into groups for an hour and presented our ideas for how we might pick up the pieces.

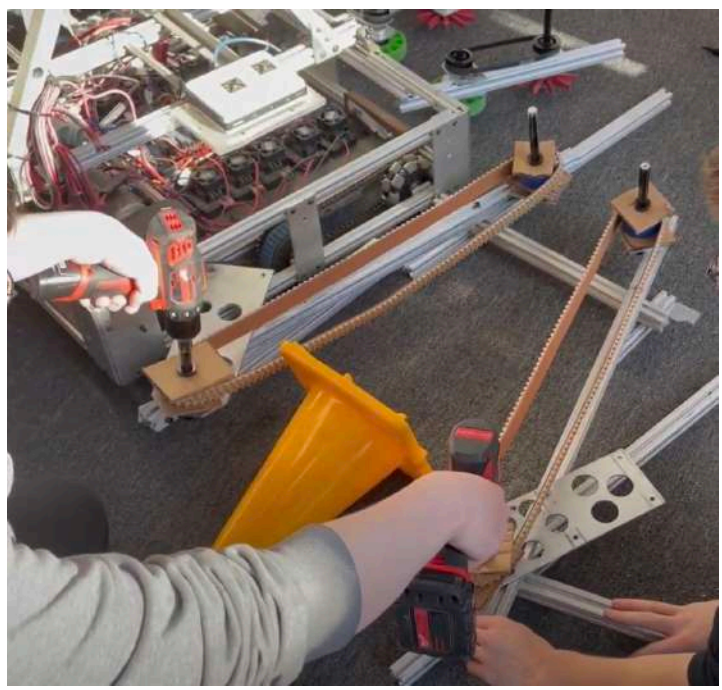
Our first conversation did not really find a obvious solution to the challenge like we had hoped, so we decided to prototype a simple system to intake game pieces. This system was 3 pieces of 80-20 with 4 pulleys and wheels attached to it and 2 drills driving the wheels. When trying this we saw that this testing was too inconsistent / flimsy and did not really provide any useful information.

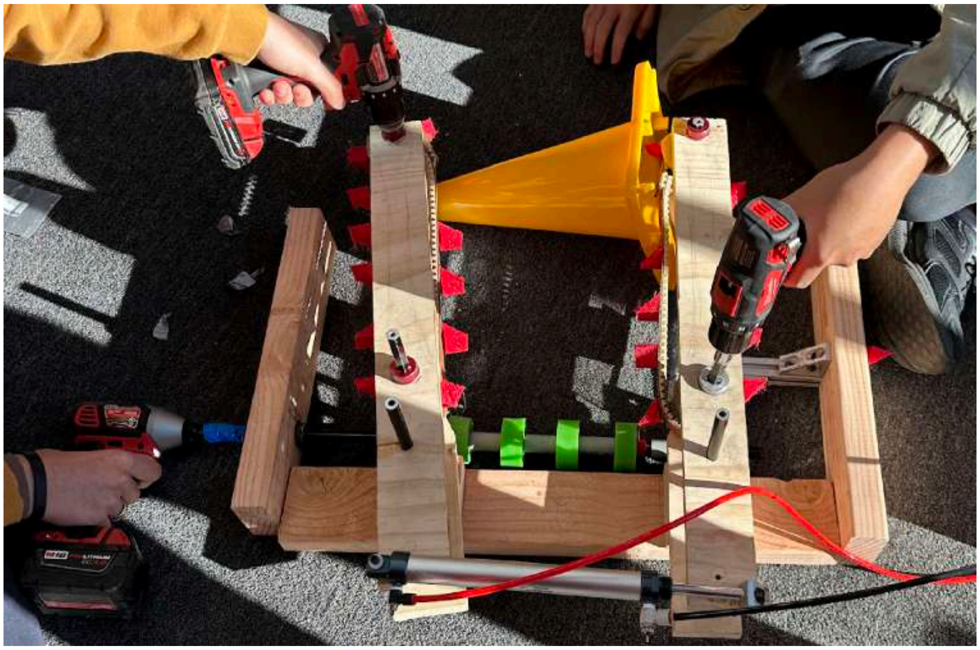




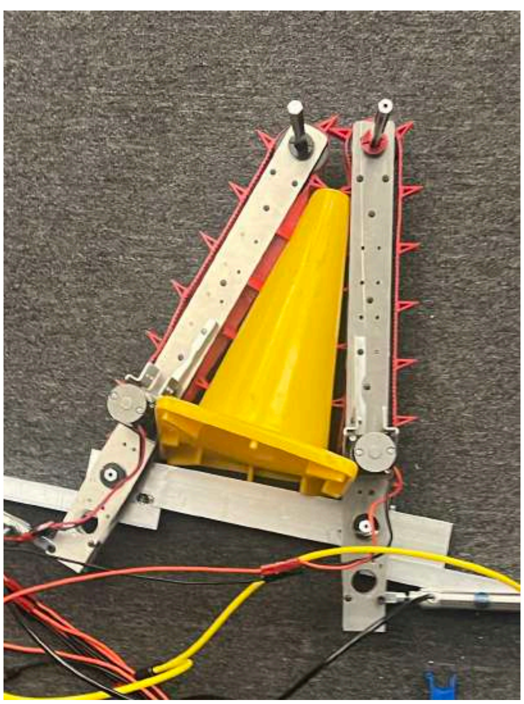
Programmers were getting restless for a robot and started helping build. We needed to get something built to start testing so we started building a stronger prototype out of plywood and belts. This prototype did not really have measurements or documentation but it finally showed forward movement in the "intake rabbit hole" the team was stuck in.

At first this idea was tried on the same 80-20 as the previous intake design. Again, we saw a problem with this lack of consistency in drill operating and human subconscious impact. In this version of the idea two belts would be driven pulling the game piece in and one of the arms would close towards the other. This idea could not really be rated due to its lack of accuracy.





Finally, we took the ideas and made a somewhat solid intake prototype. It was still drill operated but now a piston was used to close the arms and it proved to... still not really work effectively. The belt's teeth were falling off and the pieces were loose. But, we had seen what we needed to see from these prototypes to find a solid understanding of what a working intake would look like.



Belt Intake V3

Finally, the intake made it out of the plywood and drill driven test phase. We started to CAD and machine out the idea. In this version, the idea was the arm would open, and once the cone or cube was somewhere in the interior of the intake, the pistons would close, capturing the pieces and bringing them to the front face of the intake using 3D printed belts.

PROS	CONS
- Cones would come to the same configuration when being captured from their back side. - Allowed for a wide pickup range of game pieces.	- 3D printed belts were inconsistent and were breaking. - Cubes weren't being captured and when being pulled through would get launched out.

Wheel Intake v2

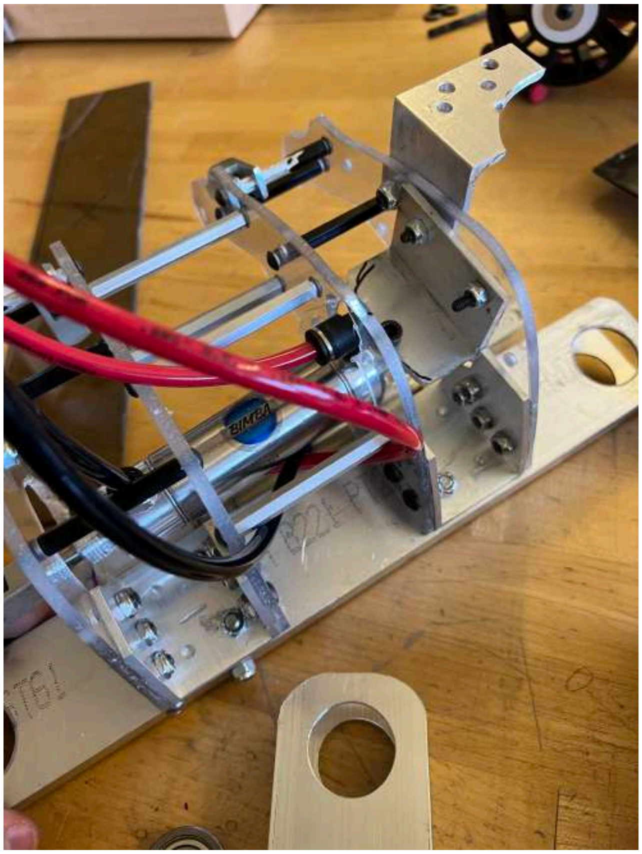
This prototype dealt with the pros and cons of the previous one and attempted to solve the problems. The belts were replaced with wheels. These wheels were 2 layers to avoid the cube being launched out when reaching the end of the intake and they gave more grip on the game piece. This design worked really well and Resulted in both cubes and cones being able to be taken to a fixed position.

PROS	CONS
- Captures and centers both game pieces. - Wide capture range. - Works very consistently. - Can score low if arm breaks.	- Heavy / extend far forward. - Part of intake rides on the ground due to weight. - How would the pistons be mounted?

Other Intake Prototypes

The goal of this prototype was to be able to pick up the game pieces with any orientation and then reorient it. The prototype performed well with picking up cones and cubes but on some occasions would fail. Though this prototype did not work it provided us with key information to better understand how to make a more reliable final build.

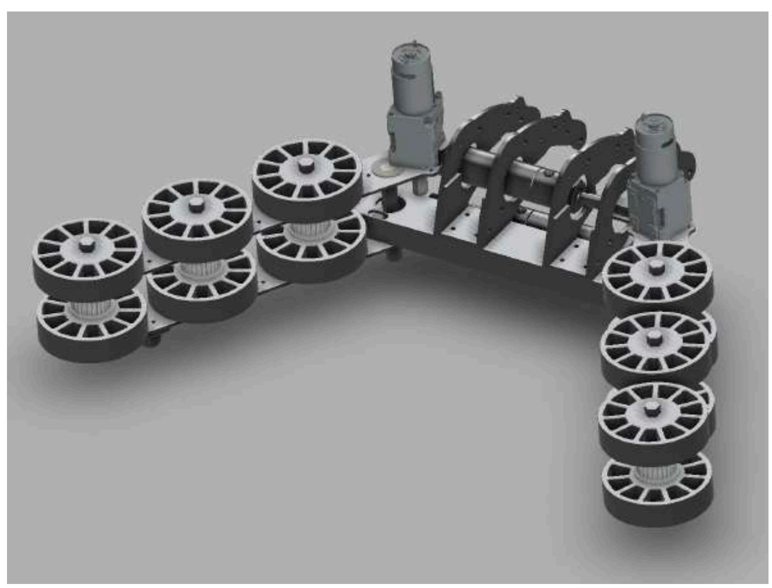
This intake function by lowering onto a cone or cube from any orientation resulting in the hinged protrusions falling underneath the game piece. As a result, the piece would then be unable to move downwards relative to the structure. While this type of mechanism could consistently pick up game pieces, the idea was abandoned because it had no ability to reorient game pieces.



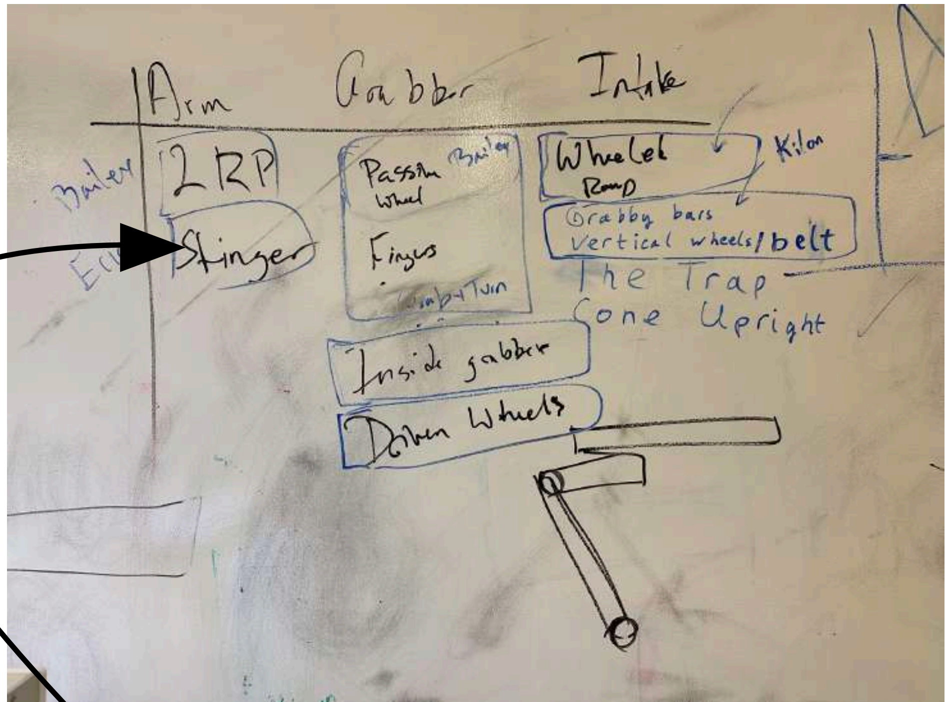
The piston to open and close the intake was placed between both arms once we decided not to bring the game piece through this middle section. We also modeled 4 pieces of plate that would connect the intake to the robot on a rotation point within the body's frame so we could bring the intake up when not in use, and in starting configuration.

Intake Decision:

This model shows the intake's geometry. The arms open 12.5 degrees, allowing for half an inch on either side of clearance from the robots frame. This is to account for the wobble in the Polycarbonate to stay in regulation. When the intake is closed, both wheels touch forming the shape of a cone.



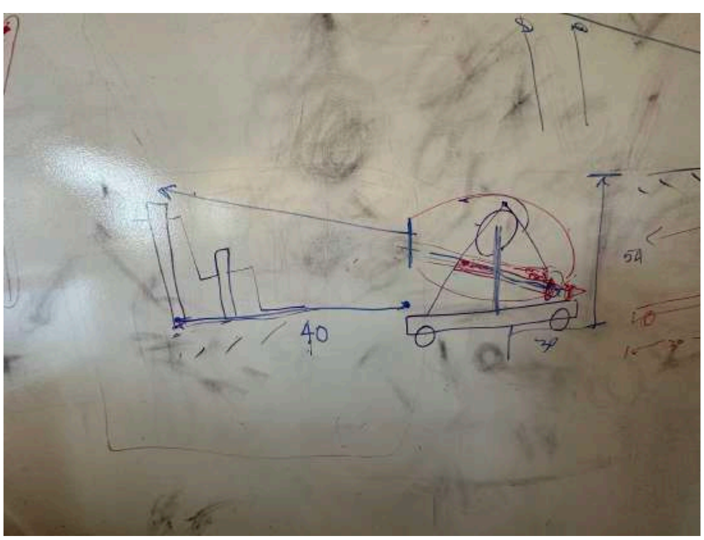
Arm Prototyping



Arm options:

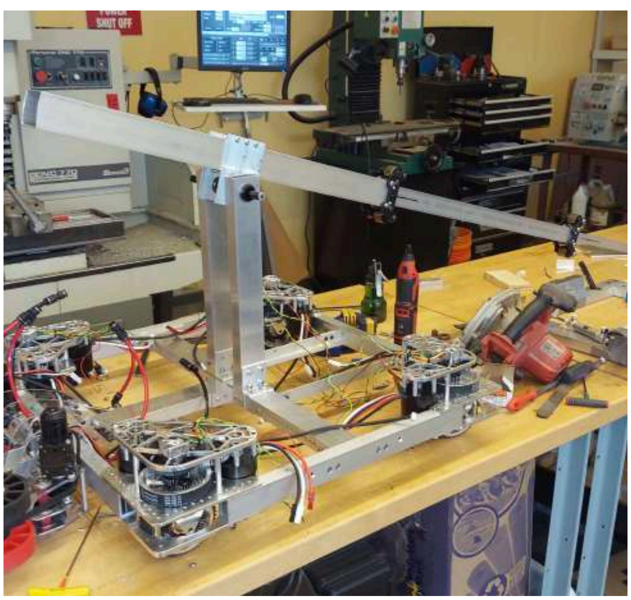
- 2 Rotation Point Arm
- Stinger arm

Stinger Arm Prototype:



Initially, we had to see if using a stinger for the arm would work and how many sections the stinger would require. We did some math and tried to account for certain factors like the grabber mechanism and found the ideal angle of the stinger and how many sections it needed.

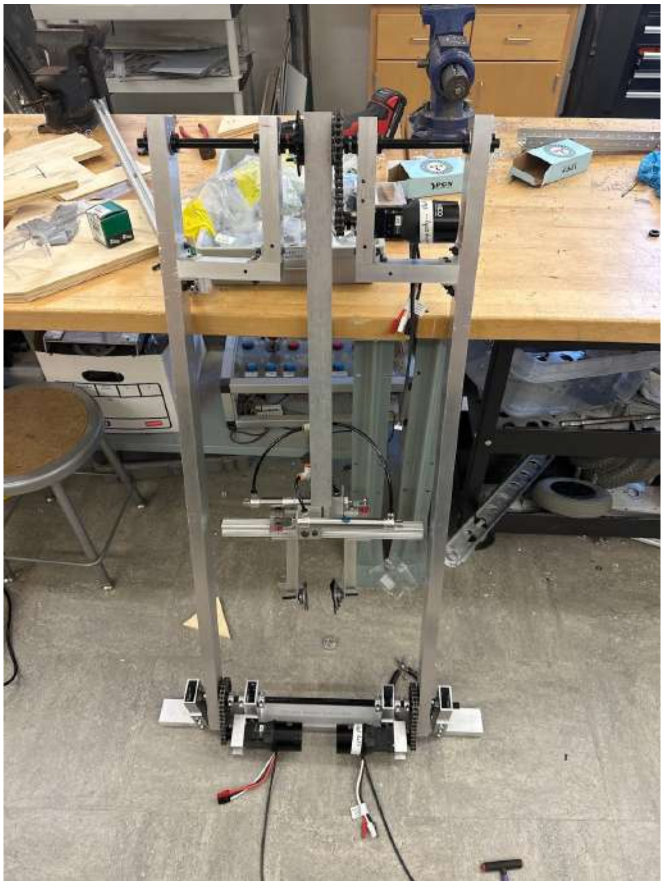
The stinger was built and attached with a clamp to our test bot. The idea worked, but was flimsy, and the braking power was unacceptable on the motor/gearbox that pulled the stinger in and out. This gearbox was subsequently switched to a ratcheting Gearbox ,which put less torque on the motor when not being run.



Now that we had seen the idea was there, the stinger was redesigned to rotate over 180 degrees, allowing for ground pickup and human station pickup. This design also was able to place game pieces (of course.) This new stinger was placed on the competition drive base to try it out.

during testing, we saw it worked, but that using the stinger would require a redesign of the grabber mechanism and potentially the intake. due to these considerations the team opted for a rotation point arm.

2 Rotation Point Arm:

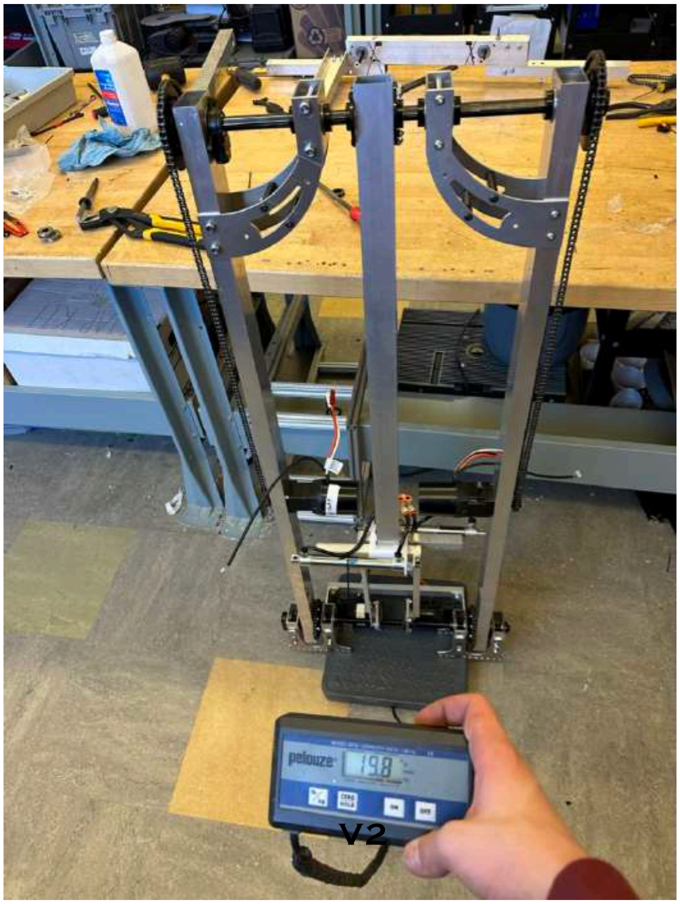


V1

The initial design of the 2 rotation point arm was built as a prototype. The design was made entirely out of 1x2 box, and proved quite effective in testing. With this 2 rotation point arm, we could place game pieces anywhere to score and pick them up anywhere on the field.

When we tried driving with the arm clamped to the drive base the robot fell over. The arm was too heavy, resulting in instability. We saw that the arm worked, but if we could not lower our center of gravity, we wouldn't be able to use it. we then went to the whiteboard and did extensive COG calculations to determine limiting constraints.

V2



We found the ideal weight of everything of the robot and came up with a elegant design for the arm that reduced its weight, especially up top. Additionally, this new design kept its original functionality and strength. On this new design, thinner pieces of box were used, some box was replaced by routed plate with standoffs and the shoulder motors were moved down, and the shoulder mechanism then driven using chain.

We mounted this new version of the 2 rotation point arm on the robot and tried it out. To use with game pieces we attached the current version of the grabber we were working with and were handing it pieces since we still didn't know how we would transfer them from the intake to the grabber. But, when testing, the arm worked great and, apart from chain tensioning issues, it seemed like the solution and no longer tipped over.

Stinger arm

PROS	CONS
- Light weight. - Easier to code.	- Cannot work without intake. - Would need a new grabber and/or intake.

2 Rotation Point Arm

PROS	CONS
- Works well + tested. - Can work independently of intake. - More robust. - Less prone to breaking.	- Harder to code. - More motors / sensors needed.

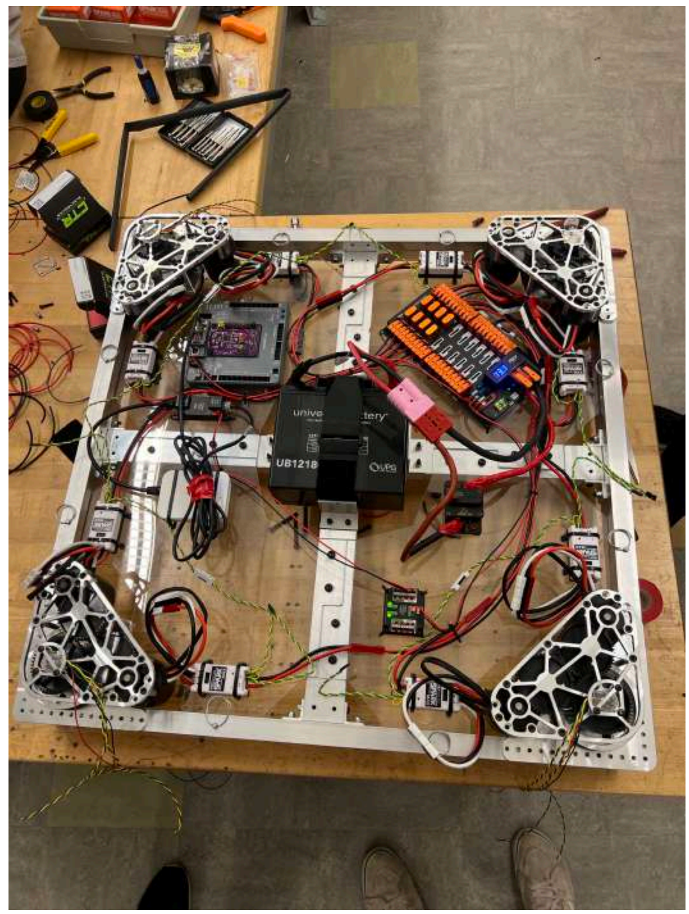
Arm Conclusion:

We decided to use the 2 rotation point arm. It worked effectively, repeatability, was robust and the robot no longer tipped over. The arm created an integrated system with both the intake and collector, which could also work independently if a component failed in competition.

Drive Base

The test bot drive base was developed in advance of kickoff to give us a general idea for how we wanted to use swerve and how we would go about building the competition drive base. The test bot was quite useful for trying mechanical components and code out when we still did not have a robot built. An example of this is when trying the stinger arm we clamped it to the test bot.

Test Bot

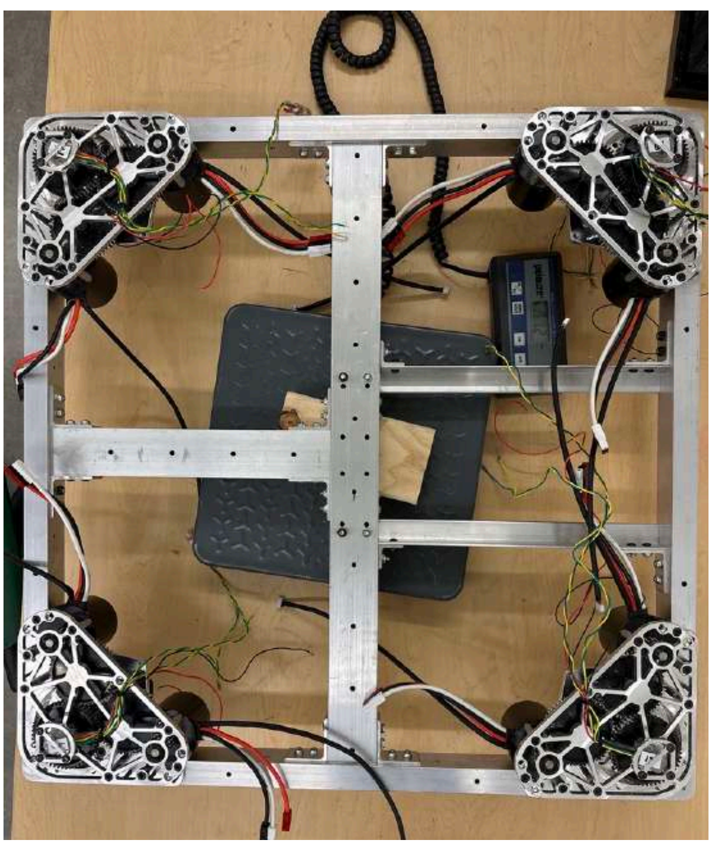


The test bot was quite useful for programming. Once we mounted a limelight to it the robot was used to teach coding to the new programming students and to calibrate and code the new April Tag features. Lastly, since we knew we wanted to try swerve this year, the test bot was a confirmational tool. It made building the competition season drive base much easier, worked effectively, and went much faster than the traditional tank drive our team was accustomed to.

24” x 27”

We originally decided on a 24”x 27” drive base. This decision took too long and we really needed to start building the robot since we were already on week 4. As a result we chose a size that would fit all our components, while maintaining a 24” side, allowing us to take up as little space as possible when on the charge station.

27” x 27”

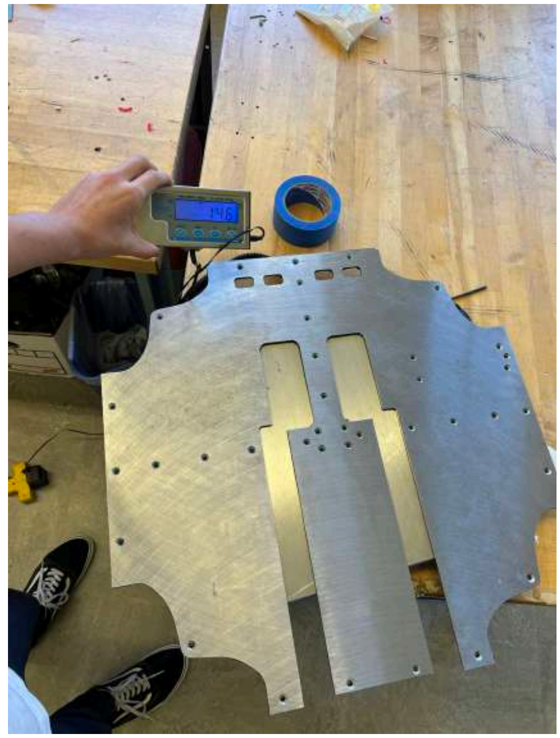


After trying to put all components on the robot we discovered the intake needed to be cutout within the bumpers. This, as a result, meant the intake wouldn’t fit unless we were 3” wider on the 24” side because of the 6” bumper rule. So, 4 days after building the first “final drive base” we built another one and started mounting the components back on it.

Steel Electronics Board:

Since the robot was not very heavy and we were worried about the robot tipping over in competition, we decided to remake the electronics board entirely out of steel and then insulated it with tape to prevent any electrical shorts.

Lastly, the electronics board was cut out of a single sheet of steel for ease of manufacturing and assembly. The design also included cutouts to have the ability to fix any component without needing to remove it.



This electronics board took up to 7 hours of machining to make but the end result was perfect. The steel electronics board allows us to tip up to 80 degrees on a side without falling over. The steel electronics board also added structural integrity to the drive base. The holes on this electronics board were also all countersunk to use flat head screws which meant when climbing the charging station we would not get stuck on anything.

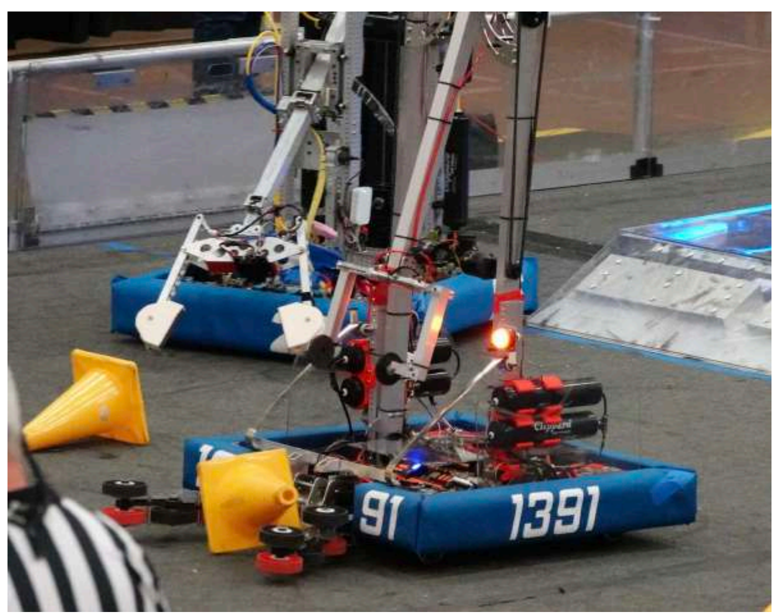
Team 1391's robot:

BAMI

Robot Final Build

Once the robot components had been put together all that was left were finishing touches. We 3D printed mounts for the air tanks, radio & camera. Additionally, we made swerve covers to protect the swerve and to mount the swerve's motor controllers because of a lack of space on the electronics board. Also, a hard stop was made for the arm to prevent over-extending.

Constant force springs were added to the intake because the pistons did not have enough force to bring them up alone. Additionally, two pieces of box were used on the intake for the cone to rest on when it was pulled in, which gave it the perfect placement for the arm to grab it. The intake motors were rotated because they came close to the bumpers, and we were worried they might get their wires pulled out.



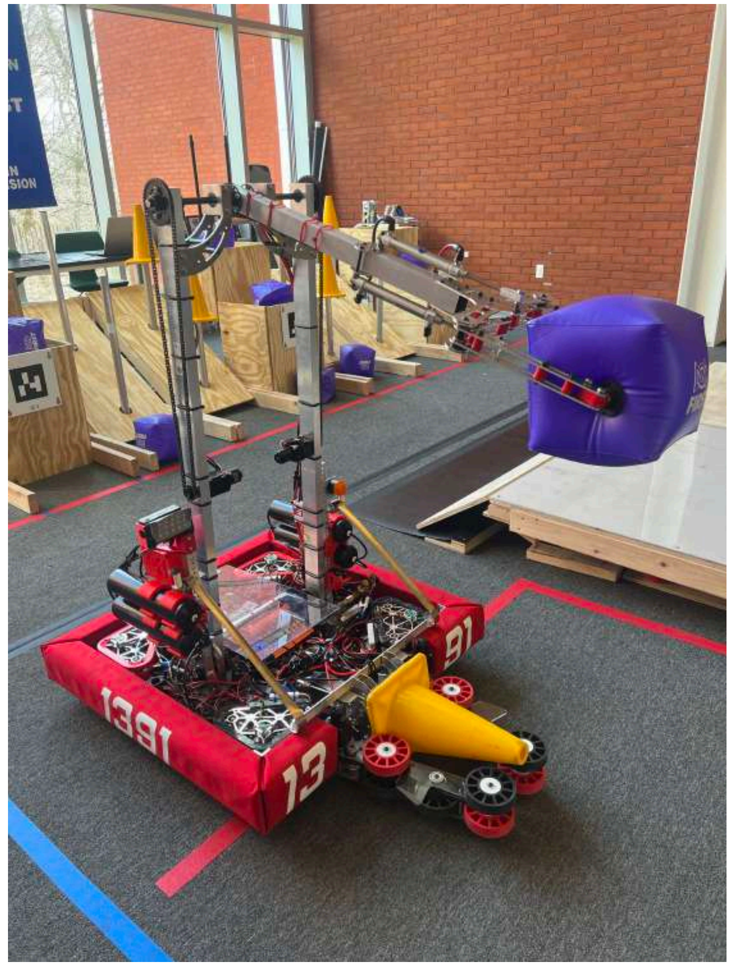
Build Evaluation

What went right:

- The arm and Intake can both work Independently. This means if either one of these breaks in game, we can still score game pieces.

- The intake has a wide game piece capturing range which means we can take game pieces from anywhere and don't have to line up perfectly with it when picking up from the ground.

- The decision to use swerve was definitely the right one. Not only was swerve less of a hassle in both drive base construction and repair, but also it allowed moving around the field to be way easier. The advantage of swerve was especially seen when balancing on the charge station, sometimes having to move to the sides making space for our alliance partners.

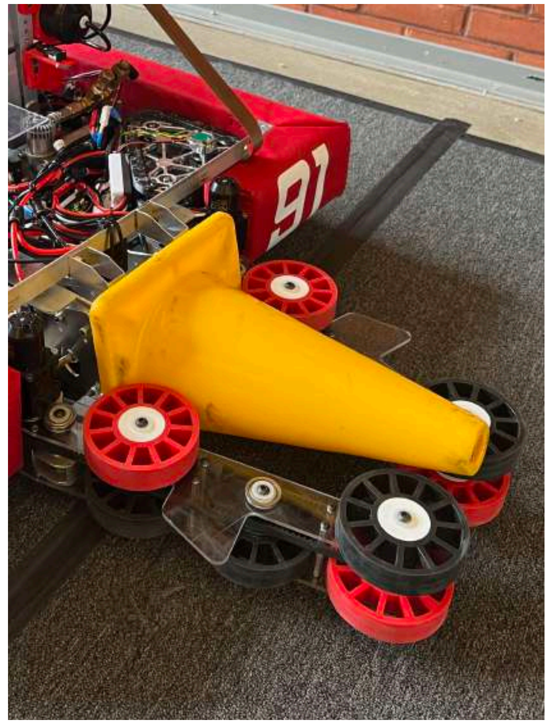


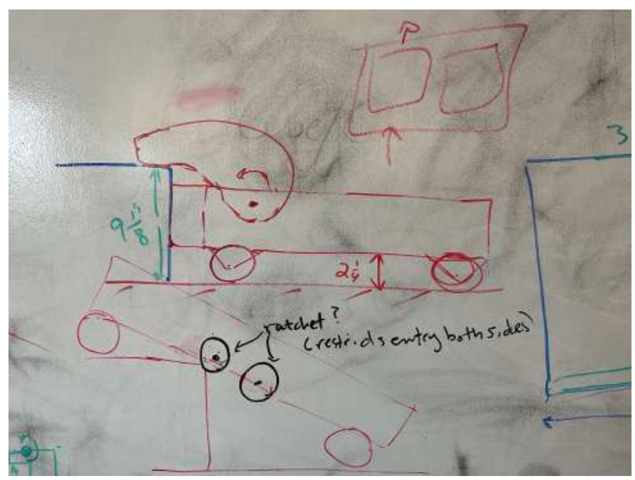
What went wrong:

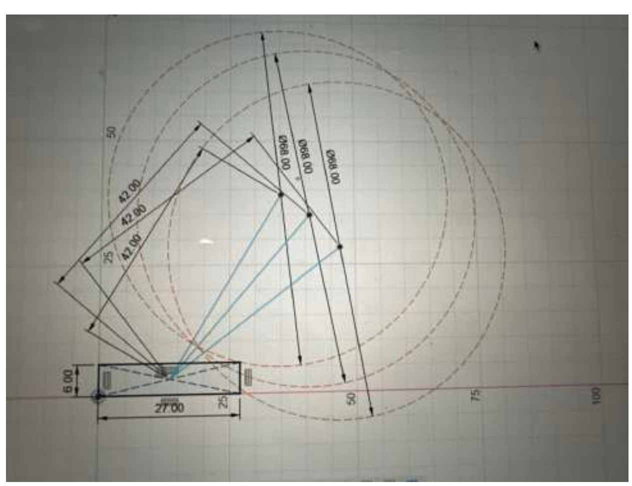
- We wasted a lot of time this season trying to solve a problem we later discovered didn't exist (climbing up the side) at the expense of developing other components more quickly.

- We should have communicated more clearly within the team about game piece transfer between the intake and grabber / arm. Fortunately, with a concerted and focused effort, we were able to solve the transfer challenges,

- We spent a lot of time talking about ideas and CADing them, rather than developing prototypes in parallel. We should have spent more time testing ideas and designs iteratively.







Week 1 Event Evaluation

What went right:

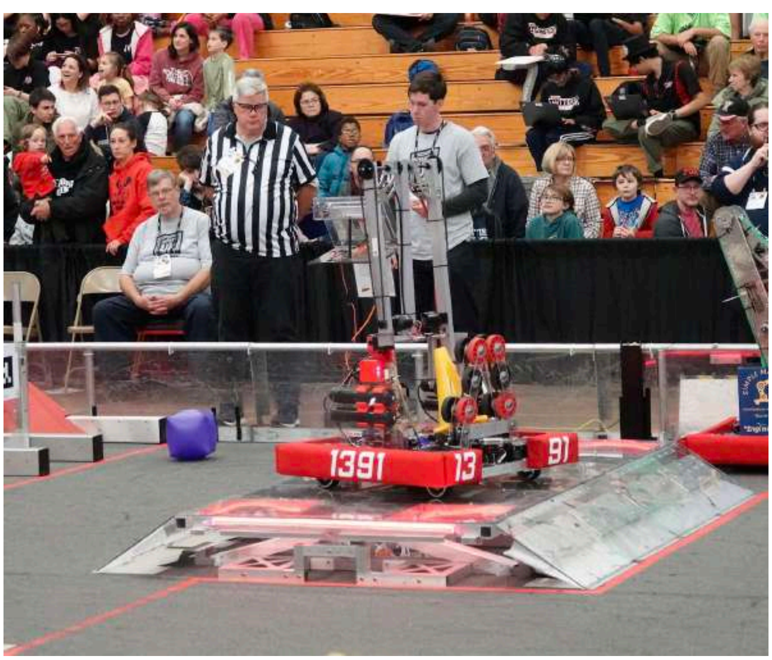
- The robot had very few minor mechanical issues, which meant we weren't fixing between matches often.
- Both the intake, transfer and scoring arm worked as designed throughout the competition with few issues.
- Autonomous worked almost perfectly at all times, and the Metal Moose won the autonomous award.
- Balancing on the charge station went really well and rarely posed any issues.

What went wrong:

- Cones would occasionally get stuck in the robot due to a incorrect grab of the cone out of the intake.

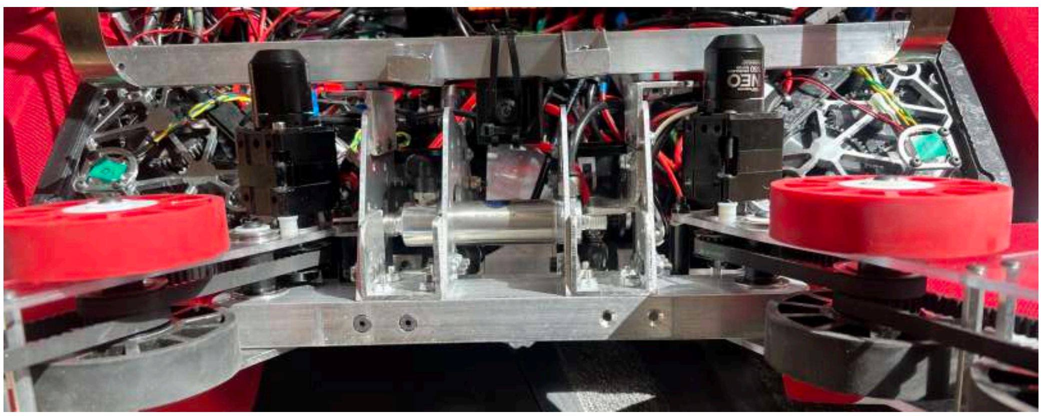
- The cube would sometimes get stuck to the grabber due to its sticky pads.

- The intake was less robust than expected, and drive chain on the arm failed twice.

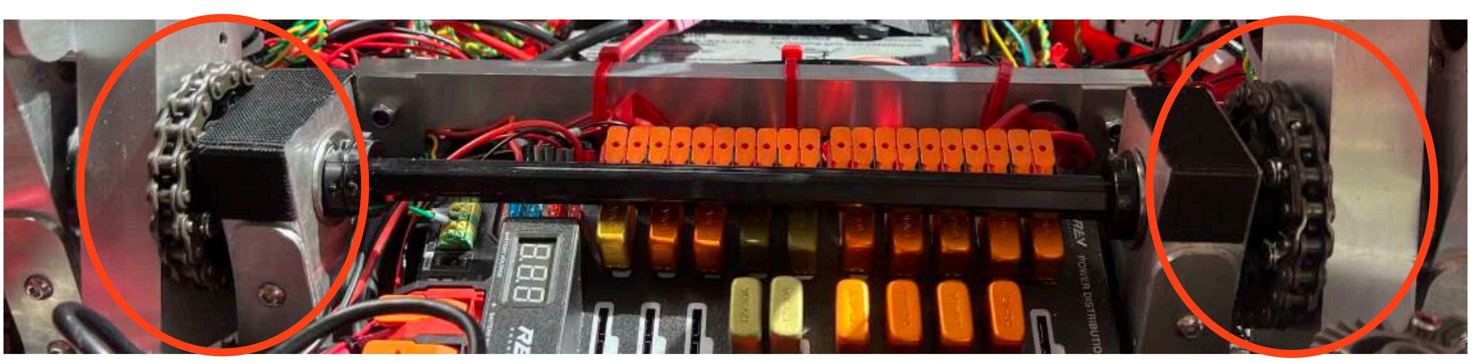


Next steps for second event:

Intake: The intake worked well but needed to be fixed and screws had to be tightened throughout the competition. To fix this, the central 2x1 box of the intake was changed from 1/16 inch to 1/8 inch to prevent this from breaking again. Also, the pistons were pulling on standoffs which broke two times in the competition so these standoffs were replaced with 1/4-20 screws which would bear the piston load much better.

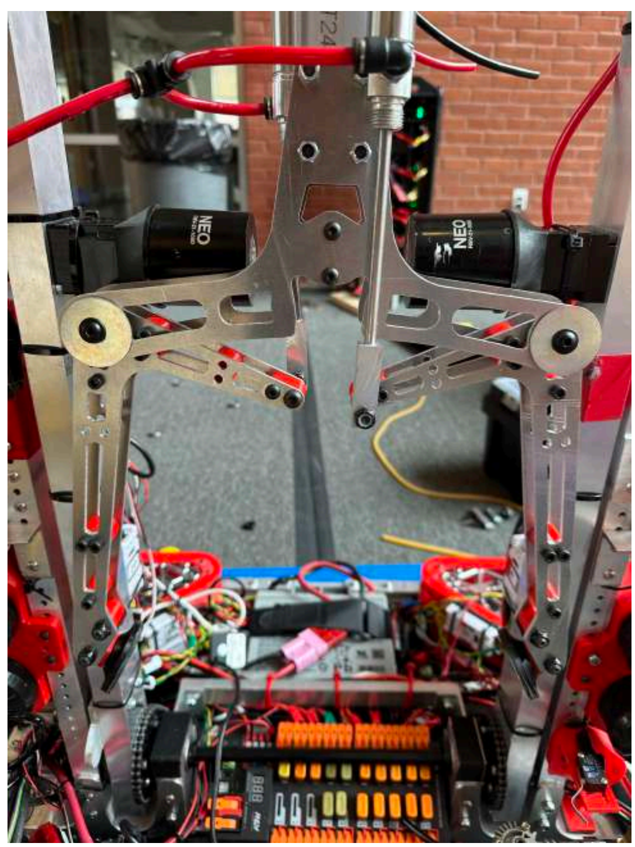


Arm: The main issue with the arm was the chain links kept snapping due to the constant force being put on them. To solve this, the #25 chain on the arms shoulder was replaced with #35 chain to prevent it from snapping again. Also, the chain was being tensioned by a hex shaft (which seemed to be another cause of chain failure) so this hex shaft was removed.



Grabber: The grabber had some issues in the first competition. Cubes would get stuck in it occasionally because it would not open wide enough. Also, the grip was not always right on cones so sometimes they fell into the robot. To solve this, we created a new grabber V4. This version is machined out of sheet metal and doesn’t work in a linear motion anymore. Once we tried it out we saw it worked much better. The new Grabber also features smaller paddles to reduce grip on cubes which would prevent the issue of cubes getting stuck on it. The standoffs between both plates were 3D printed.

Lastly, the new gripper did not fit between the NEO motors so these motors were replaced with NEO 550s with a 100:1 gearbox, nd a total gearing of 250:1 on the arm's elbow joint.



Programming

This year, our programming team started our pre-competition season off by learning the basics of robot programming in Java from returning team members, using last year's robot and an offseason swerve chassis to try out their code and get real feedback. Once the team members were feeling comfortable doing some programming for the robot, we started looking into AprilTag tracking to get a jump-start for the upcoming season. In the past, we have used Limelights to track retroreflective tape on the field, but with this being the first year for AprilTags to be introduced into FRC, there weren't any easy solutions yet. We spent most of this preseason time using PhotonVision, an AprilTag tracking software to identify and track the AprilTags, and as this only gave fairly raw data and was a little slow, it was a challenge to figure out how to use it as best we could, and we used trigonometry to begin implementing vision-based odometry updates using the AprilTags. Early in the season, Limelight released their own AprilTag tracking software, which worked out a lot better for us than PhotonVision, as it is a much faster alternative to the PhotonVision software we were using. After kickoff we began working on programs to balance automatically and to line up with apriltags using odometry and the vision updates we had figured out before the season started. The autobalance program was a great project for newer team members to work on, as the code was pretty straightforward. When the code is run, it gets the pitch and roll of the robot using the NavX gyroscope then compares that to a constant to determine if the robot should drive in a direction or not. Using both the pitch and roll allows the robot to compensate for any rotation it might have without doing any complicated math.

```
  public void execute() {
    var strength = adjustmentStrength.getAsDouble() *
kTriggerMultiplier;
    var pitch = -m_swerveDrive.getNavxPitch();
    var roll = m_swerveDrive.getNavxRoll();

    pitch = -kAngleTolerance > pitch || pitch > kAngleTolerance ?
pitch * strength: 0;
    roll = -kAngleTolerance > roll || roll > kAngleTolerance ?
roll * strength: 0;
    pitch = MathUtil.clamp(pitch, -kSpeedLimit, kSpeedLimit);
    roll = MathUtil.clamp(roll, -kSpeedLimit, kSpeedLimit);
    m_swerveDrive.drive(
      roll,
      pitch,
      rot: 0,
      Field relative: false
      );
  }
```

The AprilTag auto alignment code was a little more complicated, using the different inputs from the attached Limelight and multiple PID controllers to align with the AprilTags. We took multiple approaches to achieving this, starting with an easier method where we would use the Limelight to estimate the robot's position on the field and then get the position of the AprilTag on the field and drive the robot to that position using the wheel odometry. This worked really well, except the field estimates of the robot's position had a frustrating margin of error. Because of that we tried another approach, this time using the raw data from the camera to get the size of the tag and the position of the tag relative to the camera.

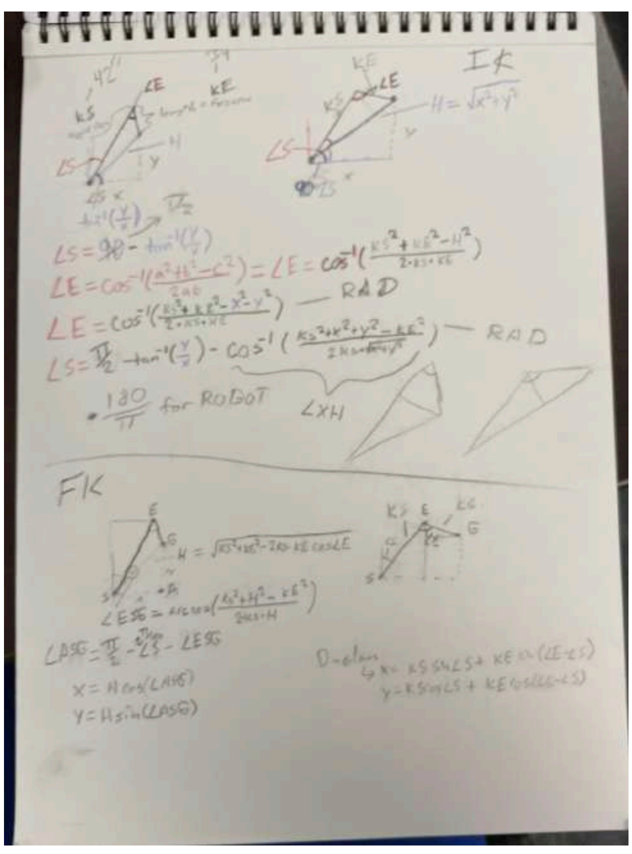
```
public void execute() {

    var xDrive = -xController.calculate(m_drivetrain.getTA());
    var yDrive = -yController.calculate(m_drivetrain.getTX());
    var rot =
turnController.calculate(m_drivetrain.getFieldPosition().getRotati
on().getRadians());
    rot = MathUtil.clamp(rot, -1, 1);
    xDrive = MathUtil.clamp(xDrive, -3.0, 3.0);
    yDrive = MathUtil.clamp(yDrive, -3.0, 3.0);
    if (m_drivetrain.getTA() < 0.5) xDrive = -2.5;
    if (m_drivetrain.getTA() == 0) xDrive = 0.0;
    m_drivetrain.drive(xDrive, yDrive, rot, true);
}
```

The function getTA() returns the percentage of the area of the camera feed the AprilTag is taking up(we found at 3.4% the robot would be the perfect distance away).

getTX()returns the offset from center(0) the tag is on the x-axis within the camera feed.

Once these programs were functional, we decided to hold off on the more specific programs and wait until we had a real competition robot to test the code on. While the build team was working on getting that robot ready, we started experimenting with PathPlanner, a program FRC teams can use to create Autonomous routine paths easily using curves. This was something we haven't done in the past, especially with swerve drive, something brand new for our team this year.

As we received the robot, the biggest challenge to work with was the arm. To monitor the angles of the arm, we have through bore encoders reading the angle of the different joints, we will refer to these joints as the shoulder and the elbow. The arm required very precise PID tuning to keep it from breaking itself or other things around it. Once we had basic control over it, we used the angles to gather a series of angle combinations that would allow us to score at each level. Once this was working, we decided we wanted to solve the inverse kinematics, which would allow for very easy control over game piece positioning. The math itself looks like this:



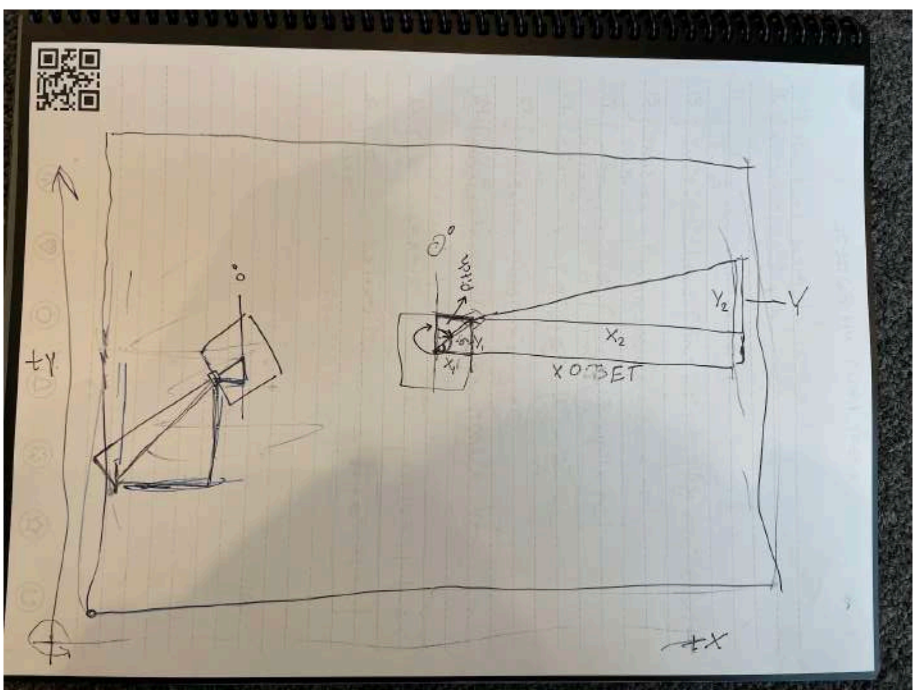
And the code looks like this:

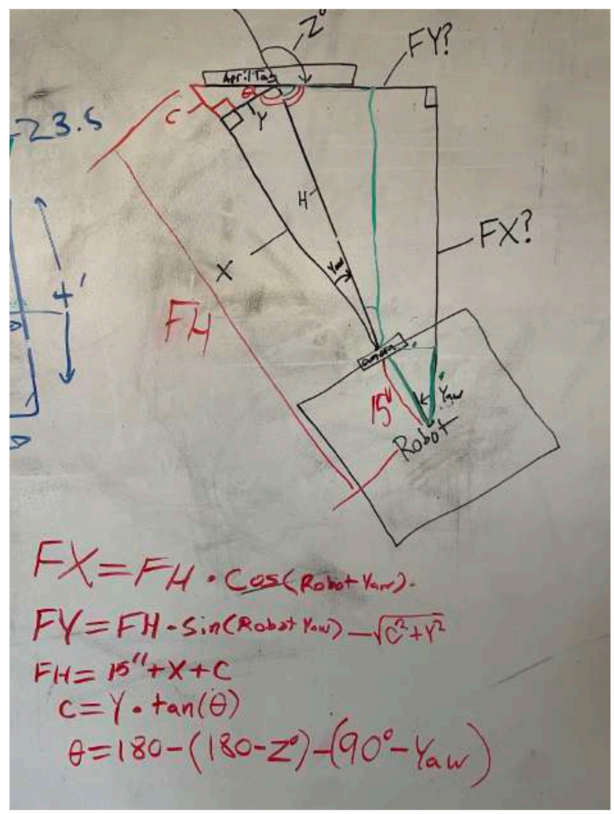
```
public double getIKShoulder(Double x, Double y) {
    if (y < 0) return getShoulderAngle();
    var x2 = Math.pow(x, 2);
    var y2 = Math.pow(y, 2);
    var hypotenuse = Math.sqrt(x2 + y2);
    Double angle = 90 - (180/Math.PI) * (
      Math.atan(y/Math.abs(x))  + Math.acos(
        (Math.pow(kShoulderLength, 2) + x2 + y2 -
Math.pow(kElbowLength, 2))
          / (2 * kShoulderLength * hypotenuse)
          )
        );
    if (angle.isNaN() || angle.isInfinite()) return
getShoulderAngle();
    else if (Math.abs(angle) < 35.6) return angle;
    else return getShoulderAngle();
  }

  public double getIKElbow(Double x, Double y) {
    if (y < 0) return getElbowAngle();
    var x2 = Math.pow(x, 2);
    var y2 = Math.pow(y, 2);
    Double angle = Math.copySign((180 / Math.PI) * Math.acos(
      (Math.pow(kShoulderLength, 2) + Math.pow(kElbowLength, 2) -
x2 - y2) / (2 * kShoulderLength * kElbowLength)
    ), Math.signum(x));

    if (angle.isNaN() || angle.isInfinite()) return
getElbowAngle();
    else if (Math.abs(angle) < 155.6) return angle;
    else return getElbowAngle();
  }
```

What these inverse kinematics equations do is allow us to provide the x and y coordinates of a point and the robot will automatically know what angles to move the arm to. The initial part are the equations and the if statements at the end are safeguards in case the code is given a bad set of coordinates, like a point the arm physically can't reach.

Once we had good control over the robot, we started creating autonomous routines, which were achieved using sequenced arm positioning commands, path planner routes, auto balancing, and apriltag alignment all to make our autonomous routines as consistent as possible.

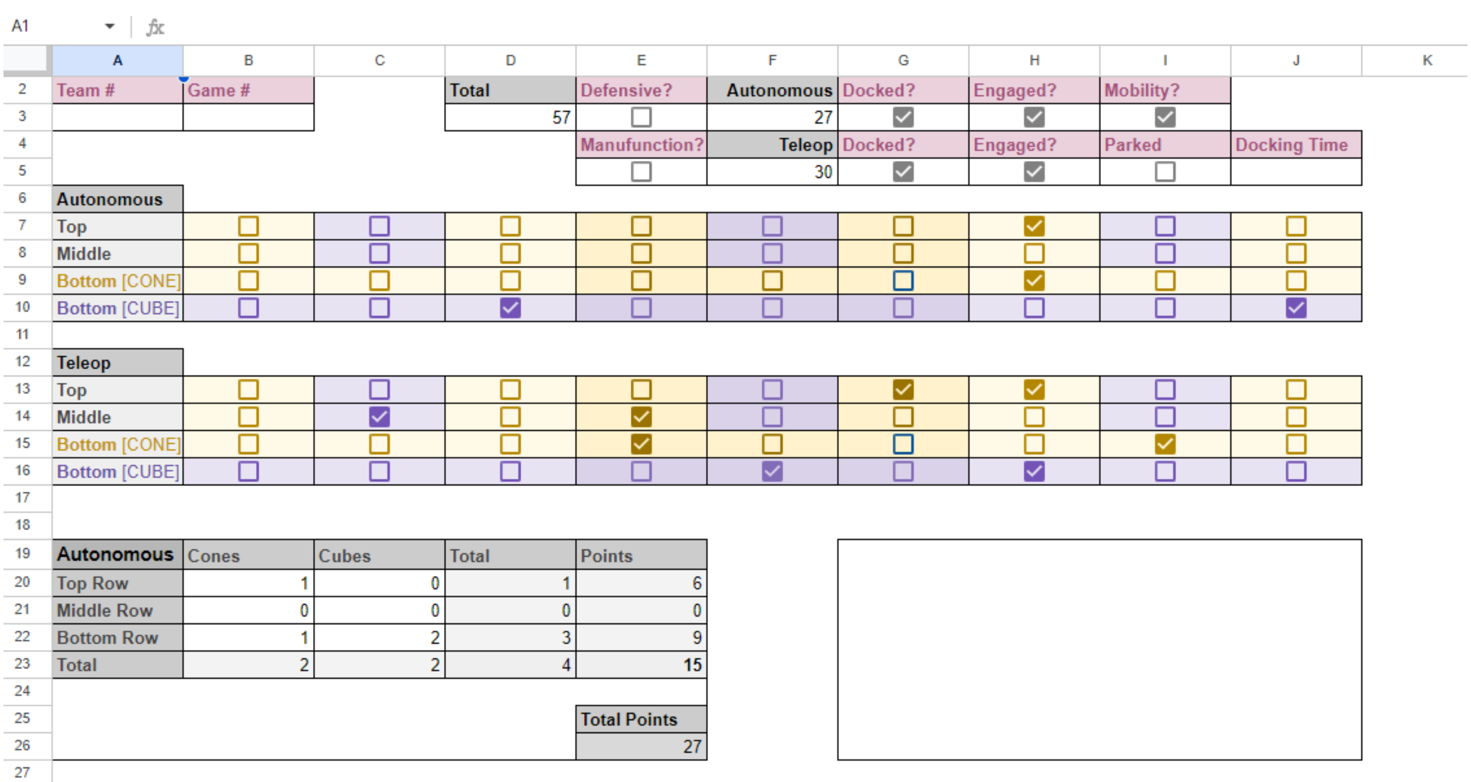
Here is some of the math we used initially to determine the robot's position relative to





Scouting

The scouting team was responsible for creating a new scouting model that would prove effective for data analysis. We brainstormed effective ways for scouters to take data based on this year's game mechanics. We had a few different ideas on how to collect and input the scouters' data, originally attempting to use another team's scouting program. We realized using the other team's program would not fit best with our plan to use a grid to take data, so we tried creating a model in Google Sheets with QR codes to collect the data.

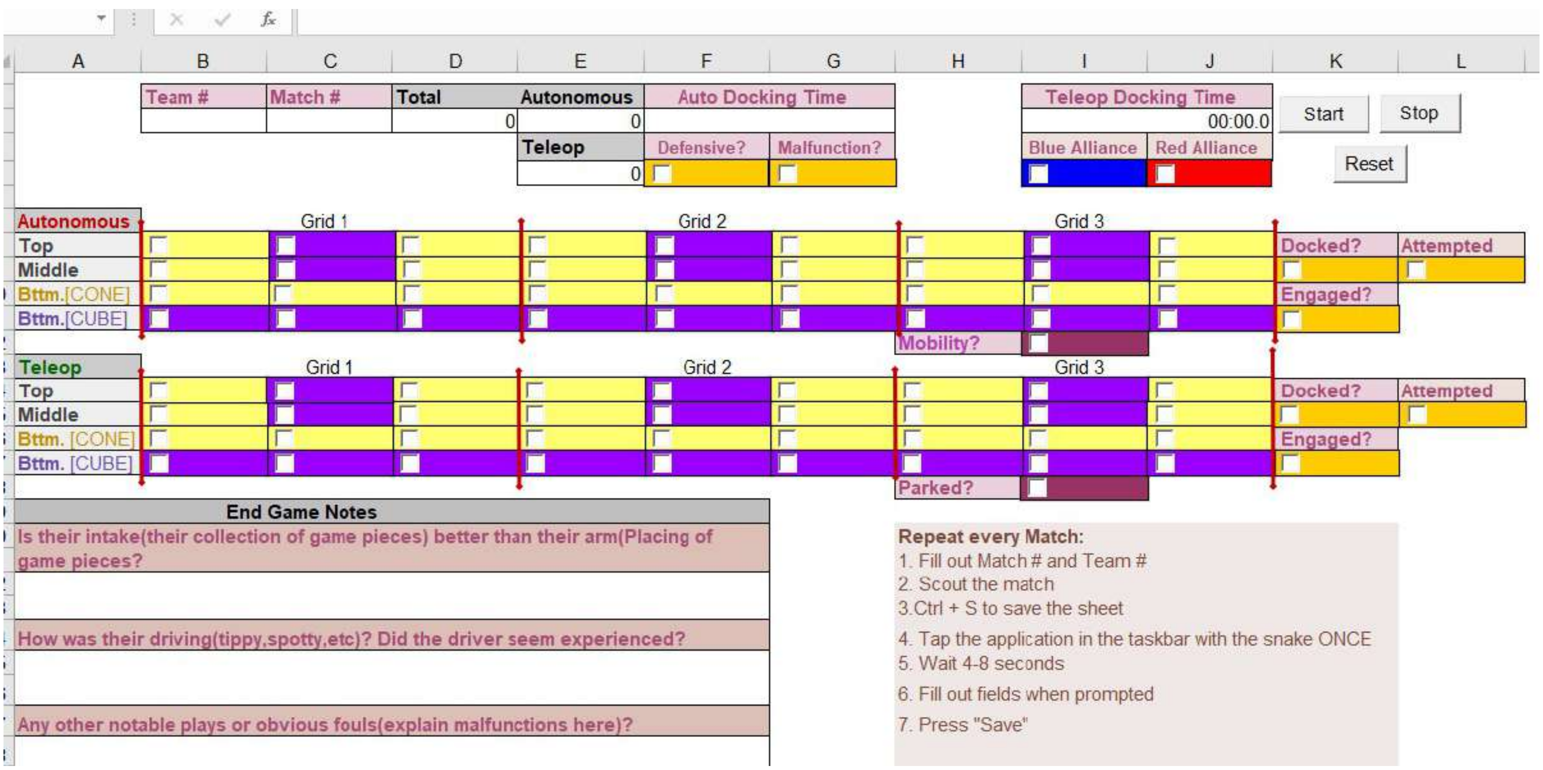


	A	B	C	D	E	F	G	H	I	J
2	Team #	Game #		Total	Defensive?	Autonomous	Docked?	Engaged?	Mobility?	
3				57	☐	27	☑	☑	☑	
4					Manufunction?	Teleop	Docked?	Engaged?	Parked	Docking Time
5					☐	30	☑	☑	☐	
6	Autonomous									
7	Top	☐	☐	☐	☐	☐	☐	☑	☐	☐
8	Middle	☐	☐	☐	☐	☐	☐	☐	☐	☐
9	Bottom [CONE]	☐	☐	☐	☐	☐	☐	☑	☐	☐
10	Bottom [CUBE]	☐	☐	☑	☐	☐	☐	☐	☐	☑
12	Teleop									
13	Top	☐	☐	☐	☐	☐	☑	☑	☐	☐
14	Middle	☐	☑	☐	☑	☐	☐	☐	☐	☐
15	Bottom [CONE]	☐	☐	☐	☑	☐	☐	☐	☑	☐
16	Bottom [CUBE]	☐	☐	☐	☐	☑	☐	☑	☐	☐

Autonomous	Cones	Cubes	Total	Points
Top Row	1	0	1	6
Middle Row	0	0	0	0
Bottom Row	1	2	3	9
Total	2	2	4	15

Total Points
27

After consulting with members of the team, we decided to use Excel sheets with flash drives instead of our original plan. It took us less time to understand the process, so we focused on perfecting the sheet, training the scouters, and creating a method to collect data.



Lastly, we created a system to transfer data from the stands to the pits during a competition efficiently, and organized a team for analyzing the data collected.

Business / Impact

Business:

Starting in preseason, we brainstormed, drafted, and finalized a brand new 5-year plan, worked on a sponsorship packet, and worked on creating outreach activities. After many meetings and a group review with the entire team, we finalized our plan. Following this, we worked together to finish our sponsorship packet and which was later sent out to a list of prospective sponsors. our individual donor fund was well received this year and lead to important funding for our program, particularly our impact and equity work. in conjunction with our sponsorship packet, we put together a letter that was sent to alumni and parents with information about our upcoming season and our needs.

Halfway through the season, we decided to split the business team to work on separate tasks. Some members worked on the essay submission for the Woodie Flower's Award, while the other members worked hands-on to create a functional and elegant pit design. Within our pit design efforts, we worked with our pit crew and our drivers to make sure all of their requirements were met.

Impact:

While the business team was hard at work, the outreach strand of business worked hard on kickstarting FLL in Bermuda. Their school system contains 18 primary schools, and we plan on providing each with a LEGO SPIKE Prime robotics kit. Although the promise is grand, the impact team has started small by keeping ourselves connected with both Bermuda's Board of Education and Government. Taking out time to teach team members how to build with the kits allowed us to prepare to send 15 minute tutorial videos. Our outreach has progressed so much so that an advertisement for our ambitions appeared on the radio. We hold ourselves accountable for the commitment of teaching fundamental programming skills through the use of LEGO's proprietary SPIKE Prime app. Continuing on this path will allow us to formulate an accessible community for STEM opportunities in FIRST Bermuda.

please see our team overview document that more fully presents our impact and equity work, past, present and future.

Thank You