210Z ECLIPSE

Engineering Notebook

Calgary, Alberta, Canada

Start Date: May 5th, 2024 **End Date:** TBD **Book:** 1 of 1

Date 1-10-2025 Event Name Rumble in the Rockies VEX V5 Robotics Competition

Innovate Award Submission Information Form

Instructions for team: Please fill out all information, printing clearly. This form should be included immediately after the Engineering Notebook's cover page. In the case of physical notebooks, this form can be printed out and placed in the notebook. For digital notebooks, this form can be scanned in and included. Teams may only submit **one** aspect of their design to be considered for this award at each event. Submission of multiple aspects will nullify the team's consideration for this award.

Full Team Number: 210Z

Brief description of the novel aspect of the team's design:

8-wheel 450 rpm drivetrain powered by 6 11-watt motors, featuring all omni-wheels that are drilled out then fitting with ball bearings. This allows us to have a nimble yet robust chassis whilst preserving durability, stability, and overall performance under varying conditions.

Identify the page numbers and/or the section(s) where documentation of the development of this aspect can be found:

Page 111 comprehensively outlines the process regarding the design and synthesis of the ball-bearing wheels.

Explain why your submission is unique from other approaches to the problem it solves or task it performs:

This intricate design is unique and innovative because of the increased power and streamlining it provides our drivetrain. Because these ball bearings spin much smoother compared to the standard wheel inserts, it allows for our drive to be more efficient and reach maximum speed rapidly. Furthermore, the ball bearings also ensure the stability of the joint supporting the wheels because of their low tolerance of oscillation, minimizing wear over time.

Our Notebook is divided up into two sections: Engineering & Programming

The Programming Volume is After the Engineering Volume

Table of Contents

2024-05-07 to 2024-09-29

Drivetrain designs/modification entries are highlighted in **red.**
Intake designs/modification entries are highlighted in **orange**.
Clamp designs/modification entries are highlighted in **green**.
Wall Stake designs/modification entries are highlighted in **blue**.
Elevation designs/modification entries are highlighted in **purple.**
Arm designs/modification entries are highlighted in **pink.**

Miscellaneous Entries are in highlighted in **black**.
Tournaments/Scrimmages are underlined.

//////////////////// ***2024-04-27: VEX High Stakes announced to the public.*** //////////////////

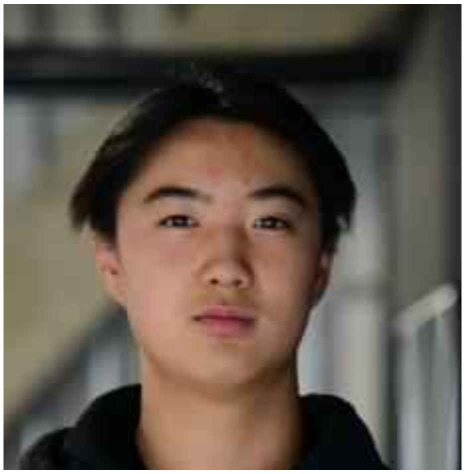
Table of Contents

2024-10-27 to 2024-19-12

Table of Contents

2024-12-16 to 2025-02-12

Meet the team



Adam Xu

- 16 years old | Grade 11
- Sir Winston Churchill HS
- Programmer/Notebook

Richard Feng

- 14 years old | Grade 10
- Henry Wise Wood HS
- Builder/Designer

Alex Su

- 15 years old | Grade 11
- Western Canada HS
- Designer/Notebook

Meet the team

Kevin Zhao

- 17 years old | Grade 12
- Dr. E.P Scarlett HS
- Programmer/Skills-Driver

Maxwell Li

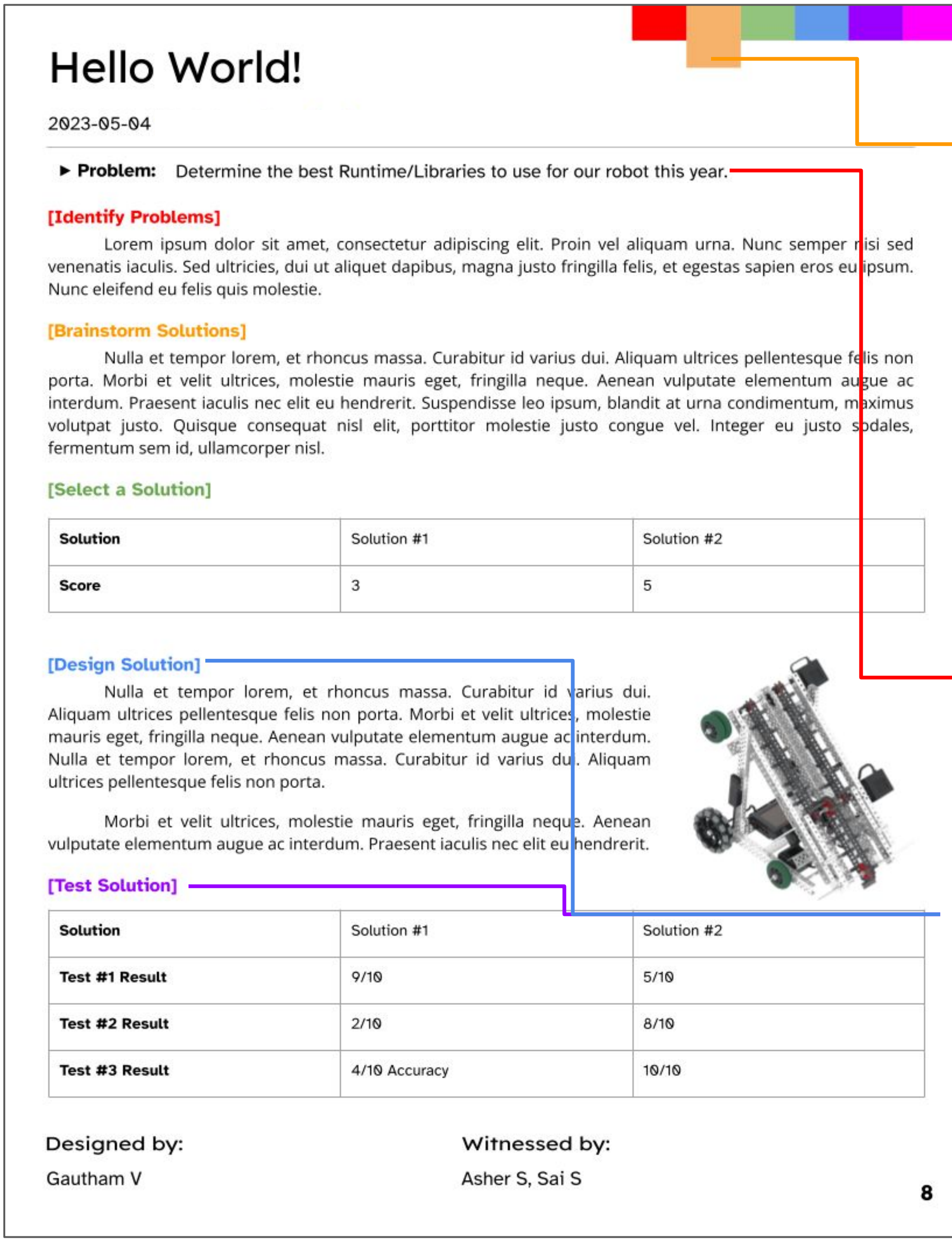
- 16 years old | Grade 11
- Henry Wise Wood HS
- Builder/Driver

We are team 210Z. A five-man team formed in Calgary, Alberta, competing in the VR5C High School competition. Each member of our team has a specific skill set that they can use to benefit our team in unique ways this season. In our spare time, team 210Z helps run our organization Western Mechatronics. A non-profit with the goal of filling the hole of missing robotics opportunities for students across Calgary. This is the team's fifth year competing, and we're looking forward to seeing what this season has to offer.

Sample Page Overview

2024-05-05

Here is a sample page detailing how this notebook will be formatted, including the robot subsystem tags from the previous page.



Hello World!

2023-05-04

► **Problem:** Determine the best Runtime/Libraries to use for our robot this year.

[Identify Problems]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel aliquam urna. Nunc semper nisi sed venenatis iaculis. Sed ultricies, dui ut aliquet dapibus, magna justo fringilla felis, et egestas sapien eros eu ipsum. Nunc eleifend eu felis quis molestie.

[Brainstorm Solutions]

Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta. Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Praesent iaculis nec elit eu hendrerit. Suspendisse leo ipsum, blandit at urna condimentum, maximus volutpat justo. Quisque consequat nisl elit, porttitor molestie justo congue vel. Integer eu justo sodales, fermentum sem id, ullamcorper nisl.

[Select a Solution]

Solution	Solution #1	Solution #2
Score	3	5

[Design Solution]

Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta. Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta.

Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Praesent iaculis nec elit eu hendrerit.

[Test Solution]

Solution	Solution #1	Solution #2
Test #1 Result	9/10	5/10
Test #2 Result	2/10	8/10
Test #3 Result	4/10 Accuracy	10/10

Designed by: Gautham V

Witnessed by: Asher S, Sai S

8

The flag at the top shows which subsystem this page is a part of. See the table of contents for a list of categories.

The **problem** we're currently facing is identified at the top to remind readers and ourselves exactly what we are currently doing.

Tags are on each page of the notebook, so the design process is evident throughout the flow of the notebook.

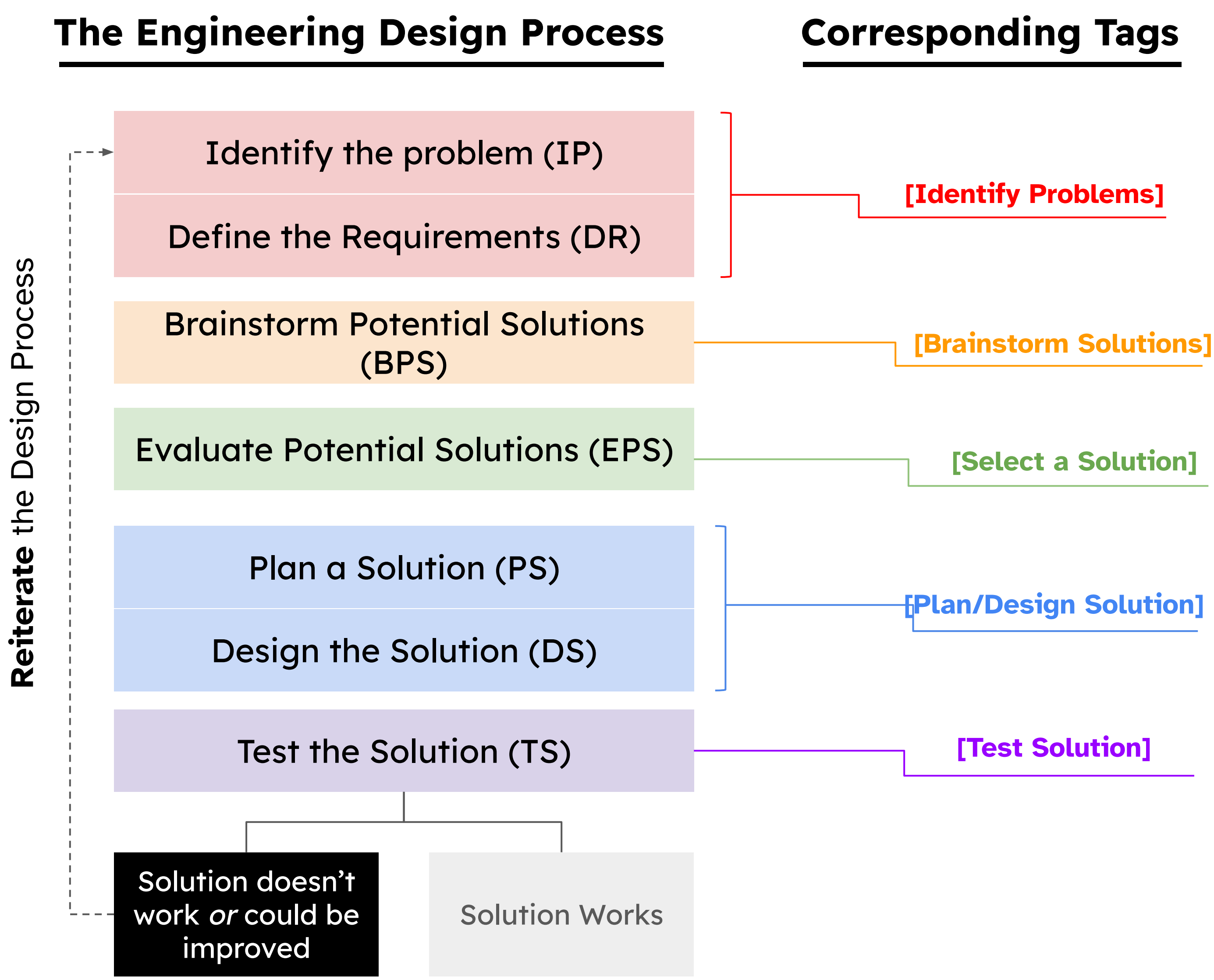
Designed by:
Adam X

Witnessed by:
Kevin Z

The Engineering Design Process

2024-05-07

The Engineering Design Process is a key component to the success of any robot. Using this method, our team is able to efficiently cover all of our solutions and decide on the design best fit for our robot. The **process** is shown on the *left*, and the corresponding **tags** used on each page of notebook are shown on the *right*.



Designed by:

Adam X

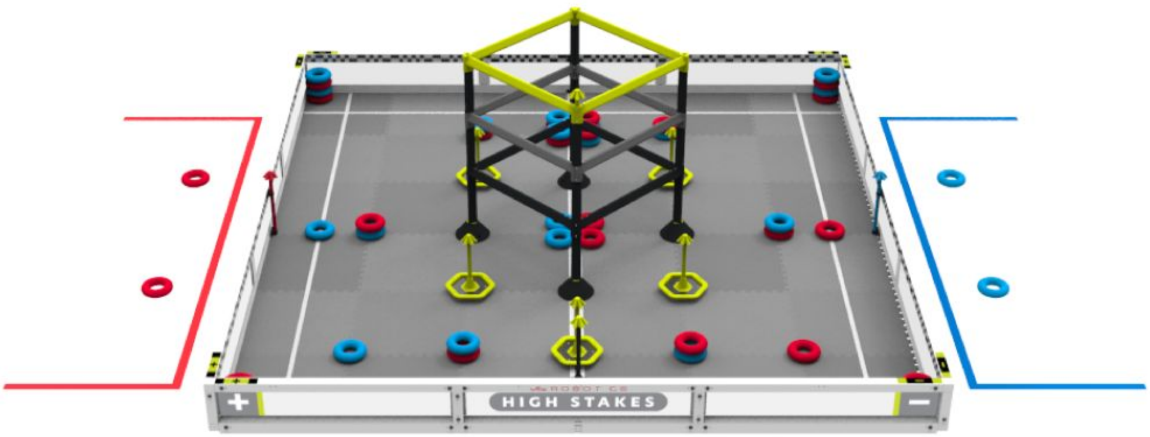
Witnessed by:

Alex S

Early Game Analysis

2024-05-10

▶ **Problem:** Field & Game Elements



[Identify Problems]

Rings (Total: 48)

- Either red or blue (Quantity: 24 Red, 24 Blue)
- 1 preload ring for each team
- Worth 1 point on any stake
- Worth 3 points for being the top ring on any stake

Endgames

- Hang (3 Tiers)
  - Tier 1– 3 points
  - Tier 2– 6 points
  - Tier 3– 12 points
- Corners (Positive & Negative)
  - Diagonal to respective corners
  - Negative - cancels out other scored rings
  - Positive - doubles points of rings in the corner

Stakes (Total: 10)

- Wall Stakes (Quantity: 2)
  - 25” (including height of field perimeter)
  - Holds 6 rings
- Alliance Stakes (Quantity: 2)
  - Holds 2 rings
  - Only scores points for corresponding alliance colour
- Ladder Stakes (Quantity: 1)
  - Holds 1 ring
- Mobile Stakes (Quantity: 5)
  - Holds 6 rings
  - Requires force to score
  - Can be scored in positive/negative corners

Designed by:

Alex S

Witnessed by:

Adam X

Early Game Analysis

2024-05-10

▸ **Problem:** Analyze Reveal Video and Make Observations

[Identify Problems]

Like Grant Cox said after the game reveal, this game is an upgraded version of Round Up. However, since they were using older 393 motors instead of V5 and they could hold 5 rings, we feel that a lot of the strategies they used are not very applicable to High Stakes. Since the game manual has been released, we will be going through both the reveal video and the game manual, making observations and brainstorming potential solutions to them. In high stakes, there are three main ways you can score:

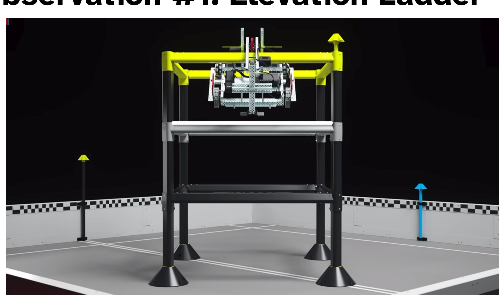
1. **Mobile Goals**
   - Scoring rings on mobile goals gives 1 point, and the top ring gives 3. A mobile goal hold a maximum of 6 rings. They are descorable, however we predict that in the early to mid season, not many teams will be able to.
   - Mobile Goals can be brought into the corners to be either doubled or negated.

2. **Wall Stakes**
   - There are two types of wall stakes: alliance, and neutral. Points are awarded the same as mobile goals. Alliance stakes hold 2 rings and neutral stakes hold 6
   - There is also the high stake, on the top of the elevation structure, which is worth 3 points since it can only hold 1.

3. **Elevation**
   - Teams can elevate themselves on the high stake during the last 10 seconds. Points 1st, 2nd, and 3rd stage elevations are 3, 6, and 12 respectively

In addition to scoring, there are also other things we would like to make note of.

Observation #1: Elevation Ladder



The first rung on the ladder is much higher than last year, 18", which is the size limit. This means that there has to be some kind of vertical expansion during the match on the robot to be able to elevate. The second rung is 32" of the ground, and the top is 46"

Designed by: Adam X

Witnessed by: Kevin Z

Early Game Analysis

2024-05-10

▶ **Problem:** Analyze Reveal Video and Make Observations

Observation #2: Positive and Negative Corners

There are two types of corners located on the field: positive and negative. Positive corners double all of the rings scored on a mobile goal, and negative corners make all of the rings on a mobile goal negative. This is the first every time VEX has had a negative element to a game, and a possible strategy could be to fill a mobile goal up with an opponent's rings and put it in the negative zone. However, logically, positive corners are much more worth than negative corners, since if two goals of the same colored rings are in a positive and negative corner, then the negative corner only negates the double effect of the positive corner, and the rings on it are still scored normally. Since scoring does not award that many points as other games, doubling your mobile goals will most likely be crucial to winning games.

Observation #3: Stake Cap

Unlike in Tipping Point, the mobile goals as well as wall stakes all have caps, making it more difficult to score and descore. To score, you will probably have to apply some kind of pressure to make sure the ring fully passes through the cap.

Observation #4: Wall Stakes

The two heights of the alliance and neutral wall stakes will make it so that wall stake mechs will need to be adjustable and versatile. Wall stakes are at a slight disadvantage to mobile goals since they can't be doubled, so we anticipate that they will be somewhat ignored in the early season.

Observation #5: Ring Size

The High Stakes rings are 7 inches wide, and 2 inches thick, meaning the inside of the ring is 3 inches. These rings are quite wide compared to other game elements from past games, so robots will need to be quite wide to be able to store rings inside.

Designed by:

Adam X

Witnessed by:

Alex S

Early Game Analysis

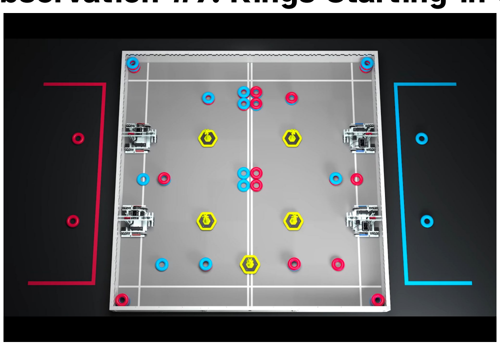
2024-05-10

▶ **Problem:** Analyze Reveal Video and Make Observations

Observation #6: 5 Mobile Goals

Mobile goals are oriented at the start of the game so that there is one in was quarter of the field, one for each robot, and the fifth to the side on the middle line. Since each robot can only possess one mobile goal at a time, the obtaining and controlling the fifth goal will be crucial to winning the game. During the autonomous period, teams will likely rush for the fifth goal on the middle line, similar to Tipping Point. To maximize the chances of reach the goal before the other team does, you could build a claw or hook mech specifically for the goal rush, something like 210Y did in tipping point.

Observation #7: Rings Starting in the Corner



Having rings start in the positive and negative corners means that you need to clear them first before you put a goal inside the corner. This task can be time consuming in a match, and the opposite alliance could take advantage of it and steal the fifth goal or score the wall stakes. Additionally, it might not be worth the resources (motors, pistons etc.) to build a separate mechanism for this task.

Observation #8: Descoring

Like mentioned earlier, we think that descoring is a difficult task to do due to the stake caps, and most likely will be ignored until the mid to late season. A way to descore rings from stakes could be to go from the bottom under the rings and to force them all up and out using some sort of arm.

Designed by:

Adam X

Witnessed by:

Maxwell L

Early Game Analysis

2024-05-10

▸ **Problem:** Analyze Reveal Video and Make Observations

Observation #9: Elevation

Actually elevating the robot up past the first stage will be extremely difficult, due to the design of the ladder. We predict that this objective will also be ignored in the early season, and most teams will probably have a simple passive stage 1 hang. Since the ladder goes straight up vertically, when the robot is pulling itself up, it needs to somehow avoid running into the rungs/stages above it so it can elevate itself higher. I think that this is an extremely difficult problem to solve.

Observation #10: Autonomous Win Point (AWP)

In order to earn an additional winpoint during the autonomous period, the following criteria must be met:

<SC8> An **Autonomous Win Point** is awarded to any alliance that ends the autonomous period with the following tasks completed, and has not broken any rules during the autonomous during the autonomous period.

1. At least 2 scored rings of the alliance's color
2. A minimum of 2 stakes on the alliances ide of the autonomous line with at least 1 ring of the alliance's color scored
3. Neither robot contacting / breaking the plane of the starting line
4. At least 1 robot contacting the ladder

In rule 2, a stake is defined as one of the mobile goals, or wall stakes, so this year, it should be relatively easy for one robot to complete all of the criteria, and that it should be relatively easy to achieve it. That being said, the game manual does state that the criteria will be modified for signature and regional events, most likely to increase the difficulty.

Designed by: Adam X

Witnessed by: Richard S

Brainstorming Ideas

2024-05-23

▶ **Problem:** Which past robots could we modify to score effectively this year?

[Brainstorm Solutions]

We found two previous seasons/games that were quite similar to High Stakes: Tipping Point (2021-2022) and In The Zone (2017-2018)

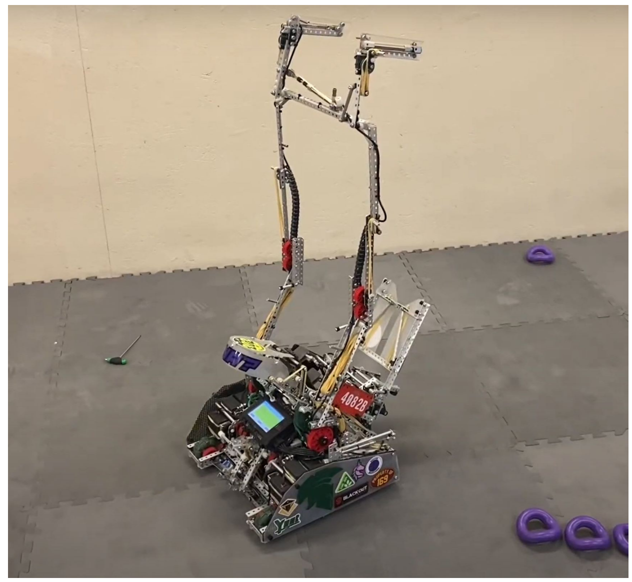
Tipping Point Bot:

Positives:

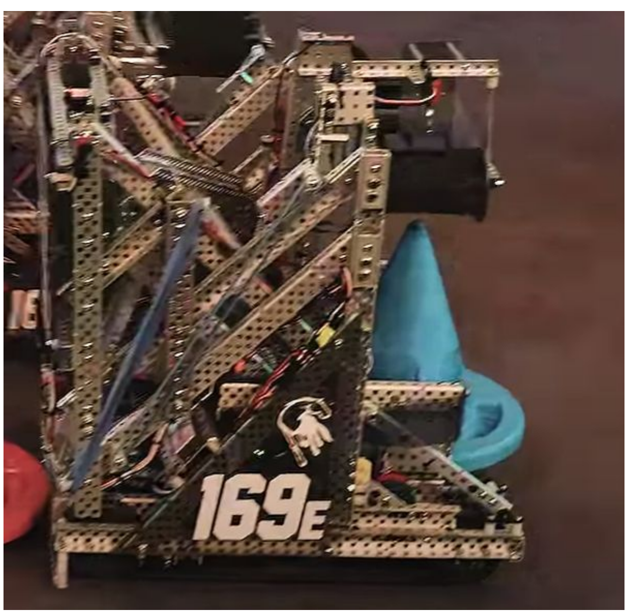
- Fast scoring
- Strong grip of mogos
- Low central Cog
- Hang Potential
- Light

Negatives:

- Cannot score on wall stakes more than once
- No use for a four bar lift on the front
- Can't directly score rings without adding pressure on the hood



Credit to Team 4082B



Credit to Team 169E

In The Zone Bot:

Positives:

- Scores on all stakes (Except Ladder Stake)
- Strong grip on mogos
- Hang Potential

Negatives:

- Unstable Cog while scoring
- Slower scoring
- Heavy

Designed by:

Alex S

Witnessed by:

Adam X

Brainstorming Ideas

2024-05-23

▸ **Problem:** How can Tipping Point Bots in score effectively?

[Brainstorm Solutions]

The "Tipping Point" meta bot has the potential to be a strong bot. By using a conveyor to bring rings to the top of the mogo it is able to score quickly and efficiently. However, in its current state it lacks any real method to score rings as it relies on gravity to score rings, which won't work due to the unique shape at the top of the mogo. Furthermore, the robot has no use for the four bar on the front and can only score on the mogos. If we could generate some pressure to press the ring on the goal and find another use for the motor on the fourbar, this robot could be a strong contender for the early season.

Credit to Team 210Y

Four Bar - A set of parallel bars that move together in unison.

Credit to Team 169E

Designed by:

Alex S

Witnessed by:

Adam X

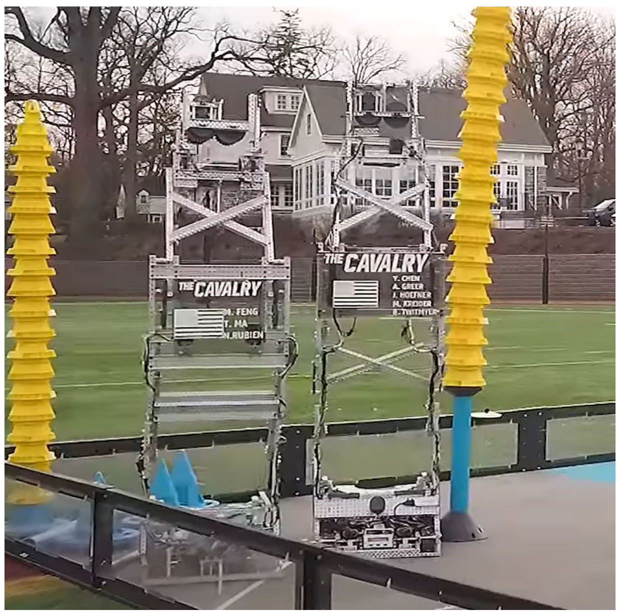
Brainstorming Ideas

2024-05-23

▶ **Problem:** How can In the Zone Bots in score effectively?

[Brainstorm Solutions]

The “In the Zone” meta bot seems like an excellent choice for this year's game. It is able to score on most stakes and has a good grip on the mogos. However, when the scissor lift fully extends, the robot's cog is very high making it super easy to tip over, putting the robot at high risk. A solution for this may be to shorten the length of each four bar; however, even with this solution, building a scissor lift seems highly unnecessary as it uses an excess amount of metal increasing its weight which could impact the friction between the robot and the foam mats, decreasing the robots speed.



Credit to 169: “The Cavalry”

Scissor Lift - A four bar facing the centre of a robot mounted on another four bar; extends the distance covered by the robot.

Credit to Team 169E

Designed by:

Alex S

Witnessed by:

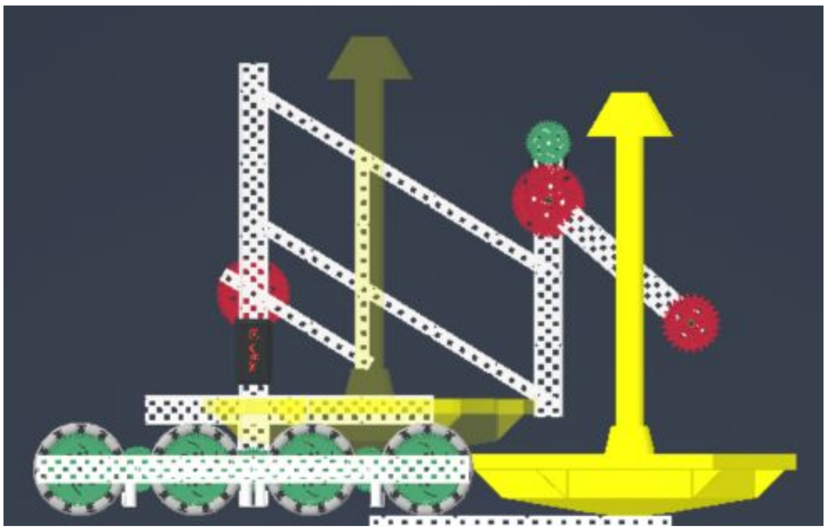
Adam X

Brainstorming Ideas

2024-05-23

▶ **Problem:** How can a new robot be created from the previous designs?

[Brainstorm Solutions]



This robot takes heavy inspiration from the “In the Zone” robot. This robot uses a four bar lift with a rotational arm at the end of it. This design has a unique pickup method not displayed in the photo above that will be discussed in the following slides. The robot also uses the same goal pickup method that the “In the Zone” meta bots had; a tray that pushes out forward and downward to wrap around the mogo. It then picks it back up, pulling in towards the centre of the robot. This robot is also asymmetrical, only having the four bar and arm on one side of the robot; we found that this was an optimal way to reduce weight on the robot as linking the mechanism to the same thing on the other side seemed unnecessary as we weren’t dealing with anything heavy enough to affect the movement of the arm.

Designed by:

Alex S

Witnessed by:

Adam X

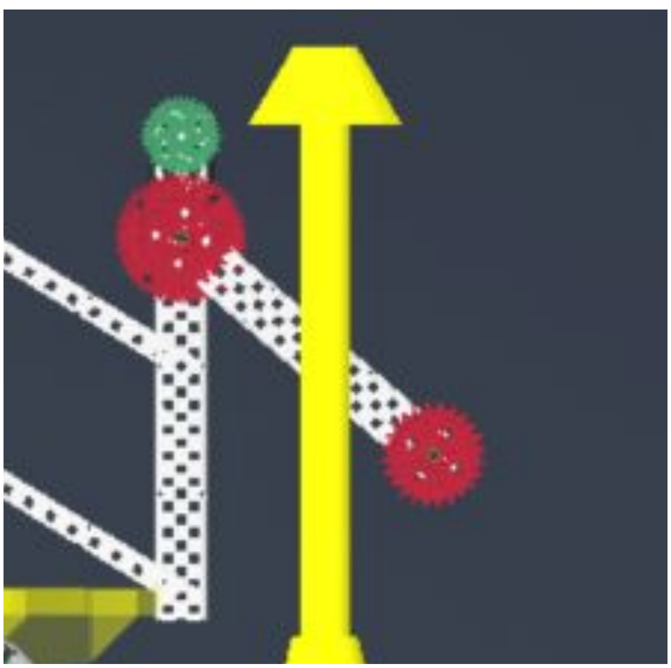
Brainstorming Ideas

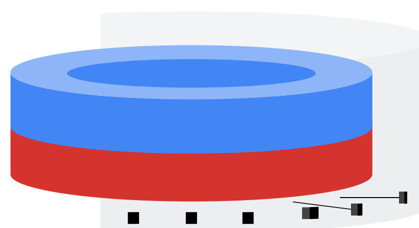
2024-05-23

▶ **Problem:** How can we pick up the rings without the standard intake?

[Brainstorm Solutions]

The “pick up” arm is on the end of the four-bar lift. It uses a 200 RPM cartridge and has a 3:5 gear ratio (120 RPM). Beside the gear is a sprocket also attached to the arm. This sprocket links to another one at the end of the arm, attached to the pickup mech. Doing this ensures the pickup mech is always in the same position at any given point. Therefore, the sprocket can be adjusted for the mech to parallel the ground. The mech itself is a polycarbonate semi-cylinder with open ends on the top and bottom. On the bottom opening is a circle of holes with Zip Ties facing inward, creating a layer that requires pressure to enter or exit the mechanism. By doing this, we can pick up rings without dropping them and without using a motor. Lastly, we chose a semi-cylinder over a cylinder as it allows us to rotate the arm to the right to pull off the goal instead of up, slightly increasing our scoring efficiency. Additionally, using this pull-out/drop-off method allows us to adjust the length/pressure of the Zip Ties without running the risk of not being able to score at all.





Designed by: Alex S

Witnessed by: Adam X

RI24H | Challenges

2024-08-03

▶ **Problem:** How can we gain experience for the season while avoiding early failures?

To avoid any major issues when building our first robot of the season, Alex, along with some members from our sister team 210K, decided to challenge themselves with the robot in 24 hours challenge. The goal was to make the most efficient robot possible with the given parameters to prepare them for anything they might face during the season. These parameters include:

- No more than two pneumatic pistons
- No more than one 150 PSI air tank
- Some method to score on wall stakes
- Some method to sort ring colour
- A 100% accuracy method to score on mobile goal
- Must have space for at least one odometery wheel

Designed by:

Alex Su

Witnessed by:

Richard Feng

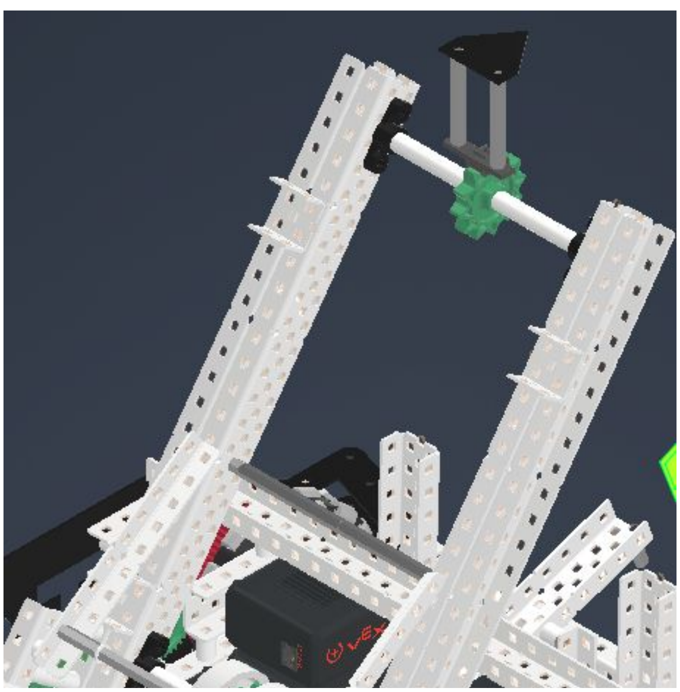
RI24H | Pre-planning

2024-08-04

▶ **Problem:** How can we earn points efficiently during the match?

Hooks Mechanism:

Since we did not have a lot of time to design and construct this robot. We settled on a sprocket conveyor that utilized standoffs and polycarbonate plastic to resemble the shape of a fishing hook. This way we could hook onto rings and pick them up for scoring.



Linear Slide Mechanisms:

Due to our shortage of pneumatic pistons, we opted to try and create a non-pneumatic system that we could use to score on the wall stakes. Our solution was to mount our entire sprocket conveyor on a set of linear slides, the slides would have an outward force provided by rubber bands and would be held down by a ratchet and slip gear. This way, when the gear rotated into the slipped portion, there would be no force to counteract the elastic force of the rubber bands and our conveyor would shoot out of the robot and high enough to hit the wall stakes.

Designed by:

Alex Su

Witnessed by:

RI24H | Pre-planning

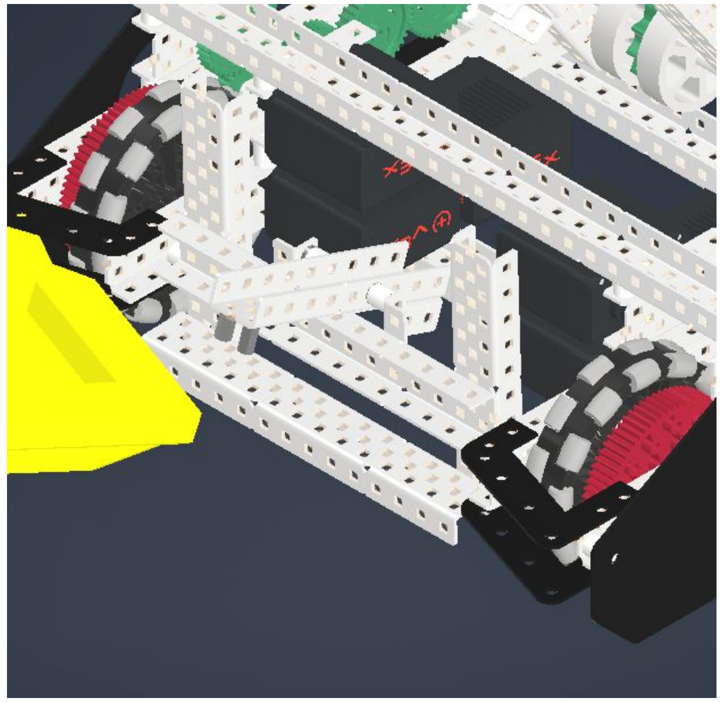
2024-08-04

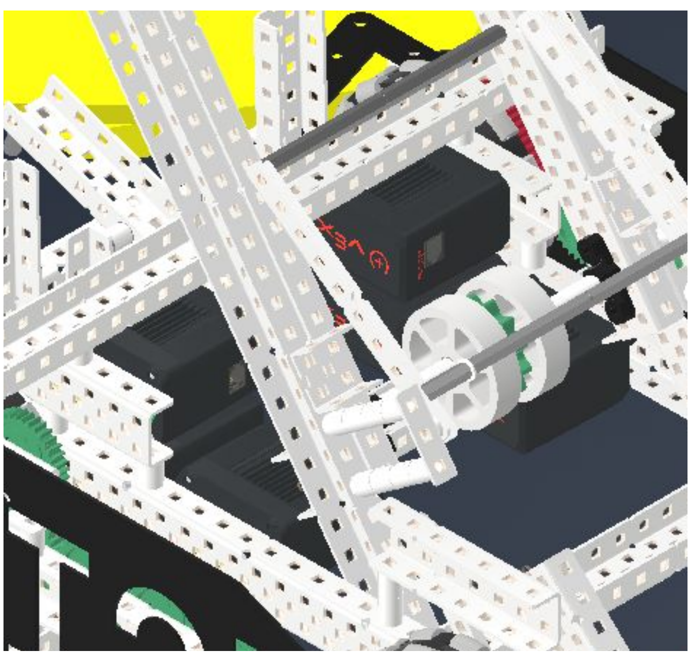
► **Problem:** How can we score efficiently throughout the match?

Omnidirectional Tilting Mechanism:

For this challenge, we decided to use our two pneumatic pistons to build a strong goal tilting mechanism. Our reasons for this decision were:

1. It gives us "possession" over it, preventing other teams from stealing the goal while we're scoring on it.
2. It allows us to find the optimal angle for the hooks to score from
3. It causes us to simultaneously pick up the goal, causing the robot to carry the goal and not drag it on the floor, lowering the overall friction force which would slow down our drive speed.





Manual Colour Sorting Mechanism:

Though we did not plan to compete with this robot, we wanted to prepare for all scenarios. In the case that we picked up the wrong coloured ring, we wanted a pneumatic free way to eject it from our system. To do this, we decided to create a thin plate that would bob up and down as a ring passed under it, but would stay in place and dismount the ring from the hook when the conveyor reversed. Allowing us to manually sort the colour of our rings during a match.

Designed by:

Alex Su

Witnessed by:

Drivetrain Brainstorming

2024-08-30

▶ **Problem:** What are some drivetrain designs that can succeed in High Stakes?

[Brainstorm Solutions]

We think that speed and maneuverability will be key in high stakes. From this, we narrowed it down to two options for our drivetrain, 450 rpm and 480 rpm, both on 2.75-inch wheels and with 600 rpm motors.

450 rpm:

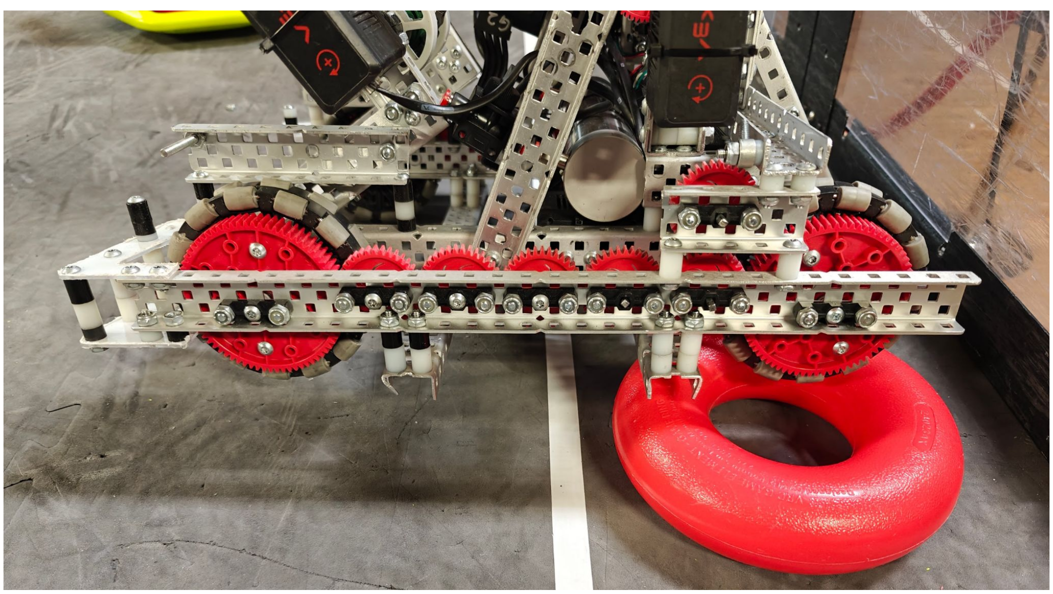
- 3:4 gear ratio (36-tooth to 48-tooth)
- More torque, slower
- 36-tooth to 48-tooth means that its a low strength, thinner gear driving a high-strength thick gear.

480 rpm:

- 4:5 gear ratio (48-tooth to 60-tooth)
- Faster, less torque

Wheels Sticking out or not

Another consideration of design is whether the wheels would stick out of the drivetrain or not. For this game, there isn't any barrier to cross or go over, so it makes no sense to have our wheels sticking out. This also takes away from our size limit and we might drive over a ring and get stuck, like in Tipping Point or Spin Up, where robots would constantly get immobilized on rings or discs.



To prevent this from happening, we won't have the wheels stick out of the drivetrain

Designed by:

Adam X

Witnessed by:

Alex S

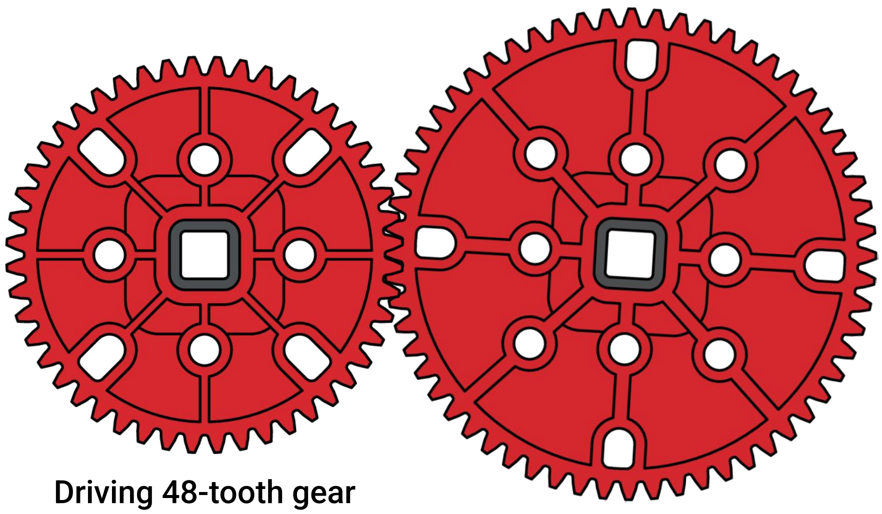
Drivetrain Planning

2024-08-30

▸ **Problem:** How can we design the an optimal and efficient drivetrain?

[Select a Solution]

As of right now, 450 rpm seems to be the most common, so we ended up going with the 480 rpm, since we wanted our robot to be slightly faster than other robots. Here is a rough demonstration of the gear ratio:



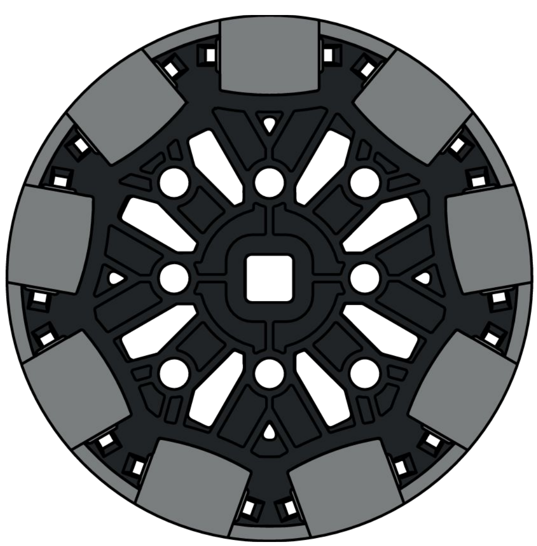
Driving 48-tooth gear (600rpm)

Driven 60-tooth gear (480rpm)

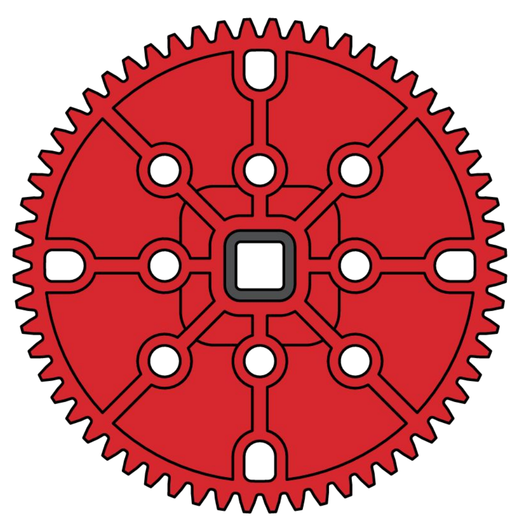
[Plan/Design Solution]

6-Wheel Drive

We decided on using 6 wheels in our drivetrain (3 on each side) instead of 4 because since we are attaching the 2.75-inch wheels with the 60-tooth gears, and they are almost the same size, we were afraid that the robot would sink into the mats too much and then the gear would rub against the mats while driving, which is not what we want.



2.75-inch wheel



60-tooth high strength gear

Designed by:

Adam X

Witnessed by:

Kevin Z

Drivetrain Planning and CAD

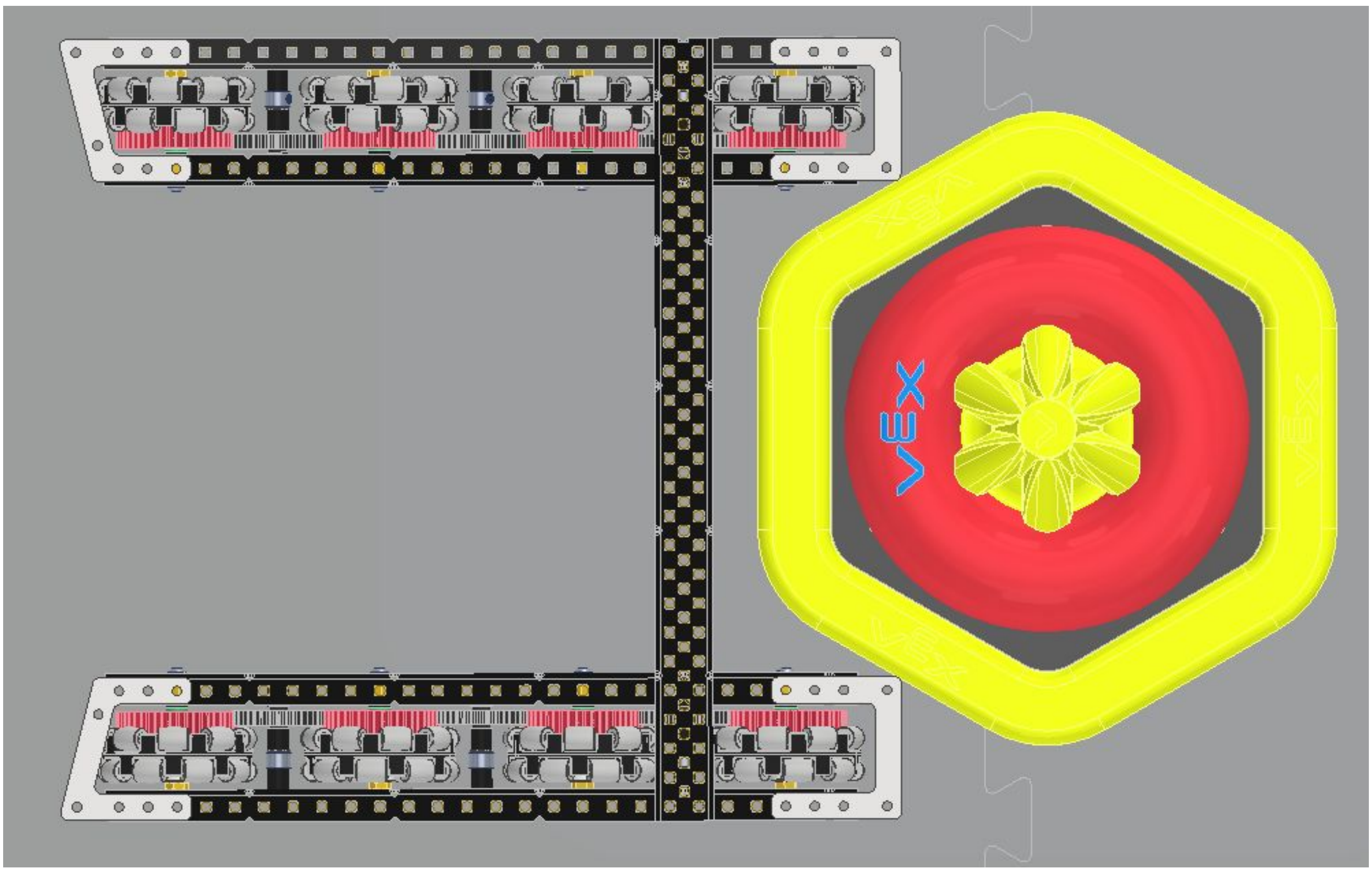
2024-09-01

▶ **Problem:** How can we continue optimize the drivetrain and CAD an initial design?

[Plan/Design Solution]

As you can see from the **previous slide**, the two parts are very similar in size, with the wheel barely benign larger. When driving on the mats, the robot will slightly sink into the mats, and we are afraid that the 60-tooth gears attached to the wheels might drag across the mats while driving with the traditional 4 wheels. Having 3 wheels on each side creates more surface area touching the mats, making the weight of the robot distributed more evenly, and causing the robot to sink less.

Alex CADed out a rough preliminary drivetrain, which resulted in the following design, and should be able to easily maneuver over the field. In the finalized CAD, we will add lexan funnels to guide rings into whatever intake mechanism we had before.



Designed by:

Adam X

Witnessed by:

Alex S

Building the Drive

2024-09-03

▸ **Problem:** How can use the materials that we have to build a drivetrain?

[Plan/Design Solution]

After looking at the parts we currently had available, we couldn't replicate the initial CAD, so we decided to use 30-hole c-channels for the length of the robot and a 27-hole c-channel for the width.

7 inches

The rings are 7 inches, which is quite large, we means need a lot of room on the middle of the robot to be able to store it, and Alex calculated the we needed a minimum of 27-hole wide. Since the main goal of this robot is to be maneuverable and quick, we decided to go with the minimum 27-holes.

[Identify Problems]

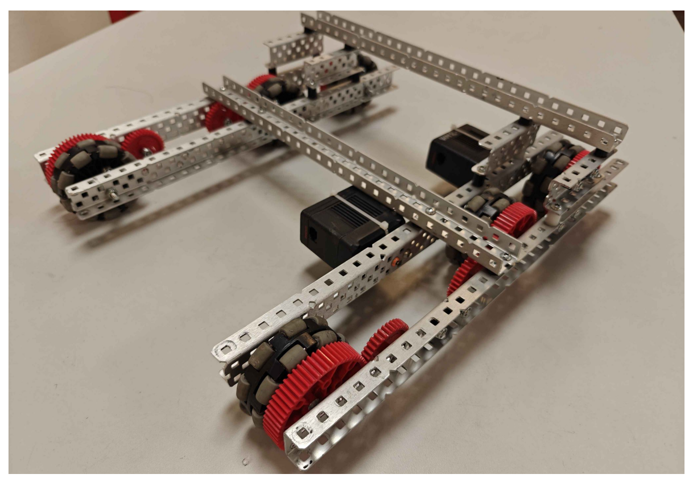
After planning out where the gears and wheels would go, we ran into another problem: **there was no good orientation.** To solve the issue, we decided to incorporate 36-tooth gears to connect the 60-tooth gears with the 2.75-inch wheels to the 48-tooth gears. Due to the uniqueness of our drivetrain, we also needed to incorporate a motor stack as well.

[Plan/Design Solution]

After day 1, the drive looked like this:

We had most of it done and there were a few things left to do:

1. **Attach the all the 36-tooth and 48-tooth gears**
2. **Make sure its structurally stable and not bent**
3. **Tune the drivetrain and reduce friction**



Designed by:

Adam X

Witnessed by:

Kevin Z

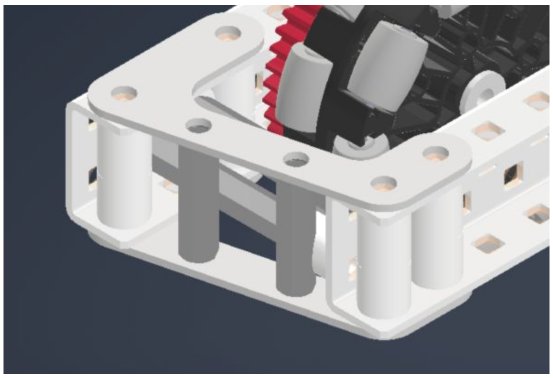
Intake Funnels

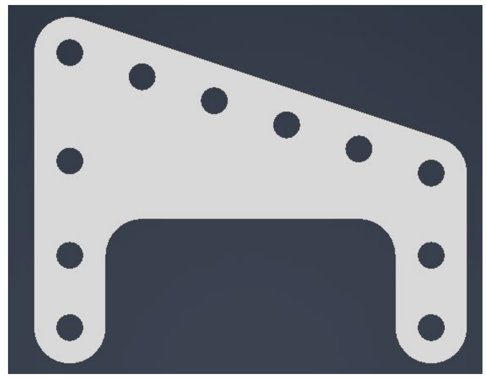
2024-09-03

▶ **Problem:** How can use the materials that we have to build a drivetrain?

[Plan/Design Solution]

After finishing all of that, we worked on the funnel, which would be attached to the drivetrain and would funnel rings into the intake while driving. This was the CAD:





The reason why we added so many holes was because we wanted it to be extremely adjustable and adaptable, in case something went wrong so we wouldn't need to make new ones

Designed by:

Adam X

Witnessed by:

Maxwell L

Intake/Scoring Brainstorming

2024-09-04

▶ **Problem:** What are some good intake designs from past years?

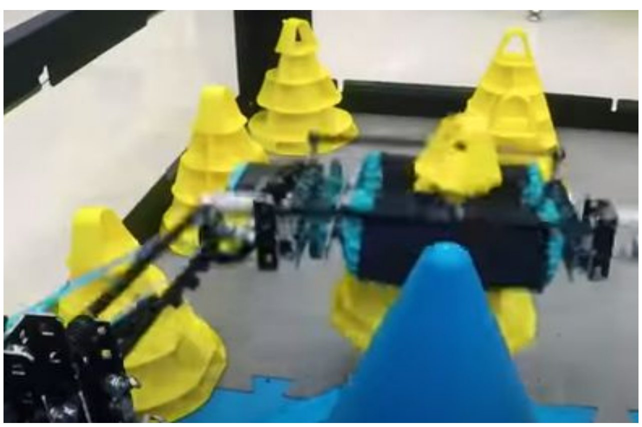
[Brainstorm Solutions]

Moving on to the intake, we had some specific criteria and things that the intake on our robot needed to meet:

- Our intake mechanism need to be able to efficiently score rings onto the goals, avoiding the cap designed to make scoring difficult
- Needs to only use 11 watts (1 11-watt motor or 2 5.5-watt motors)
- Needs to be able to hold 2 rings, since that's the max

In the Zone Intake [ITZ]

The In the Zone robot design can be directly applied to the High Stakes game, as you can score the same way in High Stakes as you can in In the Zone. The robot gains possession of a mogo and begins picking up cones and stacking them on top of each other, and the cone stacking mechanism also works on the stationary goals/pegs. Every aspect in this approach to scoring applies in High Stakes, and we can stack rings either one or two at a time with a similar arm onto mobile goals. This design is the most versatile since it can be for wall stakes as well, not just mobile stakes



48180S's In The Zone Cone Intake: *https://www.youtube.com/watch?v=A4hZSWBtsNE*

Designed by:

Adam X

Witnessed by:

Maxwell L

Intake/Scoring Brainstorming

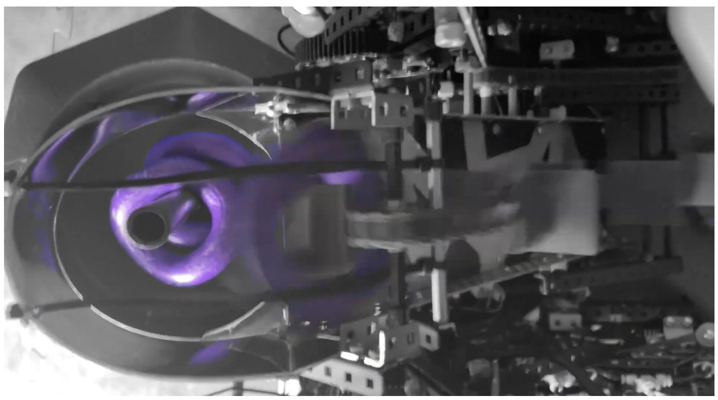
2024-09-04

▶ **Problem:** What are some good intake designs from past years?

[Brainstorm Solutions]

Tipping Point Intake [TIP]

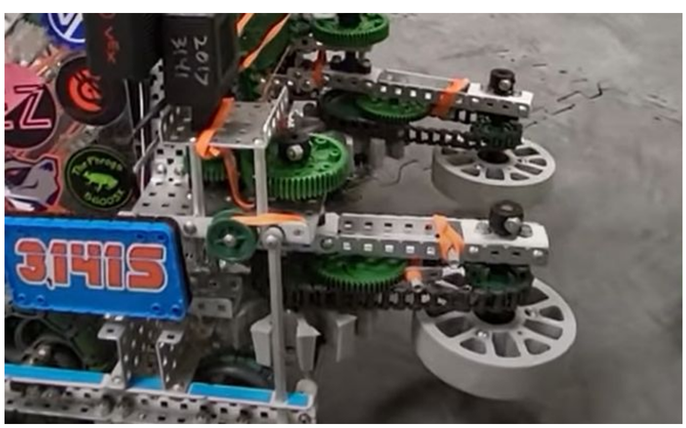
The Tipping Point intake uses flaps to carry rings up a ramp made of two 2-wide angle bars with a lexan piece in the middle. Since Tipping Point also consisted of scoring rings onto mogos, this intake style might be viable for High Stakes. The size of the intake would definitely need to be scaled up since the rings from this year are much larger than in Tipping Point, be that should be a simple adjustment. One problem with High Stakes is that mobile goals have a cap that prevents rings form easily being scored, which wasn't a problem in Tipping Point since it didn't exist.



210Y's Tipping Point Robot Intake

Change Up Style Intake [CGUP]

This intake uses two flex wheels oriented parallel to the ground on two flippers, and has been used previously to intake large balls around the same size as the Triballs in this game
Alongside this, early season reveals such as **3141S** have shown the intakes to work extremely well.



3141S's Ri-3Y Change Up Intake: *https://www.youtube.com/watch?v=LzDuHnSnENA*

Designed by:

Adam X

Witnessed by:

Richard F

Intake/Scoring Brainstorming

2024-09-05

▸ **Problem:** What are some good intake designs from past years?

[Brainstorm Solutions]

Tower Takeover Intake [TT]

Similar to the Change Up Style Intake, it uses two parallel bars with flaps on the side. One key difference between the two intakes is that the intake is mounted on hinges, allowing them to move side to side ever so slightly. While this intake worked very well for cubes, the Tower Takeover cubes were much smaller than the current game elements, and had flat sides, allowing for more surface area.

One key benefit of this intake is the fact that it can pivot upwards due to the fact it is mounted on a two bar. This could be extremely useful for match loading, if the intake was pivoted up and simply rotating extremely fast to deflect triballs into the offensive zone.

210Y's Tower Takeover Robot Intake: *https://www.youtube.com/watch?v=TMcjeYqMToE*

Claw [CLAW]

During Tipping Point, 3141B used an enormous claw to grab onto the mobile goals. However, we believe that this mechanism could also function as an effective mechanism for scoring due to the fact that a claw can easily hold the triballs due to their round shape. Alongside this, the claw can be powered with a single motor, allowing for the other motors to be used for powering the lift which the claw is attached to, or for strengthening other subsystems.

3141B's Clawbot:
https://www.youtube.com/watch?v=e7AAfQYu4B4

Designed by:

Adam X

Witnessed by:

Kevin Z

Intake/Scoring Brainstorming

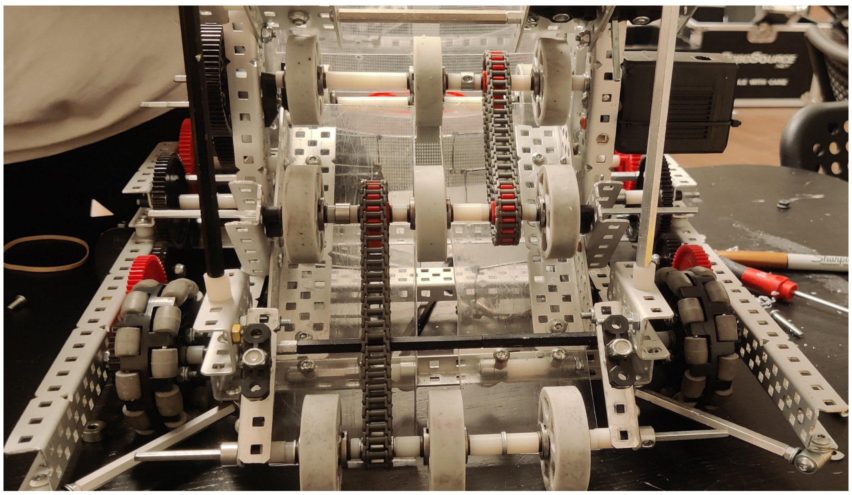
2024-09-05

▶ **Problem:** What are some good intake designs from past years?

[Brainstorm Solutions]

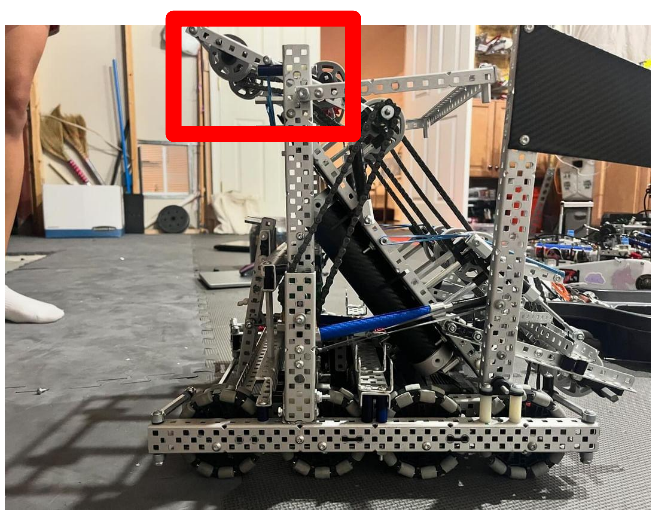
Spin Up Intake/Hood [SU]

The spin up intake is made up of a row flex wheel rollers that carried discs through the robot to a shooter. We thought that this would make a good design because of of the shape of the Spin Up discs somewhat matches the High Stakes well. If we build it well, we should only need one motor to power it, saving motors for other things.



210F's (our) spin up intake

In addition to that, it also has a hood at the end of the intake to push rings through the stake cap when scoring. The hood provides downwards pressure onto the ring so that it goes through the stake cap



2072X's hood

Designed by:

Adam X

Witnessed by:

Maxwell L

Intake/Scoring Evaluation

2024-09-05

▶ **Problem:** Out of all the previous intakes mentioned, which one is the best?

[Select a Solution]

Here is our resulting decision matrix comparing the 6 intake designs

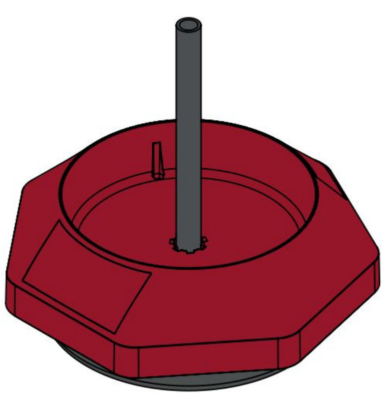
	Scoring Ability	Weight	AWP	Cycle Time	RANK // PTS
SU	1	2	2	1	1st // 6
TIP	1	2	2	2	2nd // 7
ITZ	2	3	1	4	3rd // 10
TT	3	2	3	3	4th // 11
CU	5	2	5	3	N/A
CLAW	5	4	3	2	N/A

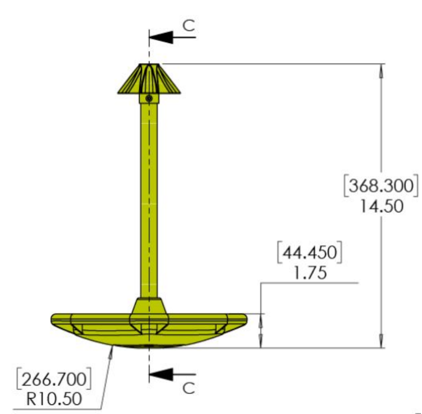
Spin Up Intake: *SU*
Tipping Point Intake: *TIP*
In The Zone Intake: *ITZ*
Tower Takeover Intake: *TT*
Change Up Intake: *CU*
Claw: *CLAW*

Each subsystem has been ranked from 1-5 based on how well they can achieve the given task. As of right now, all metrics are theoretical, and may be subject to change.

Less points = Better Solution.

We have decided to go with the Spin Up intake. Alongside this, we have completely eliminated the Change Up intake and Claw because of the fact they would need a separate scoring mechanism to move the rings up and score them. However with the other intakes, there is still the issue of applying pressure to the top of the mobile goal to push rings past the cap.





Designed by:

Adam X

Witnessed by:

Richard F

Mobile Goal Holder Brainstorming

2024-09-07

▸ **Problem:** What are some good designs from previous years to hold mobile goals?

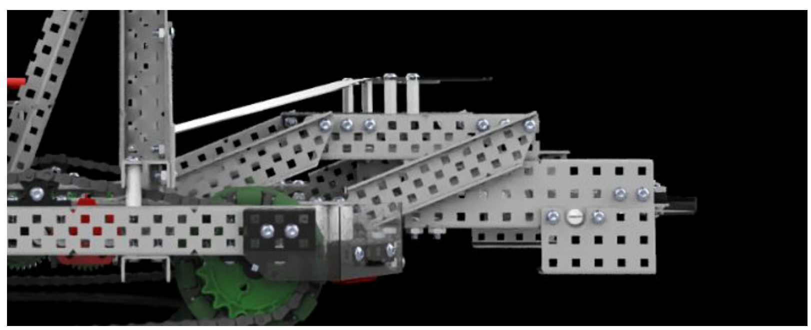
[Brainstorm Solutions]

We came up with 2 solutions, each will be graded on the following criteria:

- How quickly can the mechanism take a mogo?
- How heavy is the mechanism?
- Can it be made with minimal c-channel and motors?
- Does it take up a lot of space within the robot?

In the Zone Mechanism [ITZ]

The in the zone mechanism consists of a 4-bar linear lift and turned it sideways. This design is more complicated and will most likely need 2 pistons for sufficient power. The main advantage for this design is the fact that it is a linear system and keeps the goal level. When level, the goals are more predictable and easier to manipulate as the different height goals are only different in height and not position. Photo taken from 5225A's ITZ reveal.



5225's *In The Zone Mogo Lift*

Designed by: Adam X

Witnessed by: Alex S

Mobile Goal Holder Brainstorming

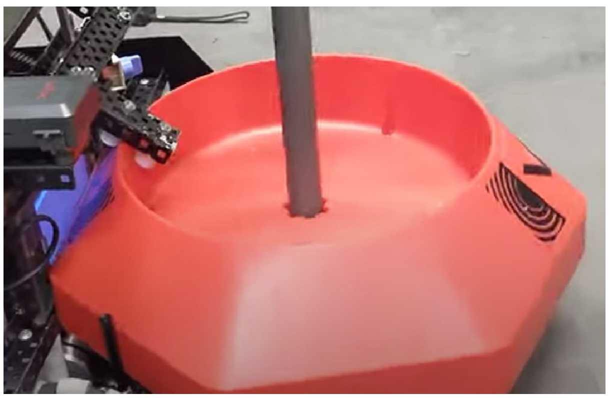
2024-09-07

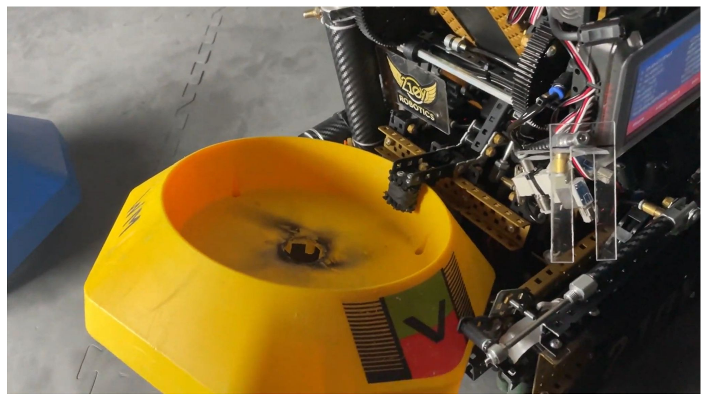
▶ **Problem:** What are some good designs from previous years to hold mobile goals?

[Brainstorm Solutions]

Tipping Point Tilter [TIPT]

The tipping point tilter has multiple designs, but they all have the same concept. There is some sort of piston clamp on the back of the robot that slightly tilts the goal into the robot, and tilts the tip of the goal to the end of the intake conveyor. Some teams only used had a single c-channel clamp, while others had a platform that went under the goal that tilted it, along with a clamp. The geometry of a High Stakes goal is similar to a Tipping Point goal, so this design seems like it would work well in High Stakes.





Designed by:

Adam X

Witnessed by:

Richard F

Mobile Goal Holder Evaluation

2024-09-07

▶ **Problem:** Which mobile goal holder is the best?

[Select a Solution]

Our analysis has been condensed into the following matrix:

	Speed	Weight	Resources	Space	RANK // PTS
TIP	1	1	1	1	1st // 4
ITZ	2	2	2	1	2nd // 7

Each subsystem has been ranked from 1-2 based on how well they can achieve the given task. As of right now, all metrics are theoretical, and may be subject to change.

Less points = Better Solution.

As shown above, we concluded that the Tipping Point clamp/tilter would be the better design mainly because of the speed and resources. The Tipping Point design uses pistons while the In the Zone design uses motors, which we are trying to preserve since we only get 8, but we have unlimited pistons. This transfers into the speed aspect of each design as well, since pistons are almost instant, tilting the goal extremely fast while the In the Zone 4-bar uses a motor with a torque ratio, making it extremely slow and teams can potentially take the goal in that time.

We plan on using 2 medium pistons for the tilter since we think that only having one piston will be too weak to tile the mogos. This means that we have one motor left for the wall stake mechanism.

Designed by:

Adam X

Witnessed by:

Kevin Z

Wall Stake Brainstorming

2024-09-09

▸ **Problem:** What are some good designs from previous years to score on wall stakes?

[Brainstorm Solutions]

For our wall stake mechanism, we looked at some of the early season robot reveals, as well as some of robots from previous seasons. These are the things we want our wall stake mechanism to do:

Speed

- How fast is the cycle time?
- Can the intake → wall stake mechanism transition be done quickly?
- Can it reach both neutral and alliance wall stake heights?

Consistency

- How consistently can it score into the low goal?
- Is the intake → scoring mechanism transition efficient with minimal jams?
- Is the mechanism simple with minimal moving parts?

Weight

- How heavy is the mechanism? Does it require pneumatics?

Redirect Arm [RA]

The redirect arm works by sitting on top of your intake and has a one-way trapdoor on the mechanism. When the robot intakes a ring, it goes through the trapdoor, and when you reverse the intake, the ring shouldn't fall through the trapdoor and should fall into the holder for the rings. It can hold two rings at a time, and arm, which can be a two, four, or six bar lift, lifts up and puts the rings on the wall stake

252H's *Redirect Arm*

Designed by: Witnessed by:

Wall Stake Brainstorming

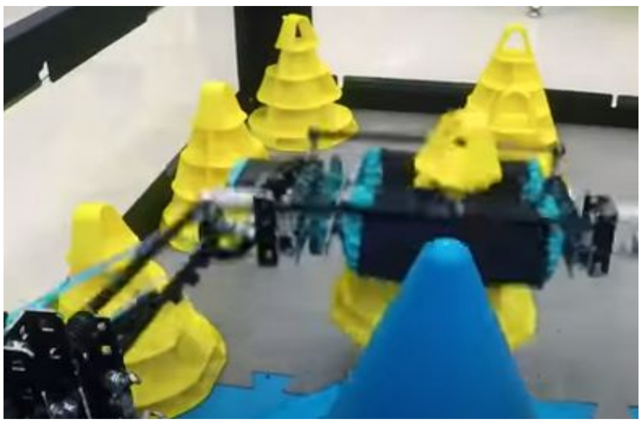
2024-09-09

▶ **Problem:** What are some good designs from previous years to score on wall stakes?

[Brainstorm Solutions]

In the Zone Intake [ITZ]

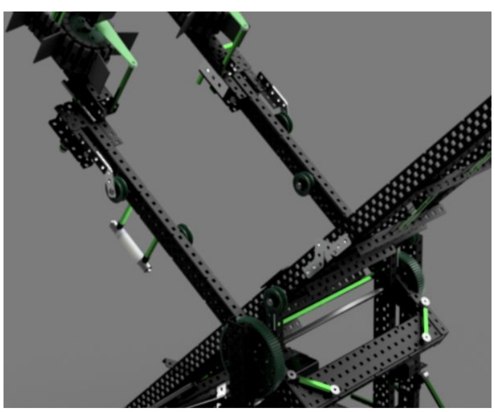
Like mentioned before, the In the Zone intake can score on both mobile and wall stakes, which is why it is mentioned in both the intake/scoring and wall stake sections. The mechanism, if modified implemented in High Stakes, can pick up the cones and lift them to the varying heights of the mobile goals, and alliance and walls stakes.

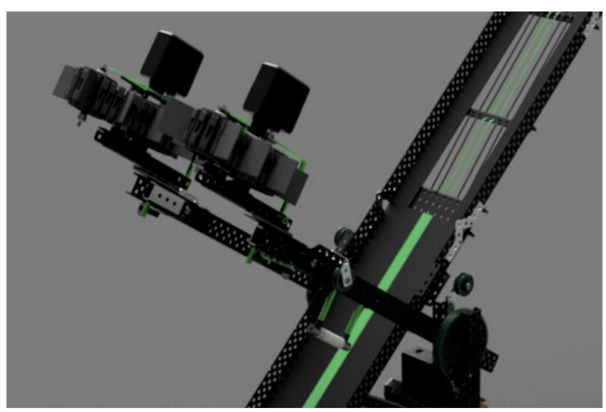


48180S's In The Zone Cone Intake: *https://www.youtube.com/watch?v=A4hZSWBtsNE*

Tower Takeover Intake [TT]

Like the In the Zone intake, the Tower Takeover intake can also have a double use, because of the two-bar mechanism that the intake is attached to. In Tower Takeover, teams had to place cubes into tall towers that are similar height. The two bar mechanism can allow it to function similar to the redirect arm, where it can rotate up and down, and score onto the wall stakes.





7K's *Tower Takeover Two bar*

Designed by:

Adam X

Witnessed by:

Maxwell L

Wall Stake Brainstorming

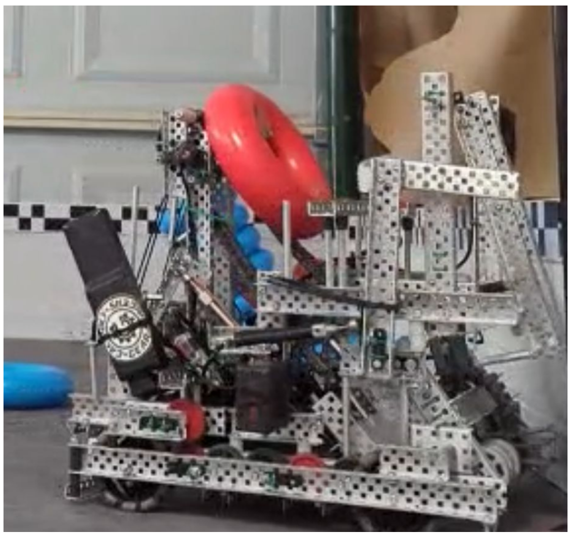
2024-09-09

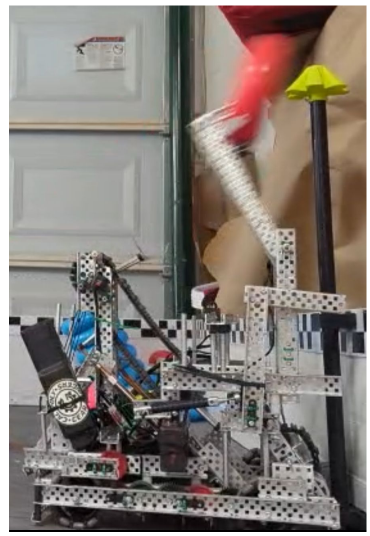
▶ **Problem:** What are some good designs from previous years to score on wall stakes?

[Brainstorm Solutions]

CLPSO Mechanism [CLPSO]

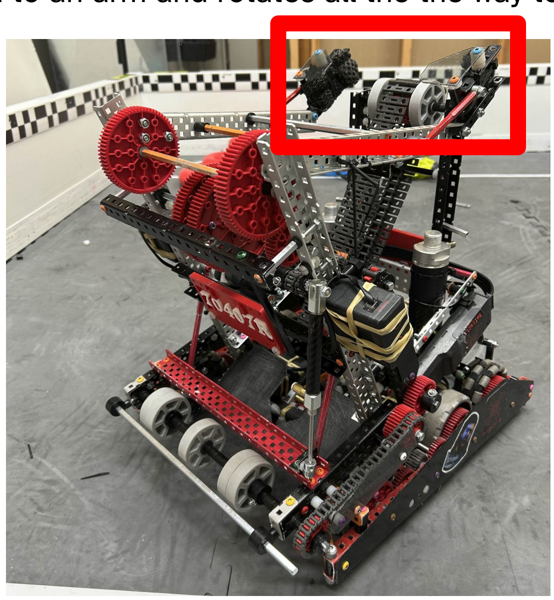
The CLPSO mechanism was made by the VEXU team CLPSO, and consists of an arm that rotates and hooks onto a ring from a hook intake, and rotates through into the wall stake. The problem with this is it has a fixed height, since the positioning and range of motion is not the most optimal.





18522R Mechanism (Lady Brown) [LB]

This mechanism was made by team 18522R, and it's called the "Lady Brown" mechanism. It works by having a mechanism that can hold rings at the end of the intake, so the rings from the intake go into the ring holder, which is attached to an arm and rotates all the the way to the front of the robot to score onto the wall stakes.



Designed by:

Adam X

Witnessed by:

Kevin Z

Wall Stake Brainstorming

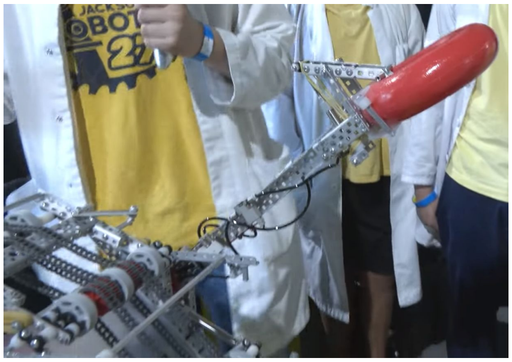
2024-09-09

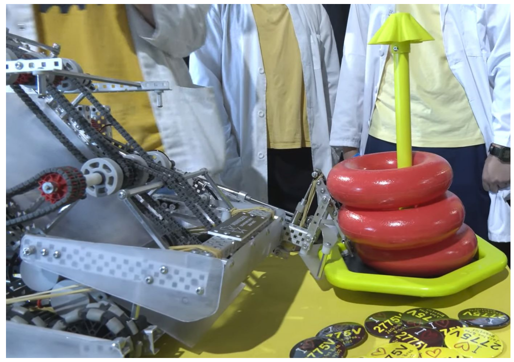
▶ **Problem:** What are some good designs from previous years to score on wall stakes?

[Brainstorm Solutions]

Arm

The arm mechanism is self explanatory given its name, it is just a rotating arm powered by a motor that can grab and score rings, as well as other functions like grabbing mobile goals. The claw end of it is powered by a piston





2775V's Claw Arm Mechanism

[Select a Solution]

Here is our resulting decision matrix comparing the 6 wall stake designs

	Scoring Ability	**Weight**	**AWP**	**Cycle Time**	**RANK // PTS**
LB	1	2	3	2	1st // 8
Arm	3	3	2	2	2nd // 9
CLPSO	5	1	1	2	3rd // 9
RA	2	4	4	3	4th // 13
ITZ	3	5	2	3	5th // 13
TT	3	4	3	3	6th // 11

Lady Brown: *LB*
Arm: *Arm*
CLPSO Mechanism: CLPSO
Redirect Arm: *RA*
In The Zone: *ITZ*
Tower Takeover: *TT*

Each subsystem has been ranked from 1-5 based on how well they can achieve the given task. As of right now, all metrics are theoretical, and may be subject to change.

Less points = Better Solution.

Designed by:

Adam X

Witnessed by:

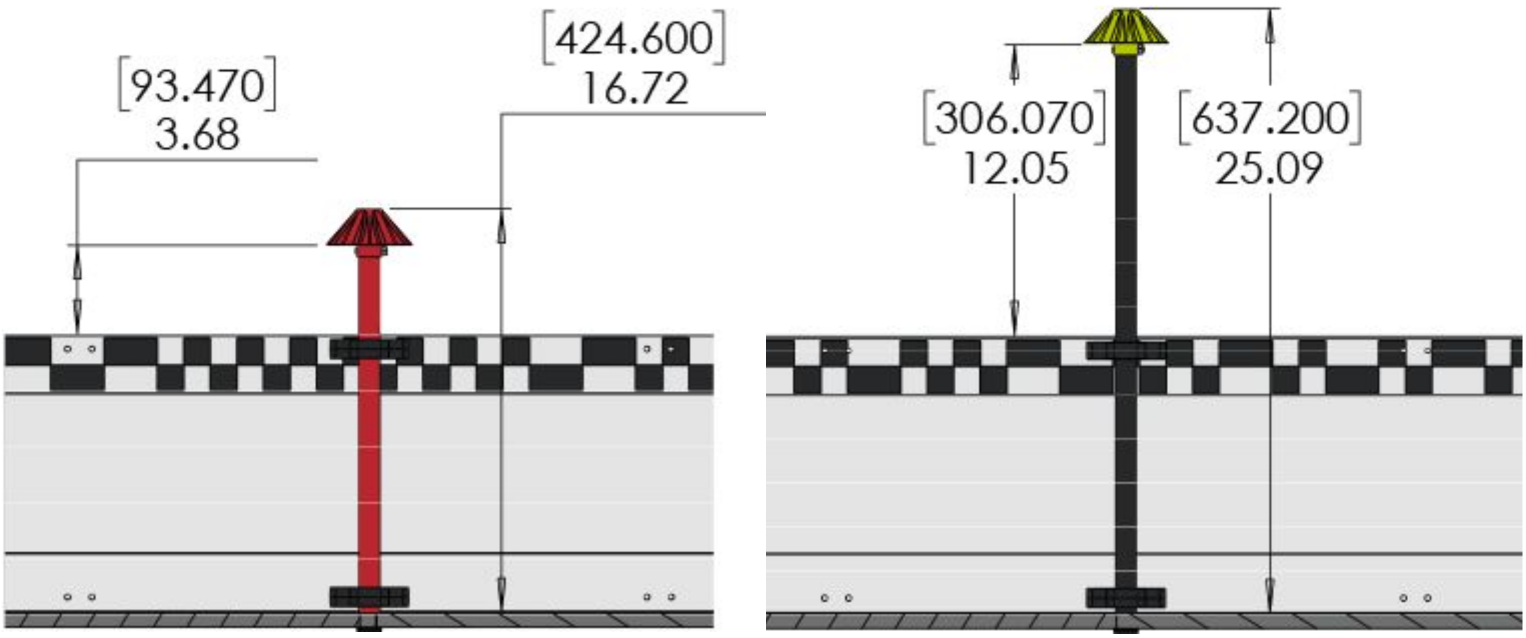
Kevin Z

Wall Stake Evaluation

2024-09-09

▶ **Problem:** Which wall stake mechanism is the best?

We have decided to go with the “Lady Brown” mechanism, or the 18522R Mechanism. Alongside this, we have eliminated the CLPSO mechanism and the In the Zone mechanism. Although extremely fast, the CLPSO mechanism is not very versatile, as it can only score on one height of wall stake, but we want a mechanism that can score both alliance and neutral wall stakes, which are at different heights. Going with the In the Zone mechanism means that it is also your intake method of scoring rings onto mobile stakes. However, we have already established that we are going with the flex wheel intake previously.



Alliance vs. Neutral wall stake. We want our mechanism to be able to score both.

Now that we’ve selected a scoring method for both wall stakes and mobile stakes, we can start planning out the overall structure of the robot. Since we need an intake ramp, as well as a mounting spot for the wall stake mechanism, we decided to make the main structure of our robot an X shape, shown in the next few pages.

Designed by:

Adam X

Witnessed by:

Alex S

Building the Initial Structure

2024-09-11

▶ **Problem:** How should we build the initial structure of both the wall stake and intake mechanisms?

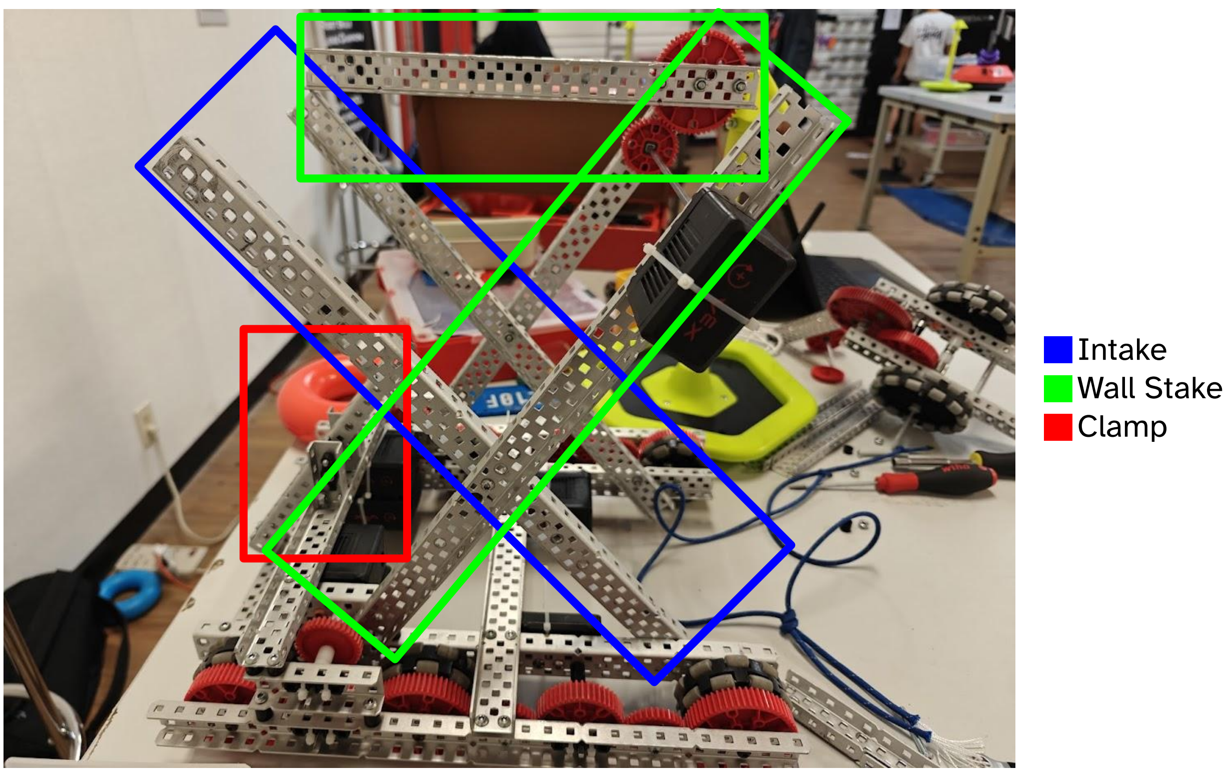
[Plan/Design Solution]

Now that we had a plan of what the subsystems of our robot would be, we began planning out what that would look like on the current drivetrain.

Subsystems we have so far:

1. Intake (flex wheel, hood)
2. Wall Stake Mechanism
3. Clamp (2-piston)

There is definitely room on the robot for more subsystems and mechanisms, which we will brainstorm about in the future. Right now, we want to focus on the critical functions of our robot, and worry about extra functions later.



Initial overall structure of the robot including 3 main subsystems

This is what we came up with. The design could potentially be changed in the future, but we believe that this is the best design as of right now. The following pages explaining the design has a very abstract timeline, as we working on all three subsystems at the same time.

Designed by:

Adam X

Witnessed by:

Alex S

Prototyping the Clamp

2024-09-11

▶ **Problem:** How can we build a good clamp?

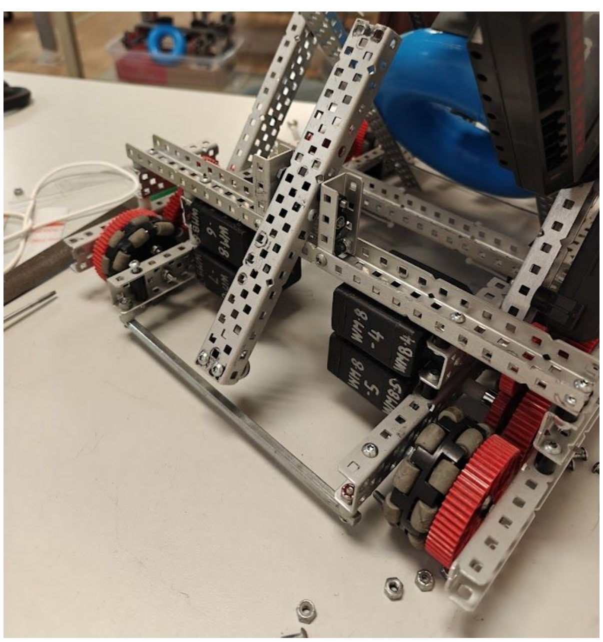
[Plan/Design Solution]

Our methodology was that in order to test the scoring from the intake, we first needed a reliable clamp. For our clamp mechanism, we focused on using mechanical advantage and levers to ensure that we had the strongest possible clamp, so that opponent's clamps can't out power ours.

This was our initial prototype with no pistons yet: It's an extremely simple design, unlike some teams that have incredibly complex structures for their clamps. It consists of a c-channel rotating on another c-channel attached to a cross-bracing for the drivetrain. On the end of the c-channel, there are two screws wrapped in an omni-wheel roller, which is supposed to help grip onto the ledge inside the goal.

[Test Solution]

When testing using our hands, we felt that too much force was required to even tilt the goal slightly, and with pistons, it would use large amounts of air every time. To reduce the force required, we simply needed to increase the amount of mechanical advantage. We did this by extending the other end of the lever, where we were eventually planning to attach the piston.



Extending the c-channel made it much easier to clamp onto the goal and actually tilt it.

Designed by:

Adam X

Witnessed by:

Richard F

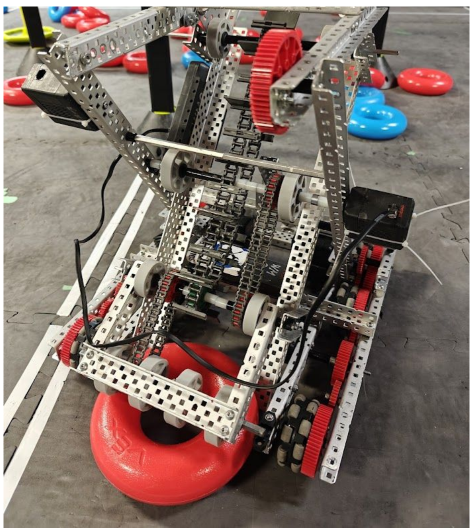
Building the Intake

2024-09-14

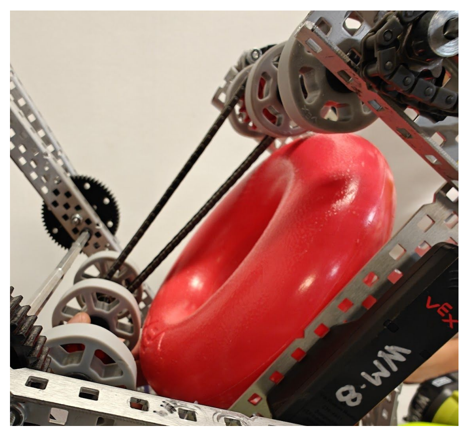
▸ **Problem:** How can we build a good intake?

[Plan/Design Solution]

We had done this intake design many times in previous years, and the concept of it was quite familiar to us. That being said, this design in High Stakes requires some sort of hood mechanism at the end of the intake that acts as a guide for rings and pushes them down through the stake cap. As this part of the subsystem is in uncharted territory, we decided to stick with what we were comfortable with for now.



All of our initial designs are planned to be extremely simple, in which we build off of and become more and more complex over time. The subsystem of 3 stages of flex wheel rollers, and a two-bar are powered by a 600 rpm motor located on the third stage, and the rest are connected through sprocket chains.



[Identify Problem]

A problem that we ran into while testing was that between the top stages, there was a deadzone, meaning the stages were too far apart that there was a section where the ring was not contacting any flex wheels.

[Brainstorm Solutions]

We came up with 2 solutions:

1. **Add another stage**
   Shifting the middle stage up or down, and then building another stage to fill the gap could add more friction if not build perfectly, which we also don't want.
2. **Use a conveyor chain**
   Using a chain with flaps, we can connect the first stage to the last and the flaps would assist the ring upwards to the next stage without adding another.

[Select a Solution]

As seen in the image above, we chose the second solution due to the potential increase of friction from the first solution

Designed by: Adam X

Witnessed by: Maxwell L

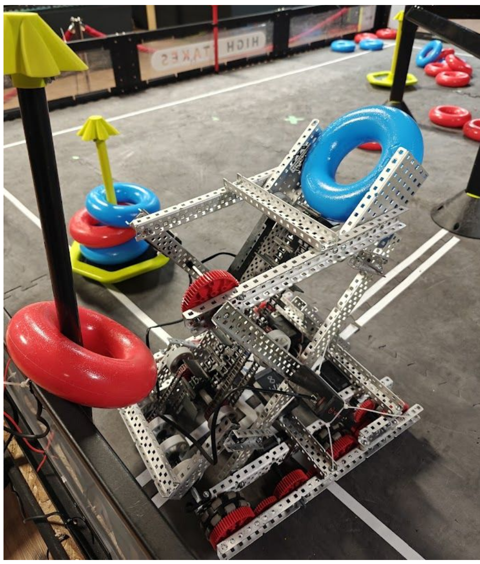
Prototyping the Wall Stake Mechanism

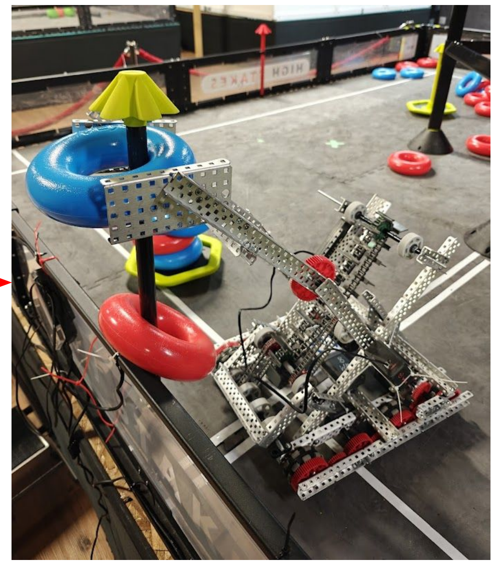
2024-09-14

▶ **Problem:** How can we build a good wall stake mechanism?

[Plan/Design Solution]

Looking at examples of the “Lady Brown” mechanism online, we found that all of the designs were on hook intake robots, but none were on the flex wheel intake we were doing. Despite this, we decided to persevere and continue with the plan. While developing the intake, we were also working on the wall stake mechanism, and we built an initial prototype:





The intake would carry a ring up to the end, and when we want to score on a wall stake, the mechanism would rotate to the end of the intake, and the ring would get held by friction from the c-channels on the side. Then, when scoring, the mechanism would rotate and the ring, still held by friction, would go onto the wall stake, and the robot could just drive backwards from the image on the right to go fulfill other tasks during a match.

[Test Solution]

Testing it with our hands, it worked well enough, and we need the intake and hood completely finished to develop the wall stake mechanism further, so we will most likely be allocating the next few weeks to that.

Designed by: Adam X

Witnessed by: Alex S

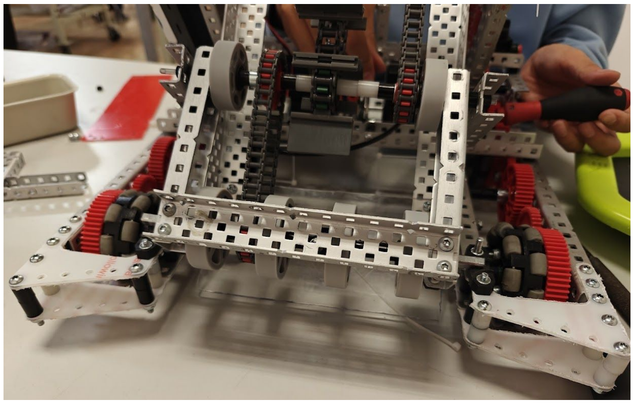
Finishing the Intake

2024-09-18

▶ **Problem:** The intake is unfinished

[Plan/Design Solution]

To finish the intake, we still needed a few things done:



1. **Mount the intake funnels**
   On **page 23,** we cadded out funnels for the intake. Their function is self explanatory, when driving into rings, they funnel rings into the center of the intake where the flex wheels are. In addition to that, they also serve as a guard to our wheels. We quickly cut them out using our CNC machine and mounted them.

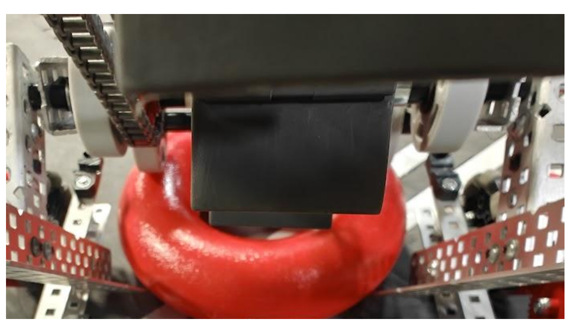
2. **Build the hood**
   This will most likely be the most time consuming. Getting one done shouldn't take too long, but the tuning process to make sure it's consistent will be a challenge.

3. **Add a lexan ramp**
   As of right now, the rings currently slide against the 2-wide angle bars when travelling up the intake, which has quite a bit of friction compared to using other materials like lexan.

[Identify Problem]

When testing with a motor on what we currently have without the hood, we discovered that the angle of the ramp was too steep, and that the angle that the ring had to change was extremely large. Because of this, the ring completely stops and the bottom of the ramp and doesn't go up any further.

The ring is stopped by the corners of the angle bar since the angle change is too steep.



Designed by:

Adam X

Witnessed by:

Kevin Z

Finishing the Intake

2024-09-18

▶ **Problem:** The intake is unfinished

[Brainstorm Solutions]

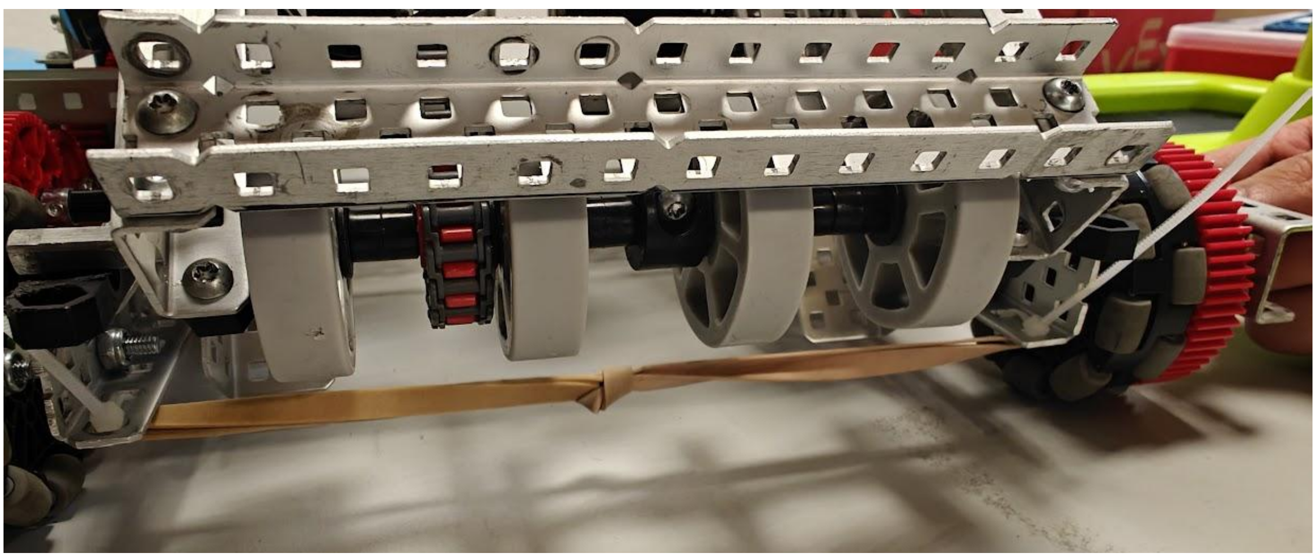
However, we struggled to identify what the best solution to the problem mentioned in the page above, where the intake ramp was too steep. We decided to test and implement all of these brainstormed solutions at the same time, as shown below:

Design #1: Rubber Band

[Plan/Design Solution]

For this solution, we stretched a rubber band across the intake, hoping that it would prop the ring up and make the angle change less steep. This was very common last year during Over Under, as almost every team had a rubber band under their intake to elevate triballs above the ground inside the intake. However, High Stakes rings are nothing like Over Under triballs, and rubber bands may affect them differently.

Implementing this design, it looked like this:



[Test Solution]

Testing this solution, it didn't work at all. Unlike Over Under triballs, the rings this year are not nearly as tall, and can't go over the rubber band, so they just stop.

Designed by:

Adam X

Witnessed by:

Maxwell L

Finishing the Intake

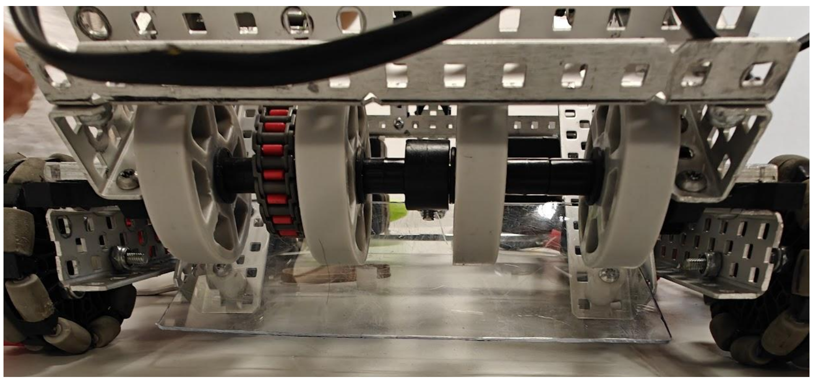
2024-09-18

▸ **Problem:** The intake is unfinished

Design #2: Lexan Ramp

[Plan/Design Solution]

For this solution, we quickly hand cut a lexan piece to go only on the bottom of the intake, as we had cadded out strips for the entire ramp that we were going to CNC out later. The ramp would be at an angle in between the ground and the ramp, acting as an intermediary stage for the ring to go up and change angle.



Using spacers, we were able to adjust the angle of the ramp find the best fit that made rings run through the intake smoothly. We decided that one ½” spacer and a ⅛” spacer worked the best.

[Test Solution]

Unlike design #1, this solution actually worked and improved the intake by great margins. From this, we are now able to further develop the intake and start working on the hood.

[Plan/Design Solution]

This solution will most likely be temporary, as we have used CAD software to design custom lexan pieces that span the entire ramp that we will cut out later.

Designed by: Adam X

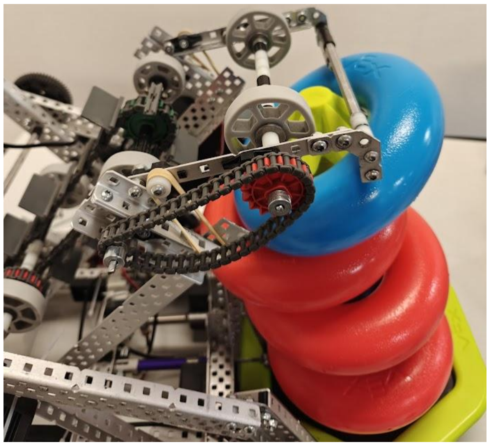
Witnessed by: Maxwell L

Prototyping the Hood/Intake

2024-09-21

▶ **Problem:** How do we build a good hood that can score effectively?

[Plan/Design Solution]



The first design we developed was a two-bar made with angle bars that rotate off of another angle bar attached to a c-channel for the previous stage of flex wheel rollers. It's powered by a sprocket chain to the rest of the intake system and runs at 200 rpm, on a 3:1 ratio. We did this because we believed that the main purpose of the hood was to push rings through the stake cap, which was a torque focused task. Decreasing the speed and thus increasing the torque, we believed it would work better. We also used a rubber band for more downforce on the ring when scoring.

[Test Solution]

During testing, the hood couldn't successfully push down any of the rings through the cap. As you can see in the image, this is because the flex wheels on the hood were in front of the cap, and it pushed one side down, but not the other.

[Identify Problems]

Mentioned above, the flex wheels only pushed down one side of the ring through the cap, but not the other.

[Brainstorm Solutions]

We did not have a great idea on what would work very well and what would not, so we just decided to build whatever solutions came to mind

Design #1: 90-Degree Gussets

[Plan/Design Solution]

As seen in the picture above, this solution consists of adding a 90-degree gusset attached to the end of the hood, and a standoff across. We hoped that since it was behind the cap and reached down lower, it could now push both sides of the ring down.

Designed by: Adam X

Witnessed by: Alex S

Prototyping the Hood/Intake

2024-09-21

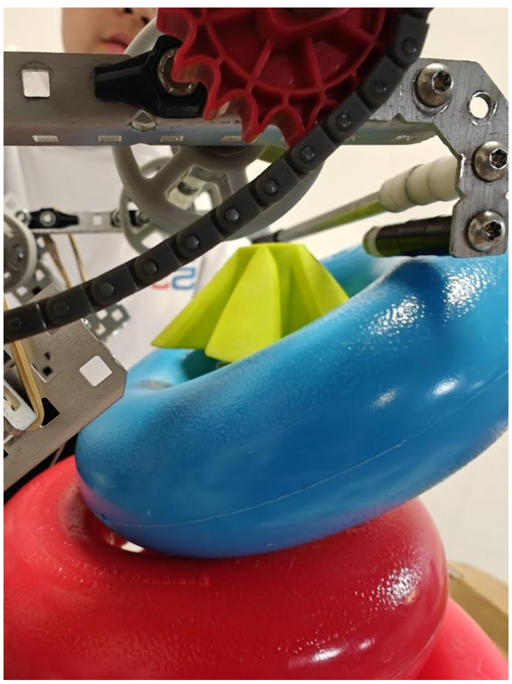
▶ **Problem:** How do we build a good hood that can score effectively?

[Test Solution]

Testing it out, it did work better than without it, but it still didn't reach down far enough to fully push the ring down. We even tried adding another row of standoffs below it, and it still didn't work as it was still too high.

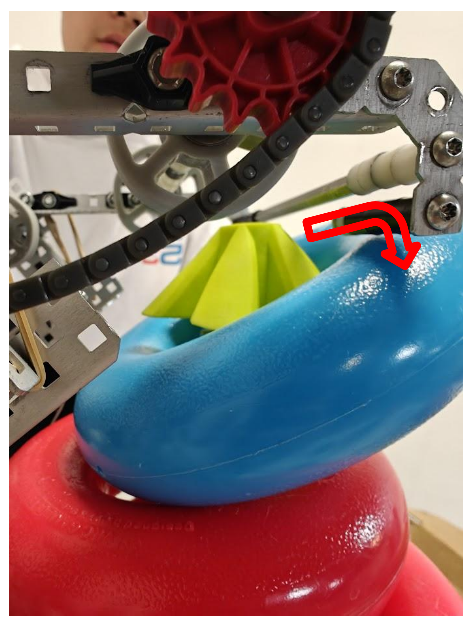
[Identify Problems]

We couldn't just infinitely stack standoffs downwards because the mobile stake has to tilt towards the robot when it's clamped, and if we go too low with the standoffs across, they will block the goal from tilting into the right position. We also tired playing around with 60 and 45 degree gussets as well, but ultimately and rinto the same problem.



Even with the standoffs, the hood still doesn't push the ring far down

From the image, the path that the mobile stake cap travels when tilted by the clamp is marked by a red arrow. If the standoffs go too far down, they will restrict the goal's path.



Designed by:

Adam X

Witnessed by:

Alex S

Prototyping the Hood/Intake

2024-09-22

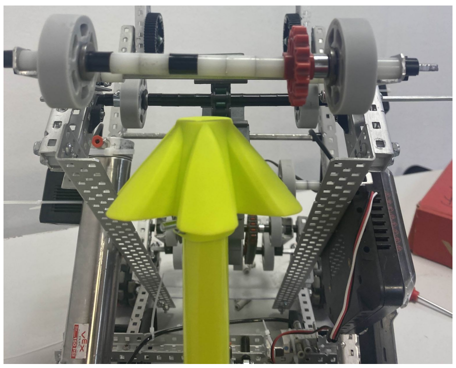
▶ **Problem:** How do we build a good hood that can score effectively?

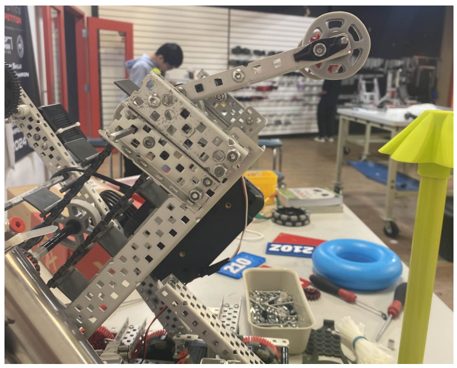
Our next solution was to ditch the gussets completely, since any solution we made involving those prevented the clamp from functioning.

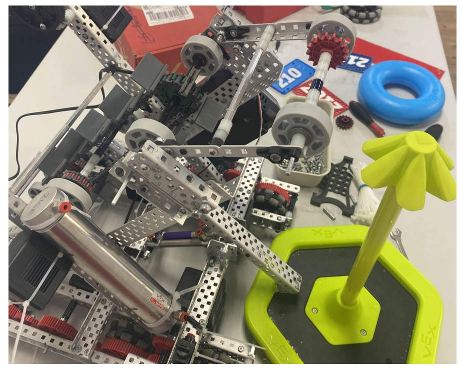
Design #2: Angle Bar Only

[Plan/Design Solution]

This solution only involves the two-bar mechanism made from angle bars that have been moved to the right. We did this in hopes that it would be far back enough that this time, the hood would push the ring all the way down through the cap.







[Test Solution]

Testing this solution out, we still faced the same problems as before, as it just wasn't far back enough. However, it was significantly better than before, and we can most likely improve on this situation and make it work.

Designed by:

Adam X

Witnessed by:

Alex S

Fixing the Intake

2024-09-23

▸ **Problem:** The flaps on the intake don't do very much

[Identify Problems]

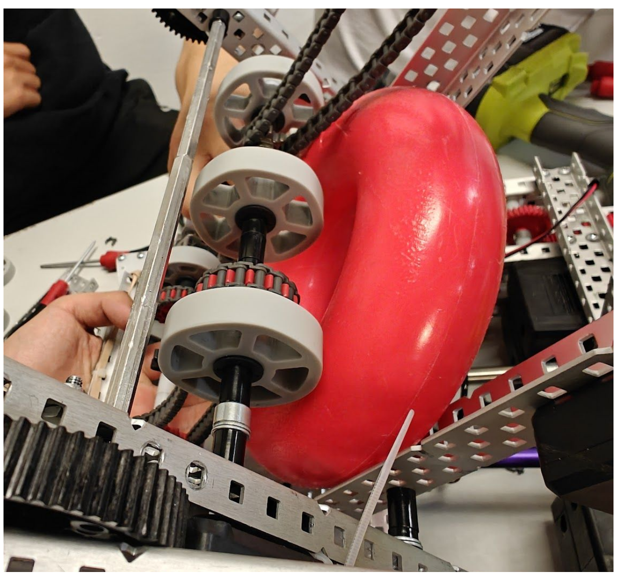
We started running into problems with out intake, where the ring would slow down a lot during when it was being carried by the flaps.

[Brainstorm Solutions]

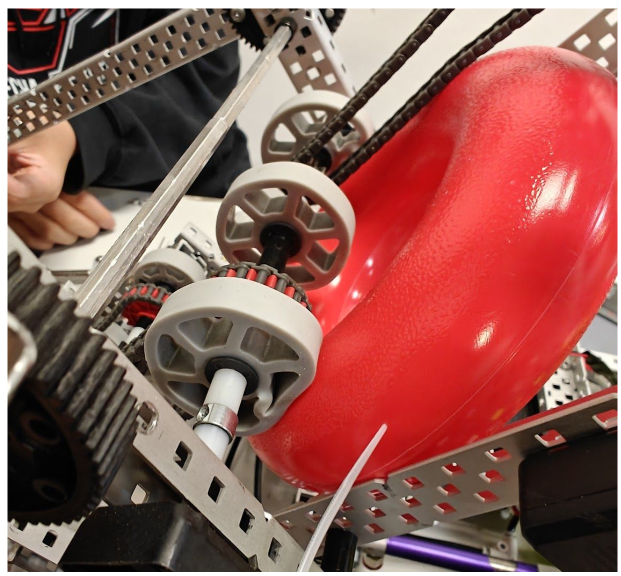
This solution to our intake problem was not working, so we decided to circle back and try another solution. We noticed that the middle stage of flex wheels didn't barely contacted the rings when going up, so we moved it down by one hole.

[Test Solution]

Testing this solution, we can see from the image that if it's one hole up, then then there is barely any contact with the ring. However, if we move it one hole down, then is too much compression between the flex wheels and the ring.



One hole up, no compression



One hole down, too much compression

Designed by:

Adam X

Witnessed by:

Alex S

Fixing the Intake and Hood

2024-09-23

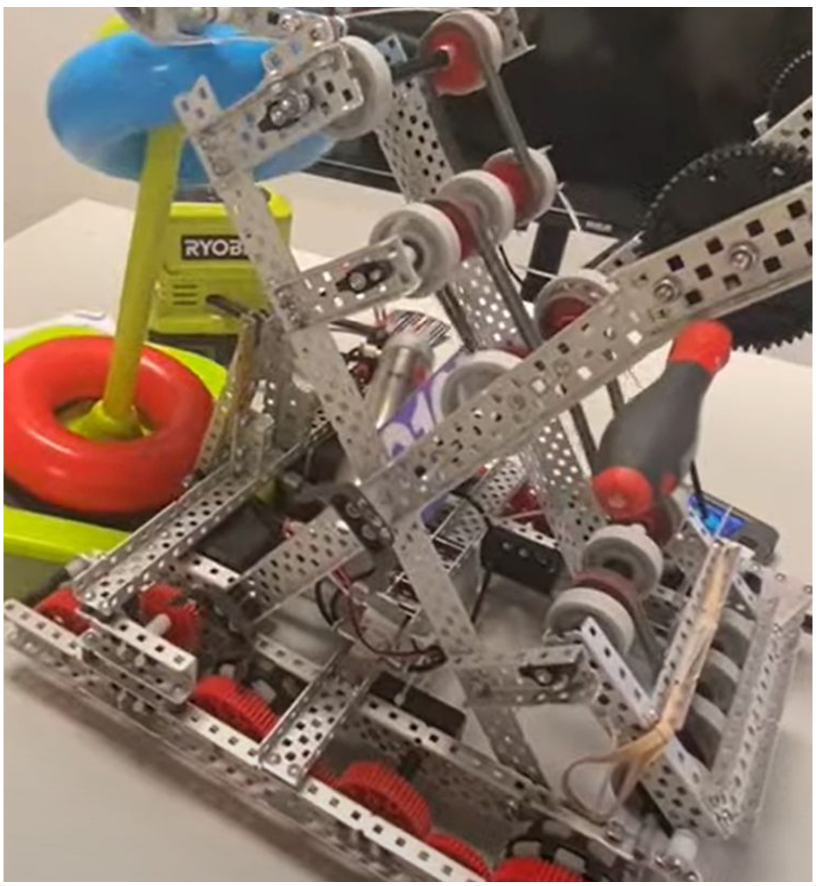
▶ **Problem:** The flaps on the intake don't do very much

[Brainstorm Solutions]

Since the previous approach didn't work, we went back to the first solution we had on on **page 39**, which was adding another stage of flex wheels. Originally, we thought that his was a bad solution due to the additional amounts of friction that would cause the same problem we had with the flaps, but if we built it well enough, then the friction added would be negligible.

[Plan/Design Solution]

We moved the middle stage down a little bit, and mounted it on a perpendicular c-channel to the intake ramp, which helped with the spacing of it. Then, we added a news stage, also perpendicular, between the middle and the last stages.



Now that all of the stages were correctly spaced and had a good amount of compression, we can start developing the hood again.

Designed by:

Adam X

Witnessed by:

Maxwell L

Fixing the Hood

2024-09-23

▶ **Problem:** The hood still doesn’t work

[Identify Problems]

The current problem we are facing with the hood is coming from two things:

1. **The hood doesn’t reach for down enough**

   Mentioned countless times before, there isn’t really a feasible solution to this that maintain the function of other subsystems, as I have talk about previously.

2. **The hood doesn’t provide enough down force**

   If the hood provided a strong enough down force on the ring, then it wouldn’t need to reach all the way down, since the ring’s momentum would be enough to push it through.

[Brainstorm Solutions]

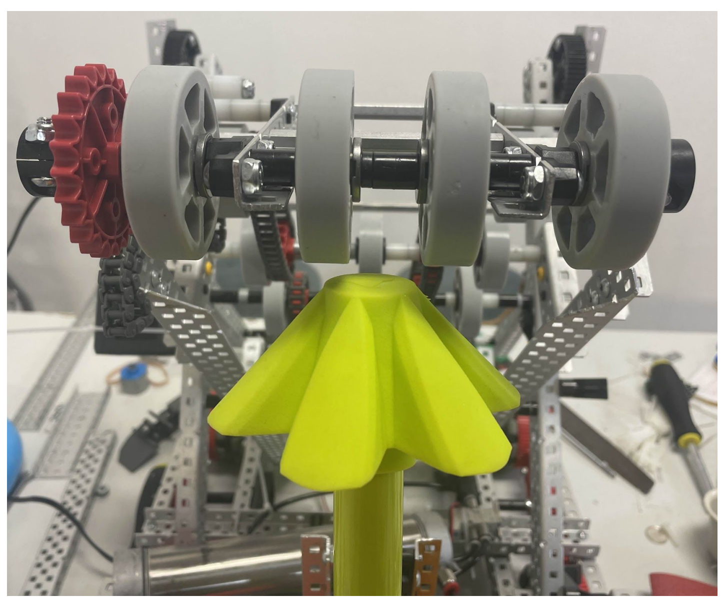
We already tried tackling the first problem, and found no success, so moving on the second, we developed some solutions. The two main ones are **#1, increasing the mass, and #2, increasing the banding**. Both of these solutions provide more down force onto the ring, which gives the ring more momentum to push through the stake caps.

[Select a Solution]

We decided to go with **solution #1**, increasing the mass of the hood. We planned to do this by increasing doubling the amount of flex wheels on hood, which would also provide more surface area contacting the rings, also increasing the chances of it working.

[Test Solution]

Testing it, we found more success that previously. However, it still isn't working 100% of the time. Often, the ring would sit halfway through the cap, and just a shake of the bot could drop it down. We decided to move the flex wheels to the outsides, so that it could drop lower and push the sides around the cap down.



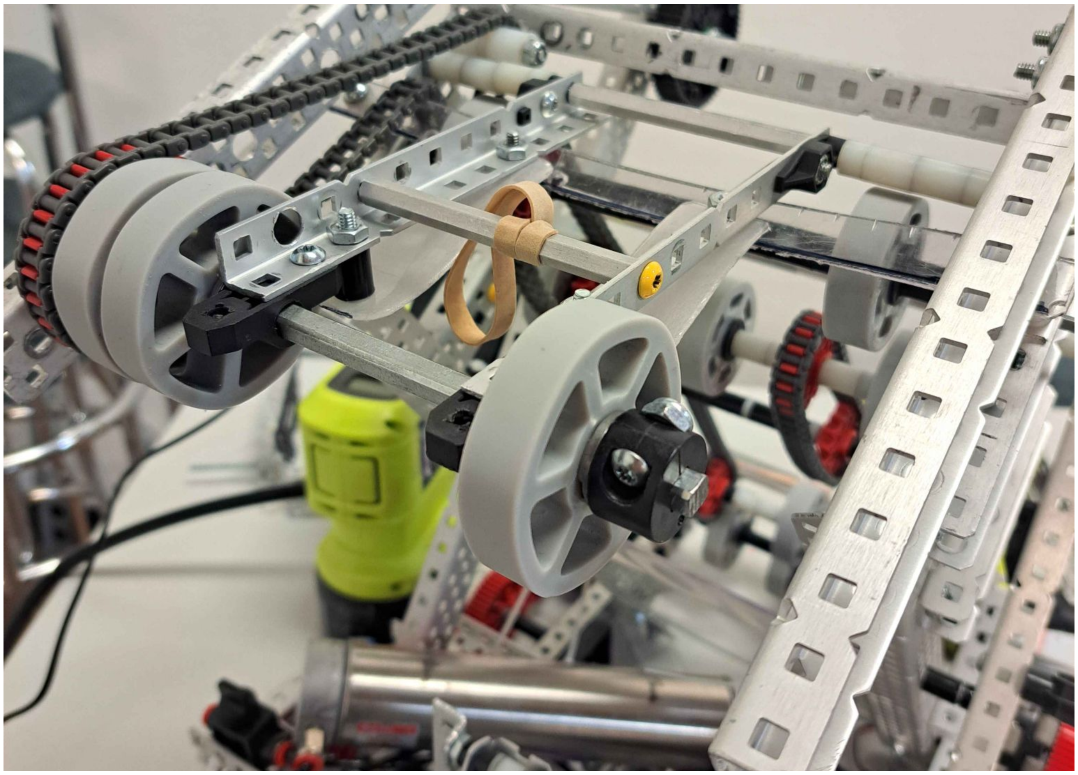
Designed by: Adam X

Witnessed by: Richard F

Fixing the Hood

2024-09-23

▶ **Problem:** The hood works, but not as good as we can make it



[Test Solution]

After making the changes mentioned in the previous page, the hood worked 100% of the time, was super efficient, smooth and consistent. This is what we want. We tried it 6 different times, intaking 6 rings, filling up a goal, and it worked all 6 times. Next steps for this is testing it while driving around on a field. Feeding the intake with our hands is not the most ideal situation, as the field tiles and driving into rings adds additional factors that are not considered when we hand feed rings.

Designed by:

Adam X

Witnessed by:

Richard F

Fixing the Hood

2024-09-24

▶ **Problem:** During actual driving, our hood has issues

[Identify Problems]

While driving, especially when turning, the ring slants sideways when going on the goal, which makes it not fully enter the goal.

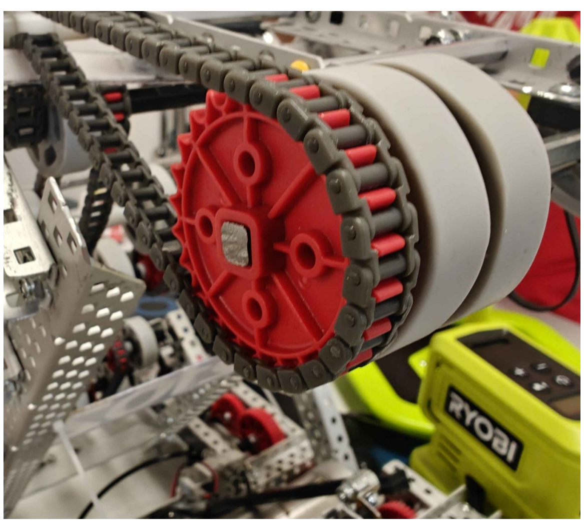
[Plan/Design Solution]

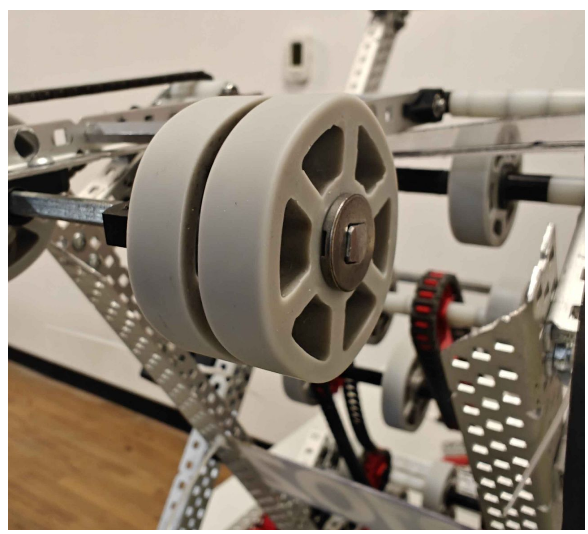
We elected that the simplest and best solution was to add a wheel on the right side of the robot, to even out the weight of the wheel. This should prevent the hood from applying uneven pressure to the ring, which should result in flatter rings.

[Test Solution]

When we added the 4th wheel on the hood, our performance while driving skyrocketed. The ring was much more consistent and it would go on the goal very smoothly.

However, when we did this, we noticed that the hood became very wide and we would have trouble fitting our lady brown as well. To combat this, we decided to remove the axle collars at the ends of the hood, and instead hammer the ends of the high strength axle to prevent the wheels from sliding off. The downside to this solution is that it’s semi-permanent, because we would need to cut the axle in order to get the wheels off.





Note the lack of axle collars on the end of the high strength axle.

Designed by: **Witnessed by:**

Fixing the Clamp

2024-09-28

▸ **Problem:** The clamp is holding us back from developing our intake and hood

[Identify Problems]

Currently, our clamp is super weak. Even at 100 psi, we are still able to move it around and untilt the goal. Our current way of user a lever to clamp the goal is not efficient, so we decided to redesign the clamp.

[Brainstorm Solutions]

One solution is to adjust our mechanical advantage of our clamp, by adjusting where the piston is mounted and the fulcrum of the clamp. However, the drawback to this design is that it would also adjust the angle of the goal, which would force us to redesign the intake ramp/hood

Alternatively, we noticed that the c-channel the clamp was mounted on received heavy amounts of torsion, which decreased the efficiency of the clamp by destabilizing the rotation joint. To combat this, we can try to add supports to the mount to stop it from bending.

Finally, we can just add rubber bands to the pistons, pulling them backward naturally. By doing this, the overall retraction force of the piston is increased. The downside to this option is that the bands may overpower the piston at low psi, leading to the pistons not being able to extend at all.

[Select a Solution]

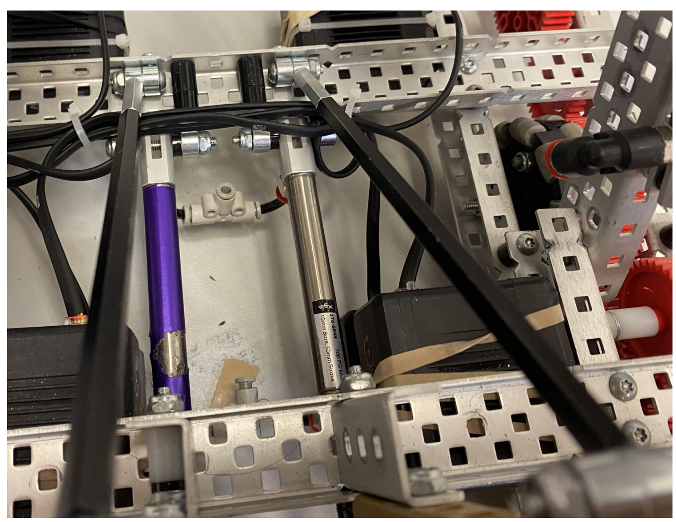
We decided to combine solutions 2 and 3, because they wouldn't fundamentally alter the clamp, but the clamp would still be augmented.

[Plan/Design Solution]

We used 2 standoffs combined with axle collars to support the mountfrom another chassis support c-channel, and we added two rubber bands onto each piston.

[Test Solution]

Our adjustments had good results. With these changes, our clamp was able to function up until ~50 psi while maintaining a good hold on the goal.



The two standoff supports, mounted on a drivetrain c-channel

Designed by:

Adam X

Witnessed by:

Richard F

Building the Wall Stake Mechanism

2024-09-29

▶ **Problem:** The prototype doesn't fit with our robot

[Identify Problems]

Our prototype wall stake mechanism has a few issues:

1. Because the c-channels are facing inward, they hit the edges of the hood.
2. It isn't grippy enough to bring the ring past the hood, which is banded downwards.

[Brainstorm Solutions]

One solution is to add non-slip to the c-channels and flip them. By doing this, we circumvent the issue of the c-channels clipping the hood, and we add grip to the c-channels to allow them to hold the ring well.

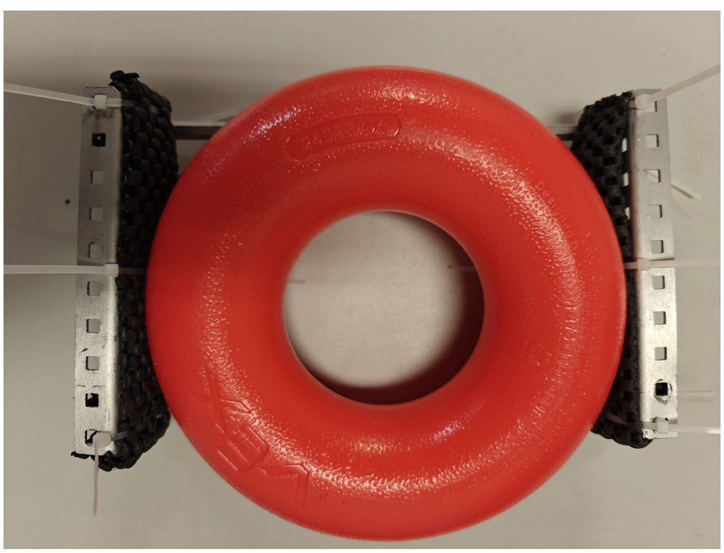
Another solution is to use foam tape instead, of non-slip, and still flip the c-channels.

[Select a Solution]

We went with the non-slip, because we found it to have a much stronger grip than foam tape did.

[Test Solution]

When we used this solution, we ran into many issues. First of all, the intake isn't strong enough to push the ring far enough into the non-slip, so the ring would not be held properly and fall out. Secondly, even if the ring was far enough, it only contacted the non-slip in a small portion. This is because the sides of the ring are curved, so only the very edge of the curvature was contacting the holder. This led to the wall stake mechanism not being very consistent when putting the ring onto the wall stake.



An isolated view of the holder and the ring

Designed by:

Adam X

Witnessed by:

Richard F

Building the Wall Stake Mechanism

2024-09-30

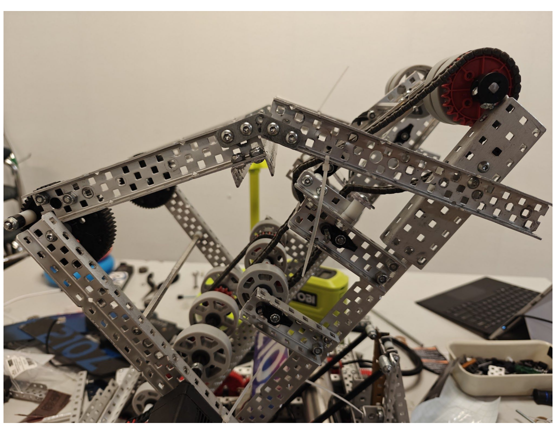
▶ **Problem:** The non-slip doesn't work well.

[Identify Problems]
As mentioned before, the ring doesn't fully enter the holder, and when it does, it barely contacts it.

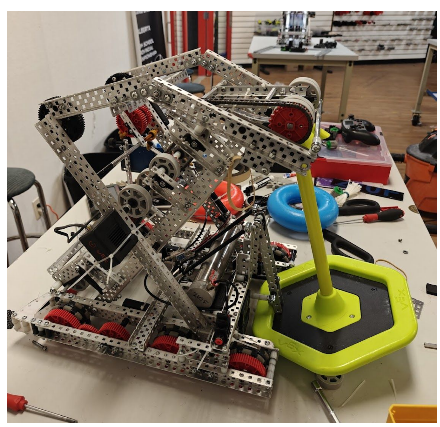
[Test Solution]
Because our only other option was to use foam tape, we tried that instead. To our surprise, the foam tape worked surprisingly well. Because it was compressible, the ring could enter pretty well, and the foam tape curved around the ring, leading to a more comprehensive grip. However, because the non-slip is a bit slippery, the ring sometimes would fall out.

[Plan/Design Solution]
To fix this, we added a small piece of metal below the main c-channel of the wall stake mechanism and put foam tape there as well. This increases the surface area of the ring that is being held. We also lightly wrapped rubber bands around the foam tape, which counteracts the slipperiness of the foam tape. We also placed an elevated standoff at the very end of the holder, for the ring to rest on while being elevated.

[Test Solution]
These changes had a good effect on the performance of the wall stake mechanism. It was able to consistently pull the ring out of the intake, and it scored relatively well as well. Our only downside currently is that it requires a high level of precision in order to score the ring. Otherwise, the ring just falls out of the holder.



Before



After

Designed by:
Adam X

Witnessed by:
Richard F

Tournament Analysis: WM Scrimmage

2024-10-27

▶ **Focus:** Analyze our performance at the Western Mechatronics Scrimmage

Western Mechatronics Scrimmage

6-1-0 : 12 / 27 / 34

Qualifications Rank:	2nd/12
Elims Alliance Partner	210K
Elims Rank:	Tournament Champions
Driver Skills Score:	N/A (No Skills at Event)
Programming Skills Score:	N/A (No Skills at Event)

Match #	**Red Alliance**	**Blue Alliance**	**Score**
Q-2	210Z, 2088W	2088B, 2088	**24** - 13
Q-4	2088A, 2088S	210Z, 2088W	0 **- 36**
Q-8	2088H, 210Z	2088R, 2088U	**30** - 10
Q-11	2088G, 2088R	2088B, 210Z	0 - **30**
Q-14	210K, 210Z	2088G, 2088S	**36** - 3
Q-16	2088G, 2088U	210Z, 2088E	0 - **40**
Q-20	210Z, 2088H	210K, 2088X	8 - **30**
SF 1.1	210K, 210Z	2088A, 2088H	**34** - 0
F 1-1	210K, 210Z	2088R, 2088X	**35** - 11
F 1-2	210K, 210Z	2088R, 2088X	**33** -14

Designed by:

Adam X

Witnessed by:

Maxwell L

Tournament Analysis: WM Scrimmage

2024-10-27

▸ **Focus:** Analyze our performance at the Western Mechatronics Scrimmage

Western Mechatronics Scrimmage

We were in quite a rush to prepare and get ready of the tournament the previous week, but despite that, we still managed to perform quite well and get the job done. The robot held up well and nothing major broke down. We were also able to make observations and analyze how well each subsystem of the robot worked, and how well they all came together. A lot of subsystems were finished and wired at the very last minute the day before (Friday 25-10-2024), as well as the code and autonomous. Another thing was the driver practice and strategy, in which we barely got any and didn't know any of the current game strategies.

Next steps for the Nov. 16 tournament we are hosting would be to refine and improve all subsystems, however we will be redesigning wall stake mechanism completely, as it was extremely nonoptimal and often got jammed. Additionally, really want to focus on autonomous and code, as well as getting lots of driver practice.

Overall, we thought it went quite well, as we ended up winning the tournament, as well as the excellence award despite being extremely unprepared.

What worked:

- Intake and hood worked extremely well, as expected.
- Drivetrain was very fast, agile, and maneuverable.
- Doinker/Arm worked very efficiently at learning corners
- Clamp worked from all angles
- Descoring autonomous worked extremely consistently (10/11 times successfully deployed.)

What didn't work:

- Wall stake mechanism didn't work as planned, was very hard to line up.
- When intaking, the ramp would sometimes bend and rub into the ground.
- Doinker/Arm became extremely bent
- Some programming controls were unintuitive.
- Intake lift wasn't wired, so we couldn't test it out.

Designed by: Adam X

Witnessed by: Alex S

Scheduling

2024-10-27

▸ **Problem:** How will we spend our remaining time leading up to the last 3 tournaments of the year?

This is a rough outline of how we wanted to allocate our time for the next 3 tournaments.

October, November, and December Calendar

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27 Now	28	29	30 Fix Wall Stake Mechanism	31	1	2
3	4 Finish Elevation	5	6	7	8	9 Finishing Touches
10	11	12 Finish Match Autons	13	14 Get Driver Practice	15 Set Up for Tournament	16 Season Opener
17	18	19	20	21	22	23 Intake, Wall Stake, and Clamp Improvements
24	25	26	27	28 Improve Autons, Driver Practice	29 Packing + Tournament Preparation	30 Bowness Tournament
1 Start Robot V2	2	3	4	5	6	7 Finish Drivetrain
8	9 Programming Adjustments	10	11	12 Driver Practice for Majority of the Week	13 Packing + Tournament Preparation	14 STEM I.A. Tournament

After the Bowness tournament, we plan to start a new robot, and simultaneously improve the old.

Designed by:

Adam X

Witnessed by:

Kevin Z

Wall Stake Mechanism Redesign

2024-10-28

▶ **Problem:** How can we remake the wall stake mechanism based on the scrimmage performance?

[Identify Problem]

While practicing using the wall stake mechanism, we found that the robot had to be perfectly lined up with the wall stake or else it would not score the ring. This was a problem in matches, as the high stress and adrenaline environment of a match limits the time to precisely line up the robot with the wall stake, and we often missed the ring. Taking this into account, we had some ideas and improvements that needed to be made to the current design.

1. **The intake doesn't fully load rings into the mechanism.**
   A major problem we faced during the scrimmage last week was that the intake didn't fully push rings into the mechanism, so Maxwell had to hold the intake for longer just so the rings would actually go into the mechanism. We temporarily solved this issue on the weekend by cleaning the hood stage flex wheels, since there was quite a bit of dust and debris on it, which gave the flex wheels more grip in somewhat worked.
2. **We can't go under the elevation ladder.**
   Due to the unusual shape of the mechanism and the fact that it sticks out at the top, we can't travel under the elevation bar. This wasn't a huge problem during the scrimmage, but if teams intentionally put mobile goals or rings under the ladder, we have no way of obtaining them.
3. **The wall stake mechanism was not strong and space efficient.**
   Since we used 45 degree gussets to connect two c-channels of the mechanism together, they bent extremely easily, causing the mechanism to deform and not work properly. Another thing was that the mechanism took up a lot of space, and was extremely large when it didn't have to be. A reason for its large size is because it is offset and not symmetrical due to the sprocket chain on the side of the hood.

From these points, we decided that the best course of action was to completely redesign the mechanism, as there was enough time until the next tournament to do so.

Designed by:

Adam X

Witnessed by:

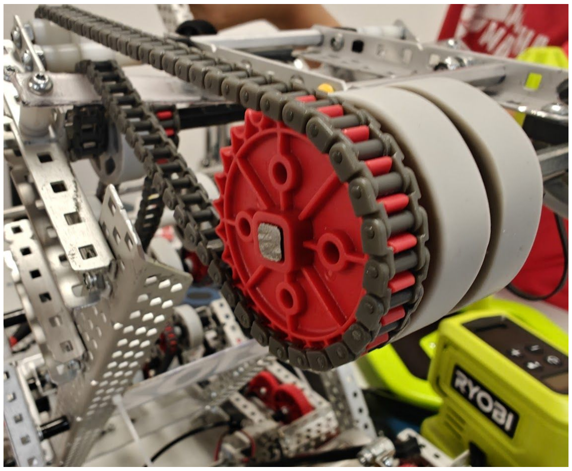
Alex S

Minor Intake/Hood Adjustments

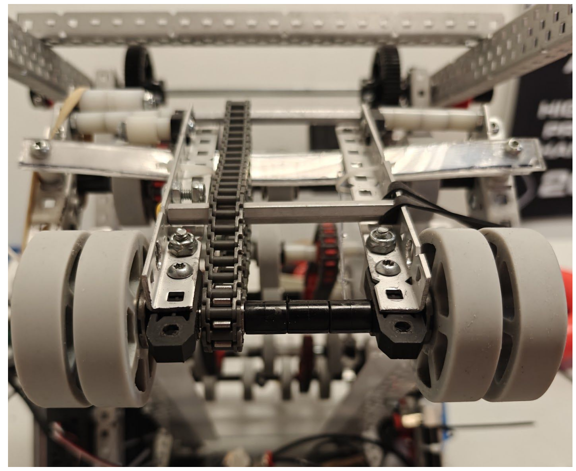
2024-10-28

▸ **Problem:** How can we change the intake to improve the wall stake mechanism?

One of the things we discussed was that in order to make the wall stake mechanism better, we needed to be symmetrical, and in order for that to happen, we needed the hood to become smaller to so the wall stake mechanism could pass through easier.



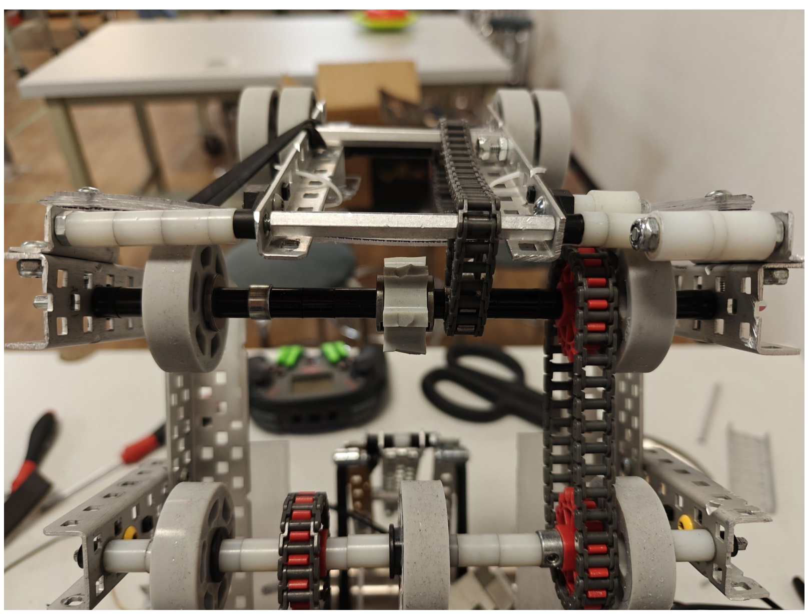
Notice the change from moving the sprocket chain to the inside to make it thinner.



We hammered in another axle to widen then ends and prevent things from falling off so we don't have to use axle collars, thus minimizing the width again.

Other minor changes include:

- Increased the speed of the hood stage, from 200 rpm to 600 rpm.
- Added a tiny cut-out flex wheel on the last stage of the intake/conveyor to help push the ring into the wall stake mechanism more.



Designed by: Adam X

Witnessed by: Richard F

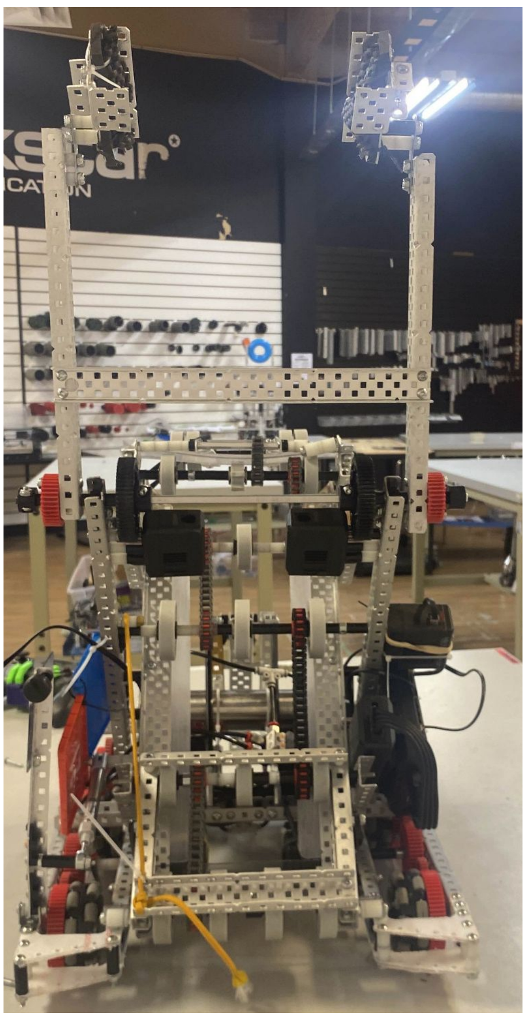
Wall Stake Mechanism Redesign

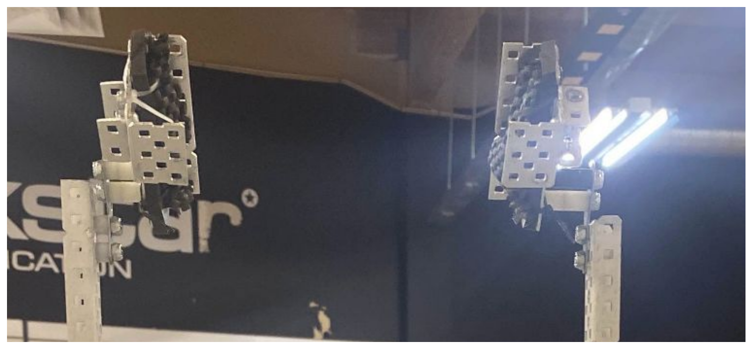
2024-10-28

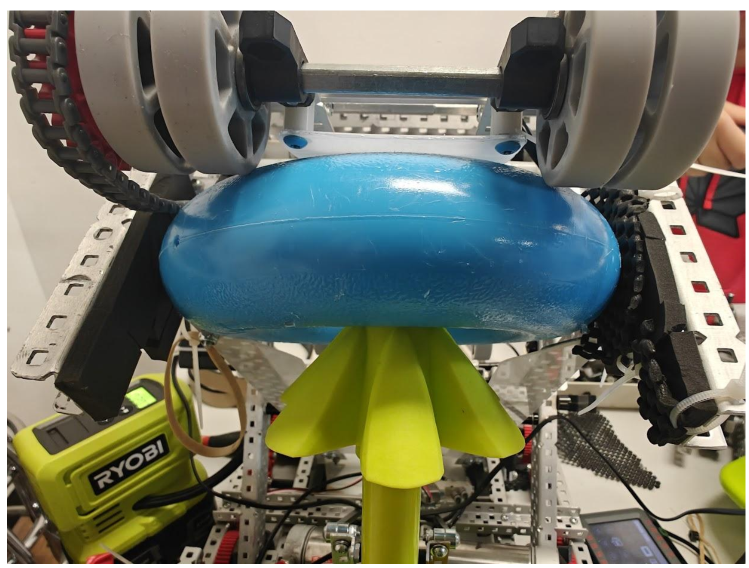
▶ **Problem:** How can we remake the wall stake mechanism based on the scrimmage performance?

Arguably the largest problem regarding the wall skate mechanism that we wanted to tackle was its inconsistency and operability. Often times, when the robot was not perfectly lined up with the wall stake, the ring wouldn't go on and would come right back, or the entire mechanism would just get stuck on the wall stake.

To solve this, we really wanted to reduce the size of it, focusing on making it stronger and more reliable. We took the entire thing apart, starting from scratch, and the first change we made was the general design of it, removing the weak gussets that we had in place originally, and using a single c-channel instead. We also changed the holding part of it, from foam tape c-channels to a 2-wide angle bar with non-slip on it.







Old Version, notice the difference in design

Designed by:

Adam X

Witnessed by:

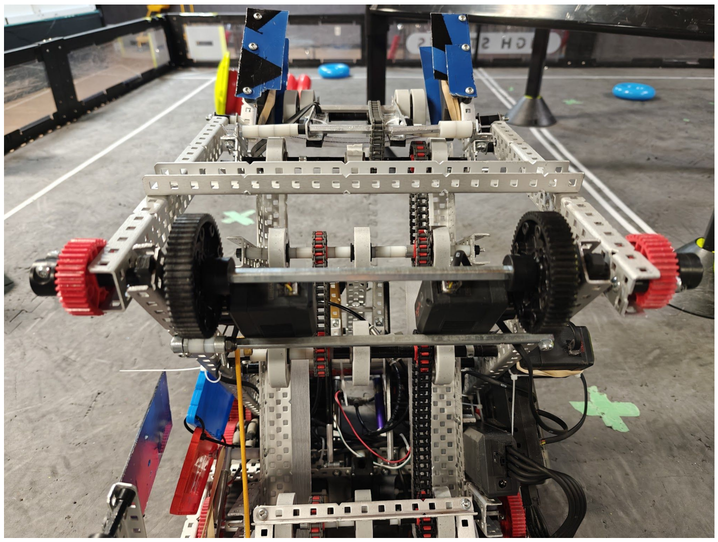
Alex S

Wall Stake Mechanism Redesign

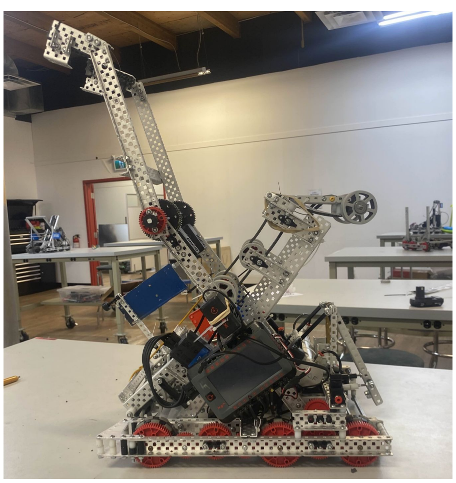
2024-10-28

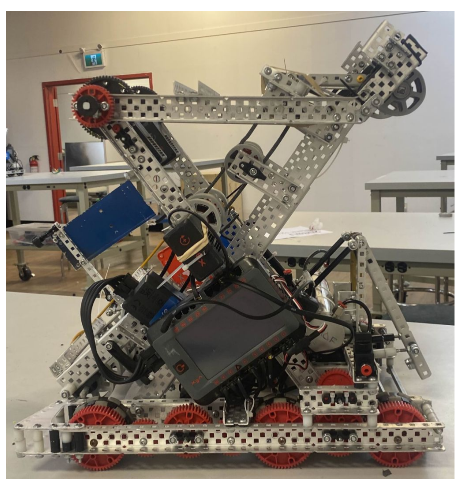
▶ **Problem:** How can we remake the wall stake mechanism based on the scrimmage performance?

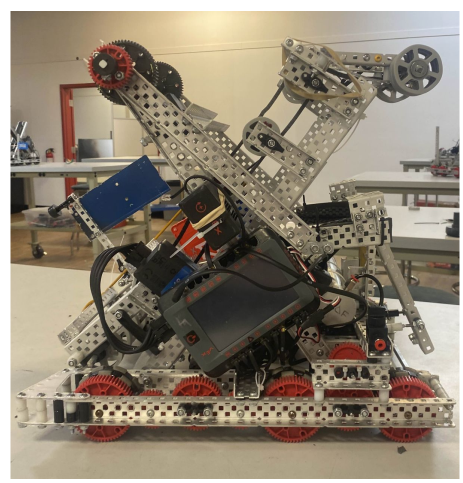
Another change we made was going from a single 11-watt motor to two 5.5-watt motors, which mathematically have slightly more torque.



On top of adjusting the design, we also wanted to change the position and functionality of it. The previous design made it so that the wall stake mechanism stopped at the perfect spot to be loaded, but couldn't go down any farther and had to stay up, preventing us from driving under the climb bar. The new design we made sits below the hood, and is also a straight c-channel, which allows us to drive under the bar now. Our wall stake alliance has three positions: one scoring, one loading, and one down for intaking.







Having the wall stake mechanism sit under the hood allows the robot to be much more compact during matches and much more space efficient, which solves one of the problems with the previous design.

Designed by:

Adam X

Witnessed by:

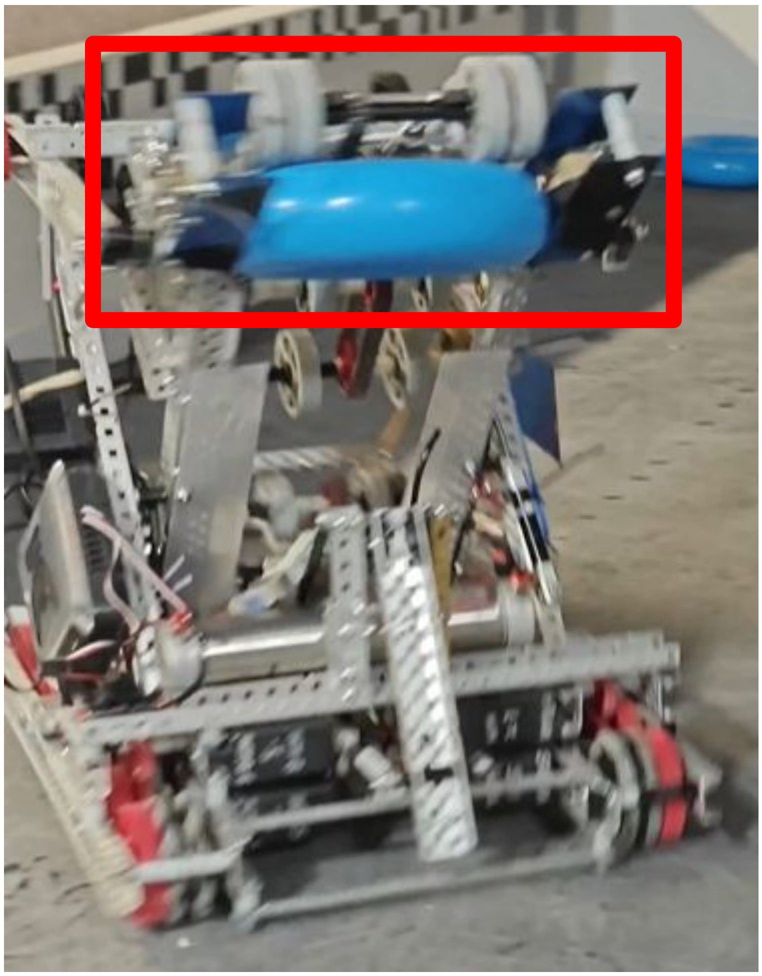
Alex S

Wall Stake Mechanism Redesign

2024-10-29

▶ **Problem:** How can we remake the wall stake mechanism based on the scrimmage performance?

Adding onto our progress yesterday, we found that the angle-bar design was not the best way to hold the rings in place, since there was no constrain at the top. We saw a lot of teams with the same design for our wall stake mechanism use lexan pieces to hold the rings in place, which seemed to work well for them. Lexan gives a firmer hold onto the rings, so the mechanism is not only reliant on the friction of the foam tape, but also constrained by the lexan pieces on the top and bottom.



We found that the lexan attachment worked much better than the non-slip netting wrapped over angle bars, and that it had a much tighter hold.

Designed by:

Adam X

Witnessed by:

Alex S

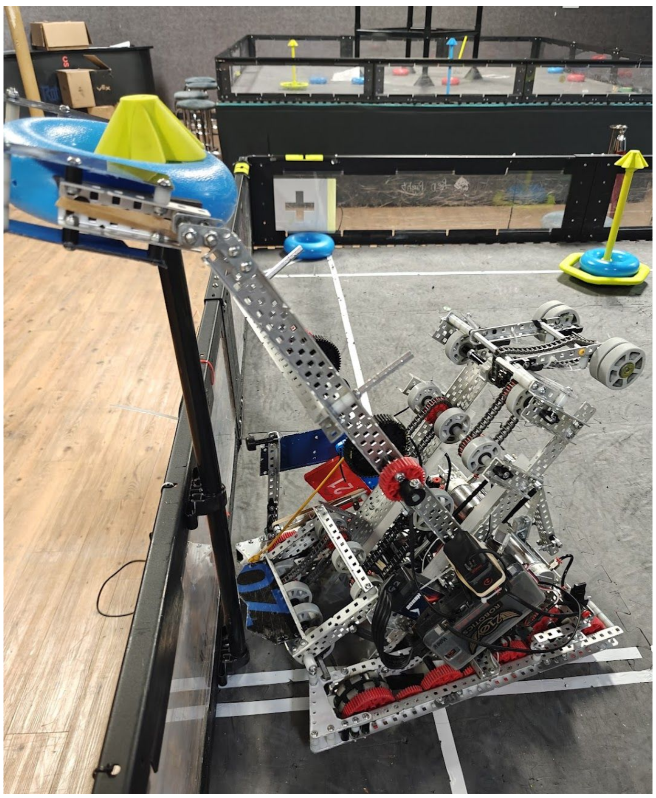
Wall Stake Mechanism Key

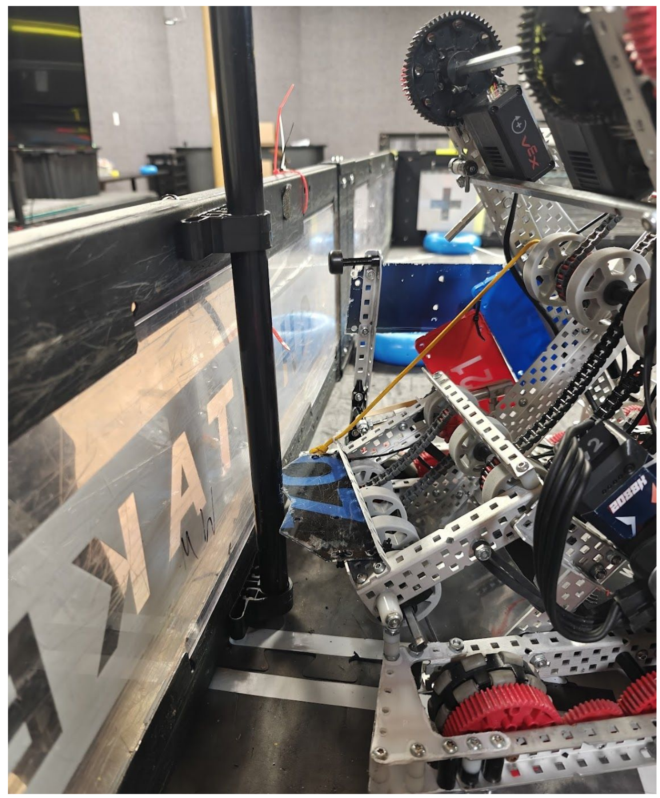
2024-10-30

▶ **Problem:** How can we remake the wall stake mechanism based on the scrimmage performance?

[Identify Problem]

Today, we decided to tackled the problem of driving and practicing with the wall stake mechanism. One thing we noticed during the scrimmage last weekend was that as the day went on, the intake cross brace kept on bending, because it was the first thing that hit the wall stake when we approached it. As we lined it up more and more, the place where it contacted started bending, and kind of created a key, or a notch for the wall stake to go into. We realized that we could use this to our advantage, and we could create a lexan piece that acted as a key for that wall stake, and had a piece cut out of it for the wall stake to fit in, making aligning the robot much easier in matches.





When testing it out, we found that it was extremely helpful, and made the scoring process much more efficient and accurate

Designed by:

Adam X

Witnessed by:

Maxwell L

Improving the intake

2024-10-30

▶ **Problem:** The intake is slow, and can't intake two rings at a time

[Identify Problem]

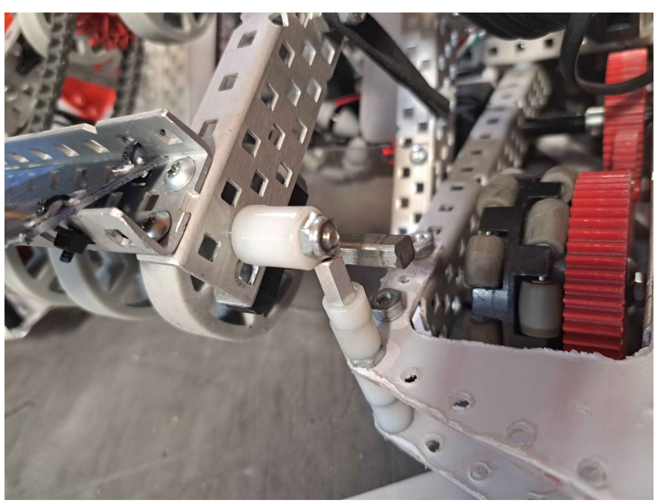
When testing the robot out after the wall stake mechanism changes, it came to our attention that our intake had high amounts of friction, and couldn't hold more than one ring at a time. Whenever we intaked a second ring, then intake did not have enough torque and power to move it up the conveyor, and it would just stop.

There were two main reasons why the intake wasn't working as expected:

1. **The front support bar was bent.**
   The support bar located at the front of the intake was kind of bent, which drastically reduced the friction.
2. **The axles were bent, and it was spaced too much.**
   In addition to the bar, the axles were also bent, which caused them to rotate unevenly, creating unnecessary friction and inconsistency in the intake. They were also spaced too wide, making the attached c-channels on the side wider, bending the entire structure.

Rebuilding and addressing these two problems, we remade the front of the intake, used new axles, and spaced everything correctly, making the intaking process much smoother overall. Now, intaking a ring took much less time than before, and it didn't stop whenever we intake a second.

Another thing we changed was the hard stop. During the scrimmage, we realized that it was too low, and cause the first stage a flex wheels to oscillate once before intaking the ring. We raised it slightly, which resulted in the intake sitting right above the height of one ring, which was perfect.



Designed by: Adam X

Witnessed by: Richard F

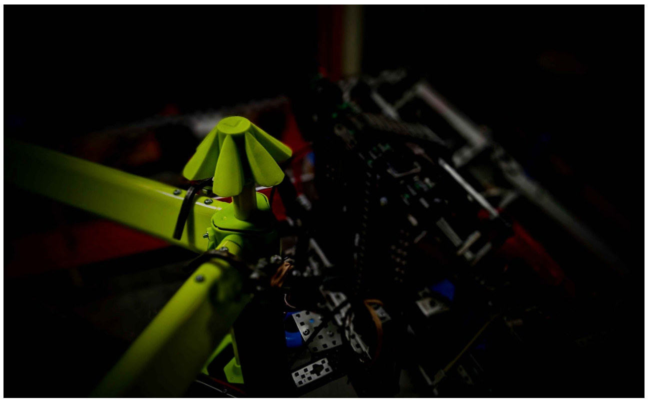
Elevation Brainstorming

2024-10-30

▸ **Problem:** What are the best designs for an elevation mechanism?

[Brainstorm Solutions]

At this point in the season, the only recorded elevations have been at level 1, with no one in a tournament going any higher. There have been teams that have claimed to get a level 3 and have posted pictures, but we haven't actually seen it happen on video yet.



Picture of **7842F's** alleged level 3 elevation

Since our next tournament is in a few weeks, we don't have the time and space on the robot to go all out and make a completely new, revolutionizing design for elevation. That's for the next robot.

For the elevation mechanism, we had two main options on how we wanted to build it:

1. **Attach it onto the wall stake mechanism, which already can already go high enough to hook onto the bar.**
   Since the wall stake mechanism rotates around the robot, it already fits the requirements of a hang mechanism as it can already reach over the bar. We just need to attach some sort of hook so it can hold onto the bard and pull the rest of the robot up.
2. **Build a separate, new mechanism.**
   We could also build a new mechanism, separate of the current mechanisms, which might work better, since we are building a specialised mechanism instead of having one mechanism perform two different functions. However, a problem we might run into is the space. As of right now, we are barely within the 18"x18"x18" size requirements, and building a new mechanism might bring us out of that limit.

Our engineering aspect of this design was made on Inventor Professional and Fusion 360 (project management), since the horizontal components of the design had to be very closely engineered to fit with the other aspects of the robot neatly.

Designed by: Adam X

Witnessed by: Maxwell L

Building the Elevation

2024-10-30

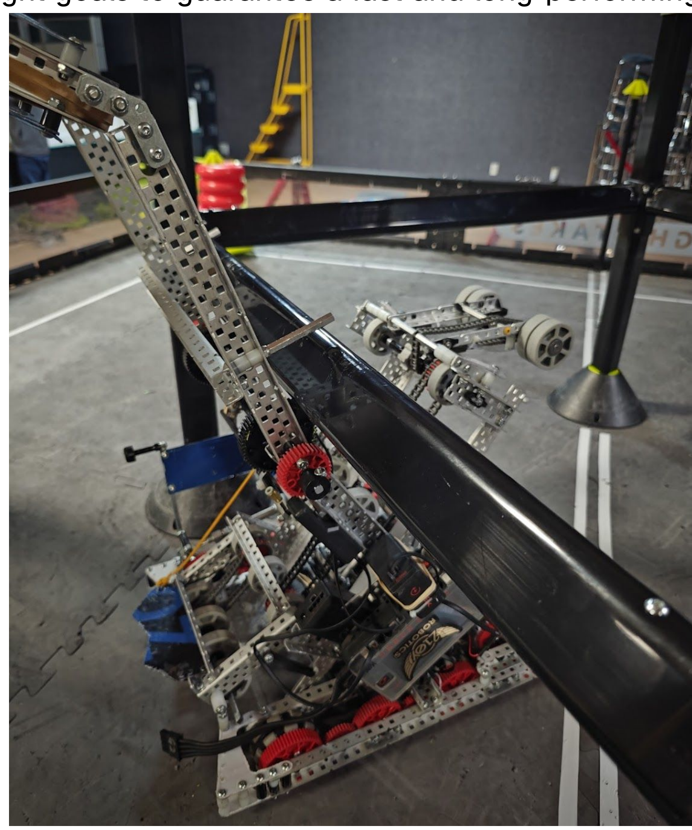
▶ **Problem:** Determine and build the design for the elevation mechanism

[Select a Solution]

Because of our lack of space on the robot and our goal to minimize the weight of the robot, we decided to build our elevation mechanism as an extension from our wall stake mechanism. This solution is also extremely simple compared to designing a completely new mechanism, which saves us time and allows us to allocate more time to programming and driver practice.

[Plan/Design Solution]

Since the wall stake mechanism can already reach above the level 1 bar and also rotates, the easiest and probably most effective solution is to attach something that allows the wall stake mechanism to hook onto the bar and not fall off. This can either be done by adding standoffs or cutting out a lexan piece. We decided that standoffs would be sufficient enough for the mechanism to work, and cutting out lexan could create inconsistencies. This alternative allows us to focus on improving existing components, rather than menially adding on more metal and unnecessary weight to the robot. Not only does this allow our robot to perform numerous tasks with the same component, but ensures that we meet our pre-set weight goals to guarantee a fast and long-performing robot that is lightweight and durable.



Designed by:

Adam X

Witnessed by:

Maxwell L

Building the Ratchet

2024-11-02

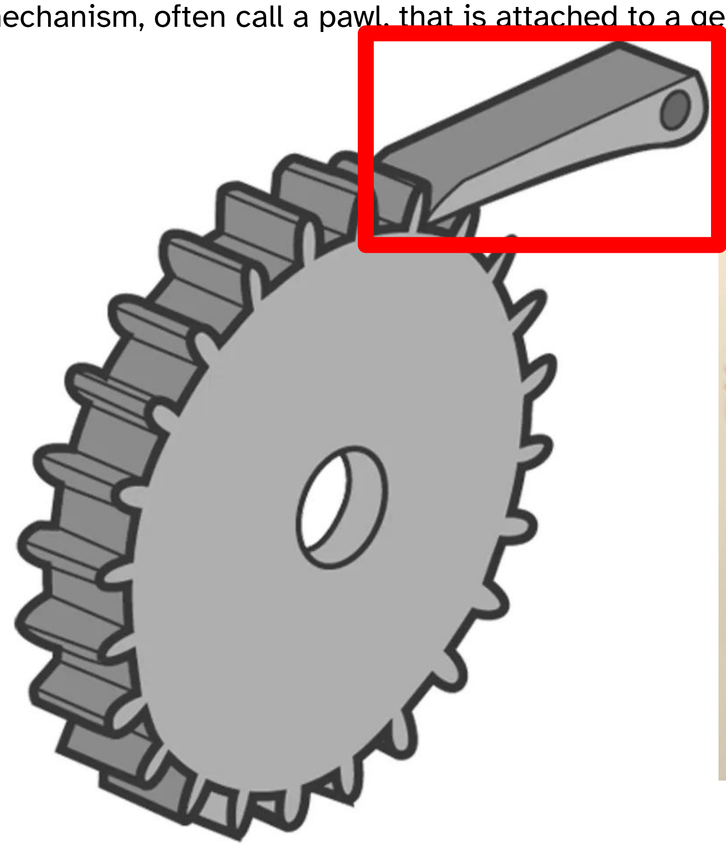
▶ **Problem:** Pure motor power isn't enough to hold the robot in place when elevated

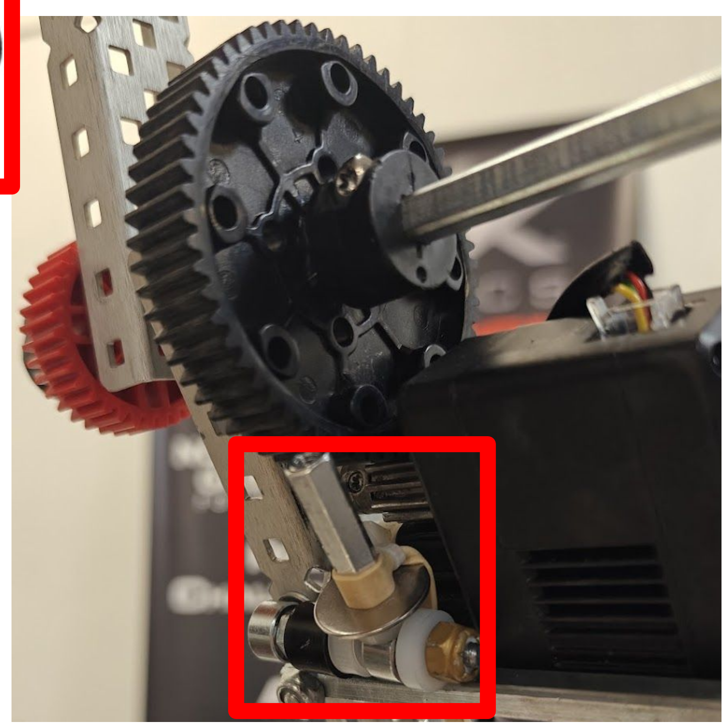
[Identify Problem]

Now that we had the elevation mechanism built, we tested it out on the field with some assistance from our hands, and it seemed to work well, the only problem was that it wouldn't stay elevated in place and it would fall back down. Even when we set the motors to hold—which means that when not moving, the motors in place—the motors alone weren't powerful enough to hold the robot up, so we need to build something that can.

[Brainstorm Solutions]

One thing that we've implementing in the past for these problems is a ratchet, which is a mechanism, often call a pawl, that is attached to a gear, and allows it to spin one way, but not the other.





When the gear spins counterclockwise, the pawl will rotate up, letting each tooth pass. However, if the gear spins clockwise, then the pawl will prevent the gear from spinning because the teeth get blocked by it.

Designed by:

Adam X

Witnessed by:

Maxwell L

Building the Ratchet Deployment

2024-10-02

▶ **Problem:** The ratchet can't be on during the match because we still need to score wall stakes

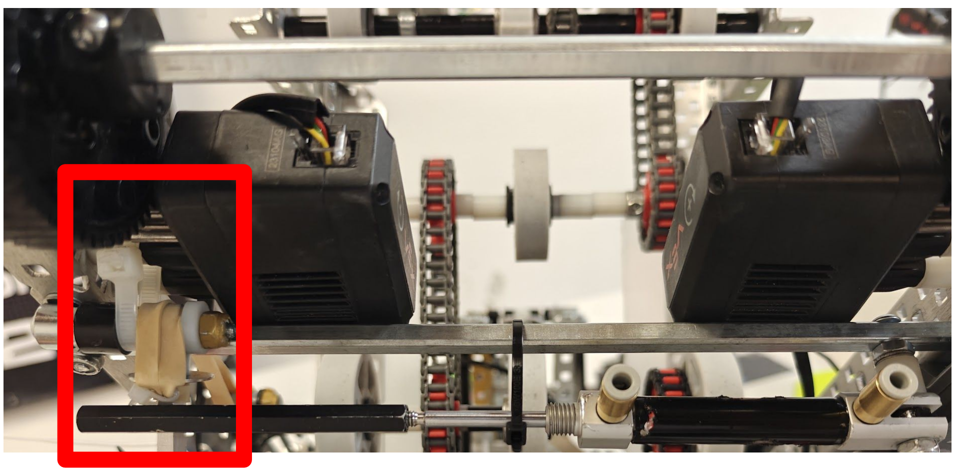
[Identify Problem]

During the driver control period, we still need to be able to rotate the wall stake mechanism both ways in order to score on the wall stakes. To do this, we need to have some sort of mechanism to deploy the ratch only when we go to elevate during endgame.

[Brainstorm Solutions]

The best and probably only way we thought of was to use a piston, which would hold the ratchet off of the gear for the majority of the match until it would then extend or retract, depending on how it's designed, and set the ratchet in place for elevation.

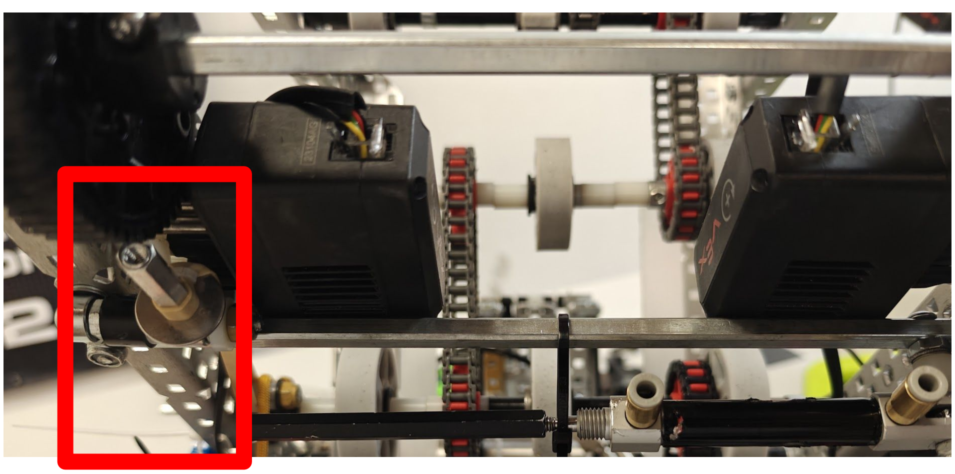
This is the design we came up with:



Ratchet off

When the piston is extended, it holds off the ratchet, and when it retracts, the ratchet activates

This mechanism is a one-time use activation (at the end of the match, since we would no longer have any function to run the arm with afterwards)



Ratchet on

Designed by:

Adam X

Witnessed by:

Richard F

Changing Elevation Gear Ratio

2024-11-02

▸ **Problem:** The two 5.5 W motors are too weak to pull up the robot

[Identify Problem]

At the end of the match, the wall stake scoring mechanism is strong enough to lift one side of the robot off the ground, while the ratchet deployment mechanism keeps it in place afterwards. However, they are simply too weak at the current gear ratio to lift the entire robot up off the ground, so the gear ratio would have to be changed in order to be able to climb efficiently and consistently every time.

[Brainstorm Solutions]

The issue with gear ratios are that they are ALWAYS a tradeoff between speed and torque (strength). This means that while we would be able to climb easier with more strength on this mechanism on a larger gear ratio, the entire function would lose speed altogether and therefore score wall stakes slower. Due to this, we had to find the perfect gear ratio with the best balance between speed and torque to ensure both a fast wall stake scoring mechanism and strong elevation (Tier 1 hang).

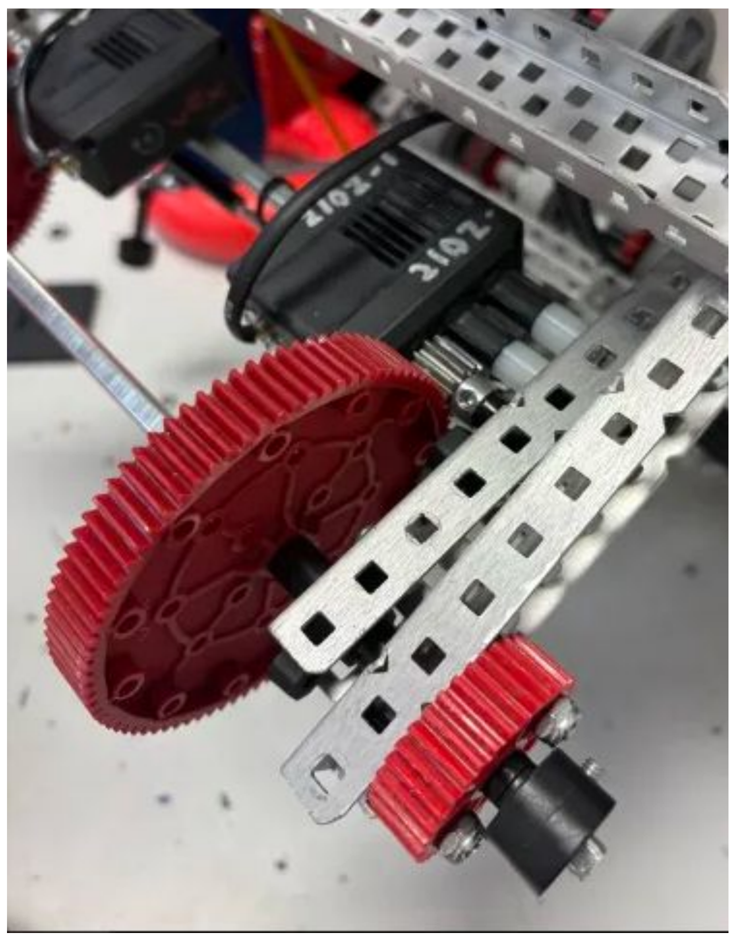
This is the new ratio we came up with:

(The 5.5 Watt Motors run at 200 rpm)

Our previous ratio was 1:5, meaning the final Rotating mechanism would spin at **40 rpm**.

Our newly balanced gear ratio (fitting for both speed and torque as a dual-function mechanism for wall-stakes and elevation) has a ratio of 1:7, meaning the final rotating mechanism would Spin at **29 rpm**.

A dual function mechanism allows us to not only save weight, but also space on the robot. These benefits are also realized with a more resource-efficient construction of mechanism.



Designed by:

Maxwell L

Witnessed by:

Richard F

Rebuilding the Arm

2024-11-02

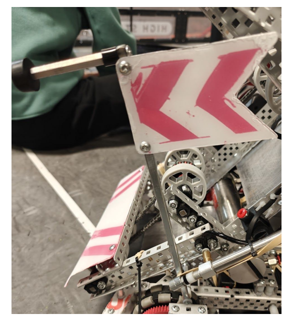
▶ **Problem:** The current one is bent from the previous scrimmage

[Identify Problem]

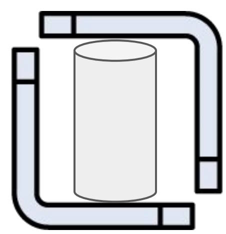
After the scrimmage last weekend, our arm was extremely bent from the blunt force it received, as well as from driver mistakes when we forgot to put it back up.

[Brainstorm Solutions]

We that we were going to rebuild the arm, but in order to prevent it from bending in the future, there were a few different ways that we could've implemented.



1. **High Strength Axle**
   Our first idea was to use a high strength axle instead of an angle bar, since high strength axles are very dense and much stronger than an angle bar. You can basically band an angle bar with your hands, but it's hard to bend a high strength axle even with a vice.
2. **Boxed Angle Bar**
   Our next idea, was to also to use an angle bar, but implement a technique called boxing that incorporates two angle bars to form a square-like shape with spacing in between to make sure it doesn't bend.



[Select a Solution]

Based on its simplicity, we chose the high strength axle, just because it requires so much less effort to build, since all we have to do is drill out the holes that it needs to be mounted to, meanwhile if we were to choose to use a boxed angle bar, we would have to go through the tedious process of boxing it which didn't seem worth it for such a small mechanism. Additionally, the high strength axle saves more space than any other solution, which we thought was a priority, even though it was at the cost of extra weight.

Designed by:

Adam X

Witnessed by:

Maxwell Li

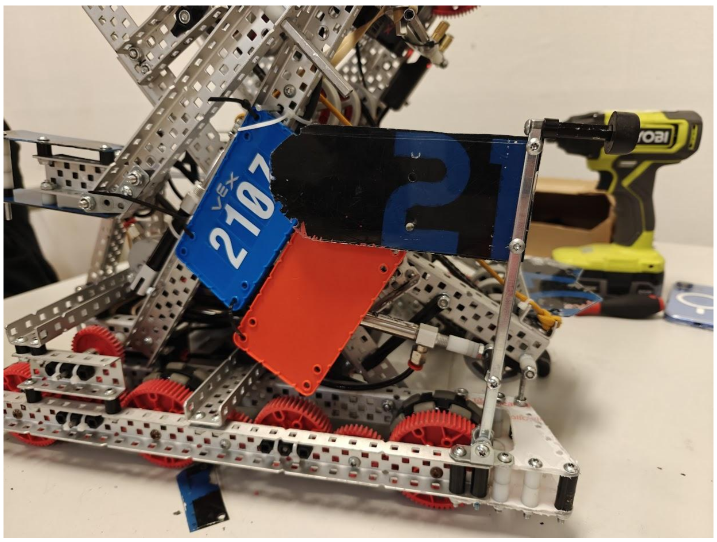
Rebuilding the Arm

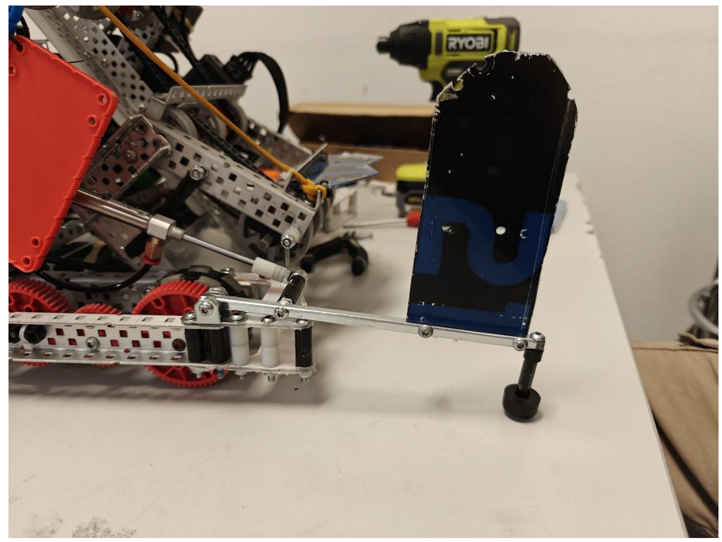
2024-11-02

▶ **Problem:** The current one is bent from the previous scrimmage

[Plan/Design Solution]

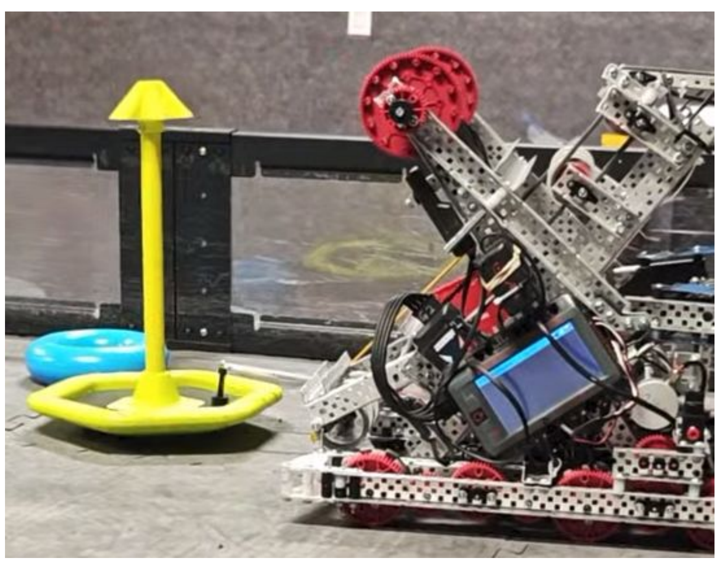
After marking and drilling out the necessary holes, the mechanism was put back together and ready for testing





[Test Solution]

One thing we noticed was that since it was now a high strength axle which is thinner than an angle bar, it actually has more range of motion and can go down farther than as an angle bar, able to reach the ground. This allows it to exert a stronger force on the modible goals when the arm clamps down on one.



Designed by:

Adam X

Witnessed by:

Maxwell Li

Improving the Elevation

2024-11-04

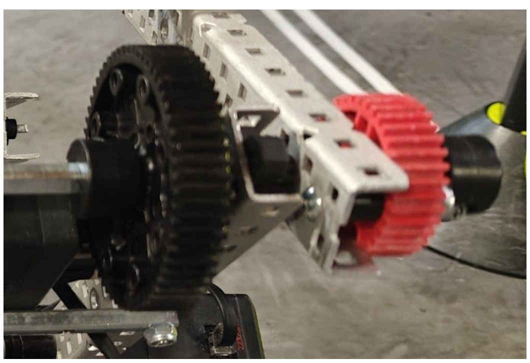
▶ **Problem:** Test and improve the elevation mechanism

[Test Solution]

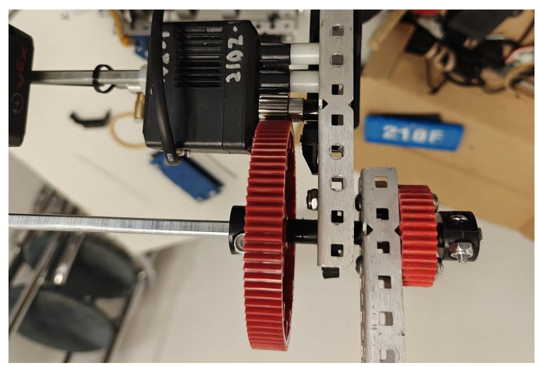
After some quick testing, we learned that the wall stake mechanism couldn't fully pull the robot of the ground, and that it needed more power to do so. Because of this, we decided to increase the gear ratio of the mechanism, trying to give it more torque.

[Plan/Design Solution]

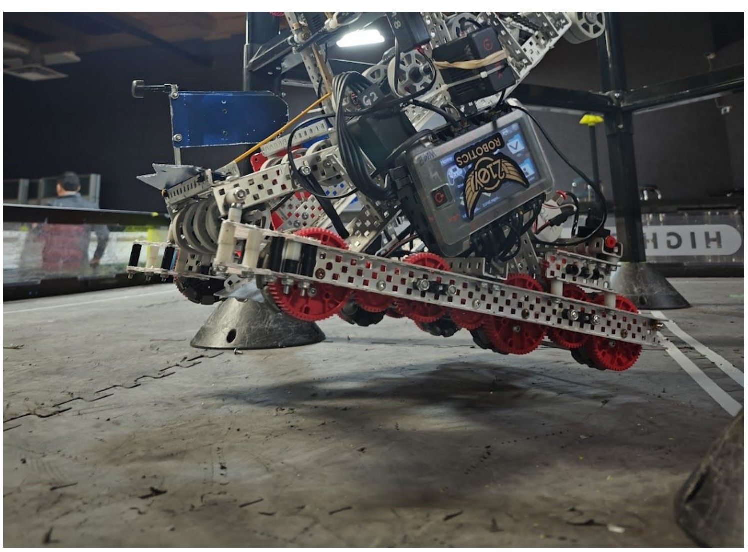
The gear ratio 5:1, powered by a 200 rpm green cartridge, so the mechanism was running at 40 rpm. We decided to change it to 7:1, decreasing the speed and thus increasing the torque.
We found that doing this greatly increased the power of the wall stake mechanism, and now it could consistently pull the entire robot off the ground.



5:1 Ratio Before



7:1 Ratio After



After we made this change, the election mechanism had some mass improvements in performance.

Designed by:

Adam X

Witnessed by:

Richard F

Tuning the Hood/Intake

2024-11-08

▶ **Problem:** The intake is very inconsistent and the hood blocks the wall stake mechanism

[Identify Problem]

When rotating up, the wall stake mechanism hits the out flex wheels on the hood, preventing it from being it in the right position. It's coded to go to a certain position in order to be able to take a ring and score it on a wall stake, and that position is very close to the point where it hits the hood, so it just stops and that point since it is so close. Because of this, the robot is unable to intake rings in to the mechanism, and it just doesn't function as intended.

[Brainstorm Solutions]

There were two ways we could've solved this:

1. **Widen the wall stake mechanism**
   This would loosen the friction grip on the rings, which would make the mechanism worse and potentially not work, se we decided not to do this.
2. **Make the hood thinner**
   Making the hood thinner would require us to use a new axle, since we hammered in the current one so we didn't need to use shaft collars. It would make the hood lighter, causing less downforce on the rings when scoring, but that can be accounted for by add more rubber bands pulling the hood down.

[Select a Solution]

For the reasons mentioned above, we decided to go with making the hood thinner

[Plan/Design Solution]

Since the hood now had two less flex wheels, it had less mass, which led to a smaller force of gravity pulling it down to push rings past the stake caps and score. To combat this, we added a second rubber band pulling down on the hood to make of for the lost force. Another thing we changed that since there was two flex wheels gone from each side, it was significantly thinner and now we could actually use axle collars to fix the components instead of hammering the ends.

Designed by: Adam X

Witnessed by: Kevin Z

Replacing Lexan with Custom Cuts

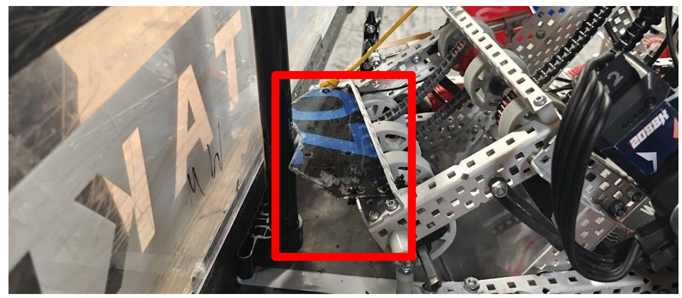
2024-11-10

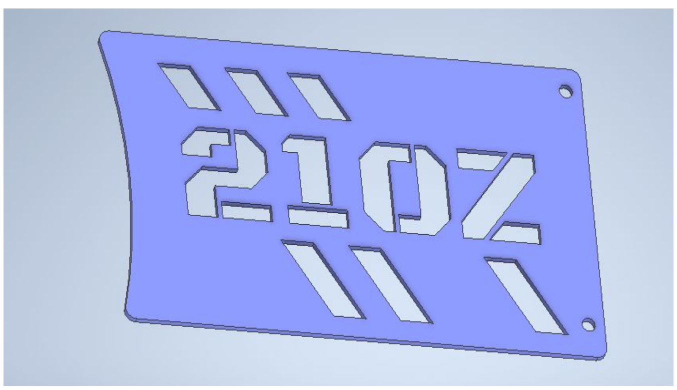
▶ **Problem:** Our current lexan pieces are old scraps from past robots and don't look good

[Plan/Design Solution]

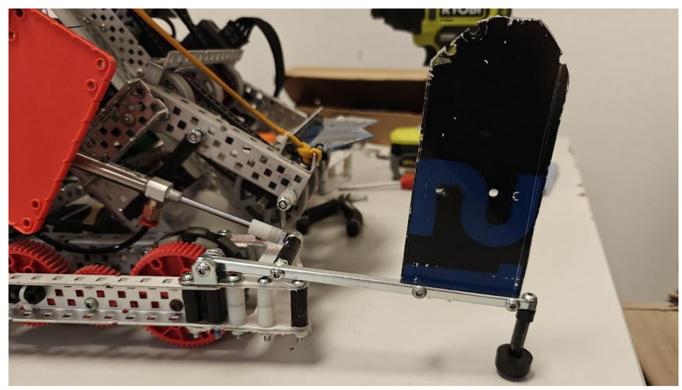
Since our club has a Computer numerical control (CNC), we are able to precisely cut out customly designed lexan pieces to put on our robot. Using a Computer-aided design (CAD software, we reverse engineered all of the big lexan pieces currently on the robot, added some team designs to them, and used the CNC machine to cut them all out to their exact dimensions.

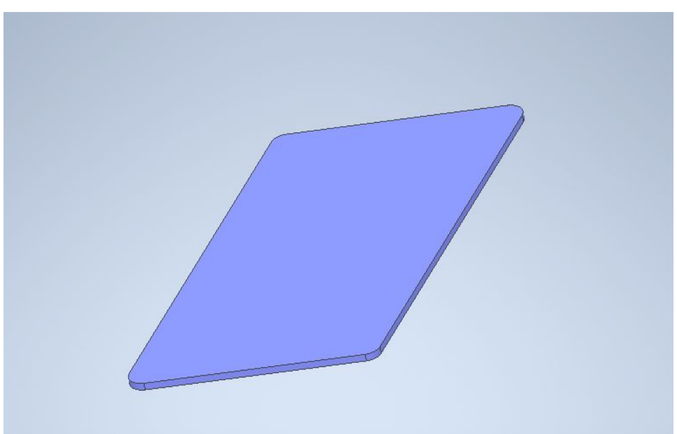
Custom lexan to replace wall stake key





Custom lexan to replace arm lexan




Custom lexan to replace hood sleds (second version)

Designed by:

Adam X

Witnessed by:

Maxwell L

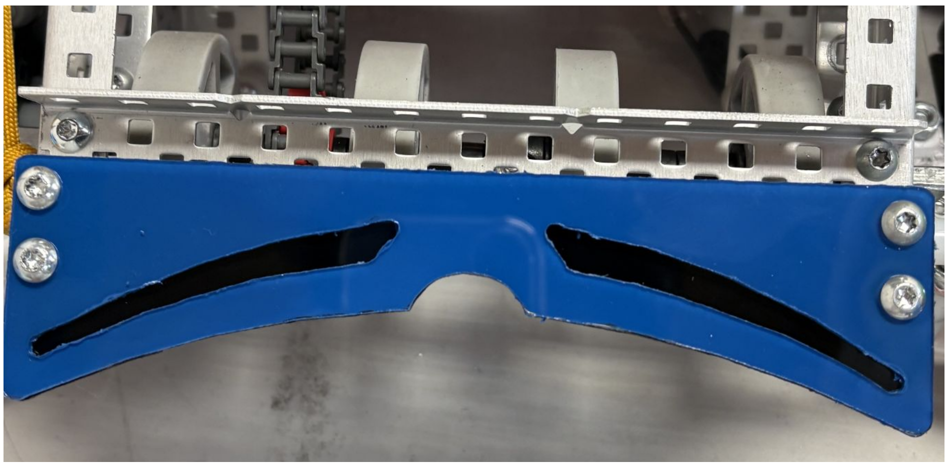
Replacing Lexan with Custom Cuts

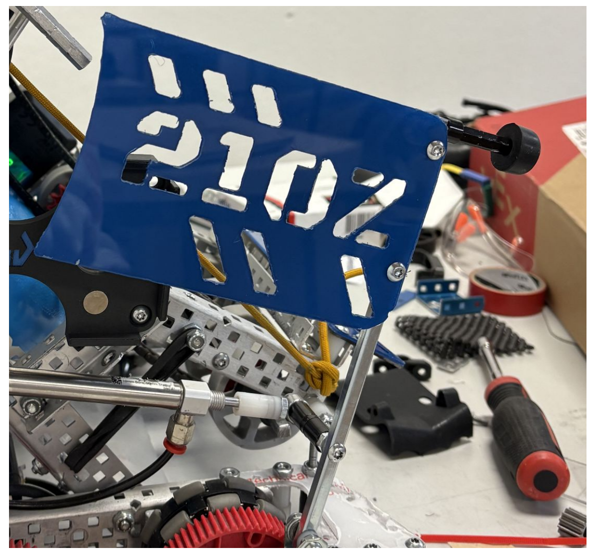
2024-11-10

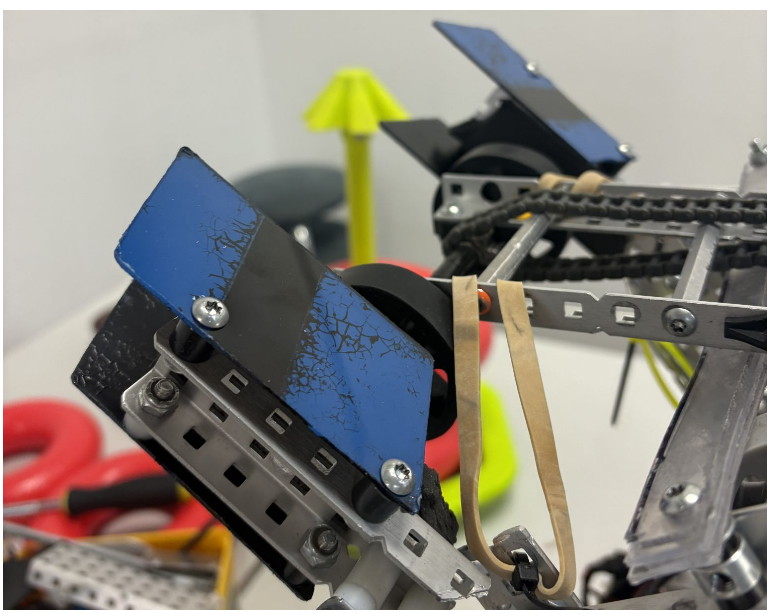
▶ **Problem:** Our current lexan pieces are old scraps from past robots and don't look good

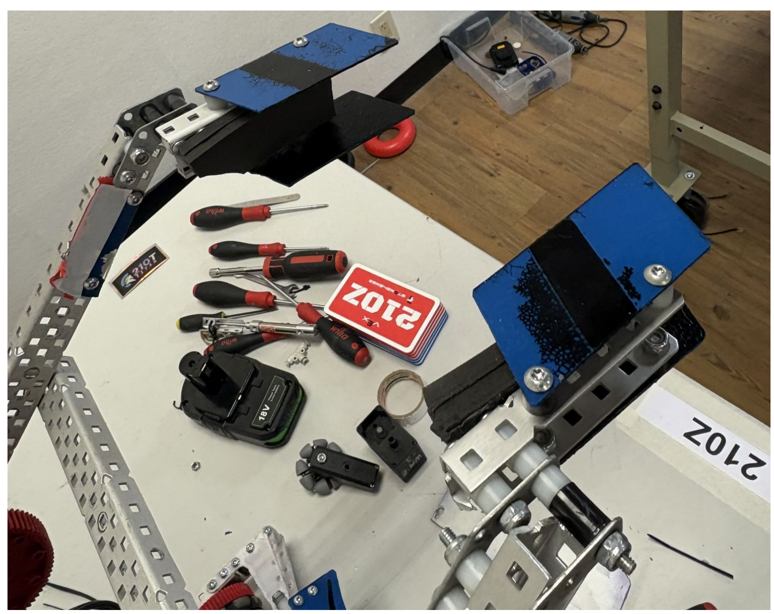
[Test Solution]

After cutting out the pieces, we spray painted them with our club colors, blue and black, which turned out very well. These pieces are not only improved with precision functionality, but are also aesthetically pleasing. CNC-cut polycarbonate and delrin plastics give us the opportunity to expand our experience in terms of functionality of each piece and aspect of the robot.









Designed by:

Adam X

Witnessed by:

Maxwell L

Tuning the Hood/Intake

2024-11-09

▶ **Problem:** Intake is having some issues

[Identify Problem]

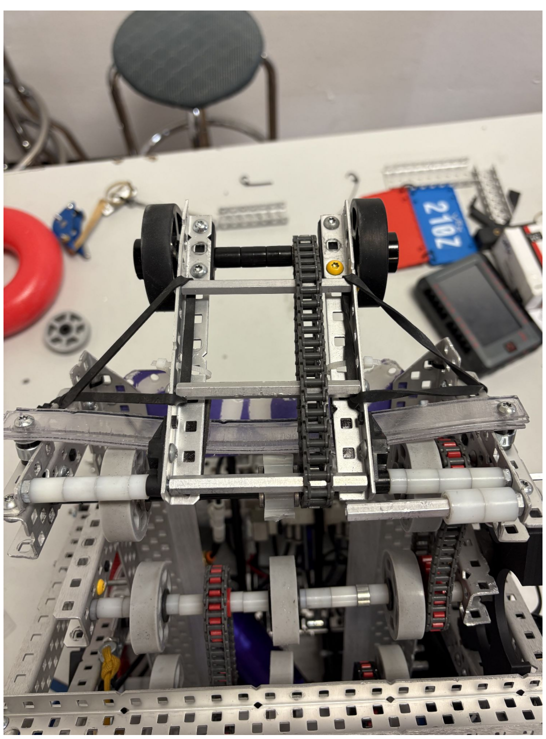
With all the changes we made yesterday, there were still some inconsistencies with the intake. Near the end of the intake, when the ring goes through the hood and onto the stake, it would sometimes get stuck on the cap, even though the hood had room to push it down. This was because the flex wheels were compressed from the rings, and had a smaller diameter that normal, so it wasn't big enough and couldn't get enough friction to push rings through the stake cap.

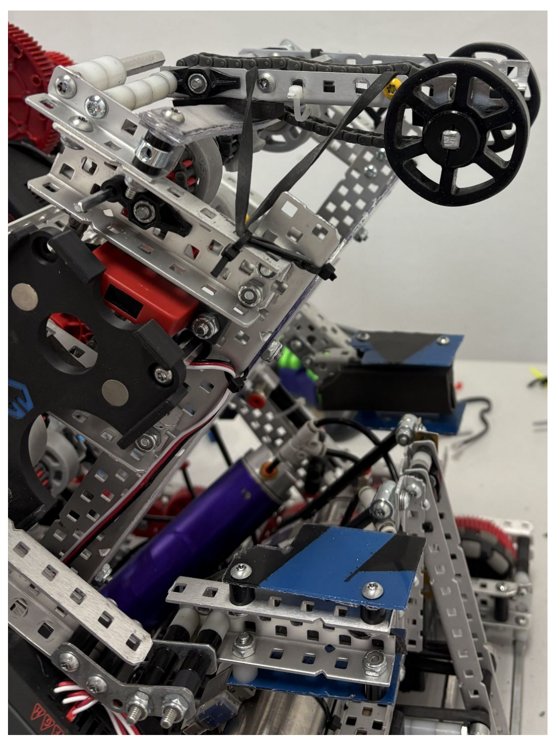
[Select a Solution]

As soon as we identified the problem, we knew that the best and most simple solution was to swap out the current flex wheels for black ones, which were stiffer. We were using the white/grey 30A flex wheels, which are very compressible and compliant, which was why they were causing us problems. The 60A black flex wheels are much stiffer and rigid, suiting our solution to the problem

[Plan/Design Solution]

Swapping out the flex wheels were quite simple and didn't take that much time, and after we swapped, the flex wheels were able to maintain their shape and successfully push down rings through the stake cap.





Designed by:

Adam X

Witnessed by:

Kevin Z

Improving the Wall Stake Mechanism

2024-11-11

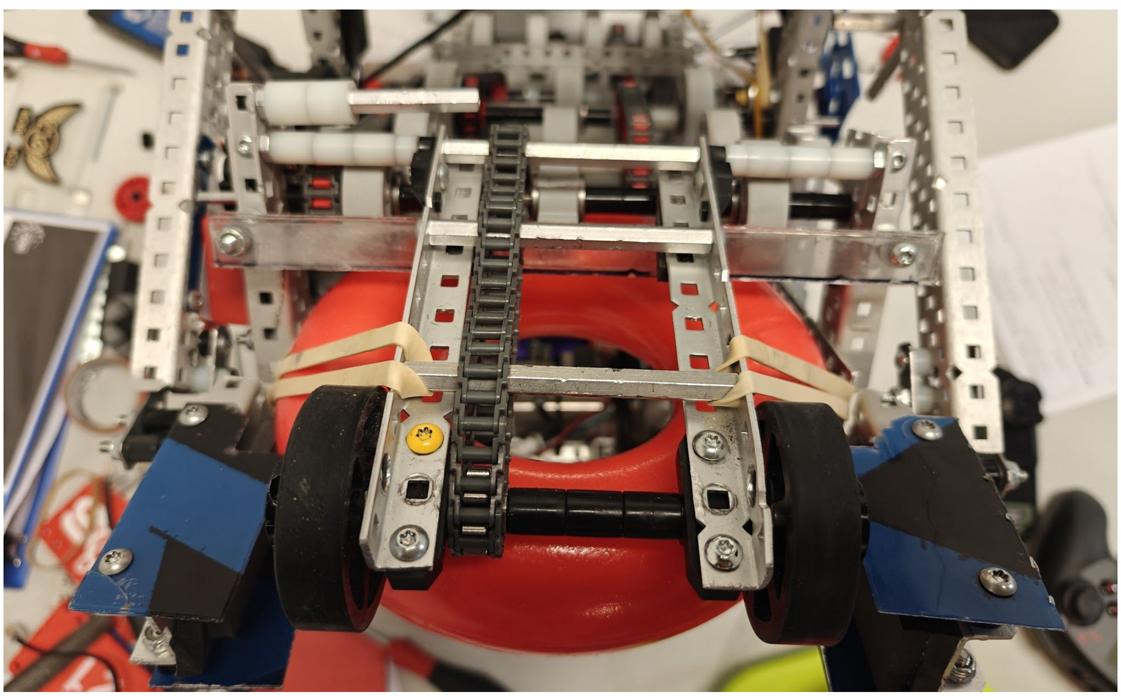
▶ **Problem:** Intaking into the wall stake mechanism is slow

[Identify Problem]

When practicing wall stakes, during driver practice, Maxwell had to constantly hold the intake down for a longer period of time, just so the ring could slowly slide into place from the pressure of the hood. This isn't what we wanted, as it wasted time and made us les efficient, and in that time, a team could defend us from scoring.

We considered two reasons why this problem was happening:

1. **There is not enough torque and pressure from the intake to push the ring**
   This problem was arising due to the high amounts of friction on the intake subsystem. Since a single 11-watt motor is powering 6 stages of flex-wheel rollers, there are many factors that can easily add up friction to create high amounts.
2. **There isn't a wide enough space for the ring to adjust**
   Going from the intake/hood into the wall stake mechanism, the ring has to slightly change angle because the two subsystems are not parallel with each other. Another factor that contributes two this is distance between the lexan constraints above and below where the ring sits. They are quite close, not allowing for much change in the ring's position.



Current intake doesn't push the ring far back enough

Designed by: Adam X

Witnessed by: Maxwell L

Improving the Wall Stake Mechanism

2024-11-11

▶ **Problem:** Intaking into the wall stake mechanism is slow

[Select a Solution]

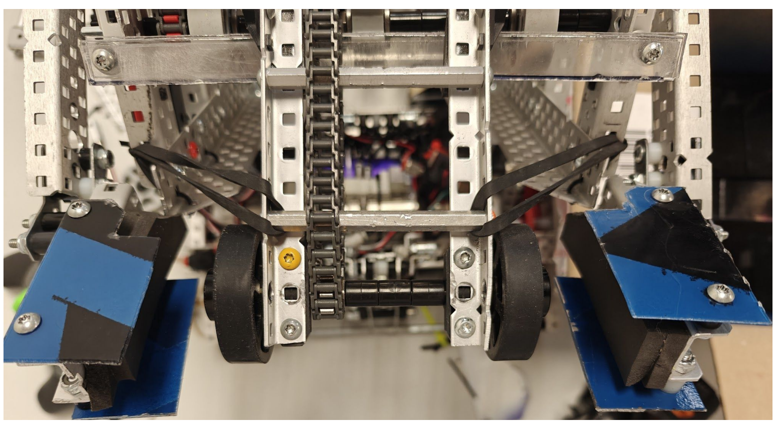
As a team, we concluded that these two issues are major problems that need to be addressed, and the solutions to both were quite straightforward, as there is only really one way they can be fixed.

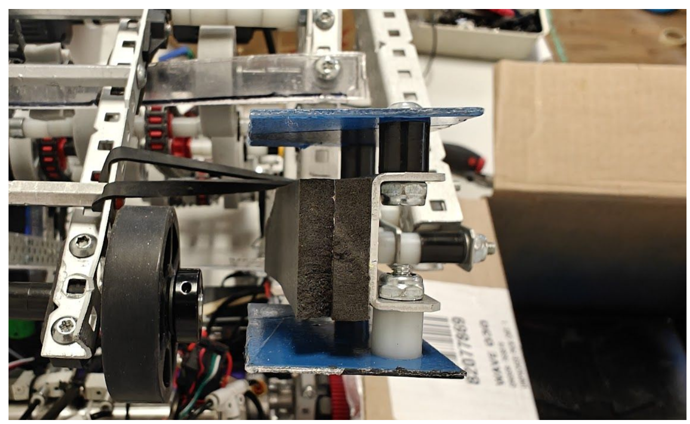
[Plan/Design Solution]

As mentioned above, the changes that needed to be made were very simple. For problem 1, the solution is to check the spacing of each axle, make sure all the c-channels are straight and parallel to each other, and replace bent axles. The solution to problem 2 is to increase the distance between the lexan pieces that are designed to hold the rings, allowing for more range of motion from the ring.

[Test Solution]

After implementing these solutions, we found that the time needed to intake a ring into the wall stake mechanism had drastically reduced, achieving our intended result, Now, we could score on wall stakes so much faster than before, which can potentially be the deciding factor in a match or in the skills challenge.





Increased the gap between lexah by a ⅛" spacer. Intake looks the same as before.

Designed by:

Adam X

Witnessed by:

Maxell L

Alliance Stake Mechanism

2024-11-11

▸ **Problem:** We can't score on alliance stakes.

[Identify Problem]

Currently, we don't have an alliance stake mechanism, which is a huge problem if we want a high score in skills, and if we want to win the autonomous period. If we were to score on the alliance stake the same way as the neutral stakes, then we would extend out of the 6" expansion limit.

[Brainstorm Solutions]

To solve this, we thought of two main solutions. Since we didn't have the resources to build a separate mechanism, we developed a couple of ways to integrate it using our current wall stake mechanism.

1. **One time use with wall stake mechanism**
   This solution would include placing a ring onto the cross-brace of the wall stake mechanism, so that when we rotate it at the start of the match, it would slide off and fall onto the alliance stake.

2. **Score using the wall stake mechanism from behind**
   This solution would work by intaking the ring into the wall stake mechanism as usual, but only lifting it up slightly to be above the alliance stake, then backing up and lowering the ring onto the alliance stake.

Designed by:

Adam X

Witnessed by:

Alex S

Alliance Stake Mechanism

2024-11-11

▶ **Problem:** We can't score on alliance stakes.

[Select a Solution]

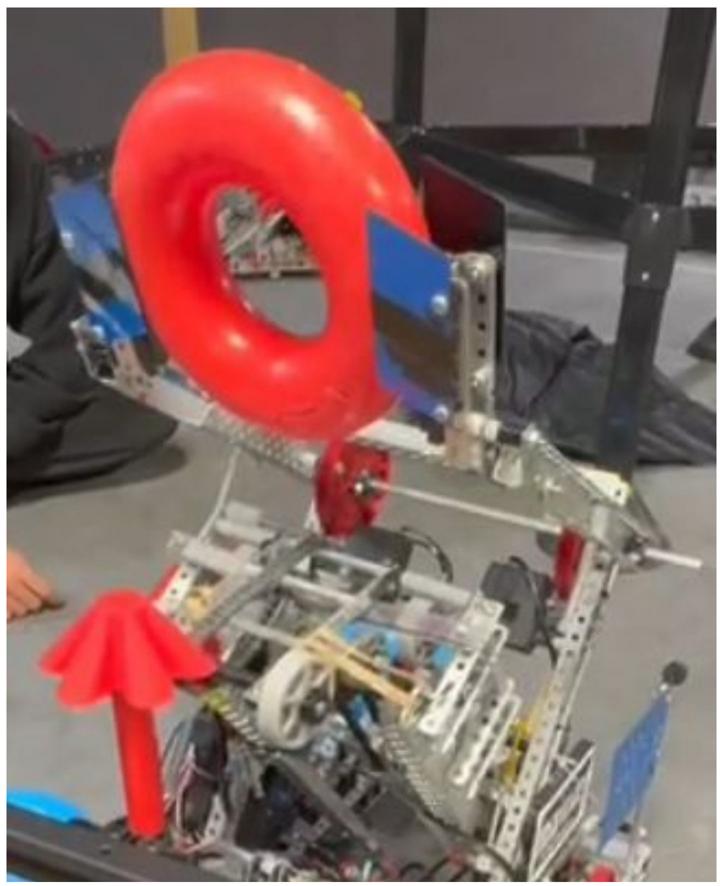
We chose the second solution for two main reasons:

1. **It's repeatable, and not only a one time use.**
   This allows us to score on the other alliance stake in skills, and also to fill up the alliance stakes with multiple rings.
2. **It's less of a expansion limit risk.**
   The first solution is risky in regards to the 6" horizontal expansion limit, as it is difficult to determine whether it is past 6" or not.

[Plan/Design Solution]

This solution doesn't the building of any additional parts and work purely by the wall stake subsystem's functionality.

Once a ring is intaked into the wall stake mechanism, it rotates up and backs up into the alliance stake. Then, the mechanism rotates down and the ring goes onto the stake.



Designed by:

Adam X

Witnessed by:

Alex S

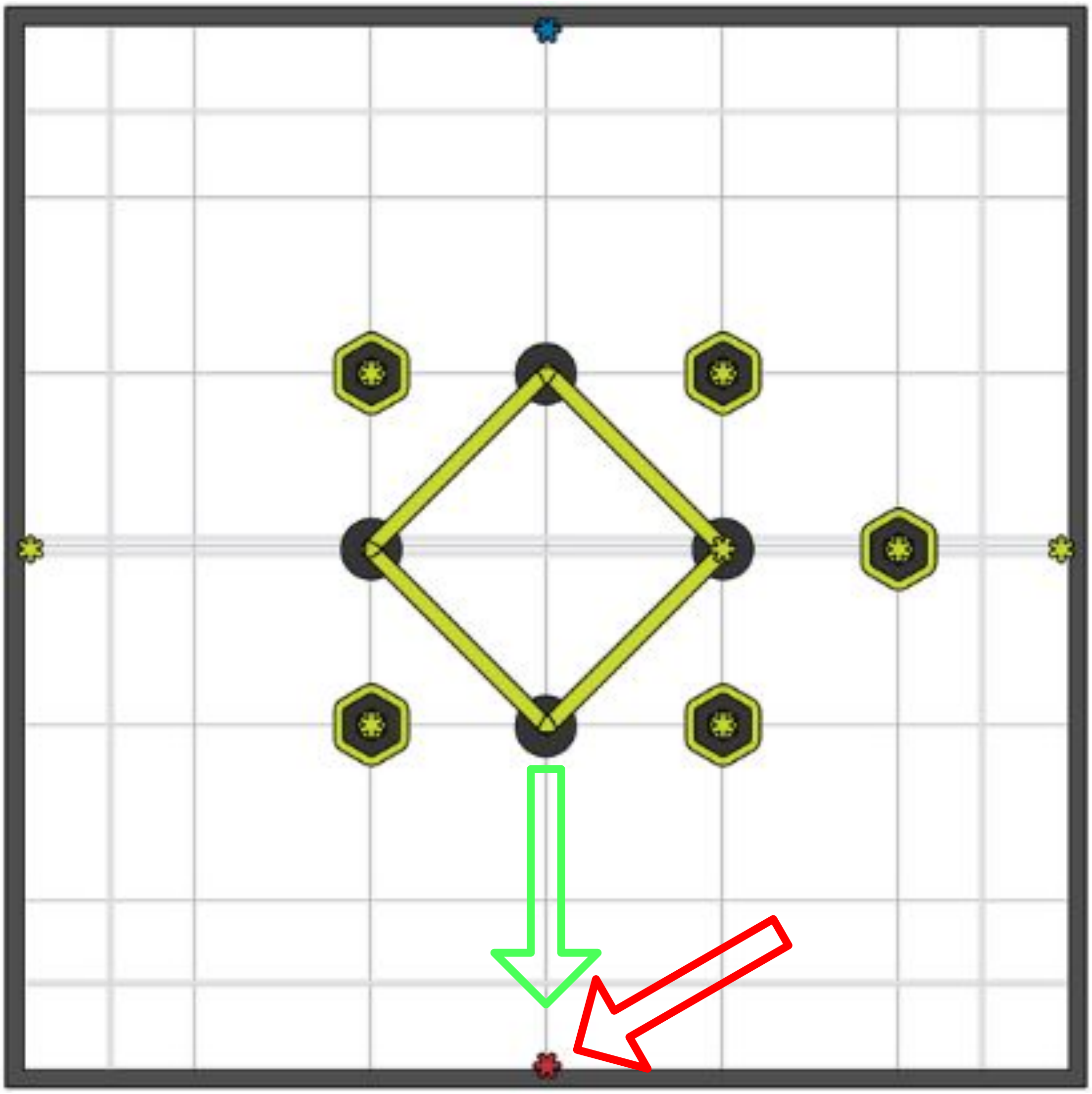
Alliance Stake Mechanism

2024-11-11

▶ **Problem:** We can't score on alliance stakes.

[Test Solution]

When testing, scoring on wall stakes was very consistent, however we discovered one minor problem: we could only score from when we backed straight into the stake. This means that we could not score when we approached the alliance stakes from an angle.



The red arrow shows that when we attempt to score from that angle, it does not end up working, and the green arrow represents the correct way of approaching the alliance stake, which is perpendicular to the field walls.

Designed by:

Adam X

Witnessed by:

Alex S

Tournament Analysis: WM Season Opener

2024-11-16

▶ **Focus:** Analyze our performance at the Western Mechatronics Season Opener Tournament

Western Mechatronics Season Opener

6-0-0 : 13 / 18 / 93

Qualifications Rank:	2nd/12
Elims Alliance Partner	210K
Elims Rank:	Tournament Finalist
Driver Skills Score:	37
Programming Skills Score:	5

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
Q-1	2088X, 3388C	210Z, 3388Z	27 - **41**
Q-9	2088G, 2088S	210Z, 3388E	12 **- 37**
Q-11	210Z, 3388H	2088R, 2088U	**24** - 15
Q-16	210Z, 2088X	2088B, 210Z	**35** - 10
Q-22	210K, 210Z	2088G, 2088S	6 - **12**
Q-26	2088A, 2008B	210Z, 2088R	**30 -** 23
QF 1-1	210K, 210Z	2088S, 3300C	8 - **30**
SF 1-1	210K, 210Z	3388C, 3388K	**34** - 0
F 1-1	210K, 210Z	3388H, 3388S	9 - **31**
F 1-2	210K, 210Z	3388H, 3388S	20 - **24**

Designed by:

Adam X

Witnessed by:

Richard F

Tournament Analysis: WM Season Opener

2024-11-16

▸ **Focus:** Analyze our performance at the Western Mechatronics Season Opener Tournament

Western Mechatronics Season Opener Tournament

One of our main problems and liabilities during the scrimmage on October 26 was our wall stake mechanism, which was extremely hard to score with and took large amounts of time. However, during this tournament, it was probably the most consistent and reliable subsystem on our robot. We went 6-0-0 in qualifications, but two of them we won through disqualifications because the opposing alliance descored our doubled goals within the last 15 seconds. In eliminations, we were selected by our sister team 210K, and we made it to the tournament finals before losing to 3388H and 3388S due to a field issue, where our arm got caught on the field wall and we were unable to move for the duration of the match. We believed this should have been a match replay, but the referees did not.

Next steps for the Nov. 30 Bowness High School would be to refine our intake, as the lexan hard stop as the hood flexes and bends the structure of it. This was a huge issue during the tournament as it gave us trouble while intaking and scoring rings. Another issue with our robot was the clamp. Our fields are very uneven, which means that some parts of the goal or higher off the ground. Our clamp was designed to be perfectly level with the goals, but because of the unevenness of the field,we had some trouble clamping goals in matches

What worked:

- Wall stake mechanism was very fast and worked consistently
- The wall stake key for aligning the robot against the wall stake was also extremely helpful and time saving
- Arm was consistent at clearing corner ring stacks and didn't show any signs of damage
- Drivetrain was consistent and reliable
- Intake lit piston worked well

What didn't work:

- Intake had high amounts of friction, could barely intake one ring at a time
- Clamp was very inconsistent
- Elevation wasn't tested, and the ratchet release wasn't wired.

Designed by: Adam X

Witnessed by: Alex S

Improving the Wall Stake Mechanism

2024-11-19

▸ **Problem:** The wall stake mechanism takes up a lot of space

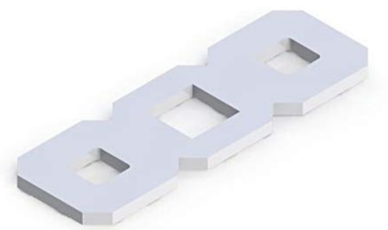
[Identify Problem]

An issue we have with the wall stake mechanism is that it takes up unnecessary space on the outside of the robot, where we can build other things. Specifically in the front, there are gears on the outside that are screwed onto the c-channel arms that allow it to rotate, and without them, they won't move.

[Brainstorm Solutions]

We can change this two different ways:

1. **Lockbar**
   Using a lockbar would allow us to screw something directly onto the c-channel that has an axle-shaped hole, and would spin it along with the gear and axle. This is a much lighter and efficient solution, as it doesn't require very drastic changes



2. **Screwing directly onto the driven gear**
   This solution involves using a long screw to screw the c-channel directly onto the large 84-tooth driven gear. Like the previous solution, this one is light and efficient as it does not require many major changes to the mechanism

[Select a Solution]

We decided to go with solution 2 because we did not have any lockbars currently available, and it didn't seem reasonable to buy a pack of some amount just for this purpose. For that reason, solution 2 was the only solution that we could go with.

[Plan/Design Solution]

This solution was quite simple. All we needed to do was take of and unscrew the outside gears that were current controlling the mechanism's rotation. The actual solution was implemented by screwing a <u>long</u> screw from the red 84-tooth gear across the main bar below, and to the c-channel of the wall stake mechanism.

Designed by:

Adam X

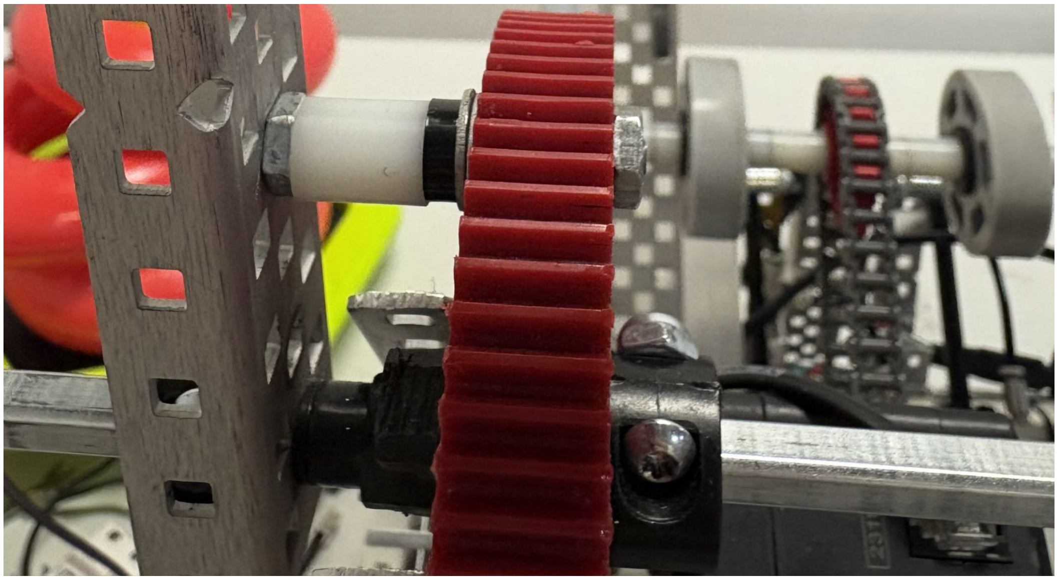
Witnessed by:

Alex S

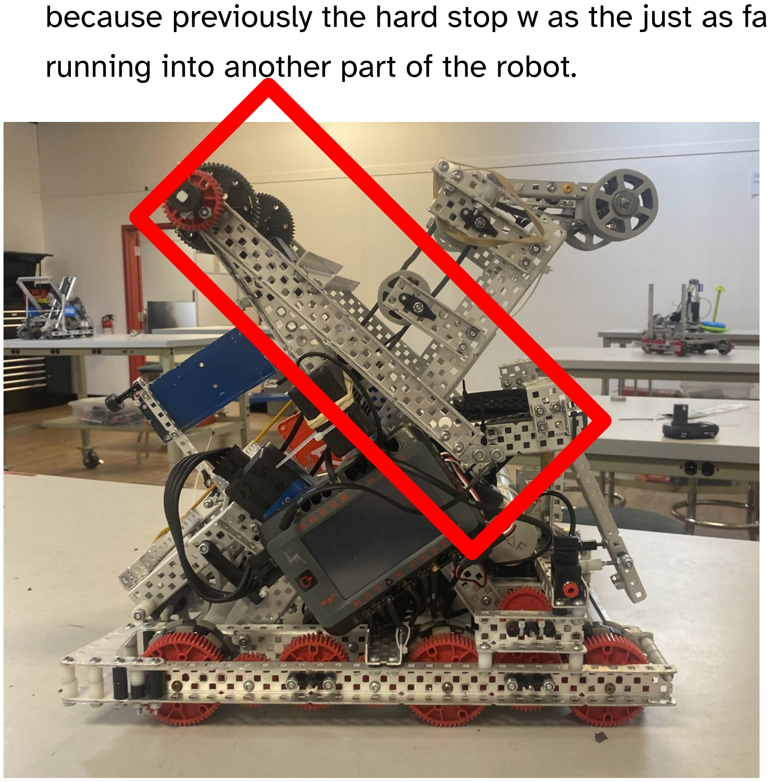
Improving the Wall Stake Mechanism

2024-11-19

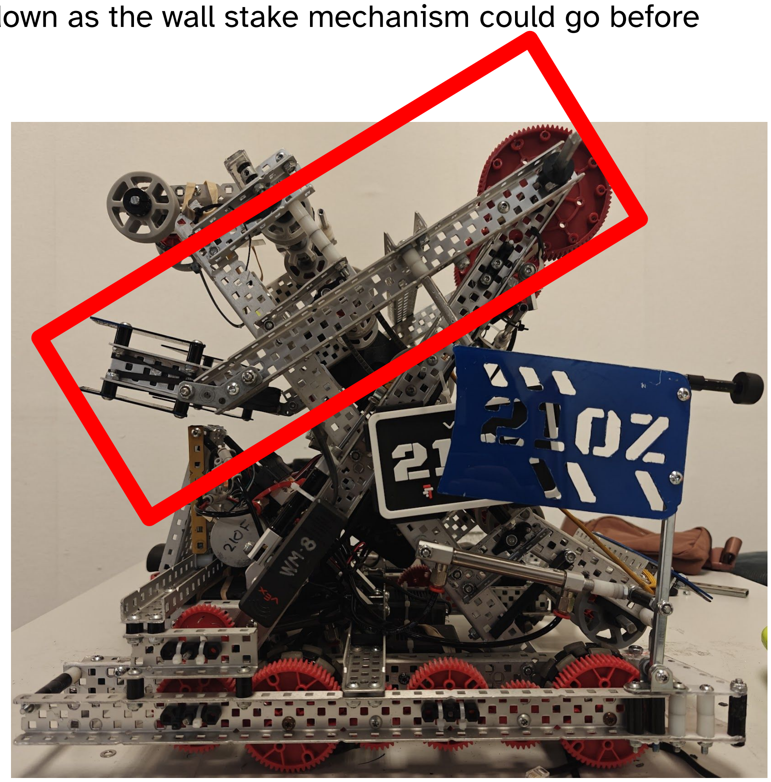
▶ **Problem:** The wall stake mechanism takes up a lot of space



This also acted as a hard stop for the mechanism because it comes into contact with the supporting c-channel. This caused the wall stake mechanism to sit slightly higher up than before, because previously the hard stop w as the just as far down as the wall stake mechanism could go before running into another part of the robot.



Before, lower hardstop



After, higher hardstop

Designed by:

Adam X

Witnessed by:

Alex S

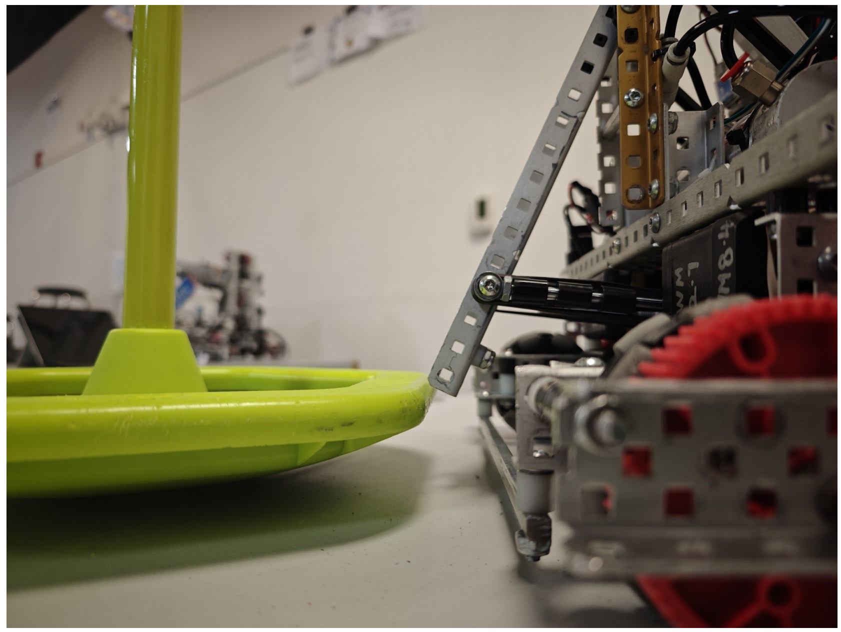
Improving the Clamp

2024-11-19

▶ **Problem:** The clamp is too low compared to the goal

[Identify Problem]

During the tournament, the floor mats were extremely uneven, which caused the goals to be tilted when placed on the mats. Additionally, our clamp was designed to perfectly be just above a flat goal, but because of the uneven mats, it impacted the performance of our clamp heavily.



From the images above, it is clear that with our current clamp design, which was built to be perfectly just above the goal, would not work on the fields that we have.

[Brainstorm Solutions]

1. **Raise the subsystem**
   The mechanism was mounted on a cross-brace spanning across the drivetrain, and the pistons were attached on a different cross-brace on the bottom of the drivetrain. We could space both of these cross-brace bars up slightly to raise the initial position of the clamp
2. **Reduce the angle of the starting position**
   This solution would solve the issue by flattening the angle of the starting position of the clamp. From the image above, if the clamp was not at such a steep angle, the bottom of it would actually be higher.
3. **Make the pistons extend farther**
   Making the pistons extend farther would essentially just do the same thing as flattening the angle of the starting position, it is just a way of implementing it.

Designed by:

Adam X

Witnessed by:

Richard F

Improving the Clamp

2024-11-21

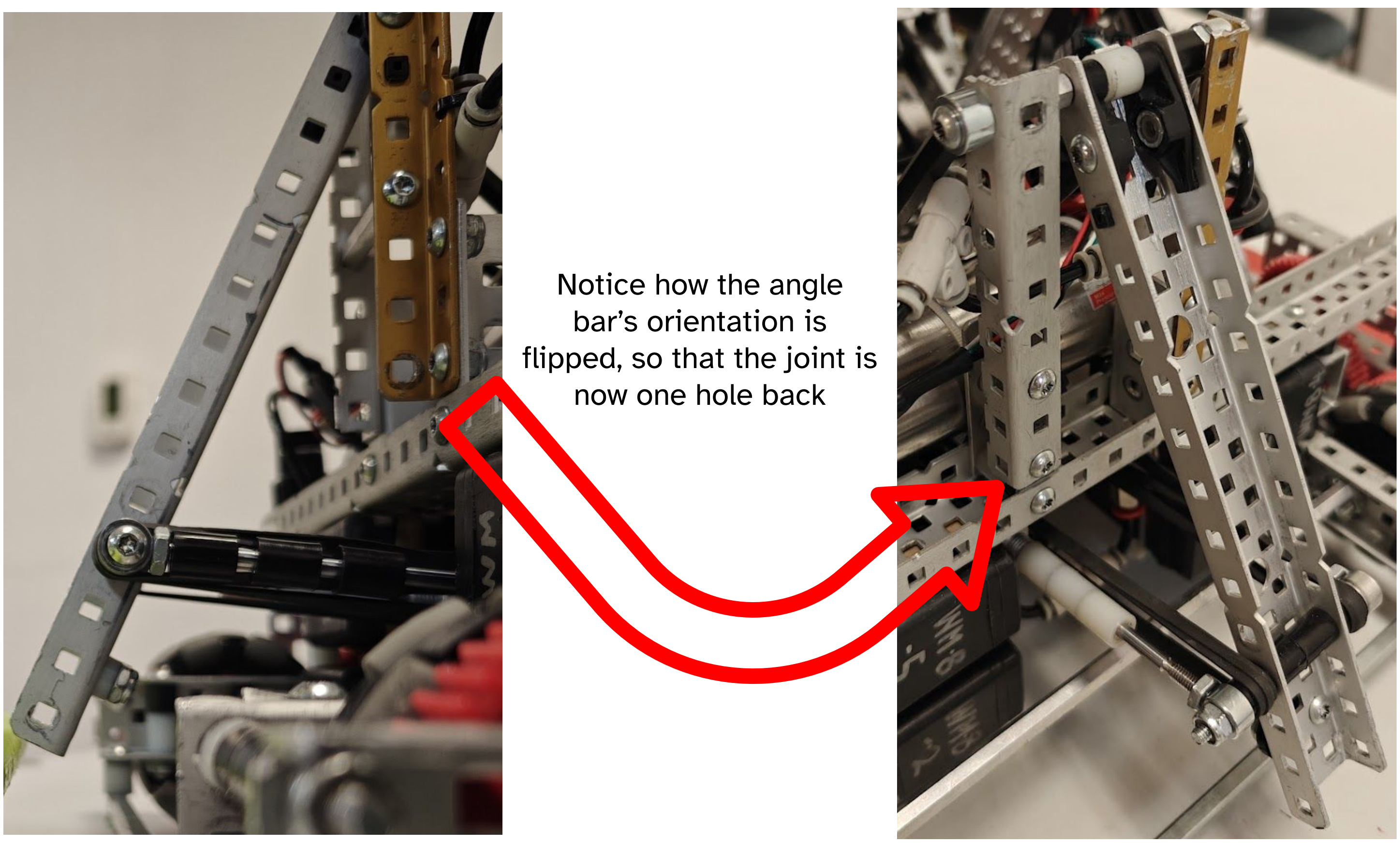
▶ **Problem:** The clamp is too low compared to the goal

[Select a Solution]

We decided to go with solution 2, mainly because the other two solutions require major, time consuming changes that we did not have the time for, with a tournament in a week. Raising the entire mechanism would require us to remove the multiple things attached to it, such as the air reservoir, and that would just require too much time. And making the piston extend longer would be very difficult, because there is no specific way to do this.

[Plan/Design Solution]

We somehow needed to extend the pistons further, to raise its initial position. We decided to implement this by moving the point of rotation, or the fulcrum of the lever back so that the angle of the clamp is flatter than before. We did this by flipping the angle bars that the joint was on, so that they were still in the same spot, but the hole was moved one back.



Designed by:

Adam X

Witnessed by:

Richard F

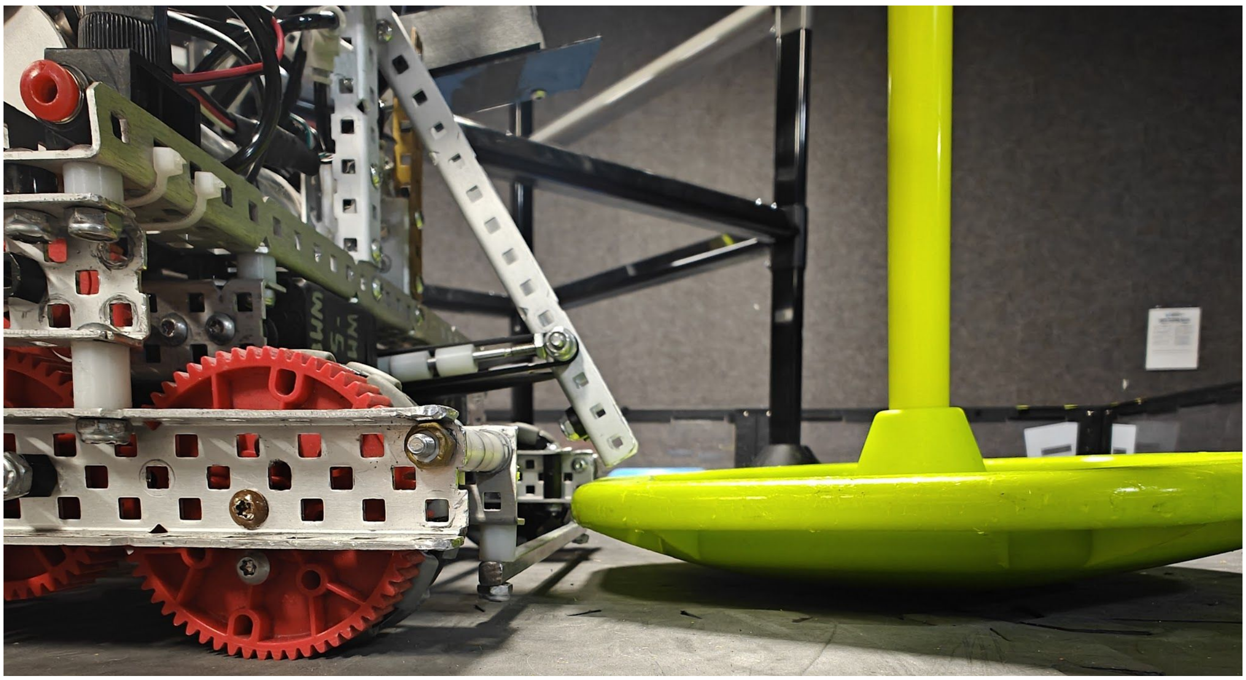
Improving the Clamp

2024-11-21

▶ **Problem:** The clamp is too low compared to the goal

[Test Solution]

Testing our solution, we found that it worked in raising the initial position of the clamp. As seen in the image below, we now have a little bit of room to work with regarding our slanted floors and unbalanced goals.



The clamp doesn't just barely go over the goal anymore, and there is some leeway.

[Identify Problem]

Implementing this solution, we ended up running into a separate problem, where if we backed up into a goal too much and clamped onto it, it would clamp onto the goal in a weird way, and would not allow the intake to score rings onto it. The goal went way too far into the clamp and tilted way too much, with the goal ending up inside of the hood.

Designed by:

Adam X

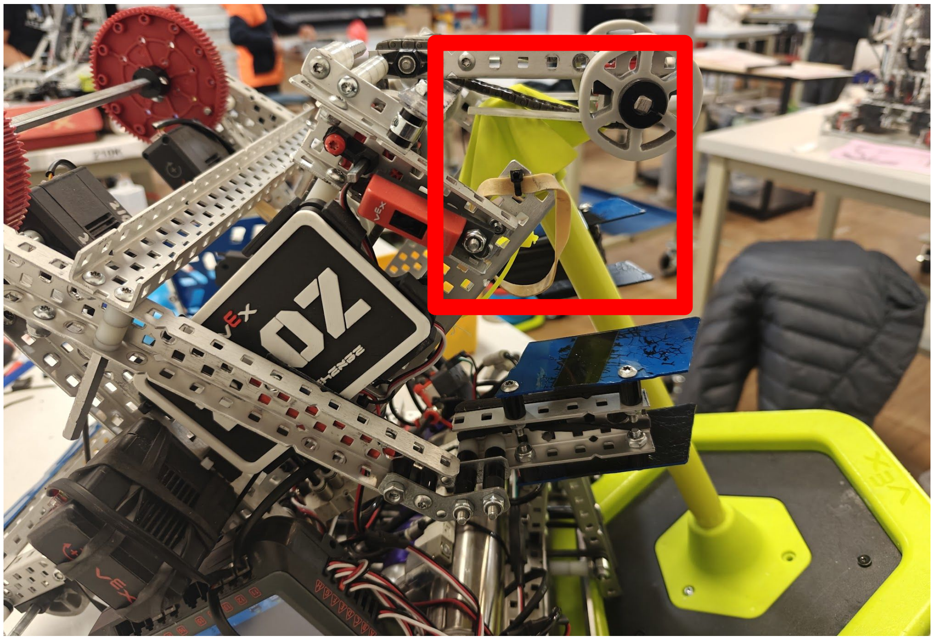
Witnessed by:

Maxwell L

Improving the Clamp

2024-11-21

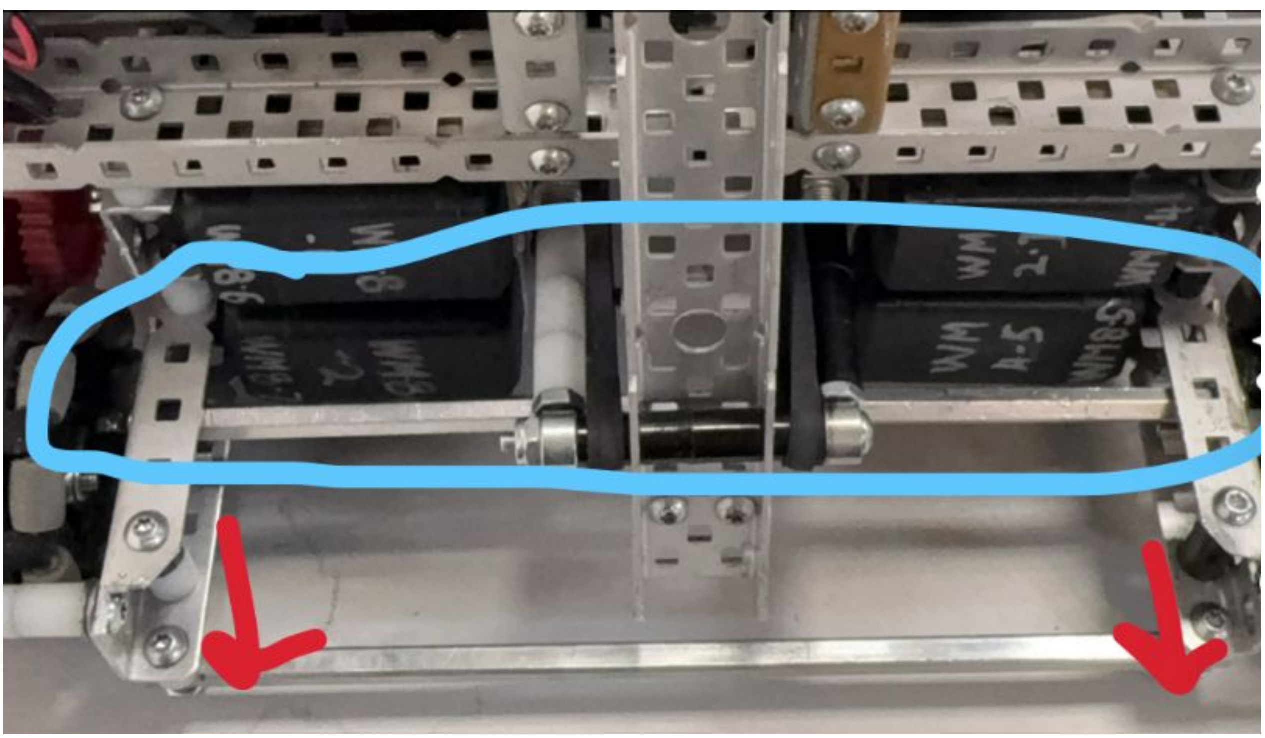
▸ **Problem:** The clamp is too low compared to the goal



The goal tilts too much and makes the intake dysfunctional

[Plan/Design Solution]

To fix this, the problem was that the standoff in the back of the clamp was holding the goal in place when we ran into it too much, so we moved it forward one hole.



Designed by:

Adam X

Witnessed by:

Maxwell L

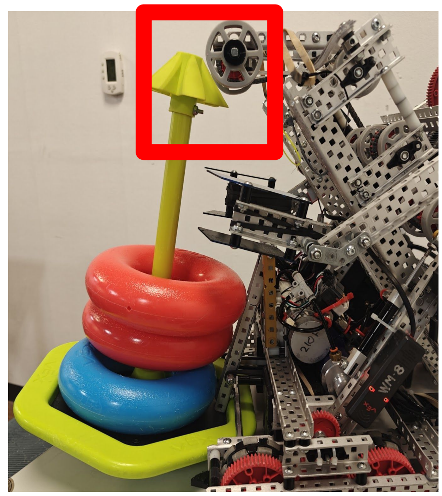
Improving the Clamp

2024-11-21

▶ **Problem:** The clamp is too low compared to the goal

[Test Solution] and **[Identify Problem]**

The solution ended up being too extreme, and the clamp made the goal tilt too little, so that the hood was too far away to score rings onto it.

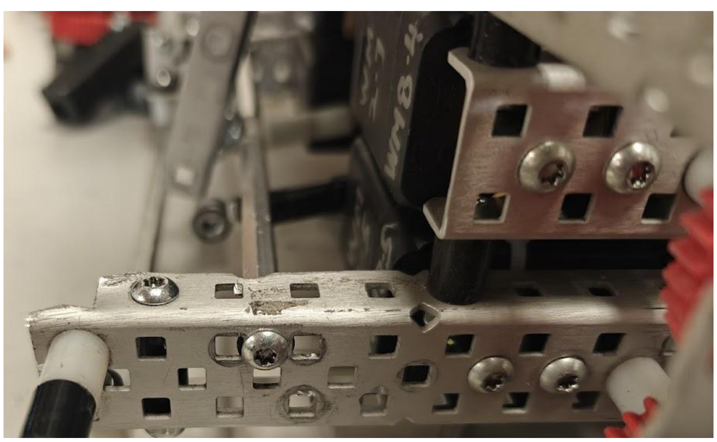


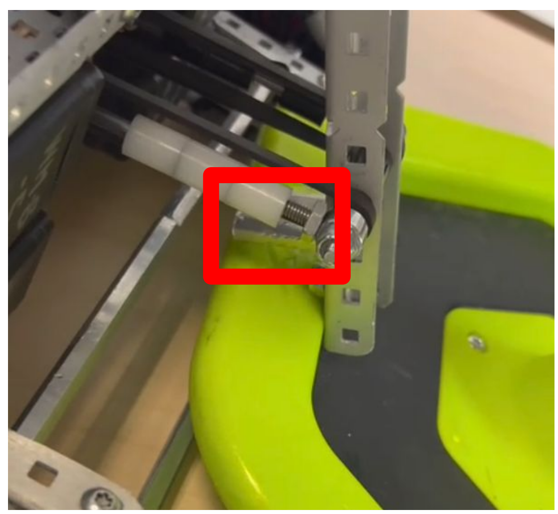
[Brainstorm Solutions]

Since moving it one hole forward didn't work, and having it one hole back didn't work, the only solution that we had was to drill out a custom hole that was in between the two.

[Plan/Design Solution]

We also decided to add two 1" standoffs to the clamp because another problem we had was that the clamp didn't hold on to the goal tight enough, and it moved around inside the clamp from time to time. The standoffs prevented the mobile goal from moving up and down, reducing the room that to move around previously





[Test Solution]

When testing this out, we clamped on to a goal and tested out the intake to see if the goal was at the right angle, and found that it was a perfect middle ground between the two previous solutions, and the. The goal will never clamp on the inside part and dig into the mats, the goal tilts perfectly on both corner and flat sides, and the goals drops fully when let go.

Designed by: Adam X

Witnessed by: Maxwell L

Tournament Analysis: Bowness Robotics

2024-12-01

▶ **Focus:** Analyze our performance at the Bowness Robotics VEX V5 Robotics Competition

Bowness Robotics VEX V5 Robotics Competition

5-0-1 : 14 / 27 / 41	
Qualifications Rank:	2nd/32
Elims Alliance Partner	210K
Elims Rank:	Tournament Champion
Driver Skills Score:	37
Programming Skills Score:	8

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
Q-5	210Z, 9409Y	45519E, 99197S	**42** - 0
Q-16	3388C, 45519B	210Z, 99197Z	16 - 16
Q-24	2088A, 3300C	210Z, 2088G	6 - **32**
Q-28	210Z, 3388H	3388N, 45519F	**54** - 0
Q-34	210K, 6659D	210Z, 45519C	19 - **30**
Q-44	210Z, 6659B	6659C, 9409X	**41** - 0
R16 5-1	210K, 210Z	2088S, 3300C	**46** - 1
QF 3-1	210K, 210Z	3388C, 3388K	**39** - 0
SF 2-1	210K, 210Z	3388H, 3388S	**37** - 3
F 1-1	9409Z, 99197	210K, 210Z	24 - **35**

Designed by: Adam X

Witnessed by: Richard F

Tournament Analysis: Bowness Robotics

2024-12-01

▶ **Focus:** Analyze our performance at the Bowness Robotics VEX V5 Robotics Competition

Bowness Robotics VEX V5 Robotics Competition

During this tournament, we went 5-0-1 in qualifications, ranking second because of our autonomous win points and chose our sister team 210K to once again form an alliance, this time as the second seed. In eliminations, we went all the way to the finals and ended up winning the tournament against 9409Z and 99197T. The robot was extremely consistent and reliable throughout the entire day, and there weren't any major flaws with it. Regarding skills, the skills field and the tournament was not built properly, specifically the poles in the mobile goals were not fully inserted into the base plate, so they were actually taller than usual. This caused our clamp and scoring system to not function properly which was extremely unfortunate.

Next steps leading towards the STEM Innovation Academy VEX V5 Robotics Competition on December 14 would be to get as much game and strategy review as possible, as well as getting as much driver practice as possible. Another priority for the next tournament is to maximize our skills score. We've set an ambitious goal of top 20 in the world, which is as of right now, 103 points combined.

What worked:

- Wall stake mechanism was very fast and worked consistently
- The wall stake key for aligning the robot against the wall stake was also extremely helpful and time saving
- Arm was consistent at clearing corner ring stacks and didn't show any signs of damage
- Drivetrain was consistent and reliable
- Intake lift piston worked well
- Autonomous paths were consistent

What didn't work:

- Intake had some inconsistencies, but it can be improved with minor adjustments
- The skills field was set up incorrectly so skills rungs didn't work
- Elevation didn't work

Designed by: Adam X

Witnessed by: Maxwell L

Improving the Elevation

2024-12-05

▸ **Problem:** The elevation mechanism doesn't work anymore

[Identify Problem]

During the Bowness tournament, whenever we went to elevate our robot at the end of a match, we found that the side that was no ratcheted would hang lower than the other side and touch the ground.

[Brainstorm Solutions]

We came up with a few potential solutions:

1. **Make the mechanical advantage on the system better**
   Pros: Not many visible changes
   Cons: Hard to know exactly what to change
2. **Ratchet both sides**
   Pros: Most likely to be consistent
   Cons: Hard to build.
3. **Reduce unnecessary weight on the robot**
   Pros: Very simple, straightforward. Not much thinking needed to develop a new solution.
   Cons: Might disable or limit other functions of the robot

[Select Solution]

We went with ratcheting both sides of the wall stake mechanism to ensure that we didn't damage it by having one side take more stress than the other and that the robot was parallel to the ground while elevated.

[Plan/Design Solution]

To build this solution, we did a little brainstorming on how we could actually build this, and we remembered something that we built in Spin Up, which was a large standoff attached to a piston that held the expansion standoffs in place so that they wouldn't fire. We thought that we could do a similar design for building a ratchet on both sides of the wall stake mechanism.

Designed by: Adam X

Witnessed by: Richard F

Improving the Elevation

2024-12-05

▶ **Problem:** The elevation mechanism doesn't work anymore

[Identify Problem]

During the Bowness tournament, whenever we went to elevate our robot at the end of a match, we found that the side that was no ratcheted would hang lower than the other side and touch the ground.

[Brainstorm Solutions]

We came up with a few potential solutions:

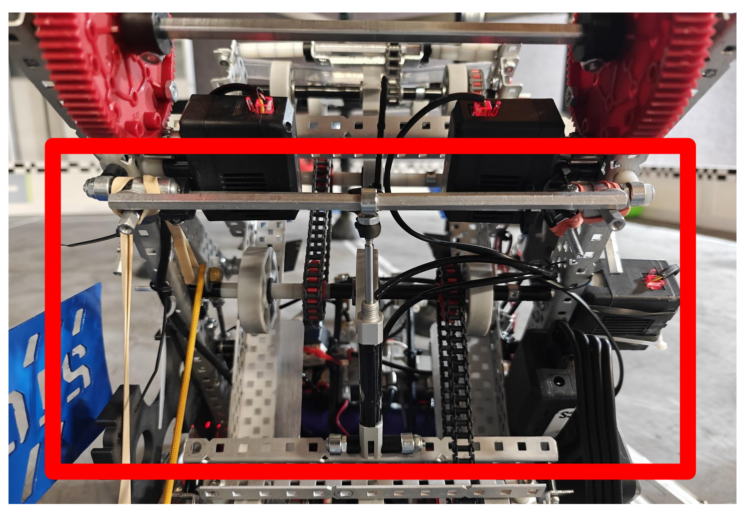
1. **Make the mechanical advantage on the system better**
   Pros: Not many visible changes
   Cons: Hard to know exactly what to change
2. **Ratchet both sides**
   Pros: Most likely to be consistent
   Cons: Hard to build.
3. **Reduce unnecessary weight on the robot**
   Pros: Very simple, straightforward. Not much thinking needed to develop a new solution.
   Cons: Might disable or limit other functions of the robot

[Select Solution]

We went with ratcheting both sides of the wall stake mechanism to ensure that we didn't damage it by having one side take more stress than the other and that the robot was parallel to the ground while elevated.

[Plan/Design Solution]

To build this solution, we did a little brainstorming on how we could actually build this, and we remembered something that we built in Spin Up, which was a large standoff attached to a piston that held the expansion standoffs in place so that they wouldn't fire. We thought that we could do a similar design for building a ratchet on both sides of the wall stake mechanism.



Designed by:

Adam X

Witnessed by:

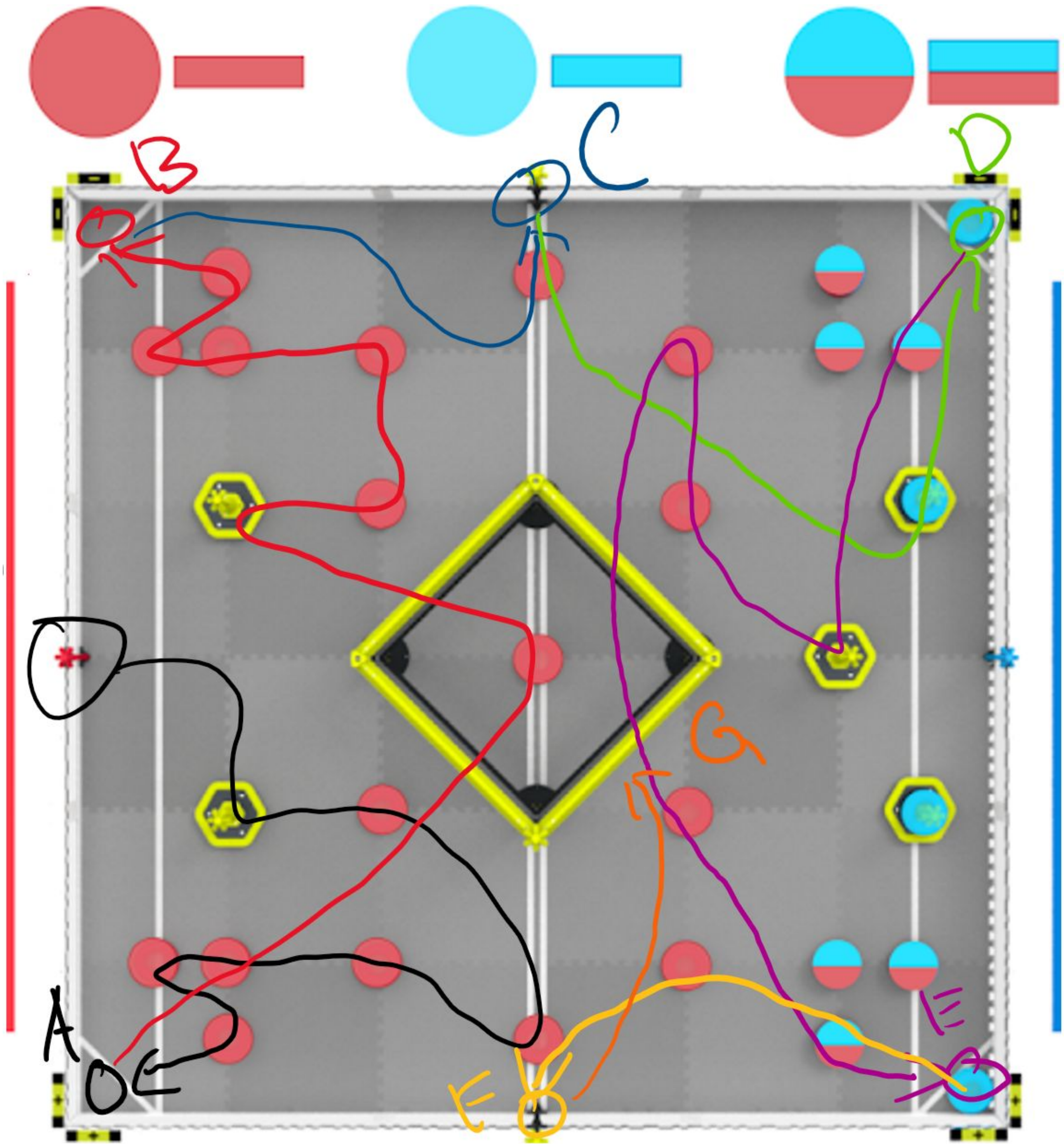
Richard F

Driver Control Skills Path

2024-12-09

▸ **Focus:** Figure out what our driver skills path will be

[Brainstorm Solutions]



Designed by:

Adam X

Witnessed by:

Kevin Z

Driver Control Skills Path

2024-12-09

▶ **Focus:** Figure out what our driver skills path will be

[Plan a Solution]

For our driver skills route, we wanted come up with something that would be comfortable for Kevin to drive and also something that was efficient.

Step	Route 1
A	Score alliance stake, clamp goal, score 6 rings, and place in the corner
B	Drive through the middle to intake the ring, clamp onto the goal, score 6 rings and place into the corner
C	Score wall stake
D	Push goal into the corner
E	Back up and clamp goal, score ~4 rings and place into corner
F	Score other wall stake
G	Climb the ladder

[Test Solution]

When running skills using this route, we had a really strong start and on average we finished step C, scoring the wall stake, at around 30 seconds or sometimes a bit earlier. However, during the second half of the run, we ran into some inconsistencies and issues. We lost a lot of time scoring rings onto the third goal and placing it into the far corner because of the pile of rings close to the corner. Rethinking our plan, we modified the second path into something we thought was more efficient on the next page.

Designed by:

Adam X

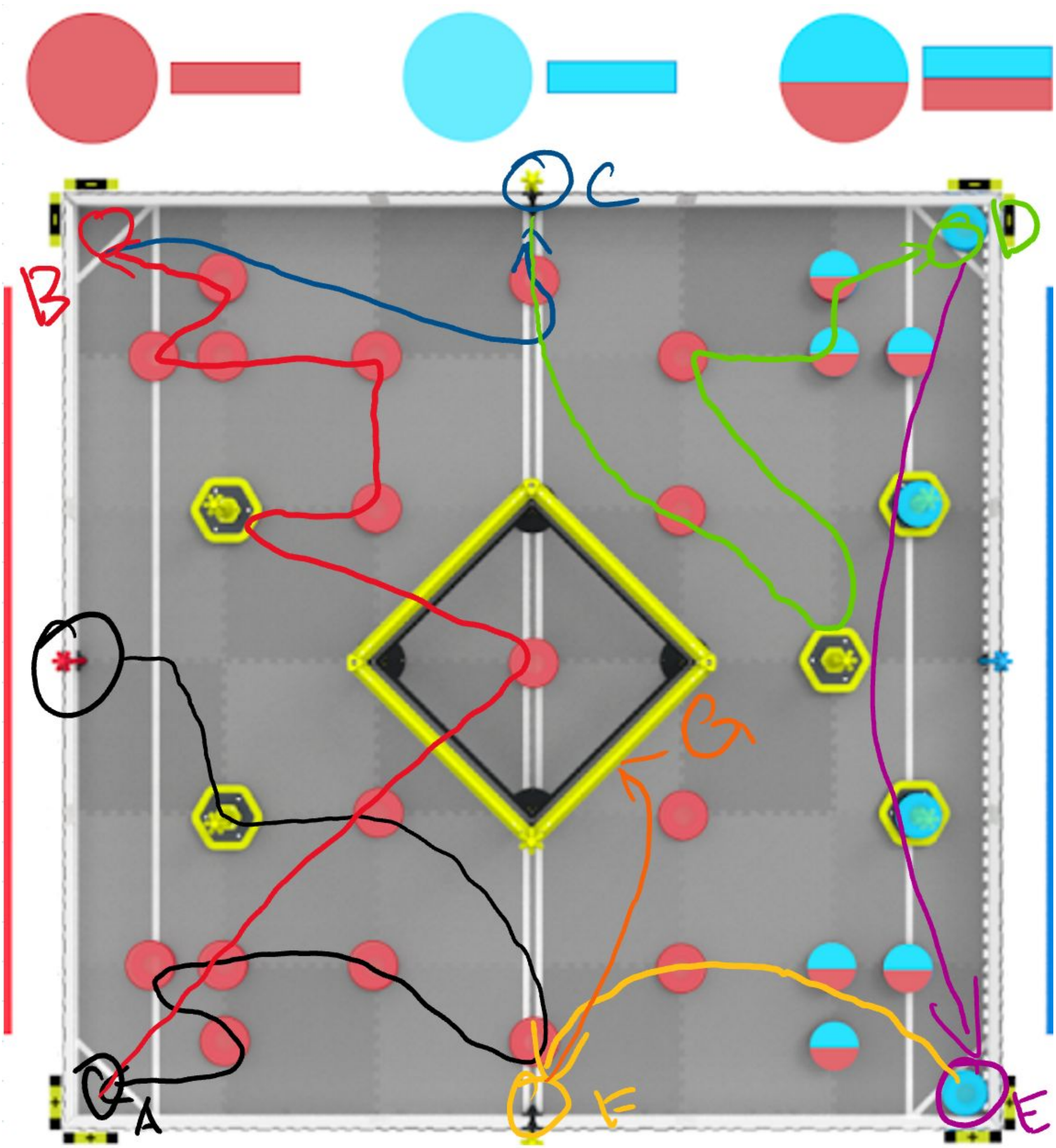
Witnessed by:

Kevin Z

Improving the Driver Control Skills Path

2024-12-12

▸ **Focus:** Figure out what our driver skills path will be



Designed by:

Adam X

Witnessed by:

Kevin Z

Improving the Driver Control Skills Path

2024-12-09

▶ **Focus:** Figure out what our driver skills path will be

[Plan a Solution]

This path requires the robot to move across the field less and should reduce the amount of time taken to score the third goal and place it in the corner

Step	Route 1
A	Score alliance stake, clamp goal, score 6 rings, and place in the corner
B	Drive through the middle to intake the ring, clamp onto the goal, score 6 rings and place into the corner
C	Score wall stake
D	Back up and clamp goal, score ~4 rings and place into corner
E	Push goal into the corner
F	Score other wall stake
G	Climb the ladder

With this path, we swapped steps D and E, which we thought would be easier for Kevin as a driver. We also made sure to intake a ring after doing the wall stake before clamping on the goal just to save a few seconds.

[Test Solution]

When testing out skills rungs with this route, we found that it was definitely a lot more efficient, and we finished step D at around 40 seconds, meaning it took us 10 seconds to complete step D. We thought this could have easily been improved with more practice, and that Kevin just needed to get comfortable with the path. The

Designed by:

Adam X

Witnessed by:

Kevin Z

Tournament Analysis: STEM IA

2024-12-16

▶ **Focus:** Analyze our performance at the STEM Innovation Academy VEX V5 Robotics Competition

STEM Innovation Academy VEX V5 Robotics Competition (HS) High Stakes Last Chance in 2024

6-0-0 : 12 / 30 / 51

Qualifications Rank:	1st/45
Elims Alliance Partner	210K
Elims Rank:	Tournament Champion
Driver Skills Score:	48
Programming Skills Score:	34

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
Q-10	6659D, 45519E	210Z, 3300A	0 - **48**
Q-15	210Z, 3300D	5760E, 99197B	**28** - 0
Q-30	2088A, 27455B	210Z, 3300E	0 - **27**
Q-35	3388K, 27455B	210Z, 27455X	25 - **26**
Q-49	210Z, 221S	5760A, 6659A	**41** - 14
Q-63	3388A, 45519F	210Z, 5760F	12 -**44**
R16 1-1	210K, 210Z	3300B, 3300G	**42** - 0
QF 1-1	210K, 210Z	5760A, 5760E	**46** - 9
SF 1-1	210K, 210Z	221S, 221X	**40** - 24
F 1-1	210K, 210Z	3388C, 3388S	**43** - 21

Designed by:

Adam X

Witnessed by:

Richard F

Tournament Analysis: STEM IA

2024-12-16

▶ **Focus:** Analyze our performance at the STEM Innovation Academy VEX V5 Robotics Competition

STEM Innovation Academy VEX V5 Robotics Competition (HS) High Stakes Last Chance in 2024

During this tournament, we went undefeated, 6-0-0 in qualifications, ranking first out of 45 teams, our biggest local tournament so far. Again, we went ahead and chose our sister team 210K to form the first seed. We went on to win the tournament and won against my school teams 3388C and 3388S, and also got second in skills. We had major issues in regards to consistency the day before the tournament and we could not figure out why. We were practicing skills and gradually as the day went on, the robot's performance just gradually got worse. We did our best to fix it the day of the competition but it wasn't perfect, and all we could do was just try to power through the day. This was also why our skills score was relatively low compared to what we were aiming for.

Now we need to hustle and get the next robot ready for the midseason competition that we are hosting in 3 weeks. We predict that combined with upcoming finals, we won't get everything on the robot working by then, so we plan to get the basic subsystems down: drivetrain, intake, clamp, wall stake mechanism, arm mechanism for corner clearing, and if we are done before then, add more features.

For the next robot, we will be building with anodized metal, which means that we will need to build more carefully, so we will be doing a lot more CAD designs with the robot so that we have an solid understanding of what to build before building it.

What worked:

- Most things that worked last tournament were the same this one

What didn't work:

- Intake broke down the day before the competition, and the issues carried onto the next day.
- None of the autonomous paths landed during qualifications, leading us to get no autonomous win points.

Designed by: Adam X

Witnessed by: Maxwell L

Team Scheduling - Calendar

2024-12-16

▶ **Focus:** Figure out our plans for the next few months

This is a rough outline of how we wanted to allocate our time for the including two tournaments

December and January Calendar

Sun	Mon	Tue	Wed	Thu	Fri	Sat
15	16	17	18	19	20	21
	Now	Finish Drivetrain				
22	23	24	25	26	27	28
					Intake Concept Completed	
29	30	31	1	2	3	4
Working Intake + Clamp Concept					Intake + Clamp Finalized	
5	6	7	8	9	10	11
		Wall Stake Mechanism Concept		Finalize Robot and Wire Electronics	Set Up for Tournament	Volunteer for Middle School Midseason
12	13	14	15	16	17	18
Midseason Tournament			Finish Adjustments from Tournament			
19	20	21	22	23	24	25
			Finialized Robot			Driver Practice + Program for the Rest of the Week
26	27	28	29	30	31	1
			Packing + Tournament Preparation	Flight, Check-in, Inspection, and Skills	**Rumble in the Rockies**	**Rumble in the Rockies**

We have a very tight timeline approaching the next tournament, so we need to use our time well.

Designed by: Adam X

Witnessed by: Kevin Z

Drivetrain Brainstorming

2024-12-02

▸ **Problem:** What is the best gear ratio for our drivetrain?

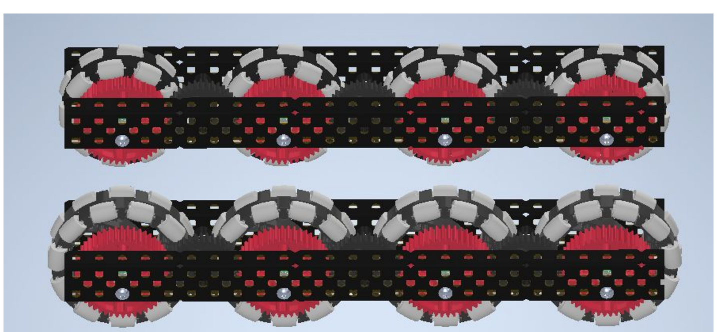
[Brainstorm Solutions]

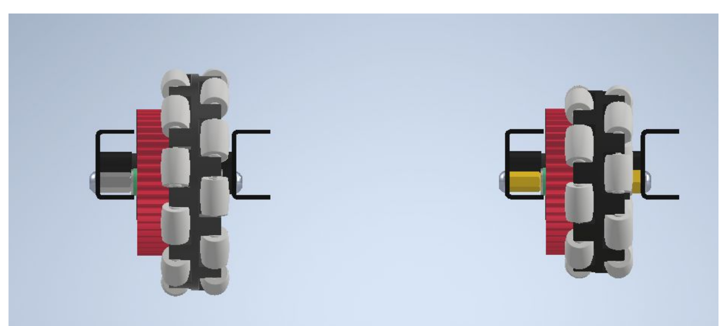
1. **Same drivetrain**
   One option we could do is stick with the same drivetrain ratio and speed that we had before. As the driver, Maxwell is used to it by now and it would be the most comfortable with him.
2. **600rpm direct drive**
   This is something we ran for worlds last year. Using the blue 600rpm motor cartridge, we could link all of the wheels using a 1:1 gear ratio.
3. **450rpm**
   Our last option was to run a 3:4 gear ratio. This design has has become very common across robotics recently, and we also ran this design previously last year.

[Select a Solution]

For our drivetrain, we decided to go with **solution 3**, 450rpm. The was because based on our previous robot, we thought that we were a little more on the slower side. Especially since we are going to two signature events as well as provincials, we will be competing against many high caliber teams, and they will all most likely be faster than us if we stay with the same drivetrain ratio. On the other hand, running 600rpm last year was very hard to drive, because of the speed and just how uncontrollable it was. We also would be giving up too much torque, which we still need to have for pushing matches and fighting for control of corners. **Solution 3** is a combination of the two, and therefore what we believed was the best solution.

Another issue we ran into was a miscommunication, as the team was split between doing 450rpm on 2.75" wheels or 3.25" wheels. To decide, we quickly CADed out a model of both wheel sizes to compare the differences.





Designed by:

Adam X

Witnessed by:

Maxwell L

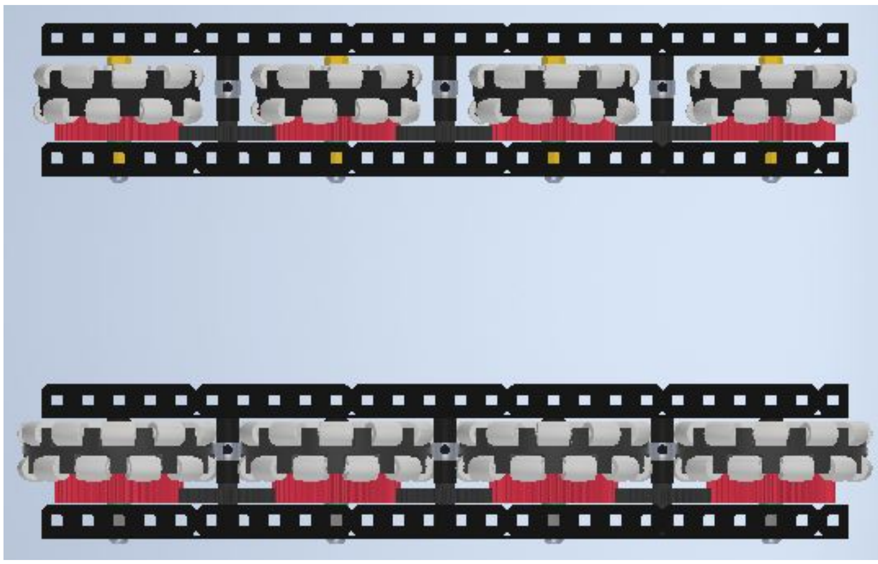
Drivetrain Brainstorming

2024-12-02

▶ **Problem:** What is the best gear ratio for our drivetrain?

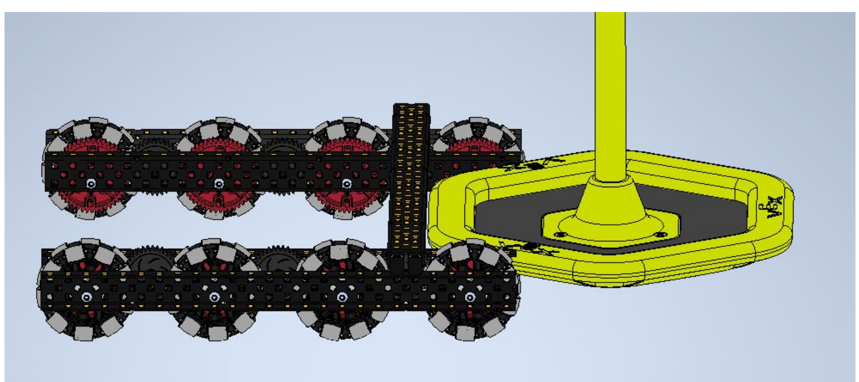
[Select a Solution]

Comparing the sizes, we thought that 3.25 wouldn't be that much of a difference size-wise, and on 2.75" wheels, it would be even slower than what we have right now, so we went with the 3.25" wheels, since we really want that extra speed.



[Plan/Design Solution]

To ensure that we maintain structural integrity in the drivetrain, we decided to focus on the drivetrain for a much longed time than we used to. First, we started off with a CAD design of our drivetrain, so we know what to build and we don't have to spend time figuring that stuff out. It also allows us to make sure that it is built properly and that it is rugged and durable.



Designed by:

Adam X

Witnessed by:

Maxwell L

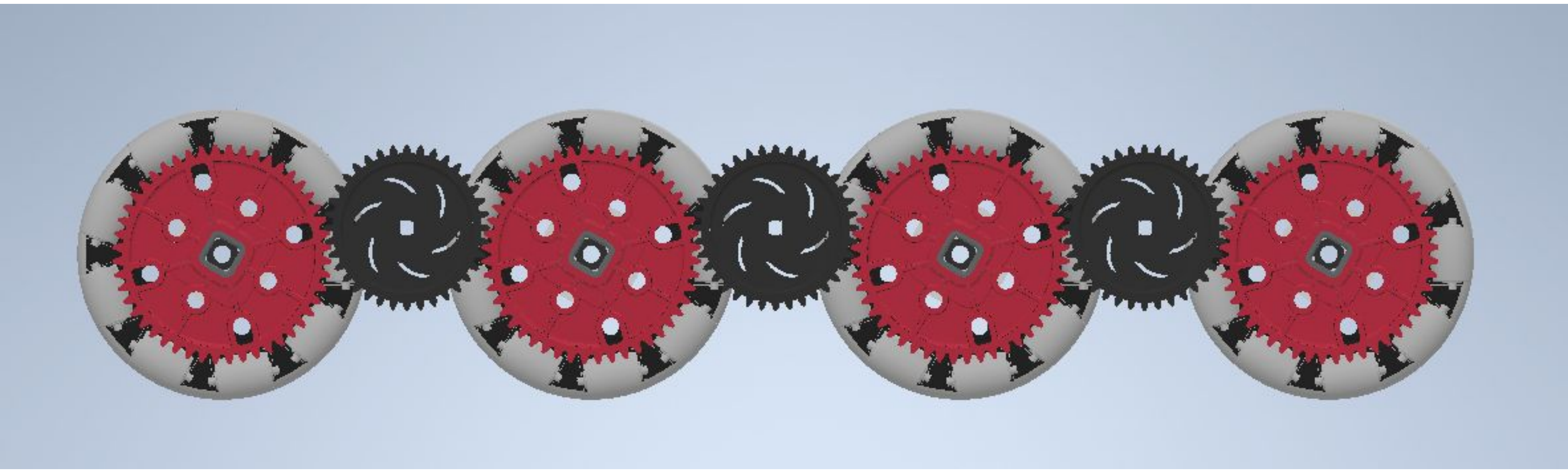
Building the Drivetrain Frame

2024-12-05

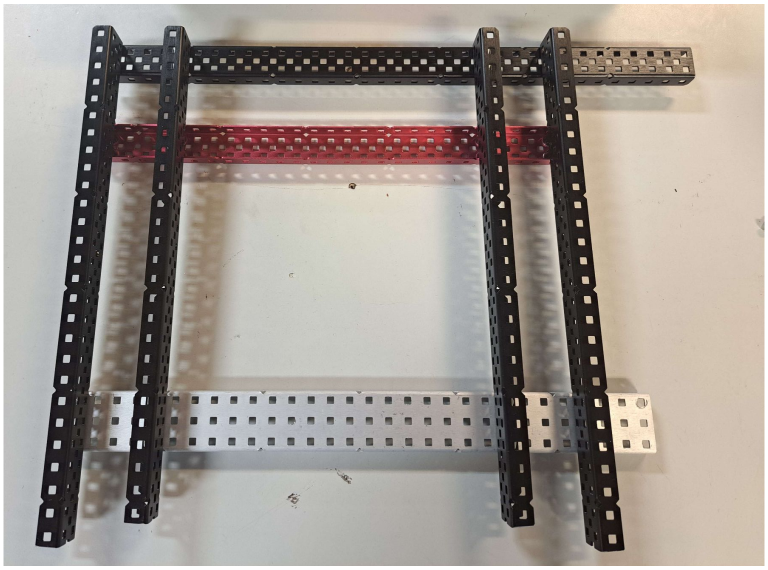
▶ **Problem:** How can we make sure the drivetrain is strong?

[Plan/Design Solution]

Here's another CAD of a close-up view of the gear ratio:



Due to the unordinary spacing of this particular gear ratio, the wheels must be on the bottom row of holes while the 36-teeth gears need to be in the middle row of holes on the c-channel.



Now that we had an idea of what to build, we really wanted to emphasize getting the drivetrain strong, rugged, and durable. To do this, we first built a 3-wide c-channel across the drivetrain to act as a cross-bracing throughout the building process, and once we were all done and all of the main bars were built, we could then remove the 3-wide c-channel. This process ensures that while we are building the drivetrain, all of the c-channels stay straight, not only does this achieve our goal of keeping it stronger, but it also reduces the friction on our wheels.

For now, all we have built is this skeleton frame of the drivetrain. Next, we need to make all of the ball bearing wheels so they are ready to be put on the drivertrain

Designed by:

Adam X

Witnessed by:

Maxwell L

Ball Bearing Wheels

2024-12-12

▸ **Problem:** How do we prepare the ball bearing wheels?

[Select a Solution]

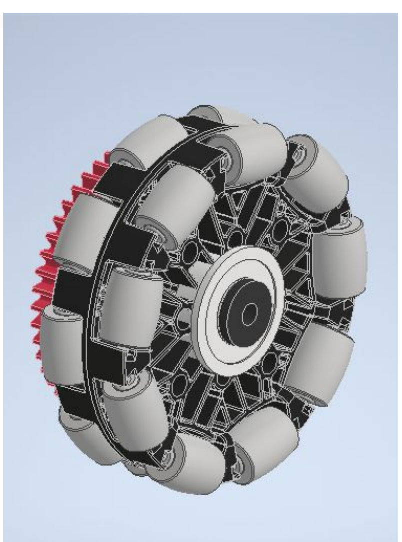
For this robot, we decided to use **ball bearings** over regular bearings for the drivetrain, but with our own little twist. The ball bearing holders that VEX makes only work on 3-wide c-channels, and their friction is also suboptimal. The ball bearings themselves are quite good and have much lower friction than regular bearings.

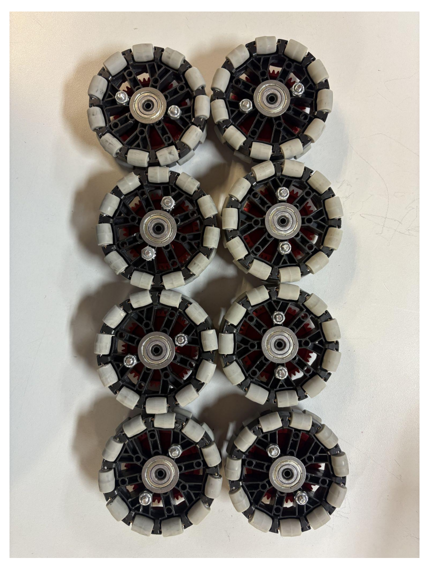
For those reasons, last year, we developed a strategy that involves using the ball bearings, but without any of the major drawbacks mentioned above.

[Plan/Design Solution]

Instead of using the holders that VEX makes, we drilled out our wheels so that we could fit a ball bearing into them. This way, we don't have to use the holders but we can still utilize them in a way that improves our drivetrain friction.

However, one downside of using ball bearings is their weight. Since they are metal and quite dense, the weight difference from a normal bearing is huge, and can increase the weight of the robot significantly. Still, we still decided to use them because of their increase to the robot's acceleration, which we thought was a key aspect of this game due to the field's openness and lack of obstacles on the ground.





Designed by:

Adam X

Witnessed by:

Maxwell L

Building the Drivetrain

2024-12-17

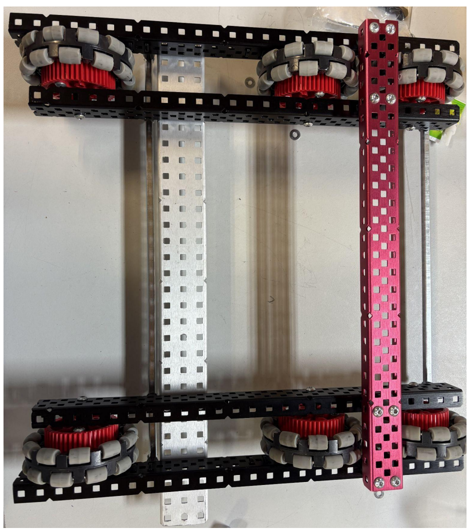
▶ **Problem:** Build the drivetrain now that we have everything

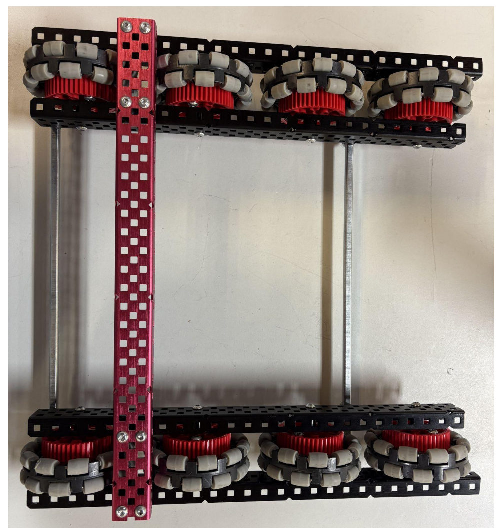
[Identify Problem]

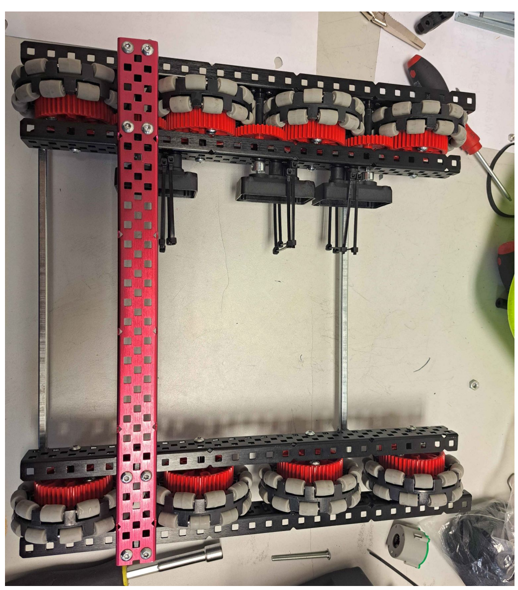
Previously, we were waiting for the motors and electronics on our old robot to become available, but now that our competition is over, we can take it apart for the new robot.

[Plan/Design Solution]

The drivetrain is the foundation of the entire robot, so we wanted it to be rugged and durable, able to support the weight of everything that we build. We did this by ensuring all bars were straight, initially putting supporting bars like the 3-wide c-channel in the picture below while building, and replacing them with real bars later.







Designed by:

Adam X

Witnessed by:

Maxwell L

Drivetrain Funnels and Lexan

2024-12-24

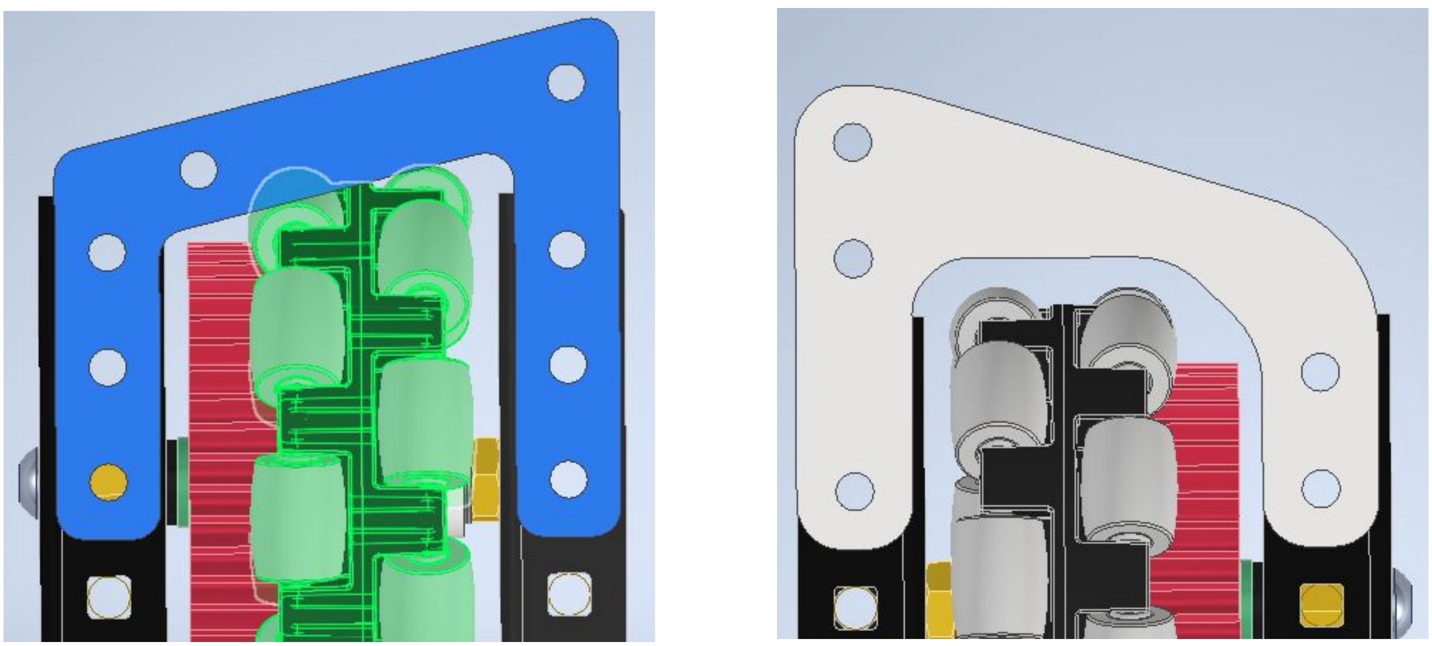
▶ **Problem:** CAD the required lexan for the drivetrain

[Identify Problem]

Now that the drivetrain is complete, we need to design lexan pieces for the ends to act as funnels and support. We decided to base off all of our designs on the previous one, and we noticed that it wasn't very compact and that we had to mount it so that it was sticking out of the drivetrain by a lot.

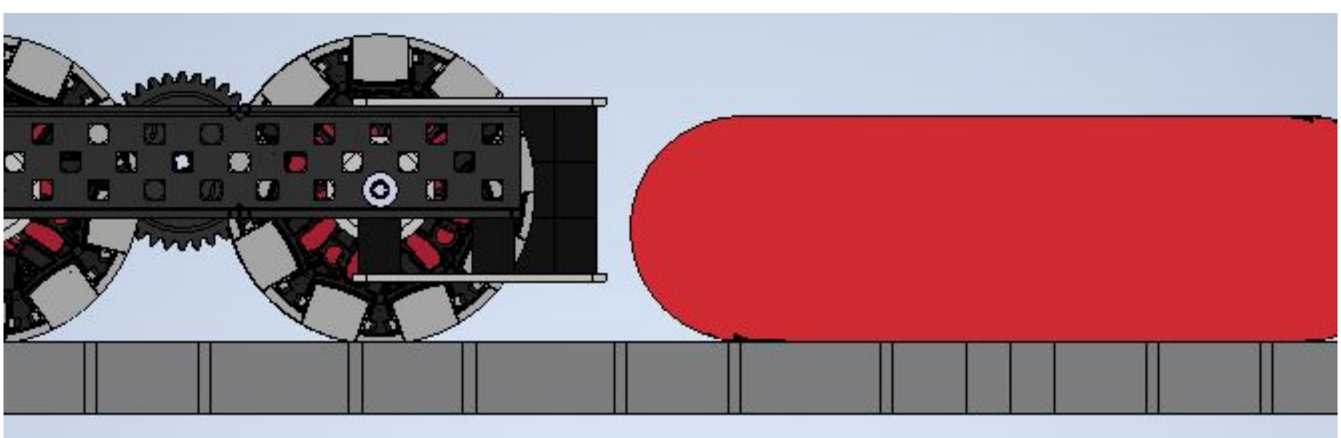
[Plan/Design Solution]

To fix this, we remade the inside of the piece so that it goes around the wheel, instead of just being a flat line. This way, we can mount it closer inside so that it doesn't hit the wheels



Blue is the old design, white is the new one

This also allows us to save more space so we have more leeway in regards to the sizing limit.



Designed by:

Adam X

Witnessed by:

Maxwell L

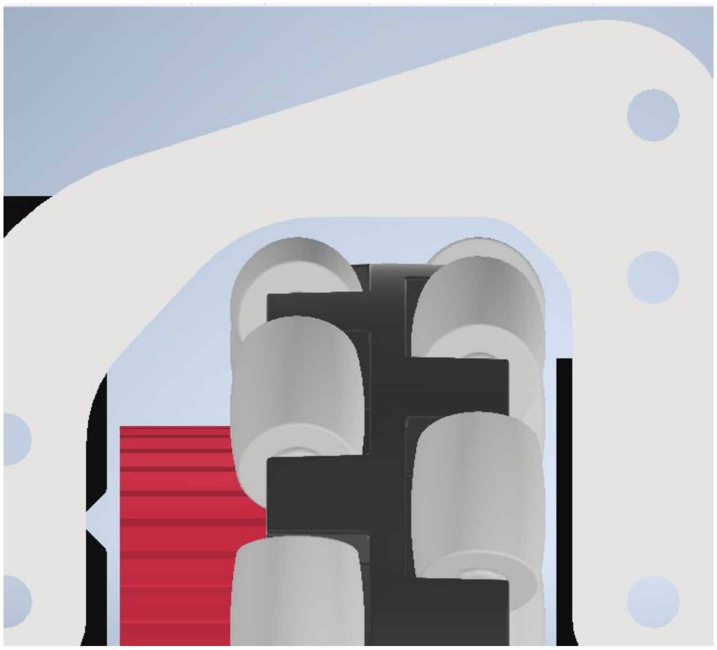
Drivetrain Funnels and Lexan

2024-12-24

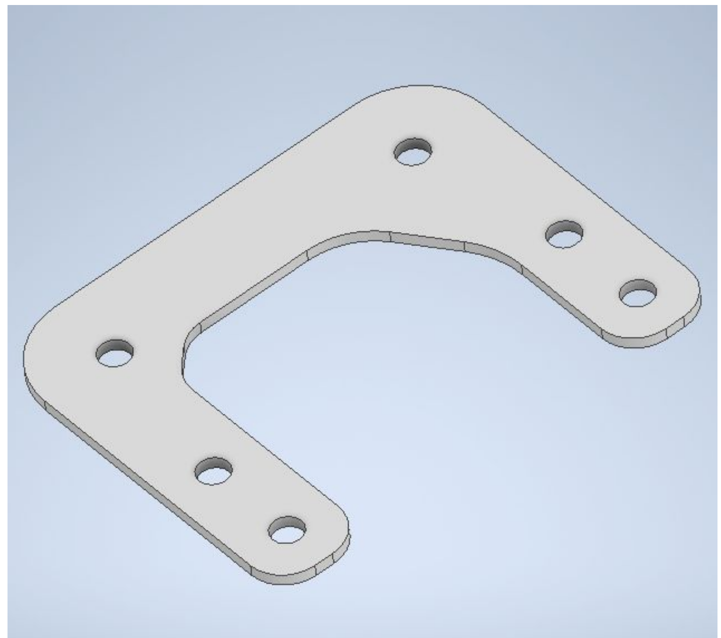
▶ **Problem:** CAD the required lexan for the drivetrain

[Plan/Design Solution]

We also wanted to have lexan on the back, so we modeled the back ones with a similar design, except the outside is flat, since we don't really need to funnel anything into the robot on the back side.

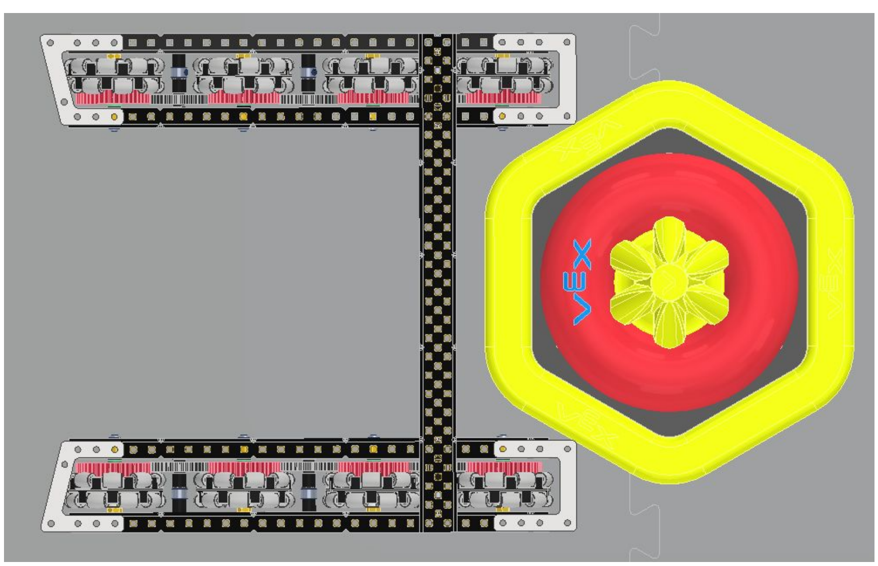


Funnel design



Design copied but with a flat side for the back

On top of providing strength and support to our drivetrain, the back lexan pieces also act as an aligner for our clamp on the back, letting the goal perfectly fit into the back of the robot.



Designed by:

Adam X

Witnessed by:

Maxwell L

Intake Brainstorming

2024-12-26

▸ **Problem:** What design should we use for our intake?

[Brainstorm Solutions]

Currently, there are only 2 major designs for intaking rings: one of them is a flex wheel hood mechanism, that we had previously, and the other design is a conveyor mechanism with hooks on it combined with a first-stage flex wheel intake.

[Select a Solution]

After comparing our Ri24 robot (hook intake) to our first competitive robot (flex wheel) we found that the hook mechanism for an intake and scoring had much more success overall, coming down to two main 3 things:

1. **Flex wheel hood is difficult to tune**
   The process of building a good and consistent flex wheel hood mechanism is very tedious and requires precise tuning, which is a very time-consuming process, as documented in our build process previously. Because of this tuning process a mistuned hood can result in the inability or issue to score.
2. **Hooks are much faster and require less moving parts**
   The flex wheel hood mechanism has so many stages, which little by little, adds up friction and makes the intake/scoring process slower. With hooks, there are only 2 stages of movement, the first stage with flex-wheel and the converter. This also adds to the previous point of hooks not having the same amount of tedious tuning as flex wheels.
3. **Hooks are more easily integrated with other subsystems**
   Since hooks bring the ring above the conveyor rather than being sandwiched in the flex wheel design, subsystems like the wall stake mechanism are more easily to be incorporated. Having no obstructions between the ring and the wall stake mechanism allows for the hold to be a more fine-tuned friction hold without the issues of accidental rubbing.

Designed by:

Kevin Z

Witnessed by:

Richard F

Clamp Design

2024-12-29

▶ **Problem:** How do we improve our clamp from our last robot?

[Identify Problems]

The main issue with the clamp design on our old robot was that because our pistons were mounted behind the the clamp and “pulled” the clamp inwards in order to clamp, a large amount of range of motion was lost because of the indirectness of the piston power relative to the goal.

Because of this, our clamp barely was able to go over the lip of a mobile goal, leading to us missing clamping the goal frequently.

[Plan/Design Solution]

In order to combat this, we realized that by moving the piston to be above the rotatory joint of the clamp joint, the exerted force of the piston would be much more direct towards the mobile goal.

We also decided to implement a locking clamp, which would ensure our possession of the goal and remove the possibility of other teams stealing our goal.

[Test Solution]

We implemented a two-stage standoff locking-clamp, with the specific mechanism being known as an over-center clamp. When the two standoff stages move past parallel, the the clamp “locks” by transferring the force exerted on the goal from the piston to the geometry of the over-centered clamp.

This worked surprisingly well, with the clamp locking without significant force required and a good amount of tilt on the goal.

The first standoff stage, connected to the piston

The over-center joint. Note the joint pressing into the angle bar

The second standoff stage, connected to the clamp

Designed by:

Adam X

Witnessed by:

Richard F

Clamp Design Iterations

2025-01-10

▶ **Problem:** Why did our previous clamp iterations fail?

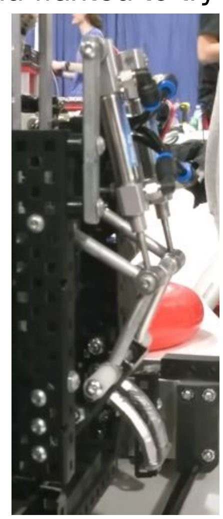
Prior to the design mentioned in the previous page, we experimented with different designs on our clamp, however, none of them worked particularly well, which is why we have our current design. There was not enough time to include these designs because we had tight deadlines for notebooks and all of these designs were quick and not intended to be permanent, meaning we did not take into account documenting them.

From the start, we always had the vision of implementing some sort of **locking** clamp mechanism, so with everything above mind, here is what we tried:

[Brainstorm Solutions]

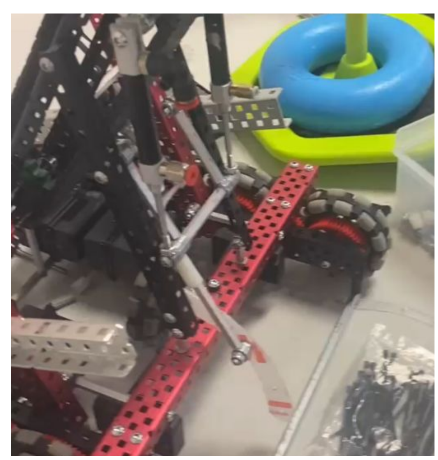
BLRS Clamp

At the Riverbots Signature Event, the FUN Robotics Network released a Pits & Parts video on VEXU team BLRS, featuring a locking clamp that we found interesting and wanted to try out:



One unique design aspected we noticed is that in order to do this, they **bent standoffs**, which we did not want to and did not know how to do, so we tried an alternative using lexan.

[Test Solution]



Testing the lexan design, we noticed that the plastic material was too compliant for our purposes, and could not 100% replicated the effect of the bent standoffs. While clamping, the goal moved around too much and the clamp did not have much of a grip on it.

Theses tests eventually led us to our current design on the previous page.

Designed by: Adam X

Witnessed by: Alex S

Building the Intake

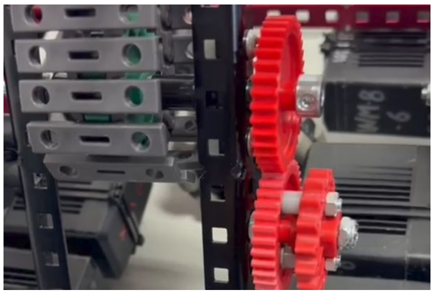
2024-12-27 to 2025-01-02

▶ **Problem:** What design should we use for our intake?

[Identify Problem]

Gearing the first stage to the hooks and motor is difficult.

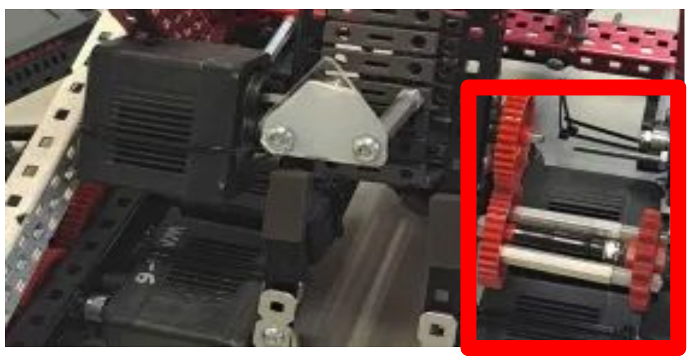
[Plan/Design Solution]



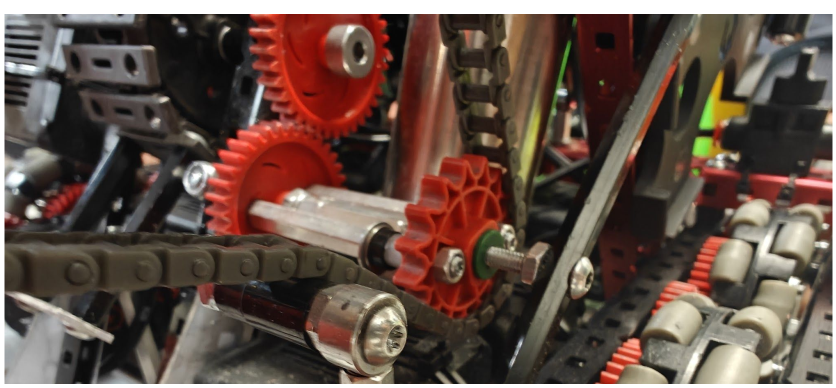
Our solution to this was to attach a sprocket to a gear and chain that to the first stage. Since the hooks need to be spinning in the opposite direction as the first stage, a gear is needed either way, it was an efficient way to chain the first stage to the hooks while saving space

[Test Solution]

Testing it out, the sprocket and chain were too close to the center of the intake, which impacted the robot's ability to actually intake the rings, so we kept extending the spacing between the gear and sprocket so that we could move it more to the side.



Eventually after making it too long, we realised that it was becoming too weak and began causing a lot of friction. To solve this, we drilled a hole for the screw-joint in a support standoff so that it could be connected on both sides, instead of cantilevered, which makes it weak.



Designed by:

Adam Xu

Witnessed by:

Richard F

Building the Intake

2025-01-14

▶ **Problem:** What design should we use for our intake?

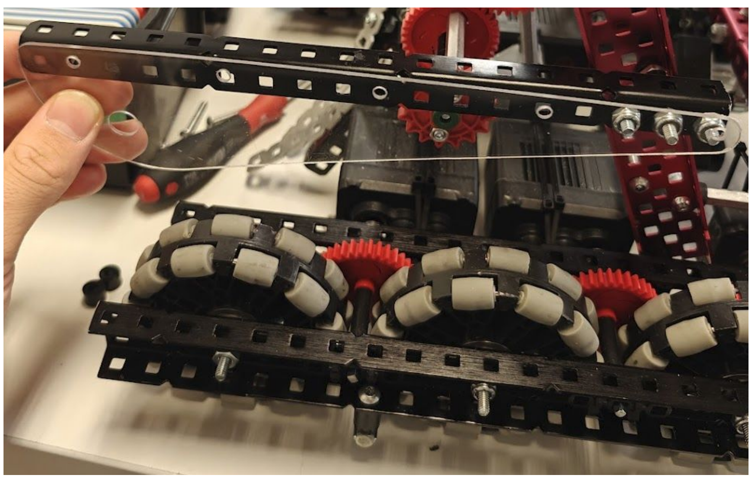
[Identify Problem]

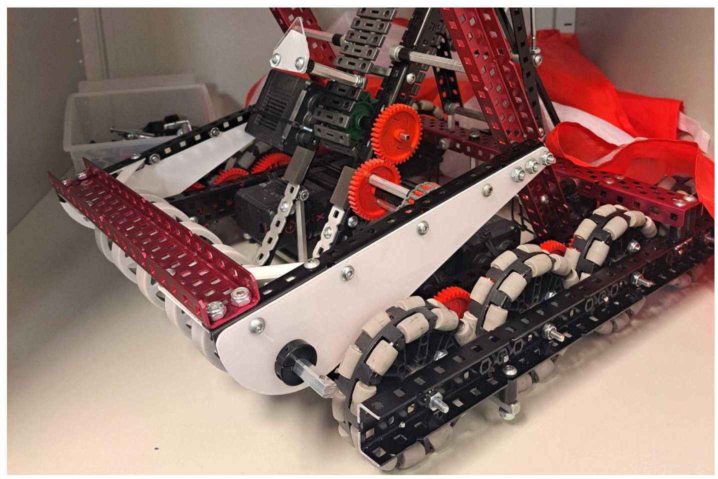
Now that we had a good concept of the hook system, we needed to build the first stage of flex wheels that would intake the rings into the hooks.

[Brainstorm Solutions]

In the past, we had a hard time getting the position and geometry of the first stage right, so instead of iterating and testing different positions, we used CAD to design a piece of lexan for the position of the axle and flex wheels.

[Plan/Design Solution]





Using lexan allows us to save tons of weight by only needing an angle bar on both sides instead of a c-channel. This design saves us the trouble of spending time on tuning the right geometry for the first stage and lets us work faster. Another unique feature of this design is that we also don't need to use bearings. The axle hole in the lexan acts as a bearing, not normally for the high strength axle, but for the high strength spacers. This allows it to freely spin in a circular shape, reducing friction because an axle is square shaped.

Designed by:

Adam Xu

Witnessed by:

Alex S

Wallstake Design

2025-01-16

▶ **Problem:** How do we improve our wallstake mechanism from our last robot?

[Identify Problems]

A few problems we found with the old design of our wallstakes was the lack of versatility it had and a lack of efficiency. The previous design was slow and could only index one ring at a time onto the wall stakes, making it slow and inefficient for us to score wallstakes (especially in skills).

Secondly, we found that a common game strategy used to prevent opponents from utilizing the positive and negative corners was to use the wallstake mechanism to tip over mobile goals. However, this strategy is only possible if the wallstake mechanism is able to rotate 270 degrees.

[Plan/Design Solution]

To accommodate for the new game strategy, we decided to shorten our wallstake mechanism such that it would be in size at max extension, allowing for 270 degrees of rotation.

To increase efficiency when scoring, we decided to include a second level of polycarbonate and foam that could be placed above or below the initial level on the c-channel, so that our mechanism could hold two rings instead of one. We also swapped our wall stake mechanism's speed to 100 RPM instead of the original 25 RPM allowing to run much faster than before.

[Test Solution]

To have a shorter wallstake mechanism, we adjusted the point of rotation on our robot to be higher up and a little closer in so that the distance between the point of rotation and the wallstake cap would be shorter, whilst also allowing us to pick the rings up off the top of the intake.

After testing both above and below the first stage of the wallstake, we found that having our second ring above the initial wallstake mechanism was the best option, allowing us to score two rings on the wallstake quickly and in one index.

[Identify Problems]

The mounting bars for the wall stake mechanism is too thin and can barely fit a ring between.

Designed by:

Alex S

Witnessed by:

Adam X

Wallstake Design

2025-01-16

▶ **Problem:** How do we improve our wallstake mechanism from our last robot?

[Brainstorm Solutions]

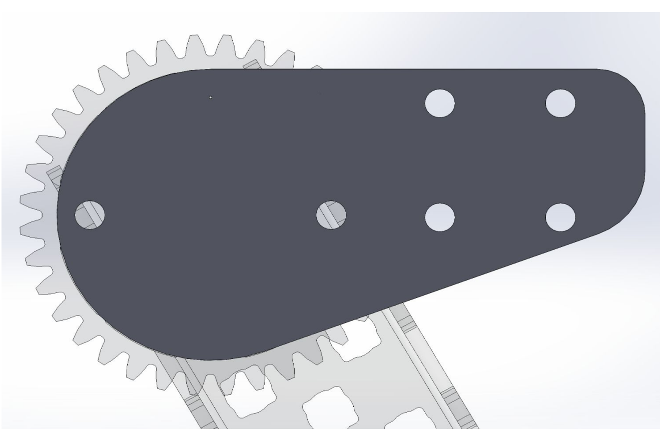
1. **Mount the Bars on the Outside**
   This would be a simple solution as it doesn't require many extra parts or effort. However, the major downside of this solution is that the c-channel rotates on a much longer joint. This makes the structure much weaker, and it would also make the geometry and friction fitting of it extremely difficult to build and tune.
2. **Make a Custom Lexan Piece that can Mount the Bar.**
   This solution is something that our WM VEXU team did, and they found a lot of success with it.

[Select a Solution]

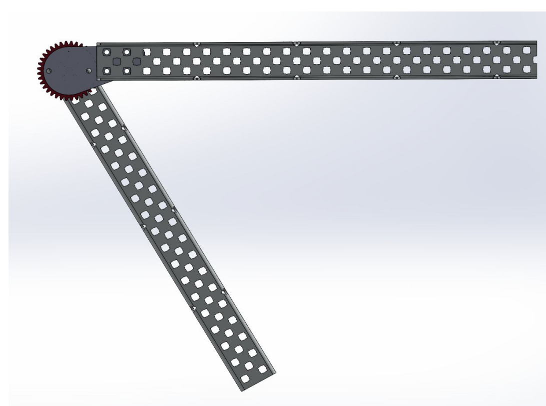
We elected to create a custom lexan piece to mount our wall stake mechanism arms. This was due to the many downsides of solution 1, and that solution 2 generally had no major downsides.

[Test Solution]

To do this, we attached the lexan to a gear that was on the high-strength axle connected to our motor. By doing this, we ensured that our lexan mount was properly anchored and wouldn't suffer from excess strain from the motor. Furthermore, we double-paned our lexan pieces to minimize any chance of the lexan snapping. We found that this worked very well, as it suffered very little bending or cracking throughout our various stress testing.



The lexan piece, mounted onto a gear in CAD.



The lexan piece attached to a wall-stake mechanism arm in CAD.

Designed by:

Adam X

Witnessed by:

Richard F

Building the Wall Stake Mechanism

2025-01-20

▸ **Problem:** How do we improve our wallstake mechanism from our last robot?

[Brainstorm Solutions]

We had two options of how we wanted our wall stake mechanism to work:

1. **Single Ring**

   This is the same design from our last robot and it is very common among all teams.

2. **Double Ring**

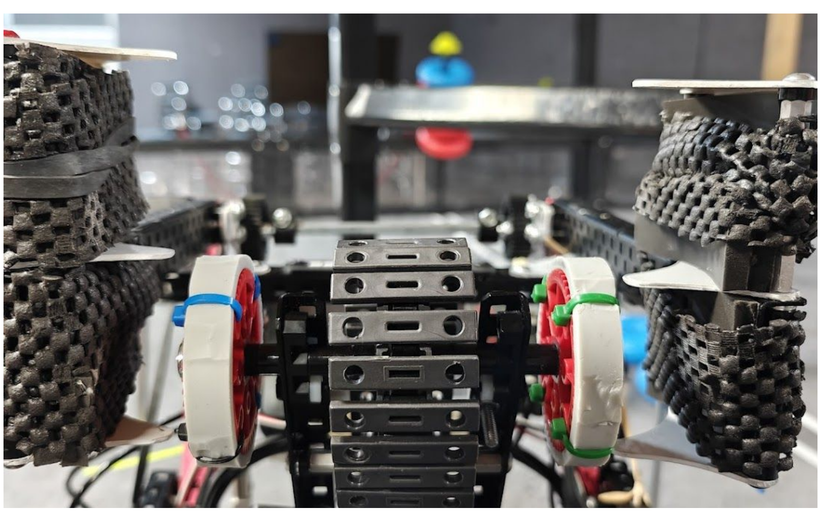
   This design is exactly what it sounds like. Instead of loading one ring at a time, this design simply does 2. We first saw this design from 2145Z on Youtube, where they successfully build a 2-ring wall stake mechanism.

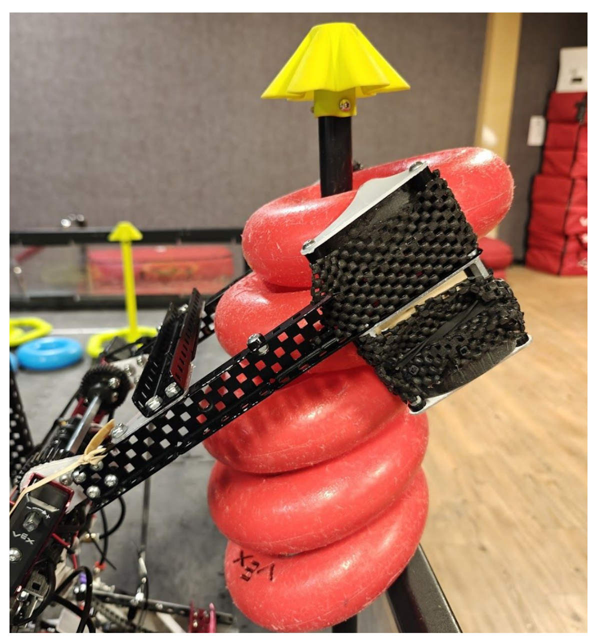
[Select a Solution]

We decided to try out the double-ring design, as even though it was tedious to build and tune compared to the single ring, we believed it was worth the risk and that we had the time to do so.

[Plan/Design Solution]

This build of this mechanism is quite simple, regardless of the single or double ring function, which is a huge reason why we continued to go with the mechanism after using it on the old robot. One minor design choice we made was to actually make the c-channels shorter, which was mentioned two pages prior, reducing the overall range of motion of the mechanism. We thought of this as a benefit more than a downside because now, we could fully extend it to knock over and upright goals with it.





Designed by:

Adam X

Witnessed by:

Alex S

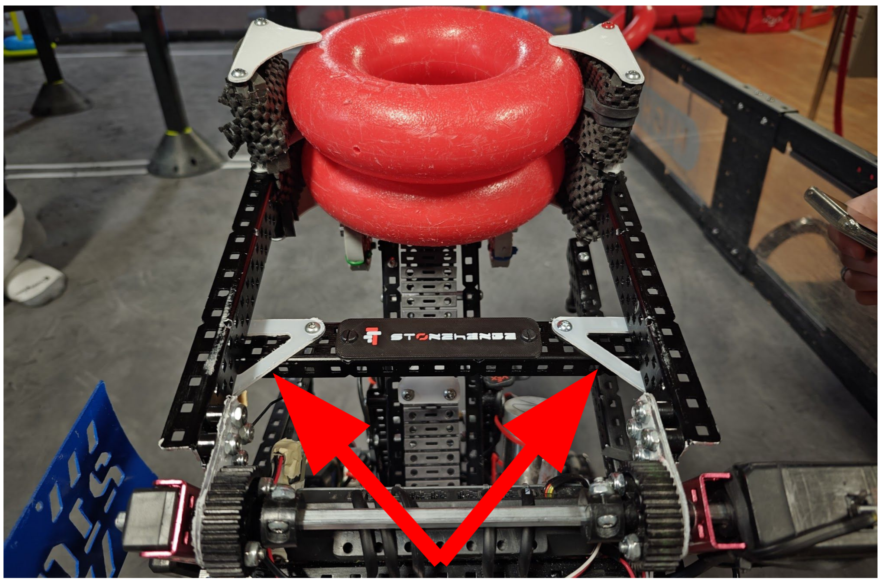
Building the Wall Stake Mechanism

2025-01-22

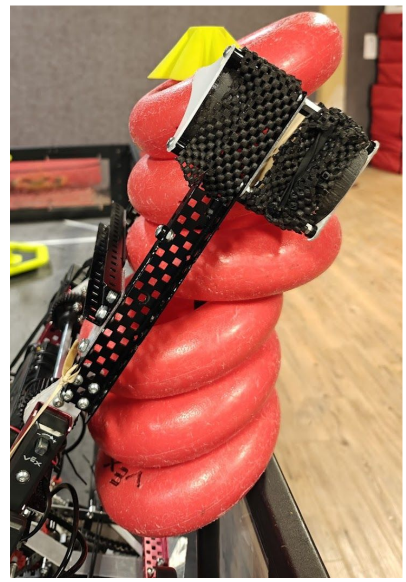
▶ **Problem:** How do we improve our wallstake mechanism from our last robot?

[Test Solution]

After going through lengths of experimenting, testing, reiterating, and tuning with different grips and materials to hold the rings with, we finally settled on a combination of lexan, foam tape, non-slip, and rubber bands to achieve what we thought was acceptable and functional for now, as seen in the images below and on the previous page.

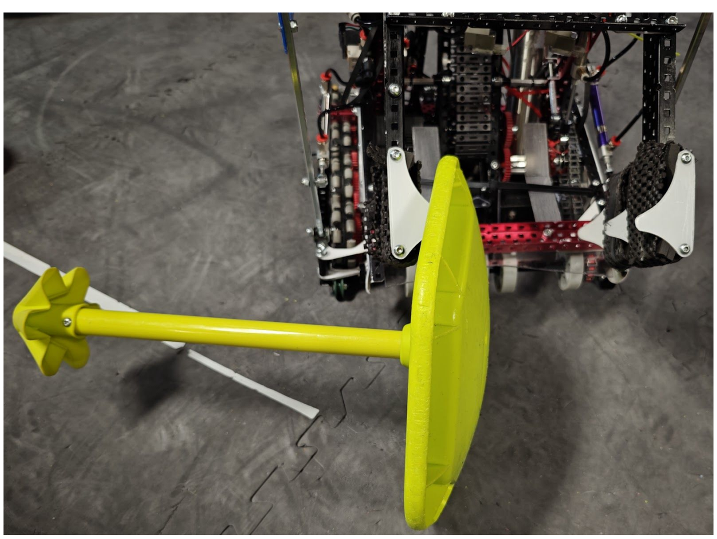


Custom lexan bracing

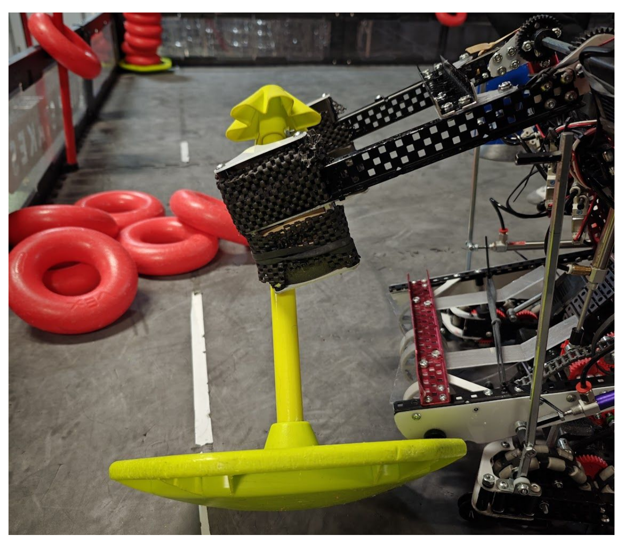


In addition to scoring, we also have the ability to descore rings

As mentioned previously, we also have the ability to knock over and upright goals:



Turning right here would upright the goal



Turning left here would upright the goal

Designed by:

Adam X

Witnessed by:

Alex S

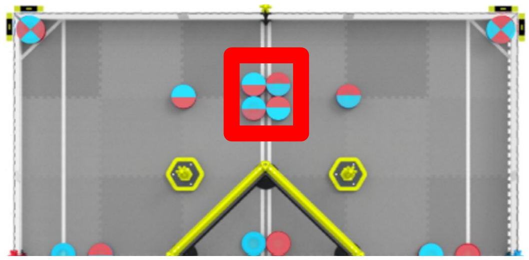
Autonomous Rush Mechanism (Rings)

2025-01-20

▶ **Problem:** How do we ensure a consistent autonomous that avoids external influences?

[Identify Problems]

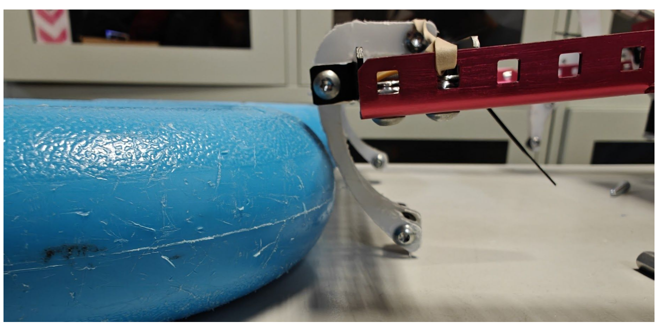
When going for neutral autonomous regions, such as the 8-ring pile, we found that they may have already been disrupted, which harms our pathing and risks unwanted errors.

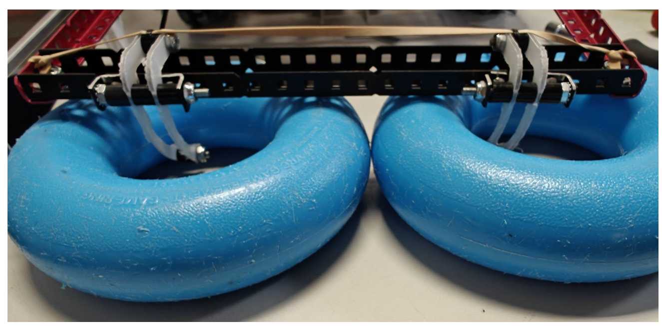


We want to be able to intake these rings freely in autonomous

[Plan/Design Solution]

We decided to create a designated mechanism for rushing rings. By doing this, we can disrupt the other team's autonomous in that region, and we can also safely ensure the security of our run. To do this, we designed a two-piston mechanism that deploys a u-shaped bracket, with two hooks mounted on it. The hooks are created such that they are on a rotating joint, and they are able to push onto the ring but not let go of it. By doing this, we are able to maintain control of the rings that we grabbed onto, while simultaneously only hook onto the rings that we want.





[Test Solution]

When this was implemented, we found that it was very consistent and capable of grabbing the rings from the stack. However, one flaw with this design was that it inhibits the range of the wall-stake mechanism by preventing it from fully rotating downwards. This results in us not being able to untip tipped mobile goals with our wall-stake mechanism, but we are still able to tip goals and score onto the alliance stake with the mechanism.

Designed by: Richard F

Witnessed by: Adam X

Goal Rush Mechanism

2025-01-20

▶ **Problem:** How can we build a mechanism that can beat other robots in grabbing the 5th goal?

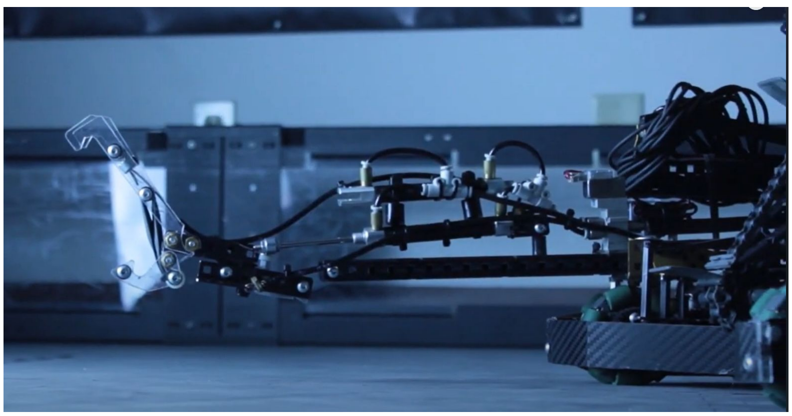
[Identify Problems]

As a team, we've noticed at a lot of tournaments, possession of the 5h goal that starts on the autonomous line is highly desired by all teams and can potentially play a key factor in winning the game.

[Brainstorm Solutions]

For this mechanism, we mainly took inspiration from Tipping Point robots, mainly because there were so many unique goal rush designs that teams made.

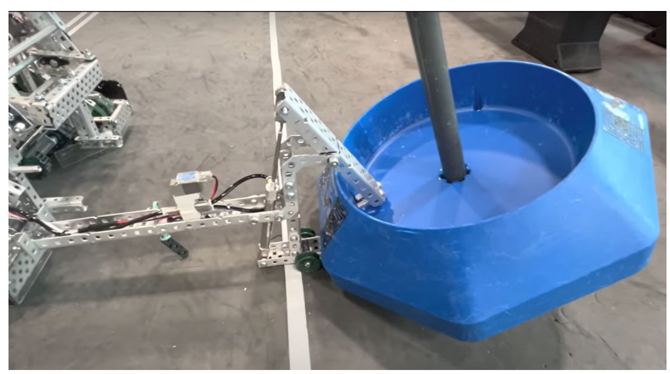
1. **210Y**



This mechanism would extend out at the start of autonomous, grab onto the goal with the claw-like structure, and pull back into the robot. It was extremely effective when used and won them every single goal rush battle.

2. **38141B**

This goal rush mechanism works a bit differently from the previous, as it doesn't retract forwards and backwards. Instead, it acts like an arm, rotating up and down into position. This robot won the world championship in Tipping Point, as the mechanism's ability to tilt the goal makes it extremely hard to grab on the other side.



3. **210X**

While this robot was not documented very much, their mechanism originated from the middle of the robot rather than the side. Like 210Y's it would extend back and forth, with the main difference being that it clamping sideways onto the pole, and not the base of the goal.

Designed by:

Adam X

Witnessed by:

Alex S

Goal Rush Mechanism

2025-01-20

▶ **Problem:** How can we build a mechanism that can beat other robots in grabbing the 5th goal?

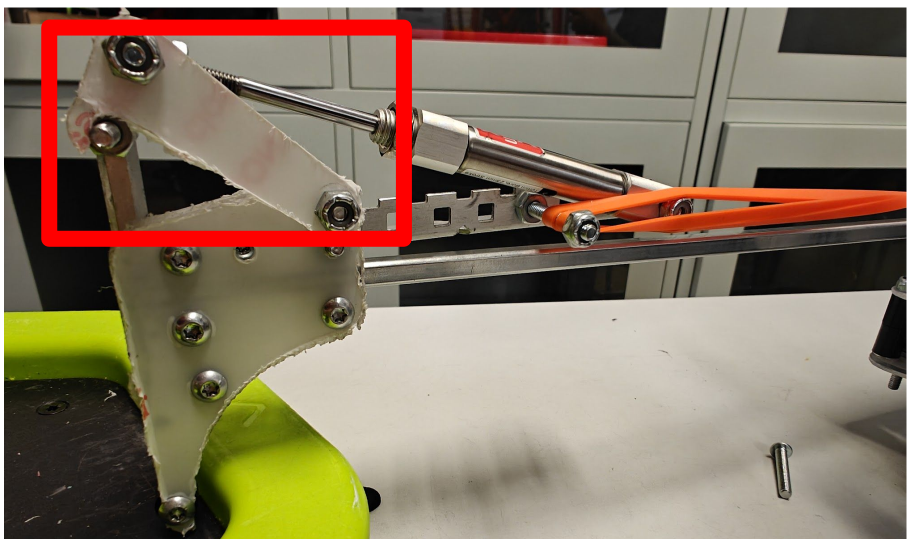
[Select a Solution]

Instead of basing our mechanism off of the solutions listed above, we decided to take our own approach to it. The shape of the Tipping Point goal was much larger than in High Stakes, so it would also take a lot of work getting those designs to work in High Stakes.

[Plan/Design Solution]

We planned to incorporate the mechanism on the arm that we already had to clear the 4-stack of rings in the corners, so that we wouldn't need to use any additional resources to build it.

Hook release for the mechanism



For this first iteration, we custom cut some lexan to fit the goal, and we noticed that we didn't account for the piston mount, which is why there is a flat piece of metal in the back. It worked by releasing the hook connected to the screw, which caused the rubber bands to pull it back.

Designed by:

Adam X

Witnessed by:

Alex S

Goal Rush Mechanism

2025-01-20

▶ **Problem:** How can we build a mechanism that can beat other robots in grabbing the 5th goal?

[Test Solution]

Testing this, we realized that it worked quite well for an initial design. There were a couple of concerns we had, especially because we were testing on our table, not the field mats

More friction on the mats.

The increased coefficient of friction from the field mats will require more force to overcome and pull the goal. This is a simple fix, as more rubber bands will create more pulling force.

[Plan/Design Solution]

On our second run of building the mechanism, we recut the lexan, this time accounting for the piston mount.

We also increased the amount of rubber bands to increase the force pulling on the goal, and the mechanism worked as intended. This mechanism could potentially allow us to always secure the 5th goal during the autonomous period, beating other teams to it, which sets us up much easier to win the match.

Designed by:

Adam X

Witnessed by:

Alex S

Tournament Analysis: Rumble in the Rockies

2025-02-02

▶ **Focus:** Analyze our performance at the Rumble in the Rockies Signature Event

Rumble In the Rockies VEX V5 Robotics Competition Signature Event

4-4-0 : 11 / 30 / 167

Qualifications Rank:	20th/53
Elims Alliance Partner	15442A
Elims Rank:	Eliminated in R-16
Driver Skills Score:	43
Programming Skills Score:	29

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
P-3	80001B, 10058A	210Z, 30134A	19 - **26**
Q-1	210Z, 1154D	11444X, 41998J	32 - **35**
Q-18	2131J, 77677X	210Z, 12125A	**41** - 26
Q-35	210Z, 15442A	10410A, 11101B	**45** - 20
Q-41	8385M, 11777C	210Z, 1908X	**31** - 13
Q-57	1281A, 89040B	210Z, 3017D	**18** - 12
Q-67	210Z, 283S	3743A, 89801B	**29** - 16
Q-80	1908B, 72711P	210Z, 8000X	23 - **26**
Q-98	210Z, 16689E	15442C, 63031Z	**35** - 25
R16 3-1	210Z, 15442A	3017D 10058A	7 - **23**

Designed by:

Adam X

Witnessed by:

Richard F

Tournament Analysis: Rumble in the Rockies

2025-02-02

▸ **Focus:** Analyze our performance at the Rumble in the Rockies Signature Event

Rumble In the Rockies VEX V5 Robotics Competition Signature Event

This tournament was our first signature event and first outside of Canada this year, so we went into it excited and ready to compete. As the only Canadian team there, everyone knew us as "that one Canadian team", and from that, we were able to make a lot of valuable connections and friends. During the tournament, we went an even 4-4-0 in qualifications, ranking 20th out of 53 teams in our division, and were selected by 5th place 15442A to form the 4th seeded alliance. Unfortunately our tournament ended there as we were upset in the round of 16. We had an extremely difficult match schedule, playing against 3 signature event winners this year and both world finalists. Skills also did not go too well as there were some minor issues both on our end with the robot and on their end with field control issues. We also ended up winning the build award because of our innovative ball bearing wheels.

Going into our next Signature Event next weekend, Mecha Mayhem, we had a few things in mind to change and improve. We learned from this experience that at signature events, getting that autonomous win point is crucial to raising your rank, even if you lost a few matches. Another thing we learned was that we lacked driver practice and knowledge on game strategy. Lastly, we realized how valuable building good connections with teams were to getting an eliminations alliance when you're a low rank.

Action items for next week are: tune our autonomous paths so they are more consistent and get more driver practice through scrimmages.

What worked:

- Overall, most things mechanisms were consistent and well working
- Ball bearing wheels worked, we got the build award ——————————>
- Goal rush mechanism worked and impressed a lot of teams.

What didn't work:

- We had a few matches where parts of the robot broke down
- Autonomous paths could be tuned more and were not very consistent
- We couldn't figure out an elevation mechanism in the hotel so we took it off

Designed by: Adam X

Witnessed by: Maxwell L

Team Scheduling - Calendar

2025-02-03

▶ **Focus:** Figure out our plans until provincials

This is a rough outline of how we wanted to allocate our time up to provincials

December and January Calendar

Sun	Mon	Tue	Wed	Thu	Fri	Sat
2	3 Now	4	5	6 Scrimmage	7 Mecha Mayhem	8 Mecha Mayhem
9 Mecha Mayhem	10	11	12 Get elevation working	13	14	15
16	17	18	19	20	21 Provincials	22 Provincials

Since we have previously competed in a tournament this weekend, most of the robot is already done and just needs some minor tuning.

Following the Mecha Mayhem 2025 Signature Event, we can focus on getting driver practice, autonomous paths, and skills.

Designed by: Adam X

Witnessed by: Kevin Z

Improving the Intake

2025-02-06

▶ **Problem:** How can we improve the intake after our performance at Rumble in the Rockies

[Identify Problems]

A consistent issue with the robot at Rumble in the Rockies was that it kept on missing rings onto the mobile goals. This problem arises from a lack of centripetal force acting on the rings, so they do not follow a circular path on the hooks and do not fall onto the stake.

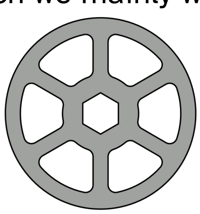
[Brainstorm Solutions]

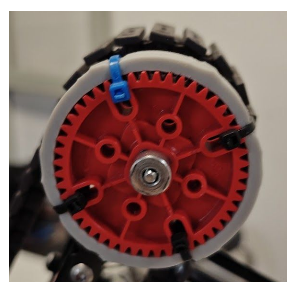
There are a few ways to solve this problem, but they all do the same thing: increase the torque/force of the hooks.

1. **Increase the flex wheel size**
   Increasing the flex wheels at the top of the intake would allow the rings to follow a more circular path at the top, therefore decreasing the likelihood of the ring unexpectedly flying out.
2. **Increase the top sprocket size**
   Increasing the top sprocket size does make that axle spin slower, but it doesn't actually change the speed that the chain is spinning at. The chain will always spin at the same speed that the driving sprocket spins, and all increasing the size at the top does is increasing its angular velocity, also increasing the force that the hooks exert on the rings.
3. **Decrease the speed of the motor**
   This compromises efficiency and speed for higher accuracy, which isn't something that we particularly want.

[Select a Solution]

We decided to go with solution 1, since that would be the easiest and we could always change it back it it doesn't work. The two other solutions require us to change something about the speed of the hooks itself, which we mainly want to stay away from.





[Plan/Design Solution]

We changed our 2-inch flex wheels into a modified wheel consisting of the outside of a flex wheel wrapped around a 48-tooth gear to slightly increase the radius.

Designed by: Adam X

Witnessed by: Richard F

Improving the Intake

2025-02-06

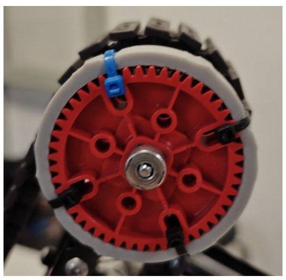
▶ **Problem:** How can we improve the intake after our performance at Rumble in the Rockies

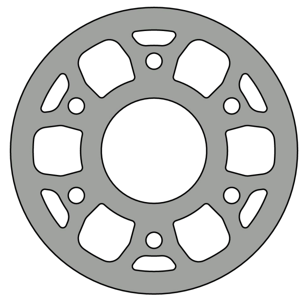
[Test Solution]

When testing out the changes we made, it worked quite well and there was a noticeable difference on the intake performance. However, we believed it could be improved even further.

[Plan/Design Solution]

So, we decided to increase the size of the wheel once again, this time to a 3-inch wheel.





[Test Solution]

This change was too drastic of a difference between the radius of each wheel, and thus caused the intake to not work at all. The 3-inch flex wheel was too large and cause the intake to completely miss this mobile stake.

[Select a Solution]

Since we couldn't increase the size of the flex wheel any further, we decide to try out the second solution of increasing the size of the top sprocket. Instead of a 12-tooth sprocket, we decided to go with a 24-tooth.

[Test Solution]

We realized this size change made is rub against the bottom of the elevation ladder, but we were still able to pass under which was fine. For the intake, there was a huge increase in how fast we could intake rings in succession, which really helped in the autonomous period where we purposefully line up 3 rings in a row to intake at once.

Designed by: Adam X

Witnessed by: Richard F

Tournament Analysis: Mecha Mayhem

2025-02-02

▶ **Focus:** Analyze our performance at the Mecha Mayhem 2025 Signature Event

Mecha Mayhem 2025 Signature Event Presented by Alberta Innovates

9-1-0 : 21 / 51 / 203

Qualifications Rank:	4th/73
Elims Alliance Partner	8198X
Elims Rank:	Eliminated in SF
Driver Skills Score:	51
Programming Skills Score:	32

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
Q-9	9543M, 98549V	210Z, 1010A	27 - **39**
Q-32	1028Z, 2088A	210Z, 5327K	31 - **28**
Q-44	210Z, 1791V	502A, 60410A	**49** - 2
Q-72	6659C, 9181G	210Z, 229V	24 - **49**
Q-91	210Z, 16610C	7842S, 45519E	**37** - 12
Q-110	210Z, 86744B	5760A, 5670F	**36** - 17
Q-119	210Z, 45519A	6508H, 27455G	**44** - 16
Q-130	210Z, 27455X	315A, 77174B	**35** - 25
Q-153	6019B, 8198X	15442C, 63031Z	38 - **39**
Q-182	221B, 68411C	210Z, 9409Y	14 - **40**
R16 7-1	210Z, 8198X	1010T, 9409Y	**44** - 21

Designed by: Adam X

Witnessed by: Richard F

Tournament Analysis: Mecha Mayhem

2025-02-10

▶ **Focus:** Analyze our performance at the Mecha Mayhem 2025 Signature Event

Mecha Mayhem 2025 Signature Event Presented by Alberta Innovates

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
QF 4-1	210Z, 8198X	3284B, 5327K	**36** - 33
SF 2-1	10B, 10W	210Z, 8198X	**34** - 37 (DQ)

This tournament went a lot better than our last, going 9-1-0 ending up in 4th place. For eliminations, we picked team 8198X to form the 3rd seeded alliance. This time, we went all the way to semi-finals, where we won that match, but unfortunately were disqualified due to an SG6 (double possession of goals) violation. Performance wise, the robot was extremely consistent each match, and we didn't have anything go majorly wrong like at Rumble in the Rockies. We set a new skills high score of 51 driver and 32 autonomous, and the robot disconnected in the last 5 seconds of the autonomous run or else it would've gotten a 37. One thing we noticed about our wall stake mechanism was that it was inconsistent during matches, and we think this is because the wall stakes in our workshop are lower than they would be on new fields at tournaments, and our mechanism is tuned to the shorter version.

The key takeaway that we gained from this experience is that **autonomous bonus is more important than possession of 3 goals**, **however**, **only** when you have a net positive point difference on the wall stakes, **which isn't hard**.

Next steps for our upcoming provincials tournament would be to coordinate eliminations autonomous and game strategy with our sister team, 210K, because that is who we plan to alliance with. We also need to remove the ring rush mechanism so we can build a better goal rush mechanism.

What worked:

- Overall, most things mechanisms were consistent and well working
- Goal rush mechanism worked and impressed a lot of teams.
- Improved autonomous routes worked very well and were consistent
- Ring rush mechanism worked every time in elims

What didn't work:

- Our improved vision of the goal rush mechanism could not be built with the current robot
- We still couldn't figure out a working elevation mechanism
- Wall stake mechanism was inconsistent.

Designed by:

Adam X

Witnessed by:

Maxwell L

Preparation and Game Strategy- Alberta Provincials

2025-02-11

▶ **Problem:** How do we want to prepare for our upcoming provincials tournament?

With our last chance to qualify for the world championships coming up in two weeks, needed to properly prepare to maximize our chances of going worlds.

There are 5 qualifying spots:

1. **Excellence Award (1)**
2. **Tournament Champions (2)**
3. **Design Award (1)**
4. **Robot Skills Champion (1)**

If a team receives more than one of the awards listed above, then the spot gets passed down to the highest ranked team in the **robot skills challenge**. This means that having a high skills score is essential to increase our chances of going to the world championship.

4 out of the 5 spots are somewhat impacted by qualification performance, and just overall match performance, meaning we want to do everything that we can to ensure that we win matches, which we should be doing regardless.

Game Strategy

Over the season, we've gradually developed our game strategy form what we've learned as well as watching other matches. In summary, we've broken it down to a few essential goals that we need to meet each game.

1. **Autonomous Bonus**
   The autonomous bonus is an additional 6 points awarded to either alliance, but it is also denying the other alliance of those 6 points, so really, it is a net swing of 12 points, making it extremely valuable.
2. **Possession of 3 Mobile Goals**
   As there are an odd amount of mobile goals, having 3 vs have 2 can make a huge difference, as a full goal scored for your alliance is worth 8 points.
3. **Wall Stakes and Negative Corners**
   Lastly, these elements are not as crucial as the first two, but regardless, can still swing the scores of a match.

Designed by: Adam X

Witnessed by: Kevin Z

Climb Design

2025-02-12

▶ **Problem:** How can we improve our climb from the past robot?

[Identify Problems]

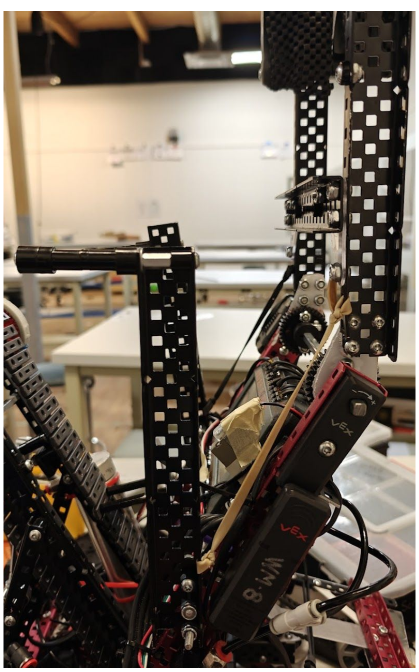
The issue we had with our climb on the past robot was that it required a piston to hang, and needed the force of the wall stake mechanism to hang. Our old robot used a piston to release a ratchet mechanism on each side allowing us to hang. However this design is not very air efficient and requires us to have enough air at the end of the match to actuate it.

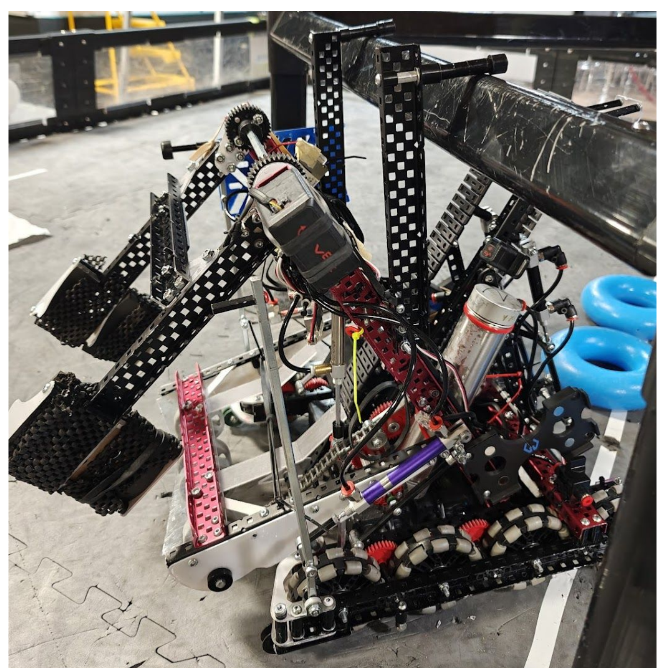
Secondly, the wallstake mechanism on the robot needed to have enough strength so that it could pull the entire weight of the robot off the ground at the end of the match to hang. If our wallstake motor was too burnt out, our robot would not hang and we would miss out of the 3 point bonus.

[Plan/Design Solution]

To avoid the case of a burnt out wallstake mechanism, we decided to have our hang rotate on a separate joint from our wallstakes. On the hang would be polycarbonate/screw-made prongs that would allow us to back into the tower and hang without the need of a full force motor to power it.

To accommodate for the lack of pneumatics on our hang, we used rubber bands and string to link our hang to the wallstake mechanism. This way we could use the wallstake to prime the hang without it being dependant on the actual strength of the wallstake mechanism and avoid using pneumatics making our hang entirely passive.





Designed by:

Alex S

Witnessed by:

Adam X

Improving the Climb

2025-02-20

▸ **Problem:** How can we improve our new climb

[Identify Problems]

Our intake ramps for the rings protrude out of the robot too much, so whenever we attempt to elevate, our robot tips over and the ramps dig into the mats, making us stuck.

[Brainstorm Solutions]

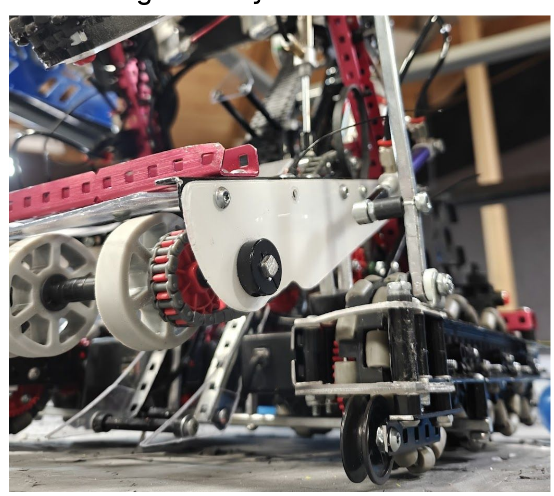
1. **Convert to an active elevation mechanism.**
   This means using a piston, motor, or other actuator to make the elevation mechanism functional. However, doing this is not resource efficient and adds on extra weight.
2. **Move the intake ramps forwards.**
   This could also be a potential solution, moving it backwards gives it more room to tip over without getting stuck. A downside of this solution is that it will most likely mess with the functionality of the intake and we would have to retune it.
3. **Build some mechanism that doesn't make us tip over.**
   Building an anti-tip mechanism could be a very strong solution as it doesn't compromise the rest of the robot and fixes the problem at the same time.

[Select a Solution]

After some thought and deliberation, we decided to build the 3rd solution, which from the descriptions above, thought was the best and most efficient solution to our problem.

[Plan/Design Solution]

Building this mechanism is extremely simple, you just need something below your drivetrain that can stop the robot from tipping too much. For this, we thought pulleys would be a good choice because they can also free spin along the ground, allowing us to spin back upright.



[Test Solution]

Testing this build, we found that it successfully solved our issue of getting stuck on out intake ramps because before we tilt that much, the anti-tip mechanism would stop us and roll us back into an upright position.

Designed by:

Adam X

Witnessed by:

Kevin Z

Tournament Analysis: Alberta Provincials

2025-02-26

▶ **Focus:** Analyze our performance at the Alberta Provincial Championship

Alberta VEX V5 Robotics Competition Provincial Championships (HS) presented by Syncrude

7-0-1 : 21 / 51 / 203

Qualifications Rank:	1st/48
Elims Alliance Partner	210K
Elims Rank:	Tournament Finalists
Driver Skills Score:	55
Programming Skills Score:	35

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
P-6	3300E, 9409Y	210Z, 3388H	28 - **29**
Q-9	5760A, 9409X	210Z, 5760E	27 - **34**
Q-18	210Z, 45519A	45519F, 99197W	**33** - 20
Q-24	2088G, 3388H	210Z, 99197A	21 - **33**
Q-38	210Z, 415519D	3388K, 86744B	**42** - 21
Q-50	210Z, 3388Z	221W, 3388C	**28** - 17
Q-57	3388A, 45519C	210Z, 210K	13 - **53**
Q-62	221X, 45519H	210Z, 221A	35 - 35
Q-72	210Z, 221B	2088A, 45519B	**37** - 0
R16 1-1	210Z, 210K	3388Z, 45519E	**42** - 13
QF 1-1	210Z, 210K	2088A, 3388K	**35** - 3

Designed by:

Adam X

Witnessed by:

Richard F

Tournament Analysis: Alberta Provincials

2025-02-26

▸ **Focus:** Analyze our performance at the Alberta Provincial Championship

Alberta VEX V5 Robotics Competition Provincial Championships (HS) presented by Syncrude

Match # (winner is highlighted)	**Red Alliance**	**Blue Alliance**	**Score**
SF 1-1	210Z, 210K	5760A, 5760E	**45** - 28
F 1-1	210Z, 210K	221X, 3388S	**38** - 30
F 1-2	210Z, 210K	221X, 3388S	15 - **31**
F 1-3	210Z, 210K	221X, 3388S	29 - **32**

We started this weekend out strong, going 7-0-1 after qualifications and placing first. We chose our sister team 210K for an alliance and lost in the finals. Overall this tournament did not go as planned and we didn't win any awards, and barely qualified for the world championship through skills, which also didn't go well. Our skills code started running differently when using the VEX V5 smart field control than when we regularly tested it, so none of our runs hit so we ended up with a 35 programing and 55 driver, securing us 4th.

This entire tournament was a huge mess and hopefully we can perform at the world championship and really show the world what we can do. From now on out, we have 2.5 months until we fly to Dallas, which is quite a lot of time to build, program and practice. The vision until we get there is to really think and plan out what we want our robot to be more than anything else.

What worked:

- Again, most mechanisms were consistent enough to get assist us in going undefeated in qualifications.
- Elevation was extremely smooth and consistent.

What didn't work:

- Our 2-ring wall stake mechanism was breaking down over time and we though that it was best to use it as a single-ring.
- Intake still occasionally flung out rings.

Designed by: Adam X

Witnessed by: Maxwell L

210Z ECLIPSE

Programming Notebook

Calgary, Alberta, Canada

Start Date: May 5th, 2024 **End Date:** TBD **Book:** 1 of 1

Table of Contents

2024-05-05 to 2024-10-20

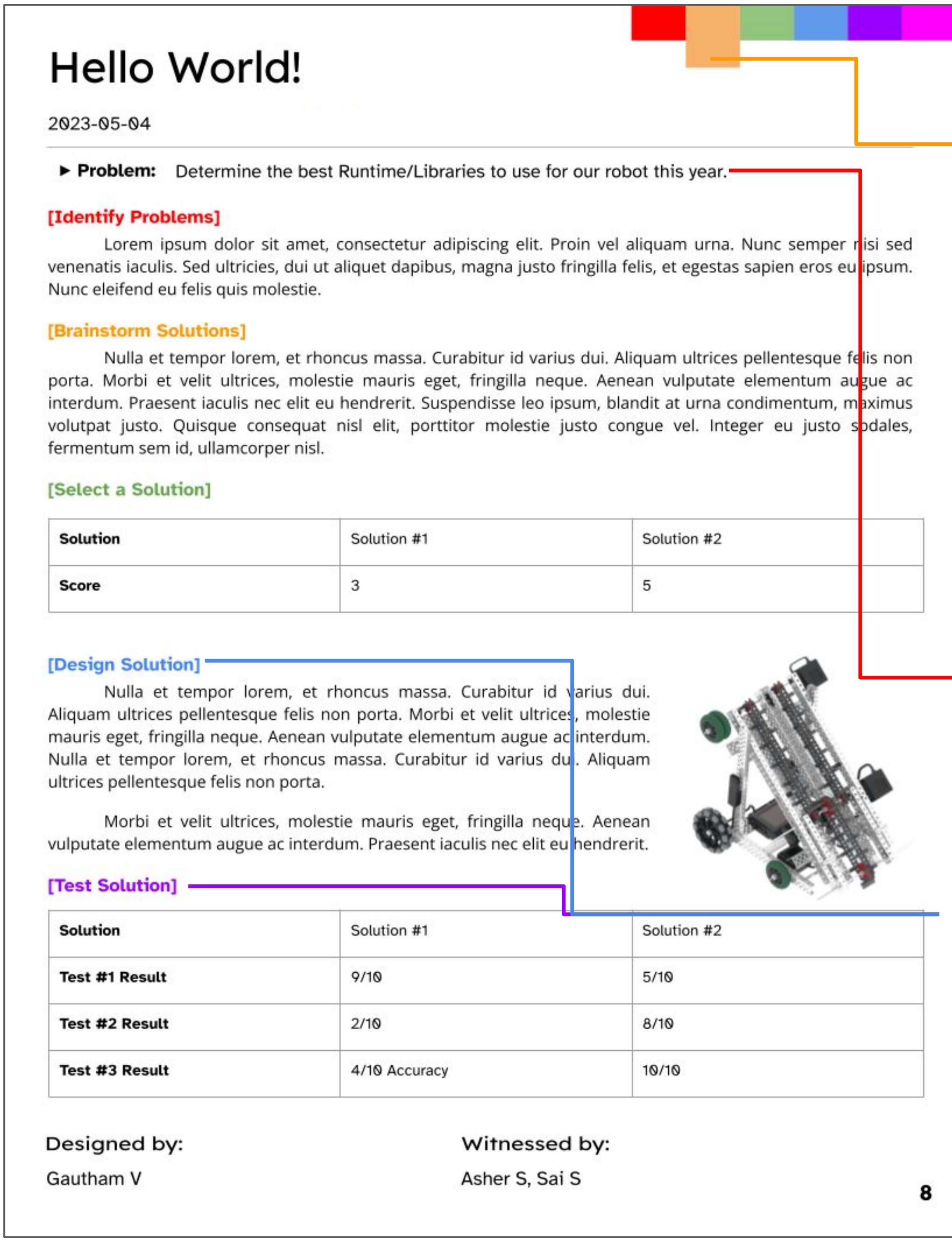
Entries regarding **Control Flow** are highlighted in **red.**
Entries regarding **Codebase Organization** are highlighted in **orange.**
Entries regarding **Motion Control** are highlighted in **green.**
Entries regarding **Operator Control** are highlighted in **blue.**
Miscellaneous Entries as highlighted in **pink**.

////////////////// ***2024-04-27: VEX Over Under announced to the public.*** //////////////////

Sample Page Overview

2024-05-05

Here is a sample page detailing how this notebook will be formatted, including the robot subsystem tags from the previous page.



Hello World!

2023-05-04

► **Problem:** Determine the best Runtime/Libraries to use for our robot this year.

[Identify Problems]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel aliquam urna. Nunc semper nisi sed venenatis iaculis. Sed ultricies, dui ut aliquet dapibus, magna justo fringilla felis, et egestas sapien eros eu ipsum. Nunc eleifend eu felis quis molestie.

[Brainstorm Solutions]

Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta. Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Praesent iaculis nec elit eu hendrerit. Suspendisse leo ipsum, blandit at urna condimentum, maximus volutpat justo. Quisque consequat nisl elit, porttitor molestie justo congue vel. Integer eu justo sodales, fermentum sem id, ullamcorper nisl.

[Select a Solution]

Solution	Solution #1	Solution #2
Score	3	5

[Design Solution]

Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta. Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Nulla et tempor lorem, et rhoncus massa. Curabitur id varius dui. Aliquam ultrices pellentesque felis non porta.

Morbi et velit ultrices, molestie mauris eget, fringilla neque. Aenean vulputate elementum augue ac interdum. Praesent iaculis nec elit eu hendrerit.

[Test Solution]

Solution	Solution #1	Solution #2
Test #1 Result	9/10	5/10
Test #2 Result	2/10	8/10
Test #3 Result	4/10 Accuracy	10/10

Designed by: Gautham V

Witnessed by: Asher S, Sai S

8

The flag at the top shows which category this page is a part of. See the table of contents for a list of categories.

The **problem** we're currently facing is identified at the top to remind readers and ourselves exactly what we are currently doing.

Tags are on each page of the notebook, so the design process is evident throughout the flow of the notebook.

Designed by:

Adam X

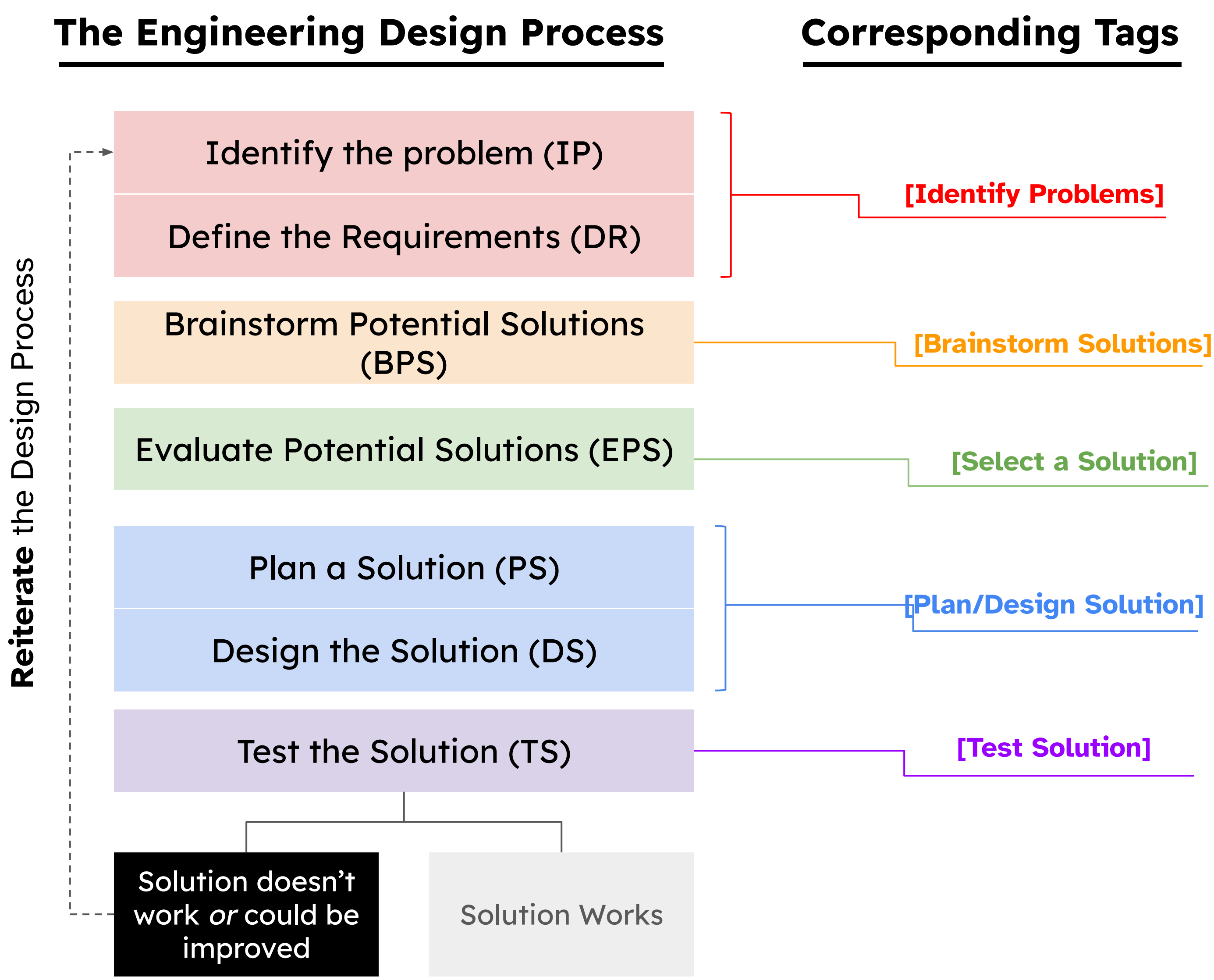
Witnessed by:

Kevin Z

The Engineering Design Process

2024-05-05

The Engineering Design Process is a key component to the success of any robot. Using this method, our team is able to efficiently cover all of our solutions and decide on the design best fit for our robot. The **process** is shown on the *left*, and the corresponding **tags** used on each page of notebook are shown on the *right*.



Designed by: Adam X

Witnessed by: Alex S

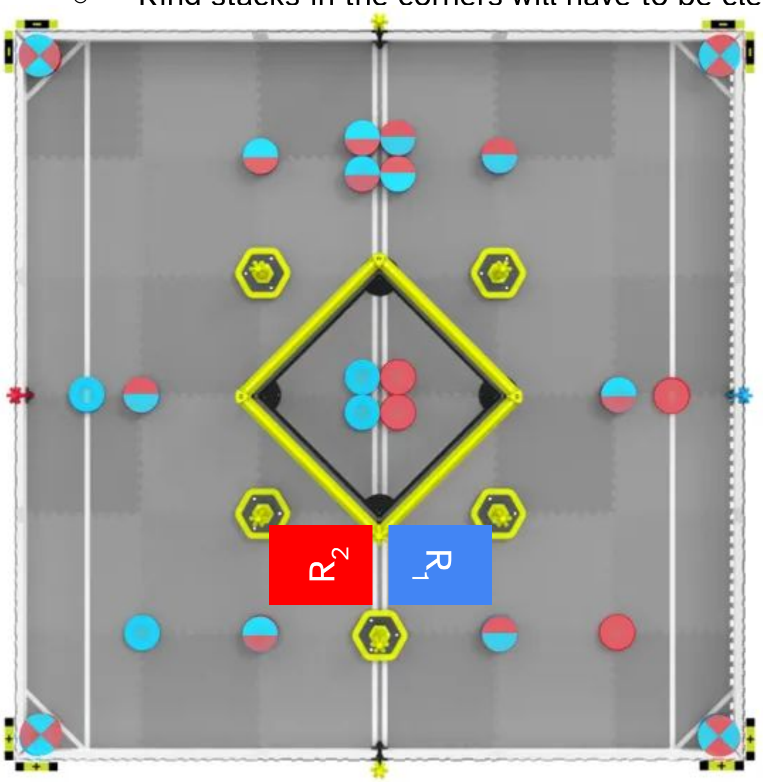
Game Breakdown and Analysis

2024-08-31 | **Miscellaneous**

Since we've already covered the game from a building standpoint in our other notebook, we wanted to cover it from a programming standpoint here. We started by identifying some problems and requirements this game may bring upon us. The following few pages will contain overarching problems we attempt to address **throughout the season.**

[Identify Problems]

- Autonomous Routines during Matches may **interfere** with other robots, which could lead to the risk of entanglement or our autons getting messed up due to outside factors
  - Sensors may be required to **detect distance** from rings or mobile goals, or to detect whether they have already been taken by opposing robots
  - Vision sensors could be used to align with the centers of rings and mobile, helping the robot intake more effectively.
  - Color sensors might be needed to sort and filter out the opposing alliance's colored rings
  - Alliance colored rings will be difficult to intake from stacks of rings
  - Ring stacks in the corners will have to be cleared to double/negate goals



Possible situation where robots interfere and interact with each other in the autonomous phase. This is a possible interaction where R_2 and R_1 are bother trying to obtain the mobile goal

The VEX V5 Rotation Sensor, which can be helpful for wheels and subsystems with precise movements

The VEX V5 Distance Sensor, for finding distances to objects

Designed by:

Adam X

Witnessed by:

Alex S

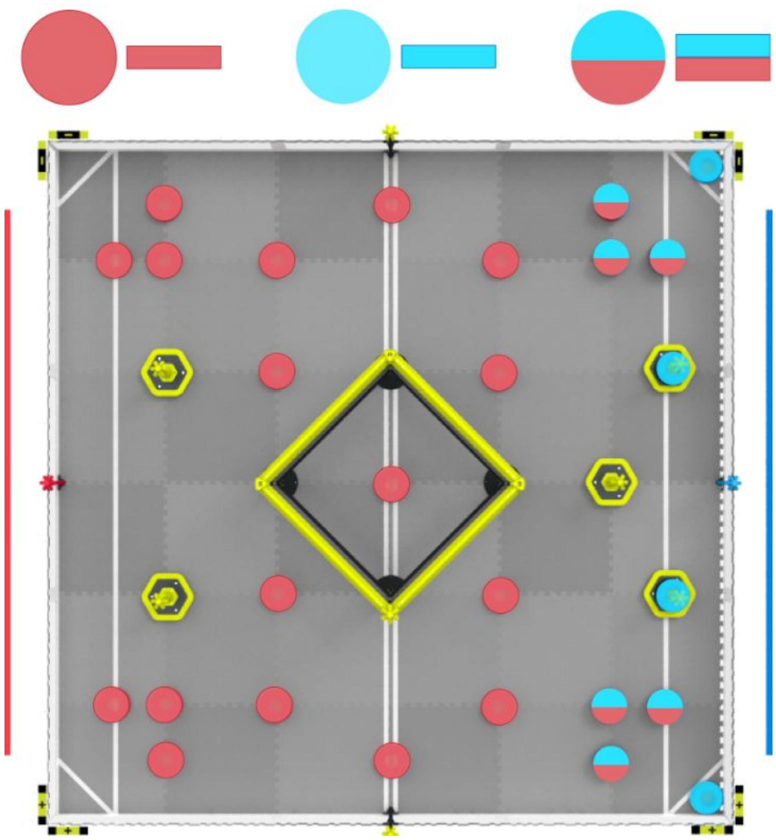
Game Breakdown and Analysis

2024-08-31 | **Miscellaneous**

We also realized the programming skills runs in this game would garner quite a bit of complexity, and a bit of luck to get through. Here are some problems we brainstormed regarding Programming Skills.

[Identify Problems]

- The field is extremely open, an will require precise tracking
- A mobile goal in any corner is 5 points
- Blue rings **only count once all red rings are scored**
  - In addition to that rule, blue rings can **only count as top rings**
  - These two rules make it so that you need to focus on the red rings only,
- Looking at the stacks of rings, they will be very hard to consistently intake the red rings each time.



Skills field orientation

The VEX V5 optical sensor, which can be used to filter out the blue rings when intaking

Designed by: Adam X

Witnessed by: Alex S

Choosing A Programming Runtime

2023-09-02 | **Codebase Organization**

▶ **Problem:** Determine the best Runtime/Libraries to use for our robot this year.

Over its years, VEX has seen dozens of runtimes, libraries, and tools to assist programmers in creating their robot's codebase. This year, we're faced with the decision of which runtime to use.

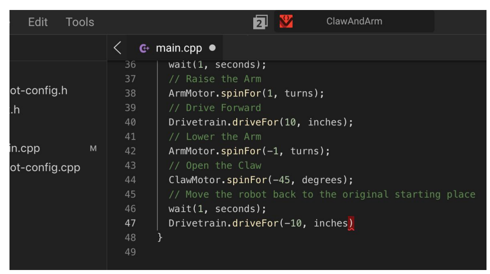
[Identify Problems]: Find the best programming runtime and compatible libraries to use for our robot this year. The runtime we choose should be:

- Versatile and simple, but effective in operation
- Used within a modern code editor, such as Visual Studio Code
- Have enough supporting libraries and community usage
- Clearly Documented, in an easy-to-understand format.

[Brainstorm Solutions]

A

The first possible solution is VEX's official programming solution: **VEXCode V5**. Since it is officially made by VEX, it is most definitely reliable and most likely bug-free, and likely boasts better system integration, allowing for better performance. That being said, it is also notorious for having subpar documentation, and isn't widely used by experienced teams



B

Our second option, which we used last year, is **VEX PROS**, an open source library developed by Purdue Sigbots' BLRS VEXU Team. It boasts direct integration into Visual Studio Code, and a simplistic and versatile API that makes intuitive sense. However, the documentation and ability to learn may be lacking. While it is the top choice for most advanced teams, given it's not official, it may need frequent kernel updates.

PROS

C

Our final option, a new and emerging option, is **vex_rt**, an emerging VEX brain kernel build with rust. Iis blazingly fast due to its method of build, and allows the use of Rust as a programming language, abstracting a significant amount of inconsistencies seen with C++. But since it is still in development, the releases are not stable, and the API is lacking.

Crate vex_rt

Designed by:
Adam X

Witnessed by:
Maxwell L

Choosing A Programming Runtime, Cnt'd

2024-09-02 | **Codebase Organization**

▶ **Problem:** Determine the best Runtime/Libraries to use for our robot this year.

[Select/Plan a Solution]

We evaluated the following aspects of each brainstormed solution in the following decision matrix.

- **Ease of Development/DevEx (/8):** How easy is this runtime to work with? Does it work easily in a well-known IDE? Is there adequate tooling and intellisense for it?
- **Documentation (/5):** How well is this runtime documented? Does the API documentation contain frequent examples and explanations?
- **API Abilities (/8):** Is the API for this runtime exhaustive? Does it cover all the features we'll be needing this season?
- **External Libraries (/5):** Does this runtime have good external libraries for us to use?
- **Community Usage (/4):** How many others in the VRC Community use this? Could we ask them if we get stuck or need help?

Choosing a Programming Runtime: Decision Matrix

	A. VEXCode V5	**B. VEX PROS**	C. vex_rt
Ease of Development (/8)	5	**8**	2
Documentation (/5)	2	**5**	2
API Abilities (/8)	7	**7**	4
External Libraries (/5)	3	**4**	0
Community Usage (/4)	3	**4**	1
TOTAL	20	**28**	9

As seen from above, we landed on **VEX PROS** as the best choice to use as a programming runtime this season. It beats its competitors in nearly every other category, and we've been using it for the past 3 years. Another reason why we chose PROS is because Vexcode V5 updates around 20ms, while PROS updates at approximately 10ms, meaning it updates faster and thus gives more accurate readings from sensors, motors, and the radio. Although it seems small, the difference is huge and can help us in matches. It's like if you have two GPS navigation apps, and one updates every 10 seconds and the other every 20 seconds. The first one will update and receive information twice as fast and as frequently than the second one.

Designed by: Adam X

Witnessed by: Kevin Z

Figuring Out our Codebase File Structure

2024-09-04| **Codebase Organization**

▸ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Identify Problems]

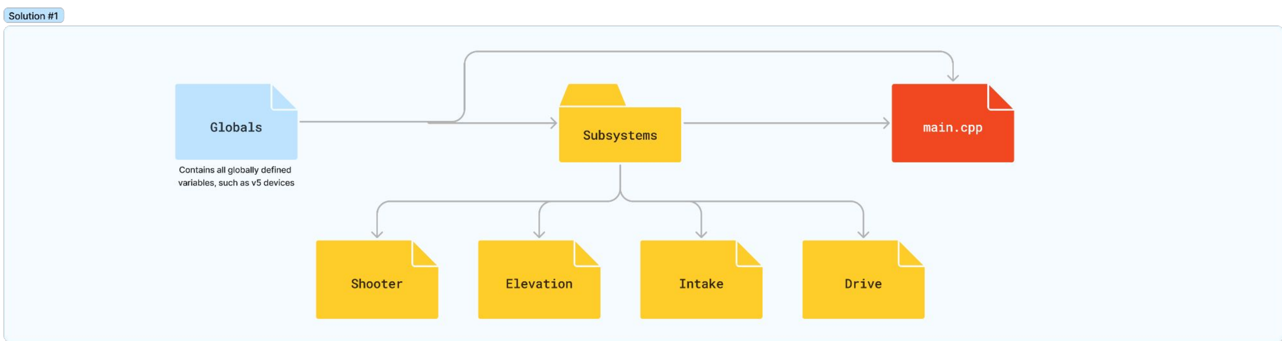
Last year, our entire codebase was split up into just eight files. Largely, this was problematic because finding certain functions in massive 400-line files was extraordinarily tedious, and modifications took quite long to make. Here are some of the requirements we need to fulfill with this codebase:

- Should be easy to find, modify, or delete functionality on-the-fly
- Should avoid egregious usage of global state, which generally decreases code clarity
- Should make use of multiple files and folders to organize like code

[Brainstorm Solutions]

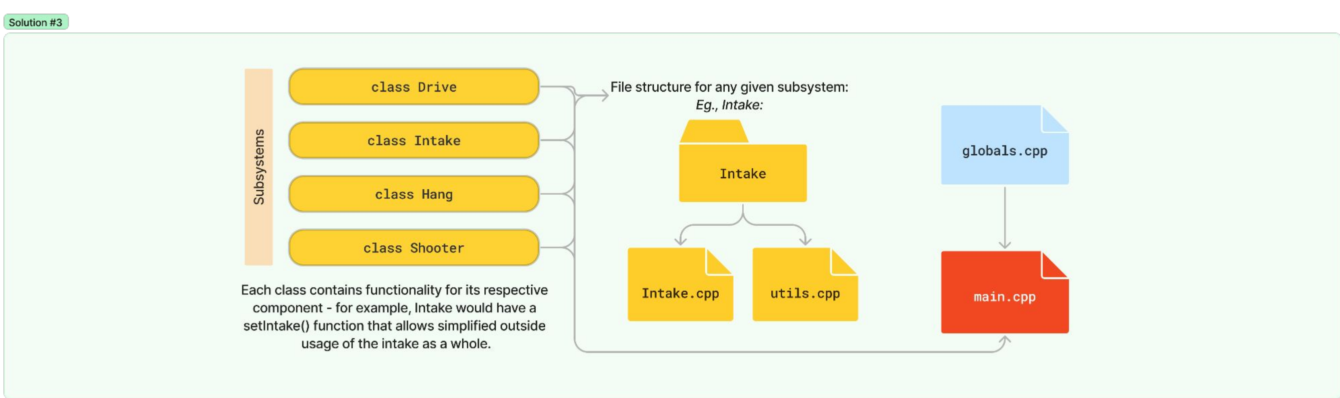
#1

The first idea we had for codebase organization we had was to organize into multiple different files for: globals, and one for each subsystem. This structure is diagrammed below:



#2

Our second idea was to use **Object Oriented Programming (OOP)** to use globally defined V5 devices (eg., a motor) to create classes for each subsystem. This structure is diagrammed below:



Designed by:
Adam X

Witnessed by:
Kevin Z

Figuring Out our Codebase File Structure Cnt'd

2024-09-04| **Codebase Organization**

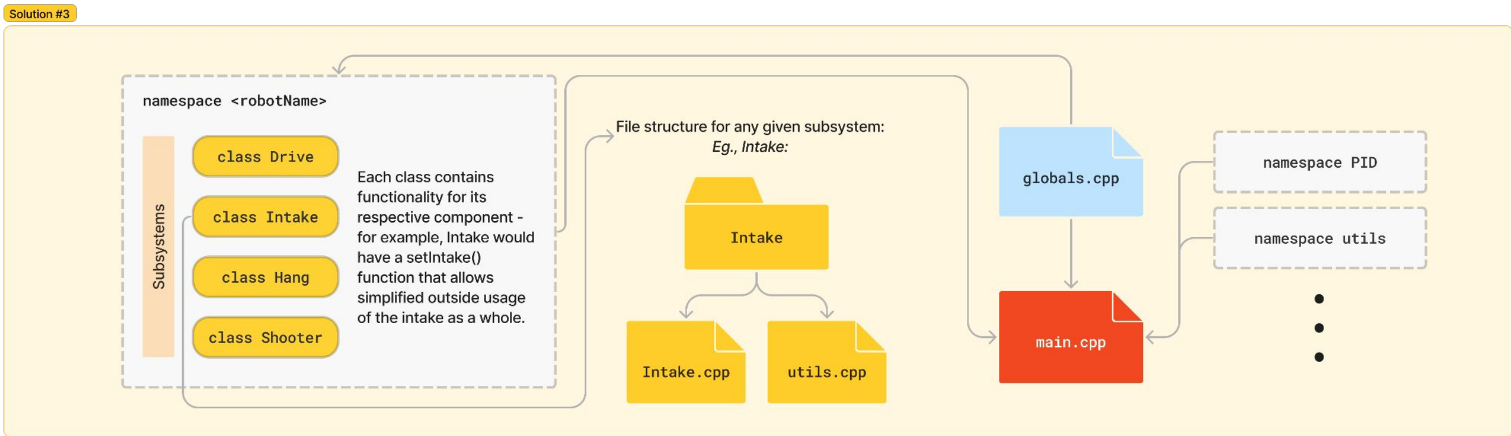
▶ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Brainstorm Solutions]

#3

Our third idea was quite similar to our second idea - but rather than just organizing the subsystem classes in a single file, we would put all of our subsystems in a **namespace** - and we'd also create classes within the namespace for other major 'sections' of our codebase, such as PID or utilities, which we will implement later on. Here is an outline of such a structure:

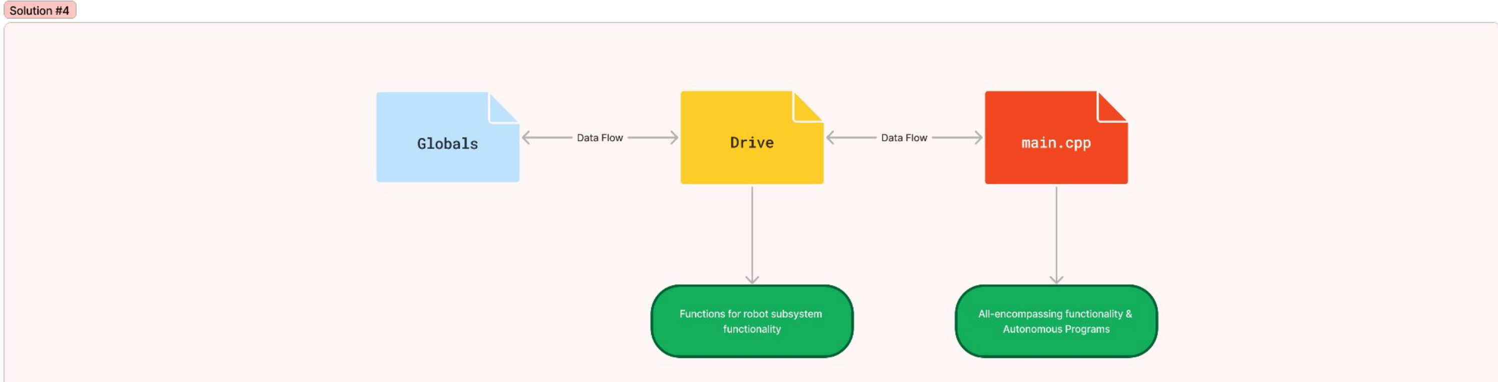
Solution #3



#4

Our final idea was to keep things as simple as possible, and focus on a functional approach rather an a OOP-based approach. This would likely have everything in just a few files, but would likely save us quite a bit of time with headers and cross-file communication.

Solution #4



Designed by: Adam X

Witnessed by: Kevin Z

Figuring Out our Codebase File Structure

2023-09-04 | **Codebase Organization**

▶ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Select a Solution]

We evaluated the four options we brainstormed by listing their pros and cons, and decided as a team what the optimal structure should be, based on those pros/cons.

	Pros	Cons
#1: Simplistic Multifiling	• Cross-file communication is simple, and headers don't have to be overbearing	• Certain files may contain a lot of functionality, making things hard to find • File names don't define functionality • Functionality is not compartmentalized • May require lots of global state
#2: OOP	• Functionality is compartmentalized, so systems are unlikely to interact with each other unintentionally • Object Instantiation of each class makes code extremely readable and intuitive • Functionality is organized by file - so finding/modifying it becomes really easy • Doesn't require *as much* global state	• OOP generally requires long and extensive header files, so adding large functionality could be tedious • Compile times would be higher • Still no division of 'sections' of code, such as a PID system from the robot subsystems
#3: OOP with Namespaces	• **All of the advantages from above still apply** • Code is sectioned by namespace, so adding functionality would be more organized • Cross-library naming conflicts prevented	• OOP generally requires long and extensive header files, so adding large functionality could be tedious • Compile times would be higher
#4: Hyper-simplistic	• No need for extensive header files • Compile times may be shorter due to a small number of files to compile	• Lots of global state needed • Requires lots of comments, not very intuitive • Long files are hard to navigate - so changes can easily create unintended side effects • Functionality lacks compartmentalization

Since **#3: OOP with Namespaces** had the most significant pro-to-con ratio, and overall seemed the most appealing, we decided to organize our code according to structure defined by solution #3.

Designed by:
Adam X

Witnessed by:
Kevin Z

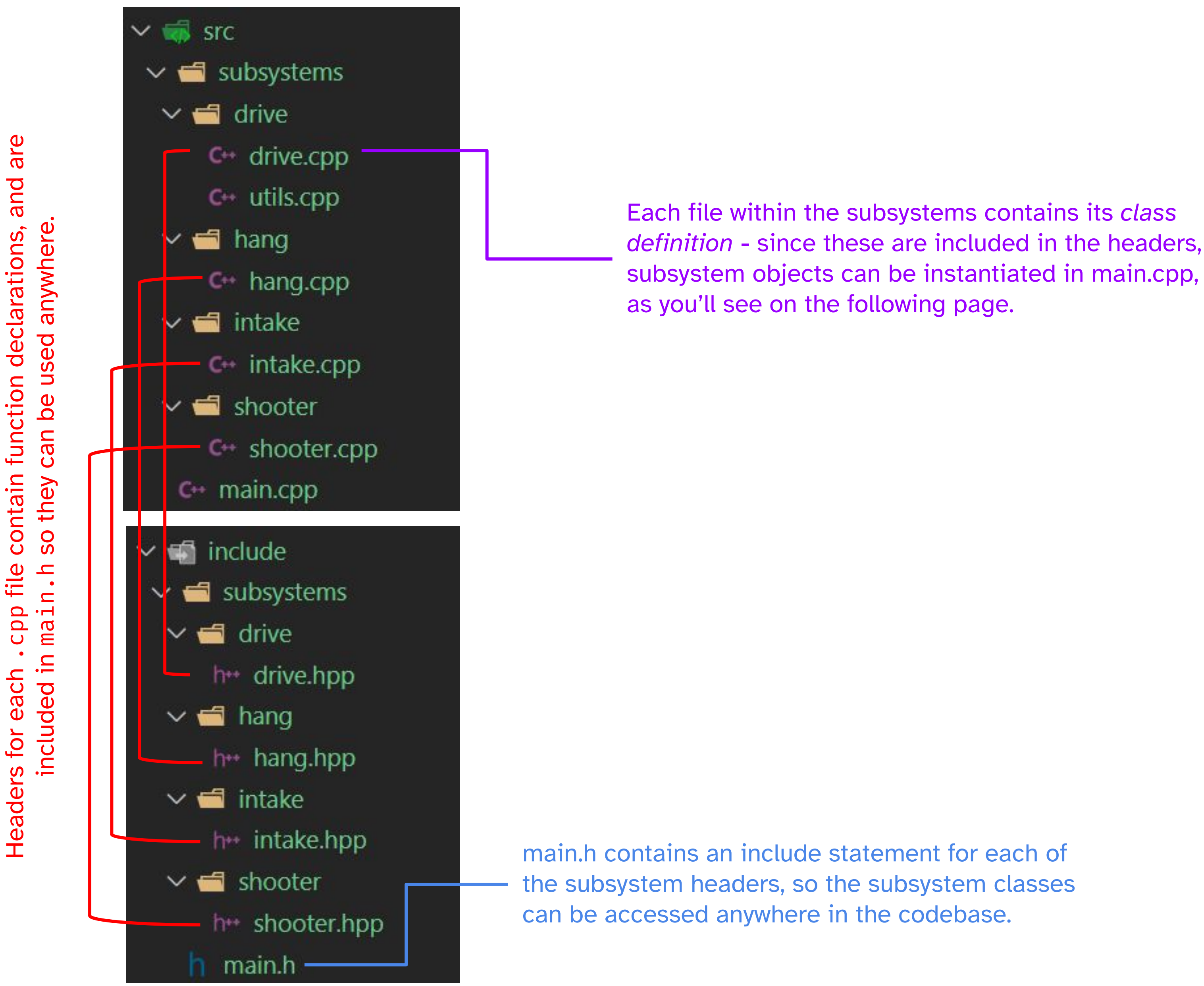
Figuring Out our Codebase File Structure

2024-09-07 | **Codebase Organization**

▶ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Plan a Solution] & **[Program Solution]**

Based on the structural diagram we made while brainstorming (page 9), we created the following file structure. Each subsystem has a <subsystemName>.cpp file as well as a <subsystemName>.hpp file, to separate the smaller utility functions that directly interact with the motors from the larger functions that use said utility functions. The next page will move from file structure to the the control flow system, but is still part of this design cycle.



Designed by:
Adam X

Witnessed by:
Alex S

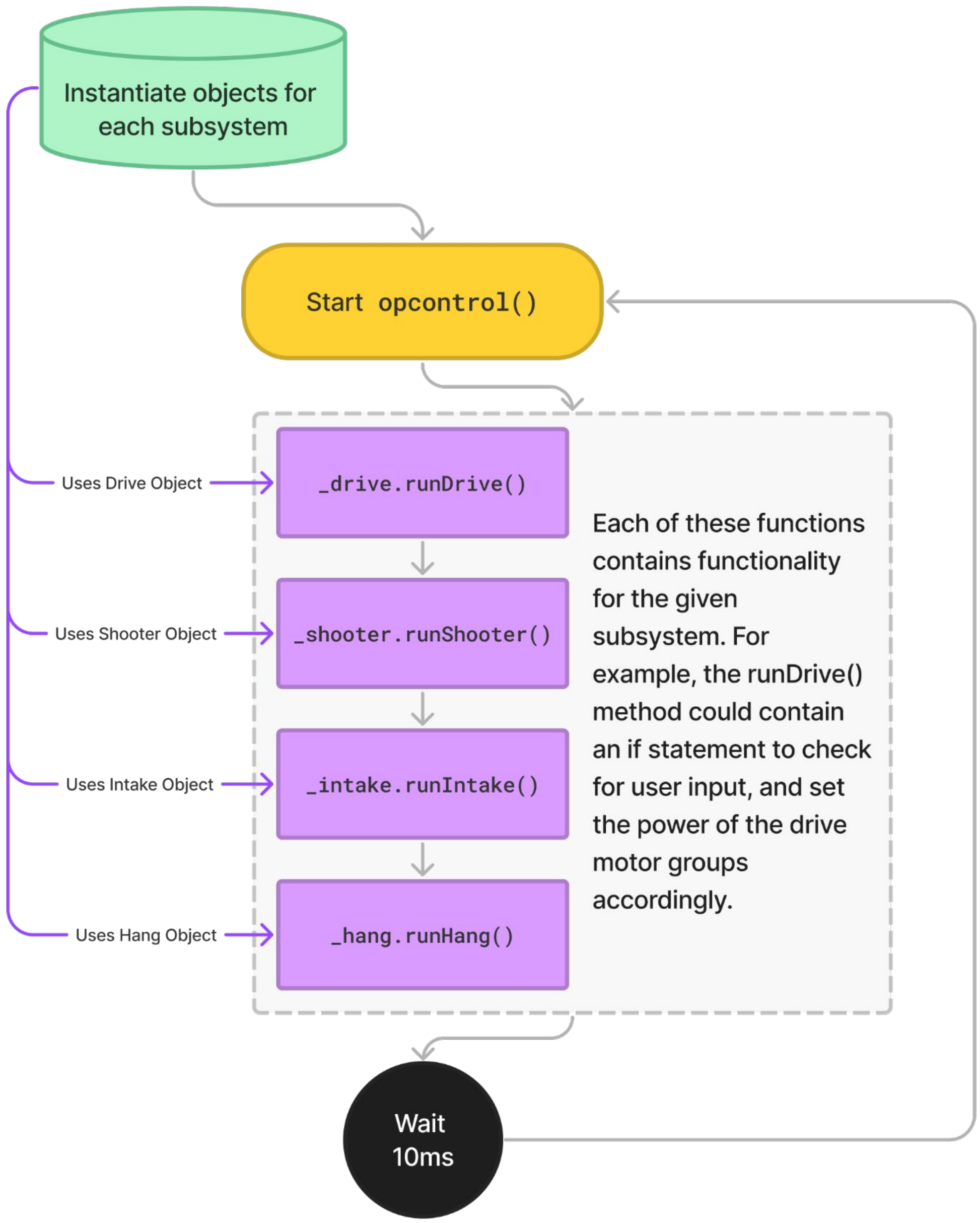
Managing Codebase Control Flow

2024-09-07 | **Control Systems**

▶ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Plan a Solution]

Continued from the previous page, we wanted to establish a control flow for our robot. All of the driver-controlled code operates within the `opcontrol()` function in `main.cpp` - inside that, we intend to set up a host of `run<subsystem>()` functions that encapsulate all driver functionality for each component within said function. We started by diagramming this with a control-flow flowchart, as shown below:



Designed by: Adam X

Witnessed by: Alex S

Managing Codebase Control Flow

2024-09-07 | **Control Systems**

▶ **Problem:** What is an effective and easy-to-modify file structure for our codebase?

[Program Solution]

Using the flowchart from the previous page, we wrote up the ‘boilerplate’ code we’ve inserted into each of the subsystem files. Below, the code for a sample `Shooter` class is shown, but the structure for each of the subsystem files from the previous page is largely identical.

shooter.cpp

```cpp
void thunderbird::Shooter::toggleShooter() {
    // Code to toggle the shooter, or something else
}
// Other utility methods would also be defined here


// This method handles central functionality,
// so it is included within the global loop in main.cpp
void thunderbird::Shooter::runShooter() {

}
```

shooter.hpp

```cpp
namespace thunderbird {
    class Shooter {
        private:
        // Private Class Members
            bool shooterIsActive = false;
            // Etc ...

        public:
        // Public Utility Methods
            void toggleShooter();
            void turnOnShooter(); // Etc ...

        // Public run<subSystem> Method - used in the main.cpp loop
            void runShooter();

    };
}
```

Headers for each `.cpp` file contain function declarations, and are included in `main.h` so they can be used anywhere.

main.cpp

```cpp
// _<subsystem> naming convention to represent the ONLY object for each subsytem
thunderbird::Drive _drive = thunderbird::Drive();
thunderbird::Shooter _shooter = thunderbird::Shooter();

void opcontrol() {
    while(true) {

        // Main loop functions
        _drive.joystickL(); // Driver controls in teleop
        _shooter.runShooter(); // Run the shooter

        pros::delay(milliseconds: 10);
    }
}
```

This function then gets used within the `opcontrol()` function, which gets invoked by the brain.

The classes defined for each are instantiated here, so the created objects can be used anywhere in `main.cpp`.

Designed by:
Adam X

Witnessed by:
Alex S

Cloud Integration: Using Github

2024-10-16 | **Codebase Organization**

▶ **Problem:** How can we use version control to manage our codebase & make changes quickly?

[Identify Problems]:

Last season, we stored all of our code locally (and using git locally) and made relatively infrequent commits to our codebase. When we went backwards to check through changes or view past implementations, we couldn't find out exactly what we had done. This season, we intend to fix this.

[Brainstorm Solutions] & [Select a Solution]

Possible Solution	**Github**	**GitLab**	**External Storage, such as a USB or on Google Drive**
Evaluation (Pros)	- World's most popular version control system - Gautham has used it in the past, so there is no learning curve - Data stored in the cloud - can be accessed anywhere - Allows for easy collaboration, so it can be stored on Asher/Sai's computers as a backup - Directly integrated within VS Code **5 Points**	- Data stored in the cloud - can be accessed anywhere - Allows for easy collaboration, so it can be stored on Asher/Sai's computers as a backup - Integrated CI/CD tools to make development potentially easier. **3 Points**	- Simple; requires no configuration **1 Point**
Evaluation (Cons)	- Have to learn/use git commands, and can be fatal if mistakes are made **1 Point**	- Have to learn/use git commands, and can be fatal if mistakes are made - Need to learn a whole new version control system **2 Points**	- Not a proper version control system - requires a lot of manual work - May not be universally accessible from any location **2 Points**
Score	5 - 1 = **4 Points**	3 - 2 = 1 Point	1 - 2 = -1 Point

Based on the above decision matrix, we decided to use **Github** as our cloud version control solution. It has built-in support for VS Code, is used by nearly everyone in the industry, and has excellent extensibility and tooling support.

Designed by: Adam X

Witnessed by: Richard F

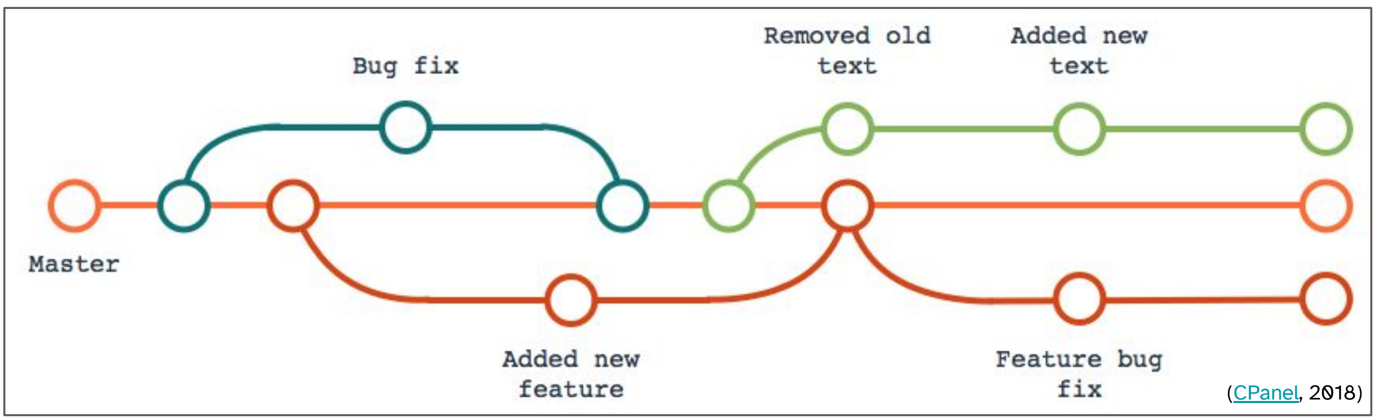
Cloud Integration: Using Github

2024-10-16 | **Codebase Organization**

▶ **Problem:** How can we use version control to manage our codebase & make changes quickly?

[Plan a Solution] & **[Program Solution]**

Now that we had decided on using Github, it as a matter of understanding how to use it, and using it effectively. The first thing to understand about github is the concept of version control, and branching. *As shown below*, the codebase can be imagined as a tree-like structure; each time we want to add a feature, we do it in a **branch** first (so it doesn't interfere with mainstream functionality), and them **merge** that side branch into the main branch, putting our code back together.



(CPanel, 2018)

Here are a list of important **git commands** we're planning to use throughout the season, and will be used to properly version control our codebase.

```
>> git stage .
```

→ 'Stage' all files to be committed, ie., tell git you are about to commit them to the repository

```
>> git commit .
```

→ Actually commit any staged files to the cloud repository

```
>> git branch <name>
```

→ Create a new git branch with the provided name, which will be a fork of the branch you are currently working on

```
>> git checkout <name>
```

→ Go and view any branch, ie., "check it out"

```
>> git merge <name>
```

→ Merge the branch provided with the main branch

Designed by:
Adam X

Witnessed by:
Richard F

Programmatic Analysis: Brake Modes

2024-10-20 | **Hardware Operation**

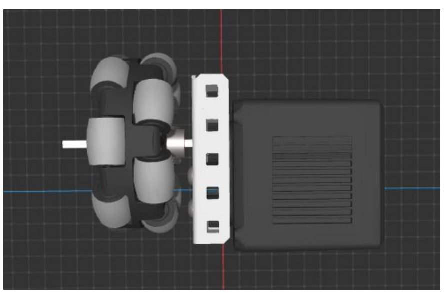
▶ **Problem:** How does the `brake_mode` of a motor affect its operation?

[Identify Problems]

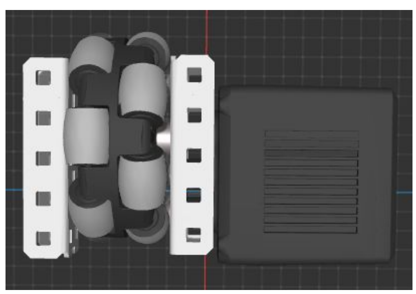
While we wait for enough parts to build our drivebase, we wanted to conduct some tests on motors and the brake modes offered by PROS. Last season, we ran everything on E_BRAKE_COAST, which simply allows motors to coast in either direction when power is not applied. This year, we wanted to investigate the other modes and see if they made a difference and could help us. Here are the requirements we identified:

- How much power can the mode E_BRAKE_HOLD resist?
- Which mode is most power-efficient, and which modes are most prone to overheating?
- Could any of these alternate modes be more effective in the driver-controlled period?

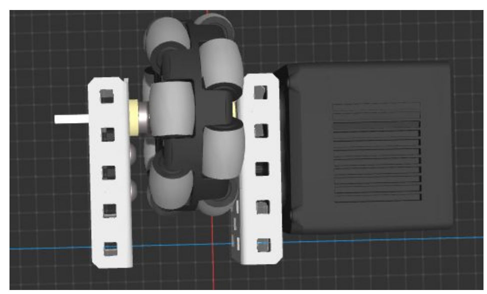
[Select a Solution]: We wanted to test the motor as if it were in a situation when being used on our robot. Mainly, this meant that it should not be cantilevered, and the drive axle shouldn't have egregious friction.



Won't work - wheel is cantilevered



Won't work - too much friction



May work, but might need extra bracing

[Design Solution]

To do this, we designed the following apparatus which resembles a small section of a drive pod. It has C-Channels on both sides to avoid cantilevering the wheel axle, and has a motor on one side to power the apparatus.

We also needed to brace the top using a 5x2 C-Channel so the apparatus wouldn't fall apart while testing.

Designed by:

Adam X

Witnessed by:

Maxwell L

Programmatic Analysis: Brake Modes

2024-10-20 | **Hardware Operation**

▶ **Problem:** How does the `brake_mode` of a motor affect its operation?

[Test Solution]

Based on the requirements we defined on the previous page, we organized the results of our testing into the following table. Here are the tests we conducted on the motor apparatus:

1. **Qualitative Analysis:** What does the given brake do, from a qualitative perspective?
2. **Burn-out Time:** When we band the wheel forward with four #32 bands, and force it to hold its position, how long does it take to overheat and give out? **[Average of 5 Trials]**
3. **Coast Distance:** After the motor goes from max power for 5 seconds to calling its `brake()` method, how far does the wheel move before coming to a stop? **[Average of 5 Trials]**

Test Results: Brake Mode **Hold**

Test Type	#1: Qualitative Analysis	#2: Burn-out Time	#3: Coast Distance
Results	When the `brake()` method is called, the motor immediately stops and uses power to hold its position. It was *noticeably difficult* to move the wheel in either direction.	65.2s	N/A - Hold does not allow for wheel slip

Test Results: Brake Mode **Brake**

Test Type	#1: Qualitative Analysis	#2: Burn-out Time	#3: Coast Distance
Results	Similar to the **hold** mode, the motor comes to a stop immediately. But by contrast, it only does this momentarily, and can be rotated in either direction with significantly lower effort.	61.4s	N/A - Hold does not allow for wheel slip

Test Results: Brake Mode **Coast**

Test Type	#1: Qualitative Analysis	#2: Burn-out Time	#3: Coast Distance
Results	When the `brake()` method is called, the motor immediately stops and uses power to hold its position. It was *noticeably difficult* to move the wheel in either direction.	N/A - no extra force applied by the motor, so it can't burn out	~260° (Could be a result of friction and built apparatus though)

While the above results are not conclusive in providing a single 'best' brake mode, it demonstrates that all three modes could be useful in certain applications. For example, in Auton, where there should be minimal motor coasting, **HOLD** may be better.

Designed by: Adam X

Witnessed by: Maxwell L

Driver Control: Chassis Movement

2024-10-20 | **Operator-Controlled**

▶ **Problem:** Find the best control-scheme for the chassis, as well as what our driver prefers most.

[Identify Problems]: What is the most effective, easy-to-use, and versatile control scheme for our driver? The following are the exact requirements we identified:

- Which control scheme (arcade, tank, single-stick, etc.) does our driver feel most at-home with?
- Which control scheme provides the most versatility and maneuverability for our driver?
- Are there certain control schemes with limitations, and should be avoid them as such?

[Brainstorm Solutions]: Here are three possible control schemes we believed we could use.

A

The first control scheme we wanted to try were **arcade controls**. In arcade controls, one stick's *vertical* axis controls the throttle power (forward/backward), while the other stick's *horizontal* axis controls the turn power, which is linearly added/subtracted to either side of the drivetrain.

B

The second control scheme we wanted to try was **tank controls**, where the vertical values of either joysticks are plugged directly into the power of either side of the drive. This can be less intuitive, but is sometimes preferred due to its possible higher degree of control.

C

The last control scheme we considered was the single-stick control scheme, which is identical to the arcade controls, but both the throttle and turn power are taken from a single joystick. While this can be more game-like and thus more intuitive, it does pose additional programmatic challenges. (see next pages)

Designed by:
Adam X

Witnessed by:
Kevin Z

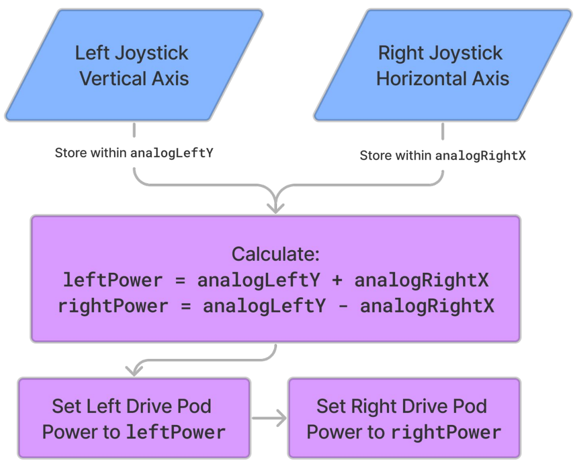
Driver Control: Chassis Movement

2024-10-20 | **Operator-Controlled**

▶ **Problem:** Find the best control-scheme for the chassis, as well as what our driver prefers most.

[Select a Solution]

Based on the above results, it was clear that the best option for Maxwell was the arcade control scheme. Despite it being harder to make complex maneuvers, we soon realized this was just a matter of practice, and if we needed more controller space, we could start using controller scuffs.



[Program Solution]: Here's how we implemented the above flowchart and control scheme.

```cpp
void Eclipse::OPControl::drivetrain_control(){
    int32_t rightXjoystick = (controller.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X));
    int32_t rightYjoystick = (controller.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y));
    int32_t leftYjoystick  = (controller.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y));
    int32_t leftXjoystick  = (controller.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_X));
    if(abs(leftYjoystick) < 10) leftYjoystick = 0;
    if(abs(rightYjoystick) < 10) rightYjoystick = 0;

    int32_t left_power = (rightXjoystick + leftYjoystick) * (12000.0 / 127);
    int32_t right_power = (leftYjoystick - rightXjoystick) * (12000.0 / 127);

    left_drive.move_voltage(left_power);
    right_drive.move_voltage(right_power);
}
```

Get the joystick values from the controller and store them

Set the left motors to the sum of the joystick inputs, and set the right motors to the difference. This ensures that the turn gets applied to either drive pod in opposite direction, which causes the robot to turn.

To prevent faulty joysticks or joystick drift, if the values are under 10, then set them to 0

Designed by: Adam X

Witnessed by: Maxwell L

Driver Control: Chassis Movement

2024-10-20 | **Operator-Controlled**

▶ **Problem:** Find the best control-scheme for the chassis, as well as what our driver prefers most.

[Test Solution]

After programming all three systems, it was time to test each one and evaluate the best one. Here's what we tested each system (qualitatively) on:

- **Driver Comfort:** Does Sai feel at-home with these controls? Or will these controls need some significant getting-used-to for Sai to be able to drive with them?
- **Driving Versatility:** Can/Could Sai make a variety of maneuvers with this? How easy are these maneuvers to control?
- **Controller-Space Utilization:** How much 'space' does this take up on the controller? Are certain configurations going to make Sai likelier to accidentally press buttons?

Test Results: Control Scheme Comfort Levels

Control Scheme	Driver Comfort	Driving Versatility	Controller-Space Utilization
Arcade Drive	**Relatively Good -** Sai seemed to get used to this setup faster than any other setups, and his driving was most smooth with this one	**Mediocre -** Sai seemed to move around quite well with this setup, but struggled on certain maneuvers such as pivoting around a single drive pod.	**Bad -** Takes up both joysticks, which could lead to accidental presses on either D-Pad, and doesn't leave either of Sai's thumbs available or outside usage
Tank Drive	**Relatively Bad -** This took Sai a while to get the hang of, and even after nearly 20 minutes of testing, he couldn't always drive straight using this setup.	**Mediocre -** While this setup opened the door to a variety of other mauvers such as easier drifting, regular driving seemed significantly harder.	**Bad -** Takes up both joysticks, which could lead to accidental presses on either D-Pad, and doesn't leave either of Sai's thumbs available or outside usage
Single Stick	**Mediocre -** While he did have an easier time getting used to this than the tank drive, he still open moved his other thumb on the joystick, expecting it to turn; thus suggesting a lack of comfort.	**Mediocre -** Sai seemed to move around quite well with this setup, but struggled on certain maneuvers such as pivoting around a single drive pod. (Same as Arcade)	**Good -** Only takes up one joystick, allowing Sai to use this other thumb to press any one of four D-Pad Buttons

Designed by:
Adam X

Witnessed by:
Maxwell L

Using PID Control for Movement

2024-10-23 | **Motion Control**

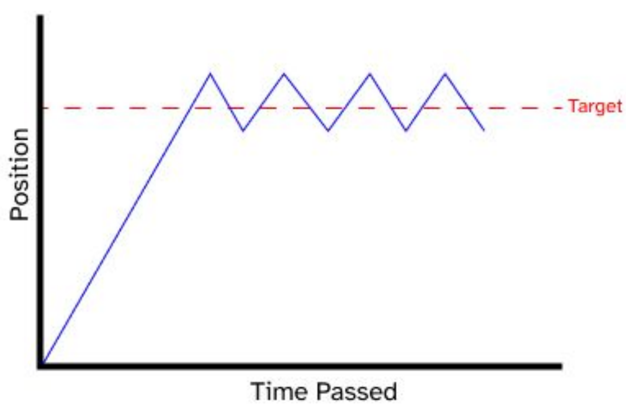
▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

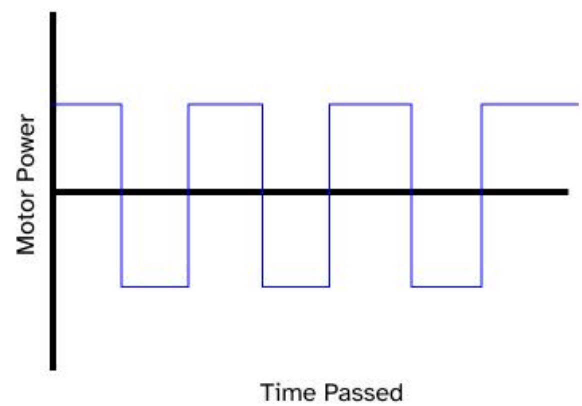
[Brainstorm Solutions]

To fulfill the requirements we stated on the previous page, we researched and brainstormed **four** solutions/algorithms that could help us achieve the goals stated. It should be noted that these solutions are **not mutually exclusive**, so we may choose to implement more than one at a time.

#1: Bang-Bang Control

This is the most simple solution we found that was on the BLRS Wiki, which was known as bang-bang control. In theory, this algorithm works by taking a current position and setpoint (target), and finds the difference, which is the measured error. If the error is positive, positive power is inputted, and if it is negative, negative power is inputted. This back and forth between positive and negative power gives the algorithm the name 'bang-bang', as it repeatedly bounces between two equal and opposite states.





In regards to robotics, this algorithm moves the robot at a given speed, until it is within a range of the target value. If it overshoots, it sets the speed to the negative of the given speed. Every iteration loop (so every 10ms), the position is checked using the IME values, and whether or not those values are within a range of the target is validated. This would be more accurate due to taking IMEs into account, and shouldn't be too jerky at moderate speeds.

Designed by: Adam X

Witnessed by: Kevin Z

Using PID Control for Movement

2024-10-23 | Motion Control

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Brainstorm Solutions]

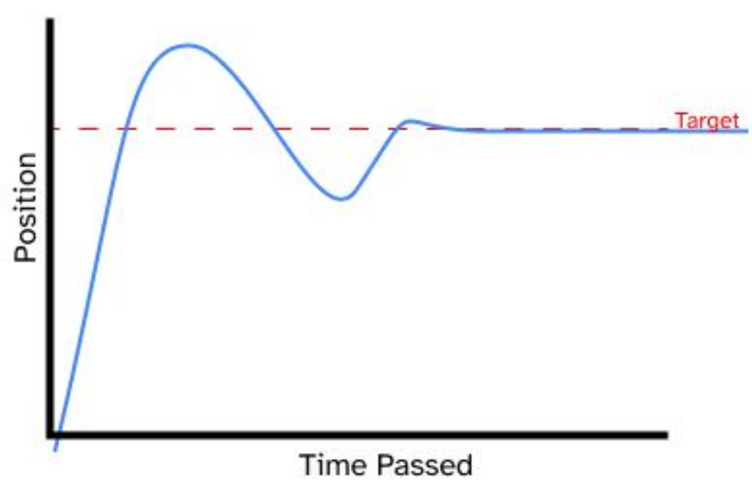
#2: PID Control

Upon doing some more research on possible motion algorithms, we learned about PID, which is the 'gold standard' of non-linear motion control. Based on the BLRS Wiki as well as An Introduction to PID Controllers by George Gilliard, we've learned that PID is an **error-correction algorithm** which bases a system's input power on the size of the error, as well was the change in error.

It takes the name PID as it stands for **Proportional-Integral-Derivative**: the three components that make up the power input. That is, the power input is the sum of:

- The **proportional** term, a value proportional to the current error (so greater error means more power)
- The **integral** term, the cumulative sum of errors (so the longer it takes to reach a target, the more power goes in
- The **derivative** term, which takes the direction of the previous power change (so it basically boosts the power of whatever direction the system is moving in.

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{d}{dt}e(t)\right)$$



Designed by:
Adam X

Witnessed by:
Maxwell L

Using PID Control for Movement Cnt'd

2024-10-23 | **Motion Control**

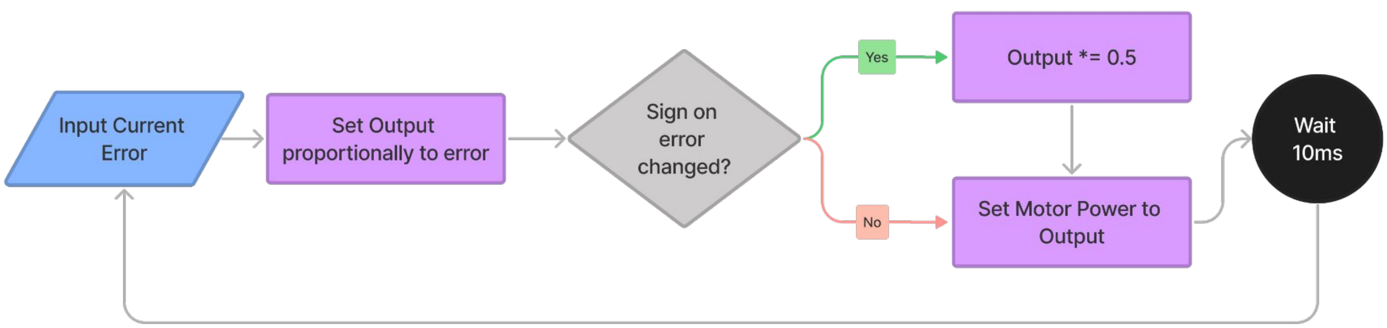
▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Brainstorm Solutions]

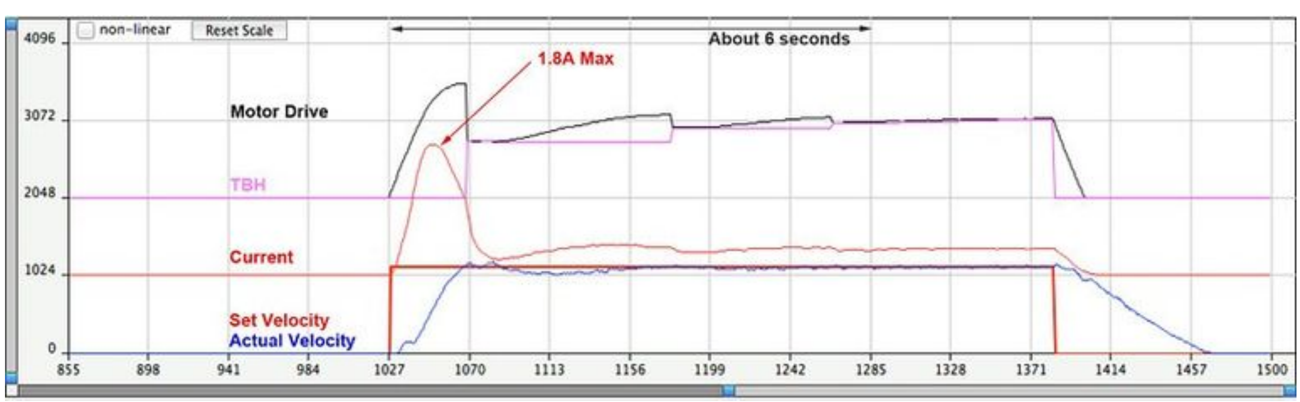
#3: TBH Control

Once again referencing the BLRS Wiki, the **Take-Back-Half** (TBH) controller adjusts the input power by a fraction of the error, gradually reducing it until the error approaches zero. It uses a simple approach of taking half of the previous control signal when the error changes sign, giving rise to the name "take-back-half". Since it doesn't rely explicitly on a derivative or integral term, it tends to be a lot more stable for loads that need to remain especially stable or accurate.

The following flowchart depicts how the TBH controller works in principle. If the sign of the error changes, the output power gets halved immediately.



In terms of robot locomotion, while this implementation is generally simpler and a bit slower than PID, it tends to better avoid overshoot and adheres more closely to the target point. The take-back-half condition prevents overshoot right as it happens regardless of tuning, so it also often avoids steady state error (where the robot stops moving before or after reaching its target.



(VEX U Team BLRS, 2023)

Designed by:
Adam X

Witnessed by:
Alex S

Using PID Control for Movement Cnt'd

2024-10-23 | **Motion Control**

▸ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Select a Solution]

Based on the brainstormed solutions on the previous few pages, we used a decision matrix to evaluate each solutions and pick the best one(s). We started by listing our the evaluation criteria, as you can see below

- **Programming Simplicity (/4):** How simple is this to learn and implement?
- **Interpolation Quality (/10):** How *smoothly* does this algorithm allow the robot to reach its target position? Does it cause jerky motions, or abrupt stops?
- **Closed Loop Quality (/7):** How much of the error data from the IMEs is taken into account? Is only the moment-of error taken into account, or are previous error values also considered?
- **Tuning Time (/5):** How difficult will this algorithm be to tune and debug?

Choosing a Programming Runtime: Decision Matrix

	#1: Bang Bang Control	**#2: PID Control**	#3: TBH Control
Programming Simplicity (/4)	**4** Super simple to program, the algorithm is basically just a loop	**2** Harder to program, since it requires a more complex calculation	**3** Simpler than PID, but still has more complex error correction
Interpolation Quality (/10)	**5** Often causes jerky motion at higher speeds, so the robot has to move slowly	**9** Works really well to get smooth motions fast	**6** Works well for most motions but can't be as well-tuned as PID
Closed Loop Quality (/7)	**4** Doesn't constantly try to get closer to its target; just within a range	**6** Tries to get as close as possible until a time limit is reached	**6** Tries to get as close as possible until a time limit is reached
Tuning Time (/4)	**3** Only the error range must be tuned	**2** Three variables need to be tuned	**3** Gain value needs to be tuned
TOTAL	16	**19**	18

Based on the matrix above, we decided to go with **PID** as our primary motion control algorithm. While the TBH controller came very close, after discussing with an Alumni, most said that PID was most definitely worth the effort, so that's what we ended up going with.

Designed by:
Adam X

Witnessed by:
Richard F

Using PID Control for Movement Cnt'd

2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Plan a Solution]

Proportional Integral Derivative Controller (PID Controller)

Formula for a PID Controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Each portion of the equation is responsible for handling a specific part of correcting robot movement as it attempts to reach a target position.

Proportional Controller

In an ideal scenario, the robot's motion would maintain a constant speed. However, achieving an instantaneous change to a specific velocity is impractical, and a gradual acceleration is necessary. The Proportional Controller addresses this by continuously reducing the error between the robot's current position and the target position. This error is multiplied by a constant, kp, resulting in a velocity that diminishes over time as the robot approaches its desired target.

This approach ensures smoother linear movements, promoting consistency. It's important to note that when velocity is calculated based on position, initial speeds may appear very fast, leading to minor abrupt movements at the start. To address this, implementing a maximum speed threshold and constraining the robot's velocity within this limit can effectively mitigate any initial jerks.

Designed by: Adam X

Witnessed by: Alex S

Using PID Control for Movement Cnt'd

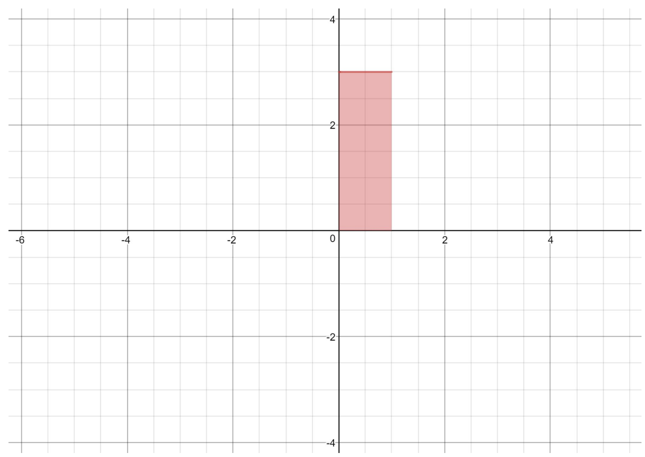
2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

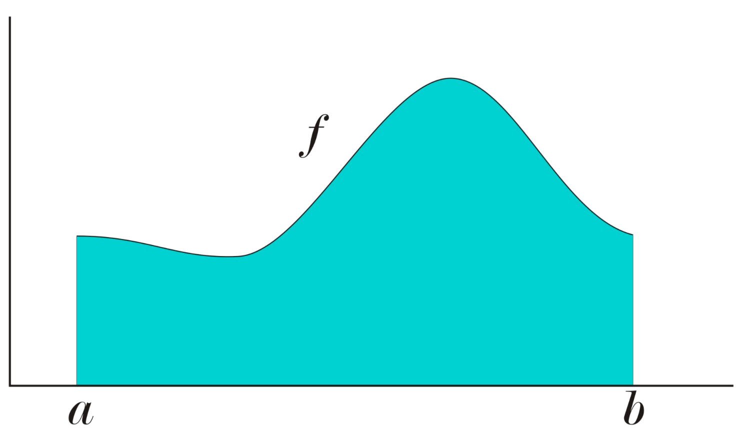
[Plan a Solution]

Integral Controller

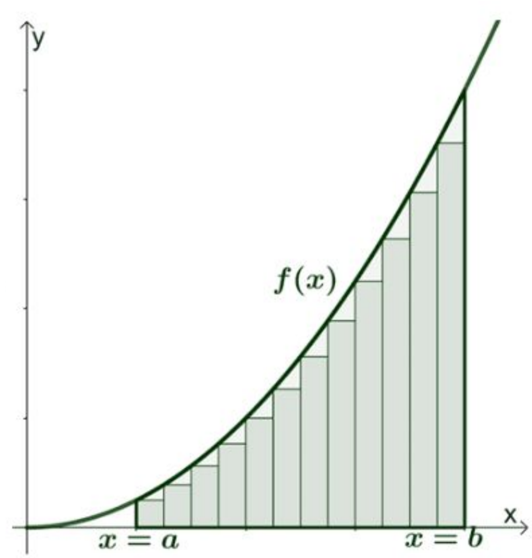
Calculating the area for simple geometric shapes is straightforward using standard formulas. For instance, the area of a rectangle can be determined with the formula: Area = Length * Height



However, when faced with more complex or abstract shapes, conventional formulas may not apply. Consider the challenge of determining the area beneath a curve like this:



For such shapes, a traditional formula won't suffice. The method for determining this type of geometry is by utilizing something called an integral calculus. Instead of relying on a predefined formula, integration involves breaking down the shape into infinitesimally small components, often rectangles under the graph in this context. By summing up the areas of these infinitely small rectangles beneath the curve, we can effectively compute the area of the graph, regardless of its specific shape.



Designed by:

Adam X

Witnessed by:

Alex S

Using PID Control for Movement Cnt'd

2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Plan a Solution]

Integral Controller

The formula for the integral of the error of the robot may be represented with:

$$K_i \int_0^t e(\tau)d\tau$$

In the controller's operation, the integral component is implemented by continuously updating an iterator within a running loop while the robot is in motion. This iterator serves as the cumulative sum of all errors encountered during the movement. The purpose of maintaining this sum is to address potential undershooting of the target location. The loop, executing at a frequency of every 10 milliseconds, ensures that the iterations occur frequently enough to simulate an almost infinite sum of errors, effectively mimicking the behavior of an integral. In a temporal context, if we consider error as a function of time, the integral of our error corresponds to the sum of all errors encountered up to the current iteration of the loop.

$$\sum_{k=a}^{b} f(a) = \int_a^b f(t)dt$$

Designed by: Adam X

Witnessed by: Alex S

Using PID Control for Movement Cnt’d

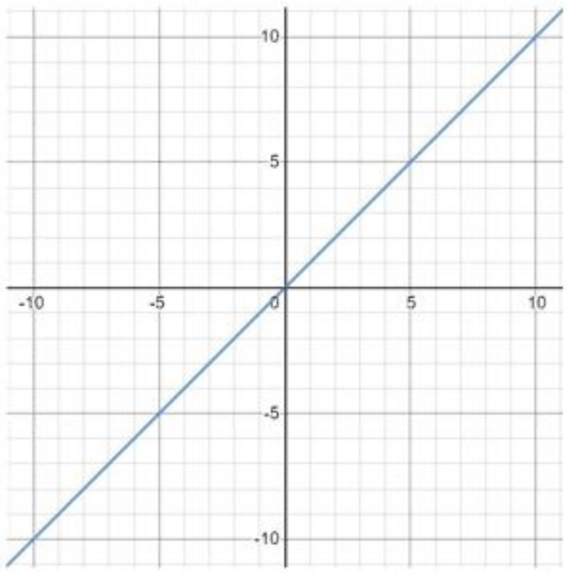
2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Plan a Solution]

Derivative Controller

The derivative is a mathematical principle that aids in comprehending the rate of change of one quantity relative to another. It measures the rate at which one thing is changing with respect to another. To illustrate, contemplate a basic linear equation:



$$y = mx + b$$

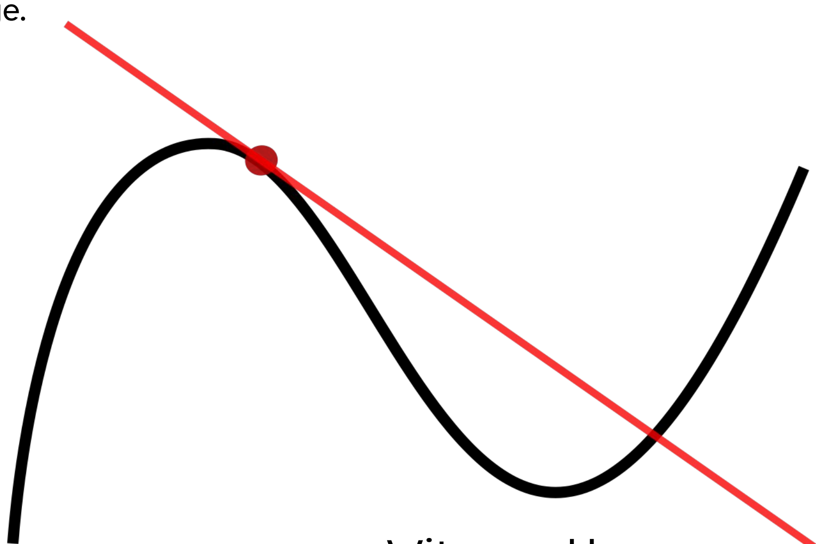
y = y coordinate

m = slope

x = x coordinate

b = y intercept

The m value represents the slope, or the rate of change. In the case of a linear equation, the rate of change remains constant. The derivative of a function provides a new function that conveys the rate of change of the original function at every point. This derivative function informs us about whether the original function is experiencing an increase or decrease, and it can also indicate the degree of steepness in the change.



Designed by:

Adam X

Witnessed by:

Alex S

Using PID Control for Movement Cnt'd

2024-10-23 | Motion Control

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Plan a Solution]

Derivative Controller

How does this work within the robot? In each iteration of the process, we observe the robot's slope, representing the "rate of change." The derivative section of the controller keeps a record of this rate and multiplies it by a constant Kd. By monitoring the rate of change, the derivative component can effectively address unforeseen variations, such as overshooting the target. It responds by applying a reverse voltage to realign the robot and guide it back on track toward reaching the desired position, this is known as a damping value.

$$K_d \frac{d}{dt} e(t)$$

Putting the PID Together

When we combine all portions together, we get this graph. Which represents the performance created by a PID controller.

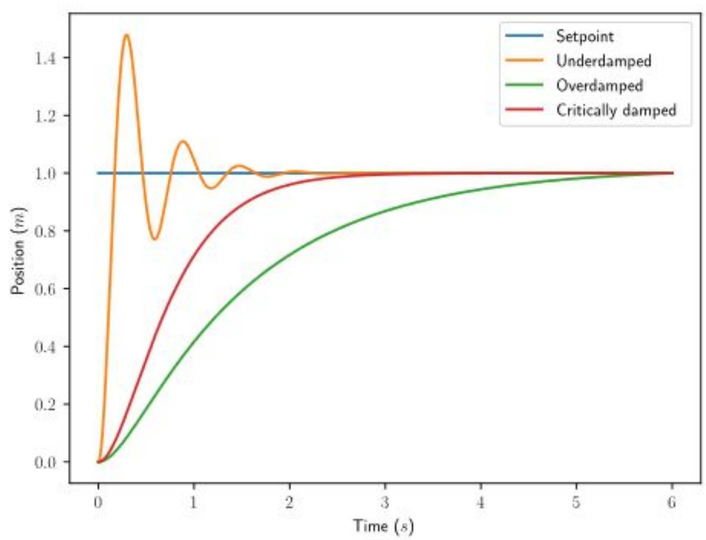


Figure 2.8: PID controller response types

Designed by:
Adam X

Witnessed by:
Alex S

Using PID Control for Movement Cnt'd

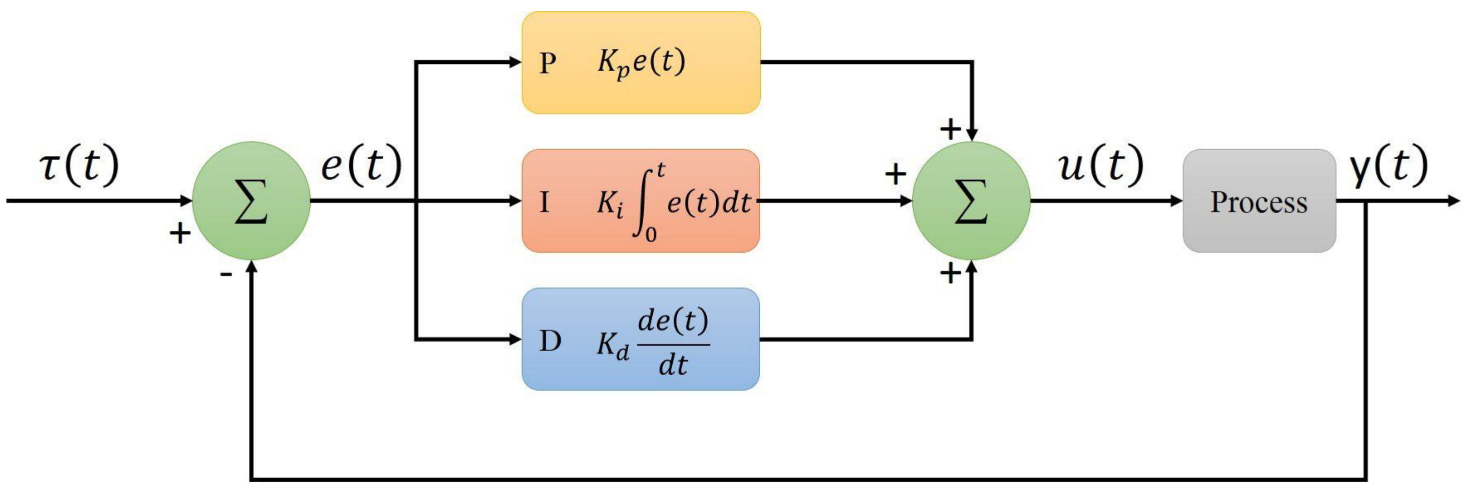
2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Plan a Solution]

Putting the PID Together

Thi diagrams explained the process of a PID Controller extremely well. The process involves looping through each component of the PID Controller to derive an optimized voltage value for precise lateral movements. Once the voltage is calculated, it can be readily applied to the drivetrain motors, resulting in a significantly improved and accurate control of the system.



The feedback controller is able to account for any changes in potential error during movement, such as friction on one end of the drivetrain, or hitting an obstacle during a rotation. In the event of such scenarios, the PID Controller is designed to effectively consider and adjust for the error, ensuring the system stays on its intended path.

Designed by:
Adam X

Witnessed by:

Using PID Control for Movement Cnt'd

2024-10-23 | **Motion Control**

▶ **Problem:** How can we use closed-loop error-correction algorithms for effective robot locomotion?

[Program Solution]: Here is the generic PID class' `compute()` method, which returns the amount of power that should be outputted by the PID controller based on the current and previous error. Within this method, all PID-related variables are automatically updated, so all internal state is self-handled.

```
double Eclipse::Translation_PID::compute_t(double current_pos, double target)
{
    t_pid.t_error = target - current_pos;
    t_pid.t_derivative = t_pid.t_error - t_pid.t_prev_error;

    if (t_pid.t_ki != 0){ t_pid.t_integral += t_pid.t_error; }

    if (util.sign(t_pid.t_error) != util.sign(t_pid.t_prev_error)){ t_pid.t_integral = 0; }

    double power = (t_pid.t_error * t_pid.t_kp) + (t_pid.t_integral * t_pid.t_ki) + (t_pid.t_derivative * t_pid.t_kd);

    // std::cout << "output" << power << std::endl;

    if (power * (12000.0 / 127) > t_pid.t_max_speed * (12000.0 / 127)){ power = t_pid.t_max_speed; }
    if (power * (12000.0 / 127) < -t_pid.t_max_speed * (12000.0 / 127)){ power = -t_pid.t_max_speed; }

    t_pid.t_prev_error = t_pid.t_error;

    return power;
}
```

A. Calculate error and update derivative for this iteration.
B. If the error crosses 0, then it means the integral is now all in the wrong direction (since the bot needs to change direction), so reset the integral term
C. This is the main output of the PID function
D. Set a prevError variable for reference on the next iteration

Designed by:
Adam X

Witnessed by:
Maxwell L

Accommodating for Two Drivers

2024-11-15 | **Operator-Controlled**

▶ **Problem:** We have two drivers, each with different preferences

[Identify Problems]

We have two drives: Maxwell for matches and Kevin for skills, and each of them have different preferences regarding what controls they are most comfortable with.

[Brainstorm Solutions]

A

The first idea that came to mind was having a boolean variable, meaning it is either true or false, that indicated to the program that if the driver controls were for a match, or for skills. This would then be downloaded into different code slots onto the brain, one for match controls and one for skills.

```
if(skills == false){
        driver.driver_control();
}
else{
        driver.skills_control();
}
```

B

The second solution was to manually edit each if statement in the driver_control function and also download it into different code slots in the brain. Although it is a valid solution, it would pose very tedious and time consuming if a major component in the code needed to be updated, and would require repeated adjustments

C

The last idea that we considered was to implement a multi=purpose graphical user interface (GUI) that had many functions including selecting a skills or match control. This solution wouldn't require any downloading to extra code slots onto the brain, which saves time.

Designed by:

Adam X

Witnessed by:

Maxwell L

Accommodating for Two Drivers

2024-11-15 | **Operator-Controlled**

▶ **Problem:** We have two drivers, each with different preferences

[Plan a Solution]

	#1: Boolean State	#2: Manual Changes	#3: Graphical User Interface
Programming Simplicity (/4)	**2** Quite simple, just need to add a few functions and other minor details	**1** Doesn't require any additional changes	**4** Coding a graphical user interface (GUI) takes a lot of time and can be very complex
Efficiency (/5)	**3** Quite a few changes still need to be made, code needs to be updated on both slots, but less than manual	**5** Time consuming changes need to be made every time the code is updated	**1** No changes need to be made for different driver codes, only a button needs to be pressed
TOTAL	**5**	6	5

**The lower the number, the better*

Based on the matrix above, we selected using a boolean state variable to determine whether the driver code would be for matches or skills.

[Program Solution]

The first thing we did was create separate functions for skills with different controller buttons and labelled them as skills. Then, we created a global boolean variable called "skills" that when false, ran the match control, and when true, ran the skills control.

```
void Eclipse::OPControl::driver_control(){
    driver.drivetrain_control();
    driver.power_intake(100);
    driver.prime_wall_stake();
    driver.power_wall_stake();
    driver.activate_clamp();
    driver.activate_doinker();
    driver.lift_intake();
}

void Eclipse::OPControl::skills_control(){
    driver.drivetrain_control();
    driver.power_intake(100);
    driver.prime_wall_stake_skills();
    driver.power_wall_stake_skills();
    driver.activate_clamp_skills();
    driver.activate_doinker_skills();
    driver.lift_intake_skills();
}
```

```
if(skills == false){
        driver.driver_control();
}
else{
        driver.skills_control();
}
```

By default, variable "skills" is set to false, but that can be adjusted in main.cpp

Designed by:

Adam X

Witnessed by:

Maxwell L

Wall Stake Macro Logic

2024-10-20 | **Operator-Controlled**

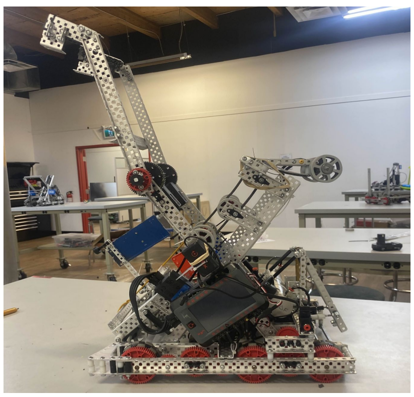
▶ **Problem:** What is the best way of controlling our wall stake mechanism?

[Identify Problems]

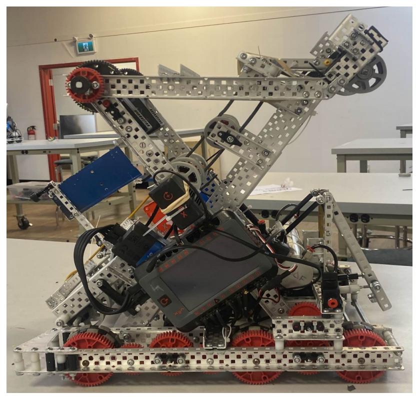
The wall stake mechanism is quite complicated to control, and there are many ways that we can go about doing this.

[Brainstorm Solutions]

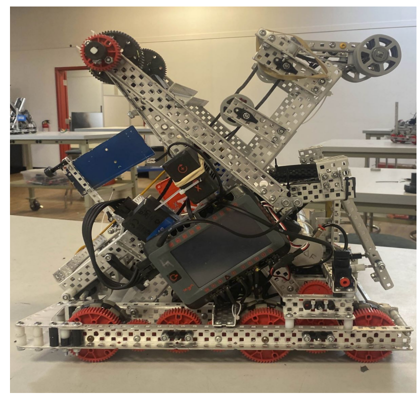
Many teams use something called a **macro**, which is supposed to simplify controlling movements by executing a sequence of commands as a single input or in our case, a button press. First, let's define the states and positions that we want the wall stake mechanism to be in.



Scoring



Prime



Down

#1: Fully Manual

We considered doing a fully manual control, which would be the simplest to program, wouldn't add any other complexities. However, this solution would take away from the possibility of any precise movements to a certain position, like when its in its prime position. Having manual controls would take more time to line up and move the mechanism into those positions, because of the reduced control and precision.

#2: Partially Manual

This solution would operate similar to the first, where the driver still has manual control over the wall stake mechanism, but there is a button that automatically moves the mechanism to the prime position. There would solve the previous problem of lack of precision, because using by using motor encoders or sensors, then we can accurately move the mechanism into the right spot.

#3: Automatic

This solution would work with two buttons. One would alternate between the prime and down positions, so it would be pressed to either bring the wall stake mechanism down or to bring it to prime position. The other button, when pressed, would rotate the arm into a neutral stake until it reaches a certain position, then rotate back to prime position.

Designed by: Adam X

Witnessed by: Maxwell L

Wall Stake Macro Logic

2024-10-20 | **Operator-Controlled**

▶ **Problem:** What is the best way of controlling our wall stake mechanism?

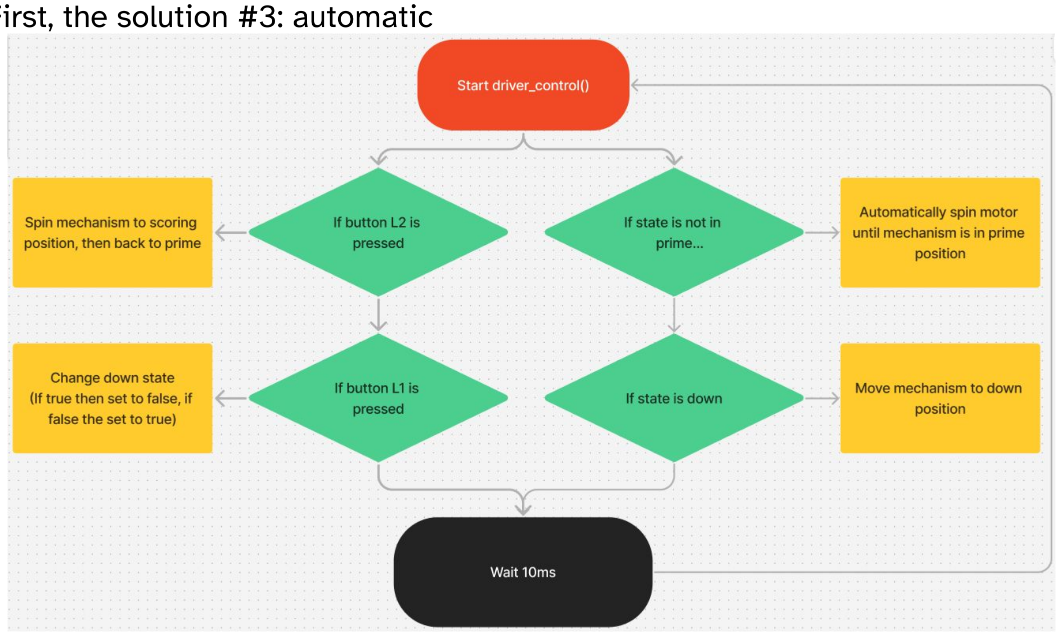
[Plan a Solution]

	#1: Fully Manual	**#2: Partially Manual**	**#3: Automatic**
Programming Simplicity (/2)	**1** Quite simple, just need to add a few functions and other minor details	**1** Doesn't require any additional changes	**2** Coding a graphical user interface (GUI) takes a lot of time and can be very complex
Driver Efficiency(/5)	**5** Quite a few changes still need to be made, code needs to be updated on both slots, but less than manual	**3** Time consuming changes need to be made every time the code is updated	**2** No changes need to be made for different driver codes, only a button needs to be pressed
TOTAL	7	**4**	**4**

There was a tie between solution 2 and 3 from the decision matrix above, each having its own pros and cons. This worked out quite well, as Kevin, who was driving skills, wanted more control over the mechanism while driving, so he wanted the partially manual, and Maxwell felt more comfortable with the automatic controls.

[Program Solution]

Now that we have selected the two solutions, here's the visualization of the logic that operates the two solutions. First, the solution #3: automatic



Designed by: Adam X

Witnessed by: Kevin Z

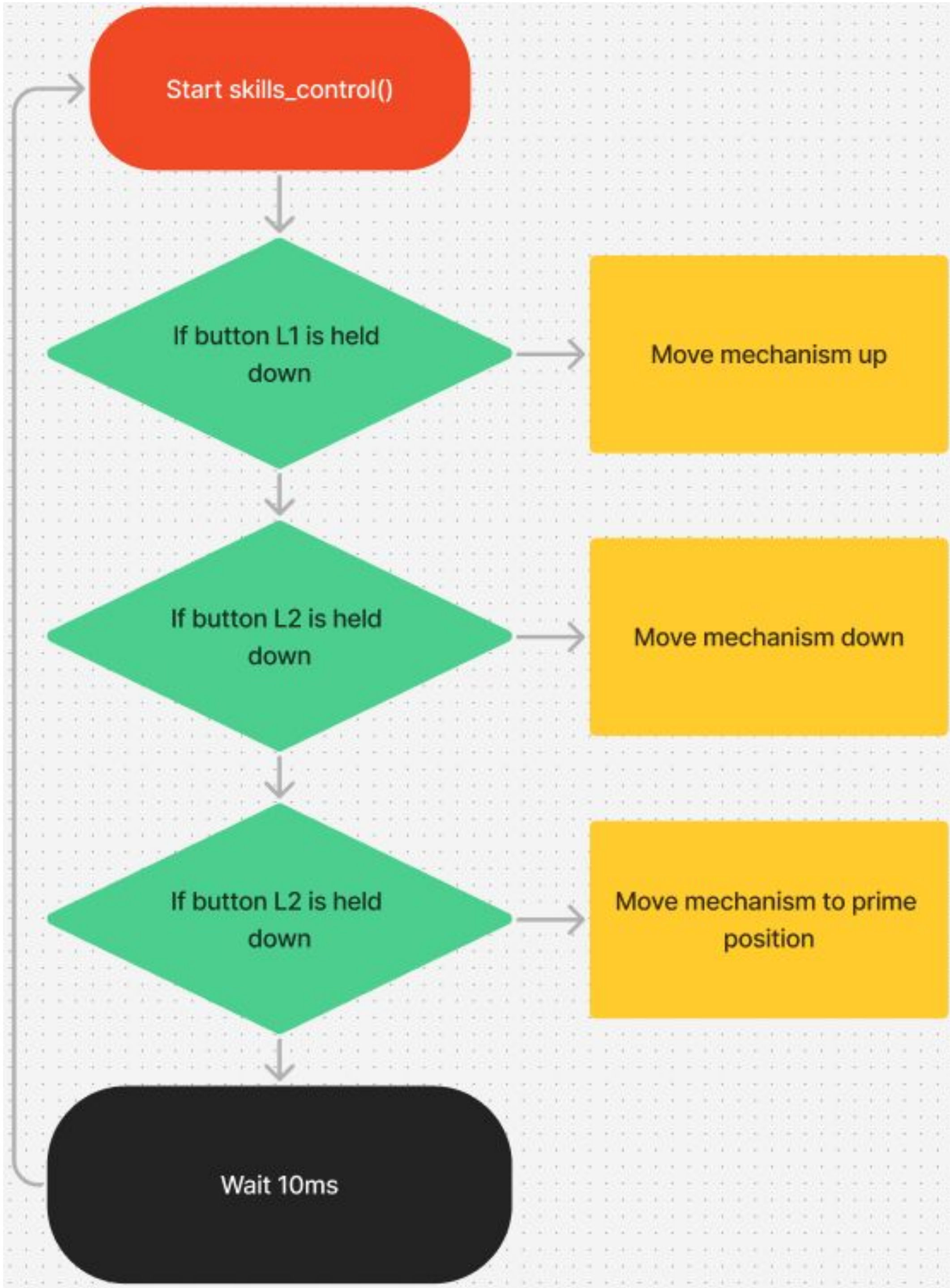
Wall Stake Macro Logic

2024-10-20 | **Operator-Controlled**

▶ **Problem:** What is the best way of controlling our wall stake mechanism?

[Program Solution]

Now, for the partially manual controls:



Designed by:
Adam X

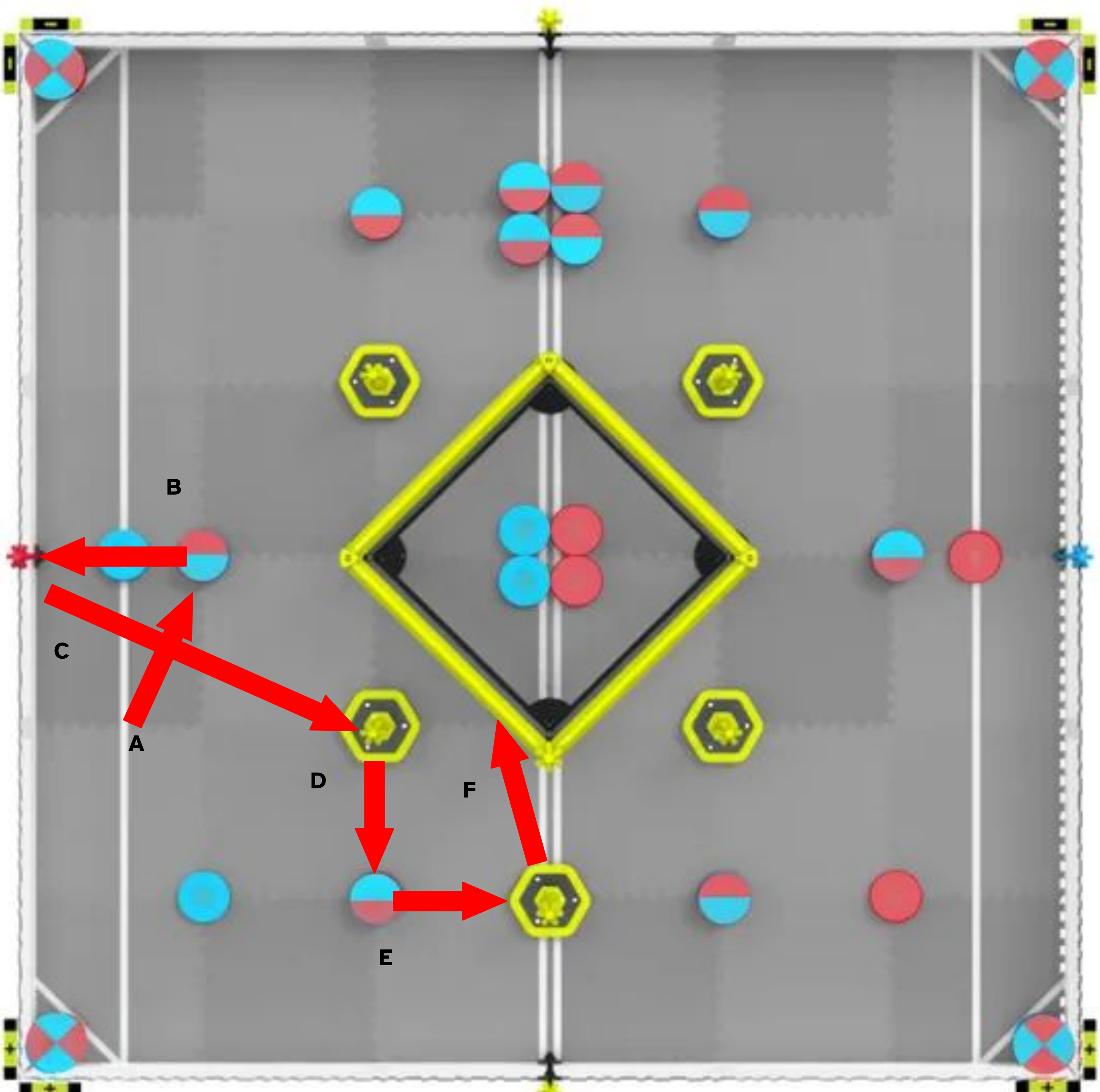
Witnessed by:
Kevin Z

Autonomous Paths: Solo AWP

2024-11-27

▶ **Focus:** Figure out what our autonomous paths will be

[Plan a Solution]



Designed by:

Adam X

Witnessed by:

Maxwell L

Autonomous Paths: Solo AWP

2024-11-27

▶ **Focus:** Figure out what our autonomous paths will be

[Plan a Solution]

During this session we collectively decided what the route for the **solo autonomous win point** route would be. Because the teams in our region were relatively weak and the autonomous win point criteria was relatively achievable from one robot, we decided to develop a solo autonomous win point route. The requirements are:

- Score at least 3 rings
- Score on at least 2 different stakes
- Both robots must be off and not contacting the starting line
- One robot must be contacting the elevation ladder

Step	Route 1
A	Drive forward to intake the red ring on top using the intake lift piston
B	Drive backwards and score onto the alliance stake using the pre-load
C	Drive forwards to clamp onto the mobile stake
D	Drive forwards and intake the red ring on the bottom, then drop the goal
E	Turn and drive backwards to clamp the 5th goal on the autonomous line
F	Drive forwards into the elevation ladder to contact

This route would be the only one that we would run during qualifications. During eliminations, we would most likely end up using the same route as well

Designed by:

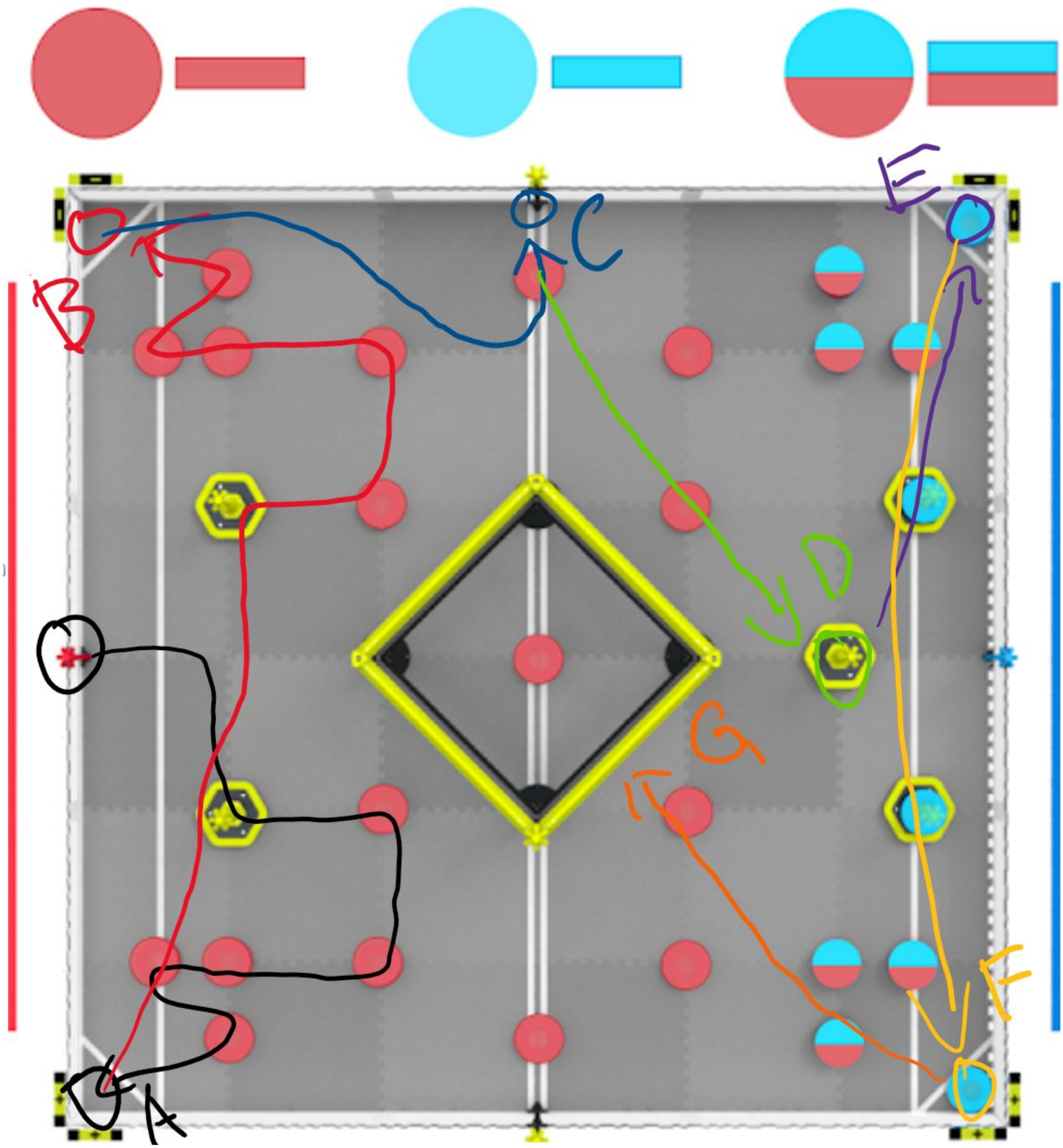
Witnessed by:

Maxwell L

Autonomous Paths: Skills

2024-12-09

▶ **Focus:** Figure out what our autonomous paths will be



Designed by:

Witnessed by:

Maxwell L

Autonomous Paths: Skills

2024-12-09

▶ **Focus:** Figure out what our driver skills path will be

[Plan a Solution]

For our autonomous skills, we wanted something that was a bit less complicated than the driver, but still high scoring

Step	Route 1
A	Score alliance stake, clamp goal, score 5 rings, and place in the corner
B	Clamp onto the second goal, score 5 rings and place into the corner
C	Score wall stake
D	Back up and clamp goal, score 1 ring and drop it
E	Push goal into the corner
F	Push other goal into the other corner
G	Climb the ladder

This path only scores 5 rings onto each of the first 2 goals and places them into the corners. Then, we only score 1 ring onto the third goal and try to push 1 goal in each of the back corners.

[Test Solution]

When testing out programming skills runs with this route, we found that the first half was extremely consistent up to step B, and step C mostly hit. Any step after that was very inconsistent and would vary from skills run to skills run

Designed by:

Adam X

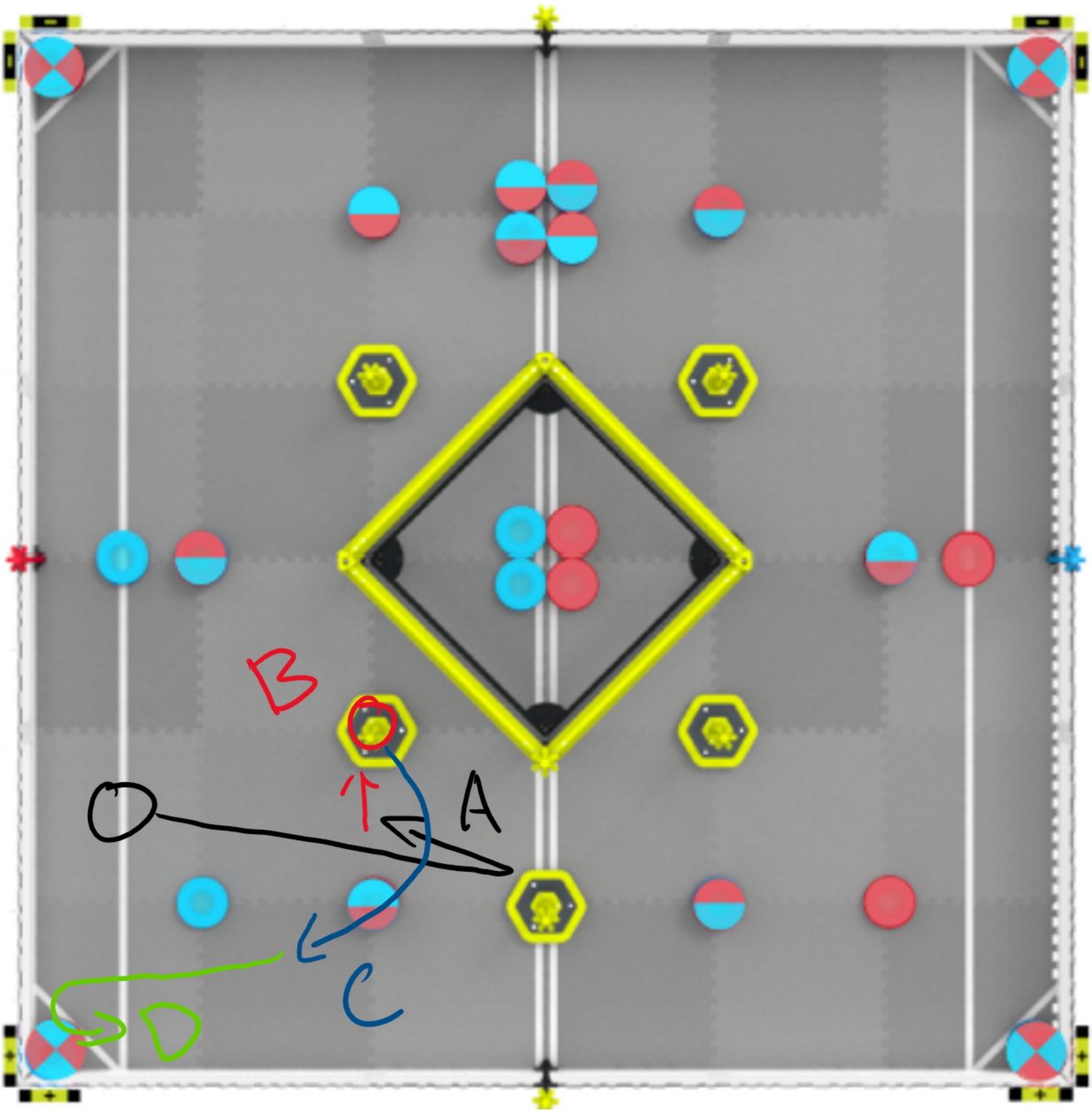
Witnessed by:

Kevin Z

Autonomous Paths: Rush

2024-12-12

▶ **Focus:** Figure out what our eliminations autonomous path will be



Designed by:

Adam

Witnessed by:

Maxwell L

Autonomous Paths: Rush

2024-12-12

▶ **Focus:** Figure out what our eliminations autonomous path will be

[Plan a Solution]

For eliminations, we really are stressing on getting possession of the fifth goal on the autonomous line, since that is a huge decider on who wins matches.

Step	Route 1
A	Immediately drive forwards, grab the 5th goal with the doinker arm, drive back, and release the goal
B	Clamp onto the goal and score the preload
C	Drive backwards into the 5th goal, intake the red ring on the bottom of the stack.
D	Drive to corner and clear

Designed by:

Adam X

Witnessed by:

Kevin Z

Designing a Graphical User Interface (GUI)

2024-12-25

▶ **Problem:** What do we want our graphical user interface to do?

[Identify Problems]

A graphical user interface has many uses in a match, and should be able to perform multiple tasks. We also have no way of actively choosing which autonomous path we want to run, and no way to see all of the sensor readings.

[Brainstorm Solutions]

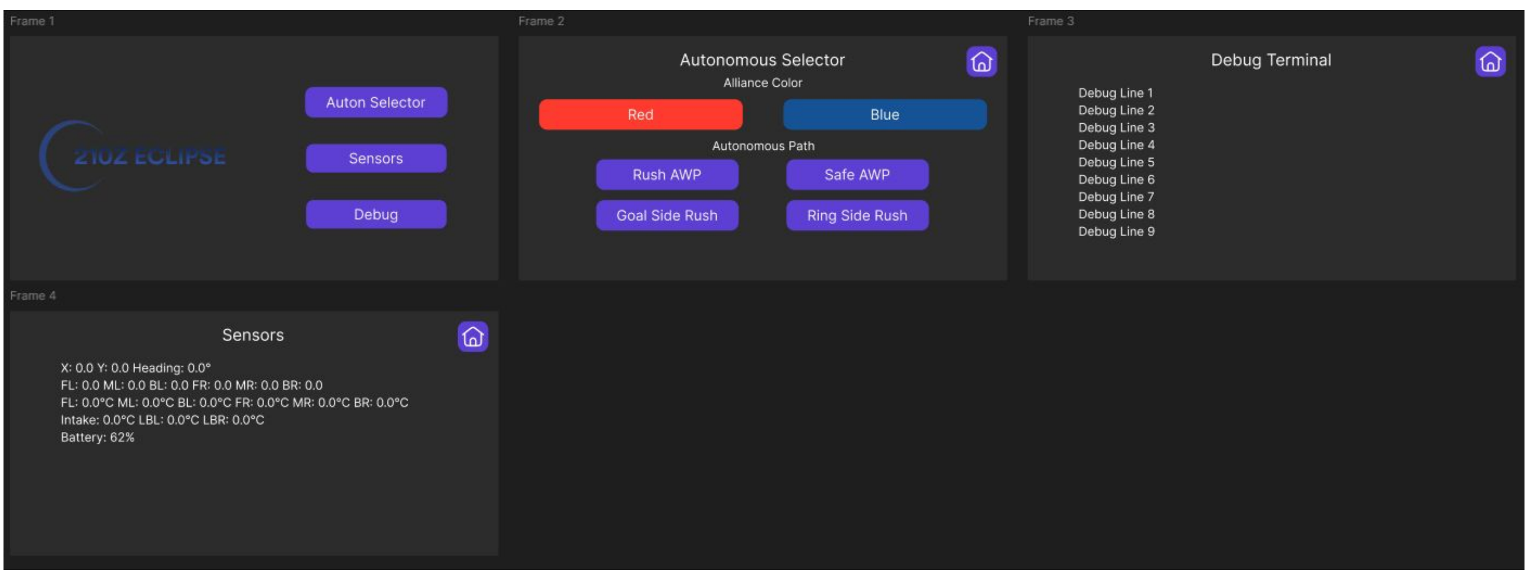
Here's a rough plan of the actions we want our graphical user interface to be able to perform.

1. **Have an autonomous selector**
2. **Be able to act as a debug terminal**
3. **Display in-game sensor metrics and data**
4. **Select between skills and normal driver controls**

It should be run during matches, so that we can select between the different screens for the different functions, and also perform all of the functions listed above.

[Program Solution]

After designing a quick visual prototype using Figma, and it looks like this:



The screen in the top left is the starting menu/home screen, with 3 buttons to each respective page. Each page also has a home button in the corner to return to the home page.

Designed by: Adam X

Witnessed by: Zechariah W

Designing a Graphical User Interface (GUI)

2024-12-25

▶ **Problem:** How do we actually program the graphical user interface?

[Brainstorm Solutions]

Now that we knew what we wanted it to do and how we wanted it to look, how do we actually code it to make it into a reality?

#1: PROS LLEMU (Legacy LCD Emulator) API

The first option is to just use the built-in lcd methods in the PROS API that can create a custom display in the LCD display of the brain. These methods are quite simple to use and understand, as there are only a few of them. They are apart of the pros::lcd class and consist of methods like clear_line, set_background_color, and more.

#2: LVGL (Light and Versatile Graphics Library)

LVGL is the most popular free and open-source embedded graphics library to create beautiful UIs for any MCU, MPU and display type. A version of it is already built-in to PROS, which means no extra installations are required.

#3: Robodash Library

Robodash is a GUI toolkit for the VEX V5 brain that provides GUI tools that take better advantage of the brain's LCD display and a system to improve compatibility with templates that provide LVGL-powered GUIs.

[Plan a Solution]

I decided that **solution 2**, LVGL was the best solution due to its versatility. Both other solutions are either just built-in templates or too simple to make something that I want to be fully adjustable and customizable. LVGL also offers a wide range of features that can be used to create modern and visually appealing user interfaces, including support for animation, anti-aliasing, fonts, images, and widgets such as buttons, sliders, and lists. In addition to that, LVGL provides a powerful layout engine that makes it easy to arrange and position objects on the screen so that I can directly enter the figma positions of each object as the location in LVGL and everything should be in the right spot.

Designed by: Adam X

Witnessed by: Zechariah W

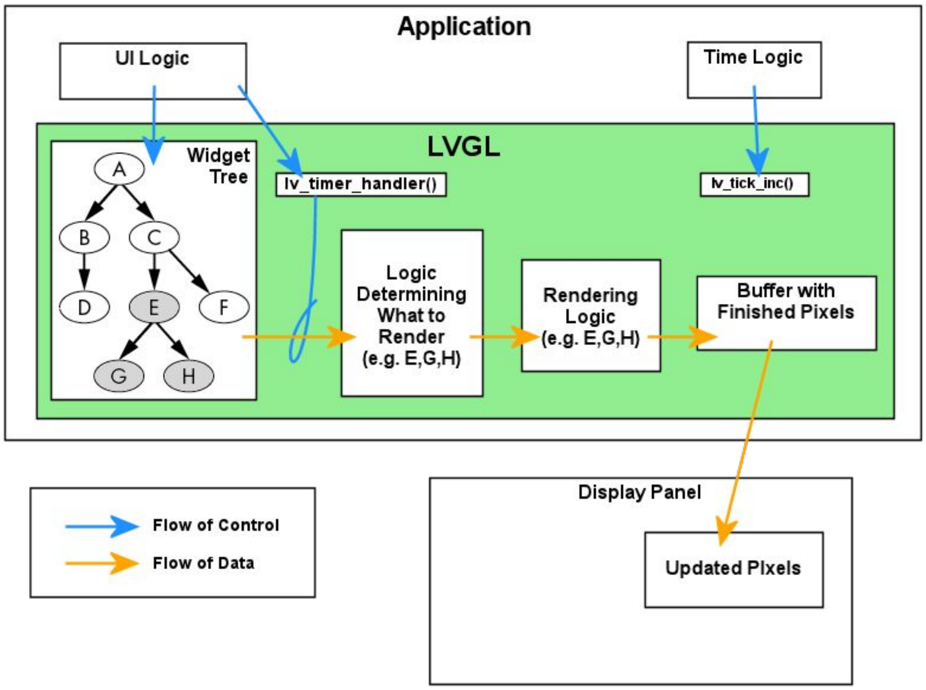
Designing a Graphical User Interface (GUI)

2024-12-25

▶ **Problem:** How do we actually program the graphical user interface?

[Program Solution]

This is what the control flow of LVGL looks like:



You create one display for each physical display panel (one vex brain), create screens on them, add widgets onto those screens. To handle touch, mouse, keypad, etc., you create an input device for each. The tick interface tells LVGL what time is it. Timer Handler drives LVGL's timers which, in turn, perform all of LVGL's time-related tasks:

- periodically refreshes displays,
- reads input devices,
- fires events,
- runs any animations, and
- runs user-created timers.

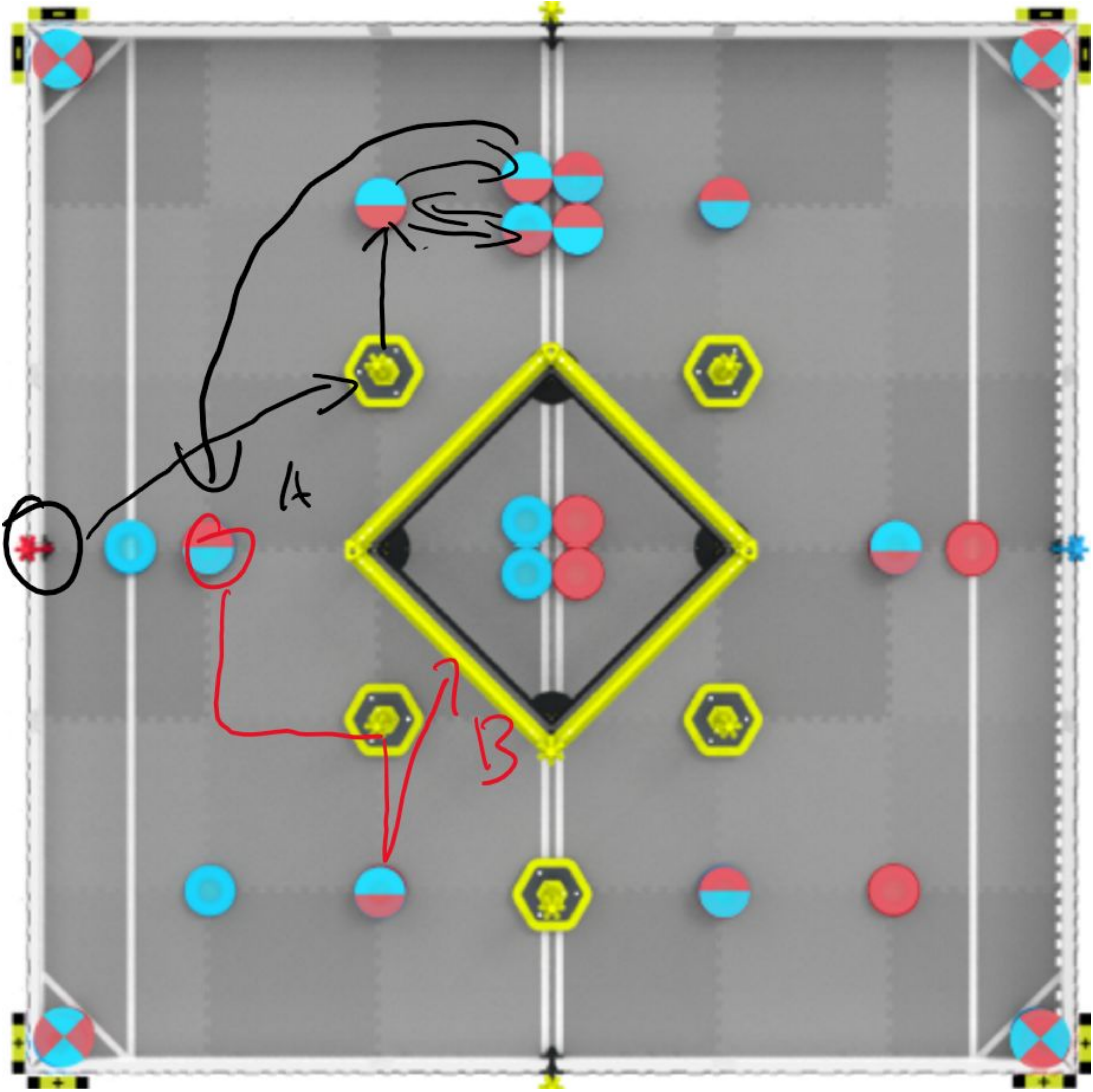
Designed by:
Adam X

Witnessed by:
Zechariah W

Autonomous Paths: Solo Win Point

2025-01-26

▶ **Focus:** How will we guarantee the autonomous win point?



Designed by:

Adam

Witnessed by:

Maxwell L

Autonomous Paths: Solo Win Point

2025-01-26

▶ **Focus:** Figure out what our eliminations autonomous path will be

[Plan a Solution]

For eliminations, we really are stressing on getting possession of the fifth goal on the autonomous line, since that is a huge decider on who wins matches.

Step	Route 1
A	Score the alliance stake, clamp onto the goal on the negative corner side, and score three rings on it. Then drive to the ring stack in the middle and drop the goal
B	Intake the ring on the top of the stack, clamp onto the positive corner side goal and score to rings onto it. Then go and touch the ladder

The methodology behind this path is that we've learned that getting the autonomous win point is so crucial to increasing our ranking. We also cannot control our alliance partner's capabilities, so it is best of we can secure the autonomous win point by ourselves, just in case our alliance doesn't have any autonomous.

Designed by:

Adam X

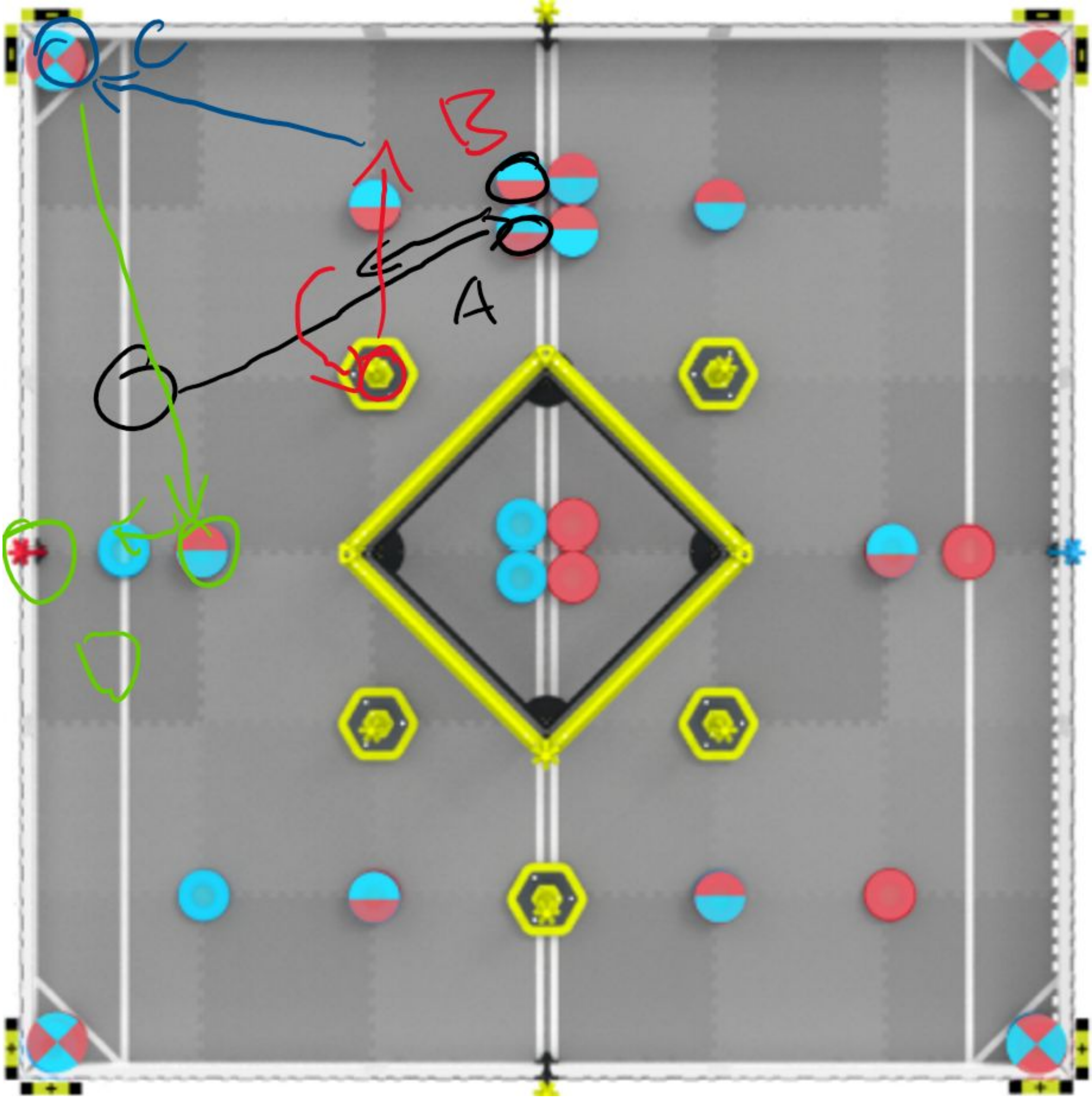
Witnessed by:

Kevin Z

Autonomous Paths: Ring Rush

2025-01-29

▶ **Focus:** Figure out what our eliminations autonomous path will be



Designed by:

Adam

Witnessed by:

Maxwell L

Autonomous Paths: Ring Rush

2025-01-29

▶ **Focus:** Figure out what our eliminations autonomous path will be

[Plan a Solution]

The eliminations path is for when our alliance prefers rushing the 5th goal on the positive side. Using our ring rush mechanism, we can quickly grab the two rings from the stack of 8 and pull them back

Step	Route 1
A	Immediately drive forwards, grabbing the two rings from the stack of 8, then drive back, lining them up with the stack of 2.
B	Clamp onto the goal and score all three rings
C	Drive into the corner, intaking one of the rings from the stack of 4
D	Drive to the middle of the field, intaking the preload that we left at our starting position and score it on the alliance stake

Designed by:

Adam X

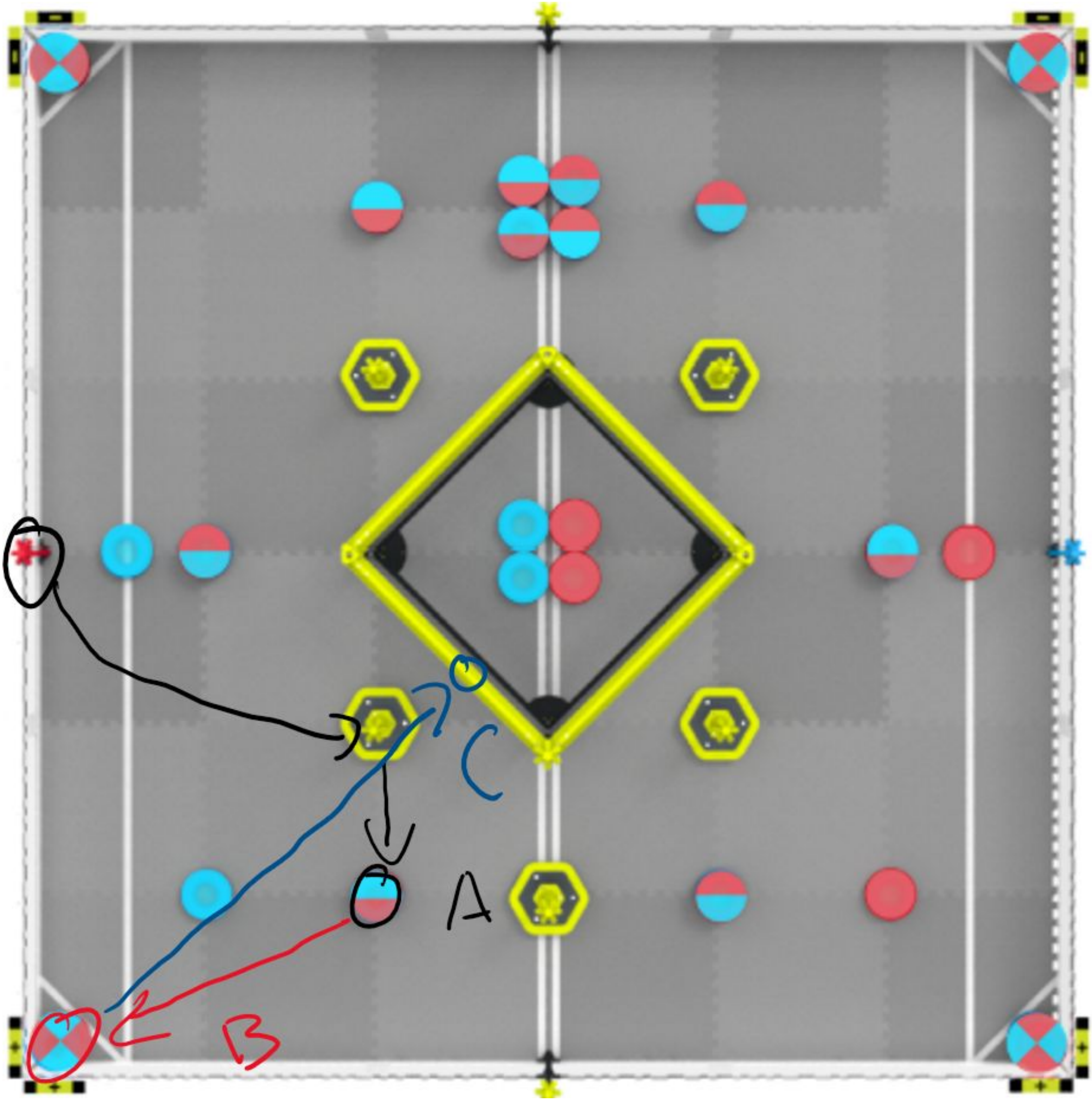
Witnessed by:

Kevin Z

Autonomous Paths: Safe Half Win Point

2025-01-29

▶ **Focus:** What can we do when our alliance partner has a decent autonomous?



Designed by:

Adam

Witnessed by:

Maxwell L

Autonomous Paths: Safe Half Win Point

2025-01-29

▶ **Focus:** What can we do when our alliance partner has a decent autonomous?

[Plan a Solution]

When our alliance has a decent autonomous, but doesn't get the full win point, we need to meet the remaining criteria and secure that win point

Step	Route 1
A	Score the alliance stake, clamp onto the goal, and score a ring from the 2 stack
B	Drive to the corner and intake 2 rings from the stack of 4 rings
C	Drive back and contact the ladder

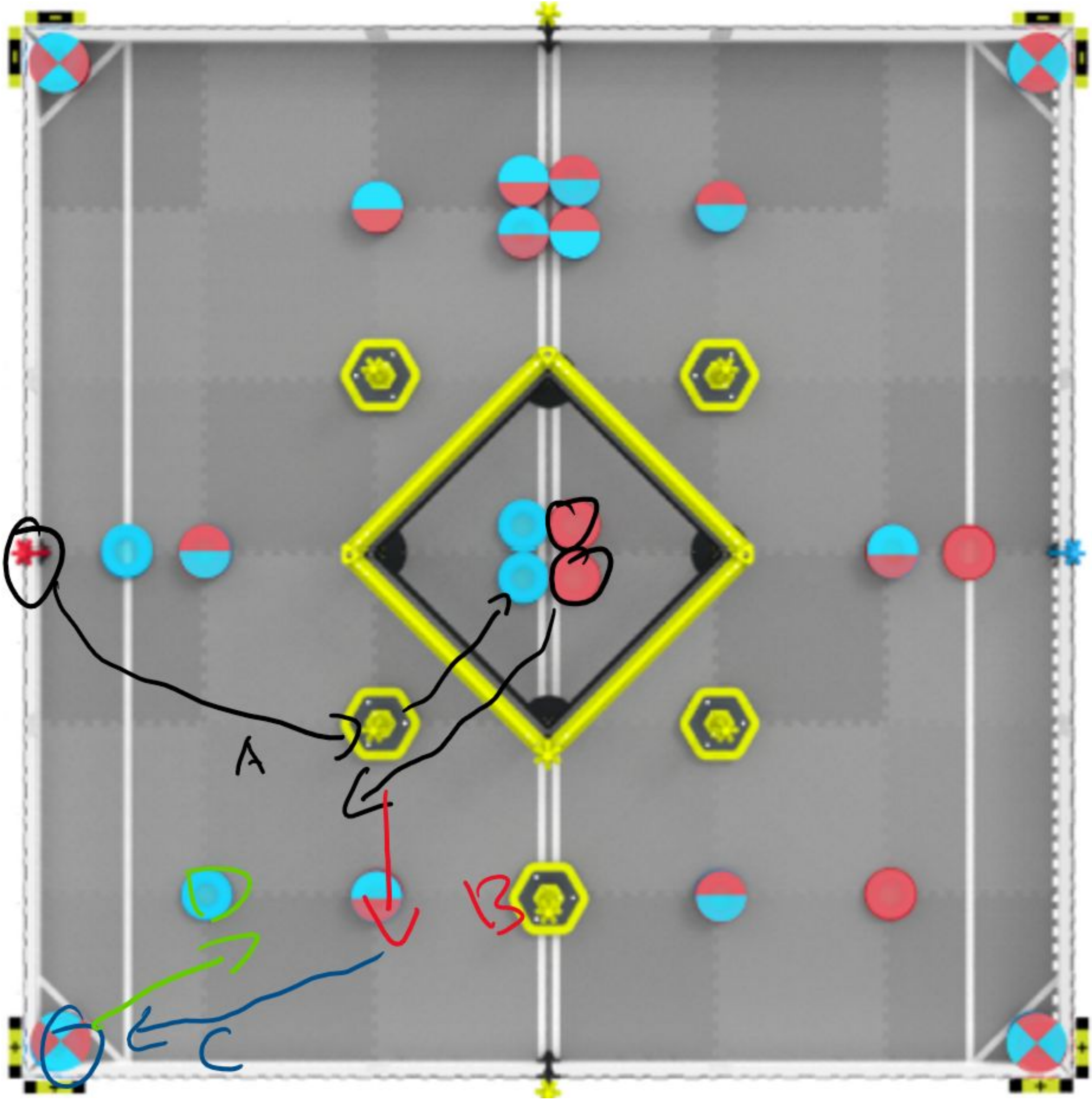
Designed by: Adam X

Witnessed by: Kevin Z

Autonomous Paths: 5-Ring Positive Side

2025-02-17

▶ **Focus:** Figure out what our eliminations autonomous path will be



Designed by:

Adam

Witnessed by:

Maxwell L

Autonomous Paths: 5-Ring Positive Side

2025-02-17

▶ **Focus:** Figure out what our eliminations autonomous path will be

[Plan a Solution]

As mentioned in the other logbook, we wanted to alliance with 210K for provincials, and since they have a ring rush autonomous on the negative side, this path matches up with theirs very well.

Step	Route 1
A	Score the alliance stake, clamp onto the goal, then using our two arm mechanism, take 2 rings from the other side of the autonomous line and pull them back, lining them up with the 2 stack of rings.
B	Score all three rings
C	Drive into the corner, intaking two of the rings from the stack of 4
D	Back out of the corner, dropping the goal, then orient the robot to attack the other positive corner or 5th mobile goal

Designed by:

Adam X

Witnessed by:

Kevin Z