JDHS FTC ROBOTICS

Team The Hydraulic Hydras #9384

2023-2024

TABLE OF CONTENTS

ABOUT US

The Hydraulic Hydras are a FIRST Tech Challenge team from John Dewey High School from Brooklyn, New York. Currently the team is made up of 5 students ranging from 10th-12th grade and 2 mentors. All of our students are from John Dewey High School aside from Aiden who is from Midwood high school located in Brooklyn New York. Our rookie year was in 2014. This team has evolved throughout this time leaving legacies behind allowing new members to learn from past mistakes and adapting older ideas from previous robots to newer robots.

As a team we embody FIRST core values in our everyday lives. Whether it be being gracious throughout our day or being professional while helping someone who needs it. Our team has set these core values to create not only a better work environment but to also have a safe space for all.

The Hydraulic Hydras may be an old team however all of the students on the team have only 1 year of FTC experience. Due to this this season was a big learning process for our team.

GOALS

Our goals for the future as a team is to stick together and create a safe environment for everyone that wants to join our team whether it be from the classroom of outside of the classroom. On top of this we want to create award winning robots that will make it to higher level games which include the super qualifier and world champs.

OUR JOURNEY

As a team of students coming from all walks of life, our Journey as a team was and still will be difficult. Due to individual problems, not having the funds to buy new parts and only having 3 months to build a fully functioning robot our team has made some serious progress. Due to our achievements throughout this build season as a team we feel like we have made something special. Not just a working robot but friendships that will last us all throughout our lives.

FINANCES

Throughout this season finances has been a major issue for our team. Budget cuts within the DOE has heavily affected our team making us unable to build our robot completely based on our cad since we were unable to afford materials that we need. Throughout all of this we were able to afford most of our materials through selling candy within our school and mentors chipping in.

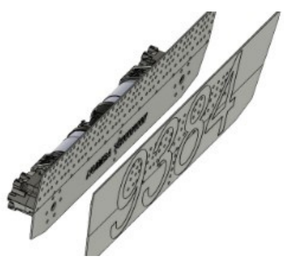
CHASSIS

Due to the limitation of pre-made parts from FTC vendors we decided to make our own custom chassis. This allows us to properly mount our mechanisms and wheels in the exact space we want to. We are unable to do that with pre-made parts. This chassis was originally made to learn how to use fusion 360 but worked for this years game. It will be around 17 x 17 x 4.25 inches tall. We began working on a chassis as an offseason project.

June 8

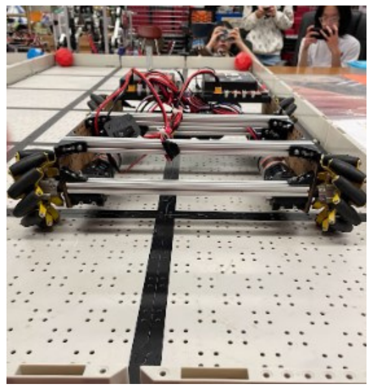
chassis

For our second design we revised the wheel placement. We put them further into the chassis allowing us to make side plates to completely cover the wheels allowing us to prevent accidental damage to the wheels from the sides and back of the robot



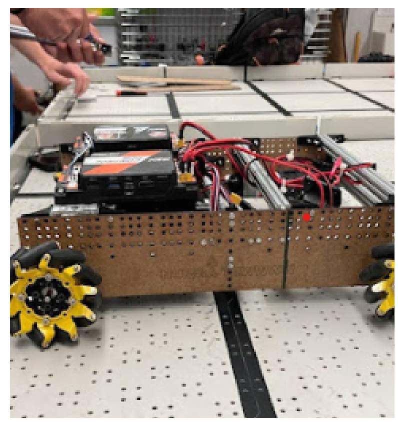
CHASSIS

Our first design, surprisingly was very good. The only issue we saw was with wheel placement since they poked out a bit.



June 9

Add a little bit of body text



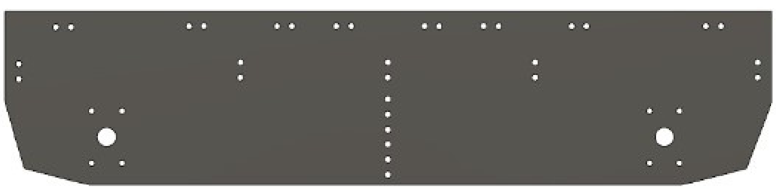
chassis

After creating the side plates we decided to add bearing blocks to hold up the wheels shafts perfectly straight.

June 22

chassis

Once figuring out where the extrusions will go we added holes for mounting.
The inner plate stayed the same

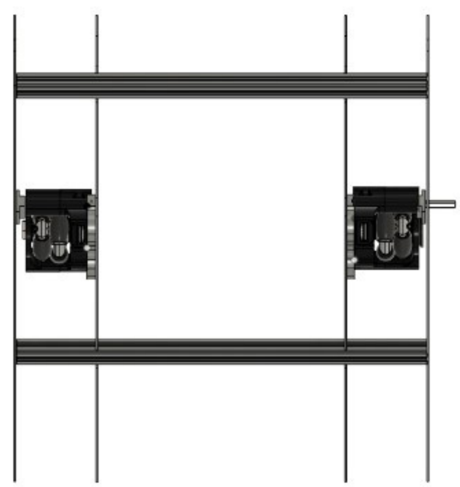


August 12

chassis

August 16

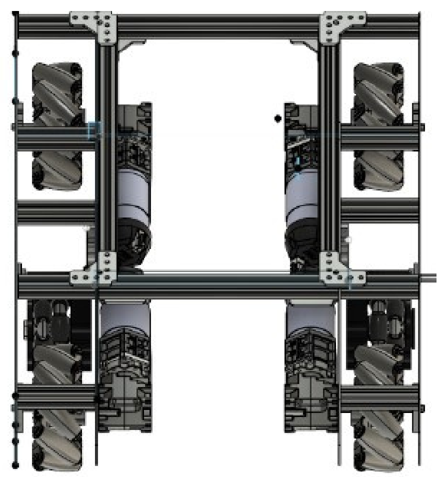
After the redesign we added 2 odometry pods to the inside of our chassis.



chassis

August 20

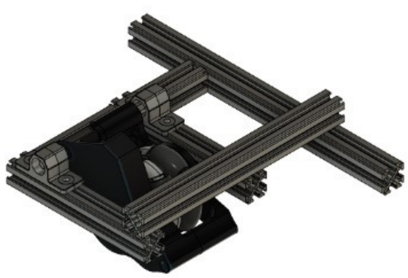
After noticing that the chassis can be more sturdy with extrusions hugging the inside of our plates we decided to add that to our cad. We also added extrusions for stability between the inner and outer plates.
On top of that we added our motors, 90 degree gearboxes, wheels and shafts.

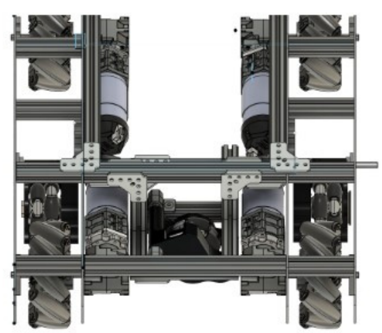


chassis

August 21

Once finishing the side plates we added the final odometry pod to the back of the chassis and added some metal brackets for stability.

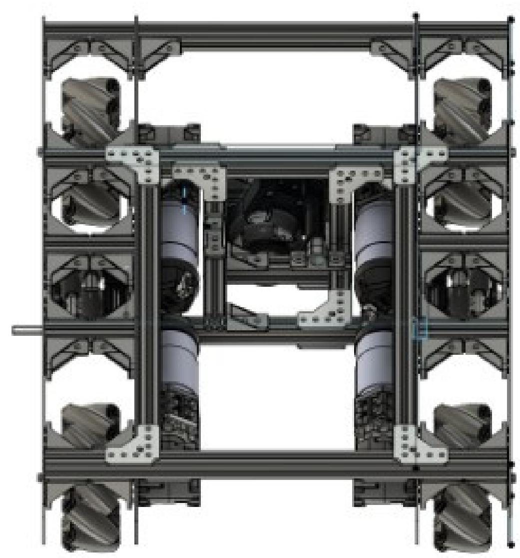




chassis

September 10

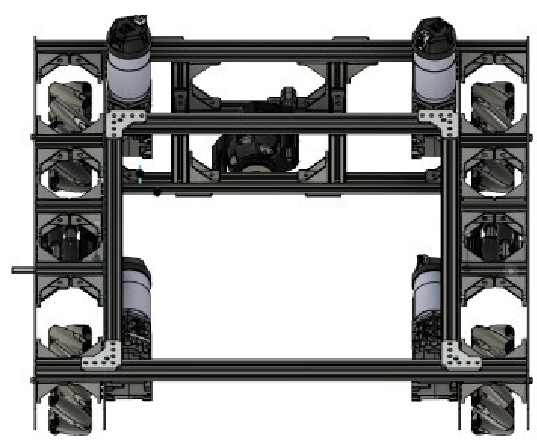
After seeing the game reveal and noticing that this chassis works with the game we decided to start the process of finalizing the design.First we added some more brackets to add stability to match what the final design of this chassis will look like on the field



chassis

September 12

After deciding on making a 4 bar linkage on our lift we moved the middle extrusion back as much as possible to account for the length on the 4 bar. We also moved the back 2 motors upwards.



September 12

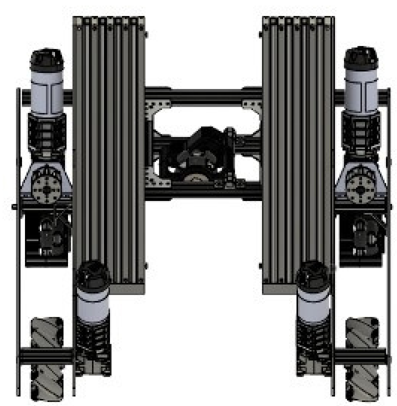
chassis

Due to the idea of making multiple designs of every mechanism, we decided to remix the whole design but keep the side panels the same. Motors and sensors will still be in similar areas but the extrusions and mounting hardware slightly changed.

chassis

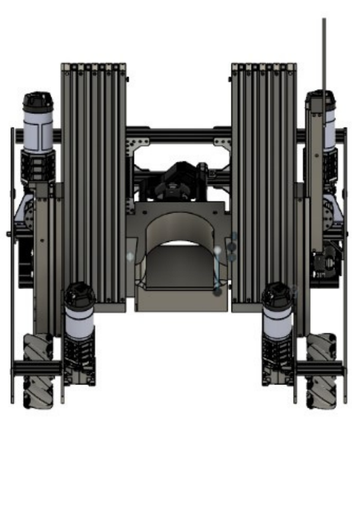
September 25

Once we finished the final dual stage climber we put it in the chassis CAD.



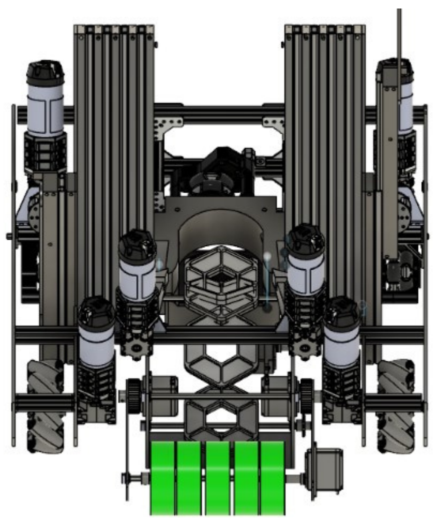
chassis

Once we finished the final lift we mounted it on the chassis and put holes for mounting. We also added the outtake mechanism for our robot.



chassis

October 1

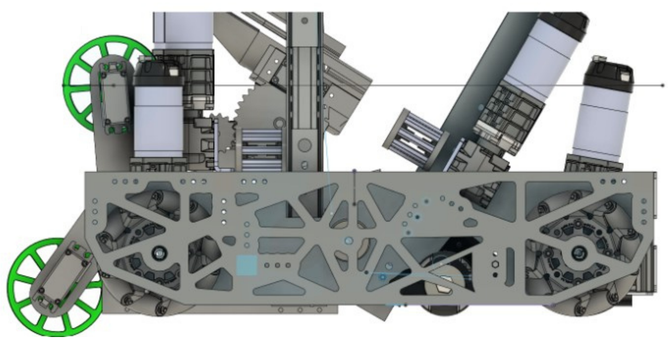


We added our intake to the chassis.

chassis

October 3

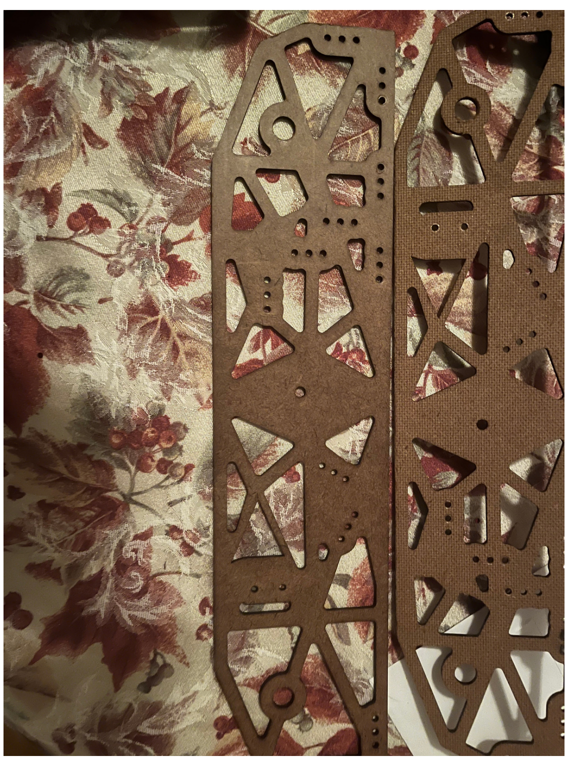
We cut holes in our outer plate to lower the weight of our robot being made of ⅛ thick aluminum.

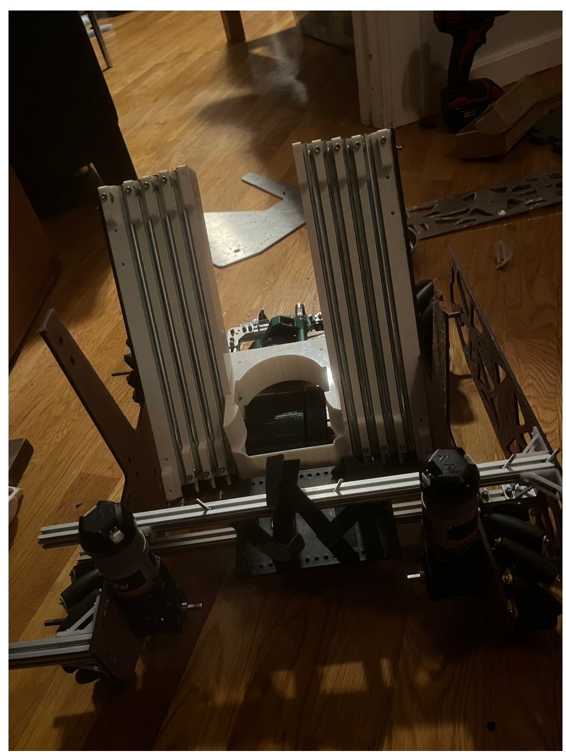


October 10

chassis

We cut out our chassis with the laser cutter using Masonite. This allowed us to see if our mounting solutions would work for our robot.





chassis

We cut out our chassis with our CNC machine out of aluminum. This is our final chassis for competition.

ODOMETRY MODULES

An odometry is a form of localization that uses external encoders to generate the position or pose of your robot. Odometry is really beneficial during the autonomous period as it can fine-tune the robot's traversal and the heading the robot keeps. We developed our own design and opted to use three odometry modules. We began working the modules as an offseason project.

Odometry Modules

July 4

resources

We decided to use this Odometry case due to the slick design and the built in holes for a suspension. These holes on the side will allow us to use springs to keep the wheel always on the ground.

Odometry Modules

July 5

Our first design was based off of a design on GrabCAD. We decided to use this design because it allowed for an easy way to extend the case itself. This allows us to use of a bigger omni wheel instead of the standard Rev omni wheel. The bigger wheel allows for better traction and overall movement.

Odometry Modules

July 16

We decided to add a bearing on the end of the odom case for smoother movement of the wheel.

Odometry Modules

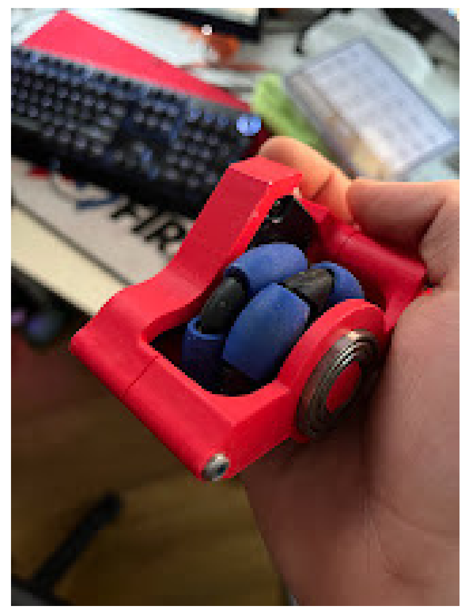
August 16

After adding the module to the chassis we noticed that the nuts and bolts sticked out of the case and caused issues for mounting. We decided to add holes for the nuts and bolts to stay in.

August 20

Odometry Modules

Since we finished the CAD, we decided to do a test print using a cheap material called PLA (polylactic acid). Since it worked perfectly and held up throughout all of our tests we decided to keep it for our final robot. We also added caps onto the bearings of our module to allow for dust to be kept out of them.

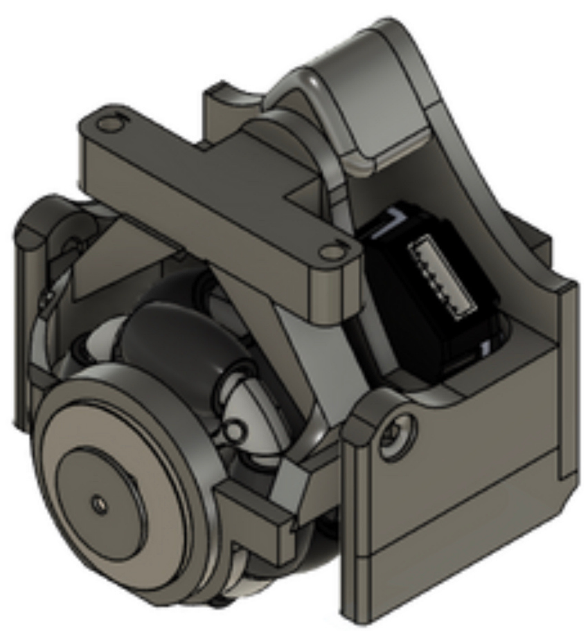


Odometry Modules

Due to the screws on the previous design being used for a pivot point, the screw would often bend and not allow for accurate readings. We decided to use a 5mm hex shaft instead. This reduces bending within the part and will be more stable overall.

Odometry Modules

due to failure with the last design for the odometry we created custom odometry pods to create suspension underneath the robot



Odometry Modules

we added springs to the custom odometry pods to help us create suspension

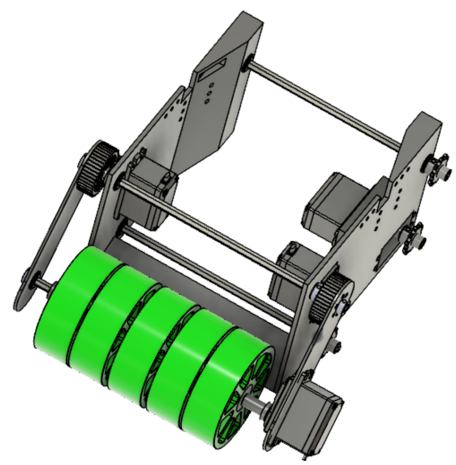
INTAKE

An intake is a mechanism used to take in game pieces. This year we needed to intake multiple game elements so we dicided to use an active intake also known as a motorized intake to spin the pixel into the robot. We would also like our intake to have the ability to lift up and intake the stacked game pieces.

September 25

Intake

We began working on the intake cad and we wanted to be able to increase and decrease the height of our roller wheels to be able to intake from the stacks during autonomous.



This servo will allow for the increase and decrease of height for our wheels.

Intake

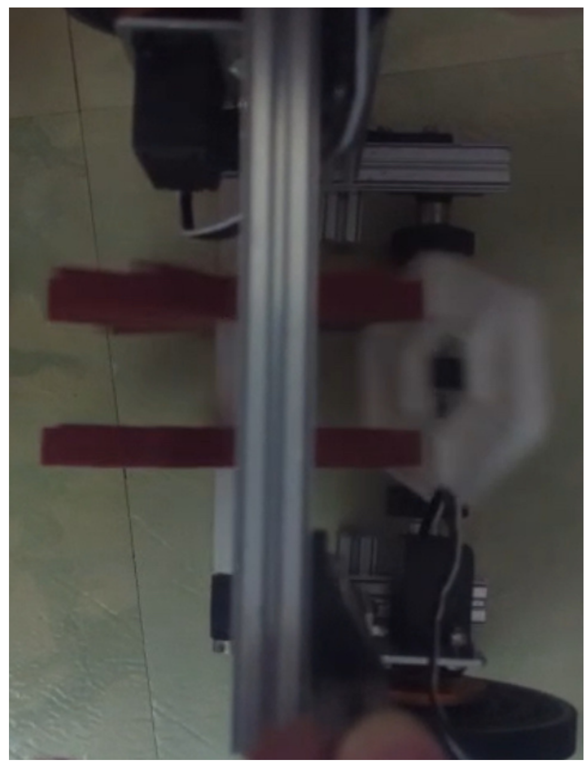
October 16

When prototyping the intake we noticed that the wheels were too heavy for the servos to keep up. This made us redesign the whole intake mechanism once we had the final chassis. We will make this new design have fixed rollers instead of the gummy wheel intake that can flip on an angle like we had before.

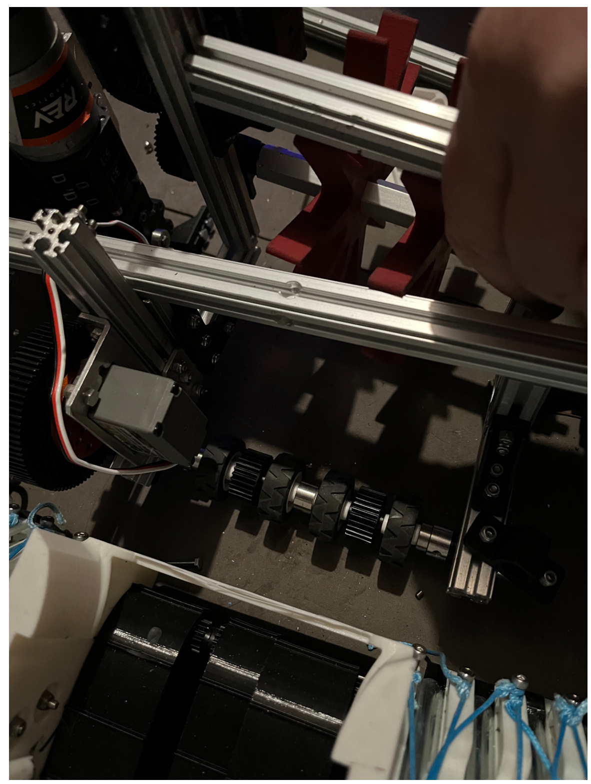
December 1

Intake

Since we had the chassis mostly assembled we decided to begin working on the new intake. This intake worked perfectly for what we are trying to do. This intake has 2 sets of wheels, one set of star wheels and another set of roller wheels. When working together they force a pixel into the rollers and shoot it into our chassis.



Intake

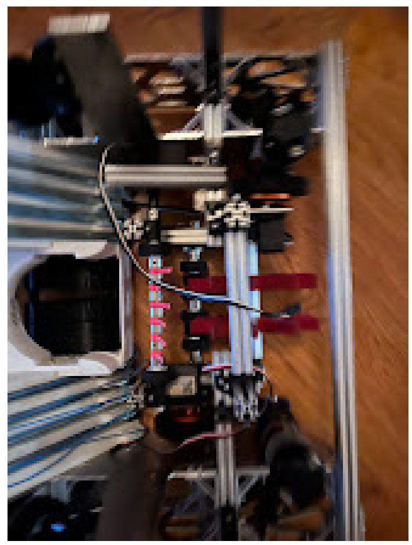


We mounted the intake on our chassis. It is not consistently intaking the pixel into our outtake. However it does intake it perfectly into our chassis.

Intake

December 5

We perfected the mounting of our intake to allow for it to precisely pick up the pixels. We added crossbeams with an opening for our star wheels to poke out of to allow for free movement. We also added another set of rollers using zip ties to force the pixel into the outtake.



LIFT\LINEAR SLIDE

A lift or linear slide is a mechanism that allows for linear movement in a certain direction. This year we needed to lift upwards and to do so we decided to make our own lifting mechanism. Our team used rev linear slides previously but this year we decided to use drawer slides as the base of our lift and 3d printed spacers to allow for mounting between the sliders. Aside from the drawer slides themselves, this design was made completely by our team.

October 14

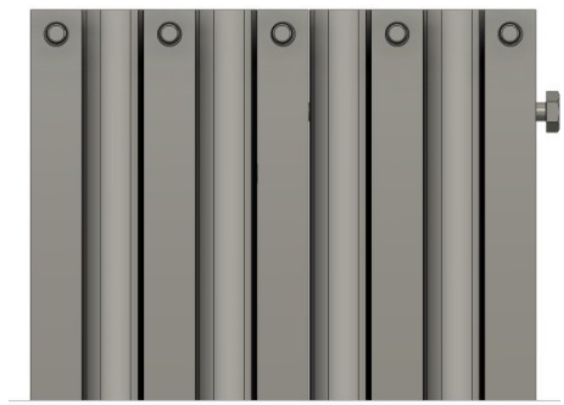
Linear Slides

This is the original version of our CAD and is very similar to the rev linear sliders (pulley mounting wise) but are way smoother. This design was not printed out due to reworking the way the pulleys were going to be mounted.

Linear Slides

October 16

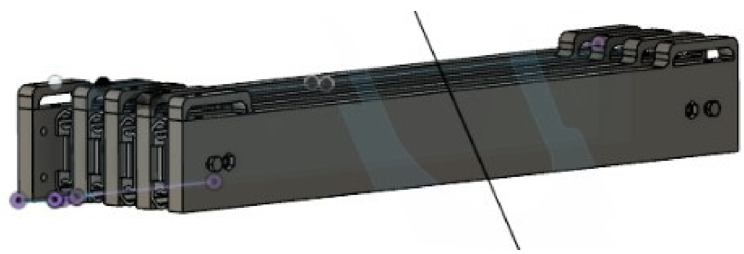
This version has the screw holes go all the way through the printed part. This allows for extra rigidity on the pulley. We also seen that we should add another printed part on the back of our slider since it could mount a string.



October 20

Linear Slides

We decided to add covers over our pulleys because we noticed that the string would fall out easily due to our sliders being at a 35 degree angle.



Linear Slides

October 29

We manufactured all of the linear slide parts and began assembling our slides. During this process we noticed that the hexagonal shapes that allow for the nuts to stay in place were not holding the nut in place. We resolved this error by making a hole instead of a hexagon to allow for a nut-driver to fit inside of the hole which can tighten the nut. On top of this we remade the brace that absorbs the impact of the lift and incorporated it into every stage of our lift.

Linear Slides

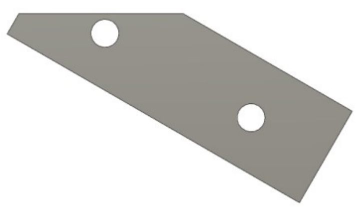
November 11

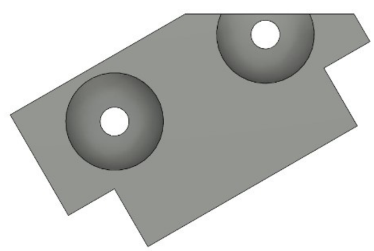
We manufactured the new 3d printed inserts and they bolted into the slides perfectly.

November 15

Linear Slides

While bolting our lift onto the chassis we noticed it was sliding on the floor of the field. We seen that it was because the shock absorbing pieces at the bottom of our lift were not high enough.

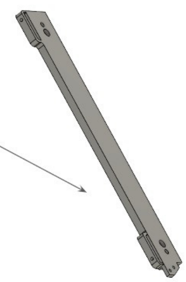


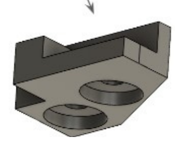


Linear Slides

November 22

After running the robot with the lifts and outtake on, we noticed that it would touch the field. This led us to redesigning the way our tpu braces and the bottoms of our lift.





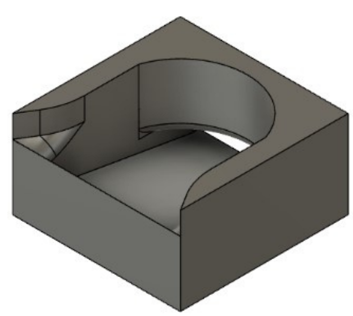
OUTTAKE

An outtake is a mechanism that allows for a game piece to be brought out of the robot and scored. Since we needed to outtake 2 game pieces we opted for a mechanism that can hold 2 game pieces and drop them almost simultaneously. To do this we are using an active outtake. This mechanism uses a 3d printed conveyor belt to convey the pixel onto the backdrop. Everything within this mechanism is 3d printed aside from the servo motor, gears and hex shaft. Before this we used a simple ramp design.

Outtake

Due to this years game allowing us to hold 2 game pieces at once, we wanted to make a design that would allow for us to outtake 2 gamepieces quickly. The first idea that came to mind was a box that had a ramp which would allow for the game piece to passively be scored. This design would have been completely 3d printed.

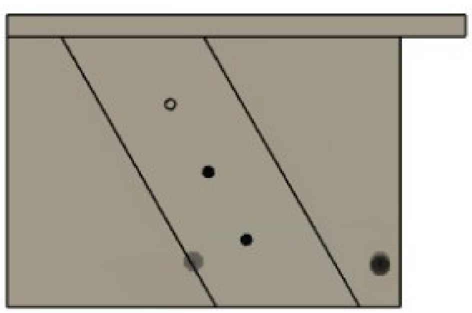
September 22



September 25

Outtake

We added mounting holes for a lid and holes on the side for it to be mounted onto the final stage of our lift.



Outtake

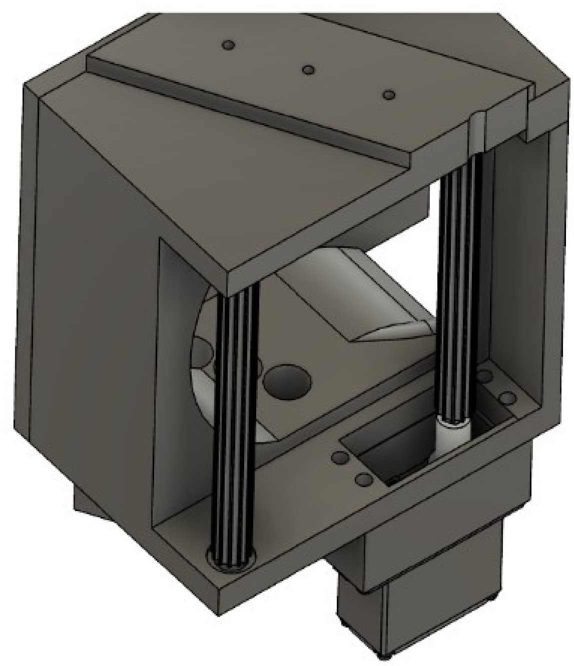
September 28

This outtake design did not work when we tested it because the hexagons did not move down the slope. We decided to make it an active intake with the motor pushing the pixels into the backdrop. This failure made us rethink how we could incorporate a scoring mechanism into our robot

October 2

Outtake

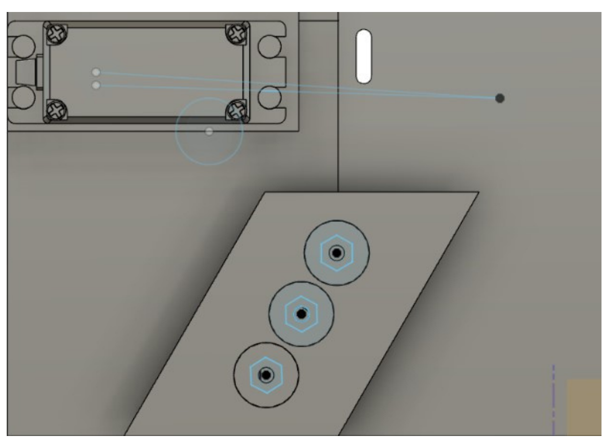
Since our outtake didn't work, we decided to add a conveyor belt to it. This will move the game pieces out of the outtake. It will be completely 3d printed aside from the hex shafts.



October 4

Outtake

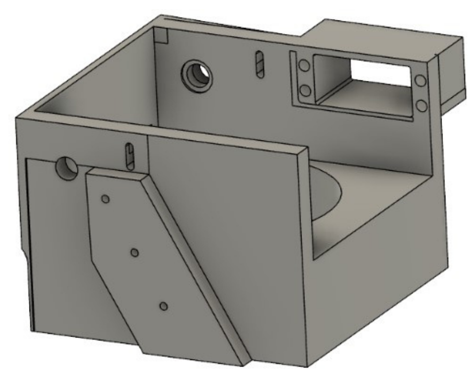
We added a belt tensioner
to the box.



Outtake

October 7

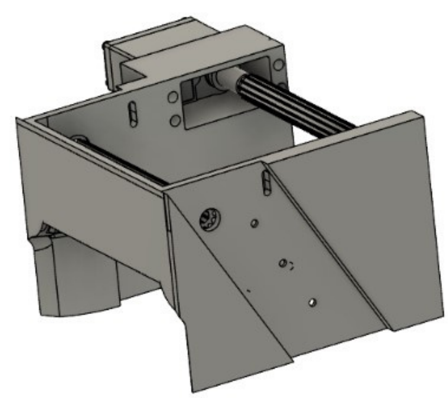
We noticed that we were unable to put the hex shaft inside of the outtake. We added holes to slide in the hex shaft into our outtake. Bearings will be placed inside of these holes to make the conveyor belt move smoother.



October 21

Outtake

After assembling everything we noticed the outtake slightly pushed the lifts too close together. Since the inside of our lift needs a little bit of space we added slots in our outtake for our final stage lift to bolt onto. We can use washers or spacers to increase the space between the outtake and lifts if we need to.



Outtake

We added a ramp onto the outtake to help with the pixel getting into the outtake from our intake.

December 8

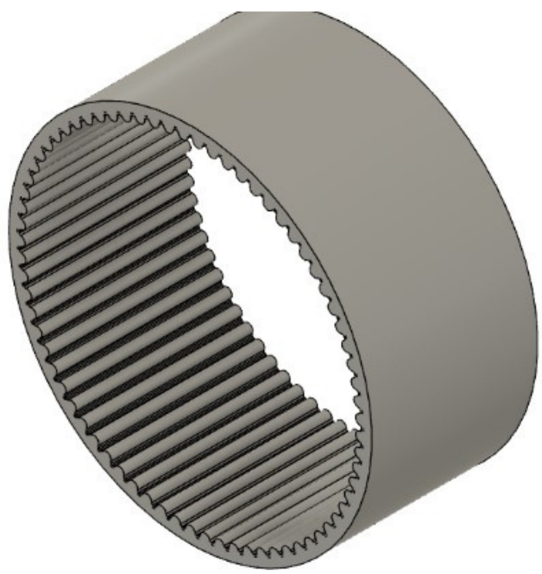
OUTTAKE BELT

Due to us using a belted outtake we decided to 3d print the belts. At first this seemed very difficult but it has been working fine throughout all of our tests. This belt is made out of 95A TPU (Thermoplastic Polyurethane) and is extremely flexible. Due to this, it is perfect for belts that aren't under heavy load. Once under heavy load the belt begins to warp making it less precise.

Outtake Belt

October 3

We CADed out the first belt and printed it using 95A Tpu (Thermoplastic Polyurethane). This filament is flexible and allowed for the belt to be properly made.



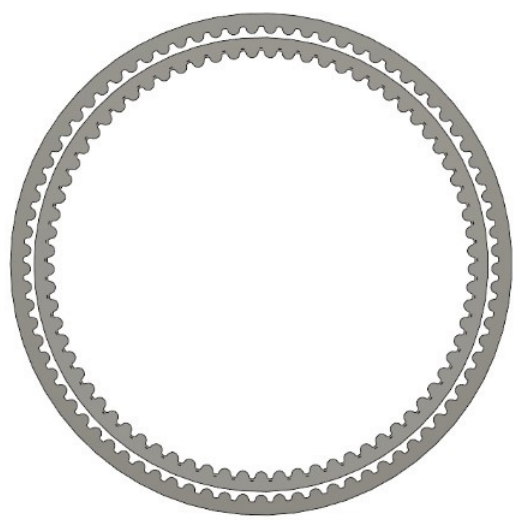
Outtake Belt

October 8

Once printed we assembled it and noticed that it provides too tight of a fit on our gears. The image shows that the belt as too tight which led it to become deformed. This not only harmed the belt but it harmed the servo as well.

Outtake Belt

Our second belts diameter is slightly bigger to make sure it doesn't strain the motor. Its outer diameter is increased by .275 inches.



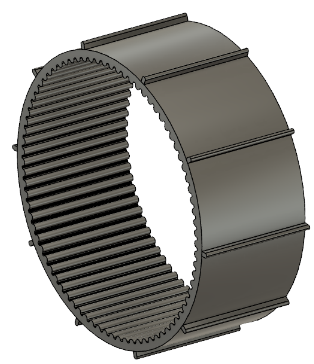
Outtake Belt

Once assembled we noticed it was slightly too big. This ended up being a good thing because once the servo spun it allowed for the belt to contract and allow for game pieces to be delivered. When it was off or pushing the other way, the belt expanded and blocked game pieces from falling out of our outtake.

November 12

Outtake Belt

We decided to add small riqges on the top oif our belts to allow for them to catch onto the bottom of the game piece.



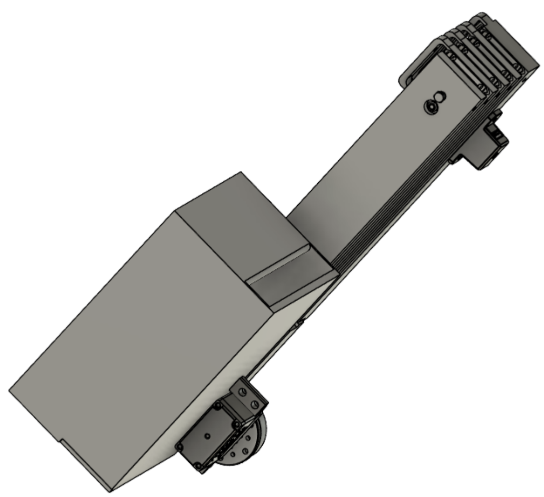
December 10

Updated Outtake

After the failure of our outtake during our first competition we noticed that we need to redesign our outtake. During this time we went back to our freight frenzy robot for inspiration and noticed that the scoring mechanism we used at that time could work for this years game.

Updated Outtake

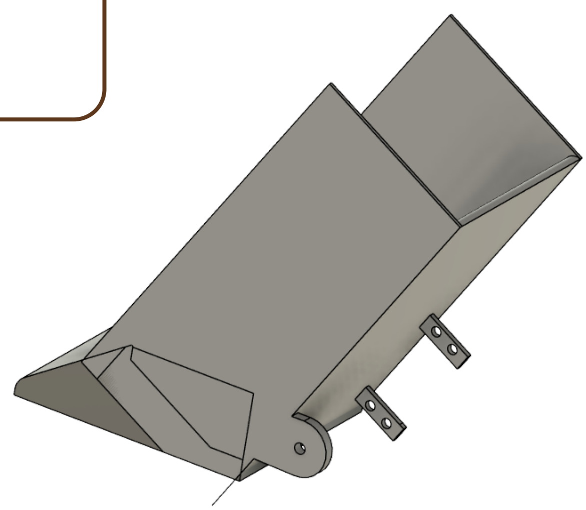
We created the base of the outtake by adding mounting solutions for a servo and a spot for 2 pixels to rest. When we want to score we rotate the whole box allowing for it to score on the backdrop.



Updated Outtake

December 12

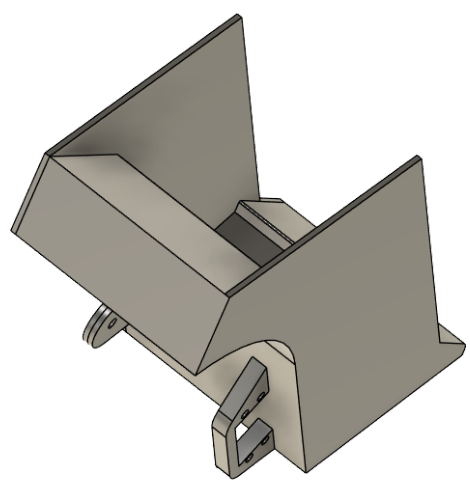
After noticing that this design will cost a lot of material to manufacture we changed the design to allow for it to be made with less filament and to have a more astatically pleasing design. We also noticed that without an idle bolt on our second pair of lifts the whole outtake would sag



Updated Outtake

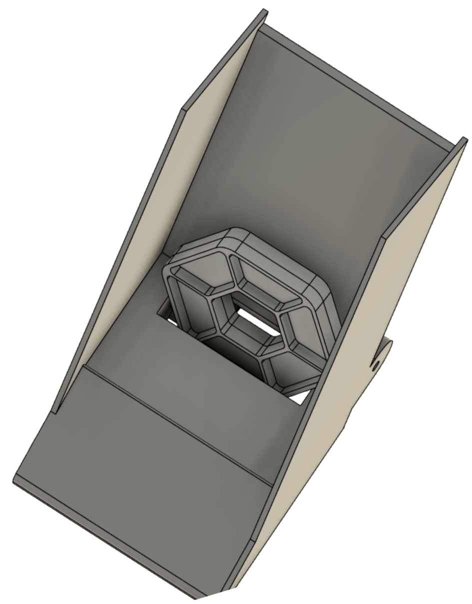
After testing our outtake we noticed that the servo mount needed to be more rigid and that there needs to be a curve for the pixel to follow when being deposited. We solved these issues by making a thicker servo mount and creating an arch to guide the pixel.

December 19



Updated Outtake

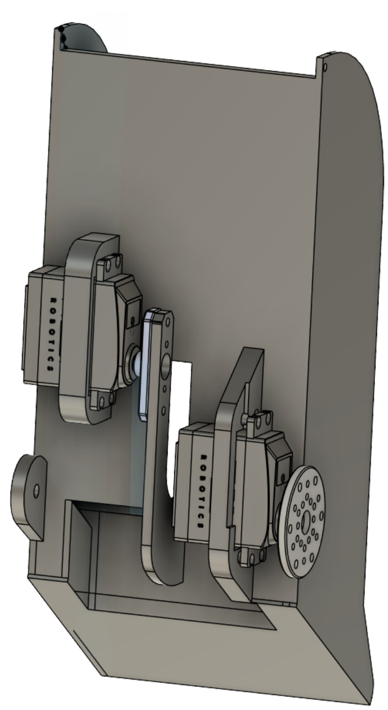
When testing the new design we noticed that the arc we made needed to be flexible. Due to this we reverted back to the previous design but kept the new servo mount and changed the inside of our box. We achieve this flex by using zip ties to guide the pixel off of the outtake. The inside of the box is also slightly changed because there is a hole there. This hole allows for one pixel to be set lower then another. This can help our intake drop one pixel at a time.



December 23

Updated Outtake

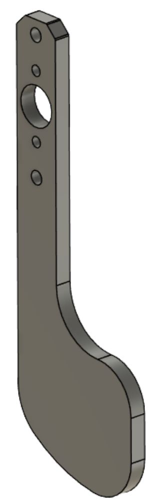
After printing out the outtake we still seen issues with consistency when it comes to our outtake dropping pixels independently consistently. We resolved this issue by adding another servo that can hold down the pixel through the use of a finger.



December 27

Updated Outtake

We updated the finger to have a smaller form factor. It is also more curvey.



Climber

We updated the finger to have a smaller form factor. It is also more curvey.

Climber

December 27

We updated the finger to have a smaller form factor. It is also more curvey.

Climber

December 27

We updated the finger to have a smaller form factor. It is also more curvey.

Climber

December 27

We updated the finger to have a smaller form factor. It is also more curvey.

Climber

We updated the finger to have a smaller form factor. It is also more curvey.

Climber

December 27

We updated the
finger to have a
smaller form factor.
It is also more curvey.

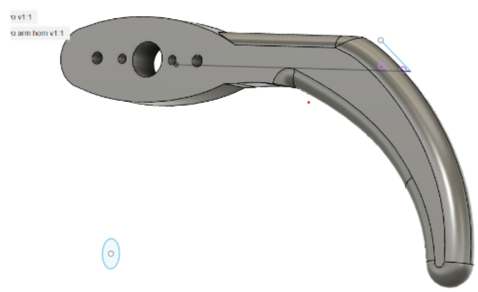
Pixel Claw

Due to the design of the chassis there was limited space and intake thats unable to extend. Making the bot struggle with the stack especially in auto. So new method of accessing the stack was needed which began with something at first resembling a finger. Thus the servo hook attached to the intake had to go through some design changes such as figuring out the location of the servo mount and angles the hook needed to not interfere with any other mechanism on in the front of the chassis.

January 23-24

Pixel Claw

The first design was just to see if the idea would be possible. When mounted it showed some potential but the angles of the tip limited most of the little function it had.



Pixel Claw

January 24-25

With version 2 the servo mount was moved to the front right of the chassis as it was interfering with the supports around it. This forced the hook to be extended in many directions to avoid the intake and drone launcher which lead to this. In testing it performed well except for only being able to reach one pixel and being extremely brittle. This lead to the base near the servo horn to break into 3 pieces .

Pixel Claw

January 27-28

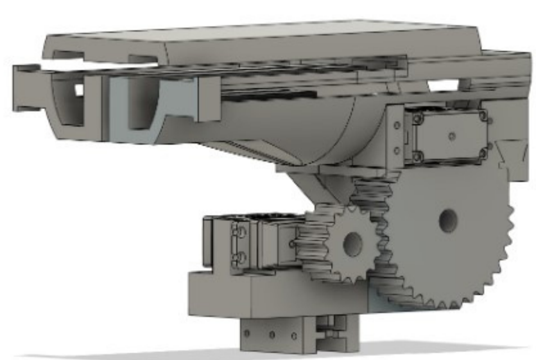
Version 3 is a continuation of version 2 extending the tip to reach the second pixel. While having a hollow section running from the base to the start of the hook to allow for the installation of a brass rod to reduce the chances of the shaft breaking.

Drone Launchers mrks 1-12

Drone launchers designated as MRKs 1-6 are equipped with dual sets of racks. The lower rack is designated for securing the rubber band, while the upper rack is intended for securing the aircraft. A servo mechanism located in the upper rear section is responsible for initiating the release, complemented by two horns situated at the front which are responsible for tensioning the rubber band.

Mrk 1

Mrk 6



Launcher mrk 1

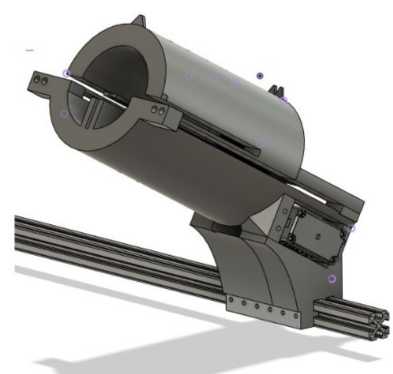
September 12 -14

Version 1, was designed primarily for as a concept. Hence, it features a simplified configuration, with a fixed angle, a rev servo, and a basic rack structure to accommodate the drone

Launcher mrk 2

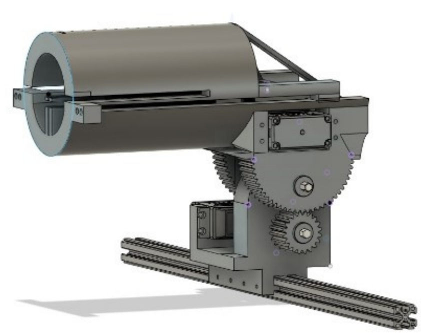
Version 2 is based on the design of Version 1, with the addition of a protective cover. This serves to prevent any potential loosening or damage to the plane, increasing the security and durability of the launcher.

I



Launcher mrk 3

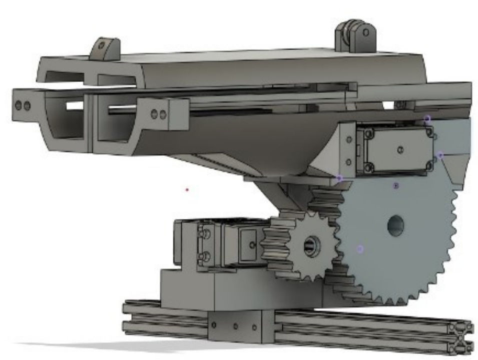
Version 3 represents a stopgap from Version 2. It introduces an angle adjustment system through a set of gears actuated by a rev servo in the base. Version 3's cover was 3d printed. However, the main flaw of this version is a flimsy cover that broke apart before testing



September 26

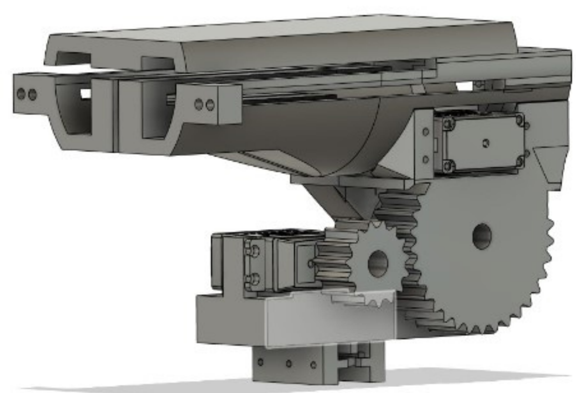
Launcher mrk 4

Version 4 signifies a redesign aimed at optimizing material usage, thereby enhancing durability and reducing spatial footprint. To achieve this, the servo previously located directly under the gear set was moved to the front of the primary gear assembly. Lastly, gear tooth dimensions were adjusted, lowering the risk of gear skipping.



Launcher mrk 5 pg#1

Version 5 builds on the design of the MRK 4 design, engineered to be more straightforward printing process with reduced reliance on supports. By optimizing the design for enhanced printability and minimizing the need for supports, version 5 achieves higher durability then the previous versions



Launcher mrk 5 pg#2

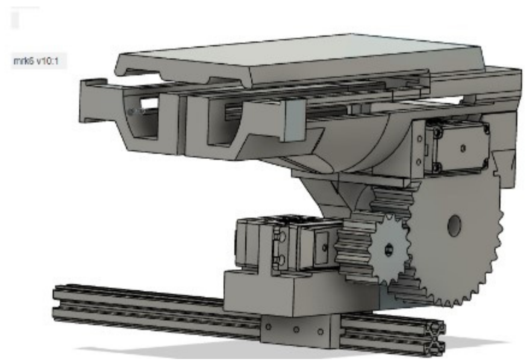
Version 5 is the first fully 3D-printed and prototyped launcher within the series. This prototype showed the concept is able to function successfully launching the plane to a height over 30 inches at about a 45-degree angle. Additionally, the launcher achieved a distance ranging between 6-8 feet.

Some failures of the design were:
Limited access to the upper servo.
Long resetting time.
Unable to bolt in the upper servo.
Low print tolerance causing fitting issues

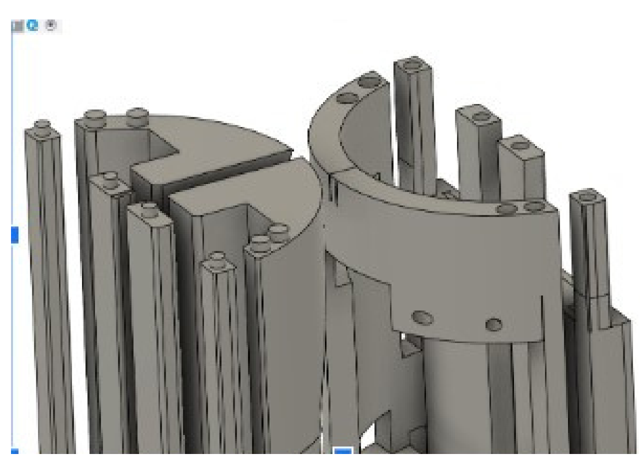
s

October 2-5

Launcher mrk 6 pg#1



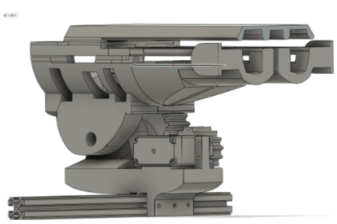
Version 6 represents the second prototype , incorporating modifications to enhance accessibility and effecenticey . Notably, this iteration features improved accessibility to the upper servo. Moreover, the addition of shields around the horns mitigates the risk of misfires. To optimize functionality, the upper section is extended, enabling heightened tension. Streamlining assembly, pins have been added. Additionally, a hex shaft is introduced within the lower servo gear, increasing efficiency.

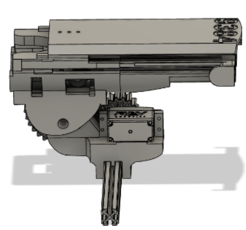


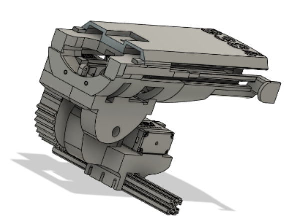
Launcher mrk 7-9

October 11-19

Moderate changes such adjusting the size and shape of the base and access opens are seen in the 3 versions with testing being done with the previously printed launchers v7’s base prevented any angle above 30 degrees. V8 adds a extended servo access, more efficient base, and bolted roof guard with the roof sect having issues with bolt length. V9 builds on these concepts with a dented roof sect to prevent bolts from remaining in the open lastly its base can be easily slid of an extrusion from above by loosening a few nuts



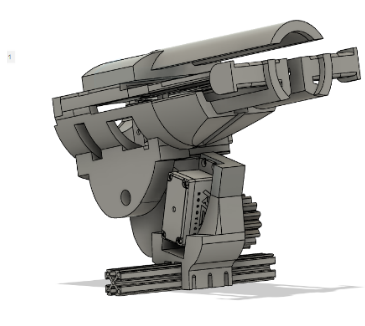


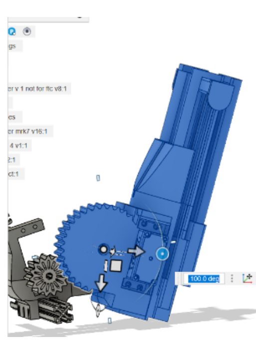


October 20-22

Launcher mrk 10

In version 10 base is modified by rotating the servo 80 degrees to save space while being able to rotate over a 100 degrees while some other changes include rounding the roof sect to reduce friction. In testing it was able to consistently launch the drone around 12 feet





November 3

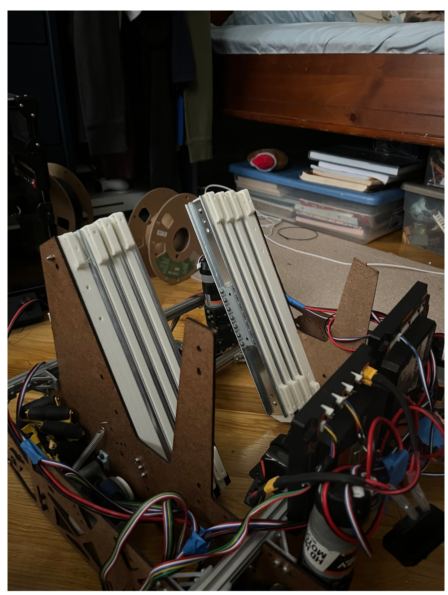
Assembly

We put the wooden chassis together with our odometry sensors to see if all of the spacing is correct. It was.

November 5

Assembly

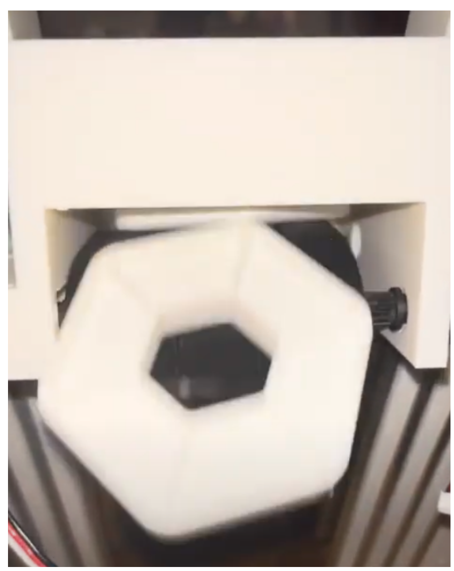
We began to put our lifts onto our wooden prototype chassis to see if it would mount onto the chassis properly. It was perfect.



November 21

Assembly

We began to put the final stage of our lift that has the outtake attached to it. Once mounted we seen if it could score on the backdrop and it did.



Assembly

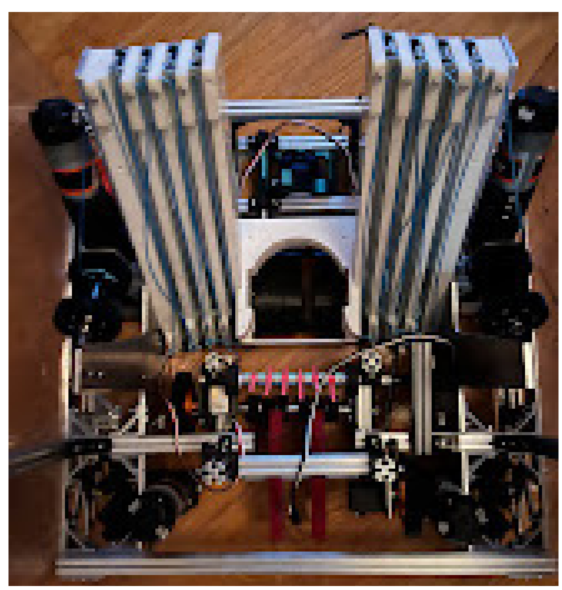
We got the all of the aluminum plates cleaned up and tapped. We decided to assembled the whole robot. We also added Lexan pieced that start at 11.5 inches tall and curve up to 11.75 inches tall too allow for the Lexan to touch the stage door instead of our lifts.

November 27

Assembly

December 7

We completely finished the intake and a spot for our launcher to sit on we also planned to mount the electronics onto our robot.



TEAM PROP

Using color, the team prop will get detected but the spike mark will get detected as well. This will mess up our autonomous due to the object was randomized and placed. Detecting the shape is more accurate as it will always scan for the precise shape. Due to this, our programmer tea opted for a prop with many different symmetrical shapes so our camera can scan our prop based on the shape of it. This allows us to get a very precise autonomous scanning.

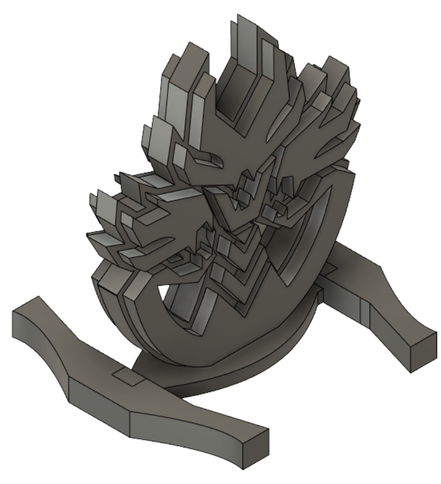
November 2

TEAM PROP

Our original team prop was made using our logo from previous years. The shape of it allowed Tea to tune it for our autonomous.

TEAM PROP

Our new team prop uses our new logo so teams know it is our prop by just looking at it.



December 4

March -
December 9

PROGRAMMING

INTRODUCTION

We initiated our project by coding and testing the drivetrain to assess its speed capabilities. In the first week, Tea, our lead programmer, began crafting a specialized programming library aimed at enhancing the intricacy of our code. This custom library, currently in the developmental phase, has been a long-standing side project for Tea since the commencement of his senior year. Dubbed "Hydraulic Lib," this library is slated for public release once it completes beta testing. Its purpose extends beyond our team, as it is intended to serve as a valuable resource for both newcomers and seasoned members of any robotics team worldwide. The scope of the library is not limited to our team alone, emphasizing its potential to contribute to the improvement of programming practices across the global robotics community.

PROGRAMMING
HYDRAULIC LID V1

The genesis of Hydraulic Lib traces back to its initial manifestation as a set of rudimentary utility files designed to simplify the coding process for our programmers. Initially conceived as mere auxiliary components, the project underwent a transformative evolution when our lead programmer, deeply immersed in Java programming, recognized the efficacy of consolidating disparate projects into cohesive files. Subsequently, a concerted effort was made to explore the nuances of creating custom libraries.

This journey commenced in March 2023, and over the course of several months, culminated in the development of the inaugural library in its beta version by September 2023. The intricate process involved meticulous research and dedicated effort on the part of our lead programmer. It is noteworthy that the library's evolution is an ongoing endeavor, with our lead programmer and other team members empowered to contribute enhancements and updates. This collaborative approach ensures that Hydraulic Lib remains a dynamic and continually refined resource for the programming community.

March -
December 9

March -
December 9

PROGRAMMING

HYDRAULIC LID U2

Within Hydraulic Lib, an encompassing repository of mathematical equations has been meticulously formulated. This library not only houses these mathematical constructs but is also equipped with an evolving pathing sequence. This sequence, currently in the development phase, facilitates the creation of a robot's path through the issuance of commands and trajectories. Importantly, it enables curved and spline movements while concurrently preserving a consistent or adaptable heading.

Moreover, Hydraulic Lib boasts a bespoke PIDF Control system—Proportional Integral Derivative Control. This system is designed to intricately fine-tune a robot's drivetrain or mechanisms. Its versatility extends to various drivetrain configurations, including X-drive, Mecanum, and Differential Swerve drive. The controller encompasses kinematics tailored to different drivetrains, thereby offering a comprehensive solution for precise control across diverse robotic platforms.

PROGRAMMING
OBJECT DETECTION

March - December 9

In our autonomous mode development, we employed TensorFlow object detection for training our object, a process that consumed approximately 2-3 hours but yielded highly accurate results. Subsequently, we seamlessly integrated our trained object into our Java code. This integration, coupled with a distinct enum class, enabled us to ascertain the placement side of the Team Prop during the match randomization process. During the initialization phase, our robot initiates the camera stream and, leveraging the trained object and enum class, accurately determines the location of the Team Prop. This streamlined approach ensures precise and efficient identification during the early stages of the match.

PROGRAMMING

DRIVE TRAIN

March -
December 9

In configuring our drivetrain, we opted for a mecanum drivetrain design. To achieve omnidirectional movement, we developed a bespoke code specifically tailored for the drivetrain. This custom code employs linear algebraic equations to enable the drivetrain to navigate along omnidirectional paths with precision and agility. Using this custom code we could also control the speed of our robot while driving as the program contains a method that creates speed states for our robot.

```java
double [] wheelspeeds = new double[4];

// Denominator is the largest motor power (absolute value) or 1
// This ensures all the powers maintain the same ratio, but only when
// at least one is out of the range [-1, 1]
double denominator = Math.max(Math.abs(y) + Math.abs(x) + Math.abs(rx), 1);
double leftFrontSpeed = (y + x + rx) / denominator;
double leftRearSpeed = (y - x + rx) / denominator;
double rightFrontSpeed = (y - x - rx) / denominator;
double rightRearSpeed = (y + x - rx) / denominator;

if (gamepad1.left_bumper) {
    powerMultiplier = 0.5;
}   else {
    powerMultiplier = 1;
}

wheelspeeds[0] = leftFrontSpeed * powerMultiplier;
wheelspeeds[1] = leftRearSpeed * powerMultiplier;
wheelspeeds[2] = rightFrontSpeed * powerMultiplier;
wheelspeeds[3] = rightRearSpeed * powerMultiplier;

ws[0] = wheelspeeds[0]; // left front
ws[1] = wheelspeeds[1]; // left rear
ws[2] = wheelspeeds[2]; // right front
ws[3] = wheelspeeds[3]; // right rear
```

March -
December 9

PROGRAMMING
LINEAR SLIDES

The cascade sliders in our setup underwent meticulous mathematical and coding considerations to ensure precision and delicacy in their operation. Although their physical manifestation may seem like a straightforward motor for vertical movement, our code transforms them into Motion Profiled motors. Motion profiles, in this context, delineate a trajectory for our reference throughout a specified time interval. At each point within this interval, the motion profile dictates the desired reference position. Leveraging motion profiles, our cascade sliders execute a controlled ascent during the autonomous phase, facilitating accurate scoring on the designated backdrop.

March -
December 9

PROGRAMMING

ODOMETRY MODULES

In the autonomous phase, our odometry modules undergo calibration to convert encoder ticks into inches. This conversion is crucial as it enables our robot to accurately record its position in the (x, y) format. As the robot maneuvers, the (x, y) position dynamically changes based on our initialization parameters. Utilizing the recorded positional data, we systematically build a trajectory for the robot. This trajectory is formed by connecting waypoints, and it facilitates both linear and splined movements. Consequently, our robot achieves asynchronous and precisely controlled motion, enhancing its overall performance during autonomous operations.

March -
December 9

PROGRAMMING

THE JOURNEY

Throughout our study and research endeavors, we diligently documented our insights in a compact journal, a project spearheaded by our lead programmer. The objective is to complete this journal by the conclusion of the current season. Within its pages, a comprehensive repository of knowledge unfolds, encompassing our learnings in both robot programming and broader software engineering principles. Notably, the journal explores the transformative potential of software engineering in diverse aspects of people's lives.

As the current season unfolds, we have plans to conduct online workshops aimed at imparting the fundamentals of Java programming to young audiences. This initiative reflects our commitment to knowledge-sharing and community outreach. Additionally, the journal will serve as a valuable legacy, passed down to incoming recruits who aspire to embark on their programming journey. This transfer of knowledge ensures a seamless transition and contributes to the ongoing growth and development of future team members.