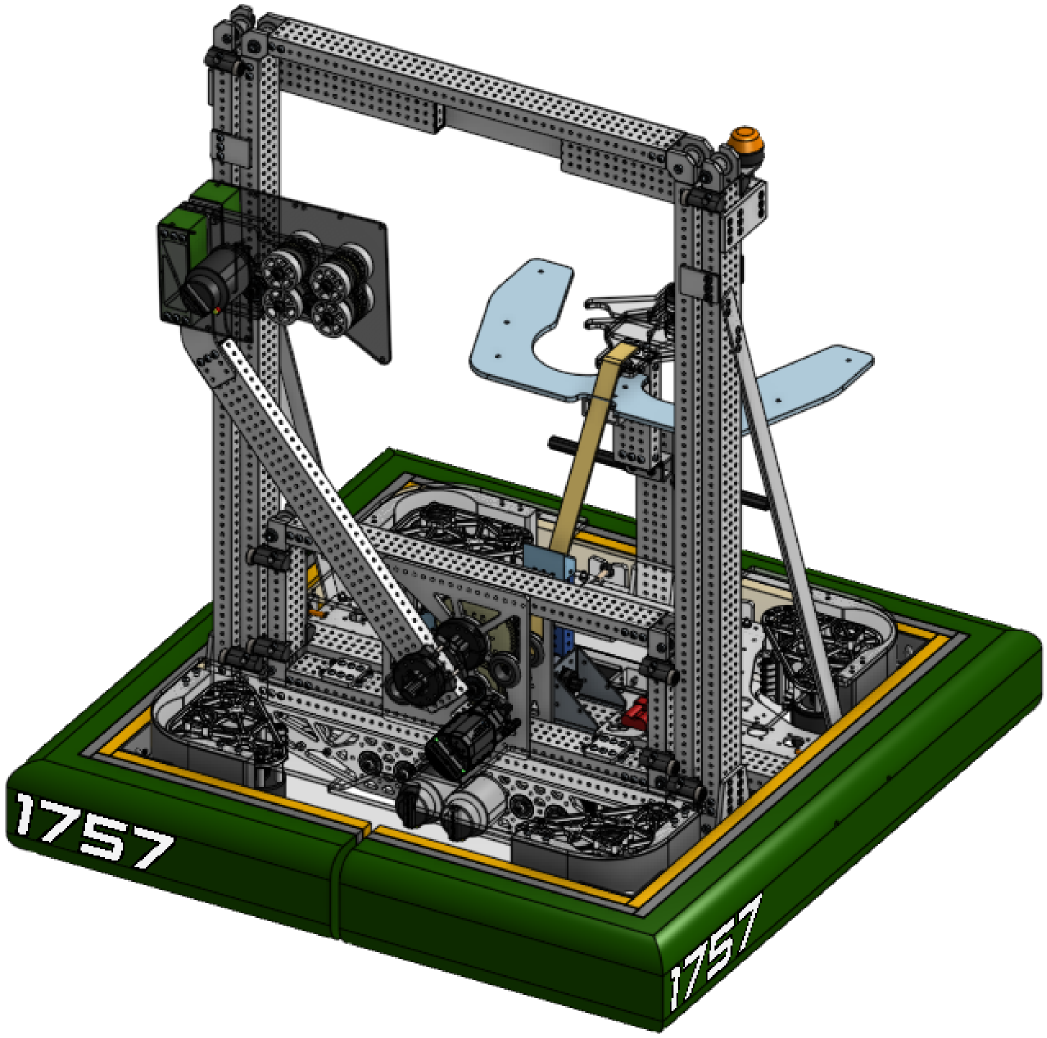
PROUD
MEMBER



WPI DISTRICT
EVENT EDITION

PERSEUS

WOLVERINES
1757

2025 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2024-2025 Season. After our meteoric rise in competitiveness in 2023, 2024 was not the year we were expecting. We flew a little too close to the sun, and in an attempt to build the best, most capable robot we possibly could, we lost focus on what made us special. We are a team that has always done best when we isolate core robot competencies and perform those skills as consistently as possible. Despite our technical failures, we did manage to do something we thought impossible, we won the Impact Award at the WPI District Event, so despite our less-than-spectacular performance on the field, at least we have validated that our small team approach is still ensuring that we make an impact not just in our FIRST Community but our community at home as well.

We spent a lot of time in the offseason focusing on recruiting, training new students on topics like CAD, design, and electronics, and our ever-growing community outreach efforts. We did go to one off-season event, which was River Rage in NH in October.

Finally on a cold winter day in January gathered up in our classroom, dusted off the CNC, warmed up the 3D printers and eagerly awaited the reveal of the 2025 FIRST Robotics Competition REEFSCAPE. 6 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5851 lines of code, 104 git commits, 13,430 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "Perseus" for the 2025 FRC Season.

Perseus's namesake, the Greek Demigod and Son of Poseidon is most famous for slaying the Gorgon Medusa. We thought the REEF in this year's game reminded us a little of her twisted hair, since our robot's goal is to conquer the reef like Perseus conquered Medusa, we thought it was a befitting name. Also, there was a nice similarity with the name of last year's robot, Proteus (another Greek sea god), since both robots share a common subsystem design, their names appropriately share a similar lineage. [1]

We are super excited as this is our 3rd year participating in the Open Alliance. We have found the Open Alliance teams and their open and timely build season updates so helpful to our team and we are thrilled to share our own progress and hopefully help other teams the same way the alliance has already helped us. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2025-build-thread-open-alliance/474230/4

We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained or a few hard-learned lessons we learned along the way.

URI District Event Update

URI Had its ups and downs; we finished day 1 of qualifications ranked 32 of 36 Robots and went home feeling pretty defeated and looking to move on to our next event, then something amazing happened. Day 2, with four matches left to play in qualification, we won all four matches and managed to climb 22 ranks to finish qualifications ranked 10th overall. Eventually, being Named 6th Seed Alliance captain. In the playoffs, we quickly dropped down to the losers bracket but managed to rack up 2 wins against the 7th and 4th seed alliances before losing in the lower bracket Semi-Final. Overall, it was a pretty successful event for us; of there was one other Award as well, which was pretty cool.

1. Its not actually that mythological in origin. Team 6328 Stole our Robot name, and this was the Backup Name

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

Our process is heavily influenced on the format developed by Team 125 the neutrons which involves a systematic breakdown of the game.

This process begins with identifying every individual skill a robot would need to accomplish any part of the game's tasks broken down as granular as drive forward to as complicated as being able to pick up game pieces off the floor without input from the driver. A new record for this year, we identified **129** potentially valuable robot skills that we might expect to either be barebones required or beneficial to the season. These robot skills aren't necessarily the *best* ways to do given tasks, so we then followed with which scoring objectives and which skills complete which scoring objectives by number
This list is in no way exhaustive, a lot of the sensor/indicator parts are generally helpful for the entire bot, so they aren't always reflected in the beneficial column.

After this we identify all the ways the robot can score a point in the game. After we map out every way the robot can score we go back to our list of skills and determine which skills are needed to do each of the scoring objectives. Afterwards, we conduct time-based analysis, taking into account the maximum number of match points and the time needed to complete those tasks. Obviously, those with the largest point values and the least amount of time to complete will receive the biggest consideration during build season.

we looked at 7 different archetypes. Superstar bots that will be good at everything and have a four piece L4 auto and a deep cage endgame. Average bots that will have the ability to score both game pieces, have a 2 game piece L2 auto with a shallow cage endgame. kitbots with a one piece auto, source intake and park endgame. "Everybot" which is a kitbot that can shallow climb and process algae. Algae specialists which can score in net, reef and possibly a shallow cage endgame. Coral specialists with a 3 L4 coral auto, presumably knock off but not score algae and a shallow cage endgame. Finally the Endgame specialist with a leave auto L1 scoring processor and a deep cage endgame. Assessment of likely alliance combinations were made and our bot specialization was then influenced by the estimated match and ranking point each type of bot is able to score.

Based on our analysis, the team unanimously decided that L4 coral was a priority for the bot. This is due to L4 being the most effective scoring method while being a coral-dominant bot whether it be in auto or teleop. However, L4 is difficult to construct in our opinion, so while we will design for L4, we may end up going back to L3 as a backup plan for when we need to pivot our priorities. On average, we estimate that the amount of game pieces we can score during the 15-second Auto phase is either three corals or two algae with our current estimated speeds.

IDENTIFYING DESIGN CONSTRAINTS

As we mentioned earlier, we were too ambitious with our robot design last year, which showed in our lackluster performance on the field. This caused us to seriously reevaluate our priorities regarding robot design. Eventually, this led us to revise one of our 3 axioms of robot design. These Axioms have been our guiding principle over the past four seasons and have served us well. With the new revision, we think we have settled on a final draft.

TEAM 1757 AXIOMS OF ROBOT DESIGN	
2022-2024	**2025**
Build Enough Hardware so that Software can continue to improve our Performance.	Build Enough Hardware so that Software can continue to improve our Performance.
Get all the Ranking Points Yourself	Be Able to Contribute to all the Ranking Points materially.
There will be lots of Robots; Vision and Partial Automation will Make you Stand out	There will be lots of Robots; Vision and Partial Automation will Make you Stand out.

THE 1757 RAPID DEVELOPMENT MODEL

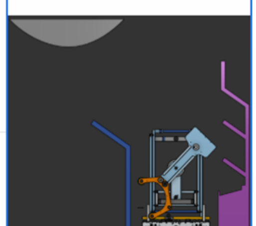
DEFINE
- Clearly Identify the design requirements of the system

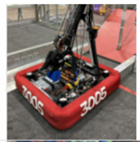
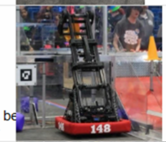
PROTOTYPE
- Design and Build a prototype that can be used to test design assumptions and Test

REFINE
- Use what we learned from testing to develop a final design

DEPLOY
- Fabricate final version and intergate into overall robot systems

ARCHETYPE CONSIDERATIONS

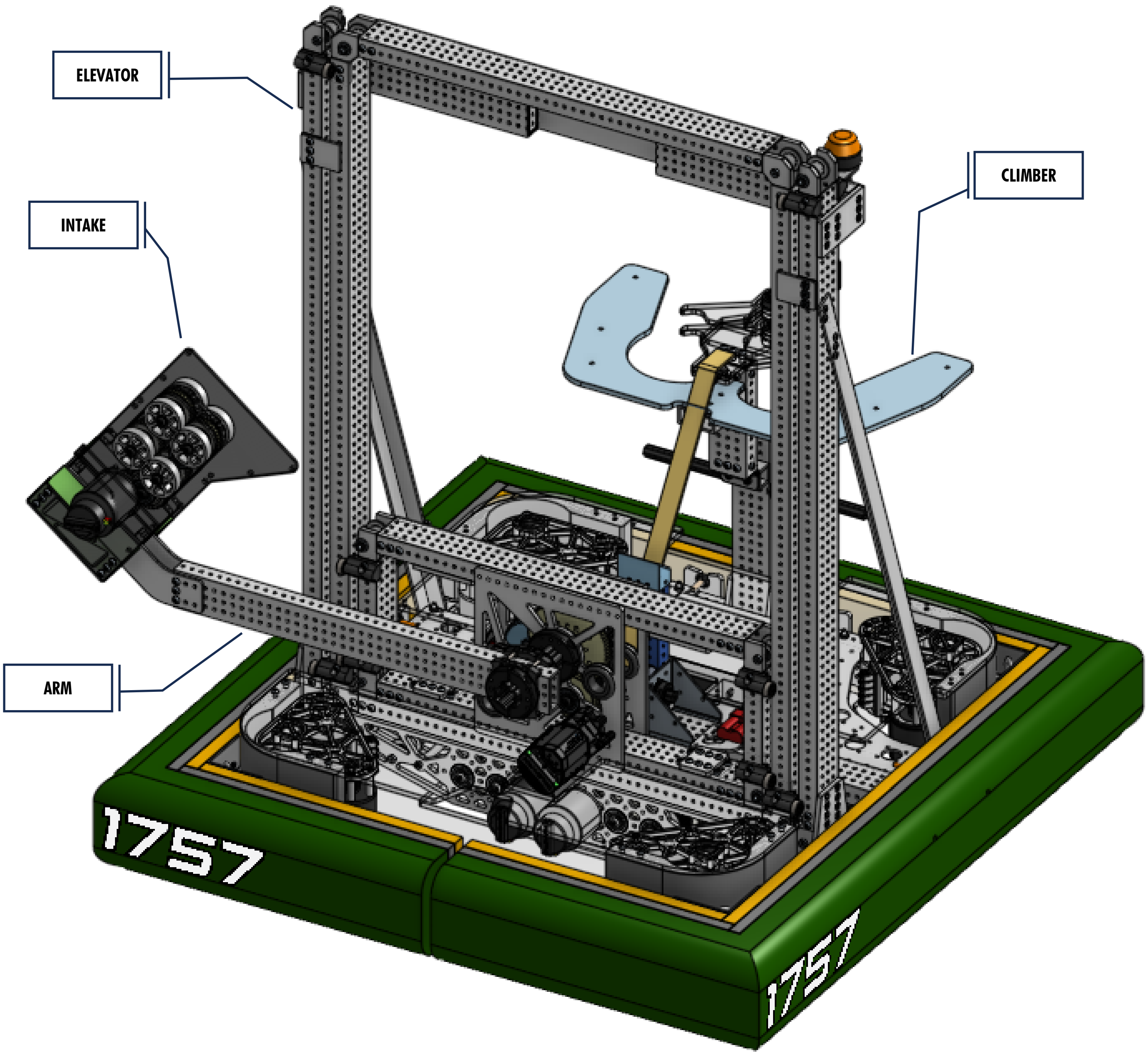
2	Name	Pro (features)	Con (limitations)	Pictures
3	Quokka HCR (side elevator with pivot)	-Score coral L1-L4 -Ground algae pickup + reef grab -Processor and net -Proven concept -Already have elevator and pivot arm design from last year -4 Dof (Elevator + arm pivot + coral gripper + algae gripper) -Extra space for climber on bot	-May be difficult to make/design -Has high extension (must be sturdy) -CG concentrated on a side -Big + slow intake -No current climb	
4	Quokka Oops all coral	-Score coral L1-L4 -Single scoring mechanism -Already have elevator and pivot arm -Most space for climber to be decided later -3 DoF (Elevator + arm pivot + gripper) -Can add in algae later (expandable)	-no climb designed yet -L4 can be a lil sketchy -Missing out on algae scoring	
5	Cranberry Alarm Ri3D	-Score coral L1-L4 -Single scoring mechanism -Already have elevator -3 DoF (Elevator + arm pivot + gripper)	- no climb designed yet -limited space for climber - big space allocation of hopper - L4 can be a lil sketchy - no net	
6	Rusthounds Ri3D	- Score coral L1-L4	- can't do algae processor - no net	

7	3005 2023	- buildable - "simple"	- not defined sketch (yet) - might be difficlt reach - frame perimiter stuff	
8	148 2023	- uses elevator - cad exists on multiple variations - L1-4 - flexible end effector ideas	-complicated to build -no climb -we havn't done anything like this be -need to be adadpted to this year	
9	2468 2023	- Tilted elevator has better stability - 3 DOFs (elevator, pivot, gripper)	- Not yet proven for this year? - frame perimiter potential issues	
10	Penn state Ri3D (except climb)	- can pluck algae while getting coral - L1-4 - algae ground and processor	- cg for climb - large space allocation for climb - startup for intake is weird	

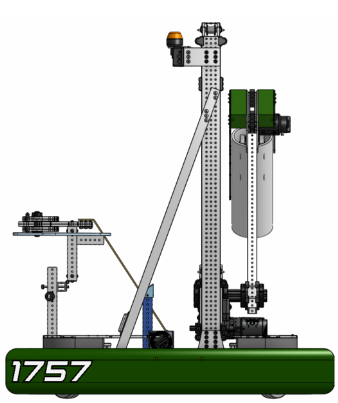
Through a collective vote within the team, we decided on building a Quokka-style all-coral bot. It is a good base bot to start, as it ensures our scoring capabilities without making it too complicated to design.
For simplicity, we are considering designing a bot without an algae intake. We decided that scoring in the barge net was not worth enough points in a match to warrant the time and effort that it would take us to design an algae intake. We believe that it may be advantageous to specialize in one area rather than being just average in all categories during alliance selection at competition.
We are also considering using the coral mechanism to knock loose algae from the reef to help others during a match. We've also noted that it may be advantageous to be specialized during alliance selection, as being good in one sector is better than being mediocre in all. You know what they say: "Jack of all trades, master of none."

The team ultimately decided to make a side mounted elevator and intake mechanism. his helps us to stay compact, and the fact that we can reference our bot's elevator-intake from last year is an added bonus.

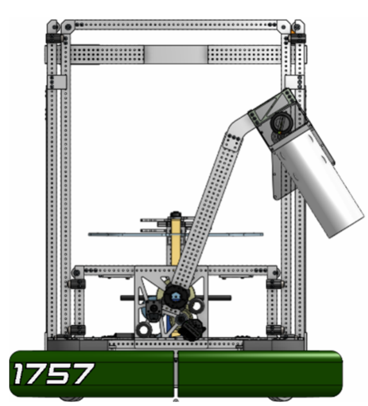
Final Robot Design



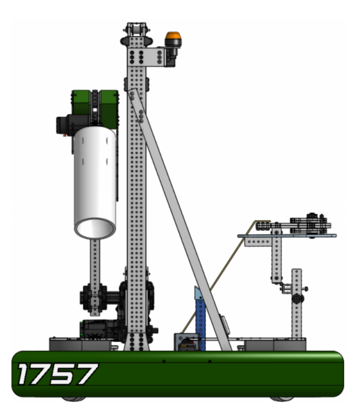
ISOMETRIC VIEW



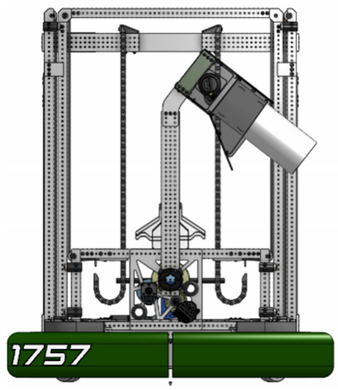
FRONT VIEW



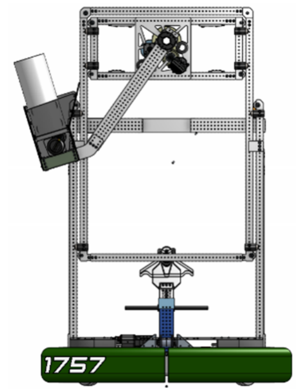
SIDE VIEW



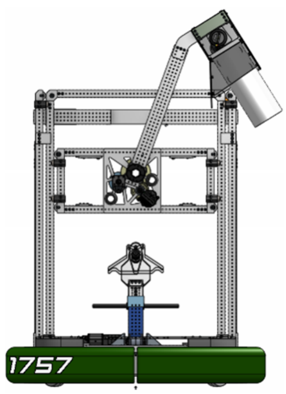
REAR VIEW



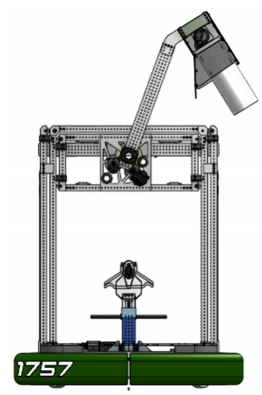
TRAVEL



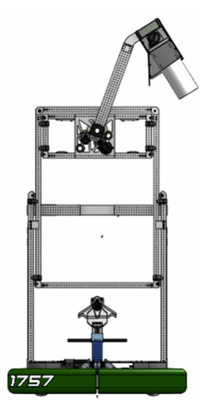
LOADING STATION



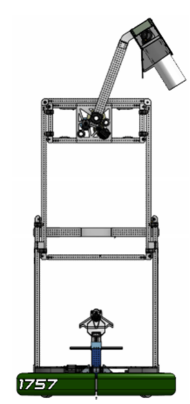
L1 SCORE



L2 – SCORE



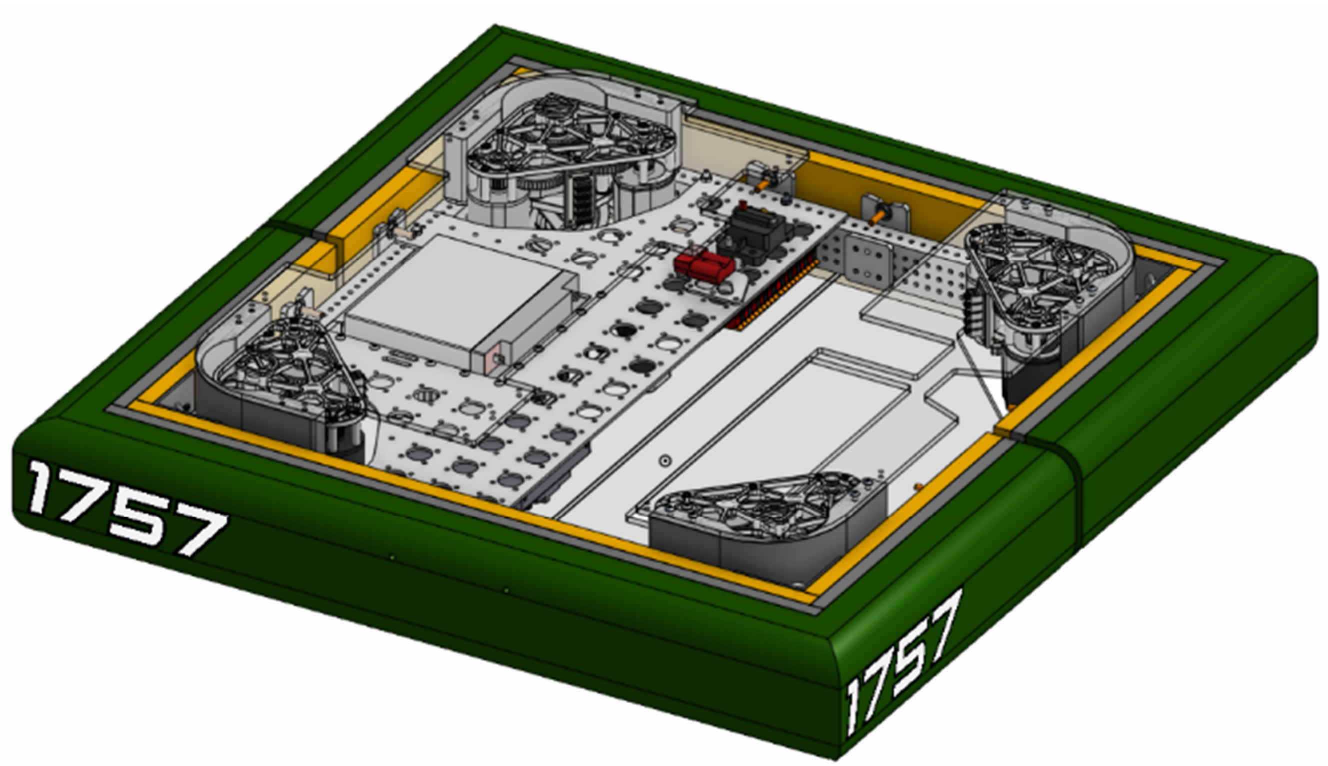
L3 – SCORE



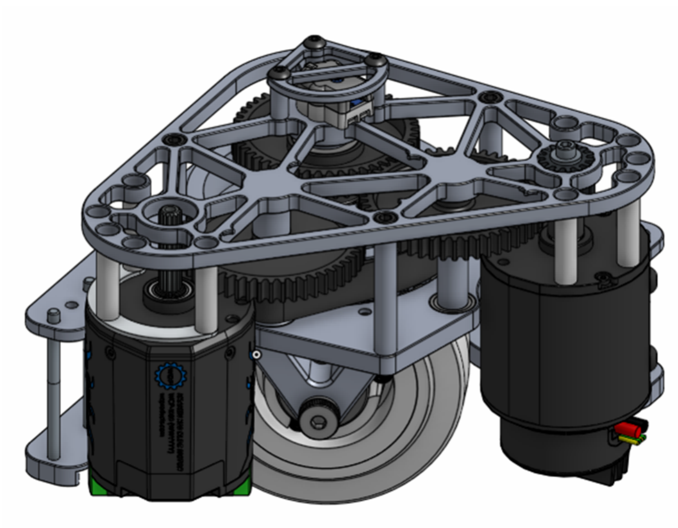
L4 – SCORE

ABOVE: Side View Images of the Robot Mechanism at Various Critical Points in its operation

MAJOR SYSTEM #1: DRIVE TRAIN

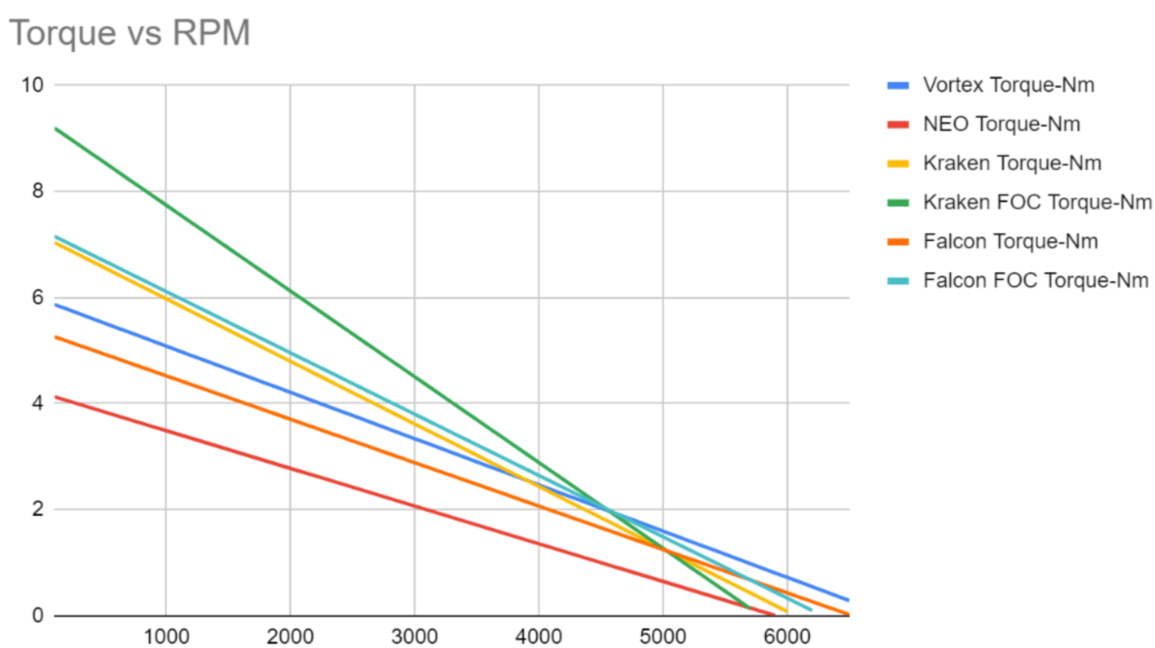


1.1 - SWERVE DRIVE MODULES



Perseus SDS MK4i Swerve Module Configuration	
Powerplant (Steering)	Falcon 500
Powerplant (Propulsion)	Kraken x60
Gearbox Configuration	L3
Overall Gearbox Ratio	6.12:1
Unadjusted Free Speed	17.1 ft/s
Drive Wheel	4” Coulson Caster

Perseus is our 4rd competition bot that has utilized a Swerve Drive for its drive train configuration. Each swerve drive modules utilize two motors, on ois used to power the propulsion (forward and back) of the Drive Wheel, and the second motor is used to adjust the rotation of the drive wheel. A Swerve drive system is holonomic meaning the robot can move in any direction relative to the field with changing the orientation of is chassis



Above: Comparison of Torque vs RPM for Various Brushless motors commonly used in FRC Robots, as you can see the Kraken motors have far superior Torque performance at Low RPM, this means more power to get the robot moving faster (Graph: Courtesy of CTRE)

1.2 - ELECTRONICS SUBSYSTEM

Continuing a trend from the last 2 years, we have an underslung electronics system; this means that we have an electronics pan mounted on top of the drive rails, and all the electronics are installed upside down underneath it. This allows for wide open access to the electronics systems when the robot is tipped over (very carefully) on its side. There is no need to remove robot components in order to access the electronics, and the electronics are protected from any stray game pieces or from getting snagged by another robot. This does leave the electronics exposed to the floor, however, and with our low ground clearance, this could be a problem. To alleviate any potential disasters, a 0.25” thick polycarbonate skid plate is mounted to the underside of the drive rails to protect the electronics bay. The Polycarbonate plate is easily removed by unscrewing a few 10/32 countersunk bolts.

Electronics System Major Components

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) CTRE Pigion 2 Accelerometer
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (1 ea) Limelight 3 Computer Vison Camera
- (4 ea) Kraken x60 Brushless Motors w/ Integrated Talon FX Motor Controller
- (7 ea) Falcon 500 Motors w/ Integrated Talon FX Motor Controller
- (1 ea) VH-109 Vivid Hosting Radio

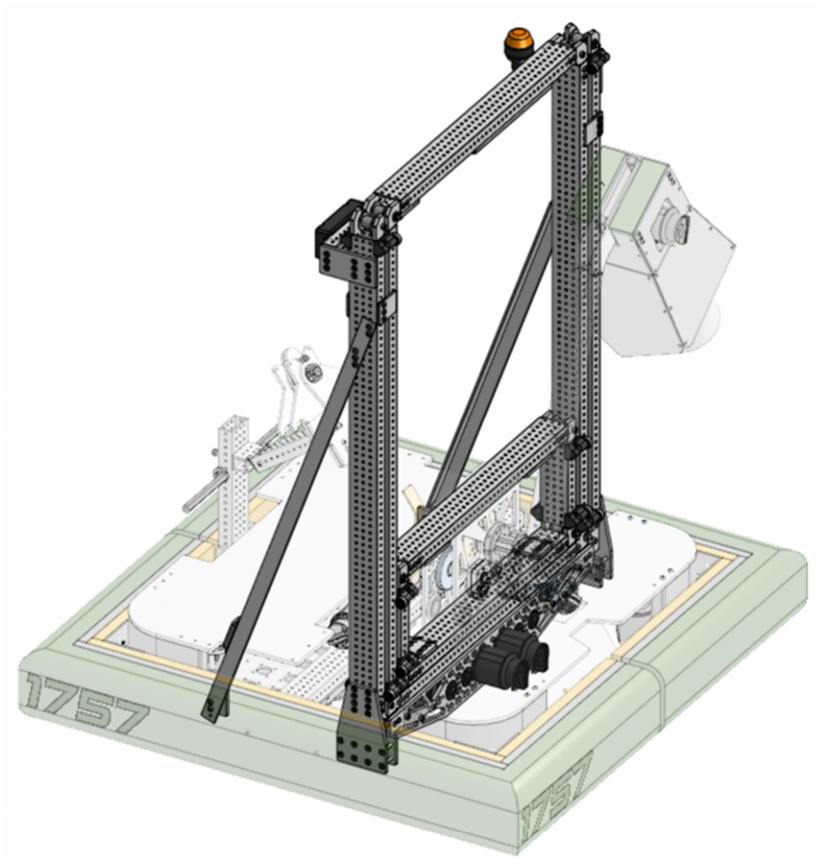
1.3 –ROBOT STATE INDICATORS

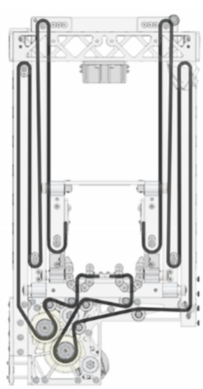
We Added LED RGBW Strips to the frame of our robot so we can easily signal to the driver what the robot thinks its current state is in for Example, if the driver tells the robot to align itself to the target the Lights Will Flash Yellow and remain Yellow until the Robots vison system says that the Shooter is locked on to its target, when the robot is locked on the lights turn green. This allows the Driver to use the Auto Aiming Features without taking his eyes off the robot itself. The second driver has a more detailed view of the state of various sensors on the bot and can call out any relevant info to the primary driver.

Just For Fun

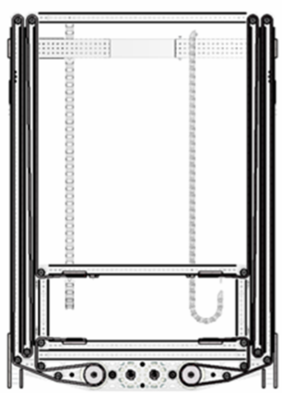
Sometimes things are useful sometimes they are just for fun, we spent a few hours working on the code so that when the robot was on but not enabled the LEDS change to an animated Rainbow sequence. As soon as the match is over the robot displays Rainbow RGBs, does it serve any functional purpose…No, Is it entertaining…Yes

MAJOR SYSTEM #2: ELEVATOR





2024 – Proteus

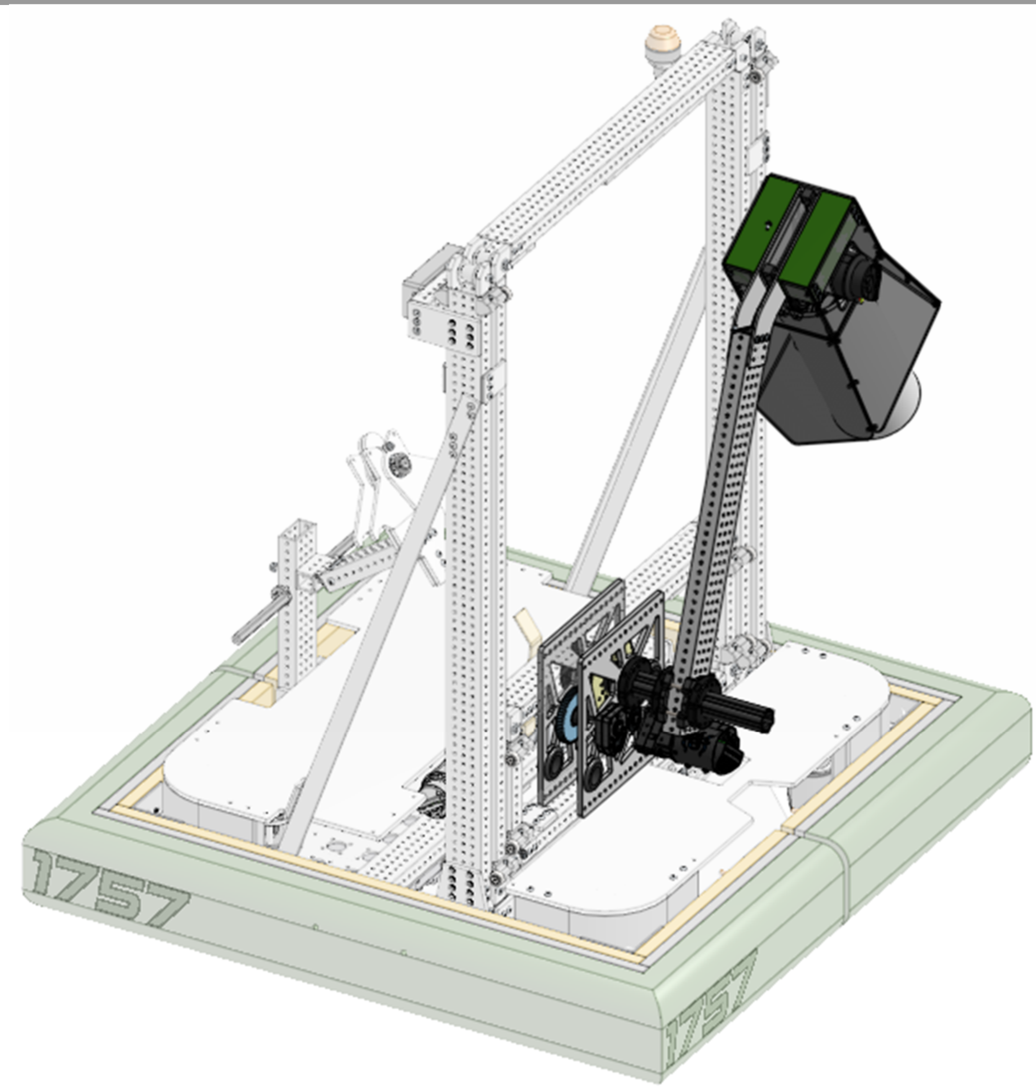


2025 - Perseus

Perseus features a continuously belted elevator similar in design to the one we used on last year's robot Proteus, with some notable improvements from last year.

Unlike last year, where we ran an extremely complicated belting pattern within our gearbox, the belting pathway is greatly reduced. Also instead of using the COTS solution from WCP for the elevator carriage bearing block we developed a custom solution based off of a design by team 111 Wildstang.

MAJOR SYSTEM # 3: ARM



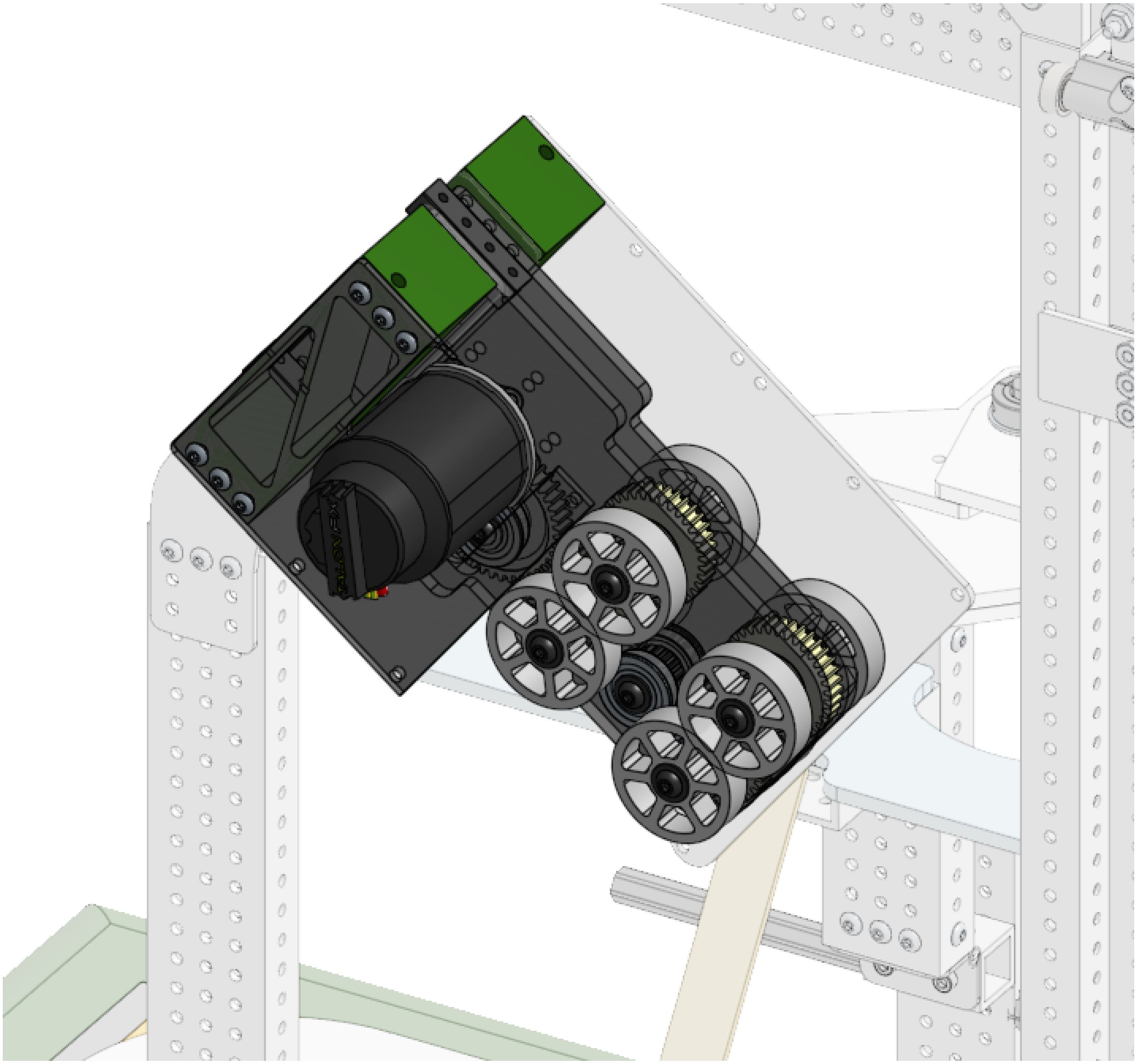
Our Arm is affectionately known as the Hockey stick of doom. It's a fairly simple design with the Intake mounted slightly off the axis of a straight metal tube arm. The Arm is fixed to a Max spline shaft, which is free-floating through the Gearbox assembly contained in the innermost stage of the elevator subsystem. Unfortunately, due to the position of the Falcon 500 that powers the Arm gearbox, we cannot rotate the arm a full 360°. We initially identified this as a high-priority design objective, but we just ran out of time in the build season and had to prioritize other objectives. If we have time between week 4 and DCMP, we have an idea to use a COTS 90° offset gearbox to move the entire motor assembly inside the elevator carriage.

Electrical Wiring—We pass all the wires for the intake through the hollow shaft of the MAXSpline Tube. We utilized a similar design last year, and it works pretty well to prevent wire strain by not wrapping the wire leads around a shaft multiple times.

DESIGN TO MINIMIZE CYCLE TIMES

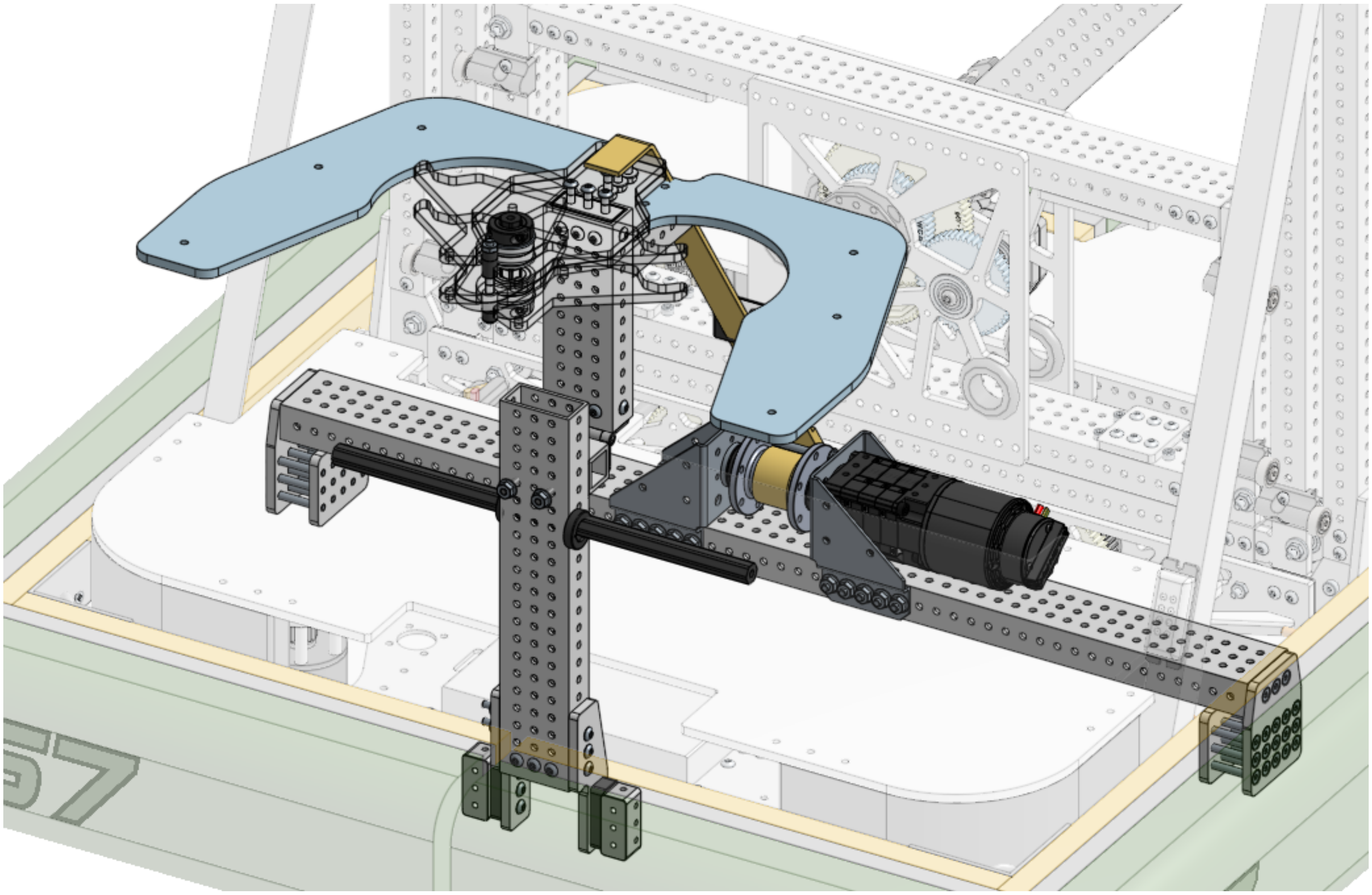
Probably the most critical statistic for any robot this year and many years past is Cycle Time, how long it takes your robot to collect a game piece and score it. The shorter your cycle time, the more scoring opportunities you will have in any Match; more scoring opportunities mean more points, which means more wins. Every little discrete function you need to accomplish each cycle adds time to your cycles. One thing we did with this year's robot to help minimize this is that our Arm intakes Corral on the opposite side of the robot from where it scores it on the reef; this means that we don't need to turn the robot 180° between the coral station and the Reef, it's small savings, probably 1-2secs, but in a 2:15 long match if you can cut 1-2 secs off of each cycle, that could give you 2-3 additional scoring opportunities per match.

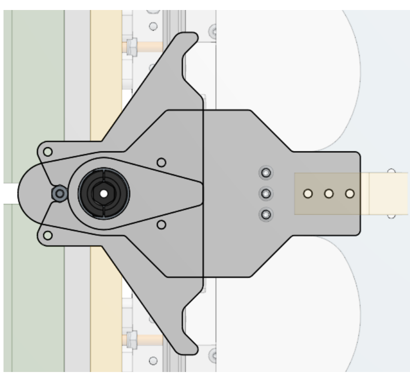
MAJOR SYSTEM # 4: INTAKE



We copied this intake from Team 448 Their Open Alliance thread and publicly available CAD. Go check out those crazy Dutchmen on their Open Alliance Build Thread https://www.chiefdelphi.com/t/frc-4481-team-rembrandts-2025-build-thread-open-alliance/472303/216?u=sharrington9614

MAJOR SYSTEM # 5: CLIMBER





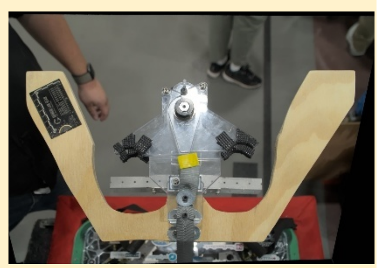
Our Deep Climb Mechanism features a spring-loaded barb that passively slides between two cage Posts. The passive hooks then positively engage on the back of the post, and the winch pulls down on the hook, pivoting the cage upright by 90° and lifting the robot up at the same time.

It took a lot of trial and error to find the right ratio of the distance between the Hook and the fixed point on the robot that the cage pivots on in order to get a successful climb.

Special Thanks to Team 118's Everybot Climber Design.

URI – RAPID CARPENTRY EDITION

Our original design did not have the large blue "VADER" wing on the arm that helps to collect and guide the cage into the climber hook. During controlled driver practice, it was easy to drive up and hook onto the cage. We seriously underestimated how often the cages would be swinging or spinning when we went to climb, and through practice and our first 4 qualification matches, we did not manage a single successful deep climb, One Trip to Home Depot, a Jig Saw, and 2 Googly eyes later we had a 100% success rate with our Deep Climb



SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020.

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible. We have a programing team of 5 students and they all program remotely for the most part so being able to spilt up tasks and work on multiple branches of code at once is great

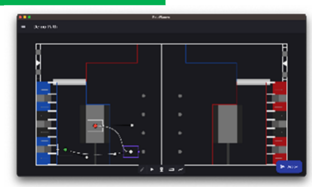
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



This is our Fouth full season using PathPlanner as part of our autonomous software stack. The simple to use graphical interface allows us to quickly generate new autonomous sequences and deploy them in a matter of minutes

LESSON LEARNED – ENSURING CODING SUCCESS!

One of our code policies is to have at least two other students of the team review the code before being merged into main. This has two main pros/cons

- **Pro:** multiple students on the team know whats going on with code, not a single source of failure
- **Con:** if one student has a bunch of back to back, individual complete changes, they have to wait for other students to completely understand to use on the bot

The con part has hurt us especially nearing competitions last year, where we just had one student who made a lot of last-minute integration changes to fix the bot, but we had a “magic branch” which defeated the purpose of main. More students this year have stepped up, and we hope that by just finishing faster these on-the-fly changes are mitigated

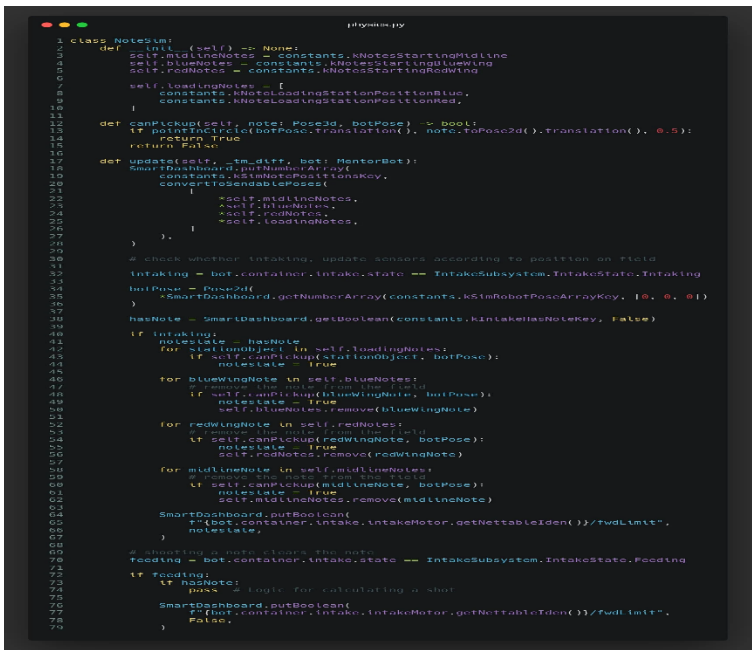
SOFTWARE: DRIVE

Using our same custom swerve base for this year, the drive subsystem is pretty box-standard. A few of our major optimizations include utilizing second-degree swerve kinematics to increase efficiency as well as compensating for a rotating and moving robot Intake:

A Brief Tangent

Two years ago our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important. The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

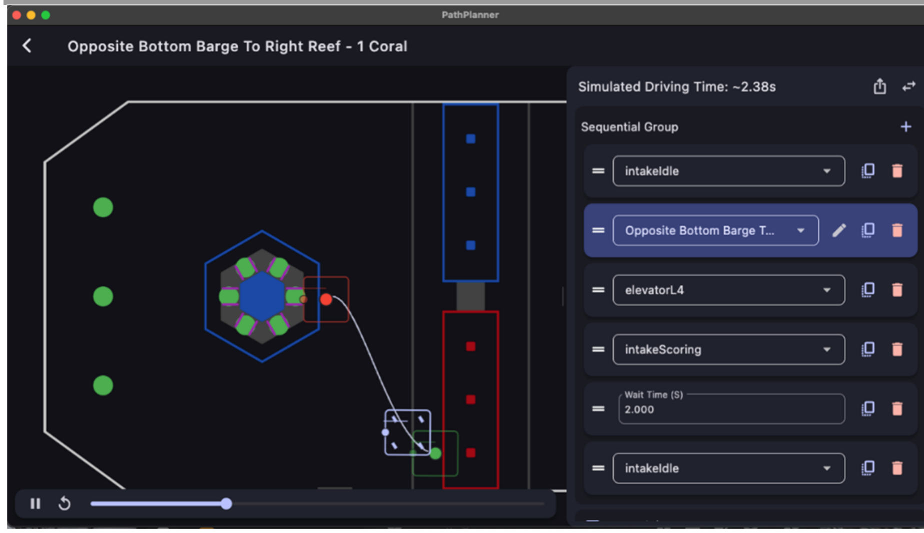
SOFTWARE: INTAKE



```python
class NoteSim:
    def __init__(self) -> None:
        self.midlineNotes = constants.kNotesStartingMidline
        self.blueNotes = constants.kNotesStartingBlueWing
        self.redNotes = constants.kNotesStartingRedWing

        self.loadingNotes = [
            constants.kNoteLoadingStationPositionBlue,
            constants.kNoteLoadingStationPositionRed,
        ]

    def canPickup(self, note: Pose3d, botPose) -> bool:
        if pointInCircle(botPose.translation(), note.toPose2d().translation(), 0.5):
            return True
        return False

    def update(self, _tm_diff, bot: MentorBot):
        SmartDashboard.putNumberArray(
            constants.kSimNotePositionsKey,
            convertToSendablePoses(
                [
                    *self.midlineNotes,
                    *self.blueNotes,
                    *self.redNotes,
                    *self.loadingNotes,
                ]
            ),
        )

        # check whether intaking, update sensors according to position on field

        intaking = bot.container.intake.state == IntakeSubsystem.IntakeState.Intaking

        botPose = Pose2d(
            *SmartDashboard.getNumberArray(constants.kSimRobotPoseArrayKey, [0, 0, 0])
        )

        hasNote = SmartDashboard.getBoolean(constants.kIntakeHasNoteKey, False)

        if intaking:
            notestate = hasNote
            for stationObject in self.loadingNotes:
                if self.canPickup(stationObject, botPose):
                    notestate = True

            for blueWingNote in self.blueNotes:
                # remove the note from the field
                if self.canPickup(blueWingNote, botPose):
                    notestate = True
                    self.blueNotes.remove(blueWingNote)

            for redWingNote in self.redNotes:
                # remove the note from the field
                if self.canPickup(redWingNote, botPose):
                    notestate = True
                    self.redNotes.remove(redWingNote)

            for midlineNote in self.midlineNotes:
                # remove the note from the field
                if self.canPickup(midlineNote, botPose):
                    notestate = True
                    self.midlineNotes.remove(midlineNote)

            SmartDashboard.putBoolean(
                f"{bot.container.intake.intakeMotor.getNettableIden()}/fwdLimit",
                notestate,
            )

        # shooting a note clears the note
        feeding = bot.container.intake.state == IntakeSubsystem.IntakeState.Feeding

        if feeding:
            if hasNote:
                pass  # Logic for calculating a shot

            SmartDashboard.putBoolean(
                f"{bot.container.intake.intakeMotor.getNettableIden()}/fwdLimit",
                False,
            )
```

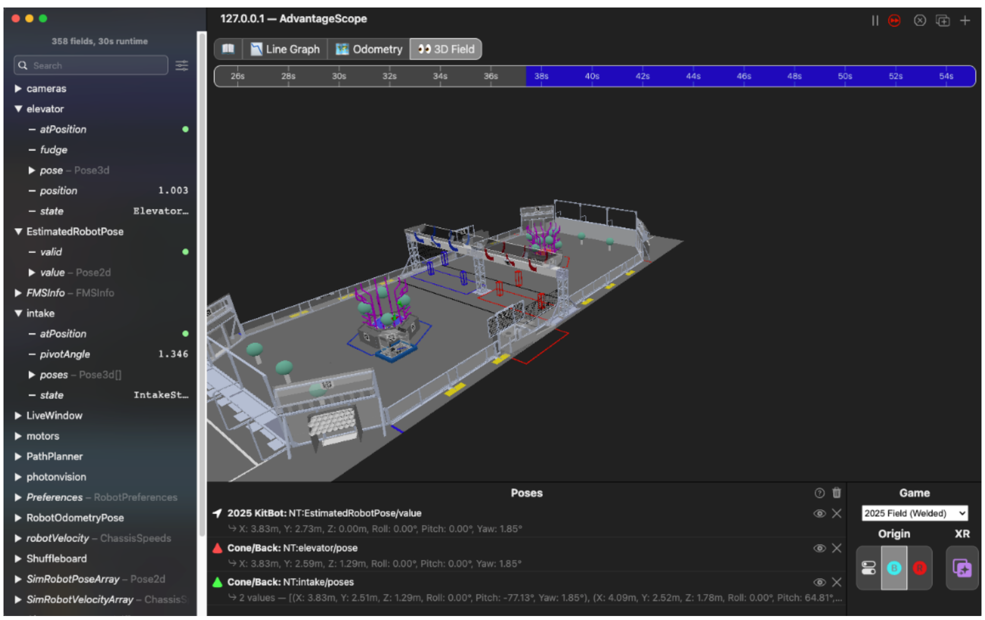
Using our full field positioning The intake is a complicated mechanism, and the most varied in state machines. It has two requirements from a software side: quick intaking and positional awareness. The first of which is completed mechanically, and executed with a velocity control on the motor itself, the second of which with positional awareness involves utilizing the two presence sensors on the intake. We calculated that, at the speeds we want to run the motor, the note has a variation of 6in due to just the timing of the roboRIO, to compensate for this there is a secondary positioning sequencing that slowly moves the note until a presence sensor doesn't detect, at which point the location of the note is known and it can move to a safe position. This is also used in amp and trap scoring to hold a known position.

SOFTWARE: AUTONOMUS



We are utilizing pathplanner to quickly build autonomous pathing sequencing. Because of the ease of control for each subsystem, having target states and knowing whether each subsystem has completed its task, waiting for actions to finish before continuing on paths is simple to do.

SOFTWARE: SIMULATION



We historically get our bot done criminally late in the game. simulation and code helps us stay ahead and ensure that the time between robot being mechanically and electrically done, and when the bot can be competition-viable is reduced as much as possible. Each motor is individually simulated from CTRE and incorporated to give a 3 dimensional representation of the bot's pieces from what their motor encoder readings give. This allows the majority of the systems to be tuned within simulation because of accurate modeling of cad-measured inertia values Vision is also simulated based on the field of view of each camera, and variation is added to ensure the robustness of the vision system.

ENGINEERING TEAM

Technical Captain

Landon Bayer*

Technical Vice-Captain

Anya Jiang

Business Captain

Claire Peng*

Business Vice-Captain

Elizabeth Lowney

Programing Lead	*Design Lead*	*Open Alliance Lead(s)*	*Outreach Lead*	*Media Lead*	*Finance Lead*
Ivan Cai	**Jacob Kaplan**	**Yuhan Chen**	**Alex Theofilou**	**Celina Yau**	**Constantina Flevarakis & Julian Sarofim**

STUDENT MEMBERS

Luke Szigety	**Henry Marsland**	**Carson Koukkos**	**John Muttiah**
Cam Howe	**An Pham**	**Zoe Jin**	**Hamsini Siddi**
Duanmu Jiang	**Erin Hwang**	**Isabel Wu**	**Gabriel Arroyo**
Lily Xu	**Marshall Atkins**	**David Confoey**	**Daniel Romanelli**
Kevin Bai	**Benjamin Thomas**	**Adi Dani**	**Kaelyn Kotfila**
Erik cerulli	**Kevin Liang**	**Owen Monahan**	**Cara Sullivan**

WHS Faculty Advisor

James Looney

Senior Mentors

Dwight Meglan **Chris Aloisio°** **Steve Harrington°**

Mentors

Anthony Gelsomini **Manny Barros°** **Mark Holthouse** *Mentor Emeritus* **Charley Marsland °** **Amber Maxwell**

Eric Yamaguchi° **Luke Maxwell°**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*	**Catherine Tseng** *Treasurer*	**Manny Barros°**	**Cheten Gopal** *Clerk*	**Jack Tseng**

**Denotes High School Seniors* *°Denotes FRC Alumni*

SPONSORS

Cardathea	Westwood High School	Gene Haas Foundation	Medtronic
Act Fasteners	Intuitive Foundation	IRA Motor Group	Teradyne
Deadham Savings	Needham Bank	Schneider Electric	Poirier
Comella’s		Print Master	
McWeeney Family	Peng Family	Gopal Family	Gelosimi Family

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

Special thanks to everyone who makes this team possible Including

Our Students
Our Mentors
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Timothy Piwowar - Superintendent
Caitlin Ahern - Asst. Superintendent
Cindy Cox - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Asst Director of Facilites
Amy Davenport - Principal
Aishleen Marcus - Assistant Principal
Tom Millett - Assistant Principal
Andrew Miller - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff

#FRC1757
@WWRobotics1757