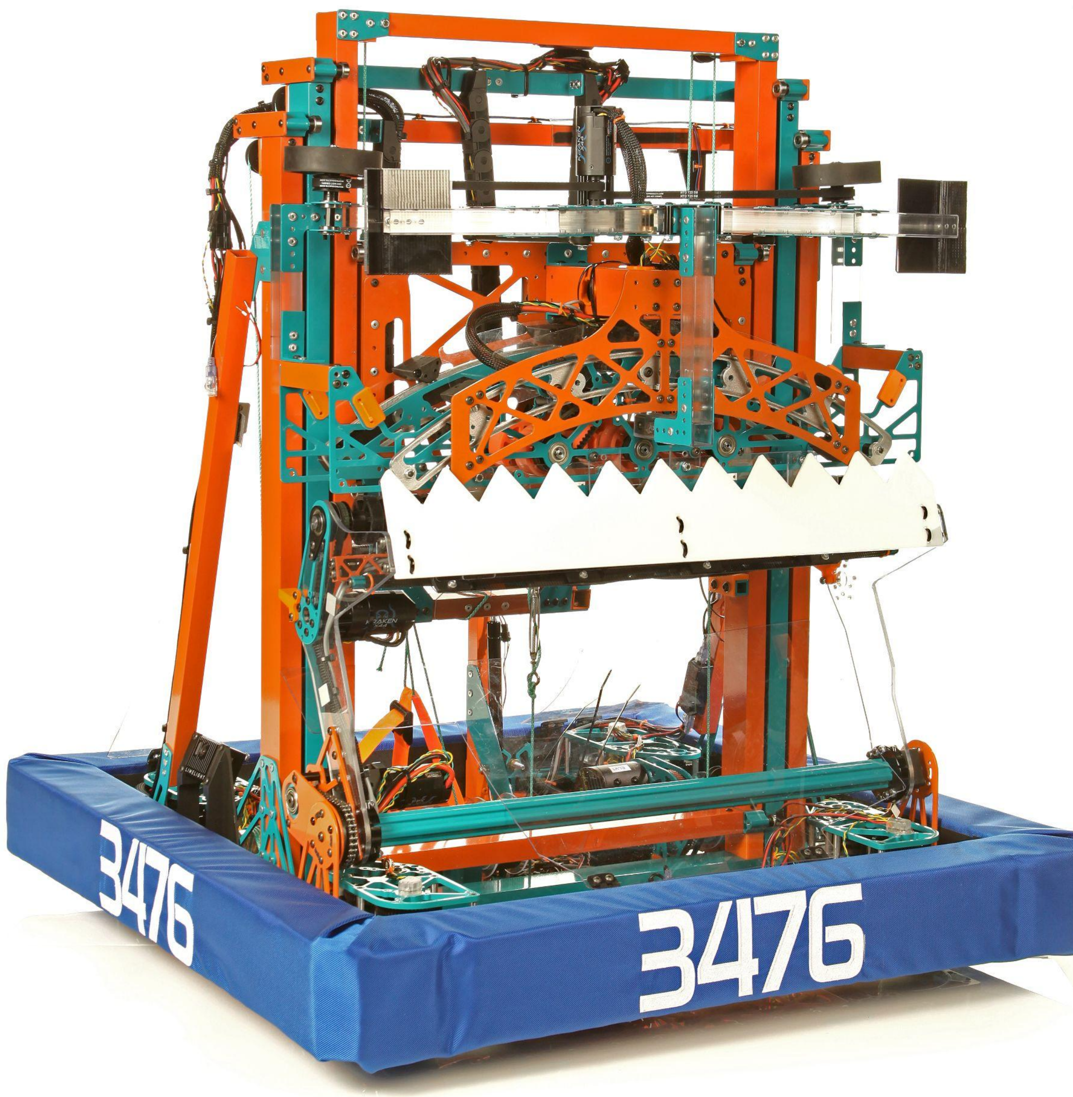
Team 3476
Code Orange
Technical Documentation

Presenting our 2025 Robot:

JAWS



“

Code Orange students have demonstrated incredible dedication, willingness to learn, and desire to help others succeed. The result is a team that is exemplary in all aspects of FIRST.

- **Karl Olson**

”

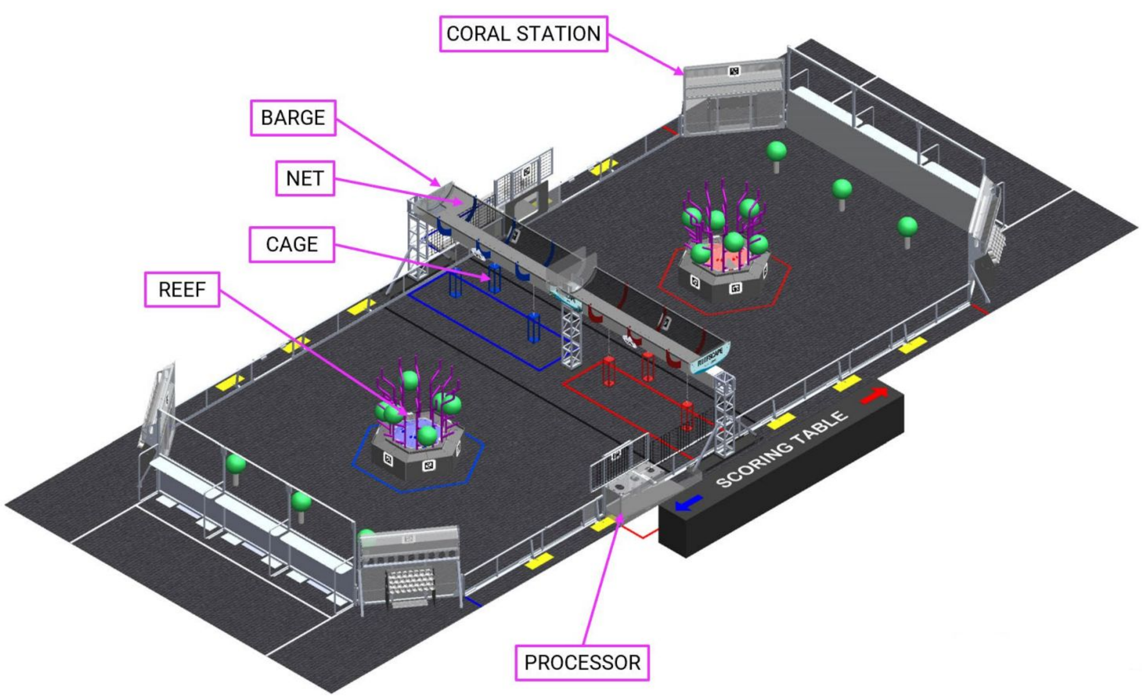
Table of Contents

I. Design Process

Design Requirements

Actions:

- Intake coral from the ground and human player substation
- Score coral and algae into all point locations
- Score coral and algae during auto, telop, and end game
- Climb during endgame for ranking points
- Fast cycles to maximize game pieces scored



Ranking Specifications
Wish, Prefer, Demand List

Robot Features	WPD
Score Algae in processor	Prefer
Score Algae the barge	Prefer
Coral in L1	Prefer
Coral in L2	DEMAND
Coral in L3	DEMAND
Coral in L4	DEMAND
Climb on shallow cage	wish
Climb on deep cage	DEMAND
Pickup coral with algae on it	wish
Coral vert ground pickup	wish
down the barrel	Prefer
perpendicular	DEMAND
Ability to center afterpickup	DEMAND
Re orientation afterpickup	DEMAND
Coral station down the barrel	Prefer
Coral station perpendicular	DEMAND
Go under shallow cage	wish
Go under deep cage	wish
Robust against cages	DEMAND
Take hits	DEMAND
Score barge in auto	wish
Score coral in auto	DEMAND
Cycle coral in auto	DEMAND
Move in auto	DEMAND
disrupt an auto	wish
Steal in auto	wish
Don't get fouls	DEMAND
Easily repairble	DEMAND
Intake algae from reef	Prefer
dislodge algae from reef	DEMAND
Don't drop game pieces	DEMAND
Hold both pieces at once	Prefer
Control algae	Prefer
Hold coral	DEMAND
Be able to play defense	Prefer
Fast coral cycle times	DEMAND
omni directional	DEMAND
Be able to preload	DEMAND
Don't tip	DEMAND
Centered COG	DEMAND
Go between cages	Prefer
Drive into cages without breaking	DEMAND
Take algae off of coral	wish
Floor intake algae	Prefer
buddy climb	wish
Buddy climb compatible	wish
touch it own it algae	Prefer

Design Requirements

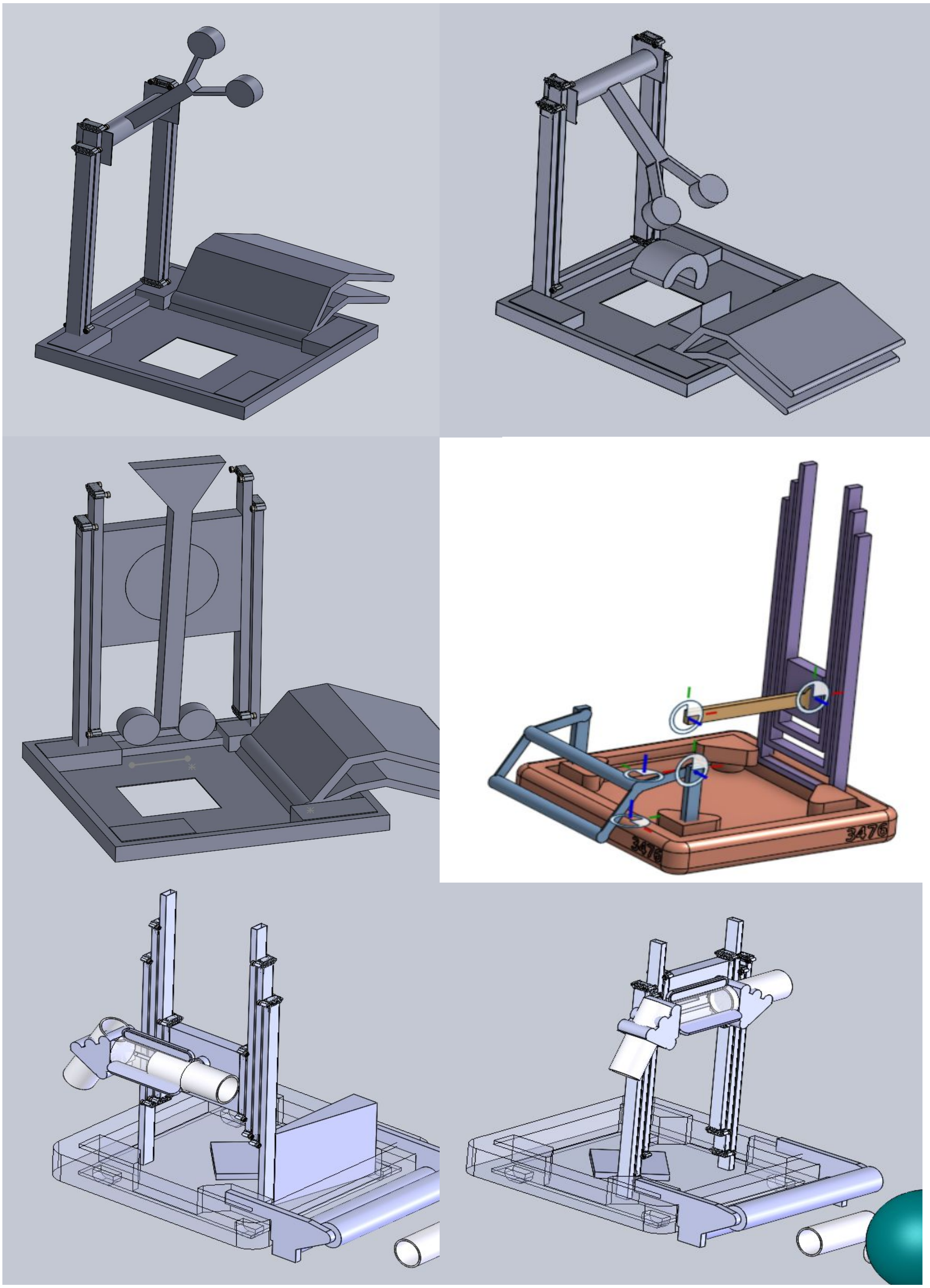
Design Requirements:

- Score in all scoring locations
- Fast and accurate
- Short cycle times
- Simple robot
- Comply with all game rules

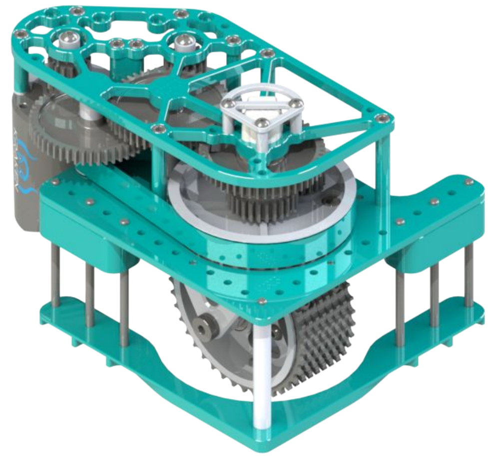
Design Considered:

- **Pivoting Arm on an Elevator:**
    - Pros:
        - Quick transition from intake to scoring
        - Fast cycling, not needing to turn around
    - Cons:
        - Long, heavy arm
        - Difficult to repair
        - Hard to implement a good scoring mechanism
- **Small mechanism that passes through the elevator:**
    - Pros:
        - Lighter arm
        - Symmetrical robot
        - Quick scoring, turning around not required
    - Cons:
        - Difficult to design
        - Difficult to repair
        - Hard to implement climber
- **Elevator with coral scoring and intake on same side:**
    - Pros:
        - Fast Cycling
        - Can score on both the left and right sides of the robot
        - Easy to implement algae mech and climb
        - Can have any type of scoring mechanism
    - Cons:
        - Heavy and tall
        - Slightly slower handoff

Overall Design Considerations



Drivebase



Design requirements:

- Fast
- Easy to wire
- Comply with rules
- Fit all subsystems
- FIt all electronics
- Doesn't tip
- Robust
- Lightweight

Designs considered:

- Swerve Gearing
  - Quickly changeable from speed to torque w/Kraken X60

- Bumper mounting design
  - L brackets to prevent screw stripping
  - Bolts and brackets to keep bumpers mounted in place
  - Changed from one piece foam to overlapping parts for more robustness
  - Mounted higher in compliance with guidelines

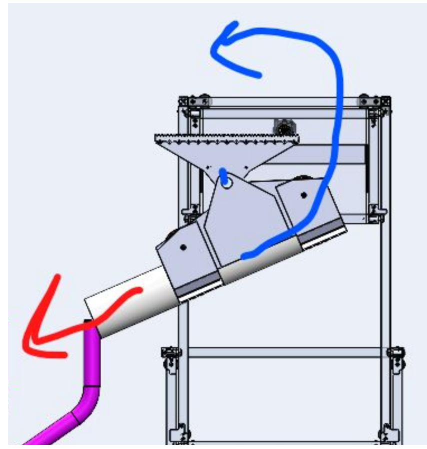
- Coral blockers
  - Mounted at an angle to deflect coral
  - Hinge used to access battery and important electronics
  - At stowed climber height to ensure coral can slide from wherever they land

- ⅛ Box tubing vs. 1/16 Box tubing - more weight vs sturdiness

- 25x25
  - Pros:
    - Small and fast
  - Cons:
    - Not enough space for electronics and subsystems
    - Easier to tip

- 30x30
  - Pros:
    - Space for subsystem and electronics
    - More stability
  - Cons:
    - Slower and heavier

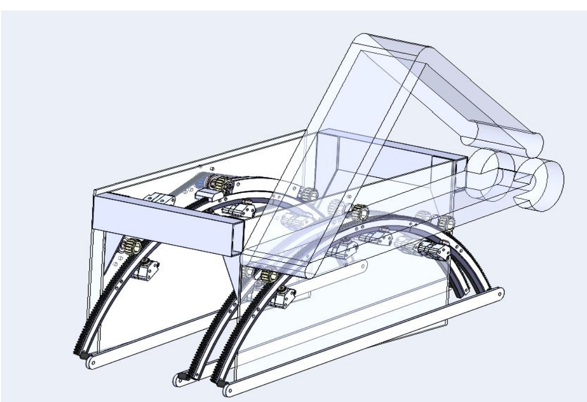
Carriage

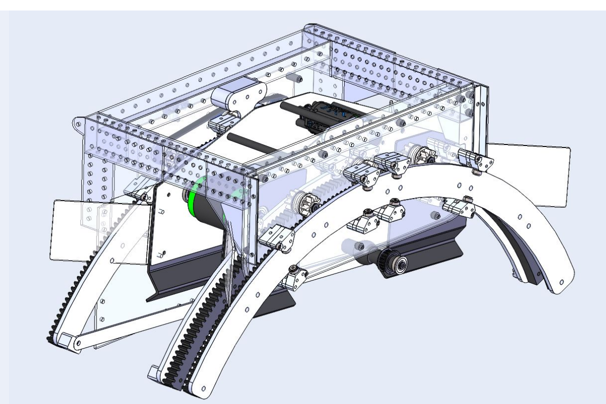


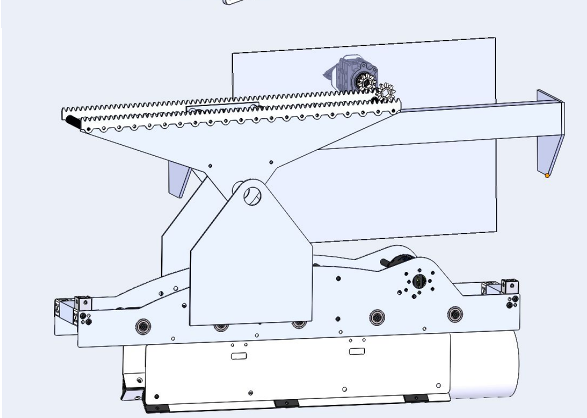
Design requirements:
- Lightweight
- Robust
- Outwards extension
- Downward rotation
- Quick rotation

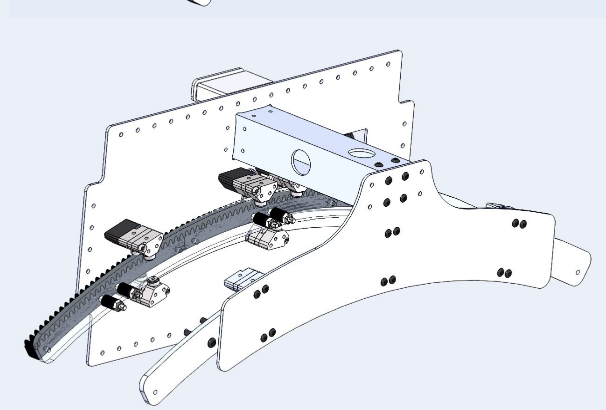
Major designs changes considered:
- Passive Rotation
  - Pros:
    - Simple, Only outward extension
    - Passive downward rotation
  - Cons:
    - Passive downward rotation
    - Can flop around while driving
    - Low scoring angle

The design of the carriage was decided with the process of elimination, starting from the simplest designs. After 5 revisions and prototypes we discovered a balance between functionality and simplicity.









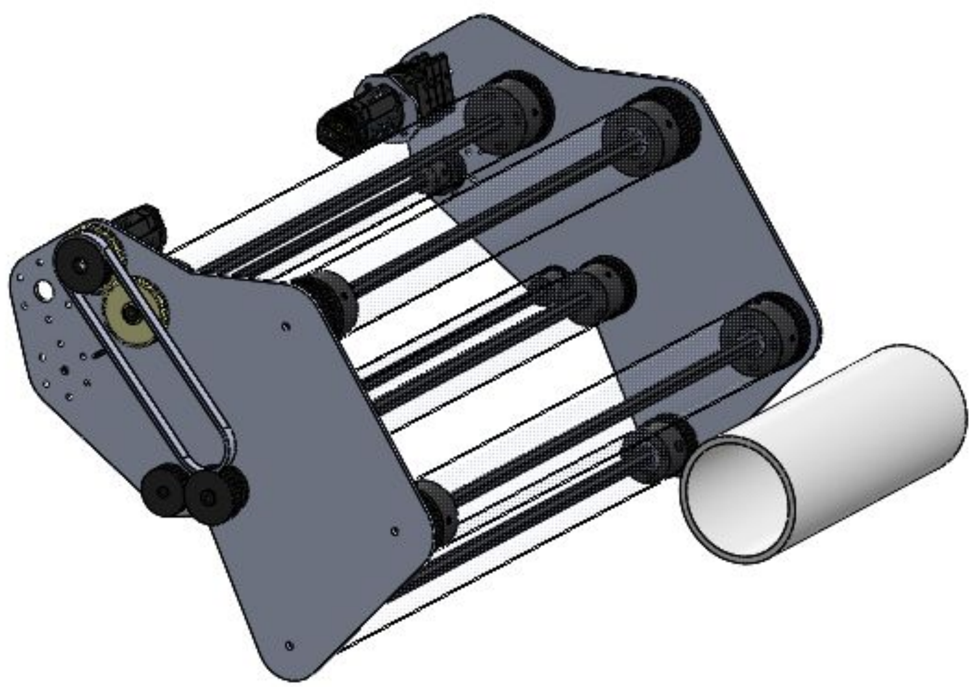
Intake

Design requirements:

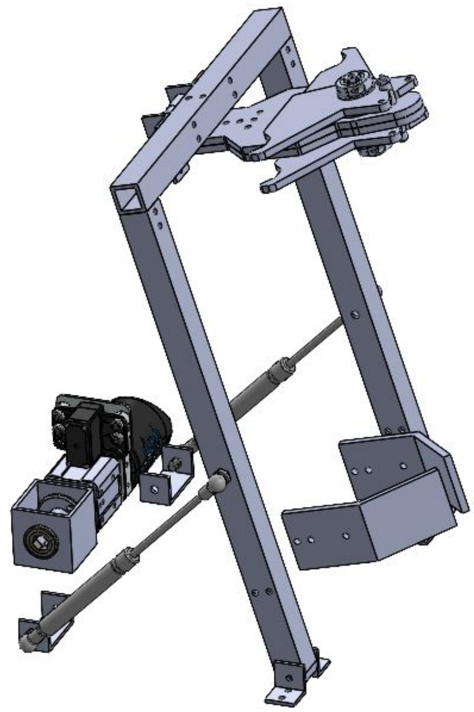
- Pick up Coral
- Make it a “Touch it own It” intake
- Be able to change the coral from a down the barrel orientation to a perpendicular position for handoff
- Lightweight
    - Move quickly to handoff

Intakes considered:

- Mecanum wheels to center the note (too heavy)
- Intake + serializer combo (too much space, too heavy)
- Double roller top on bottom (too heavy and hard to control coral orientation.)



Climber



Design requirements:

- Climb in <5 sec
- Be lightweight (<5lbs)
- Not slide down the cage
    - Especially once robot turns off: nothing should move

Designs considered:

- Arm Pivot
    - Low Pivot
        - Allows for a very robust design
        - Due to large arm radius, there is little height gained
        - Small room for error
    - High Pivot
        - Needs extra weight to raise a robust pivot
        - Small radius allows bot climb higher
        - Easy stow/take down from cage
- Latches
    - Inside Latch (Everybot)
        - Triangle shape allows for easy cage alignment
        - Only so much alignment can be done, still easy to miss
        - Very challenging to take off once completed climb
        - Hits eye bolt at top of cage, preventing a higher climb
    - Outside Latch
        - Allows for easier alignment with a funnel shape
        - Easy to remove once completed climb
        - Simple and robust, allows for extra friction

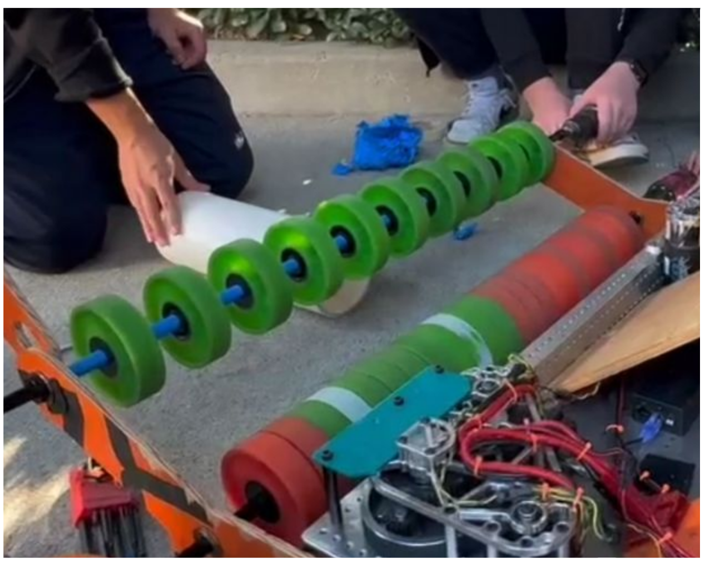
Climber (cont.)

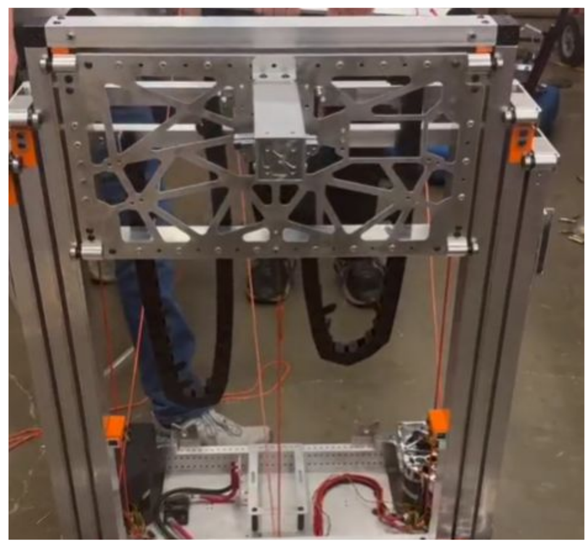
- Cage Pivot
  - Low Pivot
    - This allowed for the cage to rotate ~45 deg
    - Cage chain was far away from elevator, minimizing risk
    - Difficult to effectively implement
  - High Pivot
    - Friction based, making it easy to slip
    - Allows ~90 deg of rotation, leading to more height gained
    - Leads to the cage chain being close to elevator

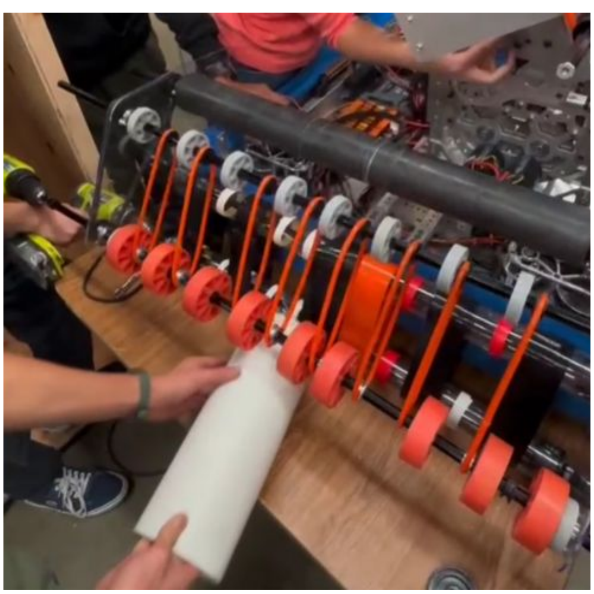
Prototyping

Next, our team breaks into smaller subsystem groups to prototype ideas based on our wish, prefer, and demand list. Each group consists of both experienced and new students, fostering knowledge transfer and skill development.

During this phase, subsystem groups brainstorm concepts and build prototypes using wood and scrap materials. Collecting data and documenting functionality through video is essential, allowing designers to reference these prototypes when determining final subsystem specifications.







CAD

Once prototyping is complete, we move on to the next phase: CAD. We select the most successful prototypes for each mechanism, and our CAD team begins the detailed design process. To ensure seamless integration, frequent meetings and open discussions facilitate coordination across mechanisms. CADers continuously refine their designs based on feedback and new ideas, then present their finalized concepts to team members and mentors for approval.

Once a design is approved, machinists take over to bring it to life. Detailed drawings are created for each part, specifying materials and dimensions to ensure precise fabrication. These 2D diagrams serve as essential guides for machining each component accurately.

CADers must efficiently design and implement these changes to keep the robot competition-ready. As a crucial part of our team, the CAD division ensures a well-integrated design that can be manufactured and assembled smoothly.

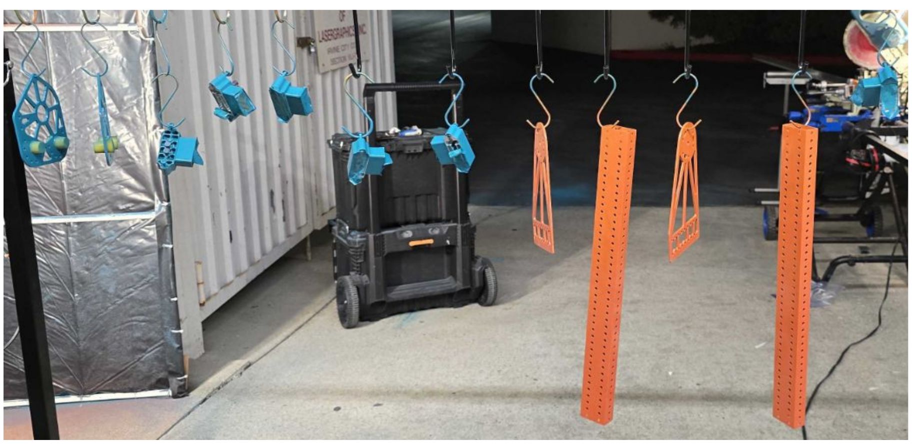
Fabrication

The manufacturing subteam is the largest on our team, responsible for transforming the CAD-designed robot model into a physical reality.

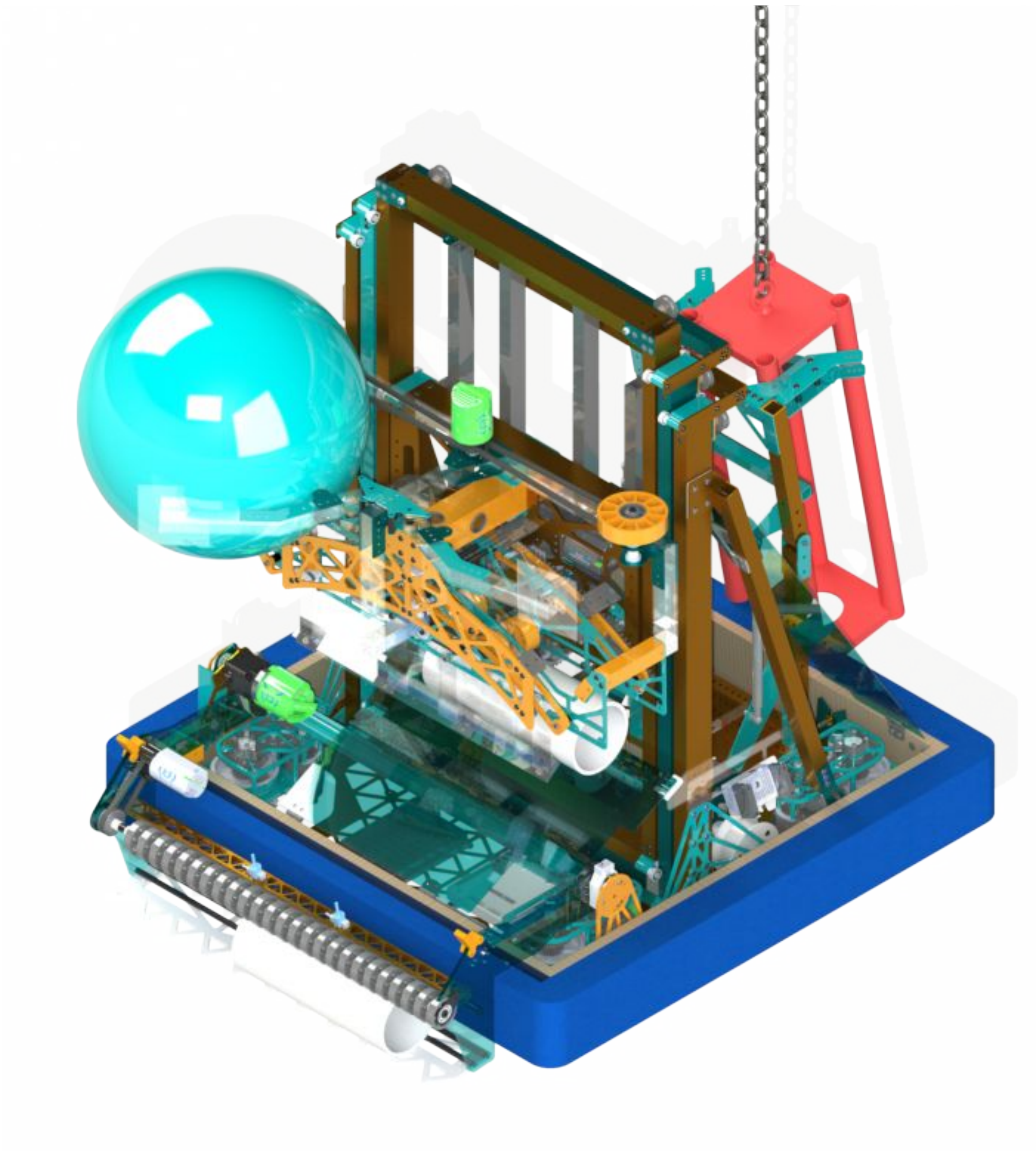
With hundreds of parts to produce, organization and precision are crucial. The manufacturing lead ensures that each part is made within the specified tolerances and properly labeled for powder coating and assembly.

Our team utilizes both basic and advanced machinery to fabricate components. This includes tools like the bandsaw, chop saw, bench grinder, and belt sander, as well as high-precision equipment such as the mill, lathe, laser cutter, CNC mill, and CNC router.

For specialized parts like camera mounts and custom pulleys, we use our Bambu X1C and Markforged 3D printers to print PLA, TPU and carbon-fiber/ nylon filament (which enabled the creation of strong, load-bearing components).

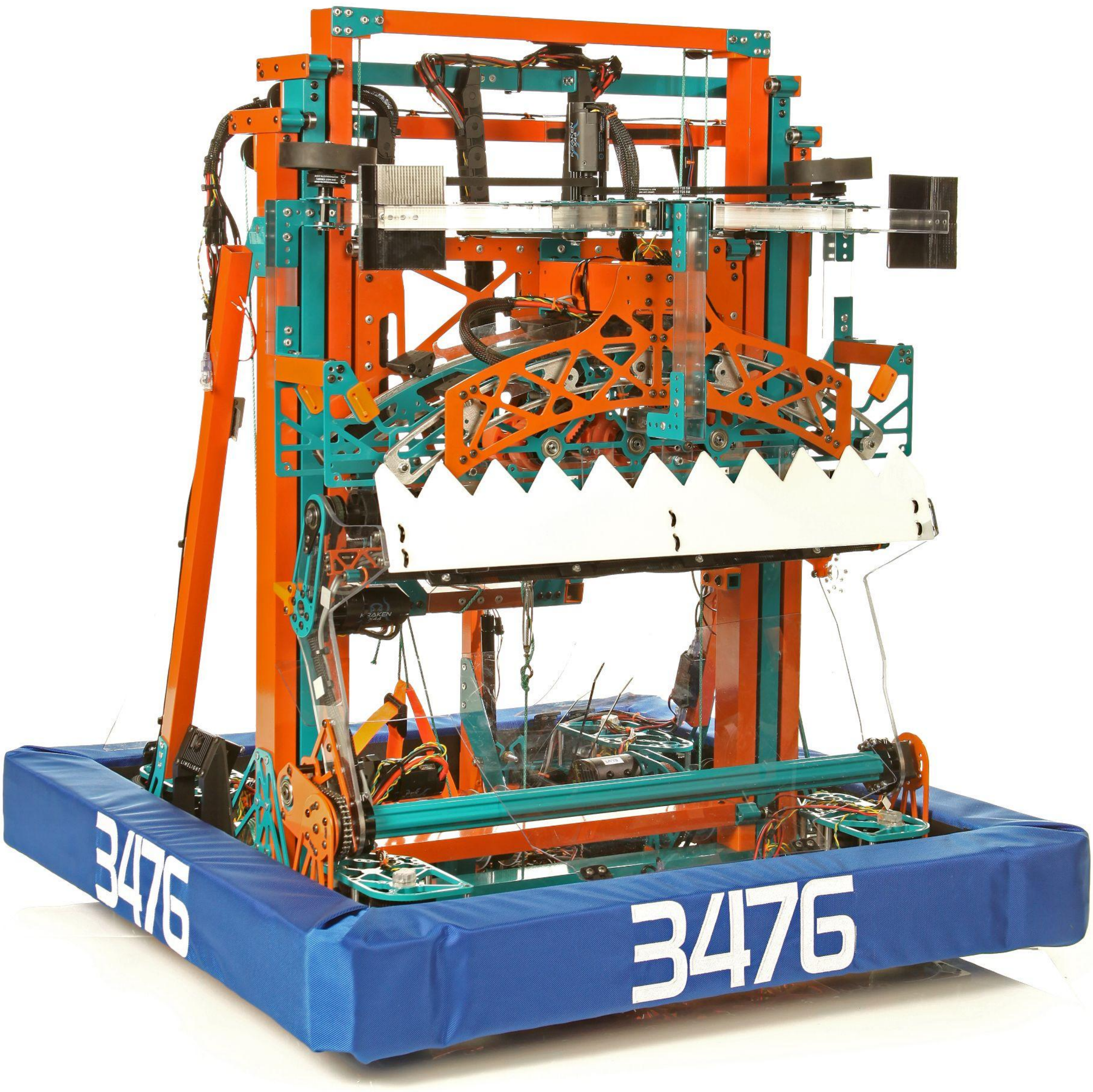


II. Mechanical Design



Final Design

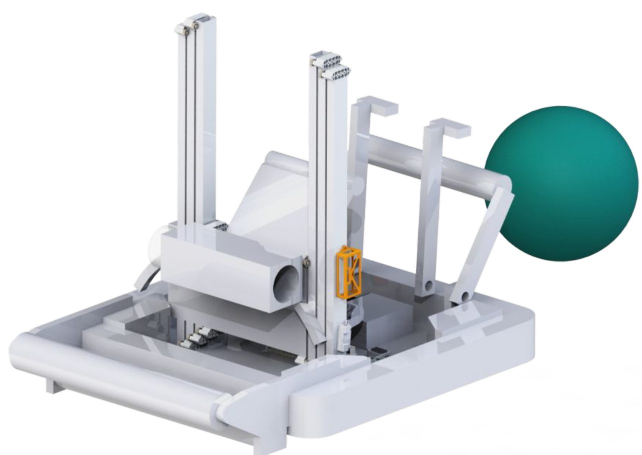
Dimensions: 29.5” x 29.5” x 41.75”



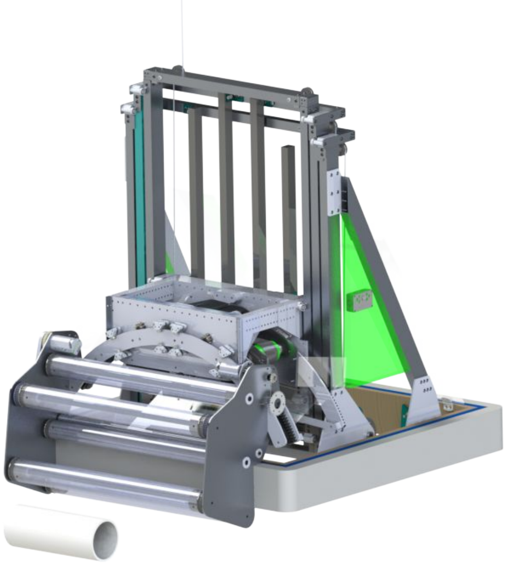
Design Process

Day 7: Simple block cad to discuss a general concept.

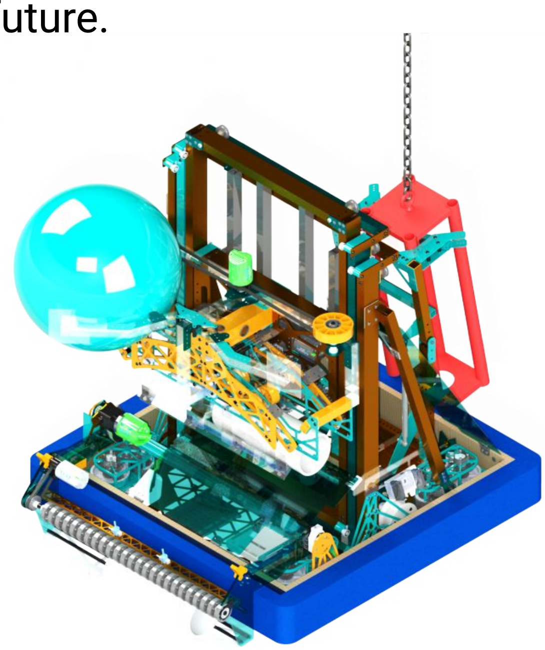
Day 14: Block cad is further expanded to include more defined areas for systems as well as mapping out electronics.



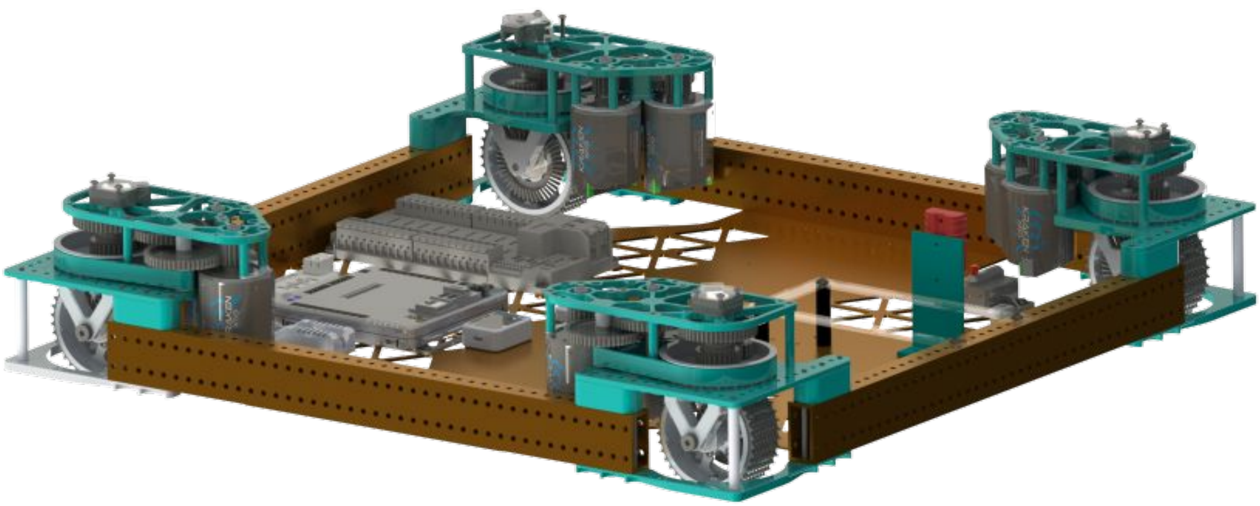
Day 21: More in depth design begins as we progress from the conceptual phase to the detailed design phase.



Day 49: Final polished CAD model with more design iterations to come in the future.



Drivebase



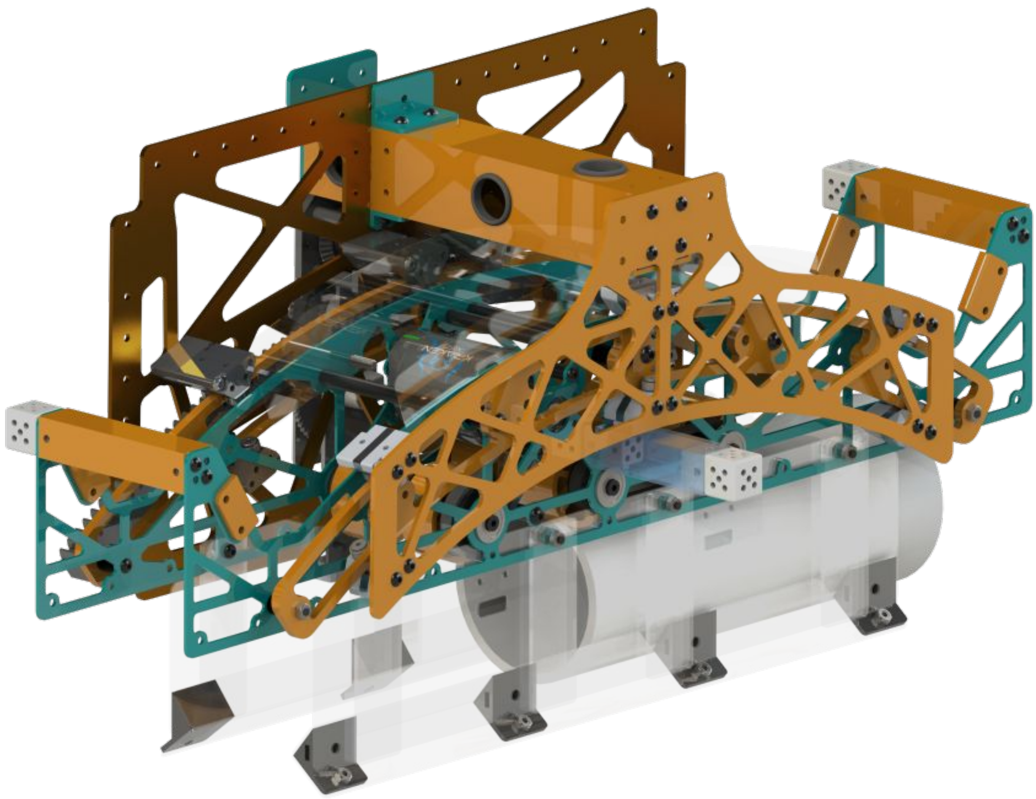
Structure

- 29.5” x 29.5” drivebase for space and stability
- Room for all electronics on drivebase
- Pocketed belly pan to save weight
- Screw and nut coupling design for quick and strong connection
- Bumpers under 20 lbs
- Guards mounted to bumper to prevent electrical component damage
- 3 material bumper design to absorb impact and create a structural bumper
- Mapped wires for clean wire routing

Swerve

- 4 MK4in L1+ swerve modules (1 Kraken X60 for drive and 1 Kraken X44 for steering) - 16T Pinion adapter
- Grip lock wheels for lightweight and high traction
- Added custom swerve plates for easy ratio change
- CTRE CANcoder for location of swerve

Carriage/Coral Scoring



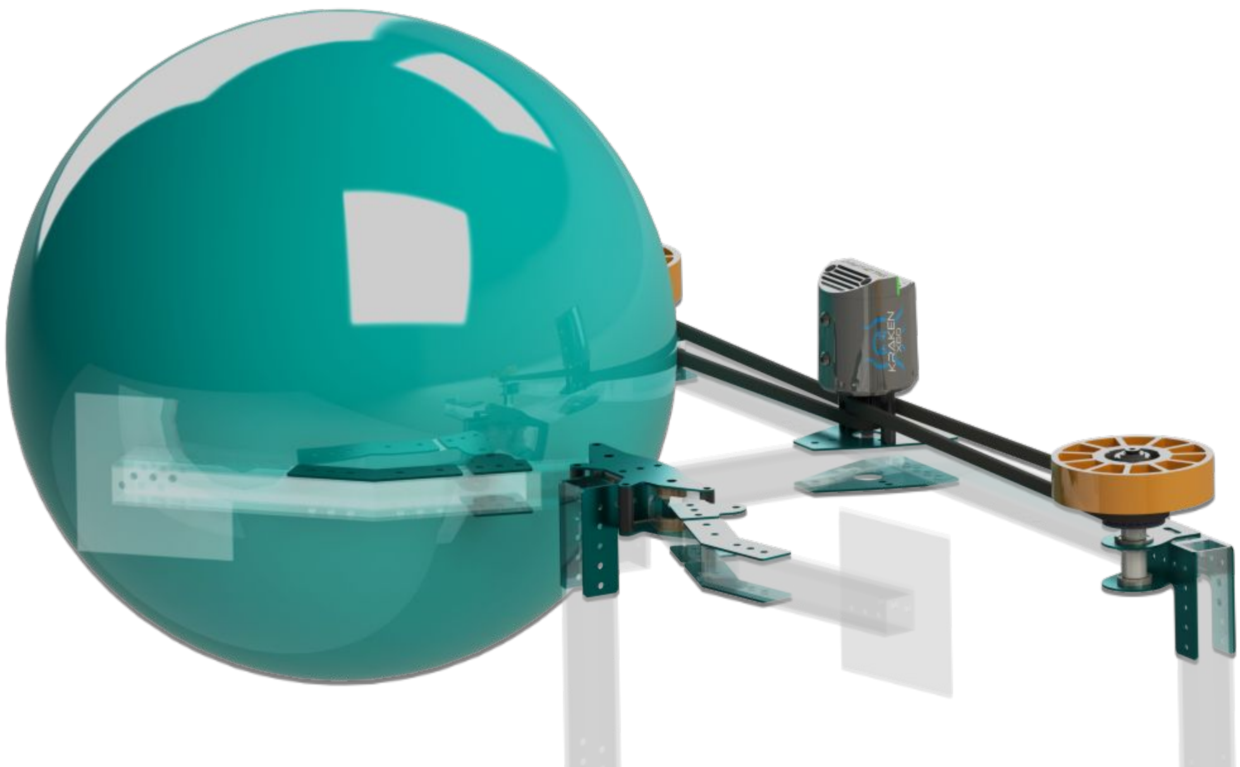
Carriage Design

- Kraken X44 to power carriage and scoring mech
- Ctre CAN coder and encoder gearbox to determine position
- WCP plate bearing blocks to run scorer side to side, Plate bearing blocks are 3d printed for Weight Savings.
- Maximum pocketing for weight savings
- 1 Kraken X44 (5:1 gear ratio), Rev 90 Degree gearbox, 10:400 pinion to rack 29 degrees of travel bi-directional

Coral Scorer Design

- 1 Kraken X44 to power movement of coral
- 3d printed carbon fiber gear for lightweight and rigid gear
- Modular polycarbonate plates to increase stiffness of grip
- Heavily pocketed plates to make movement as swift as possible
- Dual sided for maximum efficiency

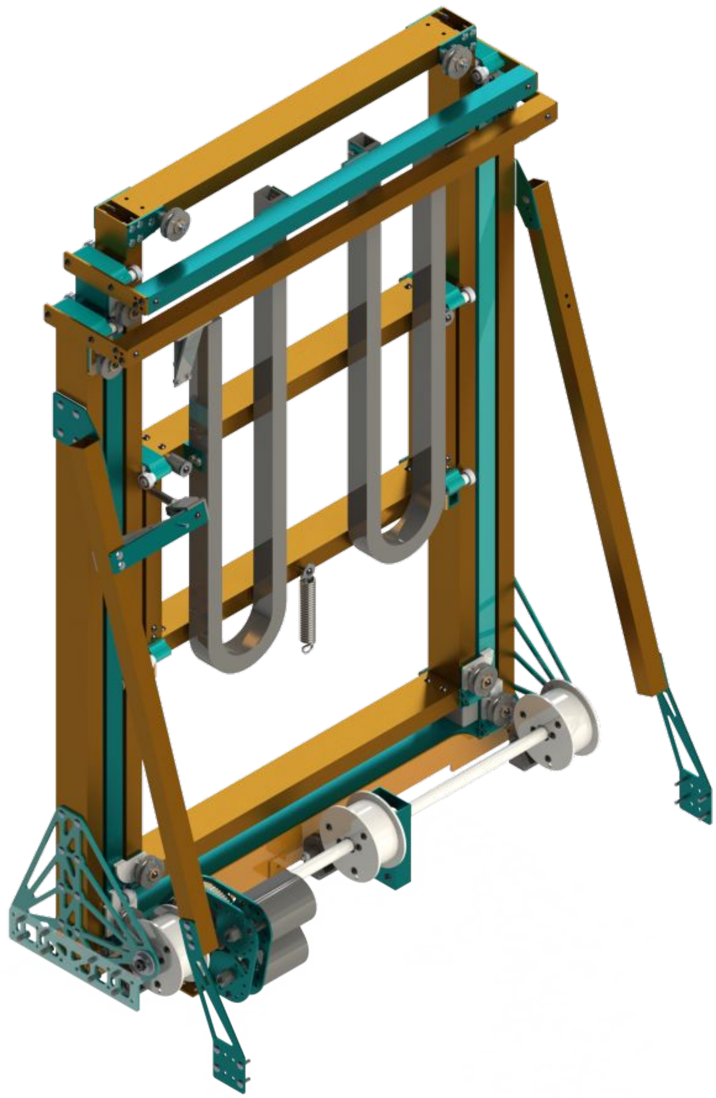
Algae Scorer



Algae Scorer Design

- Polycarbonate box tubings to keep light as possible
- 1 Kraken X44 to power movement
- Dual sided for maximum efficiency
- 11:24 gear ratio

Elevator



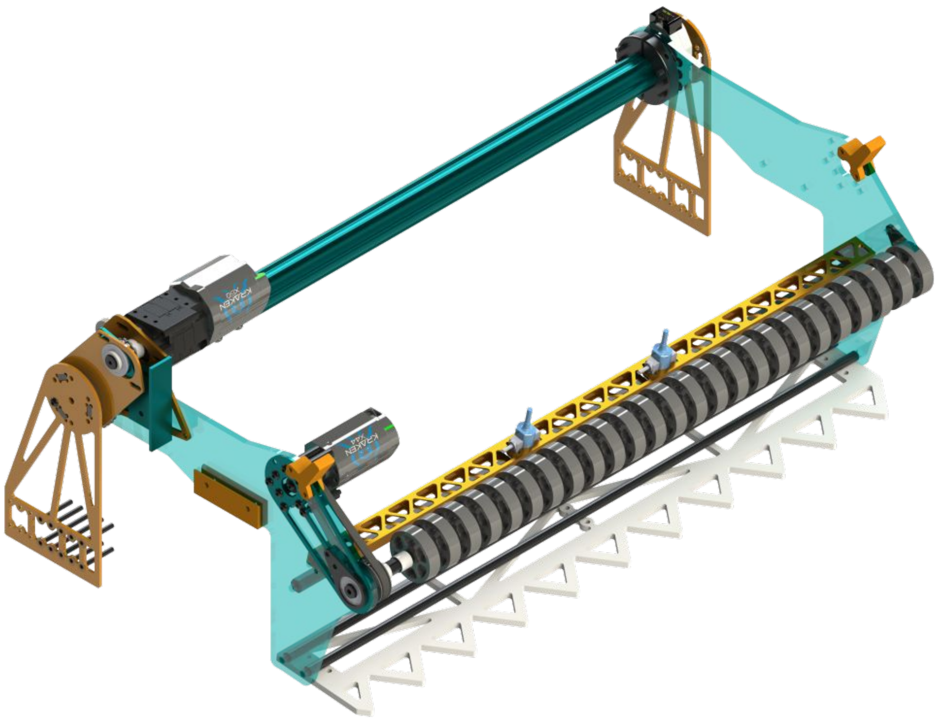
Elevator Design

- 3 stage elevator driven by a continuous dyneema rope
- Side mounted pulleys reduce cantilever of crossbar and compactness
- Travels 80 inches in <1 second and reaches a max height of 100 inches
- 2 hall sensors to determine position of algae scoring and handoff
- WCP inline bearing blocks to run box tubings
- TPU stops to reduce noise during impact

Gearbox

- Powered by 2 Kraken X60s for maximum current draw efficiency
- Single stage 7:1 gear ratio (optimized ratios for maximum speed to weight ratio)

Intake



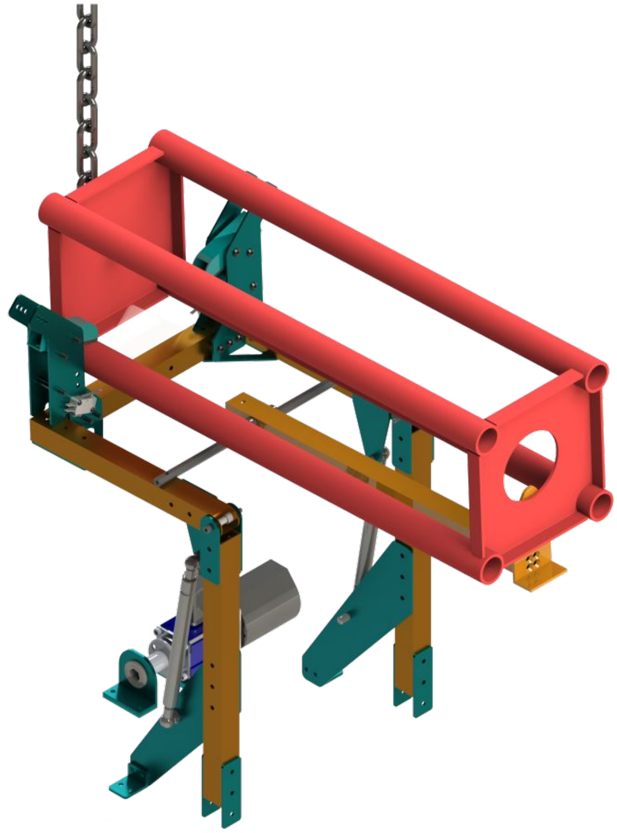
Intake Design

- HDPE plates to allow intake to slide on the ground.
- Tooth shaped design to rotate coral into the parallel orientation.
- Intake rollers is run by a Kraken X44 with 11:24 gear ratio.

Pivot Design

- Pivot motor (Kraken X44 at 1:24 gear ratio) attached to moving structure to make replacing and repairing faster and easier.
- Slotted motor mounting plate for chain tensioning.
- MAXSpline tube for torque transfer to both sides of the intake.

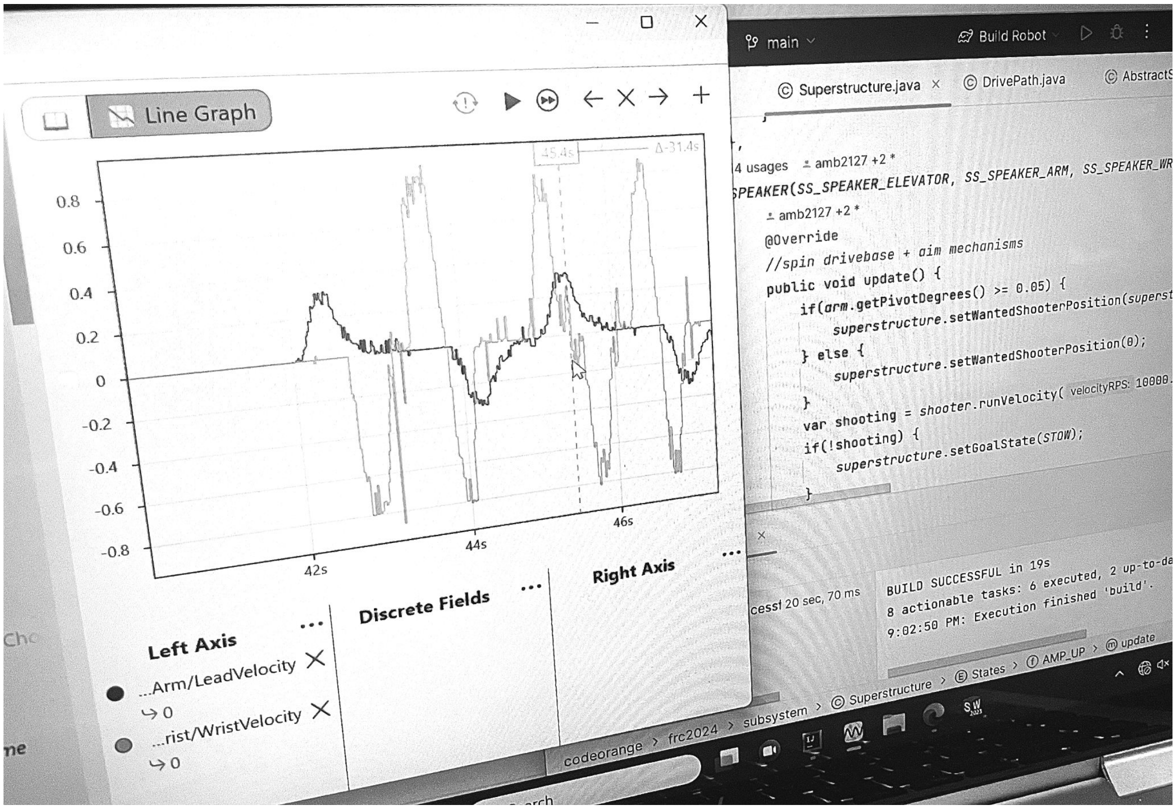
Climber



Climber Design

- Minimal alignment fast climb
- Gas Shocks allow the mechanism to move upwards and a winch pulls it down.
- 1 Kraken X60 with AndyMark Sport Gearbox (20:1 gear ratio) and ratcheting winch
- One way hooks allow the climber to latch onto the cage but not let go.
- Extension of the climber called the “foot” holds the bottom of the cage to help with optimal alignment.
- Limit switches on the hook help driver to know when the cage is fully held.
- Break beam sensors automatically stop the climber once it is at the correct climb position.

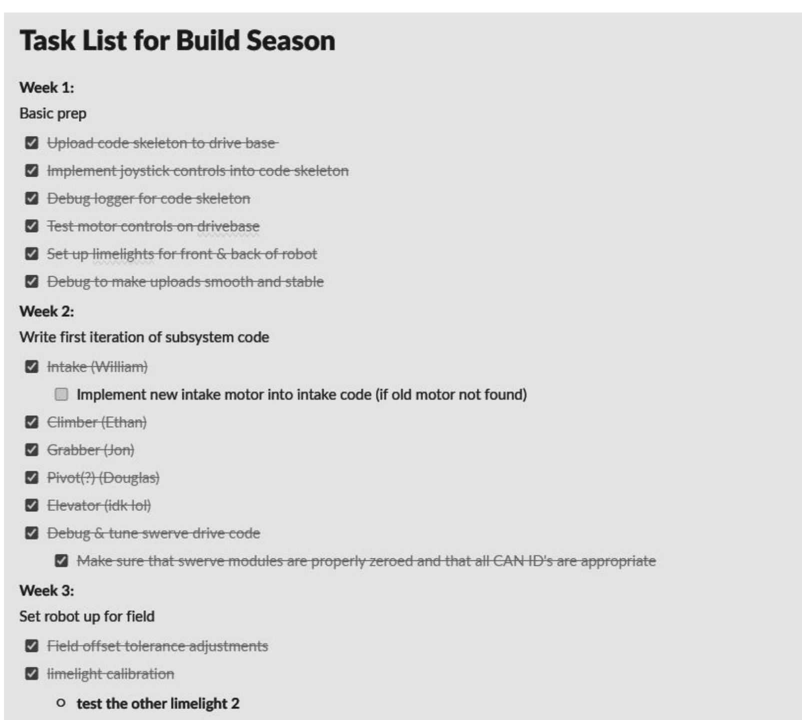
III. Programming



Programming Management

Since we have a **very small programming team (4 members)**, **organizing tasks** was a crucial step to success. Only one person in our team had previous programming experiences with FRC, so it was definitely a great learning experience for everyone.

- Organized tasks on Slack
  - Listed all tasks by week number
  - Easy to check weekly on progress
- Assigned each member to a specific subsystem
  - Able to quickly get subsystem programs written and completed
  - New programmers **gained experience** extremely quickly
  - Got the team to work together more coherently, relying on each other
- AdvantageScope was always open while testing
  - **Records all inputs/outputs** for review after
  - Ensures fixed bugs
- Elastic was crucial
  - Allowed for rapid tuning, crucial for such as small team



Task List for Build Season

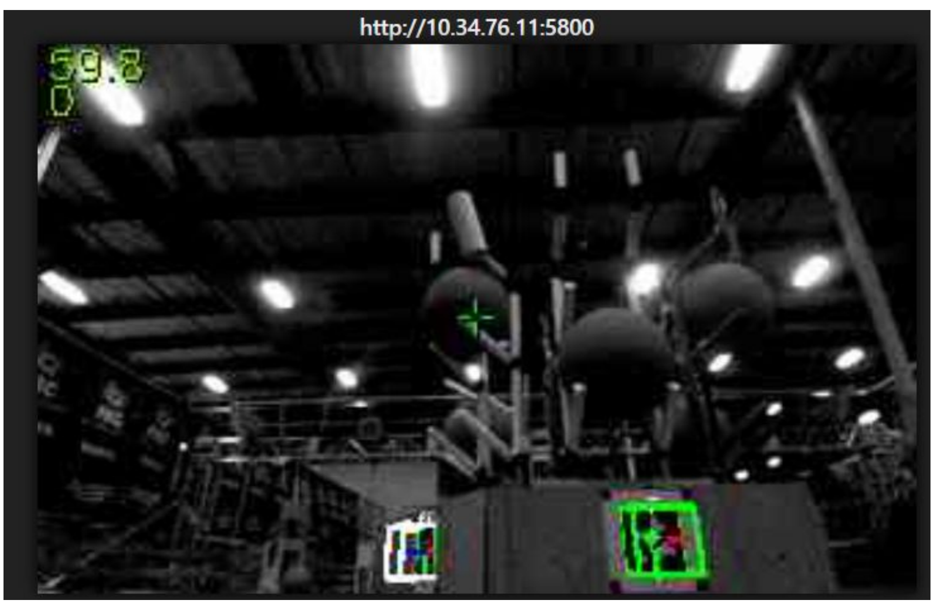
Week 1:
Basic prep
- [x] ~~Upload code skeleton to drive base~~
- [x] ~~Implement joystick controls into code skeleton~~
- [x] ~~Debug logger for code skeleton~~
- [x] ~~Test motor controls on drivebase~~
- [x] ~~Set up limelights for front & back of robot~~
- [x] ~~Debug to make uploads smooth and stable~~

Week 2:
Write first iteration of subsystem code
- [x] ~~Intake (William)~~
  - [ ] **Implement new intake motor into intake code (if old motor not found)**
- [x] ~~Climber (Ethan)~~
- [x] ~~Grabber (Jon)~~
- [x] ~~Pivot(?) (Douglas)~~
- [x] ~~Elevator (idk lol)~~
- [x] ~~Debug & tune swerve drive code~~
  - [x] ~~Make sure that swerve modules are properly zeroed and that all CAN ID's are appropriate~~

Week 3:
Set robot up for field
- [x] ~~Field offset tolerance adjustments~~
- [x] ~~limelight calibration~~
  - **test the other limelight 2**

Vision & Pose Estimation

- Limelight Vision – 2 Limelight 4s
    - **Tracks AprilTags** on field quickly
    - Detects robot **field location**
    - Easy integration and configuration
    - For **increased accuracy**, each vision measurement is assigned a level of 'trust,' and the least trusted values are thrown out
    - Is a combination of limelight and swerve module location so we aren't overly reliant on one method
- Auto routing to each coral branch
    - Uses camera pose and a set endpoint to quickly draw a path
    - Sends the information to the robot
    - Allows for rapid cycle times



Limelight detection of two AprilTags on the reef

Driver Assists

- Shooter/intake autostow
    - **Break beam** sensor attached on the wrist
    - Senses if there is coral currently in the shooter/intake
- Auto pathing/alignment for coral and algae scoring in 3 buttons with a Stream Deck
    - Allows for a visual input method that is intuitive
    - Decreases on training time for operator
    - Insures less mistakes
- Superstructure
    - Is a **finite-state machine** so mechanisms don’t damage each other
    - Allows for movement to **preconfigured states** with a single button
    - Drivers **don’t have to manually move mechanisms**

Stream Deck allows for a quick and visual input method for the operator, improving driver performance.

Autonomous

- Uses Choreo to generate time-optimized paths
    - **Object-oriented integration** of the Choreo trajectories into our code allows for easy addition of new autos whenever they are needed
    - Spreadsheet used for organization and prioritization when designing new paths
- Automates movements through superstructure
    - **Simplifies** the code even further – each auto trajectory class file is easy to read and understand
    - Avoids **unnecessary repetition** between auto paths

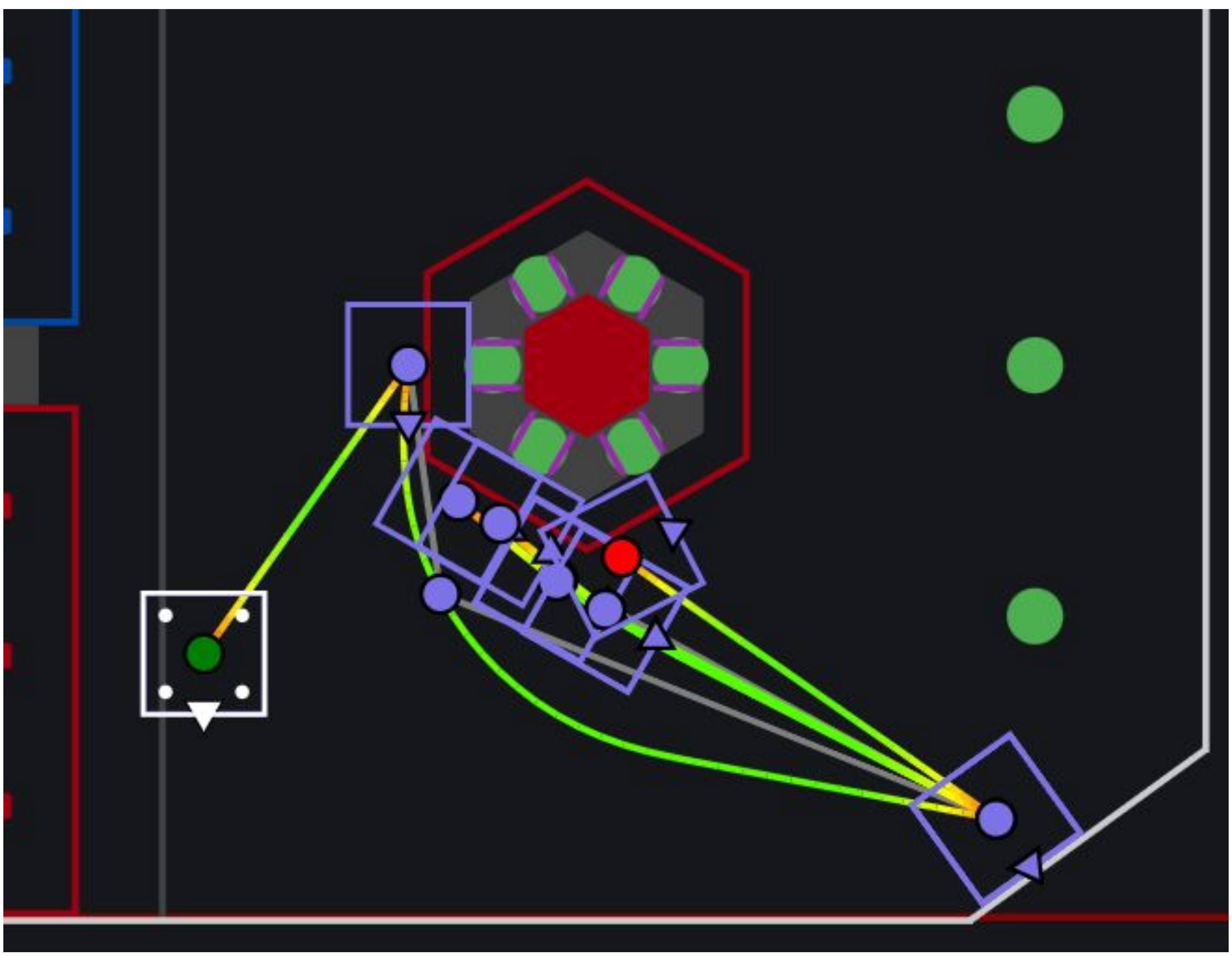
```
    if (trajectory.isPresent()){

        sequenceAction(
                new ParallelAction(
                        new ResetOdometry(trajectory.get().getInitialPose( mirrorForRedAlliance: true).get())
                )
        );
        sequenceAction(new FollowPath(trajectory.get().getSplit( splitIndex: 0)));
        sequenceAction(new SeriesAction(
                        new SetSuperstructureState(Superstructure.States.L4CORAL_RIGHT),
                        new WaitForSuperstructureState(),
                        new RunPeacesign(Robot.RobotSide.RIGHT)
                )
        );
        sequenceAction(
            new SeriesAction(
                new SetSuperstructureState(Superstructure.States.CORAL_STATION),
                new Wait( timeToWait: 0.5),
                new FollowPath(trajectory.get().getSplit( splitIndex: 1)),
                new Wait( timeToWait: 2)
            )
        );
        sequenceAction(
            new ParallelAction(
                new HandoffCoral(),
                new FollowPath(trajectory.get().getSplit( splitIndex: 2))
            )
        );
        sequenceAction(new SeriesAction(
                        new SetSuperstructureState(Superstructure.States.L4CORAL_RIGHT),
                        new WaitForSuperstructureState(),
                        new RunPeacesign(Robot.RobotSide.RIGHT)
                )
        );
```

The above program creates a sequence of actions, linking up trajectories to robot actions. By splicing the trajectory into separate chunks, we're able to loop through the sequences of the trajectory, allowing to less lines and more understandability.

The below Choreo screenshots display one of our **main autos**, which dunks the first coral, moving onto the station, and cycling 3 more corals on the branches

The triangle on the side of the robot (the square) is in the location of the front of the robot. Due to our robot's unique two-sided intake, we can take some shortcuts when cycling. Rather than needing to rotate our robot 90 degrees to dunk the corals, we simply need to turn ~20 degrees, reducing error of our autonomous paths.



Choreo allows for paths to be flipped, meaning that we only need to generate 3 paths in total to be able to play on either side.

Final Remarks

Going from an idea on a whiteboard to a fully built robot just in a few weeks isn’t easy. But we did it. The time we've spent working together has been filled with learning, teamwork, creativity, and excitement. We are incredibly proud of the robot we've built as a team and can't wait to compete with it!

Thanks to our parents, mentors, and sponsors for supporting us, believing in us, and helping us bring our robot to life!

And most importantly, we’d like to thank you for reading our technical binder, and for expressing interest in JAWS. Feel free to visit our website: “teamcodeorange.com” for more information about us and our CAD files, previous robots, Github repository, and guides.

Once again, we’re so grateful that you’re here with us on our journey. Keep your eyes peeled for more!

Thanks,
Code Orange
FRC Team 3476