TecBOT.

As we enter our 11th season, we feel confident, the path we now walk is paved with the failures and victories from the past 10 years, and we know failure is the only road to success.

SINCE 2009

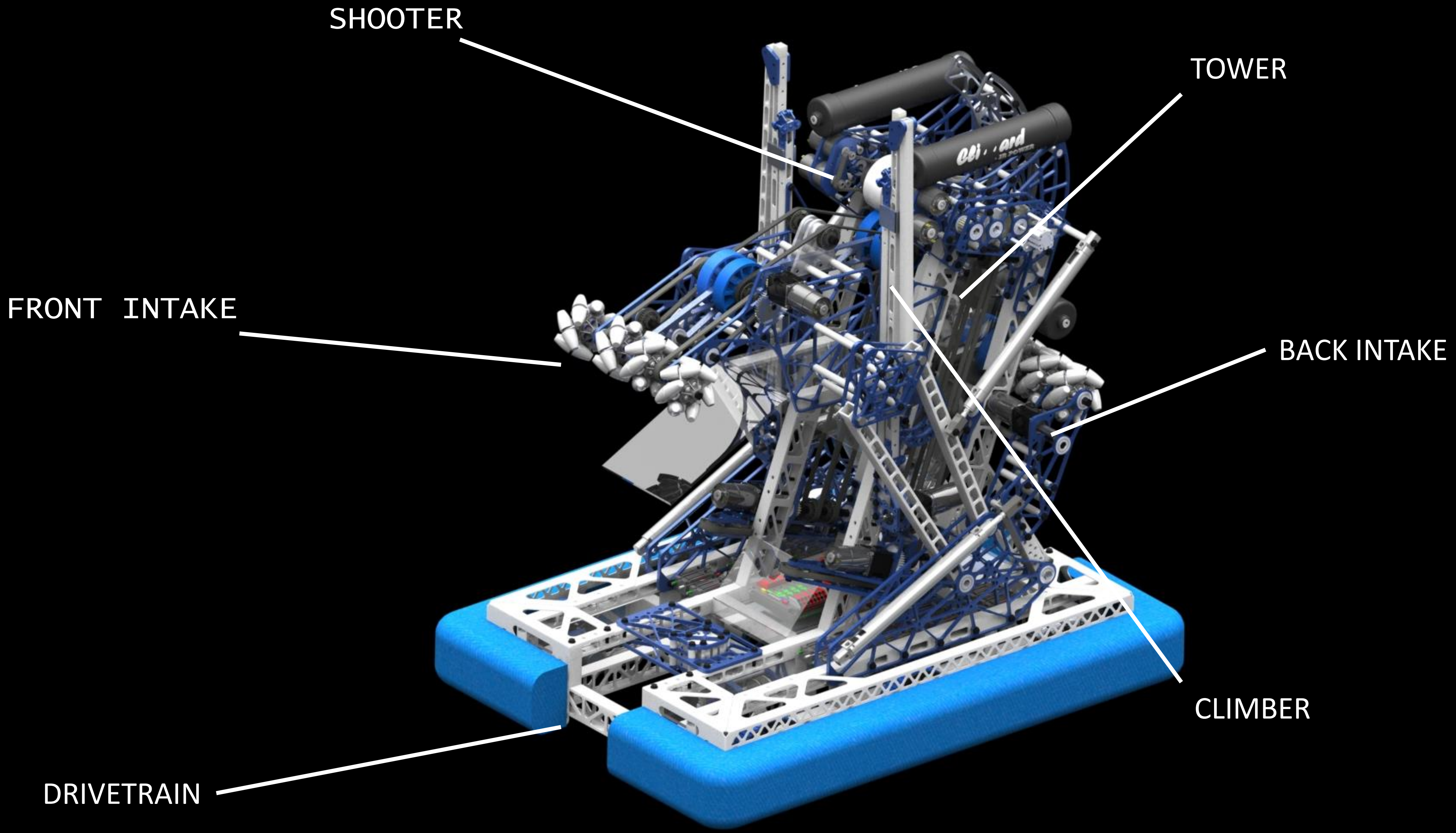
You want to know
the difference
between a master
and a beginner?
The master has
failed more times
than the begginer
has ever tried.

-Master Yoda

INFINITE
RECHARGE℠

DROIDEKA

TECHNICAL BINDER


SHOOTER
TOWER
FRONT INTAKE
BACK INTAKE
CLIMBER
DRIVETRAIN

INDEX

PAGE

Game Analysis -- 9

Robot Design Goals ----------------------------------- 11

Design --- 13

Virtual Validation and Structural Optimization ------ 16

Manufacturing -- 17

Mechanism description ------------------------------- 19

Design Highlights ----------------------------------- 39

Electrical Highlights ------------------------------- 43

Programming Highlights ------------------------------ 47

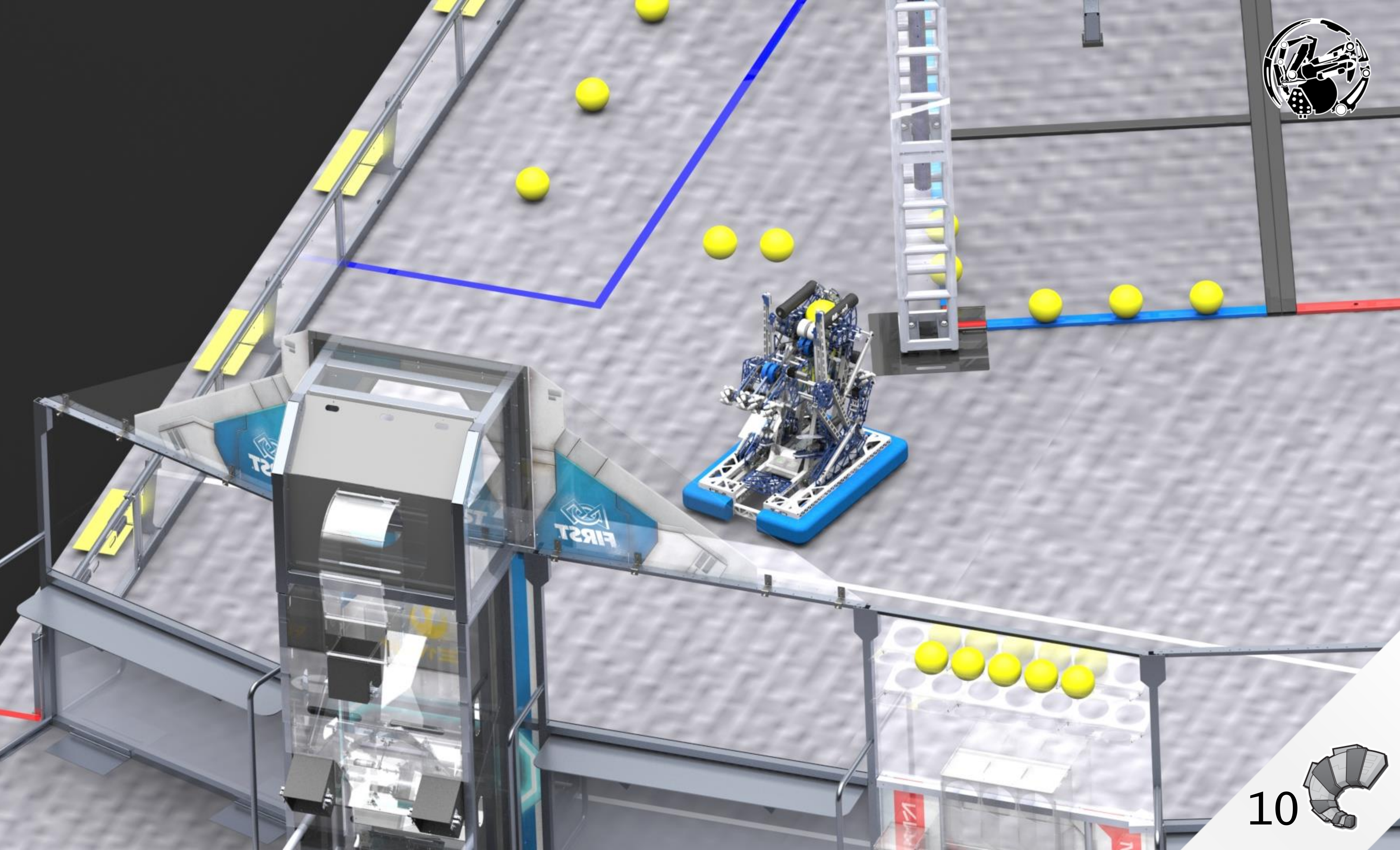
Scouting Highlights --------------------------------- 51

Sponsors -- 53

GAME ANALYSIS

Our game analysis begins right after Kickoff, when we call a team meeting to read the whole manual and brainstorm game strategies and robot mechanisms.

Our approach to this years’ robot design is based on our vision of a winning strategy, taking a solo run of the game into account. We then carry on working from the ground up, meeting each need for our robot, which we sort by priority.


FIRST
FIRST

ROBOT DESIGN GOALS

Intake

● Necessary:

○ Capable of grabbing at least 2 POWER CELLS at a time

○ Capable of grabbing POWER CELLS from either chute on the LOADING BAY

● Not necessary, but good to have:

○ Grab POWER CELLS from the front and the rear of the robot

Shooter

● Necessary:

○ Capable of shooting in either of the 3 PORTS

○ Capable of shooting 5 POWER CELLS quickly and accurately

● Not necessary, but good to have:

○ Be a turret that can pivot to aim and shoot accurately regardless of the orientation of the chassis

Spinner

● Necessary:

○ Manual drive

● Not necessary, but good to have:

○ Automated routine

Chassis

● Necessary:

○ Colson wheels for great velocity

○ Plaction wheels for defense

○ Wheel replacement of either colson and plaction depending on the match

Climber

● Necessary:

○ Scale from any point and height(middle preferably to level easier)

○ Controlled 100% by the driver

● Not necessary, but good to have:

○ Assist 1 or evens 2 robots to climb (3128 bar 2018)

Storage

● Necessary:

○ 5 power cells

Programming

● Necessary:

○ Assisted move in three Drive Types (tank, mecanum, swerve)

○ shooter speed control

○ Automatic angle adjustment

● Not necessary, but good to have:

○ Automatic Intake

○ Automatic reverse Intake

○ Automatic alignment (Shooter)

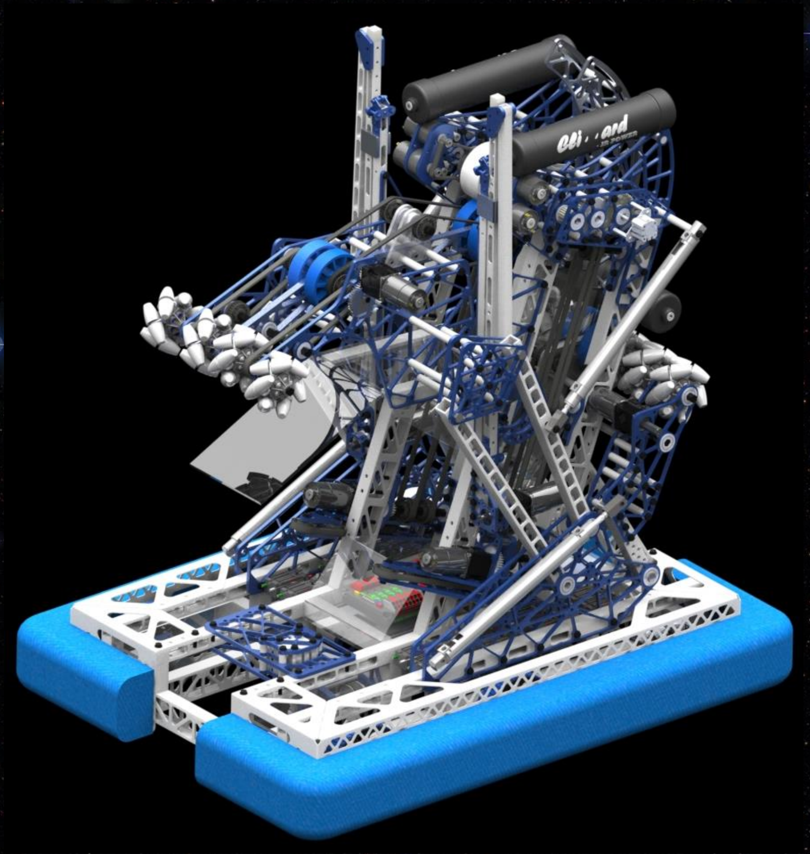
○ Automatic Rotation Control (Panel Spinner)

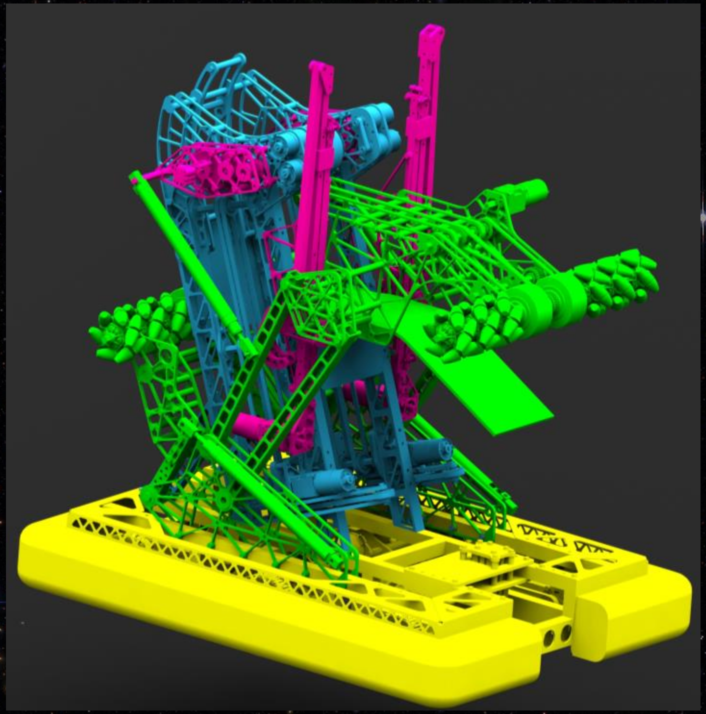
○ Automatic Position Control (Panel Spinner)

○ Automatic Climbing Action

DESIGN

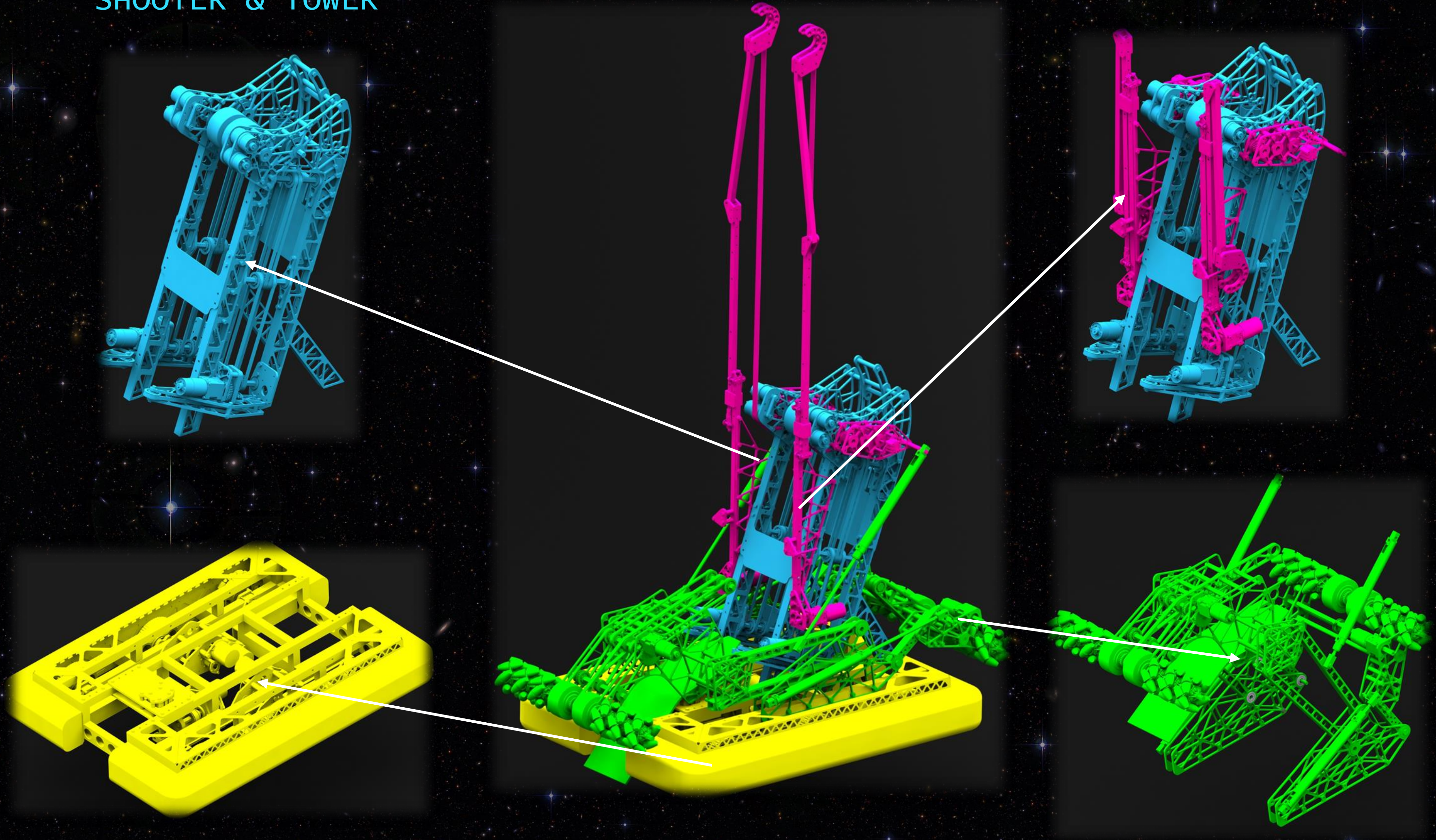
- For this season, we used the most rigorous methods of advanced simulation and design in order to fulfill the goals set for the robot.





- Each mechanism is designed individually to work as a compound, each mechanism is 100% modular, and can be replaced or swapped in a matter of minutes.

Exploded view

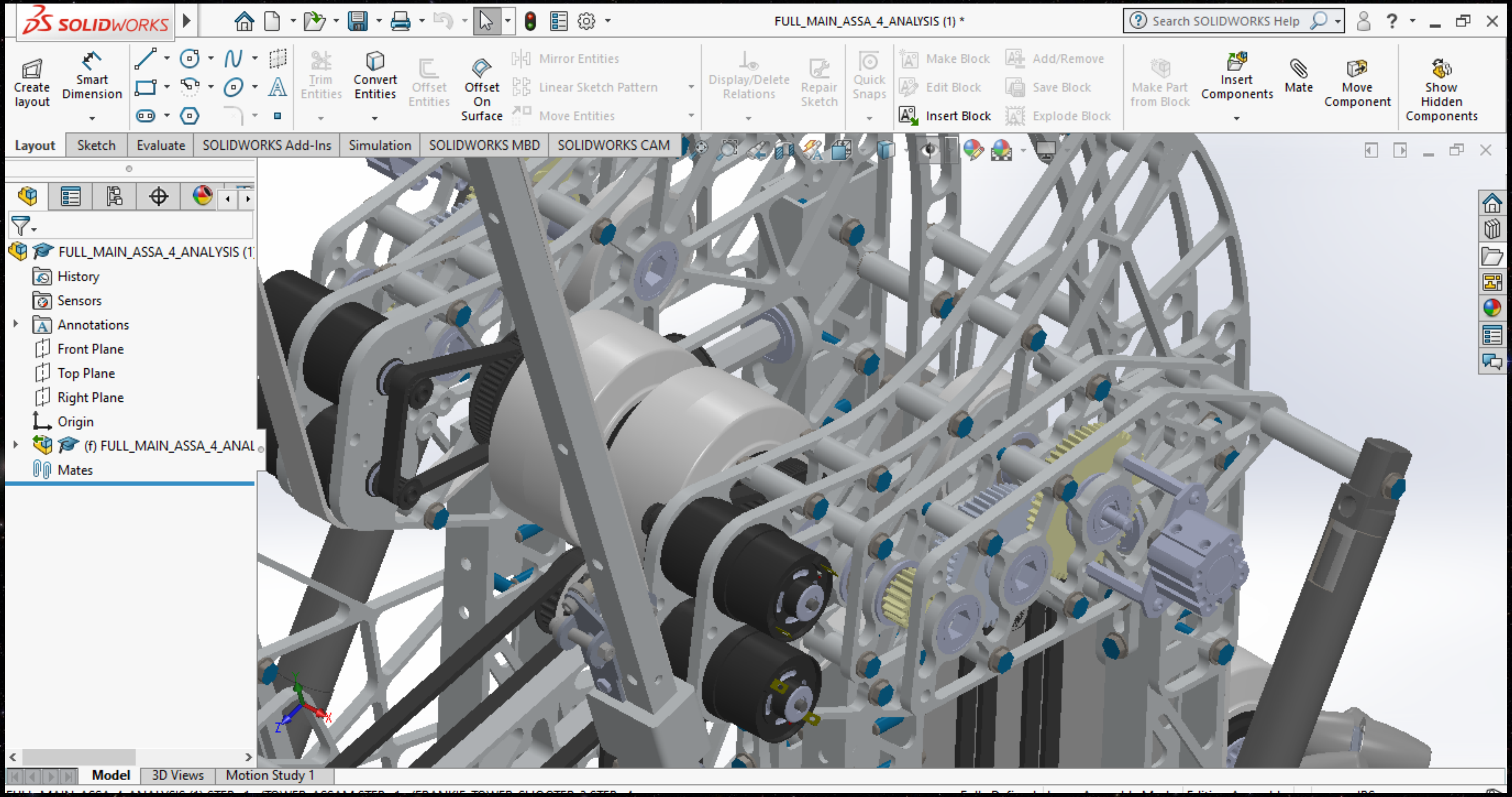


SHOOTER & TOWER

CLIMBER

DRIVETRAIN

INTAKES
(FRONT & REAR)

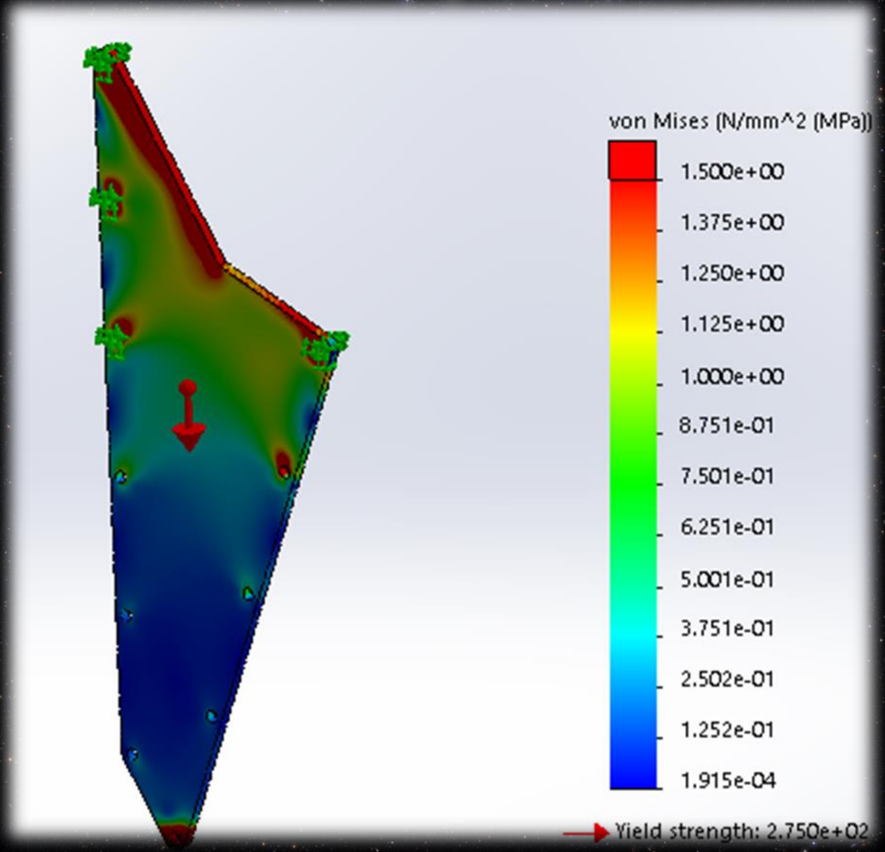


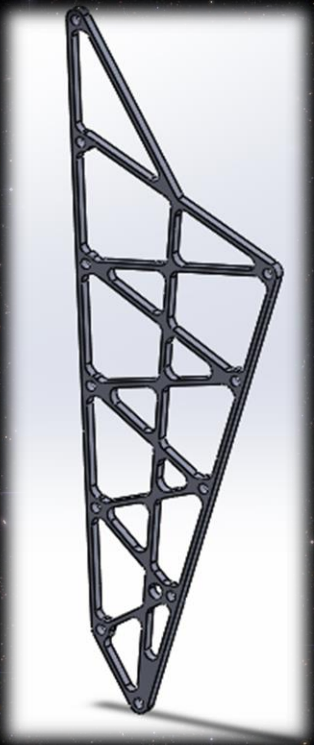
We were able to achieve modularity and strength by using many design methods, which give us many advantages but three highlight:

- Tool access feature for serviceability (meaning that our robot has both mounting holes in every plate along with a secondary hole that's suitable for the tool to get inside of next).
- Screwed extrusion, transforms every assembly into a rigid structure capable of withstanding over 300 pounds of distributed mass.
- Lightweight pieces.

VIRTUAL VALIDATION & STRUCTURAL OPTIMIZATION

- Finite Element Method has been used to obtain a lightweight design. In image below the stress contour of structural parts is showing the high stressed areas





- Topology Analysis has been used to optimize the structure and minimize the material, this method is mostly used within aeronautics and space engineering.
- We didn’t stop at implementing aeronautical methods in our design, also materials engineering has been considered, our robot is completely built on 6061 T6 SS Aluminum Alloy.

MANUFACTURING

- The majority of our robot was manufactured through waterjet, a process consisting of high pressured water cutting through metal.

- This process is not only efficient, leaving behind almost none residue, but also very quick, which helps us optimizing the time we have for the building season

- We know as a fact, most teams and brands within FRC use aluminum alloy 3036, we went to further analysis of the available alloys, and decided to use the same alloy used for aeronautics due to it’s resistance and lightness.

- In order to have a rigid structure to withstand the load of the total robot once hanged, we used advanced welding techniques that allowed us to mount every component and structure into a single, complex and rigid structure that helps maintain structural integrity.

MECHANISM DESCRIPTION

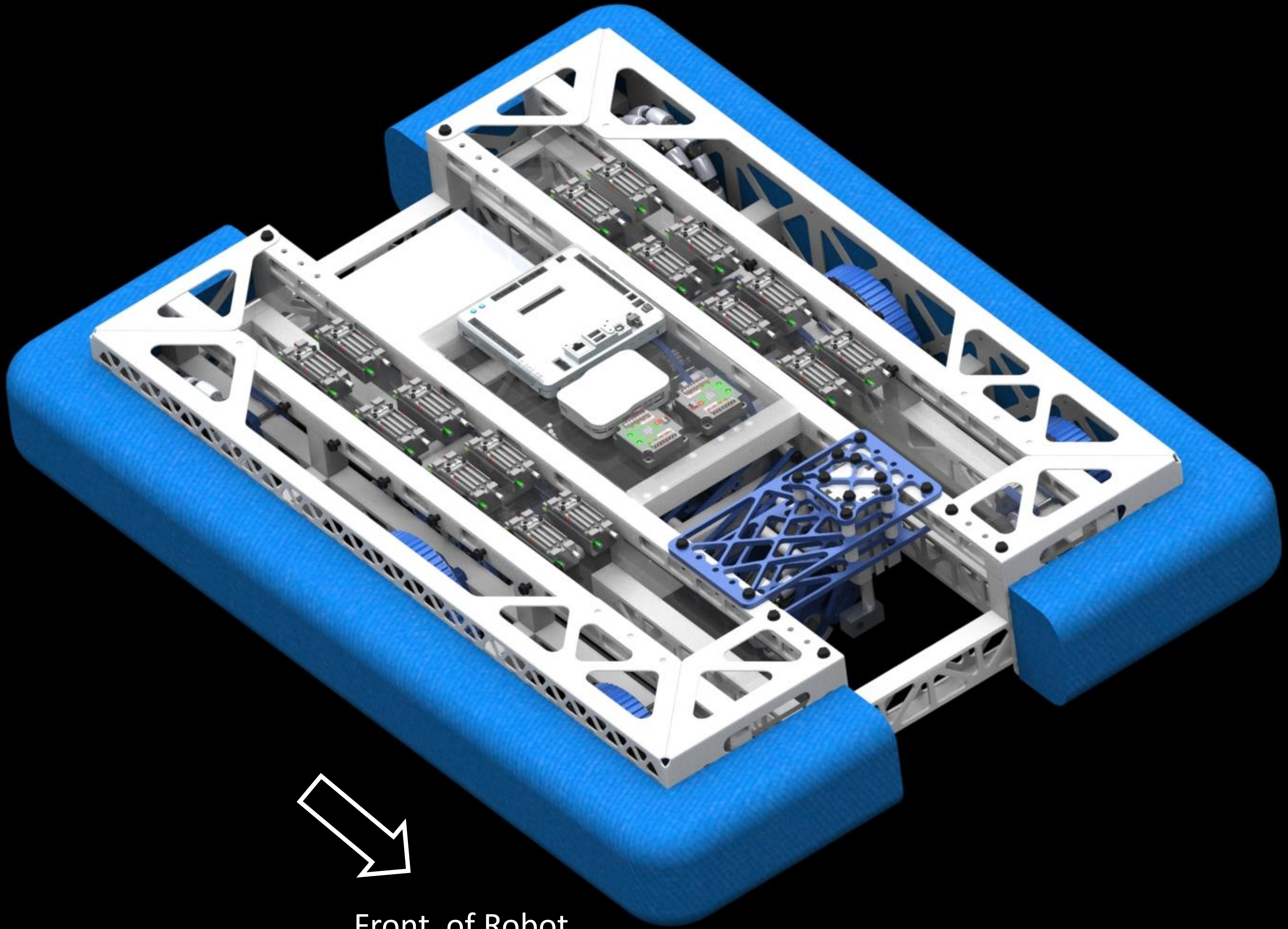
DRIVETRAIN

- Featuring our brand new Dragonfly drive, with the Dragonfly Drop Module.

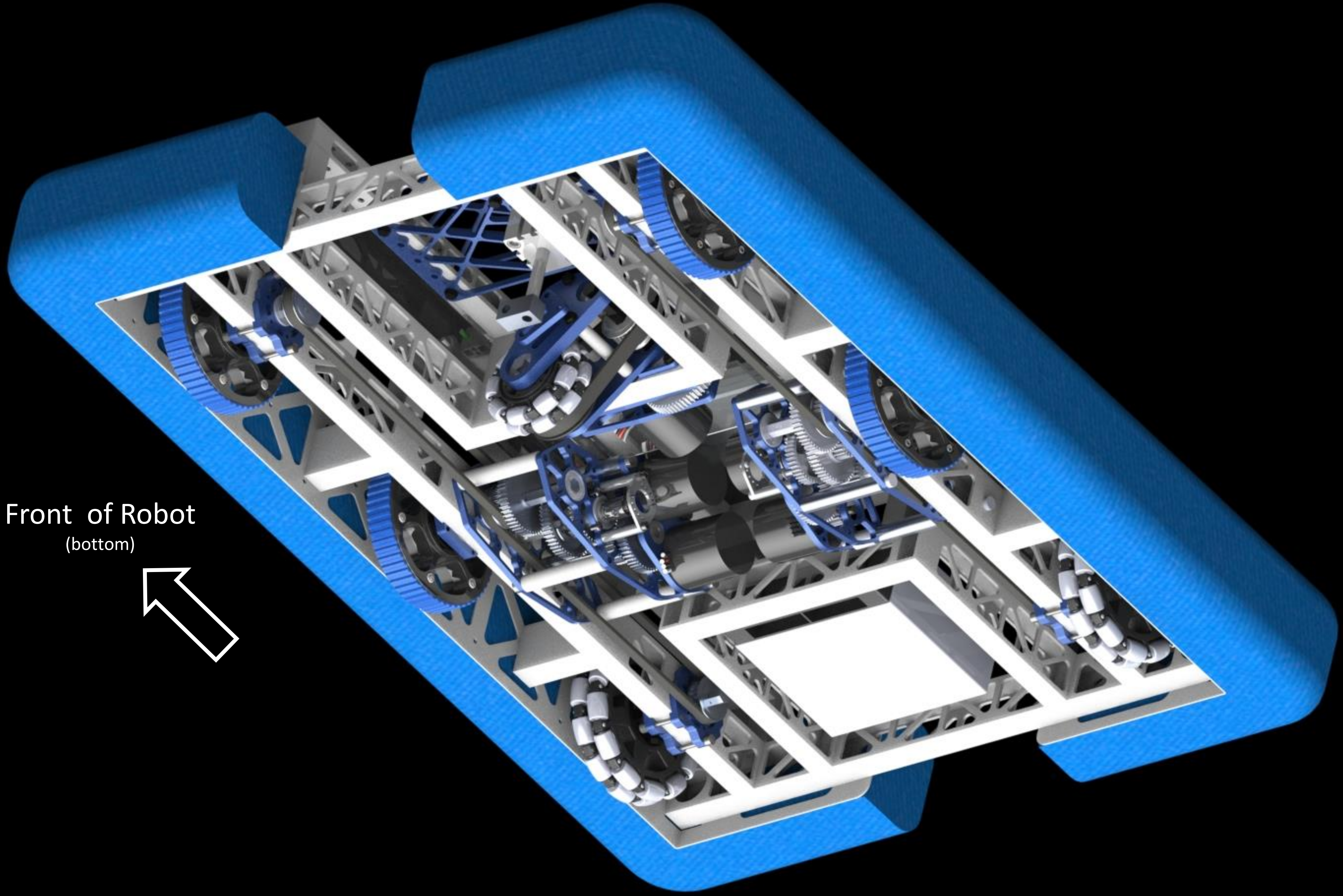
- Dual dog shifter custom gearboxes, with torque and speed gear modes.

- Powered by 5 NEO brushless motors, compact in package and provide comparable power to 8 mini CIMs.

- Front 6” Plaction wheels to gain traction and rear 6” Omniwheels to help traverse, also an additional 4” Omniwheel to switch drive modes.

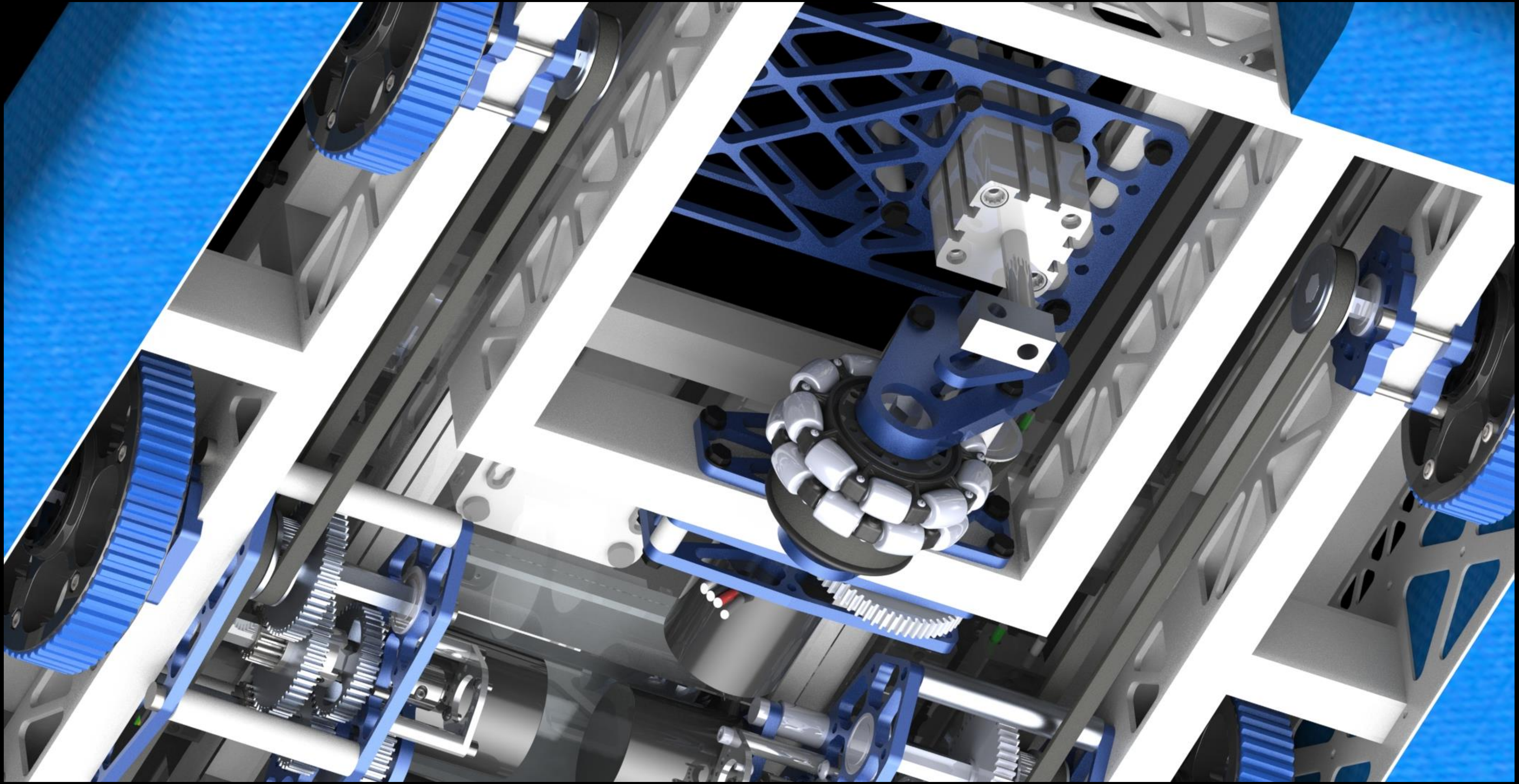


Front of Robot
(top)



Front of Robot
(bottom)

DRIVETRAIN (innovation)



INTAKE

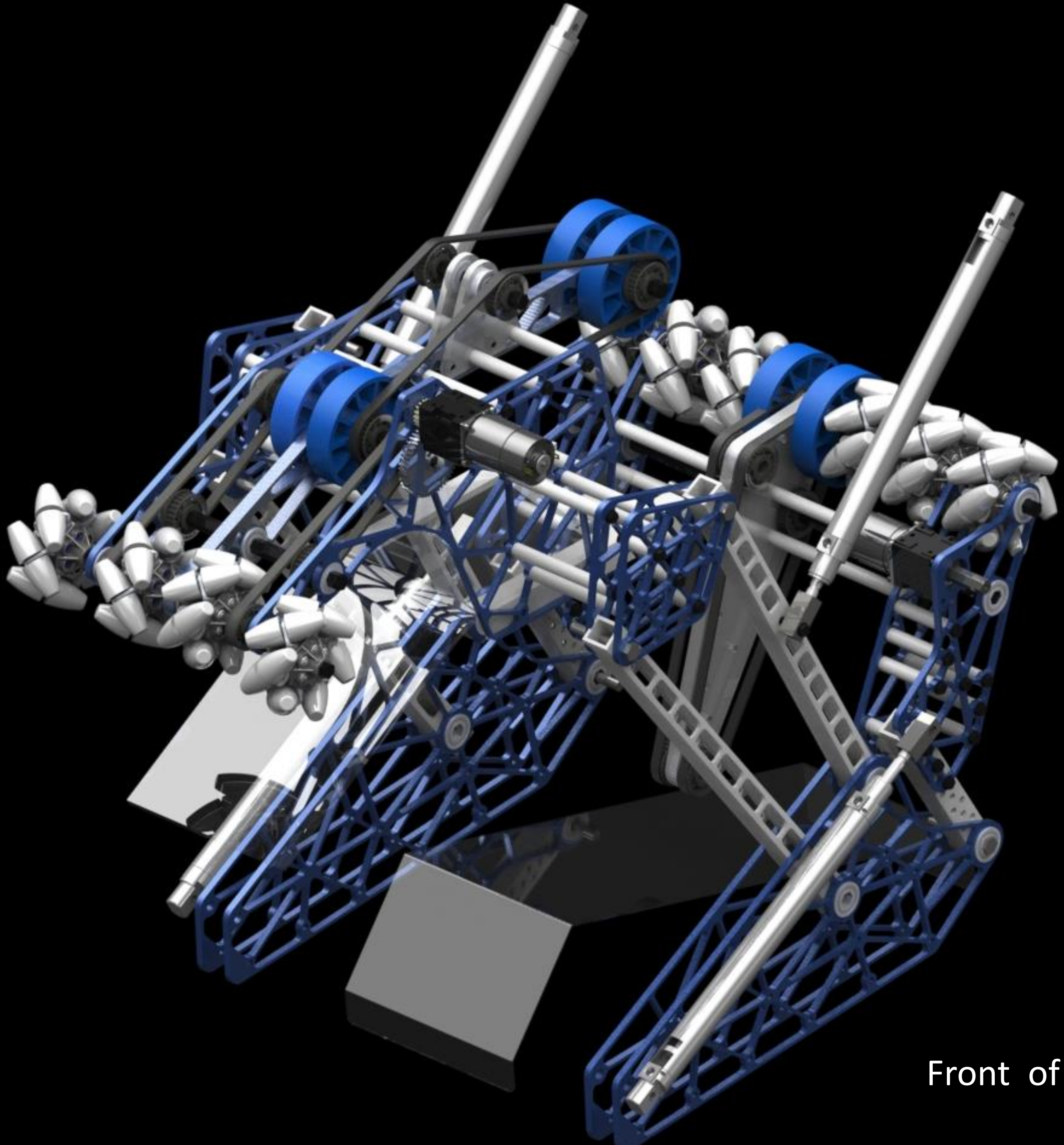
- 4” Mecanum wheels help centering the Power cells.

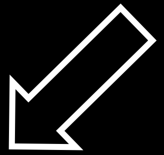
- Blue Compliant Wheels help maintain grip on the Power Cells.

- Each intake is powered by a pulley system connected with a 775 pro with a 100:1 gear ratio.

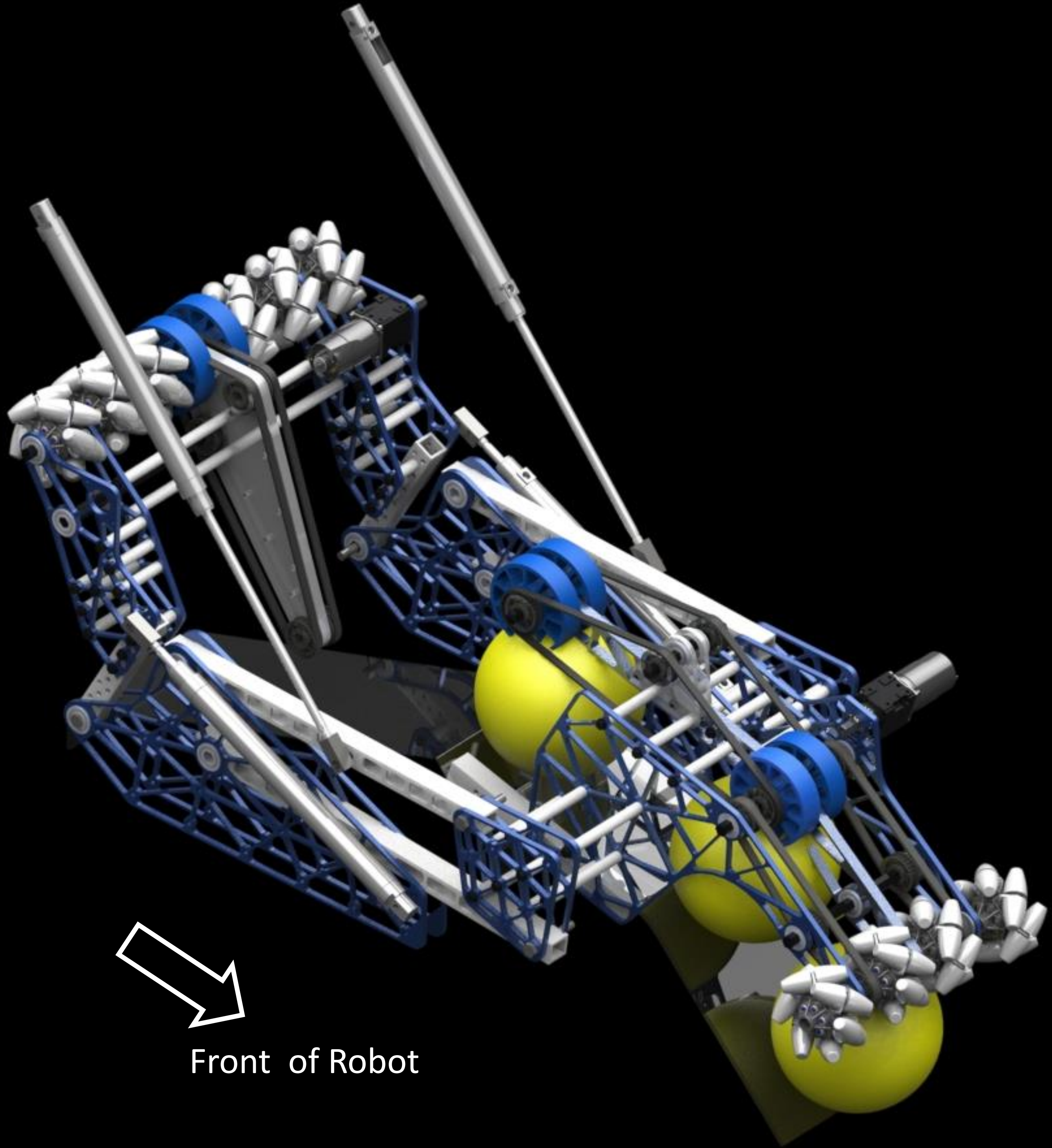
- Front intake doubles as both intake for the floor and the feeder, and as an outtake for the lower port.

- Mecanum wheels have the power to spin the Control Panel

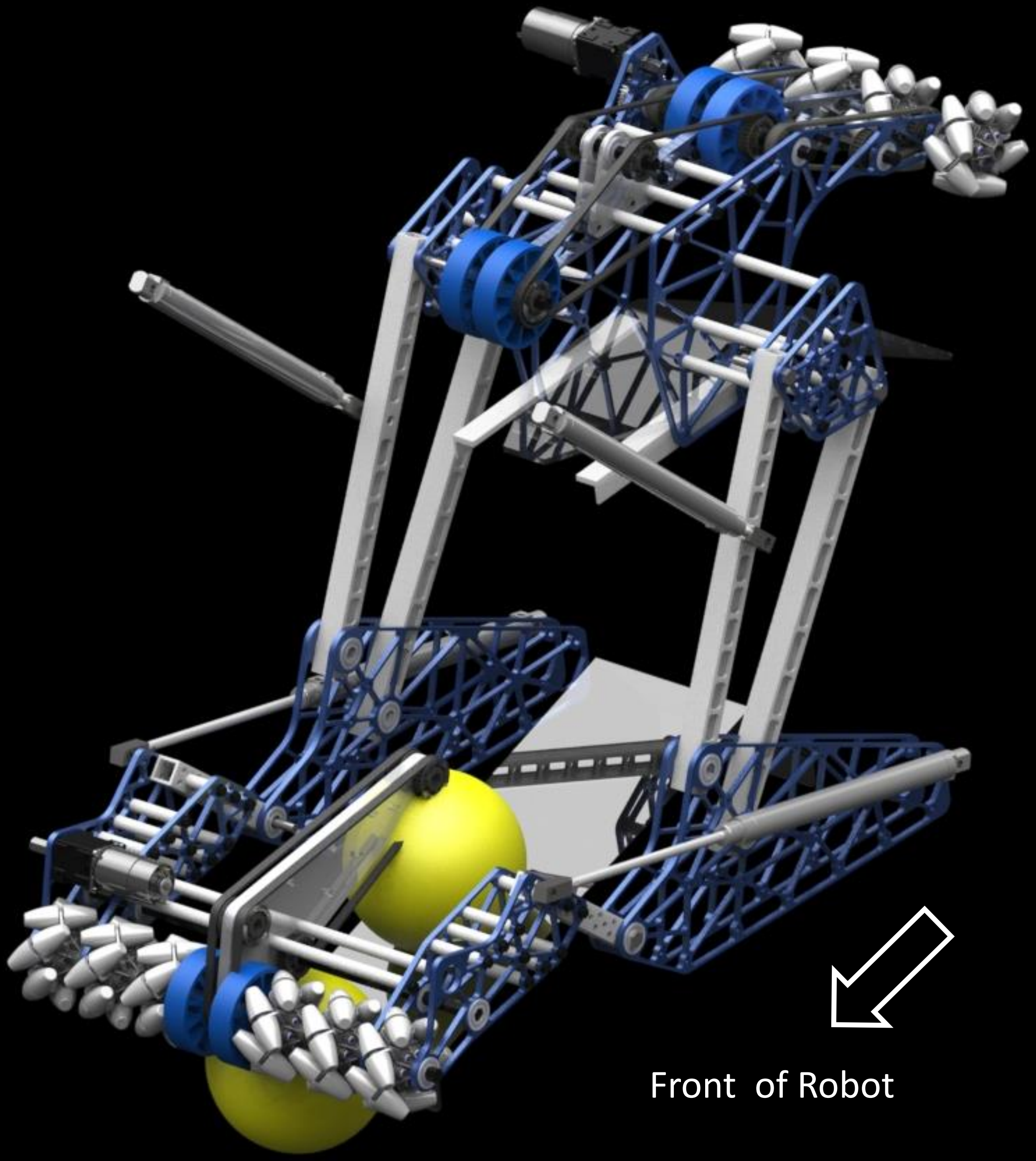


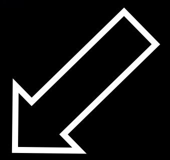


Front of Robot



Front of Robot

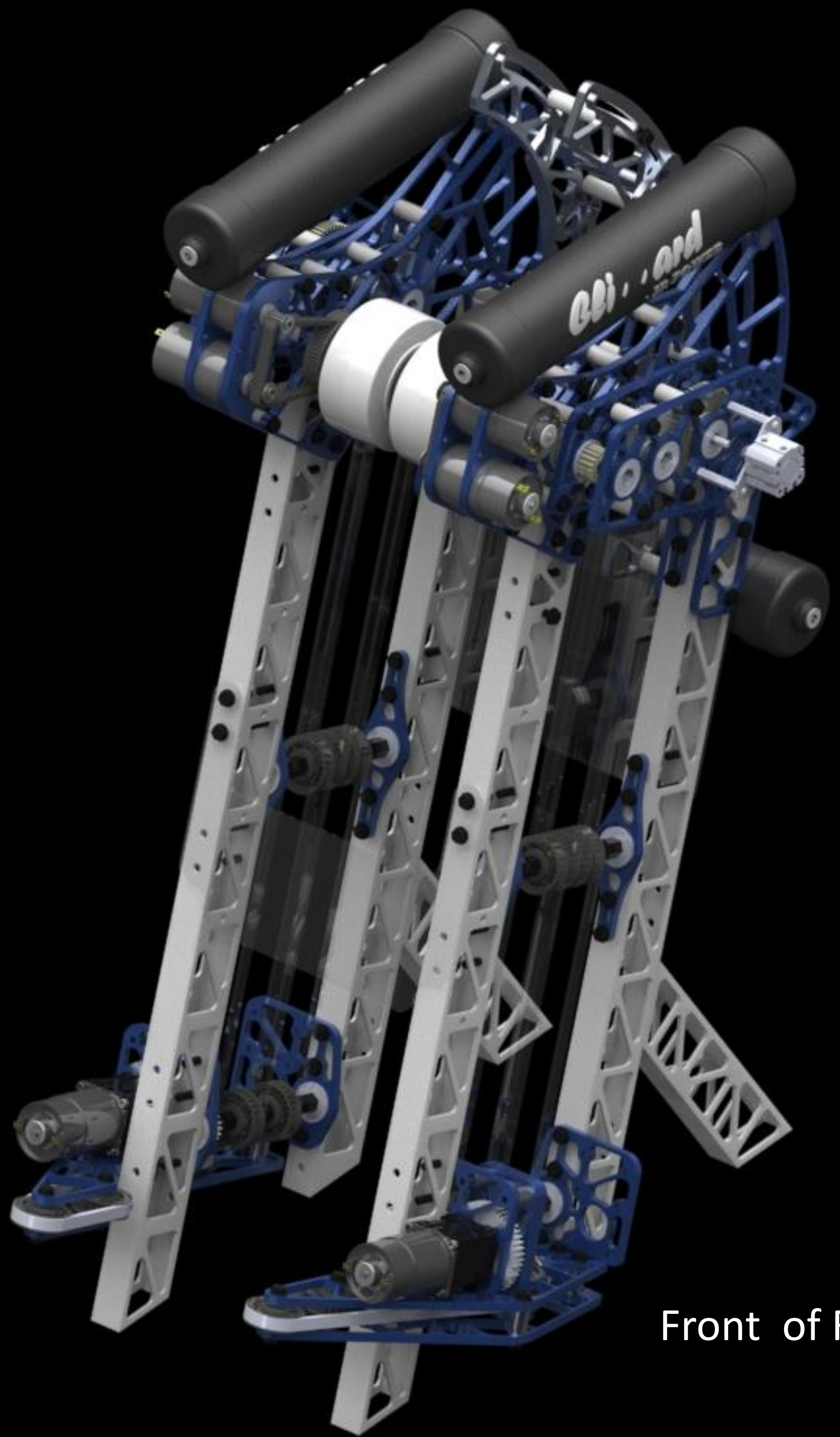




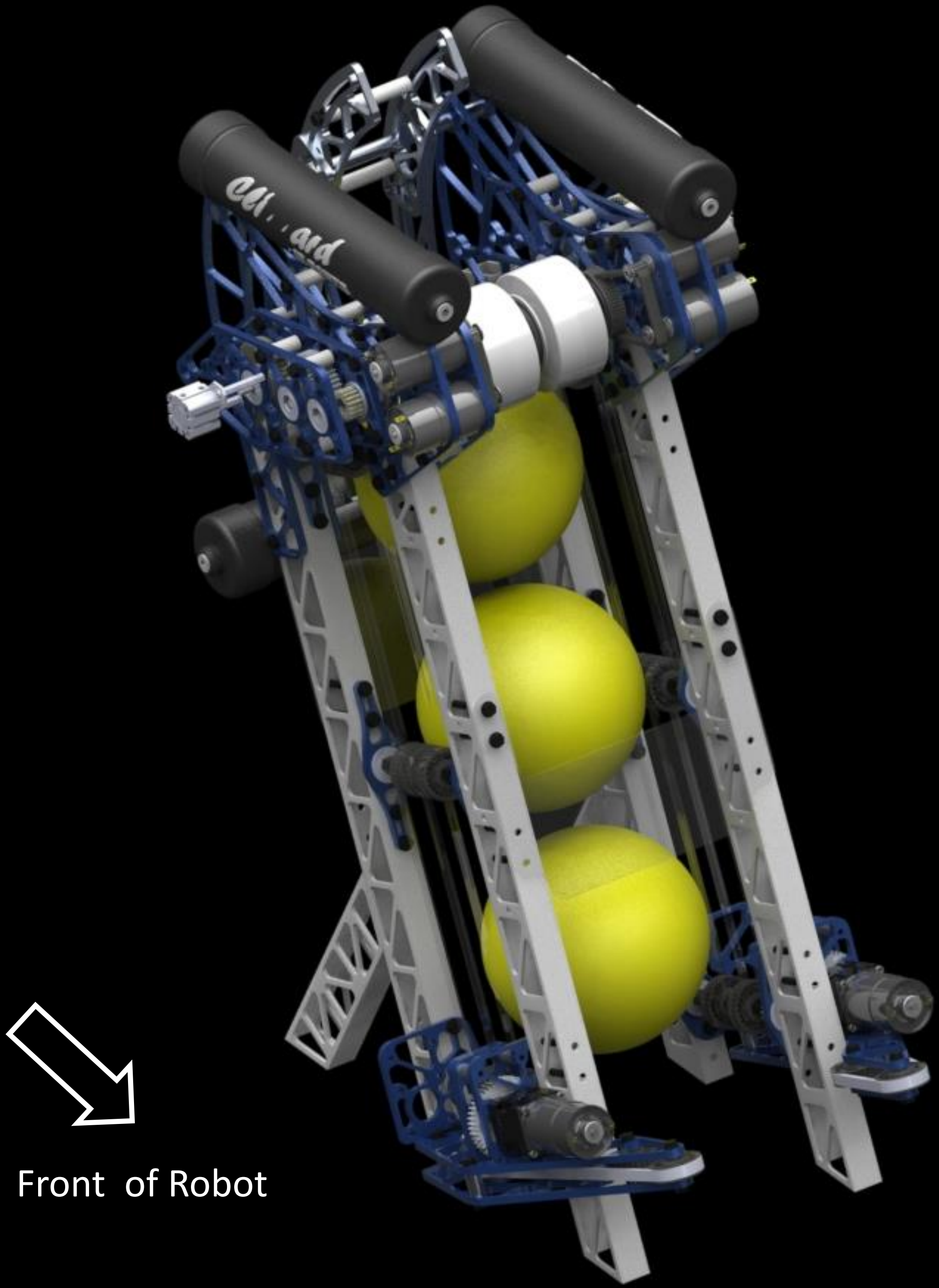
Front of Robot

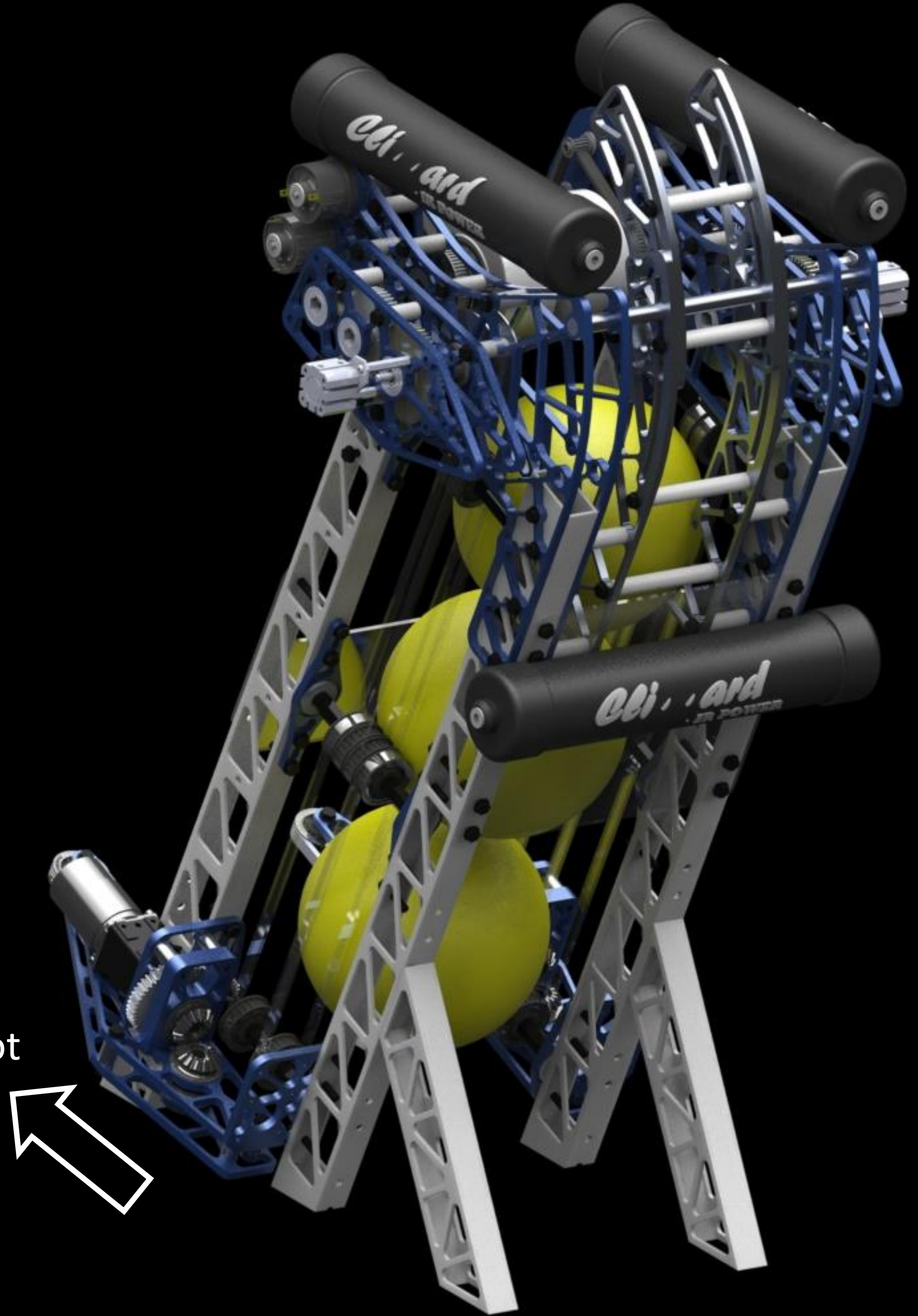
TOWER

- Capable of holding 3-4 power cells within itself
- Power Cell transportation with Belt and Pulley System, powered by two 775 pro with a 70:1 gear ratio.
- Solid structure doubles as both transport for power cells and as a solid structure suitable for climbing.
- Tall design allows for better accumulation and delivery of power cells, while also giving rigidness to the mechanisms attached to it.



Front of Robot

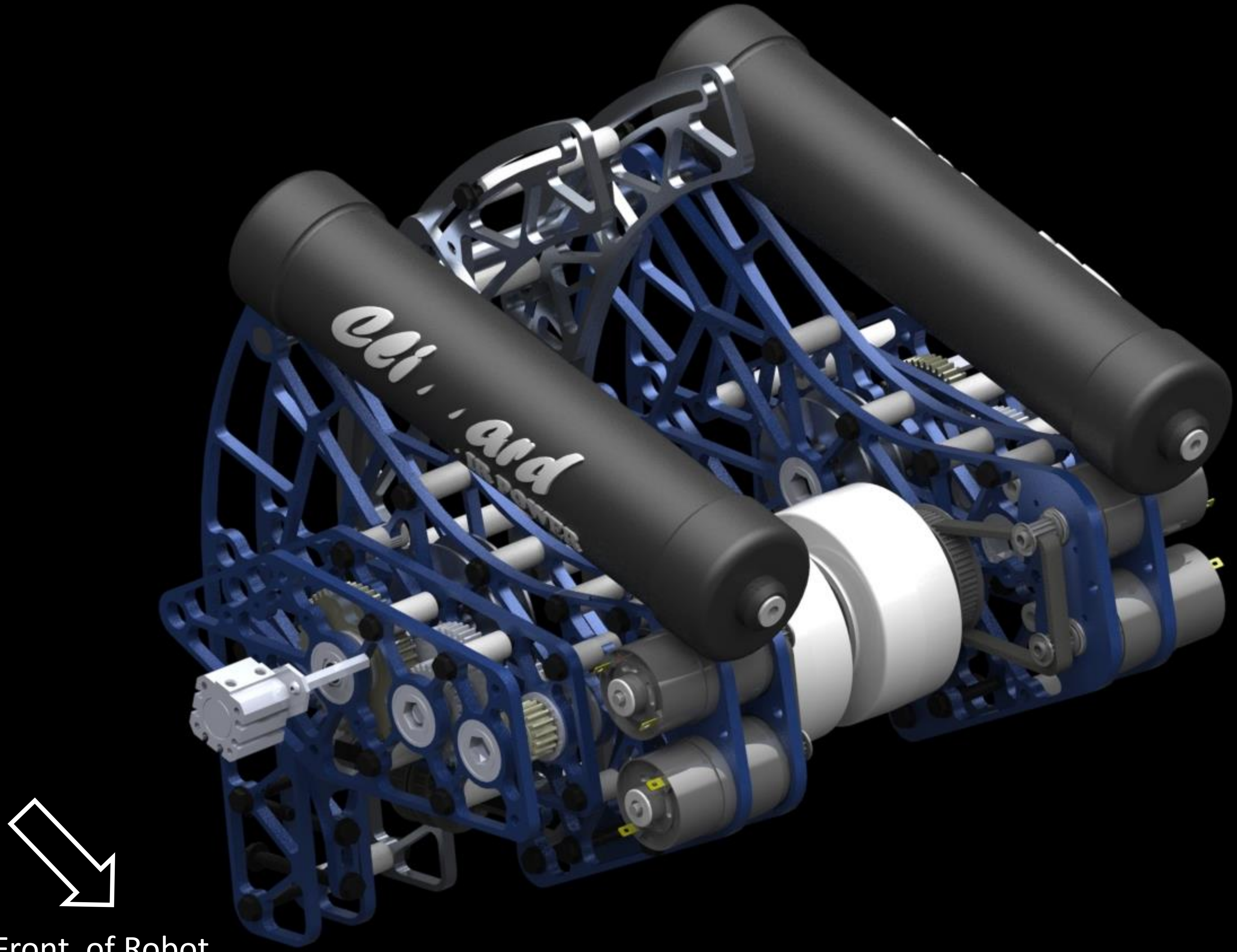

Front of Robot



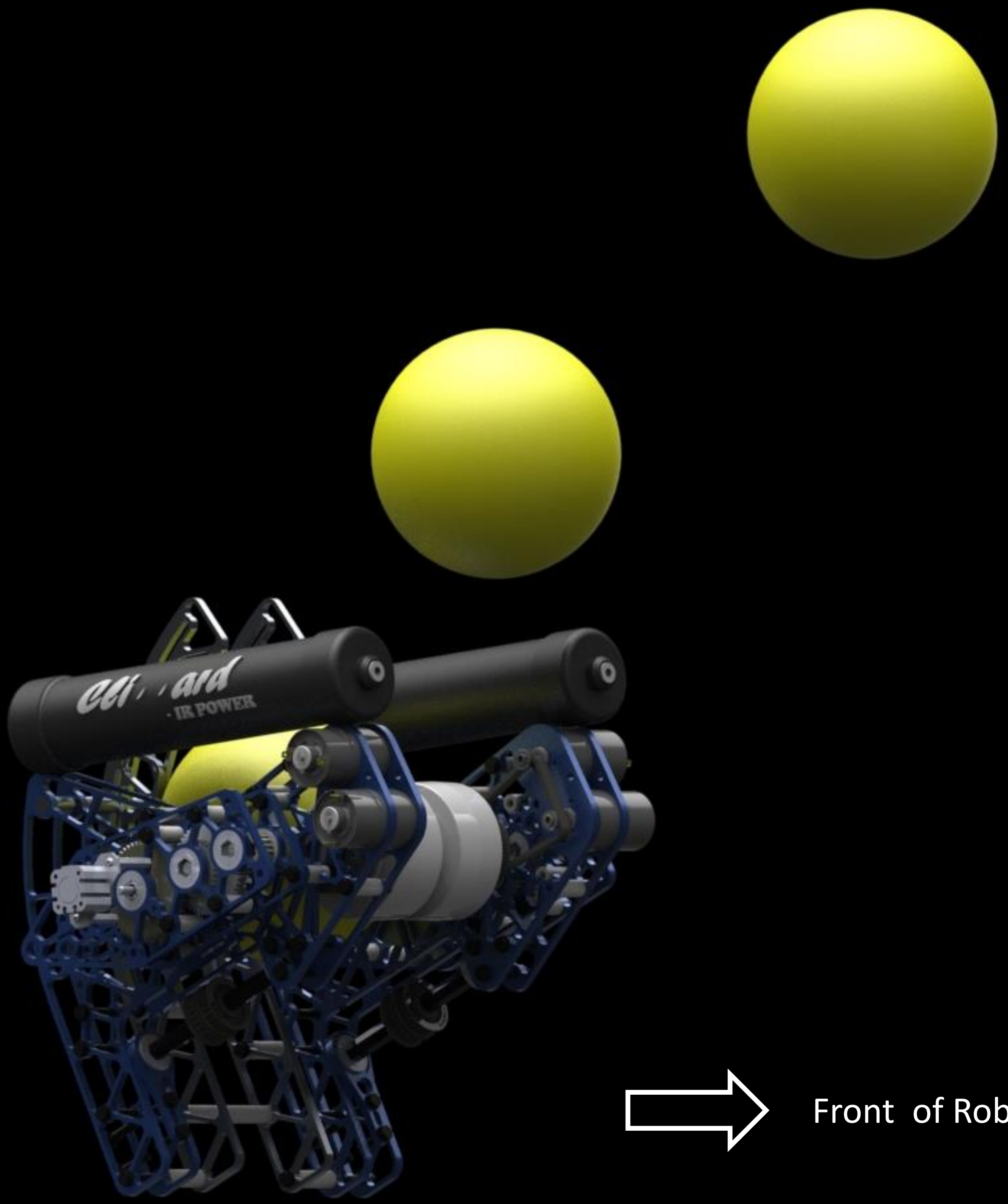
Front of Robot

SHOOTER

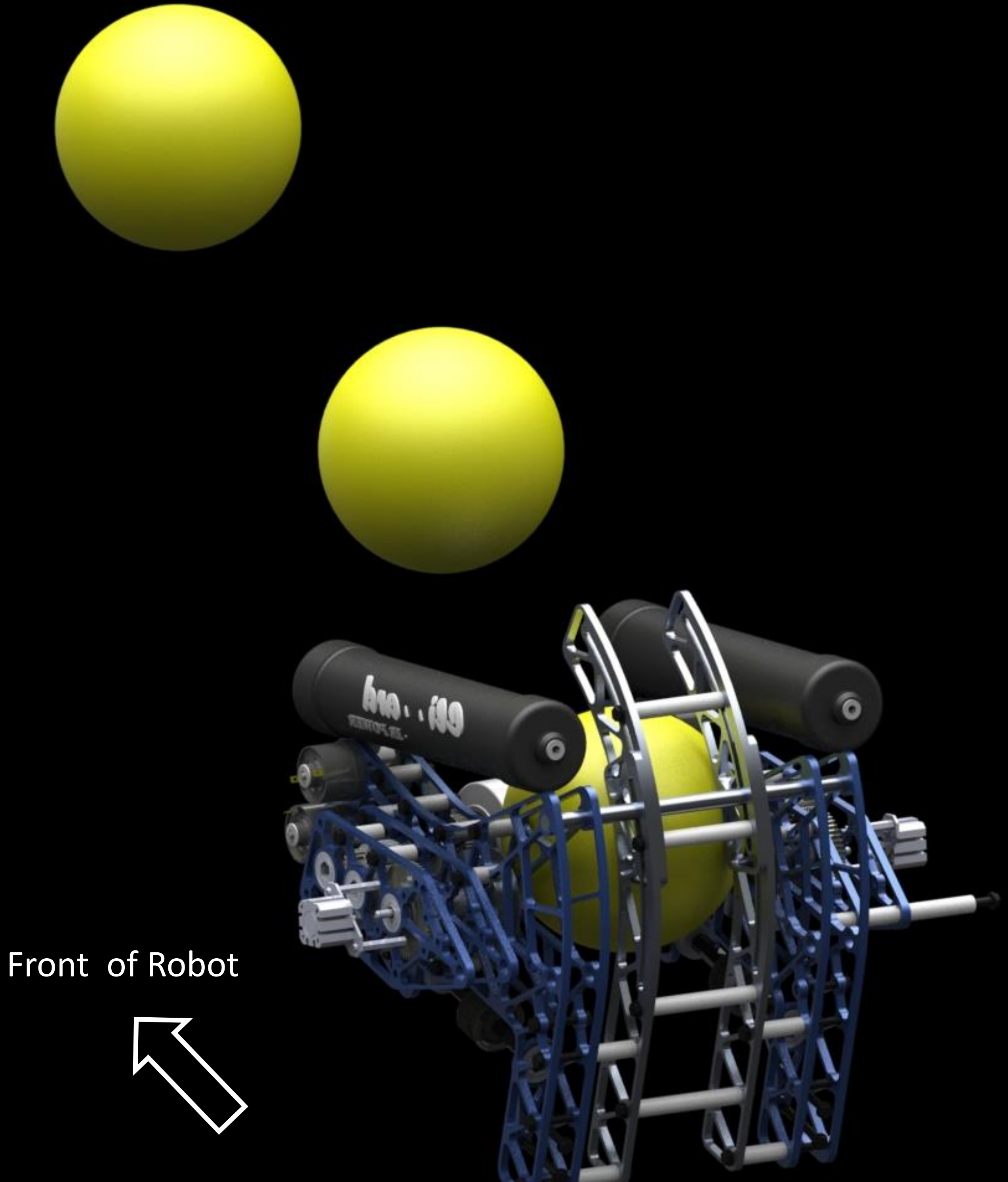
- Two 4” Colson Wheels double as both high grip shooting wheels, and a flywheel for continuous shooting.
- Powered by 4 775 pro with a 4:1 ratio given by a GT2 timing belt & pulley.
- Side plates double as both structure for pneumatics and housing for the climber’s transmission.
- Consistently shoots up to 3 Power cells per second in the Inner Port with 70%+ accuracy.
- Adjustable angle to ensure precise calibration.



Front of Robot



→ Front of Robot


Front of Robot

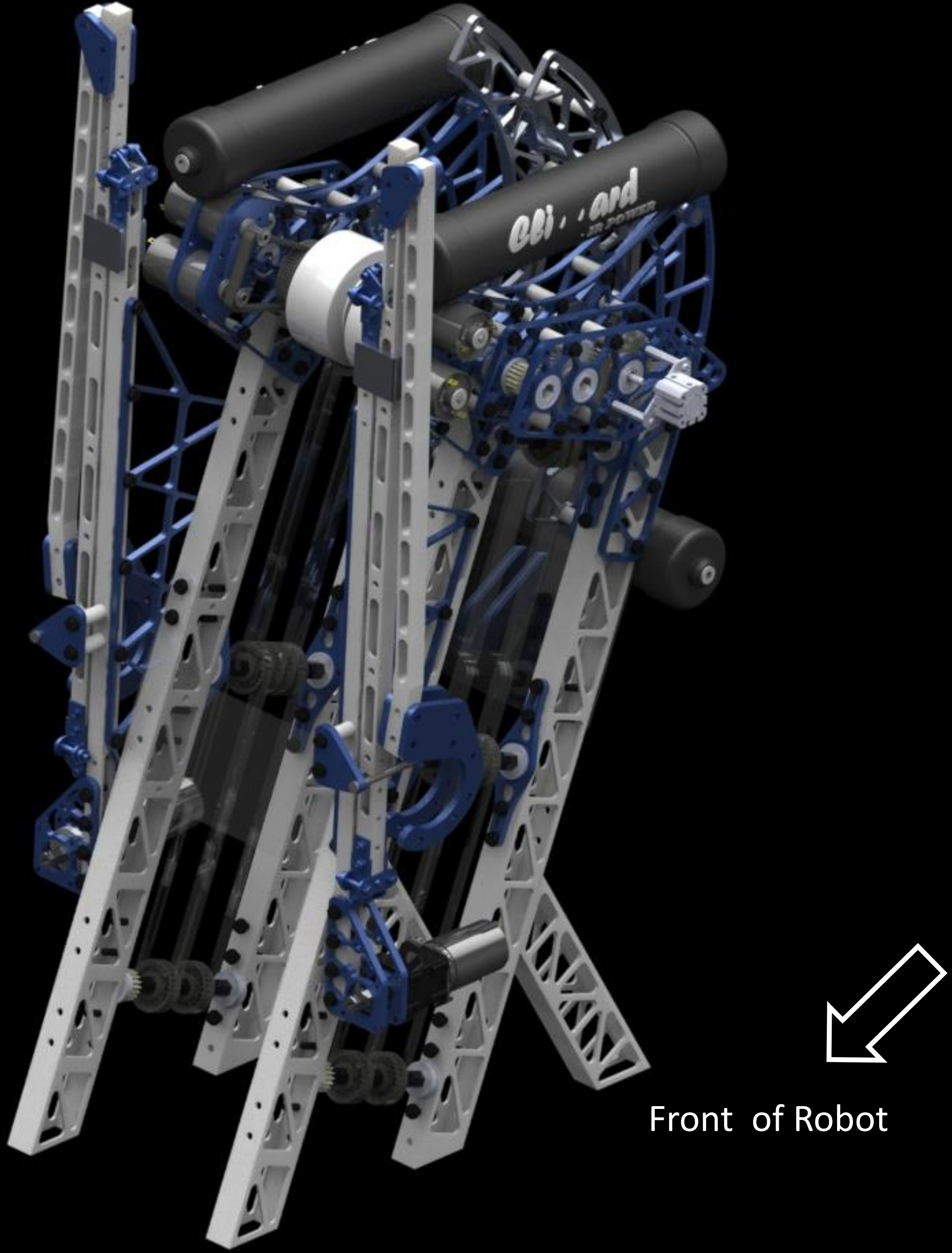
CLIMBER

- Dual deploying grabbing hooks help secure the rope.

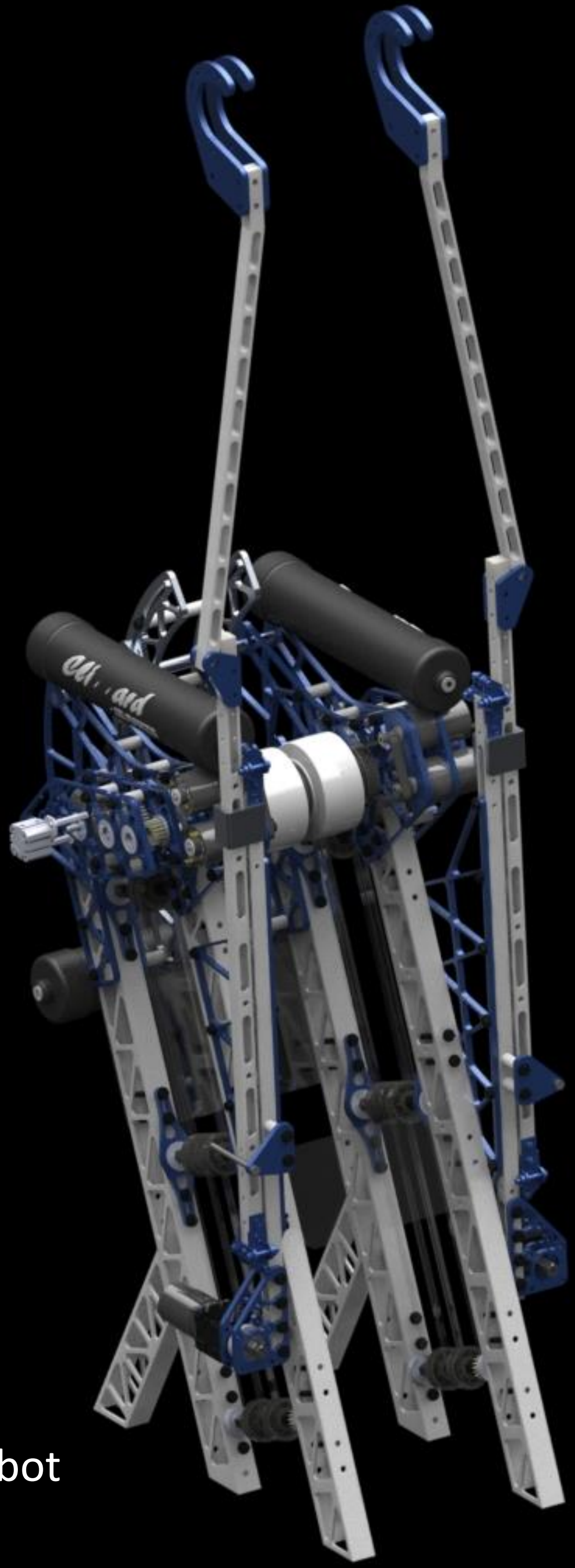
- Powered by the same output shaft as the shooter, with an additional stage to achieve an overall reduction of 45:1.

- Two parachord ropes attached to the deployed hooks, roll up two pulleys in order to achieve lift.

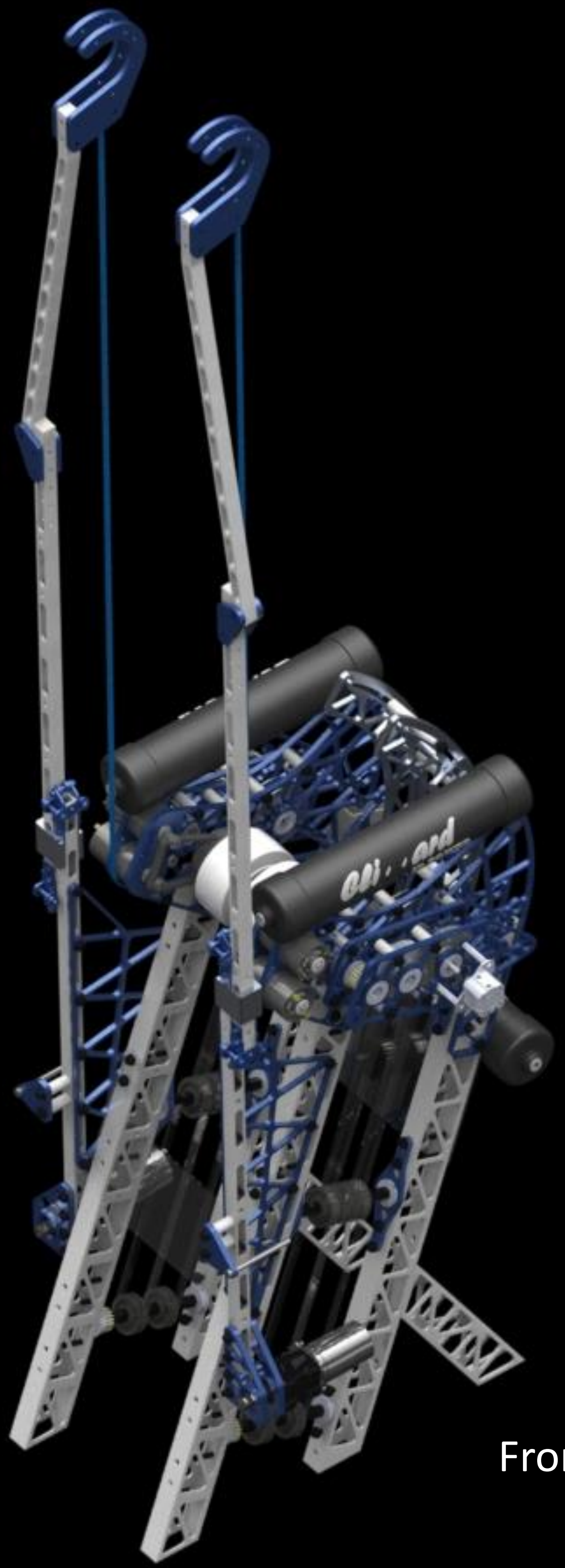
- Separate hooks elevated individually by lifts allow to reach the top and the bottom of an unbalanced Generator Switch in order to attain balance.



Front of Robot

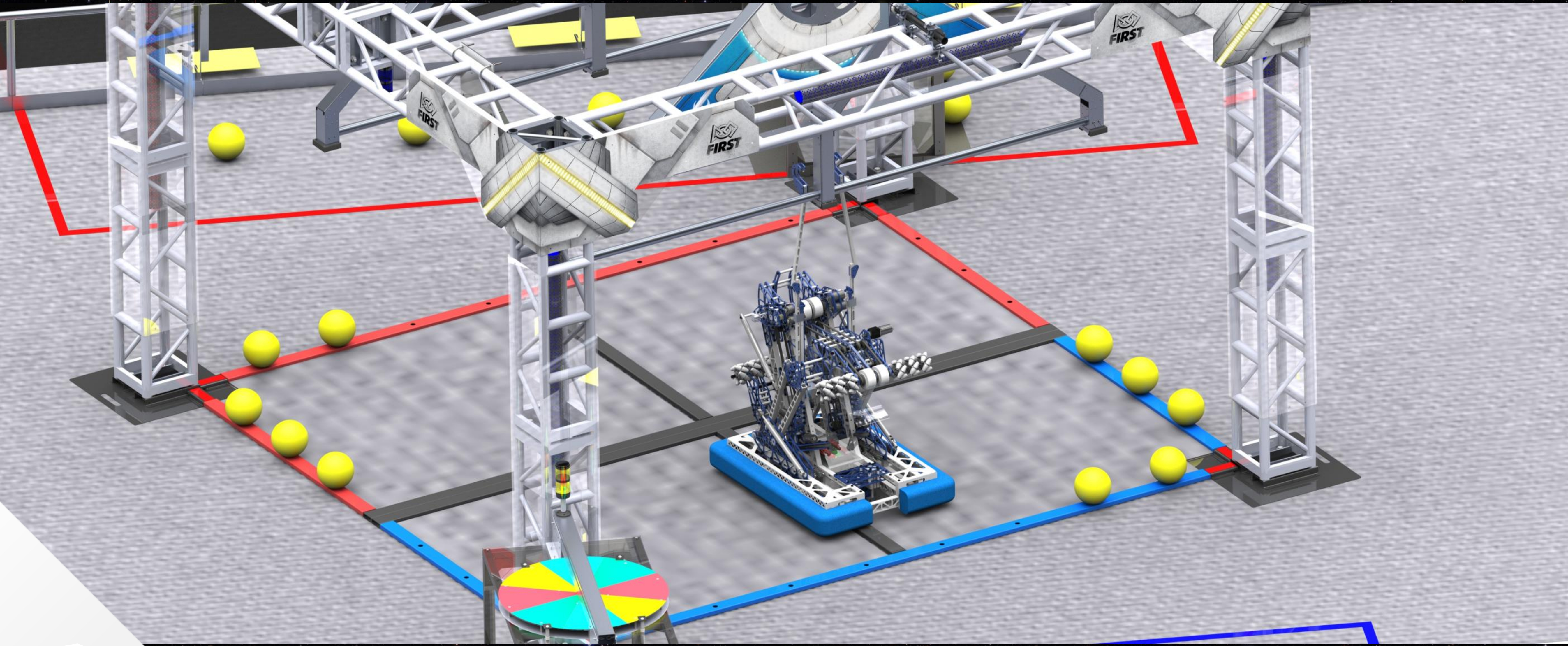


Front of Robot

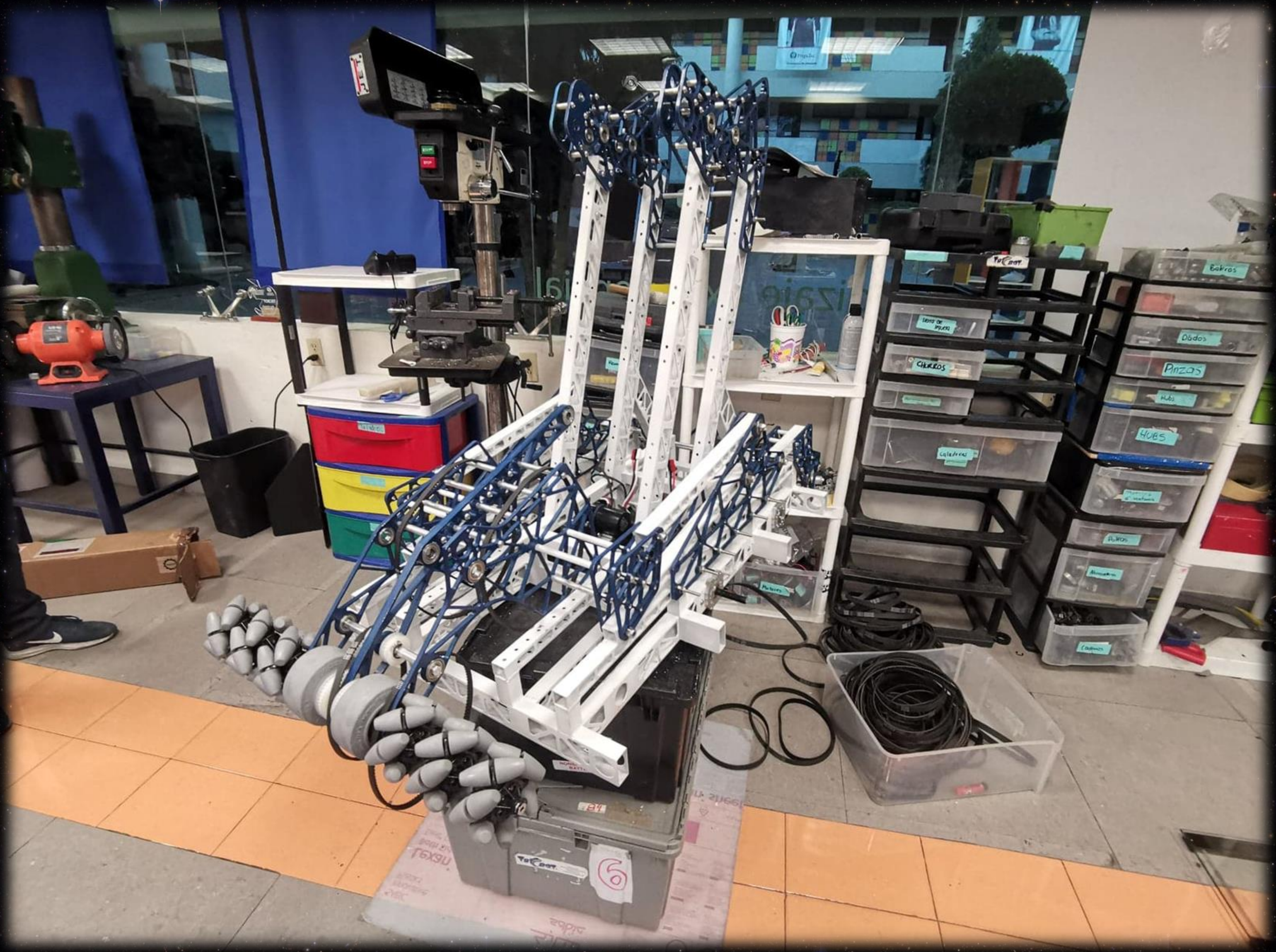


Front of Robot

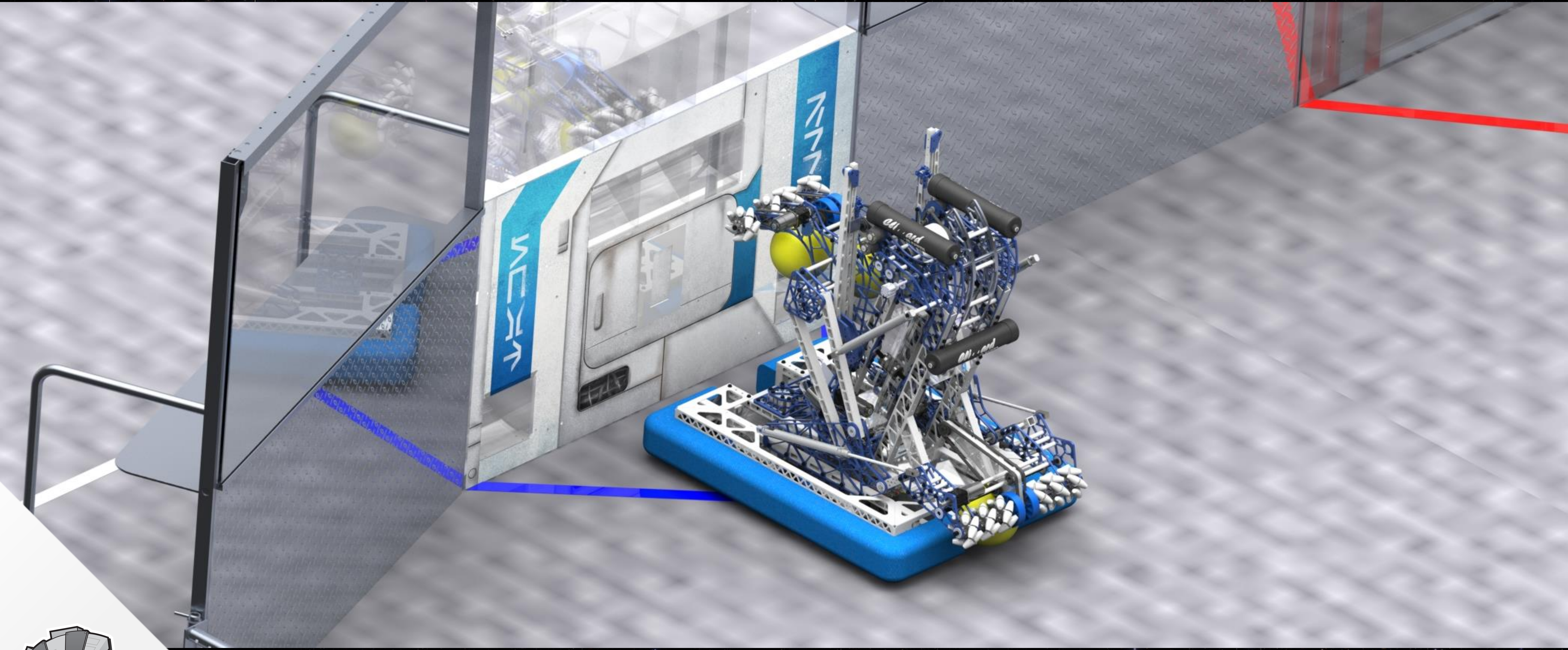
DESIGNED FOR MOVILITY



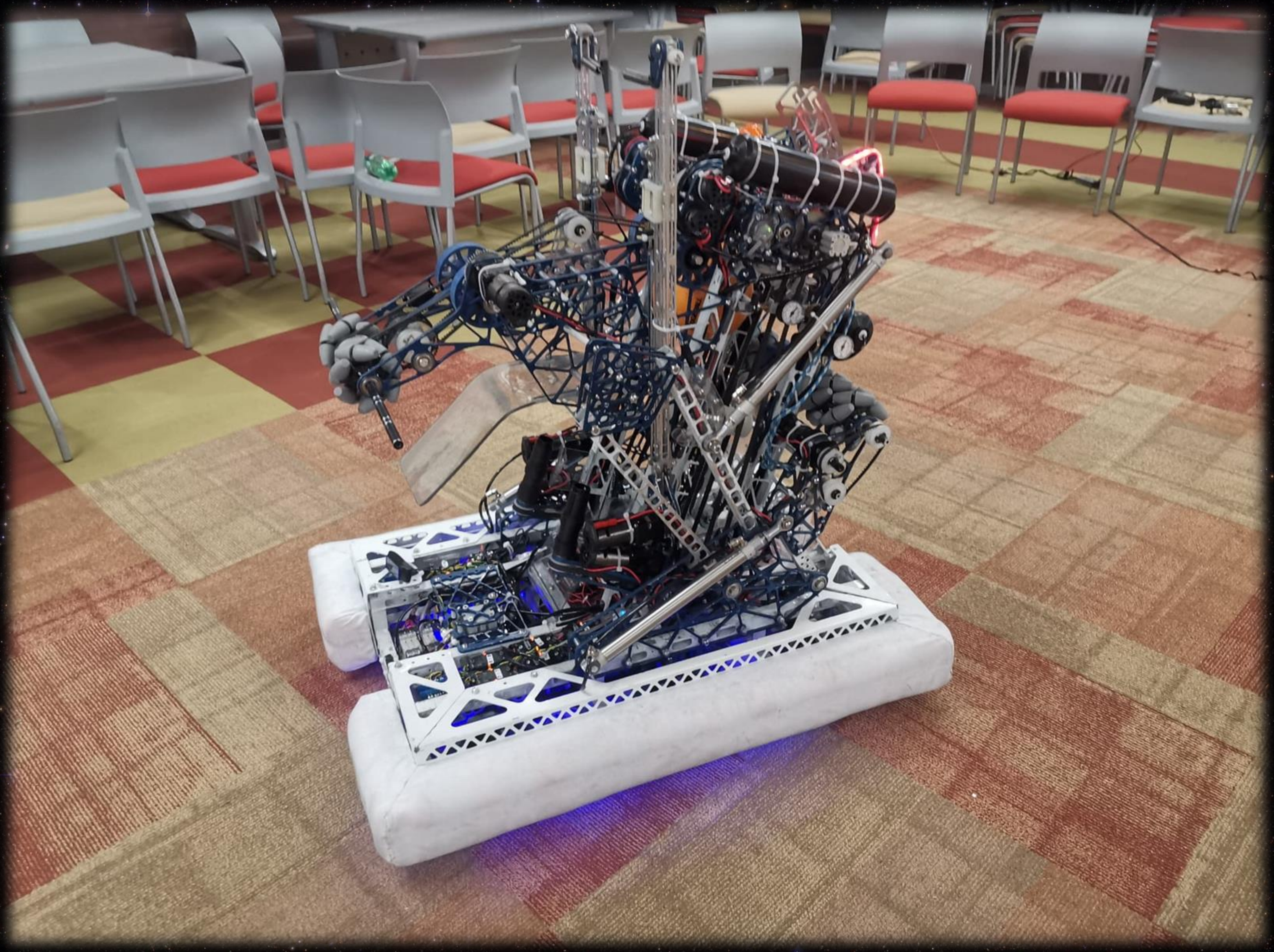
ENGINEERED TO BE LIGHTWEIGHT



DESIGNED TO BE RELIABLE



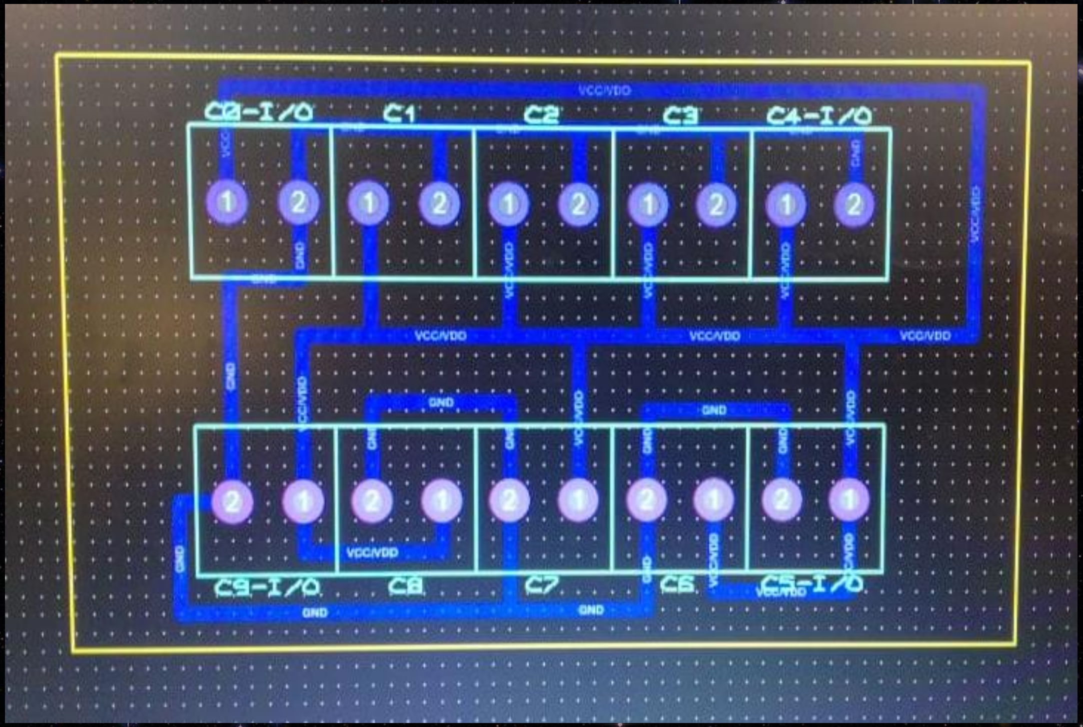
ENGINEERED TO BE PRECISE



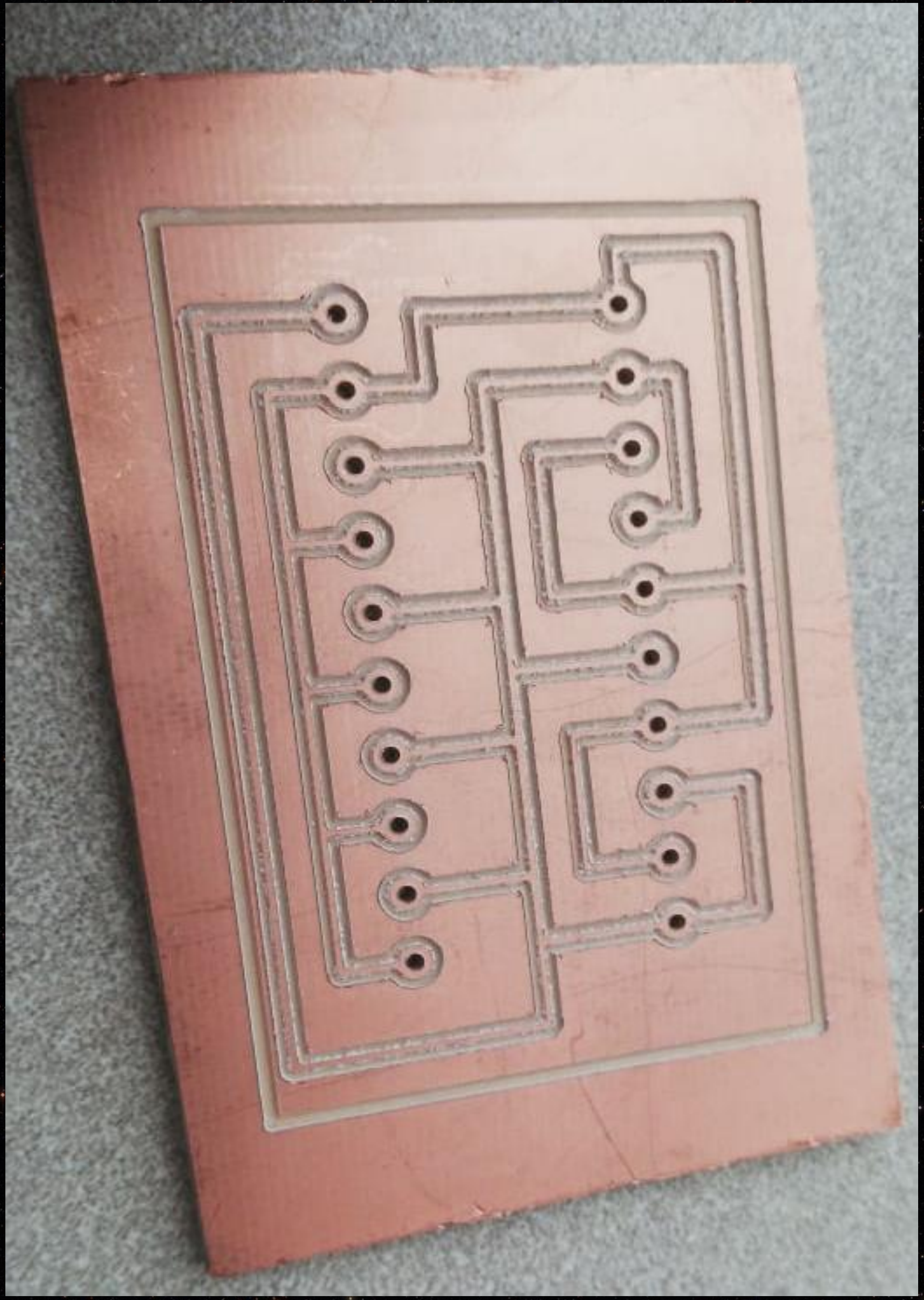
ELECTRICAL HIGHLIGHTS

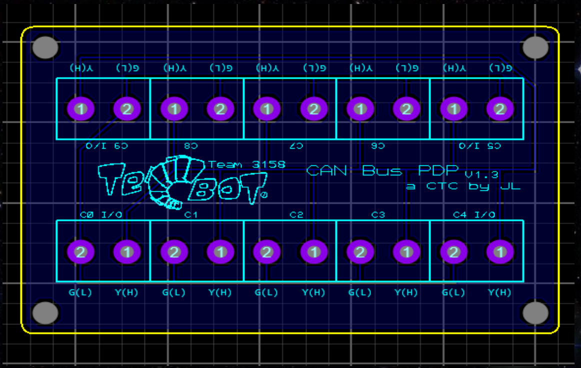
- For the past few seasons, we´ve been experiencing troubles with the CAN network, due to the topology of the circuit, when you would connect more than 8 motors, the last ones would not receive the CAN signal and they would not be visible in the RoboRio Network.
- The commonly used "Daisy Chain" system is due to present this sort of problems so we tried to connect the system with a CAN BUS module, this system helps us better distribute the signal among the motor controllers, with the clear advantage that in the unexpected case of a failure in any of the components connected to the system, the rest of the components will not be affected by the failure.
- We trusted our choice with this new component, and after weeks of design and development, we were able to design our own CAN BUS board, and we will be distributing them among other teams to help solve this sort of problems.

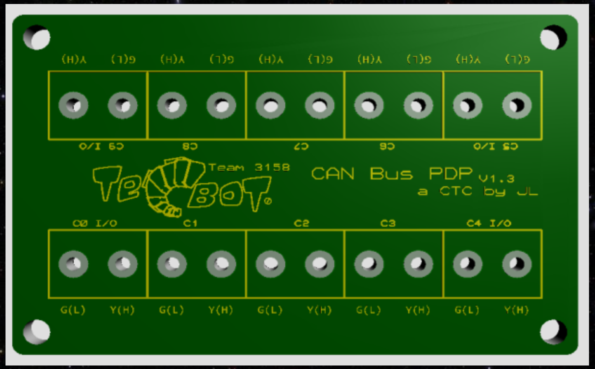
CAN BUS board (design)

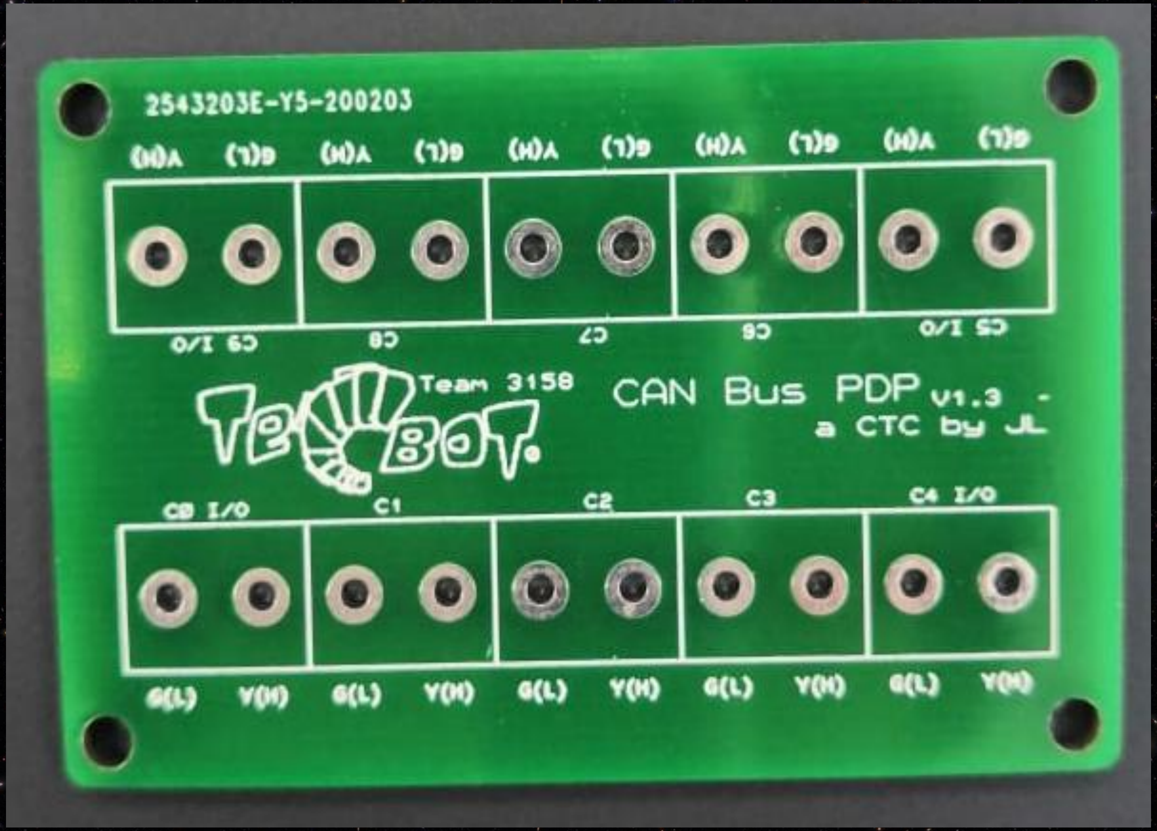


CAN BUS board (manufacturing)







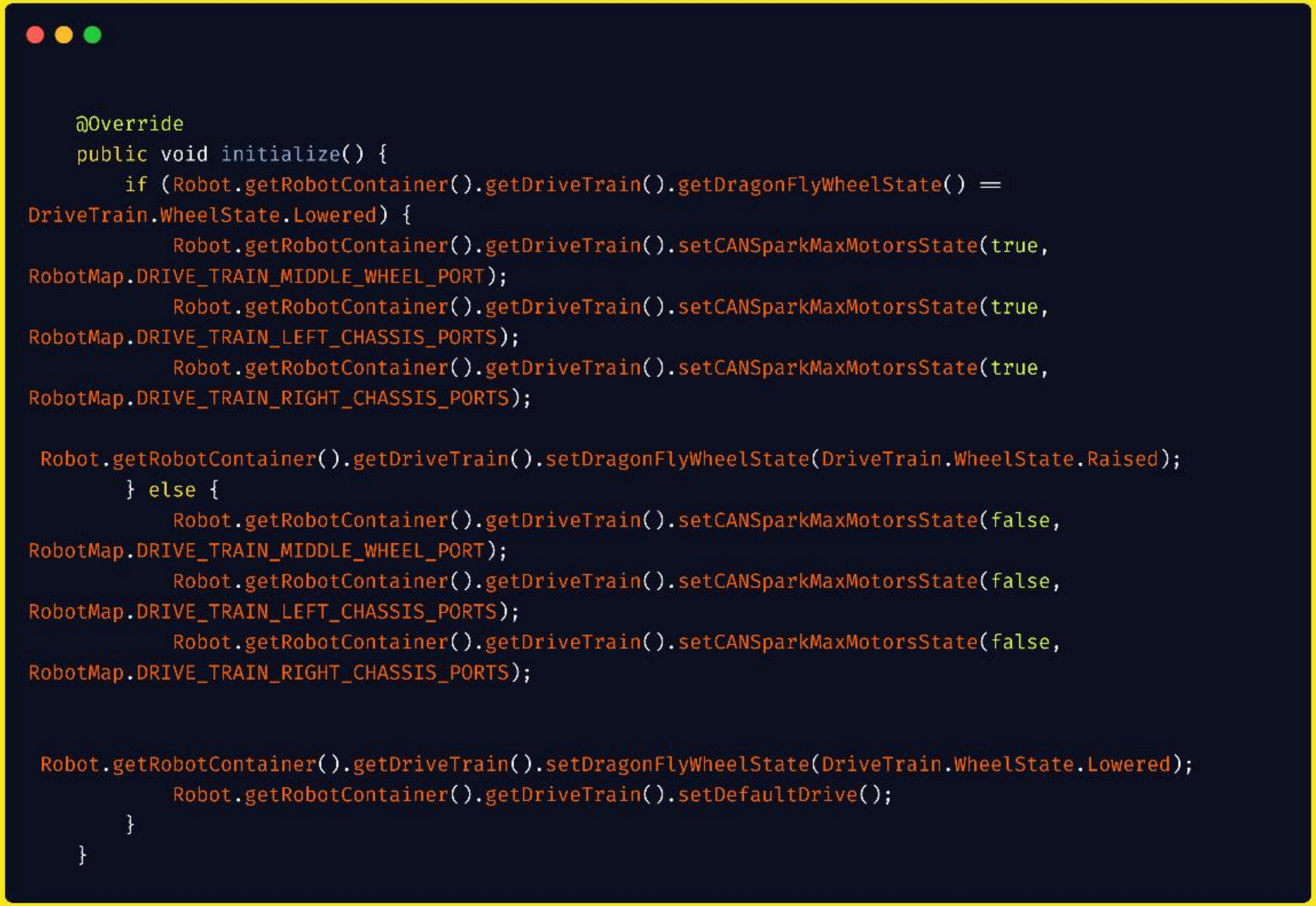


CAN BUS board (physical part)

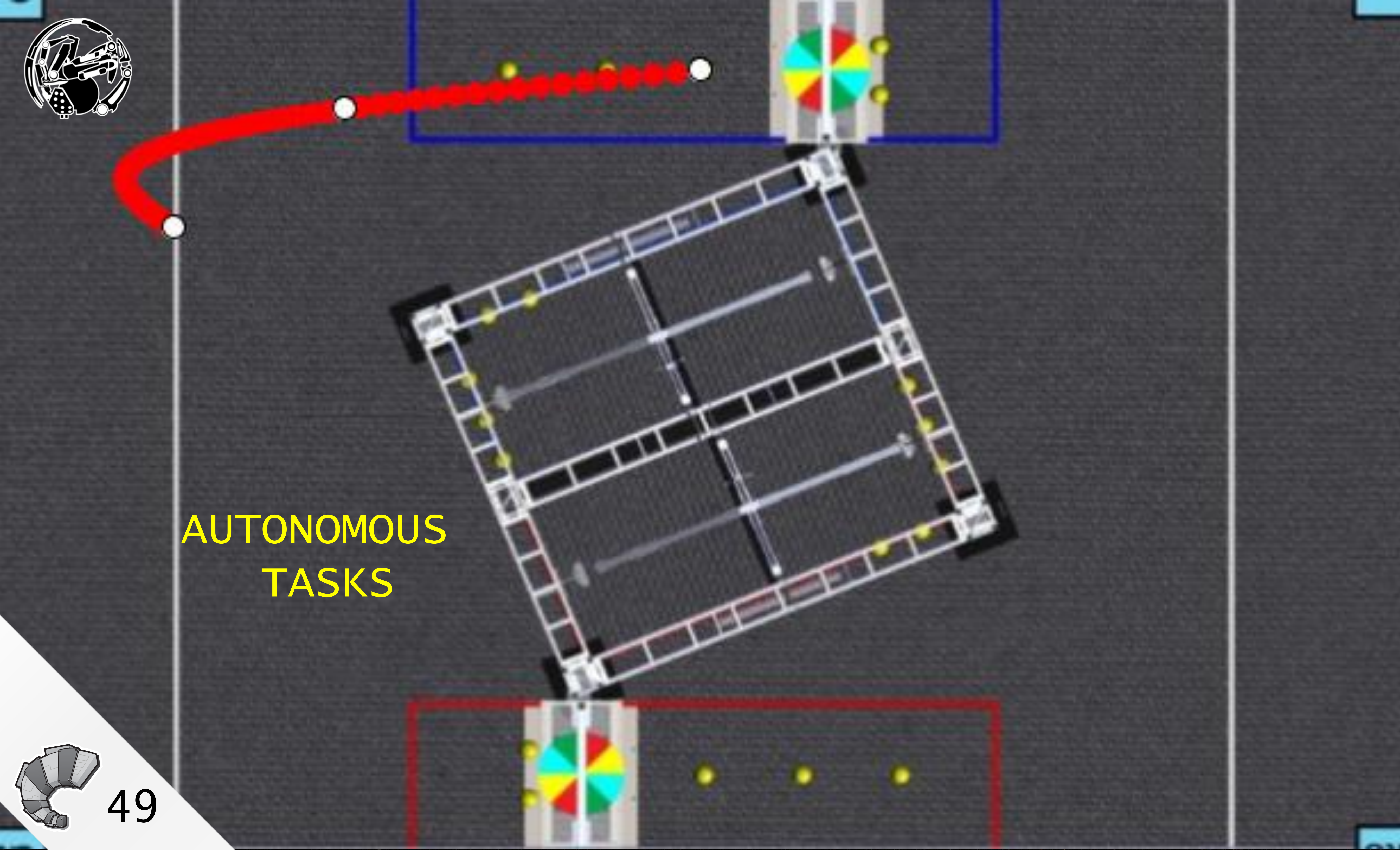
PROGRAMMING HIGHLIGHTS

- For this season, we implemented new tactics to improve our developers' performance, added new features to our autonomous and implemented our new dragon fly drivetrain. Over the years, we've improved the way we code by creating TecbotResources, a set of classes and general improvements to inputs and outputs of the robot such as speed controllers, encoders, sensors, Driver Station input (TecbotController.java), solenoids and lots of other things. We've continued to use the command-based programming for our robot, due to its level of organization. This season, developers made code reviews for each subsystem they did and used "Github Issues" to keep track of bugs as well as performance in each developer. Another implementation that we did for this season was to change the way we code by developing a plugin which allows you to create robot projects from IntelliJIdea, one of the best Developing Environments for Java

ROUTINE CODE

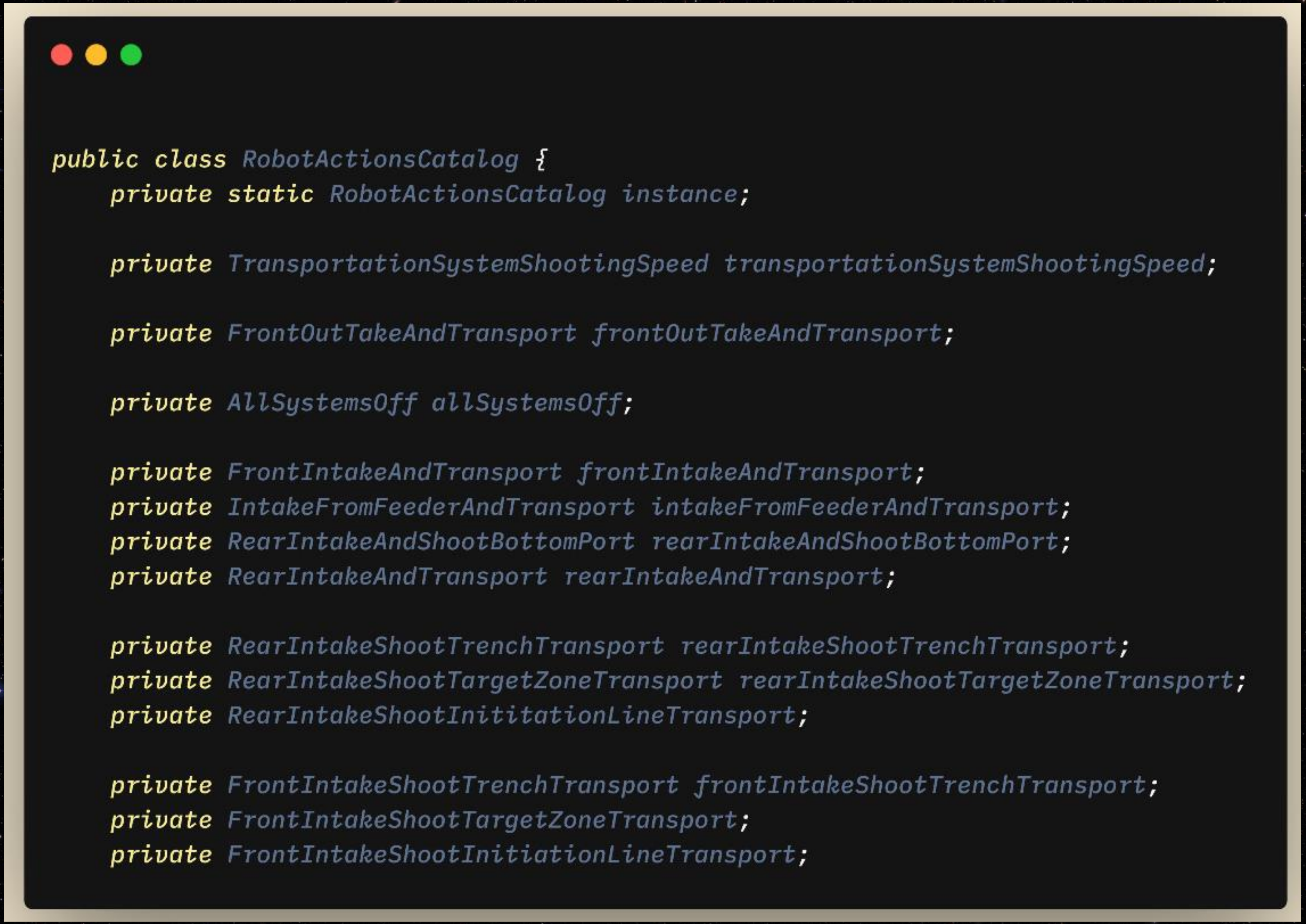


```
    @Override
    public void initialize() {
        if (Robot.getRobotContainer().getDriveTrain().getDragonFlyWheelState() ==
DriveTrain.WheelState.Lowered) {
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(true,
RobotMap.DRIVE_TRAIN_MIDDLE_WHEEL_PORT);
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(true,
RobotMap.DRIVE_TRAIN_LEFT_CHASSIS_PORTS);
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(true,
RobotMap.DRIVE_TRAIN_RIGHT_CHASSIS_PORTS);

 Robot.getRobotContainer().getDriveTrain().setDragonFlyWheelState(DriveTrain.WheelState.Raised);
        } else {
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(false,
RobotMap.DRIVE_TRAIN_MIDDLE_WHEEL_PORT);
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(false,
RobotMap.DRIVE_TRAIN_LEFT_CHASSIS_PORTS);
            Robot.getRobotContainer().getDriveTrain().setCANSparkMaxMotorsState(false,
RobotMap.DRIVE_TRAIN_RIGHT_CHASSIS_PORTS);


 Robot.getRobotContainer().getDriveTrain().setDragonFlyWheelState(DriveTrain.WheelState.Lowered);
            Robot.getRobotContainer().getDriveTrain().setDefaultDrive();
        }
    }
```



AUTONOMOUS TASKS

AUTONOMOUS CODE

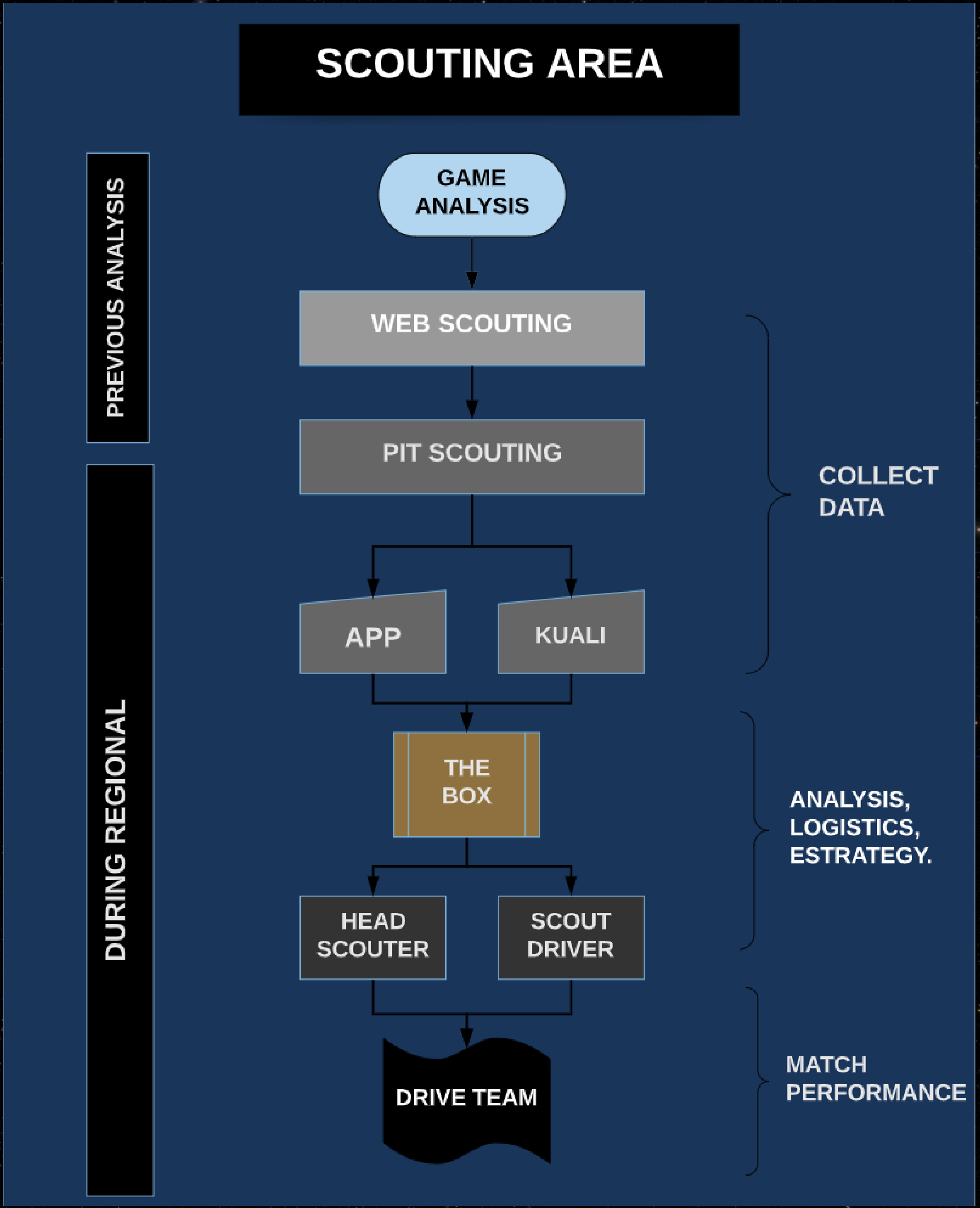


```java
public class RobotActionsCatalog {
    private static RobotActionsCatalog instance;

    private TransportationSystemShootingSpeed transportationSystemShootingSpeed;

    private FrontOutTakeAndTransport frontOutTakeAndTransport;

    private AllSystemsOff allSystemsOff;

    private FrontIntakeAndTransport frontIntakeAndTransport;
    private IntakeFromFeederAndTransport intakeFromFeederAndTransport;
    private RearIntakeAndShootBottomPort rearIntakeAndShootBottomPort;
    private RearIntakeAndTransport rearIntakeAndTransport;

    private RearIntakeShootTrenchTransport rearIntakeShootTrenchTransport;
    private RearIntakeShootTargetZoneTransport rearIntakeShootTargetZoneTransport;
    private RearIntakeShootInititationLineTransport;

    private FrontIntakeShootTrenchTransport frontIntakeShootTrenchTransport;
    private FrontIntakeShootTargetZoneTransport;
    private FrontIntakeShootInitiationLineTransport;
```

SCOUTING HIGHLIGHTS

- The last season, we implemented an online scouting record system, it served to improve the way we keep track of other team’s performance in order to better design the strategy for the upcoming matches.
- This season, we improved the tracking system and the server, so that it may serve us better, at the same time, we’re sharing the app with other teams, in order to help them improve their overall game.

SCOUTING WORKFLOW



SCOUTING AREA

PREVIOUS ANALYSIS

DURING REGIONAL

GAME ANALYSIS

WEB SCOUTING

PIT SCOUTING

APP

KUALI

THE BOX

HEAD SCOUTER

SCOUT DRIVER

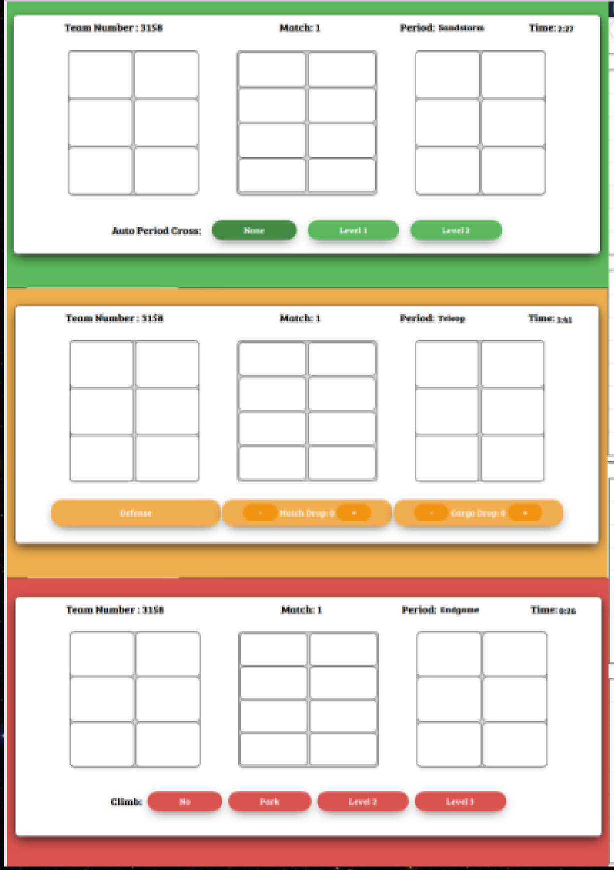
DRIVE TEAM

COLLECT DATA

ANALYSIS, LOGISTICS, ESTRATEGY.

MATCH PERFORMANCE

SCOUTING APPLICATION



Team Number : 3158 Match: 1 Period: Sandstorm Time: 2:27

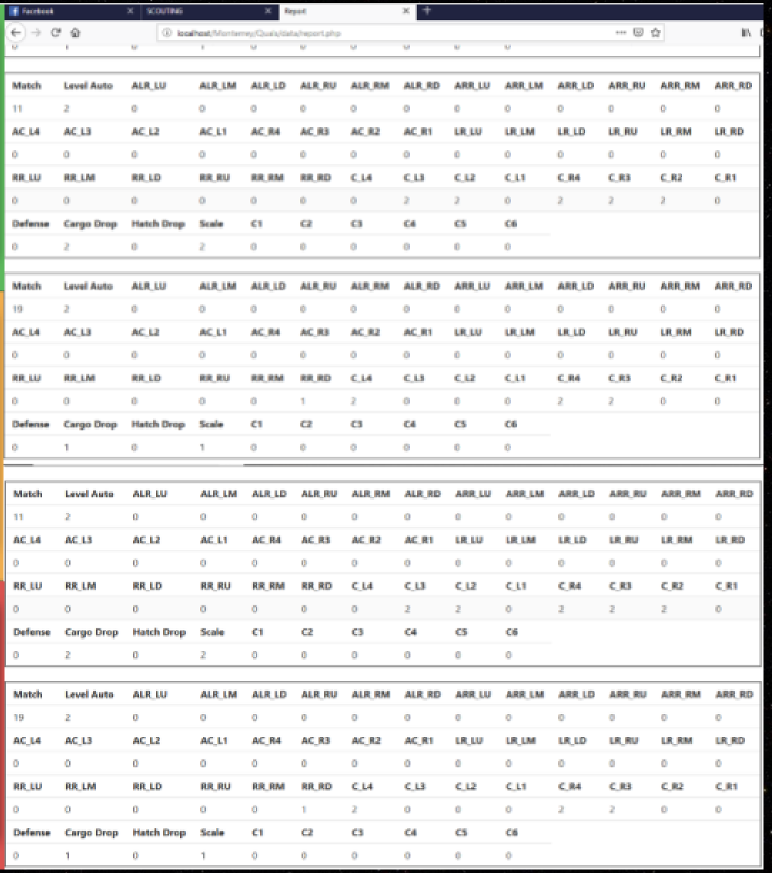
Auto Period Cross: None Level 1 Level 2

Team Number : 3158 Match: 1 Period: Teleop Time: 1:41

Defense - Hatch Drop: 0 + - Cargo Drop: 0 +

Team Number : 3158 Match: 1 Period: Endgame Time: 0:26

Climb: No Park Level 2 Level 3



Match	Level Auto	ALR_LU	ALR_LM	ALR_LD	ALR_RU	ALR_RM	ALR_RD	ARR_LU	ARR_LM	ARR_LD	ARR_RU	ARR_RM	ARR_RD
11	2	0	0	0	0	0	0	0	0	0	0	0	0
AC_L4	AC_L3	AC_L2	AC_L1	AC_R4	AC_R3	AC_R2	AC_R1	LR_LU	LR_LM	LR_LD	LR_RU	LR_RM	LR_RD
0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR_LU	RR_LM	RR_LD	RR_RU	RR_RM	RR_RD	C_L4	C_L3	C_L2	C_L1	C_R4	C_R3	C_R2	C_R1
0	0	0	0	0	0	0	2	2	0	2	2	2	0
Defense	Cargo Drop	Hatch Drop	Scale	C1	C2	C3	C4	C5	C6				
0	2	0	2	0	0	0	0	0	0				

Match	Level Auto	ALR_LU	ALR_LM	ALR_LD	ALR_RU	ALR_RM	ALR_RD	ARR_LU	ARR_LM	ARR_LD	ARR_RU	ARR_RM	ARR_RD
19	2	0	0	0	0	0	0	0	0	0	0	0	0
AC_L4	AC_L3	AC_L2	AC_L1	AC_R4	AC_R3	AC_R2	AC_R1	LR_LU	LR_LM	LR_LD	LR_RU	LR_RM	LR_RD
0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR_LU	RR_LM	RR_LD	RR_RU	RR_RM	RR_RD	C_L4	C_L3	C_L2	C_L1	C_R4	C_R3	C_R2	C_R1
0	0	0	0	0	1	2	0	0	0	2	2	0	0
Defense	Cargo Drop	Hatch Drop	Scale	C1	C2	C3	C4	C5	C6				
0	1	0	1	0	0	0	0	0	0				

Match	Level Auto	ALR_LU	ALR_LM	ALR_LD	ALR_RU	ALR_RM	ALR_RD	ARR_LU	ARR_LM	ARR_LD	ARR_RU	ARR_RM	ARR_RD
11	2	0	0	0	0	0	0	0	0	0	0	0	0
AC_L4	AC_L3	AC_L2	AC_L1	AC_R4	AC_R3	AC_R2	AC_R1	LR_LU	LR_LM	LR_LD	LR_RU	LR_RM	LR_RD
0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR_LU	RR_LM	RR_LD	RR_RU	RR_RM	RR_RD	C_L4	C_L3	C_L2	C_L1	C_R4	C_R3	C_R2	C_R1
0	0	0	0	0	0	0	2	2	0	2	2	2	0
Defense	Cargo Drop	Hatch Drop	Scale	C1	C2	C3	C4	C5	C6				
0	2	0	2	0	0	0	0	0	0				

Match	Level Auto	ALR_LU	ALR_LM	ALR_LD	ALR_RU	ALR_RM	ALR_RD	ARR_LU	ARR_LM	ARR_LD	ARR_RU	ARR_RM	ARR_RD
19	2	0	0	0	0	0	0	0	0	0	0	0	0
AC_L4	AC_L3	AC_L2	AC_L1	AC_R4	AC_R3	AC_R2	AC_R1	LR_LU	LR_LM	LR_LD	LR_RU	LR_RM	LR_RD
0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR_LU	RR_LM	RR_LD	RR_RU	RR_RM	RR_RD	C_L4	C_L3	C_L2	C_L1	C_R4	C_R3	C_R2	C_R1
0	0	0	0	0	1	2	0	0	0	2	2	0	0
Defense	Cargo Drop	Hatch Drop	Scale	C1	C2	C3	C4	C5	C6				
0	1	0	1	0	0	0	0	0	0				

THANKS TO OUR SPONSORS!

GENERAL MOTORS

Repostería Casera
PANADELY

MUNDO MAGYC

ASGROW®

GO PRINT
soluciones en impresió

atg STORE
IMPRIME EMPACA ENVIA Y MAS

CAMOZZI
Automation

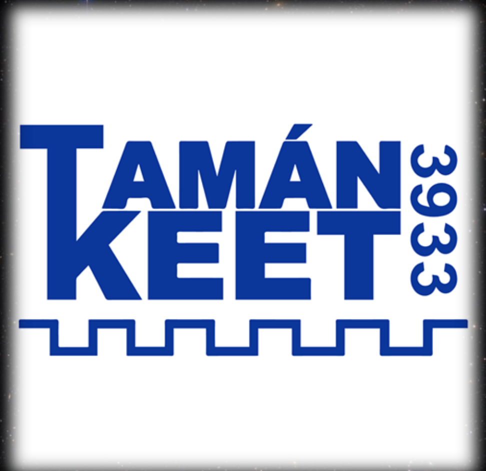
fundación azteca
de Grupo Salinas

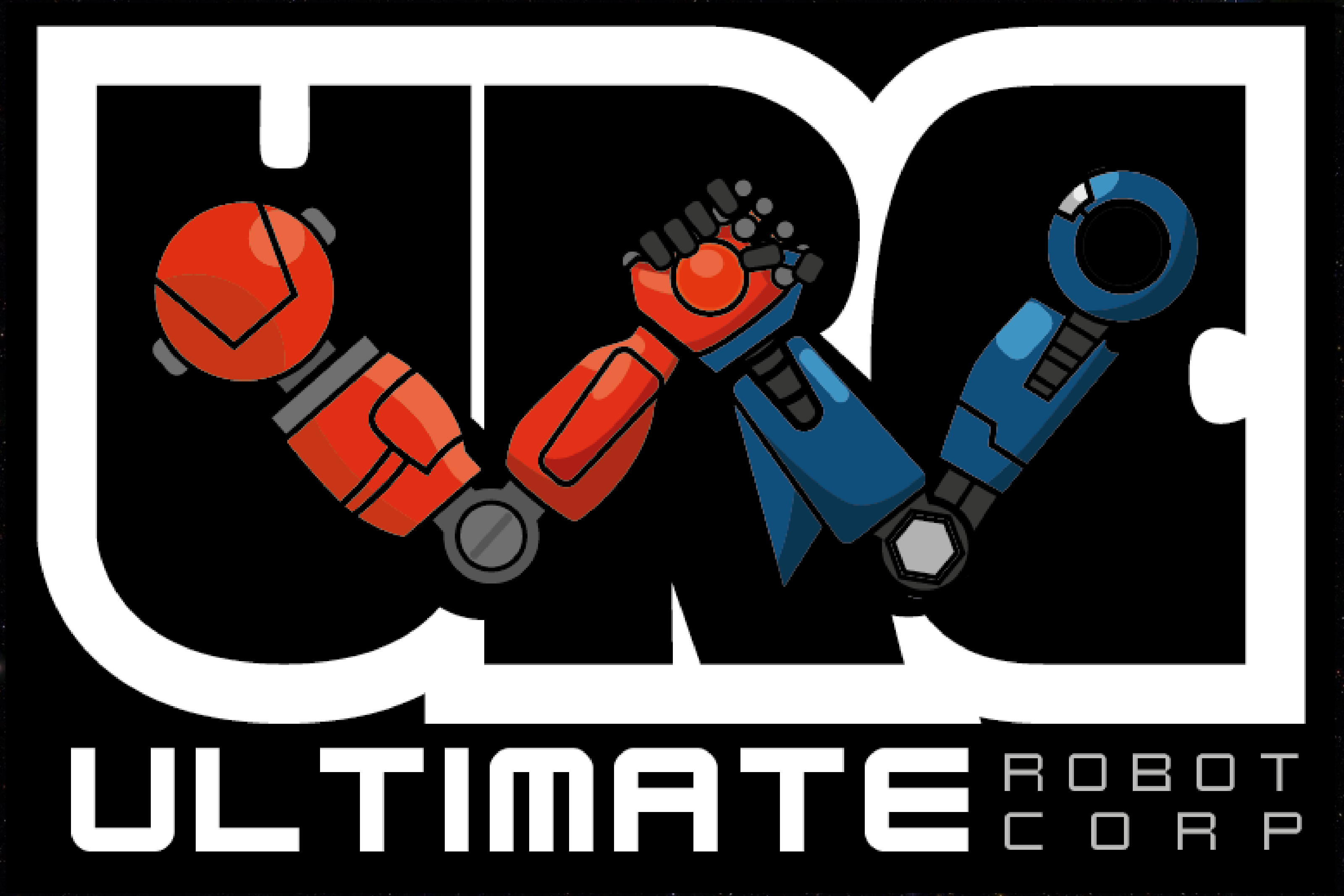
PrepaTec

THANKS TO OUR MENTORS!

THANKS TO OUR MEMBERS!

AND THE MOST SPECIAL THANKS TO …





FROM THE BOTTOM OF OUR HEARTS, THANK YOU FRIENDS,
THIS DREAM WAS POSSIBLE BECAUSE OF YOU