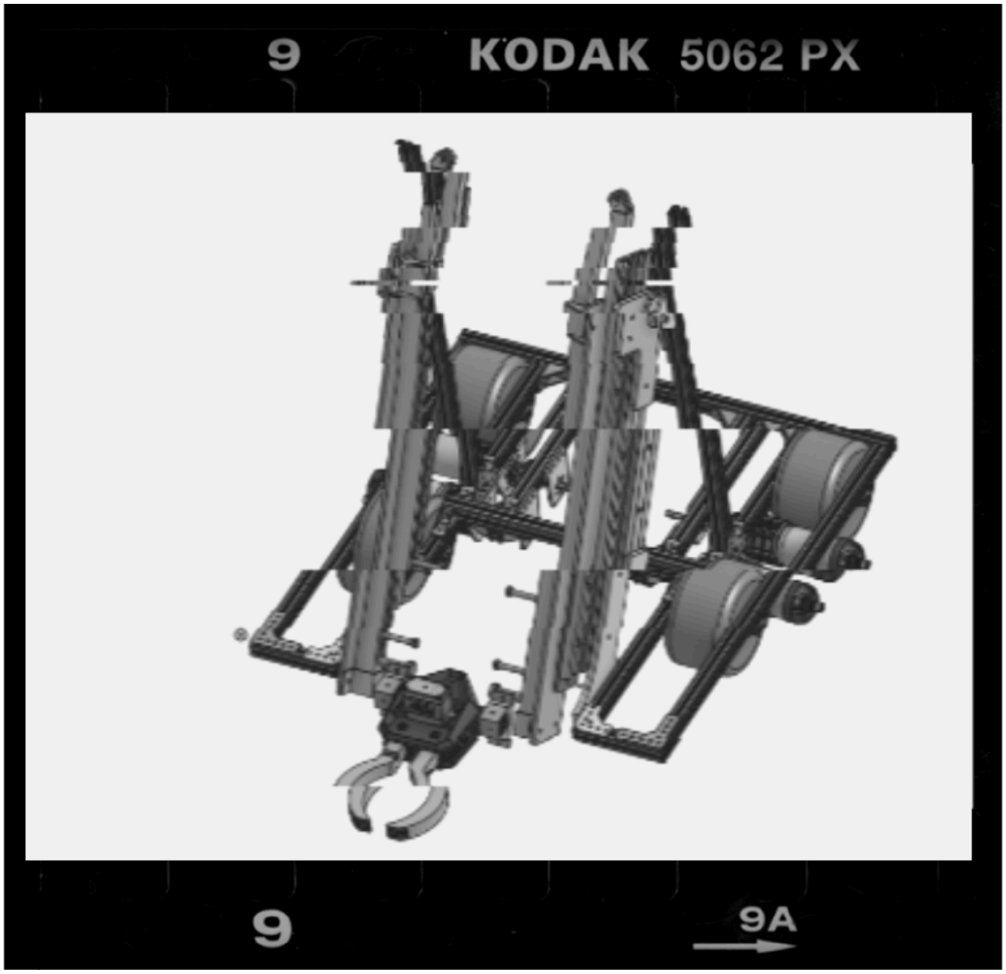
Engineering Notebook



Vegamind 22903

About us

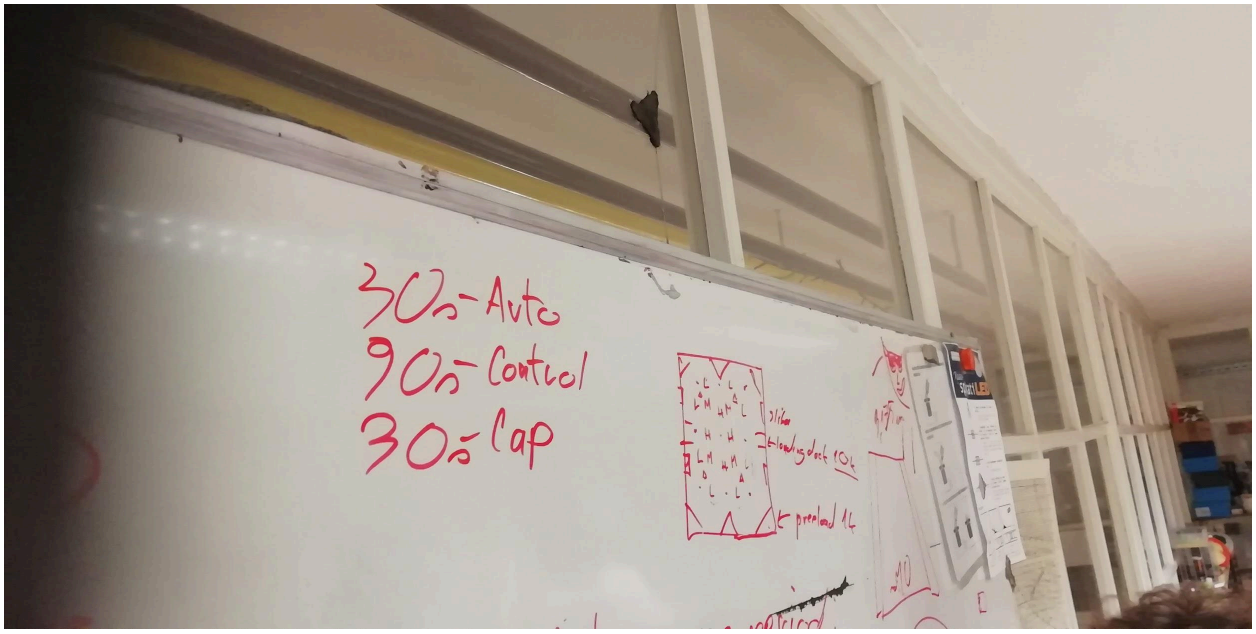
We–*Vegamind*–are a small team of seven, located in the heart of Slovenia, Ljubljana. Two mentors, Luka and David; who have enough expertise for all of us together, and five students, Ana, Jaka, Jurij, Ema, and David, with dreams as vast as the sky.
This is the first time we are competing in the *First Tech Challenge*, however, we have been to *First Global*-s a few times before–with a different team, of course. The team changes from year to year, as our oldest leave for their futures, and as the younger generations come out of their shells. We pride ourselves in our openness and despise negativity. We adore learning new things that are a part of our school curriculum, and are *always* fascinated by how interconnected everything really is.
While the start was slow, we are now a part of two organizations: *Vegova Ljubljana* (Upper Secondary School of Electrical and Computer Engineering and Technical Gymnasium Ljubljana) and *Zavod 404*. They help us gain the resources for anything we might need, from accommodation in foreign countries, to gathering different resources.

Thank you, we are excited to be a part of some bigger picture,

Team *Vegamind*

Engineering Process

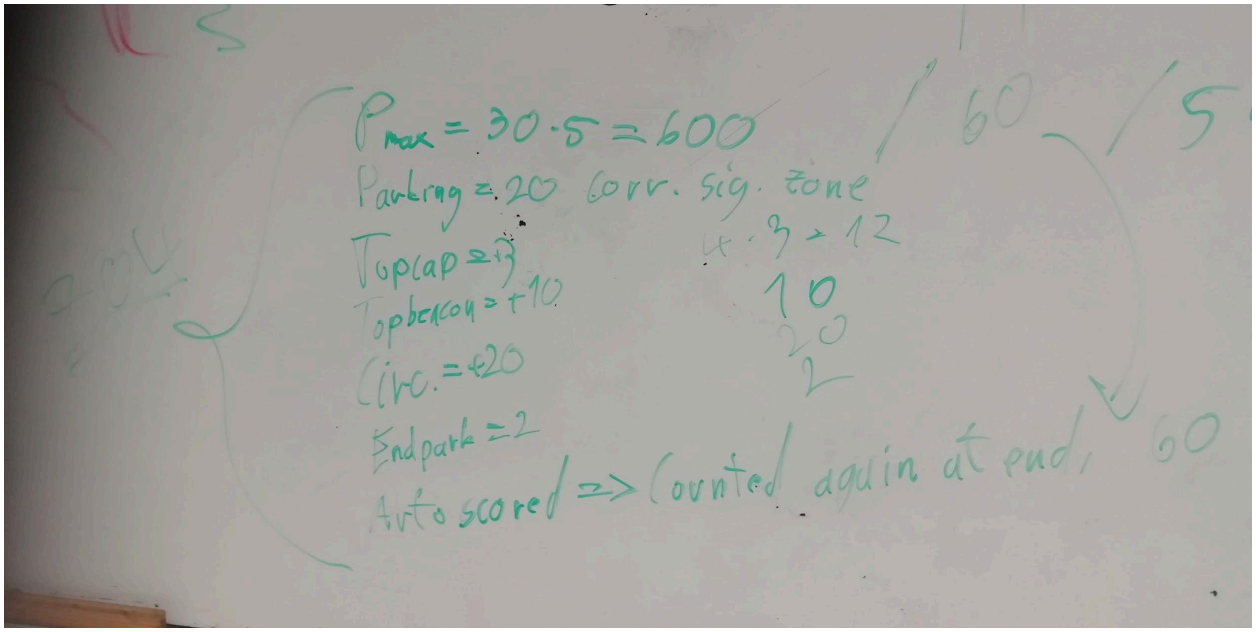
Game analysis



Picture: Time frames

Scoring

- Corner = 1
- Ground junction = 2
- Low junction = 3
- Medium junction = 4
- High Junction = 5
- Pmax = 150 + (Autonomous) 75
- Parking = 20
- Top Cap = 3
- Top Beacon = 10
- Sum points '=' 158



Picture: Trying to work out how many points is maximum

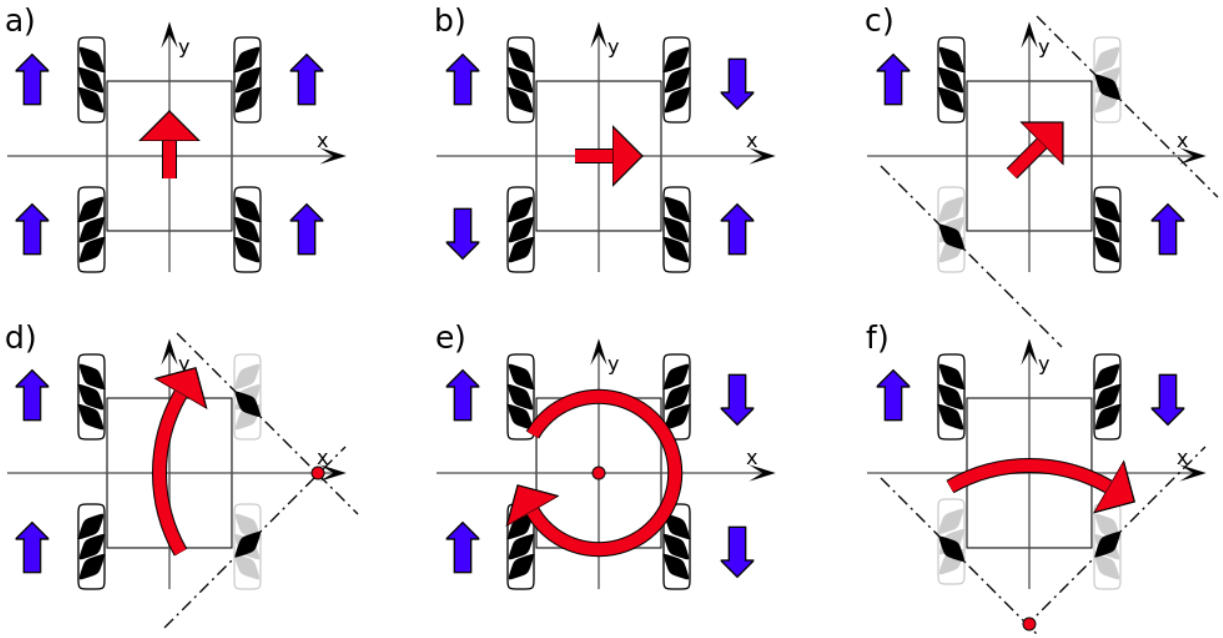
Robot design

Drive design

- Ball drive
- **Mecanum Wheels**
- Omni Wheels

Ball drive is essentially used for the sole purpose of moving a certain product into any direction it is desired. We decided that it is much too complicated to make this type of wheels work as we had no previous knowledge of their workings.
While ball drives are complicated to use, omni wheels (hereinafter OWs) and their “cousins”, mecanum wheels (hereinafter MWs) are a lot easier on the user. They are made for swiftness on any kind of stable ground, although when the field becomes uneven or the rollers wear out the functionality of the wheel lessens.
We decided on using MWs because with the OWs, there is the problem of the ‘X’ axis which requires another wheel for it to work. With MWs you need only four wheels and command them to do your bidding of where to go even if it is diagonally, forwards, backwards, or spinning around its center. With OWs you need four wheels and another one in the center of the robot for it to go sideways.

MWs in practice:



Picture: Mecanum wheels and their rotations

Robot movement is calculated by scaling wheel vectors whose components are mapped to corresponding wheels. Firstly, we read joystick input and calculate vector length. We will use the calculated length later as the power scalar. After that we take wheel vectors and scale them according to joystick input. Then we sum up 2 wheel vectors. We normalize the resulting vector and scale it up with the before calculated power scalar.

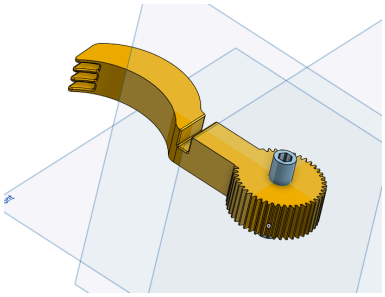
Gripper and lift

Gripper

Gripper Design:

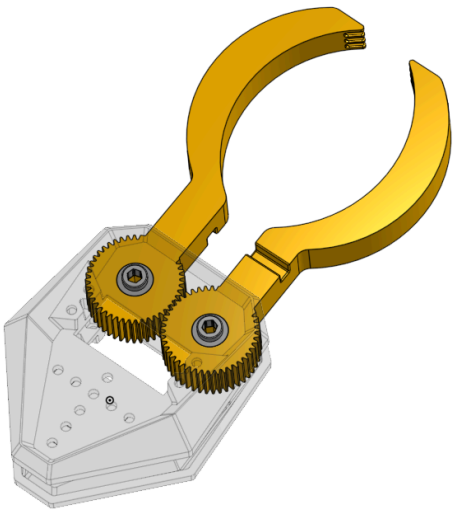
1. Classic REV with stiff jaws
2. Center-grip gripper
3. **Rubber Banded Jaws**
    a. First iteration

In the beginning we were trying to use just REV-41-1358 Claw assembly with some brackets to grasp the cone. While it may work we tried to design our own claw assembly. We still used the same mounting hardware and same design for the greater part of our claw as REV uses (mostly for simplicity). Our gripper head however supports rubber bands to squeeze the cone at the point of contact. First iteration of the gripper was sadly hard to 3D print because it had no large flat area that could be placed on the print bed. We also just guessed the size we'd need and it was too small. It can be seen on the figure below.

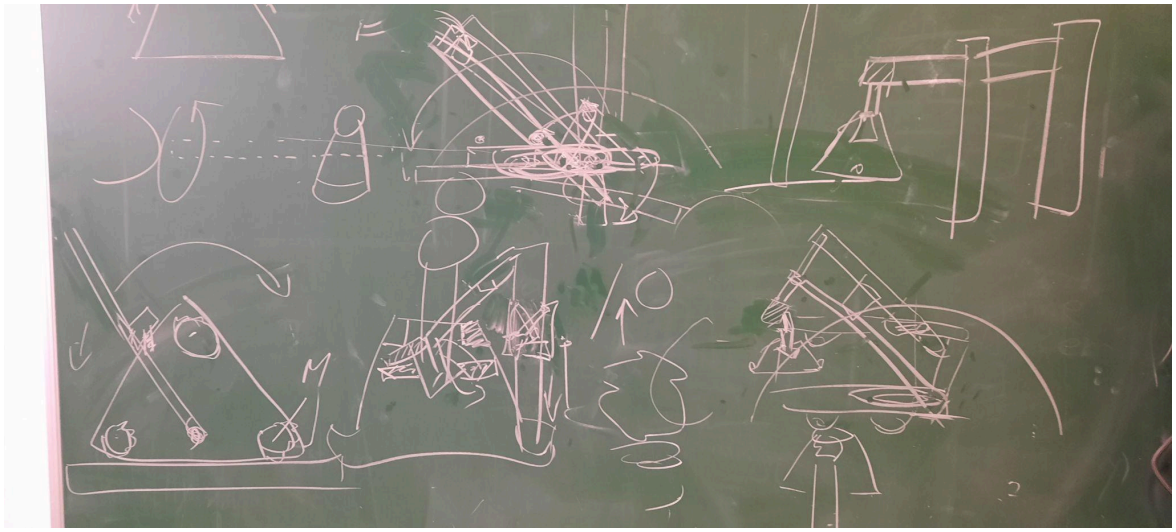


b. Second iteration

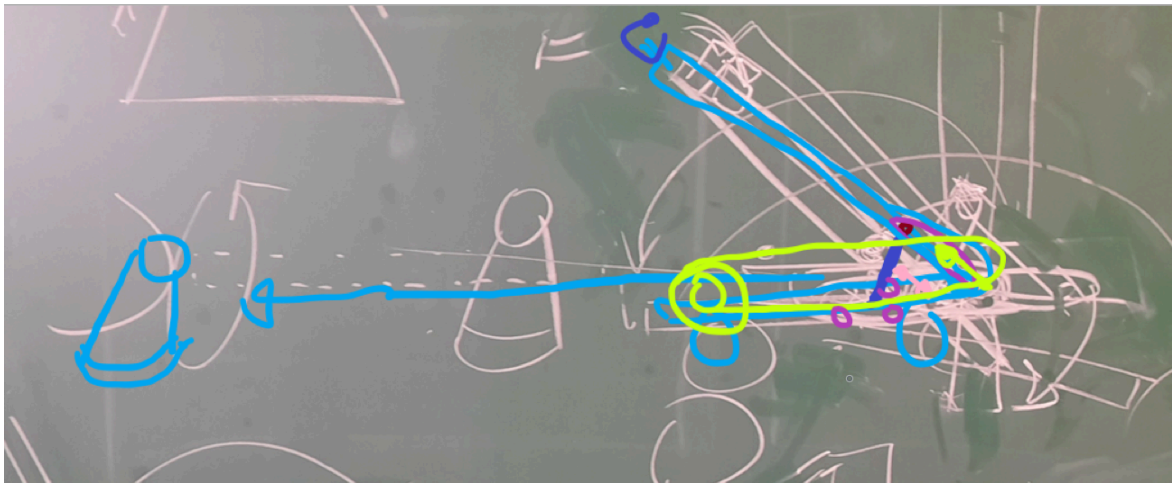
Second iteration fixes both of the problems. We made claw size adjustable and replaced long axis holders built in the gripper with Through Bore Bearings (REV-41-1326). In the figure below the design is better presented along with the transparent holders (REV-41-1358).

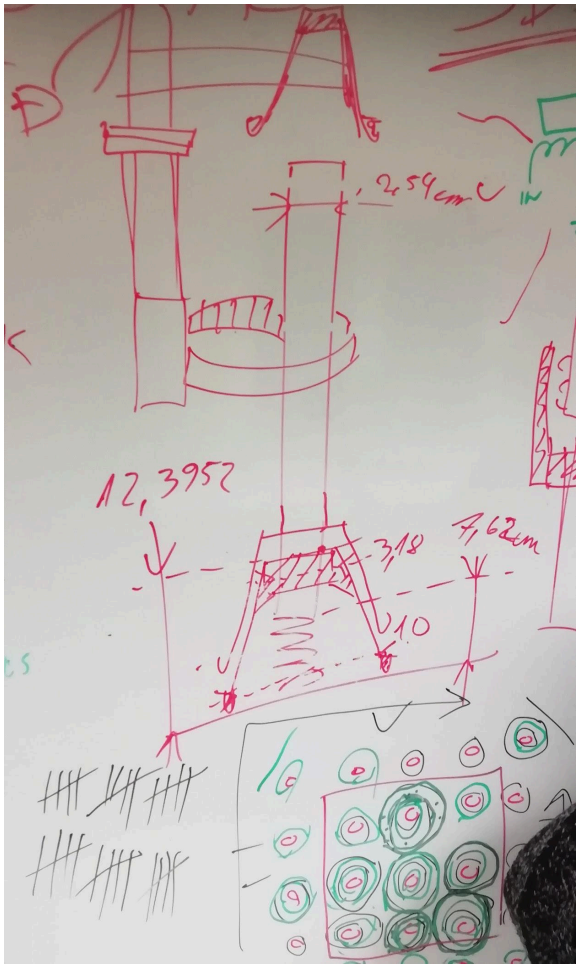


Gripper and arm design ideas:



More planning:

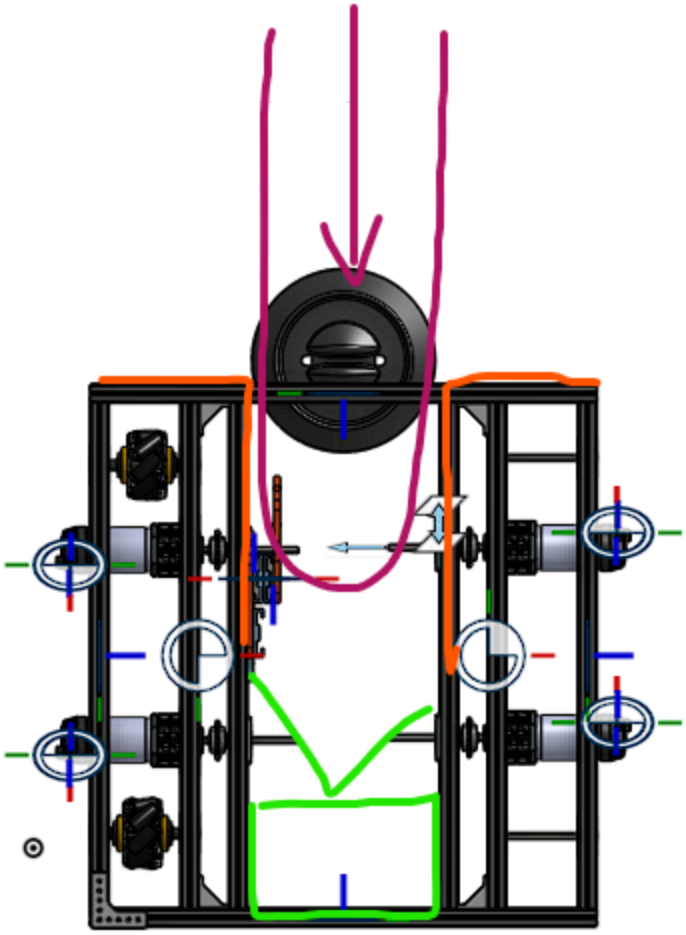




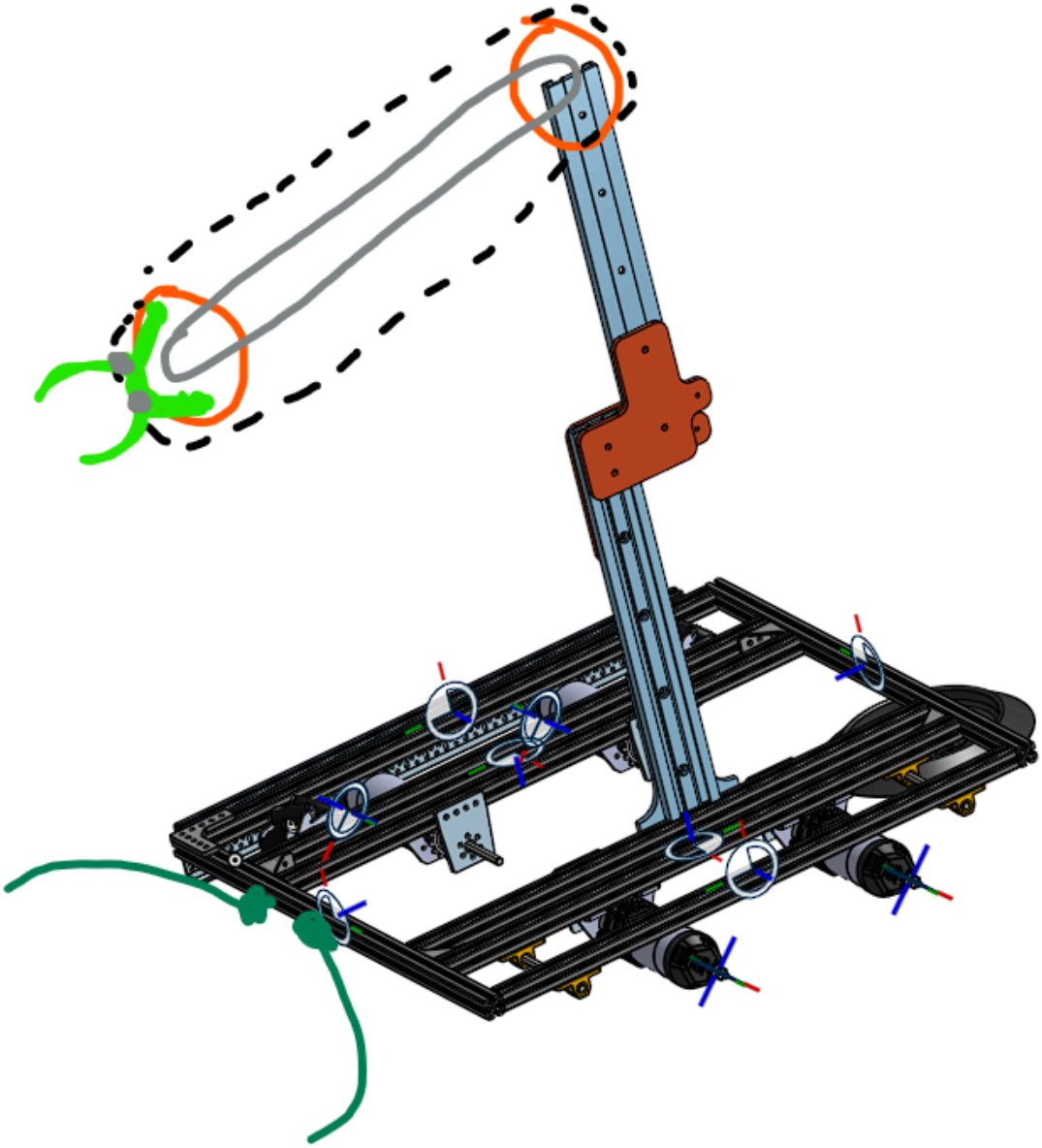
Picture: Centering mechanism for lifting mechanism and measurements for the cone while on the junction

Lifting mechanism

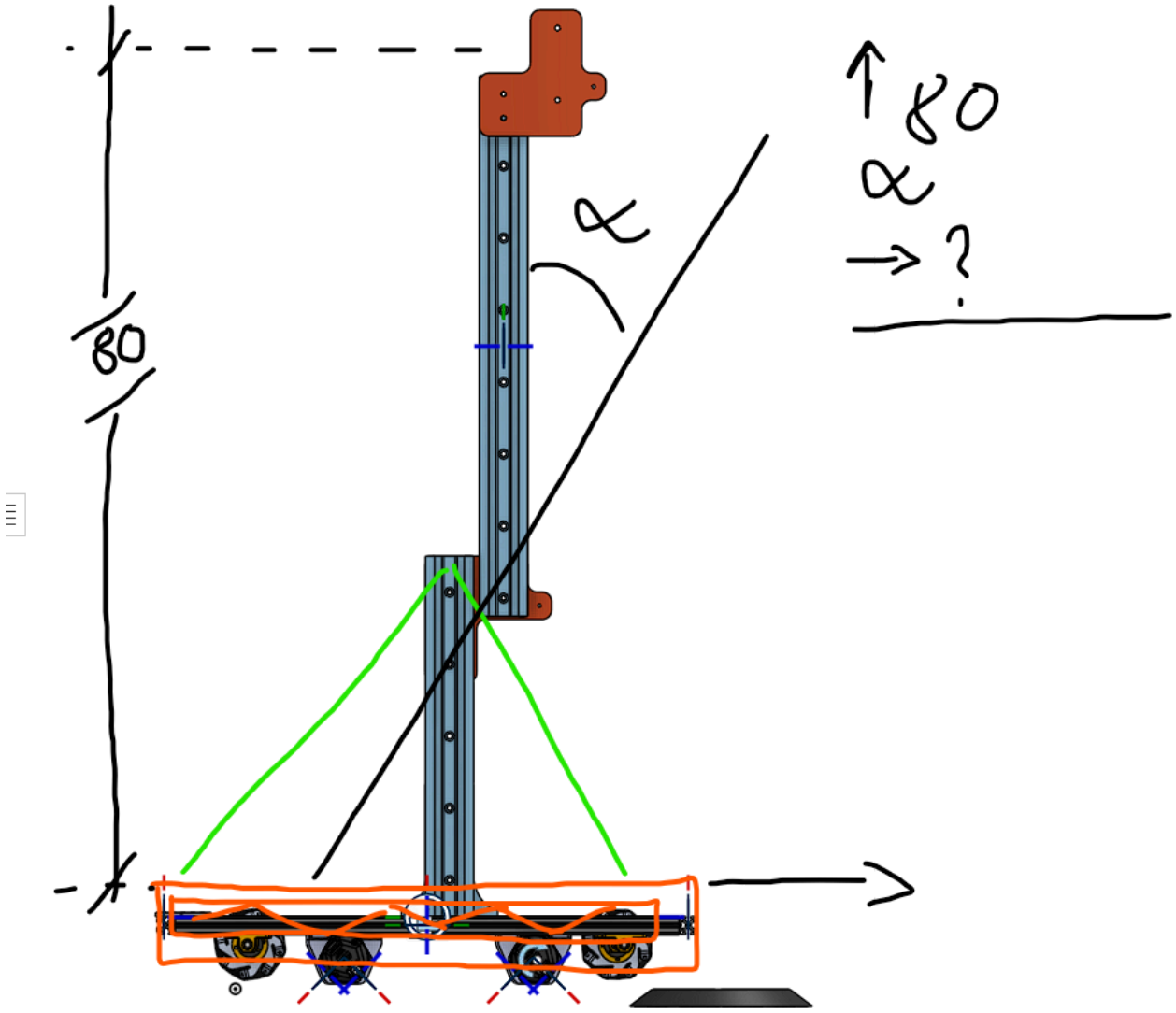
- Ball bearings on linear guides?
- Could we use linear guides from the drawer systems (perhaps in tandem)?



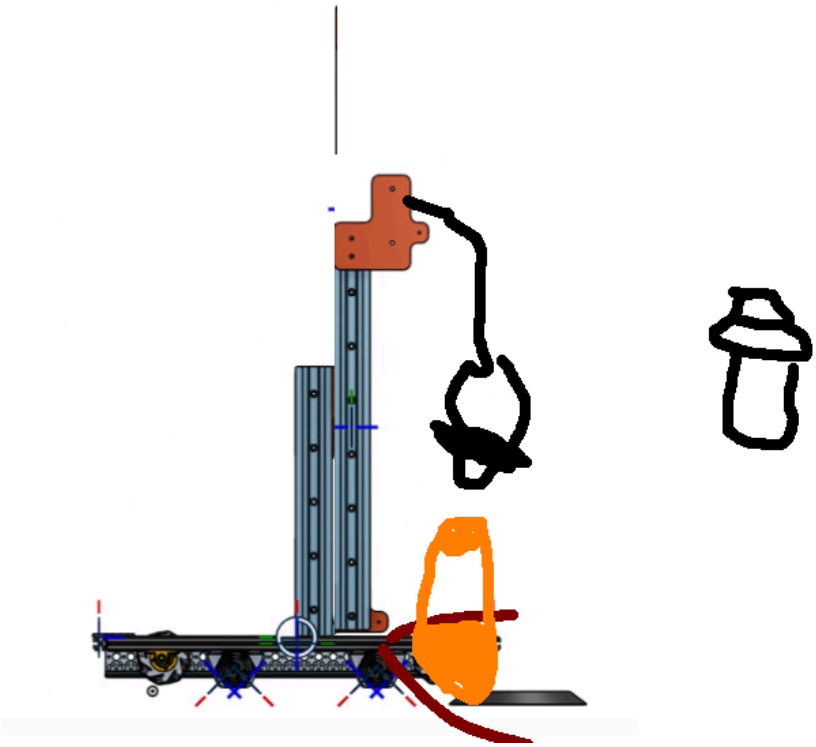
Picture: Robot Core Pickup



Picture: Third Stage Gripper

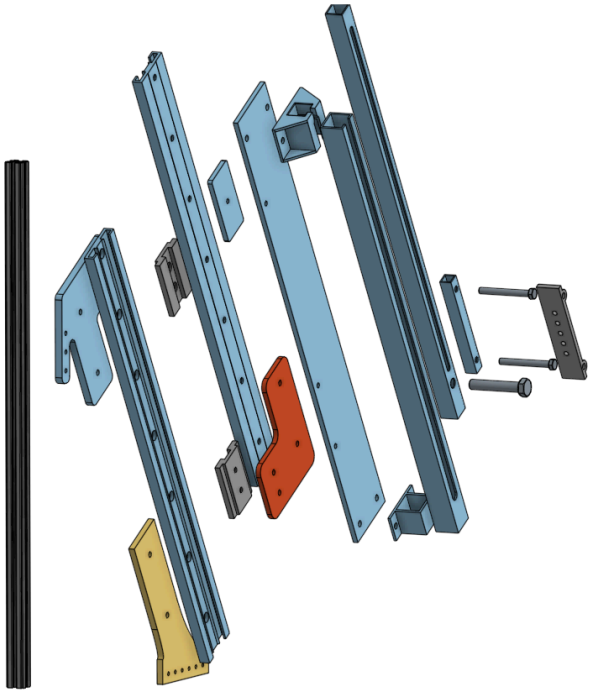


Picture: Angled lifting mechanism

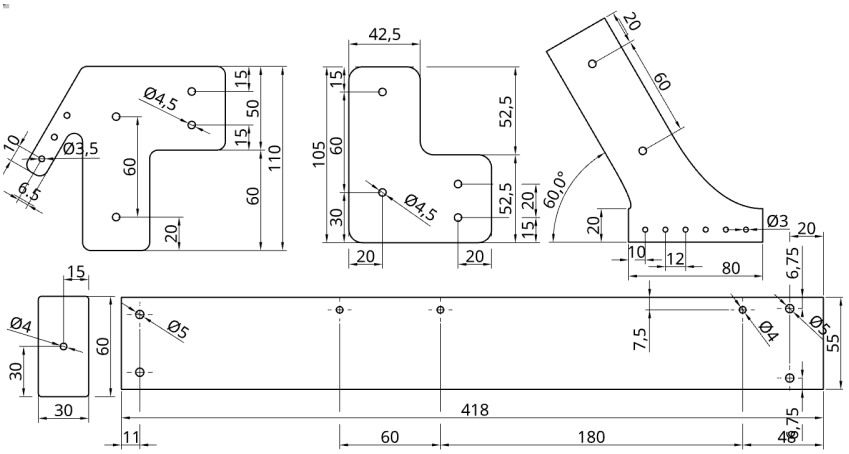


Picture: One type of the pick-up mechanism

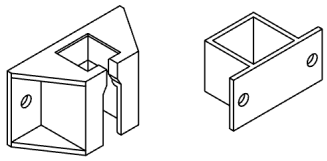
Before we landed on our current solution we drabbled around for quite a while. At first we thought the answer was to just stick the linear guides straight on top of the chassis, but soon we realized that, not only would we had had to do some serious stabilizing, we also didn't reach quite far out enough, even with the additional linear kit from REV (REV-45-1507).
Then we got an idea. We were in our local hardware store, and saw some square extrusions. It was all it was, in the beginning–an idea. But doesn't everything start that way?
We have the square extrusions now: three of them, each a different diameter. The smallest one goes in the middle and the middle in the largest.
That's how we ended up here, the idea becoming *the* solution.
This is how the whole mechanism looks together.



For mounting linear slides to chassis and combining two of those slides, we designed and laser cut adapters on detailed drawing below. Riser plates and square extrusions' adapters were also laser cut.

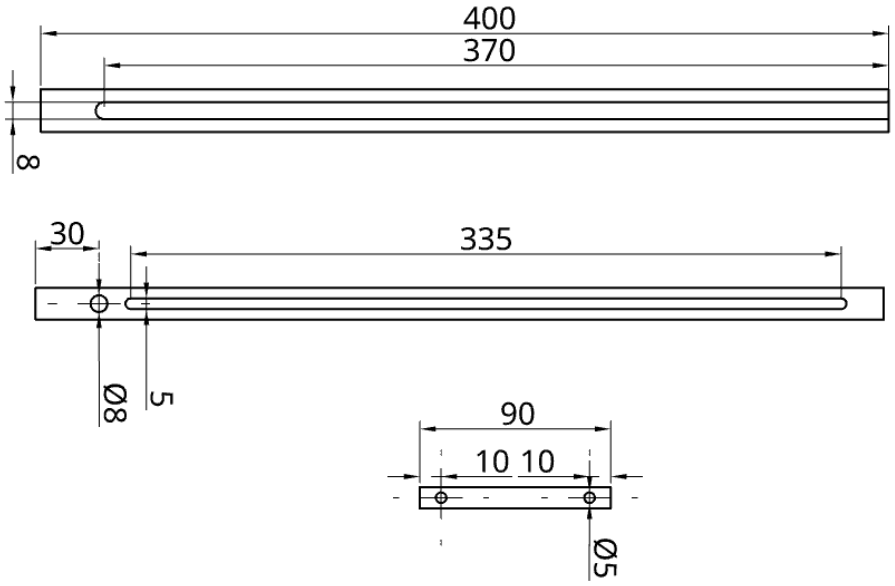


Brackets for attaching square profiles to linear rails were also 3D printed. The two adapters one on the bottom for holding the biggest of the profiles and one on the top for stopping medium profile from sliding too far but allowing the smallest one to still slide past it. Shown in the image below.



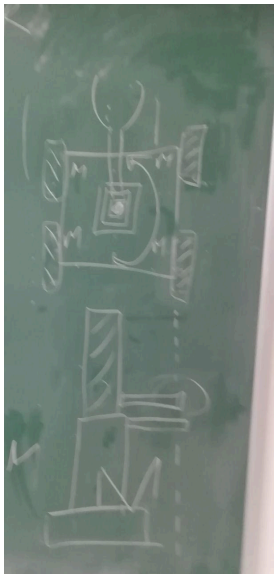
Picture: Brackets for holding the square extrusions in place

Square profiles also needed some processing. We needed to cut some grooves into them. 1st stage is a profile 20 mm x 20 mm x 400 mm that has one 370 mm long 8 mm wide groove which is perfect to fit M8 screw and it's open on one side (note: screw is stopped by aforementioned brackets). Second stage however is completely closed on both sides to prevent the third stage from sliding out. Middle stage is an extrusion with the measurements of 15 mm x 15 mm x 400 mm. It has a 335 mm long and 5 mm wide groove and a hole for the M8 screw used for sliding on the lower stage. Last stage is built from a profile 10 mm x 10 mm x 90 mm and it only has 2 M5 screws that slide in a groove and also hold out the griper. All slides are shown on the figure below.



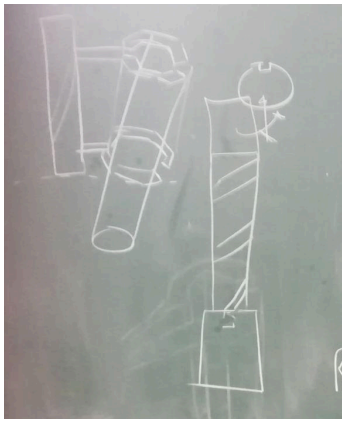
Cone manipulation mechanism

Shown in the sketch below is what we were thinking of building. An arm that's able to swing 180° in yaw and pitch axes. And of course in a perfect world that's exactly what we'd have on our robot, but due to our time constraints we couldn't build it. We will only do pitch rotation so we can pick the cone in front of the robot and place it on a junction on the back side of our robot.



Cone-Pole Targeting mechanism

We were also thinking of placing a camera on top of the gripper for ease of targeting. But as mentioned before we were in a severe time crunch and didn't even get around to starting the 3d model. On the sketch below one can see roughly what we were thinking of. Worth mentioning is also a centering gripper in front of the robot, so the cone would always be in same position relative to the robot.



In indeal world a camera would be added to the robot to make targeting poles easier but sadly we also ran out of time for that. We could also add centering gripper on the chassis to make pick-up easier or a bumper in shape of the ramp for centering.

Lifting complications

With our first design, the problem was that the cone was placed in a certain angle which wasn't ideal. We brainstormed a few ideas, such as "could we add rubber to the inside of the gripper,"

or “we could print a piece that would be in that same specific angle as the cone,” and finally we landed on a temporary solution. We took a traction wheel and put it on the left arm of the gripper–the one where the servo can be simulated. First we only tried this idea with the simulation of the servo motor in mind, but it just so happened to be good enough for now.

Software

General software

Software for our robot is coded in Java as well as our movement libraries. For our development environment, we have used Android Studio along with a custom upload tool.

Code uploader

This tool was made by reverse engineering OnBlockJava. We upload files to Control Hub using an API endpoint and start compilation. We track compilation using websocket endpoint. It saves a lot of time as the recompilation of the Control App is not required. This tool is currently still under development but it will be available as an Android Studio plugin in the future.

TeleOp

Two integral parts of our mission are driving and lifting. Both are represented by individual objects that are instantiated on OpMode initialization.

Driving

Driving was tough to get right since we are using Mechanum wheels. We are using two driving modes. Tank drive and our custom »strafe drive«.

No part could be made perfect though, so this is why we implemented the PID (proportional integral derivative) controller.

Our PID can be broken down into three separate components that affect the correction in different ways.

1. Proportional term (P): This term is proportional to the error and corrects proportional to the error magnitude.
2. Integral term (I): This term integrates the error over time and provides a correction for any accumulated error.
3. Derivative term (D): This term is the derivative of the error and provides a correction based on the rate of change of the error.

We total everything up in the end and adjust the robot's angular position.

Lifting

Target max lifting height is 82cm.

Our lifting mechanism uses two hex-core motors with encoders and a servo motor. Lifting is done in three stages. For the first two stages, the gamepad is used to set the extension percentage. For the third stage, we use separate buttons to lift the top of the arm precisely.

Pullback

Added pullback line on custom pulleys mounted on the 2 stage to reverse the motion of the arm, as it stands only second stage has it with plans for third stage to get it. First stage will probably remain without pullback line because the weight of the arm suffices to reset the motion assembly.

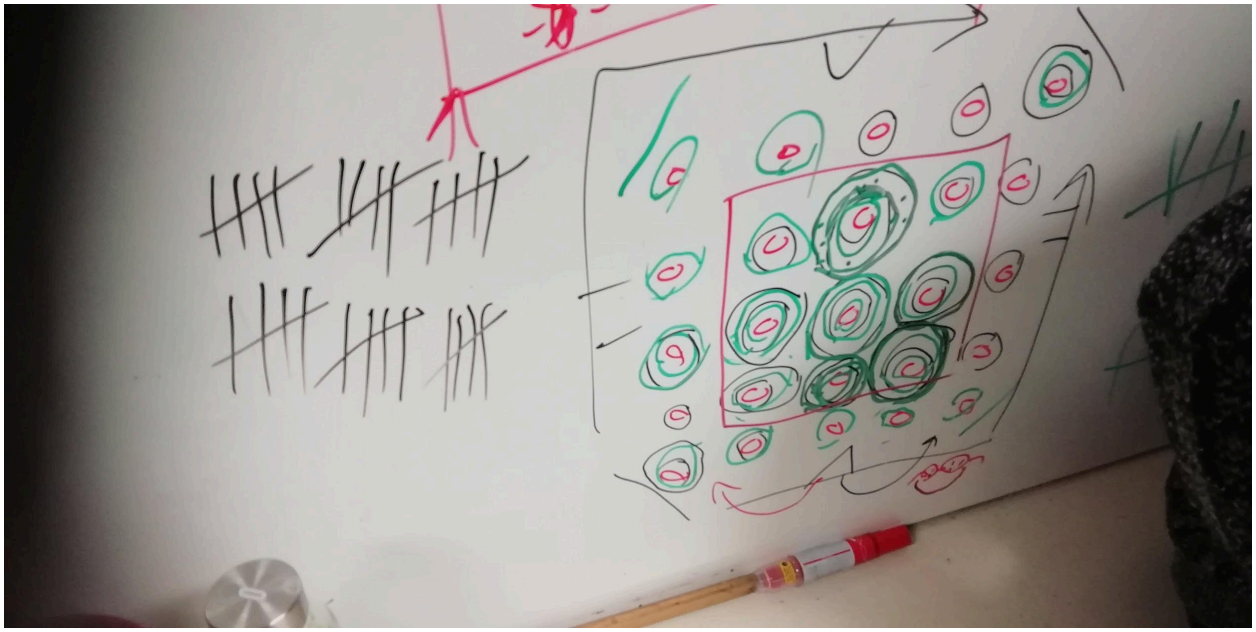
Autonomous

Since the strafe drive was already implemented we could focus on the computer vision side of things.

The game starts with robots scanning a cone in front of them. Robots must recognize the image and store the result for later. We have decided to use Vuforia. Since we made a custom sleeve for our team we had to train a custom Vuforia model. For Vuforia to recognize custom images we were required to compile a newly created model into the controller app.

Gameplay-plan

In this section we focus on field strategy and tactics we attempt to utilize on the fieldl. We played a simple game of stacked Tic-Tac-Toe, with junctions as places for Green-crosses or Red-circles, the winner had to finish this turn-based game with most points scored by PowerPlay rules whilst using all 30 cones. We are developing a simple video game to gain an edge over human players/drivers in manual mode of power play. → python, pygame, ftc, simple, strategy.
As such we decided to involve our local communities in the challenge as the playtesters for our game which will help us to decipher and predict various strategies that may or may not be implemented against or by us.



Picture: The game of tic-tac-toe we played on a whiteboard

Team Information

Luka:

A mentor, a model, and a motorcyclist, he's everything you might need except a good athlete. If you need that, look somewhere else. He's been through thick and thin at FGC since 2017, and now with FTC. What is more important, is that he doesn't plan on stopping any time soon, or to quote him: "As long as I can stand up and be here, I will." He breathes engineering, however his heart soars up there with planes.
He was a student at *Vegova Ljubljana*, the computer engineering program. Currently he is a mentor at *Zavod 404*, while painstakingly biting his way through a diploma for his last year of faculty study at the *Faculty of Computer and Information Science*.

David:

His nicknames consist of Zindo and Zindy, and if it was said before that Luka is not a great athlete, David is here to balance the mentor team out. He is a social butterfly, with a positive outlook on just about everything, really. He is made of bad jokes (which have saved the teams throughout the years more times than they'd ever say) and diligence.
As a student of electronics at the *Faculty of Electrical Engineering*, he likes to wander through the wilderness and reach new horizons. Who knows, maybe he'll discover a new dimension while at it.

Let-s start at the beginning. As a junior team this is our first appearance in first tech challagne as such we have had to adjust our perception of FIRST robotics competitions. The team had been freshly assembled for FGCs 2022 challange and had meny graduate students for whom this will be the last year of FIRST competitions as such we focus on laying a great foundation to continiue to inspire and motivate new young schollars in the field of robotics. As seid, the team had been assembled from past FGC competition team and we also had maid a considerable outreach getting new team member on board, as a school team we have had en easier time getting in touch with new people from the school, following this ideology the team members maid a great meny powepoint presentation in scholl going class to class and filling up the team member count. Next year, we hope to branch out and expand - out of our home school.

Getting a team together is the eazy part but getting work done is harder still, so accordingly we maid a plan of action to firstly analize the challange and begin brainstorming of ways we would aproach it. (include pic of whiteboard stuff) At this point

we were still mostly strangers to one another yet the skills members showed allowed us to rapidly assemble roles like, programmers, builders, designers… in the teams structure.
As a team we came up with meny ways to tackle the challagne with the help of mathematics and CAD we tried planning out the entire robot in CAD and then building it as a whole. Yet soon we found out that we are too unfemiliar with meny of production proccesess and assembly techincs to efficiently predict the mechanisms workings as such meny of them failed, battleing this problem we tried parralel development of CAD and building the daam thing.

—

This aproach was very innefficient becuse of back and forth debates betwen builders and designers. So lastly we scrapped as mush as we coud from CAD and built the robot by vision. We are also obligated to inform that we have used Onshape for our CAD platform due to its abillity to connect and colaborate eazily on meny diferent devices at once. We also had to prepare the CAD enviroment (inporting files (REV), fixing models and building examples) and train the team members, which took us a lot of time to do. As meny of senior team members were very femiliar with REV parts we designed the robot around them.

At the workshoop facility we had access to meny diferent machinery and tools to assist us with our work. All designe was based on low budget since we had little to no income and eaven less time since meny of our team engineers are close to graduation. As such we CNCed the aluminum slides 3D printed adapters and also 3D printed mechanucum wheels which we later upgraded with electrical wire shrinks for better grip. (explain all custome builds)

When we finally got the team, well, mostly yogether, we started with a few teamubilding excercises (screws falling over and scattering throughout our workshop).