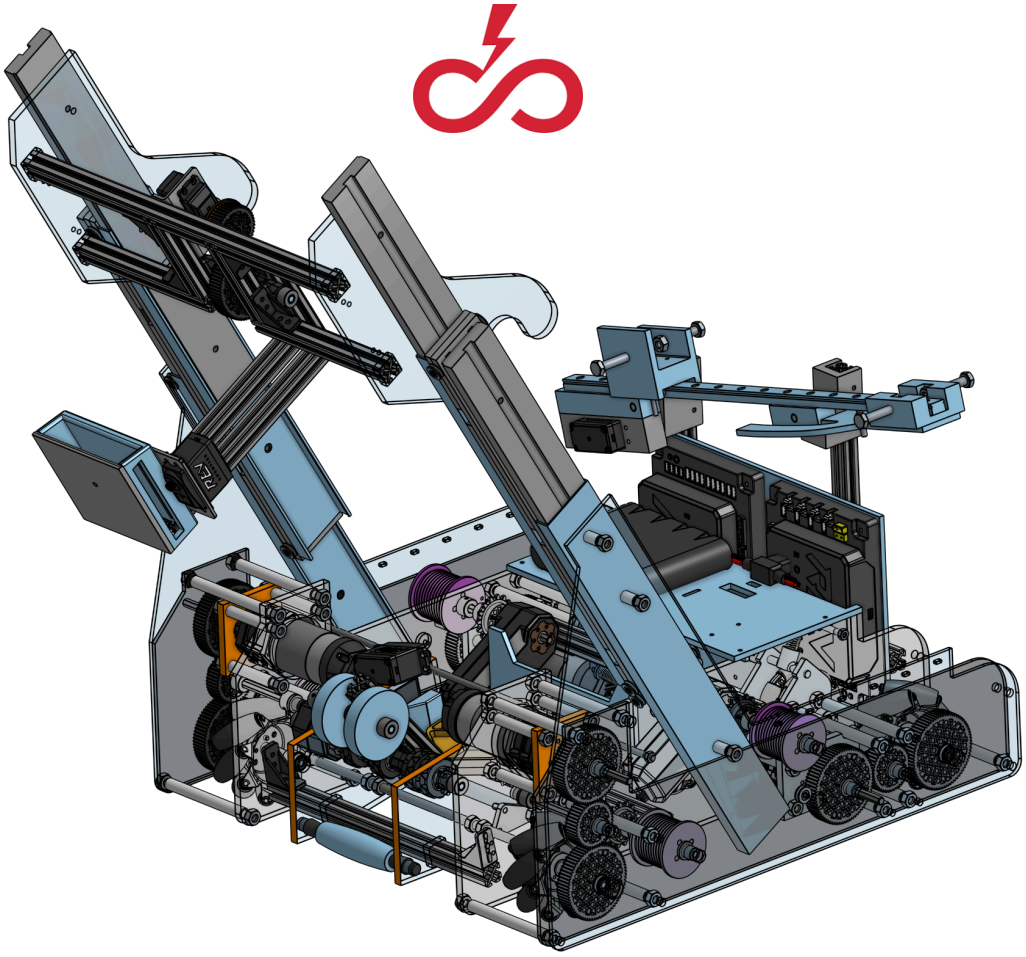
Engineering Portfolio

Team:

VEGAMIND

#22903



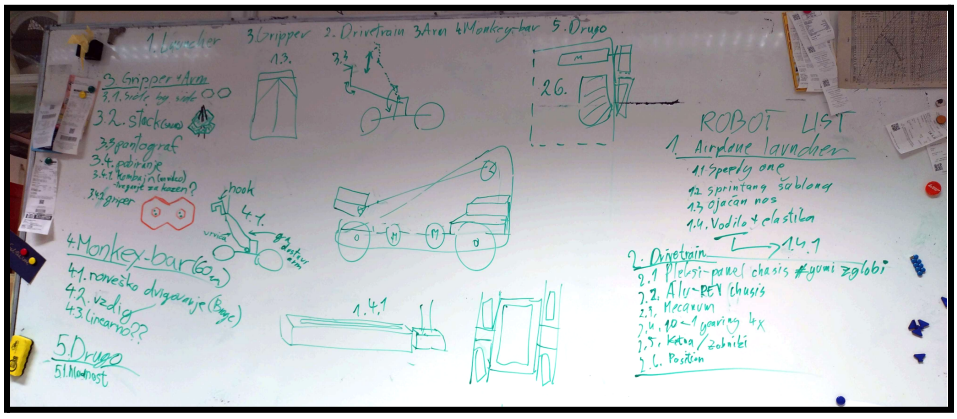
Game:

CENTERSTAGE℠

PRESENTED BY RTX

Game plan:

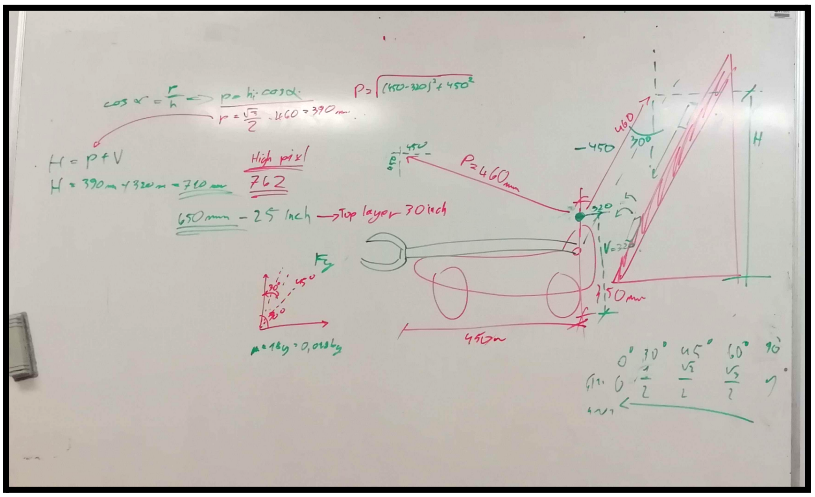
On initial few meetings we analysed the game and deliberated on the best way to complete our mission. To find the optimal design we set ourselves the following criteria:



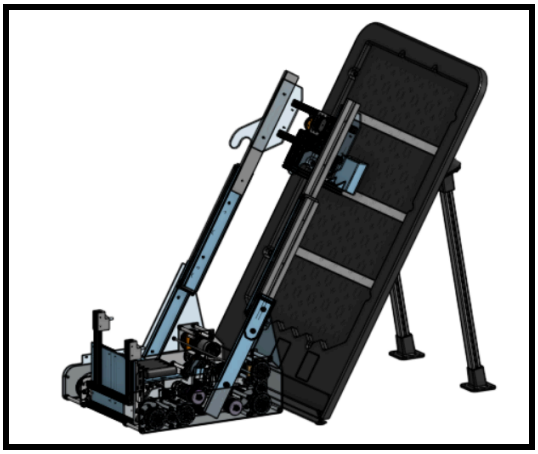
1. **Robot height** - to optimise the robot's efficiency on the field, we had to consider its height. We've decided on a maximum height of 355mm (14 inches), which will allow the robot to navigate under the *Trusses* in either direction without problem.

2. **Placing *Pixels*** - a challenge that surprised us early in our design process was the need for precise placement of the *Pixels* on the *Backdrop* without others falling off.

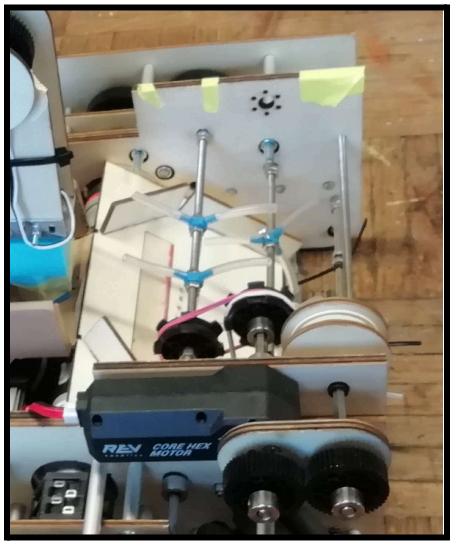
3. **Linear motion** - as we previously had trouble with either REV's Linear Motion Kit, or some simple linear slides bought locally, we set out to make the best arm we've ever built.

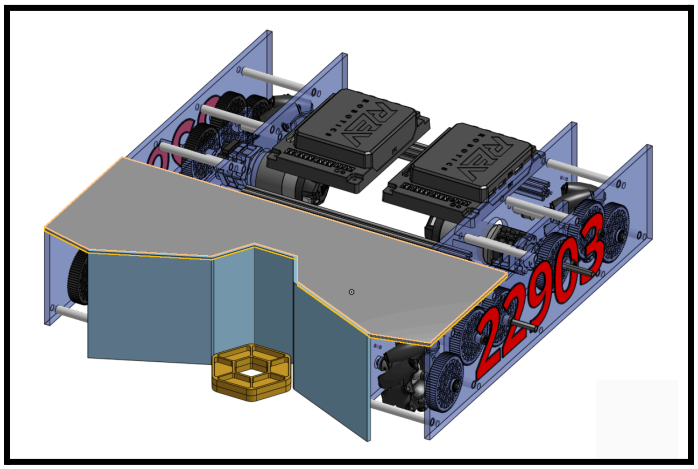


4. **Arm height** - in order to score as many points as possible, we tried aiming to reach the maximum possible height, but we ran into a problem. As we wanted a low robot, and that was more important to us, we decided on the second white line on the *Backdrop*.



5. **Intake system** - one of our first ideas when building the robot was to design a plough. It had issues with reliable control of the *Pixels*. If we want to pick them up, the robot must be moving which makes it harder to control the *Pixels* next to the walls. It's also unreliable for centering them into the pickup mechanism and insuring we do not control more than 2 *Pixels* simultaneously. We solved both issues by coming up with a belt-rollers intake system, which uses two compliant wheels to grab the *Pixels* and place them on a rubber conveyor belt. It is also able to unstack pixels that are in front of the robot.

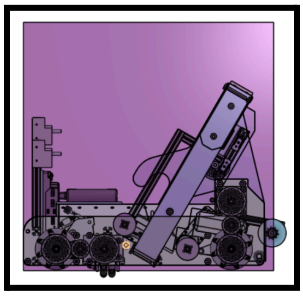


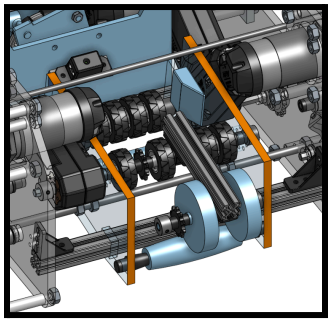


Robot design:

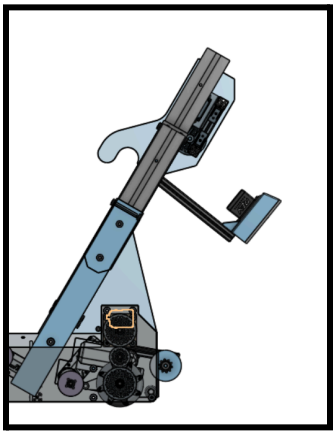
From the game plan we've identified a few key challenges to overcome:

1. **Lower height limit** - one of our key objectives was to be able to drive under the *Trusses*. To accommodate for yellow pipes on the *Trusses,* we set ourselves a new height limit. We hope this enables us to traverse the playing field more efficiently. Accomplishing this we've had to reconfigure drivetrain motors and the lifting mechanism from our initial design several times.

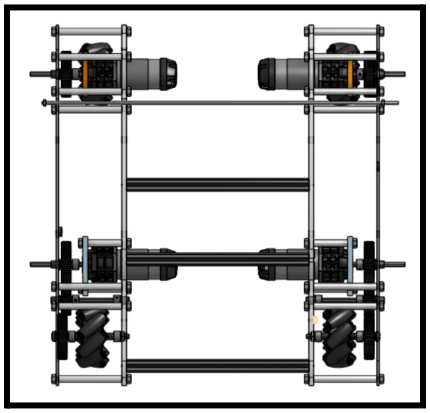




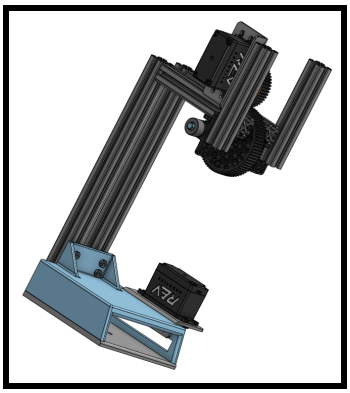
2. **Easily operated intake** - another key objective was manufacturing an easily manipulated intake with intuitive controls. We designed and built a **roller-belt intake** with a compliant wheel arm enabling us to unstack *Pixels* at the audience side (or anywhere in front of our robot). This also allows us to feed *Pixels* straight to the arm.



3. **Arm** - 2-in-1 mechanism built to accurately manipulate pixels on the *Backdrop* and lift our robot in the *Endgame* on the alliance *Rigging*. After preliminary testing, we decided to use ball-bearing drawer guides, as they were inexpensive, reliable, strong and readily available at our local hardware store. The arm is mounted at 60° so that it is parallel to the *Backdrop*.

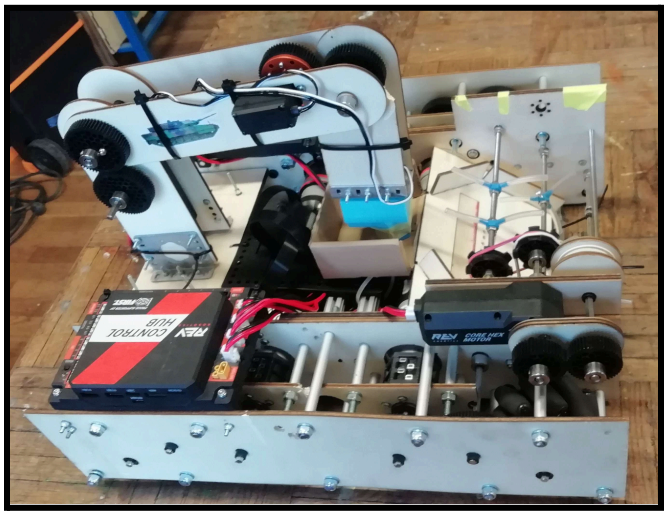


4. **Drivetrain** - the drivetrain is designed to be easy to build and quickly serviceable, enabling us greater adaptability if needed and an easily repairable system. By building a components stack resembling a sandwich structure, layering essential components between acrylic main plates and binding the structure with aluminium profiles (in the core), bushings and threaded rods (on the tracks). The drivetrain consists of two parallel tracks, each holding two wheel assemblies, consisting of a mecanum wheel and motor, joined together on an acrylic plate via a three-piece gear link.



5. **Hand** - for controlling the *Pixels,* we have decided to go with a rectangular pixel-sized box with a passthrough on one side. The *Pixel* is inserted by the belt from the intake, then the *Pixel* is locked in place by a servo motor, pressing the *Pixel* against the wall on the opposite side. The assembly is fixed in the linear guides by a swivel servo providing us with three fixed operating positions: closed, pickup and scoring. For reliability, this absolute position is also measured with a potentiometer.

6. **Rapid prototyping** - an engineering paradigm in which components can be easily modified, as designers find ways to improve them while enabling them to quickly identify when a part is “done”. It can reduce the amount of time spent on developing one part to be “perfect” while it’s already working “well enough” and can help guide engineers to spend their time on parts needing more development. As we have most of the woodworking tools in our machine shop and an 80 W laser cutter, we decided to make our robot prototype parts out of 4 mm plywood. It’s strong enough to allow us to drill additional holes in it, cut it with a saw, etc. After the bulk of design issues were addressed we manufactured new, competition-ready parts out of Acrylic.



Outreach:

Recruiting students from our school

Twice a year, before summer and after the FGC season, we visit almost every class at our secondary school to present our activities and give students an idea of what robotics involves. Most of our members joined us this way. Additionally, we also have students joining us from other STEM-oriented projects happening at our school.

Workshop - we held a 3D modelling workshop for students at Zavod 404. To get them started from the very basics, we used Onshape.

Fairs and events

Information day - information days are annual presentations held at every secondary school and are meant for primary school students to get first-hand information from students and professors of the schools they are interested in. In the student section, we are one of those projects that draw the most attention from the crowd. Most students attending don't know about the existence of FTC and FGC before meeting us, so we are always happy to inform them and invite them to potentially join our team in the future.

MOS - is a big international fair meant for different kinds of craftsmen to show off their work. We were included in the educational sections, where our robot attracted a lot of attention. We also met the Slovenian Minister of the Economy, Tourism, and Sport.

Elektrofest - is an event held at the Faculty of Electrical Engineering of the University of Ljubljana, and it's meant for secondary school students in the electrotechnical field. The festival is meant to display the most prominent electrotechnical projects happening in the city of Ljubljana. Our goal there is to spread awareness of *FIRST*, encourage other schools

to get involved with FTC, and potentially recruit new members from other secondary schools.

Informativa - is a big fair that gets all secondary schools and universities in Slovenia in one place, where students can walk around and visit the stands that pique their interest. We are one of the most prominent projects being displayed at our school's stand. We draw attention to our robot and inform future students about FIRST and robotics.

Zavod 404 final conference - is our workshop's end-of-year conference, where every project that finds a home at 404 presents itself and its activities from the previous year.

New Technology Experience

As part of this year's FGC New Technology Experience: Energy Evolution, we visited experts in the fields of nuclear energy and photovoltaics.

JSI Nuclear Reactor Physics Department - we were welcomed by Urban Pompe, one of their experts, who first presented us with the basics of nuclear power. As half of us have an electrical engineering background, he explained everything in-depth and answered all of our questions very scientifically. Later, we also visited the science exhibition on site, with him guiding us through.

The Laboratory of Photovoltaics and Optoelectronics - at the Faculty of Electrical Engineering of the University of Ljubljana, we were welcomed by Assoc. Prof. Dr. Benjamin Lipovšek and Asst. Prof. Dr. Marko Jošt, who first explained the principle of photovoltaics and their usage and then debated with us on how we could include newly learnt knowledge in our report. They showed us the laboratory and quickly explained some of the current projects they were working on.

Meeting with other teams

Stuttgart - last year's scrimmage in Stuttgart, held by Frogs 10183 and Frox 10089, was the official start for our team, so we try to attend it every year. This year, we attended it with a very early and rushed prototype version of our robot. It was great practice, and it gave us a good idea of what the game is actually about.

FGC 2023 - was a great opportunity to build deeper relationships with other teams. We connected most with Team the Netherlands (16441 - Pretty Smart Robotics), Team Suriname (9135 - STEAMSur 9135), Team INew Zealand - The Blackbots, and the Slovakian team.

DecebalTech 19105 help - we thank this team for the great help we received, especially Matei, who helped us with Road Runner setup as it was our first year using it.

Other - this year we also meet up with teams: 5980 - East Grand Rapids, 19043 - CyLiis, and 16031 - PARABELLUM.

Sponsors

HYCU - we toured our workshop with their HR representative and gave her a run-down of our activities in the previous years and our goals for the future. She loved our work and suggested a partnership.

Presentation to Mayor of Ljubljana, Zoran Janković - he accepted us at his office in the town hall, where we presented him with our activities and our goal of bringing robotics to as many students as possible. He was very interested in our team and our ideas, so he decided to sponsor us and give us a spot in the newspaper "Ljubljana" (year: 29, month: February 2024, page: 29, paragraph: 6).

Software:

Image Detection

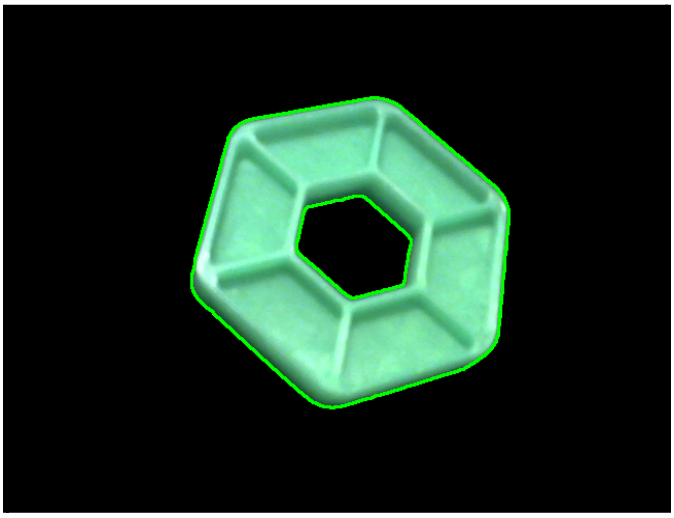
This section of the documentation dives into the inner workings of our image detection system and the reasoning behind the design choices we've made in this part of our software. First, we considered the challenge we tried to solve so we can better understand the used method and the path leading towards its implementation. Originally, our team wanted to use machine vision to detect *Pixels* on the *Backboard* so that we could develop an algorithm that would help the robot place *Pixels* autonomously even outside the *Autonomous period.*

Picking the optimal tools

Now that we understand the objective, we just have to bring a solution to life. We first had to pick the proper tools for the job. Since complex programming is the key to great autonomous Java has got an advantage over Blockly. Next, we had to decide between OpenCV and TensorFlow, since both have their unique features, we'll introduce both. OpenCV is one of the biggest machine vision libraries that comes with many built-in functionality and documentation.TensorFlow is a framework for creating and deploying machine learning models that could in our case be used for detecting pixels. We decided to use OpenCV since it doesn't require training and is in general easier to use.

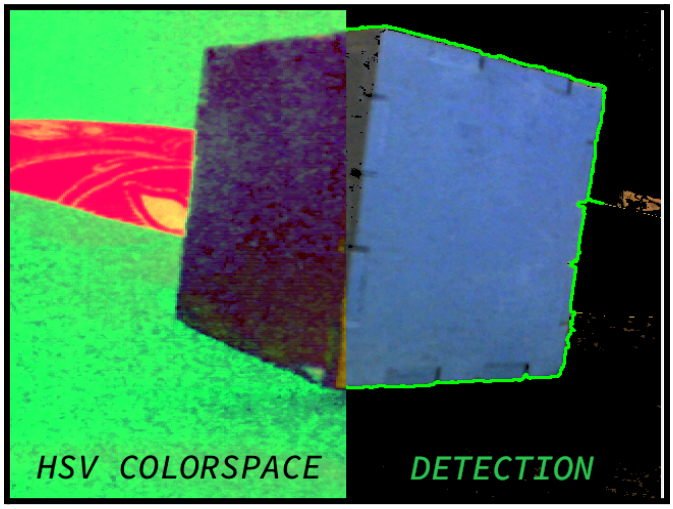
v1 Pixel detection - RGB Color and Shape Detection

The first thing that you see when you look at a *Pixel* is its unique colour and distinct hexagonal shape. Based on those observations we decided to implement *Pixel* detection based on the mentioned attributes. This is done by going through the image captured by the camera and seeing if they're in range. If pixels have the right colour and they form the correct shape together they are recognised as game elements. After the implementation, we realised that if we wanted to use pixel detection for detecting *Pixels* on the *Backboard, and* then autonomously placing them into mosaics we would also need to develop an algorithm for the optimal placing strategy.



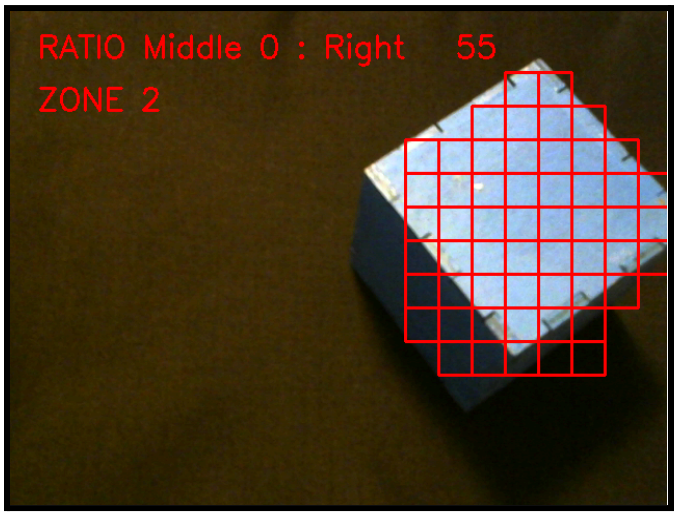
v2 Object detection - HSV Color Detection

Even though the original idea failed, we were still left with a very robust system for image detection based on colour, shape, and size. With some modifications, we now reuse that algorithm for *Team Game Element* (*TGE*). Since the world is full of imperfect conditions, that lead to shadows on the team item and consequently suboptimal detection based on colour so we tried the HSV colorspace which provides colour data separated from its luminosity. That enabled us to detect the position based on the *TGE* without the lighting playing much of a role.



v3 Object detection - Luminosity and Segment Detection

The only problem left is knowing which position the *TGE* is at. For now, we only know the position on the screen. To acquire this information, we decided to segment the camera input into a 20 x 15 grid of segments. We also updated our detection method from using colour to using luminosity. The idea stems from the fact that the camera can be positioned in such a way that it only captures the floor and the spike marks (there is a difference between their luminosities). Sadly the camera can only be reliably mounted so it captures only two of the spike marks at once. Because of these limitations, we check if any of the two zones contain a certain amount of segments, we can deduce where is the *TGE* positioned by checking if it's on any of the two visible *Spike marks* and if not we assume it's on the third one.



Control Award Submission Form:

Autonomous Objectives

The key idea to consider when developing auto is to make it as reliable as possible. Thus, for this year, we have decided to use a well-known and ruthlessly tested library Road Runner (RR). Our strategy for placing the *Pixels* on *Backdrop* is the following: do the least number of rotations and keep the path as short as realistically possible. The short path between locations allows our robot to not interfere with others and allows us to keep our codebase clean and simple. Our robot can reliably place a purple *Pixel*, preloaded in its intake on the correct Spike mark and a yellow *Pixel* on the correct *Backdrop* position.

Sensors used

1. Rev HEX encoders in combination with voltage sensors allow our robot to determine its location. After carefully tuning RR, we can track our robot's position quite reliably.
2. Control hub IMU allows us to track rotation. Rotation information is used by RR to follow predetermined paths that the robot takes as well as correct for unforeseen disturbances. Rotation information also enables our custom "strafe drive" to correct potential misalignment late in the Driver-Controlled Period.
3. Rev 2m distance sensor is used for final alignment before placing the yellow *Pixel*.
4. Logitech webcam is used to determine the randomly assigned location of the TGE. The computer vision part is done using OpenCV. For location detection, we use our custom rendering pipeline.
5. Magnet limit switch is used as a failsafe to prevent over-extending and thus damaging our lifter arm assembly.
6. A potentiometer is also used for additional tracking and mapping of the location of the *Pixel* depositor.

Key algorithms

1. "Strafe drive"

Strafe drive is a custom drive mechanism we started developing last season. It has worked reliably for us, so we have decided to improve on it this year. What makes it unique to its counterparts is mostly the control scheme. It uses the combination of PID correction and IMU compass to navigate the robot throughout the field.

2. Custom rendering pipeline

The method aims to determine the position of the *Target Game Element* (*TGE*) using camera input segmentation and luminosity analysis. The camera captures a 20 x 15 grid of segments, focusing on the floor and *Spike marks* with distinct luminosities. By analysing the segments within two visible *Spike marks* zones, the *TGE's* position is deduced. If not found in those zones, it's assumed to be on the third *Spike mark*. This approach optimises *TGE* detection despite camera positioning limitations. For more information, refer to the portfolio in the section “v3 Object detection - Luminosity and Segment Detection.”

Driver-Controlled Enhancements

Since a few of the team members (and mentors) play (or have played) video games often, we have decided to take video games’ control schemes as an example. It relieves the mental strain that could be worsened by an unintuitive control scheme. The main ergonomic feature, despite allowing free rotation, also provides the driver with predefined rotations offset by 90°. They are switched by using a D-pad on the primary controller.

Engineering Portfolio References

In our Portfolio we describe our custom rendering pipeline in detail. We also describe our RR setup and our cooperation with other teams that greatly helped with our setup.

Autonomous Program Diagrams