ISB TEAM D

ENGINEERING NOTEBOOK

86832D

10

Continued from last page

Digital Notebook Rationale

Team 86832D has decided to utilize a digital notebook format for the 2023-2024 VEX Over Under season, which will be done through a shared Microsoft Word document. In the process of making this decision, we considered the pros and cons:

Pros & Cons of Digital Notebook Use

Pros	Cons
• <u>Collaboration is possible</u>: Multiple team members can work on the notebook at once • <u>Readability:</u> Handwriting sometimes impairs understanding. A digital notebook is clearer • <u>Adaptability to Schedule:</u> The notebook can be worked on outside of working hours, allowing us to focus on building/programming • <u>Sustainability:</u> Only the exact number of pages needed will be used; physical notebooks from previous seasons tended to waste a lot of paper from unused pages • <u>Workflow:</u> The use of a digital notebook could simplify workflow, where during session times, we work on creating, and after we work on reflecting/documentation	• <u>Legitimacy of notebook:</u> Digital notebooks tend to have reduced legitimacy compared to physical notebooks • <u>No individual handwriting:</u> The different handwriting of different members, reflected in a physical notebook, may not be seen in a digital notebook • <u>Printing:</u> A digital notebook may need to be printed before some tournaments, which may be troublesome • <u>Unfamiliarity:</u> Digital notebooks are new for the entire organization as a whole so that time might be wasted in the process of learning

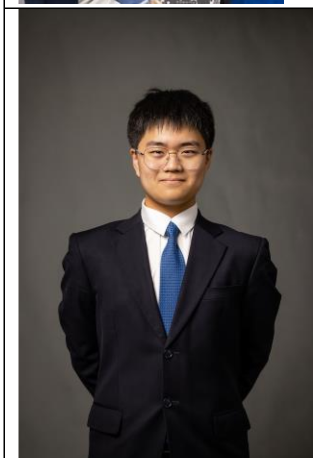
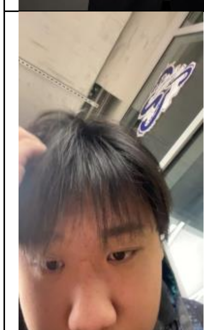
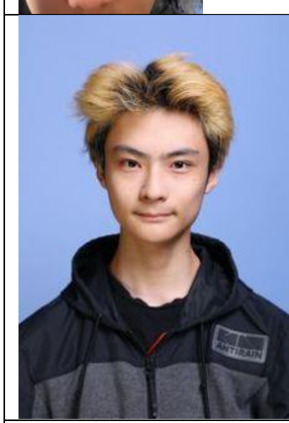
Other notebooks for teams 515R (Baxter et al.) and 3249U (Daugherty and Hartline) were referenced in the process of making this notebook. Some elements of this notebook, namely formatting, entry templates, and page organization, were heavily inspired by the other notebooks referenced. However, most of the formats/templates used were created independently in the process of making this engineering notebook for the purpose of adapting those examples to Microsoft Word.

Works Cited

Baxter, Logan, et al. *3249U Engineering Notebook*. Barnes and Noble Press, 2023. *Google Docs*, Barnes and Noble Press, docs.google.com/document/d/1c95OEhynTWcst4PPALBGL9YrfcwObkSUmUEsIXWR2iM/edit?usp=sharing. Accessed 3 Oct. 2023.

Daugherty, Logan, and Ben Hartline. *515R Revision*. 2023. *Google Drive*, drive.google.com/file/d/1HaW4U9oFkCO0W1oqMPX5Eugy24MAw_wm/view. Accessed 3 Oct. 2023.

Team Profile

	Name: Samuel Yao [Left since 2023 Dec 9] Grade: 11 (Class of 2025) VEX Robotics Experience: 2 Years Roles: Team Lead, Lead Notebooker, Designer, Builder
	Name: George Xu Grade: 11 (Class of 2025) VEX Robotics Experience: 2 Years Roles: Lead Designer, Lead Builder
	Name: Lucas Duan Grade: 11 (Class of 2025) VEX Robotics Experience: Roles: Programmer,
	Name: William Pan Grade: 10 (Class of 2026) VEX Robotics Experience: 1 Year Roles: Driver, Builder, Lead Programmer (Team Lead and Lead Notebook Writer after Dec 9)
	Name: David Nam Grade: 11 (Class of 2025)

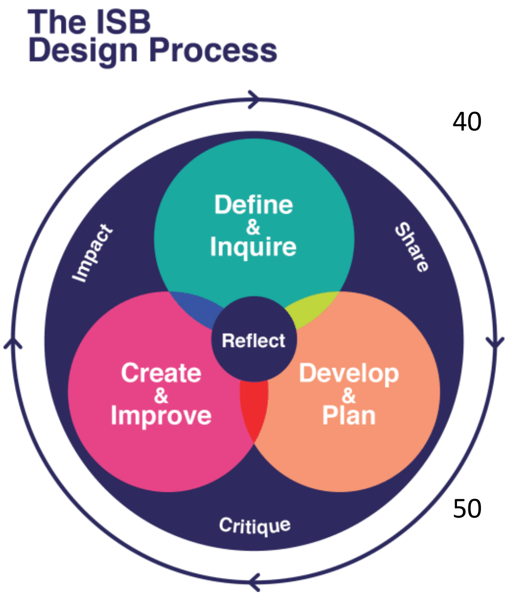
Title: Team Profile	Author: Samuel Y	Date: Nov 22 2023

Continued from last page

	VEX Robotics Experience: 1 Year Roles: Designer, Team Scout, Builder
	Name: Matthew Kim Grade: 11 (Class of 2025) VEX Robotics Experience: 1 Year Roles: Designer, Team Scout, Builder
	Name: Carolyn Liang Grade: 11 (Class of 2025) VEX Robotics Experience: New Roles: Builder, Designer
	Name: Minjae Kwon Grade: 9 (Class of 2027) VEX Robotics Experience: New Roles: Builder

Engineering Design Process

Team 86832D (and all other ISB Robotics teams) will go through the International School of Beijing Design Process (referred to as the **ISB Design Process** or ISB Design Cycle), which is a version of the design process very similar to the VRC Engineering Design Process, or the International Baccalaureate Middle Years Programme (IB MYP) Design Cycle ("ISB Design Process").



ISB Design Process, showing the 4 stages: Define & Inquire, Develop & Plan, Create & Improve, and Reflect

Simplified Engineering Design Process ("Engineering Design Process")

The various stages are as follows:

Define & Inquire relates to the Define stage in the VRC Engineering Design process, but differs in terms that this stage does not include design planning. The purpose of this stage, for our purposes, is to **fully understand the problem,** conduct secondary **research** in relation to the design and to inquire into the **skills/techniques** necessary to succeed in the design.

Develop & Plan is about the specific planning that is prior to the creation of the design. Although similar to the Develop Solutions stage of the VRC Engineering Design Process, this stage of the ISB Design Process focuses exclusively on the **modeling** of **design ideas** through **prototypes**, the **planning** for the creation of a design, and determining the **success criteria** for a design.

Create & Improve overlaps with the Develop & Plan stage partially, where prototype creation and testing also falls under this stage. This stage includes the Optimize phase of the VRC Engineering Design process as well as parts of the Develop Solutions stage since optimization to prototypes is part of this stage. Other than the **creation of a final product** that is functional, this stage also involves the **creation of prototypes** based on design choices and the **testing of prototypes** to gain **data and feedback** which allows for **redesigns** and **improvements**.

Reflect is done throughout the entire design process, similarly to how "Document it All" is emphasized within the VRC Engineering Design Process. This stage involves gaining **feedback** from others, considering the **strengths and weaknesses** of a design, explaining the **impact** of our design and **share** the product to mutually learn from others. Thus, this stage encompasses **Impact, Critique, and Share**.

Our team ultimately chose to utilize the ISB Design Process as the framework to design our robot with. Because of our inherent familiarness with this process, we believed that this will be the most suitable and most efficient process for our purposes.

To indicate the implementation of the stages, the respective stage that each heading in the notebook relates to will be highlighted, with teal/turquoise for Define & Inquire, orange for Develop & Plan and red/pink for Create & Improve. Because reflection is done constantly (through the means of this notebook), reflection is not highlighted. The highlights will be seen at the main heading of the entry and the table of contents.

Works Cited

"EdTech and Design." *International School of Beijing*, www.isb.cn/learn/curriculum/edtech-design. Accessed 11 Oct. 2023.

"Engineering Design Process." *REC Foundation Library*, Robotics Education and Competition Foundation, 2022, vrc-kb.recf.org/hc/en-us/articles/16999409685911-Engineering-Design-Process. Accessed 11 Oct. 2023.

"ISB Design Process." *International School of Beijing*, www.isb.cn/uploaded/ISB/Learn_With_Us/docs/ISB_Design_Process-Standards20180523.pdf. Accessed 4 Oct. 2023.

Team Goals & Vision

Team 86832D has decided on the following goals for the 2023-2024 VEX Robotics Competition season:

1. To earn a judged award at the TIS Robotics Challenge in February or another VEX Robotics Competition tournament. We want to hold ourselves to standards beyond the typical VRC team; therefore we believe that by subjecting ourselves to this goal, we have standards to compare and aspire towards.

2. To design and compete to the best of our abilities by committing our time and effort to the fullest. In order to achieve these aspirations for our team, we will compete to our fullest abilities.

3. To design a competitive robot that is in the top 25% of robots at the TIS Robotics Challenge, the Asia Pacific Athletics Conference (APAC) VRC tournament, and the ISB Robotics Tech Challenge. This will be determined through our ranking in robot skills and the tournament overall (both in qualification and elimination).

4. To learn from our experiences, learning practical skills throughout the tournament season that are applicable to contexts outside of VEX and the VEX Robotics Competition.

5. To complete the a full prototype of a competition robot (full design cycle) prior to the ISB Robotics Tech Challenge and the TIS Robotics Challenge, including inquiry, designing, creating and reflecting. This will ensure that we make appropriate revisions and considerations for our robot design.

Sample Meeting Entry

Page Heading (Used for larger entries/different tasks)

Section Heading

Duration of session: __ hrs				
Task	Team Members	Status	Deadlines	Notes
Task 1	Samuel	Complete	1/1/23	Finished task, but wheel needs to be fixed
Task 2	Charlie	Halfway Done	1/1/23	80% done, finishing touches needed
Task 3	William	In Progress	1/1/23	
Task 4	George	Not started	1/2/23	

Body paragraphs are used within this notebook for meeting entries, elaborating on some of the pictures/visual elements of each entry, or to elaborate on certain aspects of the robot design process that is impractical to talk about solely in visual form, bullet points, etc. The level of detail expected from all notebook entries overall is high, to the extent that another person could use this engineering notebook to recreate the designs made by this team.

Meeting Start Time: ##:## am   Meeting End Time: ##:## pm

Name	Previous Task	Status	Notes, Obstacles	Today's Task	Deadlines
Clark	Drafting x-drive design	In progress	Working on gears	Finish structure	9/33/22 Designs done
Jonah	-Not Present-	Not started	N/A	Design clamp	10/29/22 First Competition!
Joseph		Half way			
Logan		Almost Done			
Yuri		Finished			

Fig. 1: 3249U's team management table, put at the start of each entry (Baxter et al.)

The table used on the top of each meeting notebook is inspired by the task management table used by team 3249U in their digital notebook. However, instead of focusing the overall planning/objective completion on each individual team member, the table is sorted by task, then the team members assigned to that specific task.

For pictures or other **visual diagrams** included in the notebook, they **will be assigned figure numbers** (starting from 1) for reference. The sizing of visuals should be appropriate so that the content displayed is clear. The way that these figures are used within the entries, whether separate from body paragraphs or embedded/wrapped in a body paragraph, is up to the discretion of team members if the visual can be clearly discerned regardless of formatting. **Visuals with similar visual composition**, such as drawings with largely similar elements but slight differences to show iterations, **should be reduced to a minimum**, if possible, **or scaled down**.

Title: Sample Notebook Entry	Author: Samuel Y	Date: Oct 1 2023

Certain sections of the notebook will show **paragraphs in two columns**. This **is used when a significant portion of the page** (over 1/3) **consists of only body paragraphs**. This is used to improve readability. Columns should not extend beyond page breaks, so if any paragraph or column extends beyond the current page, then the paragraph should be written in the next page. Team members should note that this can be done manually via the "Layout" tab in Word.

Data tables should be used in meeting entries to represent the data collection, design matrixes or other quantitative values relevant to robot design and construction. **Design matrixes should be used** throughout the design & develop stages of the notebook **to have a quantitative means of making decisions** that are otherwise qualitative or subjective. Similarly, graphs or data tables with quantitative data should be used to show experimentation and reflection in prototyping and designing. Design matrixes and data tables that are large should be inserted as a picture after made using other software such as Microsoft Excel, but those with few columns (less than eight) should use a data table created in Microsoft Word.

The date reflected in the right side of the footer (bottom right of page) and **the date indicated** in the table of contents **will reflect the last edited date for a certain page**. Because the notebook is intended to represent a chronological order of events, previous entries and pages should not be edited. If necessary, this can be verified by looking at the "Version History" section in Microsoft Word Online. Due to slight disorganization near the start of the season, where teams were only formally decided by mid-October, entries and sections near the start of the notebook may not be fully chronological. A dedicated section of **"Pre-Season Work" indicates inquiry, planning, and designing work** that was undergone **before teams were formally decided** by current team members of 86832D, although some of the work undergone may include ISB Robotics Members competing in other teams.

The formatting of the notebook, as indicated with this entry and other preceding entries, shall be maintained throughout the entire notebook, including font size, font, highlight colors, etc. **Key words** within long paragraphs should be <u>underlined,</u> **bolded**, or **<u>both</u>**, in this given hierarchy to allow for more ease in reading, if necessary. References to outside sources should reflect MLA 9 styling with the use of in-text citations and a Works Cited placed at the end of each entry.

Large parts of the page that are not being used shall be crossed out with a line to indicate that it is not being used. This should be done with any blank space that is greater than 6 lines of font size 12 text. This can be done by going into the "Insert" tab, inserting a line in the "Shapes" menu and dragging it diagonally so it covers the blank space of the page, from the top left to the bottom left. The line should be anchored to the bottom of the page.

(Samuel) In some situations where multiple team members have contributed to a page, the person's name in brackets—such as for this paragraph— will be inserted for every section written by a given team member.

Pre-Season

Prototyping: Climbing Mechanism

Prior to the 2023 summer, George worked on creating a prototype for the potential development of high hang capabilities. A physical prototype was created to demonstrate a potential idea for this being implemented.

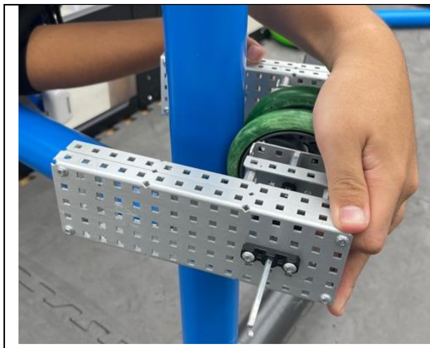


Fig. 1: Side view, prototype

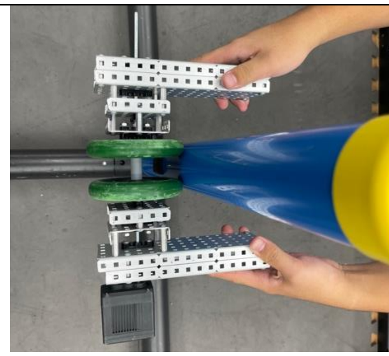


Fig. 2: Top view, prototype

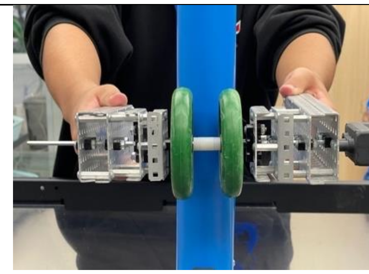


Fig. 3: Front view, prototype

Prototyping: Drivetrain

At the start of the 2023-2024 school year, robotics members from various teams worked on potential prototyping that could be used. Previously, the drivetrain used by all four teams was a regular 4-motor drive, but team members (mostly from 86832C and 86832D) wanted to explore the use of drivetrains where torque can be transferred more evenly.

The first two concepts that we wanted to experiment with are a geared drive and a chained drive. The geared drive will use gears interlocked to link the wheels together, and the chained drive will utilize sprockets connected using chains.

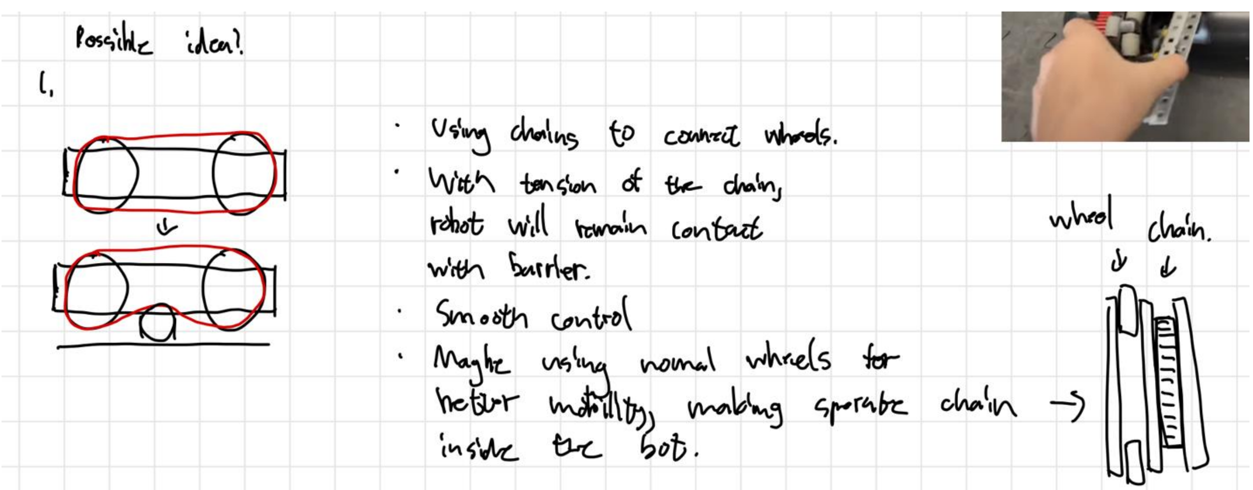


Fig. 4: Diagram/bullet points used in consideration of a chained drive, made by Myungjun Lee (86832C)

Title:Pre-Season:Define & Inquire	Author: Samuel Y	Date: Nov 13 2023

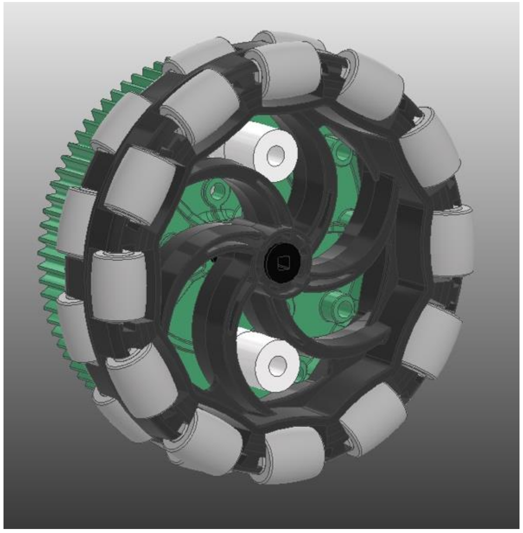


Fig. 5: (“Wheel Subassembly”)

Because the omniwheels that we have available are the older 4” and 3.25” versions, 86832D explored methods to improve the attachment of gears/sprockets since these 4” omniwheels do not have screw holes for direct attachment, risking damage to the low-strength shafts. A solution for this was inspired by an entry on the Purdue SigBots Wiki (“Wheel Subassembly”) which, although it itself is an article focused on the use of CAD, highlights a great alternative for the use of gears without risking damage to shafts.

Iterations on this design were made (as seen starting meeting entry 10/10/23) where four sets of spacers with screws were added.

260 **Prototyping: Suspension (Torsion Bar)**

In consideration of the robot having to go over the long barrier, George and I (Samuel) tried to design a torsion bar prototype that could make going over the barrier easier. This was eventually abandoned due to its excessive size and difficulty in balancing the drivetrain overall.

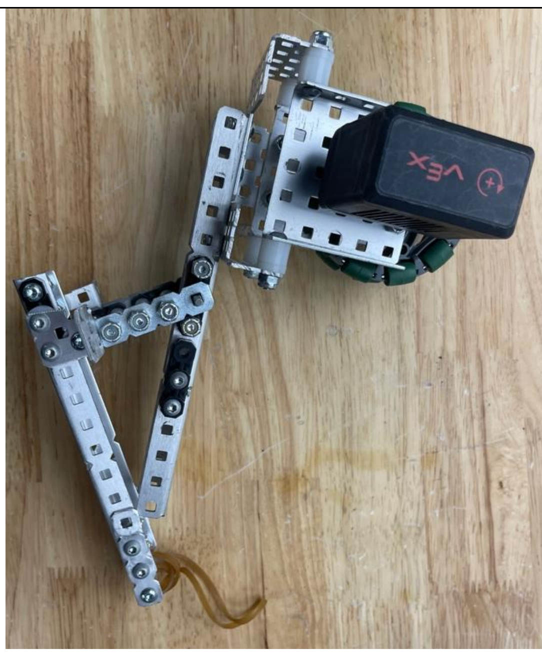


Fig. 6: Side view of torsion bar prototype

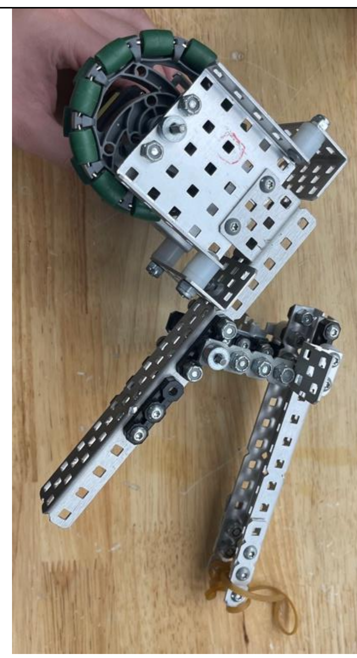


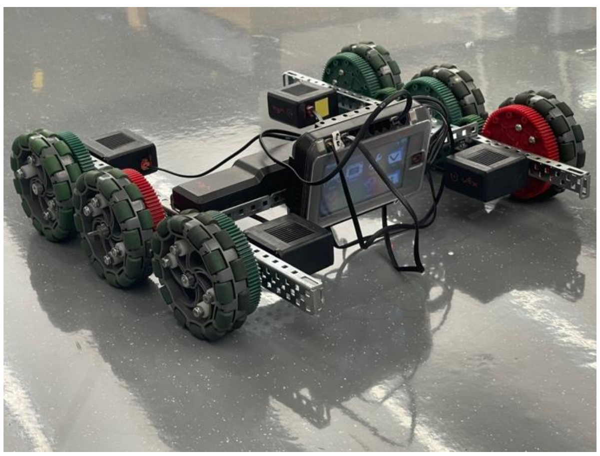
Fig. 7: Alternate side view of torsion bar

Meeting Entry 10/10/23

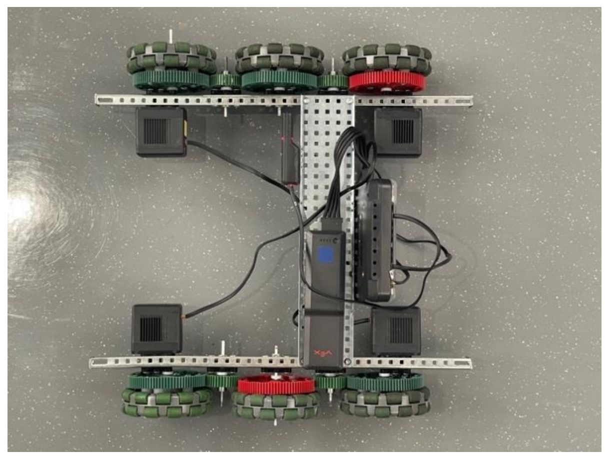
270 **<u>Prototyping: Drivetrain</u>**

Duration of session: 2 hrs. (William 1 hr.)				
Task	Team Members	Status	Deadlines	Notes
Finish geared 4-motor drive prototype	Samuel, George, Charlie	Complete	10/10/23	
Test code for prototype	William, George	Compete	10/12/23	

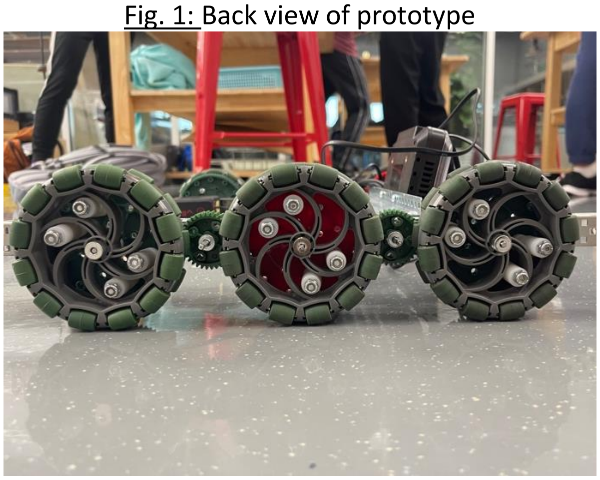
During the session, the team worked on finishing the geared 4-motor drivetrain prototype developed during the pre-season period. The motors use a combination 84T and 36T high strength gears with adapters for regular shafts along with 18:1 cartridges in four 11W motors to power the drivetrain.



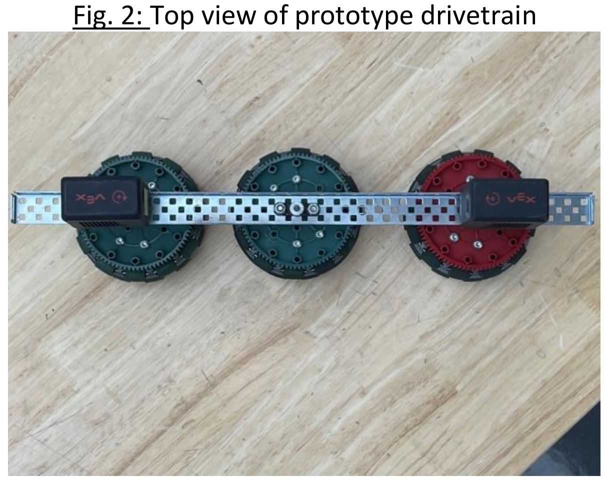
<u>Fig. 1:</u> Back view of prototype



<u>Fig. 2:</u> Top view of prototype drivetrain



<u>Fig. 3:</u> Side view of drivetrain



<u>Fig. 4:</u> Inside side view of drivetrain

Within this drivetrain, 3.25-inch omni wheels bolted onto 84T gears were used, made using white spacers and 2.5-inch screws. This makes a rigid connection between the wheel and the gear without solely relying on the normal shaft which may not effectively transfer torque between the wheels.

William, with George's assistance, coded the drivetrain. Since the driver code William developed in pre-season had issues with controller input controlling the correct motors, time was spent on debugging.

Code Snippet (PYTHON)

```
vexcode_brain_precision = 0
vexcode_console_precision = 0
vexcode_controller_1_precision = 0
DriveSpeed = 0
Stuff = 0

def Main():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision,
vexcode_controller_1_precision
    while True:
        if not controller_1.buttonL1.pressing() or not
controller_1.buttonL2.pressing() or not controller_1.buttonR1.pressing() or not
controller_1.buttonR2.pressing():
            Stop()
        if controller_1.axis2.position() > 10:
            if DriveSpeed >= 99:
                DriveSpeed = 99
            else:
                DriveSpeed = DriveSpeed + 2
                SetDriveSpeed()
                controller_1.screen.set_cursor(5, 12)
                controller_1.screen.print(DriveSpeed)
        if controller_1.axis2.position() < -10:
            if DriveSpeed <= 0:
                DriveSpeed = 0
            else:
                DriveSpeed = DriveSpeed + -2
                SetDriveSpeed()
                controller_1.screen.set_cursor(5, 12)
                controller_1.screen.print(DriveSpeed)
                vexcode_controller_1_precision = None
        wait(5, MSEC)
def Stop():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision,
vexcode_controller_1_precision
    FrontRIght.stop()
    RearRIght.stop()
    RearLeft.stop()
    FrontLeft.stop()
```

Continued from last page

```
def RightForward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    FrontRIght.spin(FORWARD)
    RearRIght.spin(FORWARD)

def LeftForward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    RearLeft.spin(FORWARD)
    FrontLeft.spin(FORWARD)

def RightBackward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    RearRIght.spin(REVERSE)
    FrontRIght.spin(REVERSE)

def LeftBackward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    RearLeft.spin(REVERSE)
    FrontLeft.spin(REVERSE)

def SetDriveSpeed():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    RearRIght.set_velocity(DriveSpeed, PERCENT)
    RearLeft.set_velocity(DriveSpeed, PERCENT)
    FrontRIght.set_velocity(DriveSpeed, PERCENT)
    FrontLeft.set_velocity(DriveSpeed, PERCENT)

def when_started1():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision, 
vexcode_controller_1_precision
    RearRIght.set_stopping(BRAKE)
    RearLeft.set_stopping(BRAKE)
    FrontRIght.set_stopping(BRAKE)
    FrontLeft.set_stopping(BRAKE)
    DriveSpeed = 50
    SetDriveSpeed()
    Main()

when_started1()
```

Fig. 5: Drivetrain Prototype Source Code (at the end of session)

In the code snippet above, initialization code is removed to save space and prevent unnecessary clutter.

In this meeting, William tested a variant of the tank drive for 86832D’s operator control system. Instead of utilizing the two joysticks in typical tank drive or arcade drive systems, the shoulder buttons (See diagram) will be used as a tank drive to control the drivetrain, with the right joystick also being used to control the set the speed of the drivetrain, similar to that of the throttle used within certain vehicles, especially how the throttle acts independently from the direction of the motors. William used a similar type of control system when participating in another team within the FIRST Robotics Competition, therefore this drive system will be tested.

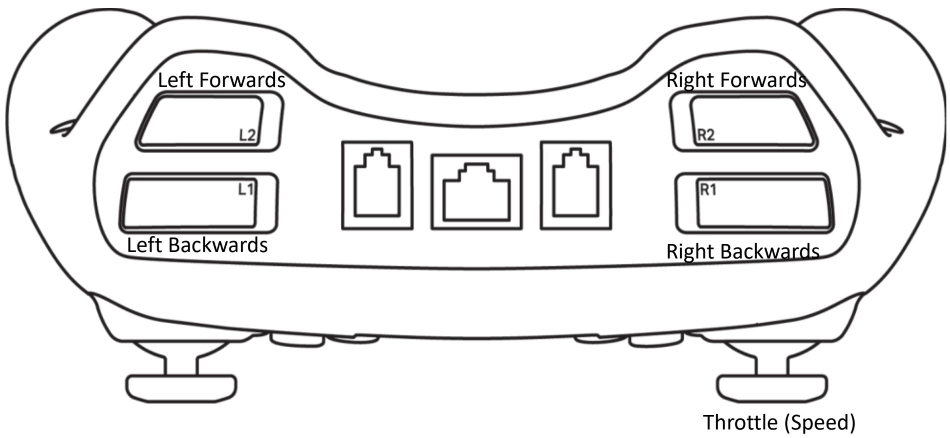


Fig. 6: Operator Control System Diagram (Tank Drive Variant)

A decision on the type of drive control will be decided in a later team meeting session using a decision matrix. Although this sort of decision will be decided by the entire team, a significant weightage of this decision will be given to William considering that he is the driver for this team. Some of the current considerations regarding this control system is that utilizing a throttle might make it difficult to switch between fast or slower speeds, which could be done easier by simply using the joysticks, and that this drive control system utilizes all four shoulder buttons instead of one of the two joysticks, reducing the buttons available to press.

The creation of this prototype drivetrain coincides the prototyping conducted by other teams within the organization. Members of 8632C, starting in the “Pre-Season” period worked on a creating a similar drivetrain using sprockets and chains instead of with gears. In the sessions next week, we plan on testing the two drivetrains side-by-side to determine the efficacy of both designs, and to decide as a team which design to go with.

Meeting Entry 10/12/23

Prototyping: Drivetrain

Duration of session: 1 hr.				
Task	Team Members	Status	Deadlines	Notes
Optimize Drivetrain Prototype	Samuel, George, Lucas, William	In Progress	10/19/23	Length of drivetrain is being adjusted after certain design considerations

Time was spent during our (shortened) session on Thursday to optimize the current prototype. Due to time limitations (our first tournament would be in December), we have decided to use our design prototype with improvements to the design.

Our current considerations in improving the design are as follows:

- The steel used on the side of the drivetrain seems to be too long, impairing the robot from climbing the Long Barrier (center bar) and reducing the area available for other mechanisms near the front/back beyond the drivetrain.
- The drivetrain (after adding necessary support structures) will be outside of size limitations, so we need to shorten it.

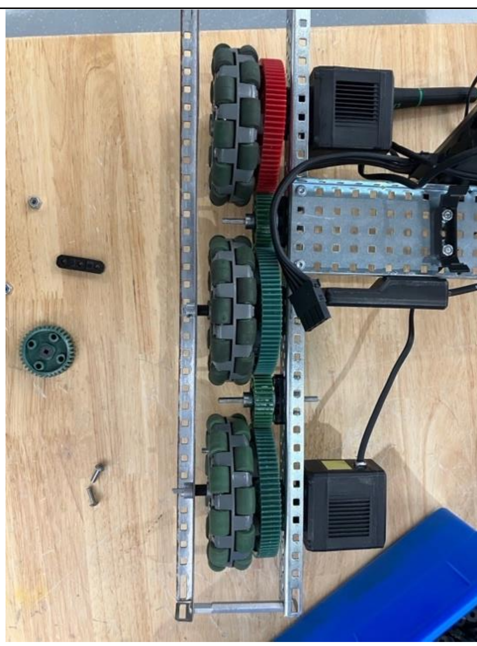


Fig. 1: Adding support structure to drivetrain

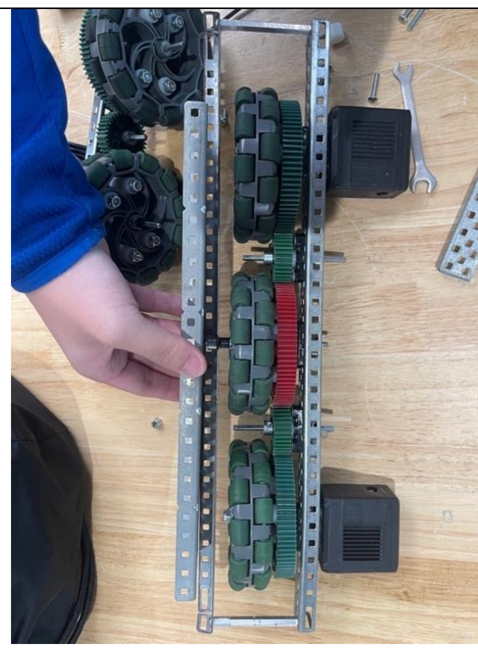


Fig. 2: Drivetrain compared to shortened length

As seen in Figure 2, we plan on shortening the steel by using 5-hole C-channels instead of the 2-hole L-channels currently being used because they come at a more suitable length and is more helpful in attaching the drivetrain (space lost can be minimized because the longer c-channels allow us to build directly on top, whereas standoffs may need to be used to build on top of the original design).

We also plan on shortening the bar in the middle (supporting the robot) by reducing the length of this piece of steel. This is done to comply with starting horizontal limits for the robot.

Due to the shortened session and the time taken in planning the improvements to this prototype design, we did not finish this task. However, we will finish it by the end of next week, since there is more time then.

Meeting Entry 10/17/23

330 Improving/Optimization: Drivetrain

Duration of session: 1 hr.				
Task	Team Members	Status	Deadlines	Notes
Optimize Drivetrain Prototype	George, William, Charlie, Samuel	In Progress	10/19/23	Changes to one (left) side complete, other side in progress

The focus on today's session was to finish improvement plans set out in last Thursday's session. At the start of the session, a sketch was made to illustrate the design of (a part) of a side of the base.

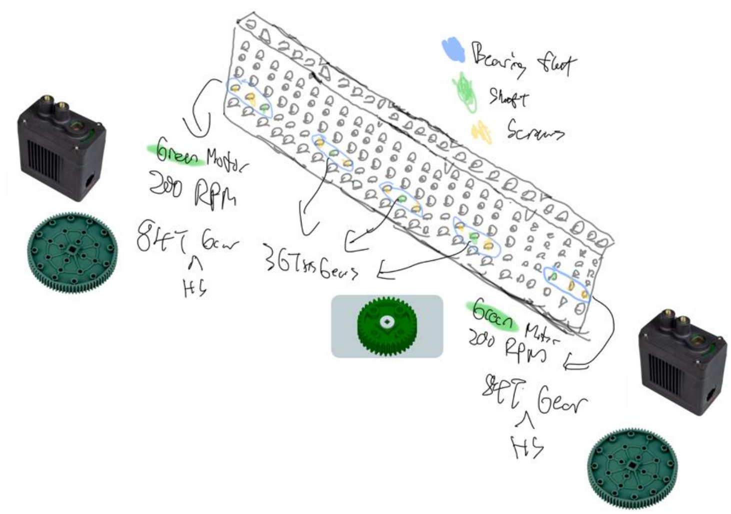


Fig. 1: Sketch made by Samuel

Fig. 2: Photo of progress

At the end of the session, we completed one side of the base and half of the other side of the base.

340 # Meeting Entry 10/19/23

Improving/Optimization: Drive Train

Duration of session: 1 hr.				
Task	Team Members	Status	Deadlines	Notes
Optimize Drivetrain Prototype	Samuel, William	Completed	10/19/23	Problems with wheel oscillation, design flaw?

In this short session with rather few members, time was spent toward fully completing the prototype base. After completing both sides of the base and adjusting the length of the steel used in the middle to connect the two sides, the base became functional.

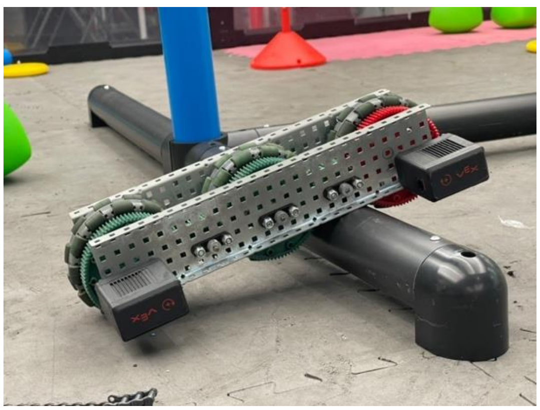


Fig. 1: Side of base placed on top of short barrier

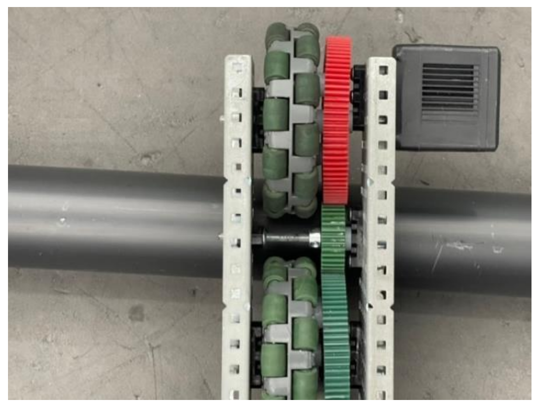


Fig. 2: Top view of base side on top of barrier

Fig. 3: (From Video) Prototype driving under goal, in possession of a Triball

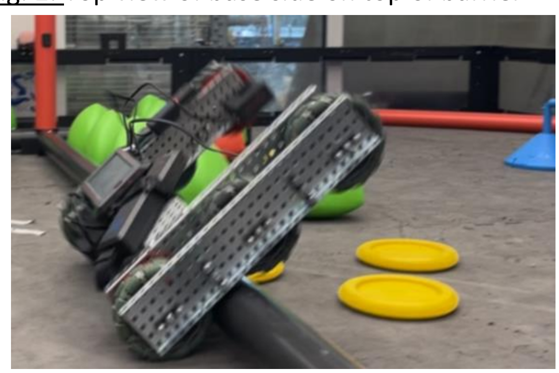


Fig. 4: (From Video) Prototype climbing the long barrier

Title: Meeting Entry 10/19/23	Author: Samuel Y	Date: Oct 20 2023

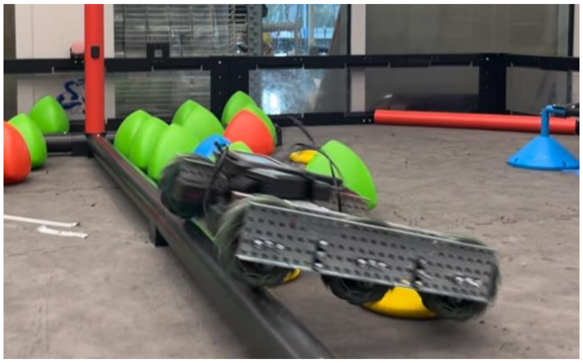


Fig. 5: (From Video) Prototype after climbing the long barrier

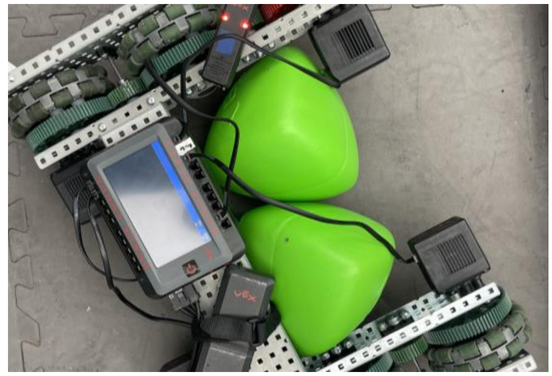


Fig. 6: Prototype (illegally) possessing two Triballs stuck within the chassis

Following a test drive, the prototype of the base is able to climb both barriers and score Triballs while possessing it into a goal. Although possible, it has limited plowing abilities due to it being concave on both sides, constituting possession and rending it difficult to push Triballs over the long barrier, for example.

Following informal (recorded) trials conducted on the field, observations were noted in the form of pros & cons:

Pros	Cons
- Ability to go over long barrier forwards & backwards - Balance of speed and torque - Ability to go inside goal and score Triballs - Ability to possess a Triball - Even distribution of power & torque among wheels - (Barely!) within starting size limitations	- Limited ability in pushing/plowing Triballs - Easy to violate rule SG7 on Possession with concave structure - Center of gravity is at back, rendering possible risk of flipping if more weight is added to robot - No ability for balancing on long/short barrier (no low elevation) - Wheel oscillation problem (see below)

When looking more closely at the video taken during the trials, one major concern with the prototype is that, when climbing the long barrier, the robot tips up at a sharp angle (as in Figure 4) before falling down once over two thirds of the robot has crossed the field. When more weight and mechanisms is added to the robot, this poses problems because this could result in a higher likelihood of tipping when climbing the long barrier. The extent of this problem depends on the mechanisms attached to the rest of the robot, but could be addressed by adjusting the weight distribution, moving more from the back to the front.

There seems to be a wheel oscillation problem with the drivetrain as well, where the robot jitters from side to side when controller input should make the robot go straight forward. This problem is still being troubleshooted, but we currently believe that this could be a result of the gears being warped/bent from the attachment method used between the gears and the wheels (see Pre-Season: Develop & Plan). This could also arise from problems within the motor itself, but more trialing is needed to determine its actual cause. After determining the cause, effort can then be taken to fix this issue.

Title: Meeting Entry 10/19/23	Author: Samuel Yao	Date: Oct 20 2023

With the completion of this prototype base, we will now focus on some other mechanisms of the robot. Current design considerations include considering a launching mechanism (to launch match loads), an intake mechanism and a low hang/high hang mechanism. Although there can be improvements made to the chassis, time constraints may result in design revisions to the base being delayed to the next design cycle, after the (possible) scrimmage in November and prior to the ISB scrimmage in December.

Develop and Plan: Driver Control System

After a brief conversation between myself (Samuel) and William, we discussed the viability of William's driver control system that utilizes the shoulder buttons. William recognized that his driving mode, although could be good for more intricate controlling, may not be the most viable due to hand cramping arising from the constant pressing/holding of the shoulder buttons. Additionally, considerations were also put forth to try different shoulder button configurations.

Since a team-wide decision has not been made yet and an analysis of choices have not yet been completed, this will be conducted once a more complete prototype of the entire robot is completed, where more testing can be undertaken.

Meeting Entry 10/21/23

Improving/Optimization: Drive Train

Duration of session: 2 hrs.				
Task	Team Members	Status	Deadlines	Notes
Add stiffeners on Drivetrain	William	Completed	10/21/23	Controller's shoulder buttons are possibly not designed for long-term pressing down
Start prototyping catapult system	Samuel	Completed	10/24/23	Used Red Motors

In today's session, I and Samuel attended and worked together for two hours; After he left the makerspace, I stayed for one more hour. Most of my time was spent analyzing and trying to fix the wheel problems.

First, the gears that attached to wheels are little not straight. Since we used the old gears, we are unable to find new gears to replace yet. But however, during today's driving test, bent gears did not affect my driving quality. In fact, when speed is set to 100%, the drive train seems to perfectly be moving straight. However, after a few seconds, it would stop for less than .5 seconds, and started slight steering. Because I already finished the stiffeners, wheels and gears should not be one reason that caused this problem; The motors have also been tested on other codes to prove that they function well. The only two possibilities are the shoulder button's intentional design, or the errors on the code.

Since the lack of alternative programs, I was unable to troubleshoot if my program caused such an issue. But I am certain that the previous design of using the shoulder buttons might be a bad idea, because not only do I get hand cramps, but the buttons also feel terrible, and sometimes I can't even visualize whether I should turn left or right now. Other than that, regarding the problem when moving, again I have no way of confirming if it is caused by the shoulder buttons and will need to wait for Lucas to make a program that uses the Joystick to verify this conjecture. I like using block-based programming via Vexcode V5, but I was forced to convert to Python after having issues with failed uploads. I cannot really use text-based, so the Drive Train program right now is a bit unreadable for myself.


Fig. 1: Drive Train with stiffeners added on

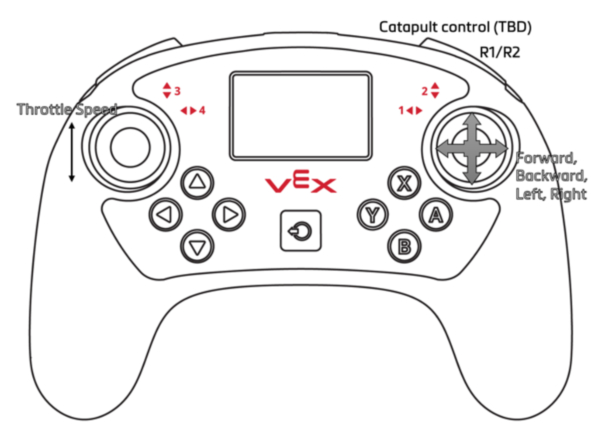


Fig. 2 New brief idea of control system

Title: Meeting Entry 10/21/23	Author: William Pan	Date: Oct 21 2023

Develop & Plan and Create & Improve: Catapult

(Samuel) I spent this session prototyping a catapult. My main goal in this process of prototyping is to conceptualize the design of what a catapult may look like and, through prototyping, understand some potential improvements that can be implemented in an elastic catapult that we intend on building.

For a catapult powered using elastic energy, the functionality of the catapult depends on a 'release' mechanism that, after rotating back the arm of the catapult which tensions the rubber bands, will release the gear which uses the elastic potential energy built up within the elastic bands to propel the catapult forward, launching the catapult.

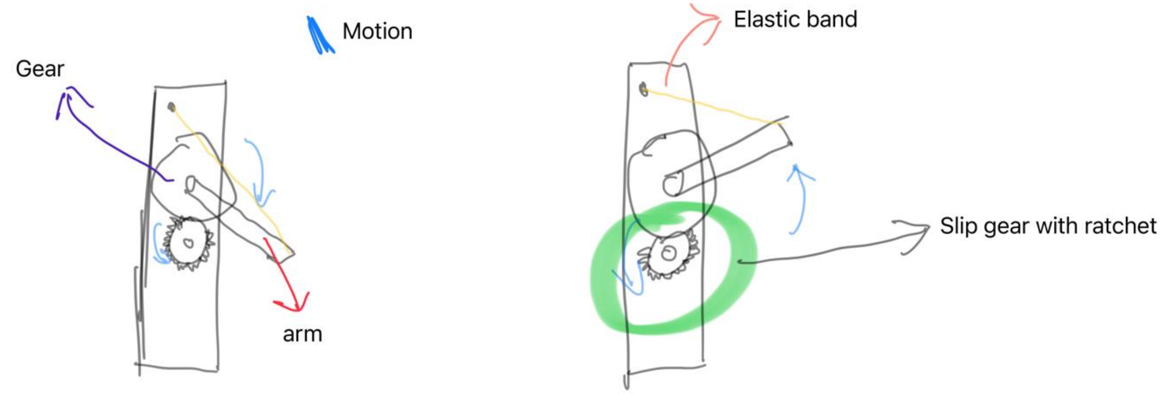
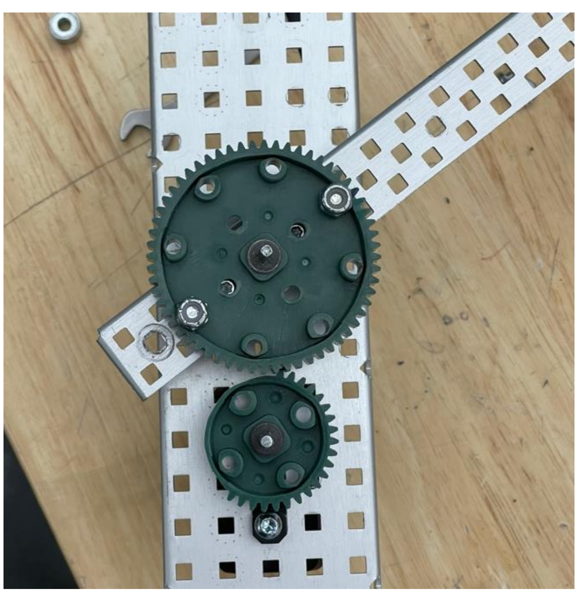


Fig. 3: (Very rough!) Sketch of core/essential catapult prototype by Samuel

After creating the physical prototype, we tested the prototype by running the red (36:1) motor connected to the catapult and using the force of our arm to simulate the release of the gear from the elastic energy of the elastic bands.



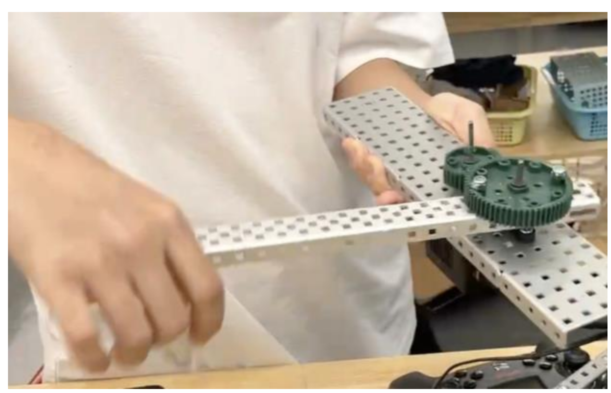


Fig. 5: (From video) The catapult in motion. Special thanks for Jun from 86832C for pretending to be an elastic band, as shown in the image.

Continued from last page

Fig. 4: Closeup of created catapult prototype

After the creation and testing of this prototype, some of the improvements that can be made include modifying the gear used from a 60T gear to a larger gear. We currently feel that the use of a smaller gear is not ideal due to the rotational speed of the larger gear not being slow enough, so the use of a larger gear such as a 72T or 84T gear will slow down the rotation of the gear. This will probably have to undergo different iterations for this to be effective.

Another consideration right now is the addition of a ratchet on the smaller gear. This is due to the fact that the torque of the motor with 36:1 motor cartridge might not be sufficient when compensating for the elastic force of the catapult. The use of a ratchet could prevent the slip gear from turning in the direction from the force of the elastic bands, allowing it to be pulled back before released once the slip gear rotates enough to revealed the shaved-off portion without the teeth of the gear.

During the simulated trial, we also realized that the RPM of the motor could be important to consider. Although the rotational speed of the arm in this prototype is not representative of the elastic forces present if the prototype actually used elastic bands, if the rotations-per-minute of the motor is set too high, it impact the effectiveness of match loading for the catapult and could result in damage to the gears (during the testing, the gears were skipping a bit).

This prototype was deconstructed at the end of the session.

Starting the session on October 24th, we will work on a second, full prototype of the catapult using the suggestions and improvements gained from this rough prototype.

Define & Inquire: Ratchets

Research into examples of ratcheting mechanisms

For the catapult that 86832D plans on implementing in our robot, we plan on using a ratcheting system for the robot in order for the use of a slip gear on the robot to be effective, since without a ratcheting system the catapult may not be effective since they need to counteract the rotational force inflicted by the elastic bands attached to the arm.

Since most of the members of 86832D do not have much experience in building a catapult, let alone a ratchet, I (Samuel) did research to see the implementation of ratchets in catapults. Because I am also unable to attend the work sessions this week, I will do background research in my own time and document this.

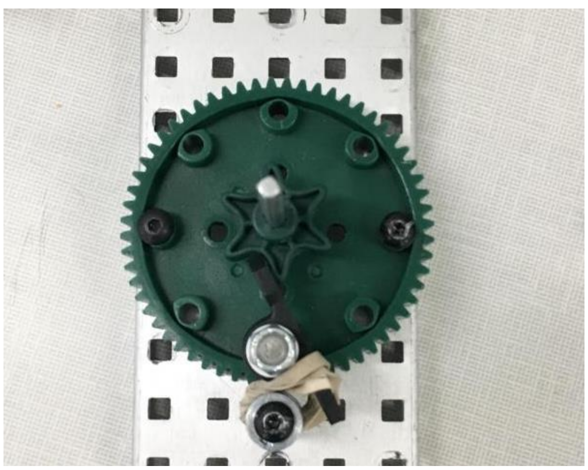


Fig 1: Ratchet design by MasterCole

Fig. 2: Ratchet design by Rogerson

Our main inspiration currently is a ratchet made using a pillow block bearing, sprocket attached to the shaft and the gear itself by MasterCole. We are considering imitating this design because the resources used to make this rachet are readily available for us and is relatively easier for us to make. The use of a standoff instead of having the elastic bands be wrapped around the metal make it easier for replacement or adjustment as necessary during a tournament. This design is also rather compact, which is more suitable for our purposes.

Another consideration we have right now include the ratchet design in Figure 2. This design is more compact, which could make it even better for us. The main concern, however, is whether a screw for the ratchet could fit onto the smaller slip gear that is used for the catapult. The use of these 12T pinion could make this possible due to its compactness, but a sprocket might be more effective to use since using a normal gear could result in slips in the rachet from the size of the teeth.

Works Cited

MasterCole. "Catapult Rachet Mechanism." *VEX Forum*, 7 Sept. 2018, www.vexforum.com/t/catapult-rachet-mechanism/49175/4?u=samuel_yao. Accessed 24 Oct. 2023.

Rogerson, Evan. "High Torque Ratchet." *VEX Forum*, 15 Nov. 2022, www.vexforum.com/t/

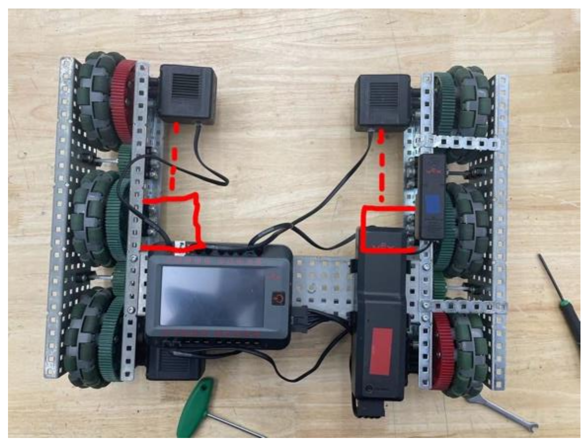
Title: Define & Inquire: Ratchets	Author: Samuel Y	Date: Oct 24 2023

high-torque-ratchet/109083?u=samuel_yao. Accessed 24 Oct. 2023.

Meeting Entry 10/24/23

Improving/Optimizing: Drive Train

Duration of session: 2 hrs.				
Task	Team Members	Status	Deadlines	Notes
Format Specified controls for drive	Lucas	Halfway Done	10/24/23	Before development of optimized program, I have formatted the needs for specific control by driver and prepare for pseudocode
Re-positioning Drivetrain motors to more inward	Lucas	In Progress	10/24/23	The motors are currently protruding from the robot's front edge, for better intake and more efficient building they need to be moved back.
Re-position Brain and Battery for future catapult builds	William	In Progress	10/26/23	The catapult requires more space, thus migrating the brain system is required.
	George		10/24/23	



In today’s session, we have begun the development stages of the robot and have made dramatic progress with identifying the needs of the driver and picking several valid control systems for him. We will continue transparent communication between the team and making sure that everyone is comfortable in their assigned roles of the final robot design and testing.

I came in the first hour of robotics, as the lead programmer I was needed to develop the main code for the robot: Intake, Drivetrain, catapult, or the final climbing stage. However, for the robot to reach its full potential, I needed to meet the needs of the driver and their habits or preferred controls - William’s style is quite unconventional, and was somewhat confusing; thus, the deadline to produce a fully optimized algorithm was delayed giving time for me to adapt and address possible changes that will further help him in doing accurate maneuvers. Me Samuel and George will discuss the potential possibilities for the control system and will finalize the list in another meeting. I believe that separate joysticks for acceleration and 360 degrees turn maneuvers will achieve better comfortability compared to the conventional driving styles of Arcade because of the straightforward nature of the control Drive.

Furthermore, I have begun to re-position the Motors of the drivetrain to more inward as currently they are protruding out of the robot’s front edge which is taking up much needed space for building. Because the drivetrain is quite complex and all connected, this will be a grueling process to re-position thus it will take some time.

Continued from last page

490 Develop/Plan: Catapult

(this entry has been lost, but content has been covered in future entries)

Meeting Entry 10/25/23

Driver Control Discussion

Duration of session: 0.5 hrs.				
Task	Team Members	Status	Deadlines	Notes
Decide on use of driver control system/ideate new systems	Samuel, George, Lucas, William,	Completed	10/25/23	

For this meeting, team 86832D discussed the current driver control systems and the strengths/weaknesses of each design currently conceptualized and considering.

Other than a normal arcade drive and normal tank drive, the two main driver control systems that are commonly used by ISB teams, we planned out and discussed the potential implementation of these control systems. Our design considerations were summed up into the design matrix below:

Decision Matrix: Driver Control System (Scale of 1-5, with 1 being lowest)

	Complexity		Controllability		Driver Preference	Total
	Button presses/joysticks	Driver control	Maneuverability	Accuracy		
Normal arcade	5	2	4	4	3	18
Normal tank	5	3	5	5	2	20
Single joystick arcade + throttle	4	4	4	4	5	21
Single joystick arcade + zeroed throttle	4	3.5	4	4	4	19.5
‘William’s drive’	2	5	3	5	4	19

The single joystick arcade + throttle drive system, conceptualized by George, is basically the combination of a single-stick arcade drive with the use of the other joystick for speed control. Similarly, the other system with a zeroed throttle, conceptualized by Lucas, is the same, except that the joystick will automatically be “zeroed” so that, at default, the speed value will always be 0. Although the main design idea to achieve this is the use of a robber band, forcing the joystick to its lowest value, we can think of alternative implementation ideas if we are to go through with this idea.

The main design considerations for our driver control system are the complexity and controllability of the system. For buttons presses/joysticks, we consider the number of buttons or joysticks necessary for the control system itself, with a lower usage of buttons more ideal because it simplifies the driver process. In terms of driver control, we rated it on the extent that this driver control system allows for the control of the drivetrain, with systems allowing for more control rated higher. ‘William’s drive’ has been referenced in previous notebook entries.

Continued from last page

For the controllability category, we also considered maneuverability and accuracy. For maneuverability, we considered the ease in which the driver can control and adjust the direction and speed of the drivetrain. For accuracy, we were considering the precision that the driver control system allowed for the positioning of the robot.

We included considerations for driver preference as well. Although this is almost entirely subjective to the opinion of the driver, it is important to consider because the effectiveness of a driver control system ultimately comes down to the preference of the driver. For this column we depended almost entirely on which drive control system William would prefer. William prefers drive systems that are more reminiscent of video game control systems, so the tank drive was rated lower due to the time it may take for William to understand and get used to it.

Ultimately, we have decided that the single joystick arcade + throttle idea seems to be the best in terms of what we have considered. However, we will still consider the use of a normal tank drive since, even though it may be difficult to master, it could be more effective in the long run.

Title: Meeting Entry 10/25/23	Author: Samuel Yao	Date: Oct 26 2023

Meeting Entry 10/26/23

Develop/Plan: Intake Mechanism

Duration of session: 1 hr..				
Task	Team Members	Status	Deadlines	Notes
Create first prototype for	Samuel, William	Completed	10/31/23	Basic design conceptualized, but needs to be refined/improved

Me and William focused on creating the **first prototype for the intake mechanism** that is planned for the robot. Although we had initially planned on continuing progress on the catapult started on the 24th of October, we felt that it was best to work on the robot when George was able to explain his current idea for the prototype. Due to scheduling conflicts, George was not able to attend this session, and did not manage to finish documentation for what he has done that session. Thus, I felt that it was most appropriate to start work on another mechanism instead.

Like the intake design conceptualized by team 86832B during the Spin Up competition season, we planned to combine **rubber bands and sprockets** to create a flexible intake that can accommodate for the uncertain orientation of the triball when picked up. The approach for this first prototype of the intake mechanism is to utilize a **top-down roller intake** where, using the roller placed near the top of the robot, the triball can be spun in and out of the robot. We decided to use **two 5.5W motors** for this mechanism because we felt that utilizing two 11W motors will be excessive for this sort of mechanism, and that the use of a single 11W motor could potentially be more inefficient compared to two 5.5W motors. Additionally, we felt that it was appropriate to use these 5.5W motors for the prototype because all of the teams within my organization have not used them yet, allowing my trial of those motors to determine the effectiveness of these new motors. We chose **not to use flex wheels** now because lacked the proper VersaHex hub adapters, and that most alternate attachment methods offered in the VEX Library includes parts that we have a shortage.

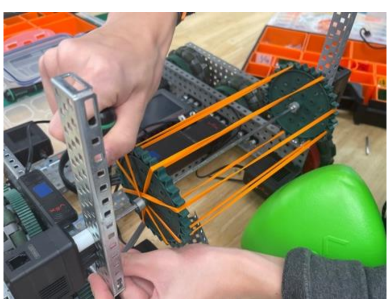


Fig. 1: William working on attaching the intake

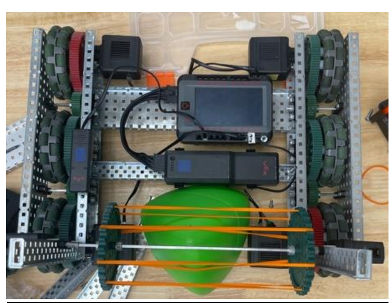


Fig. 2: Top view of the completed prototype

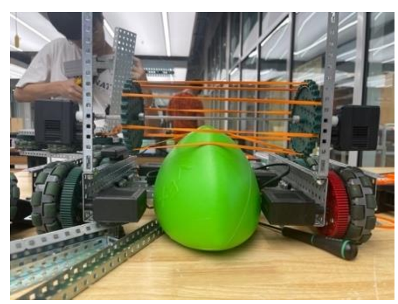


Fig. 3: Front view of the completed prototype

Due to time constraints, we were not able to test the mechanism running the motors, which I plan on doing next session. Some **potential improvements** noticed is that the mechanism could incorporate a **ramp** so that the triball is not dragged along the ground, creating frictional force and causing a risk of possession loss, the use of additional rollers. And, as Lucas has mentioned in a previous entry, the **motors at the front of the intake should be moved back** to allow for more triball space, which I plan on doing in the next meeting as well.

Title: Meeting entry 10/25/23	Author: Samuel Yao	Date: Oct 28 2023

Meeting Entry 10/30/23

Programming Edit Entry

Duration of session: 0.5 hrs.				
Task	Team Members	Status	Deadlines	Notes
Add intake functionality for intake	George, Samuel	Complete	10/31/23	Can be further optimized, see below. Buttons can be toggled.

Outside of regular robotics session times, me and George worked on developing code to allow the intake to work to test it next session. Although both of us are not primary programmers for the team, the programming required to make this work is very simple and is helpful for both of us to get accustomed to coding with VEX, for which both of us have very limited experience in doing.

CODE SNIPPET (PYTHON)

```
#Robot configuration code
Intake_motor_a = Motor(Ports.PORT19, GearSetting.RATIO_18_1, False)
Intake_motor_b = Motor(Ports.PORT20, GearSetting.RATIO_18_1, True)
Intake = MotorGroup(Intake_motor_a, Intake_motor_b)
#Variables
IntakeSpeed = 100
IntakeState = "Off"
ButtonHeld = "Off"

def Main():
    global IntakeState, ButtonHeld
    Intake.set_velocity(IntakeSpeed, PERCENT)
    while True:
        if controller_1.buttonUp.pressing():
                IntakeForward()
                while controller_1.buttonUp.pressing():
                    wait(1, MSEC)
                wait(6, MSEC)
                ButtonHeld = "CanReset"
                IntakeForward()

        if controller_1.buttonDown.pressing():
            while ButtonHeld == "On":
                pass
            else:
                IntakeBackward()
                while controller_1.buttonDown.pressing():
                    wait(1, MSEC)
                wait(6, MSEC)
                ButtonHeld = "CanReset"
                IntakeBackward()
        wait(5, MSEC)
```

Continued from last page

```
def LeftBackward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision,
vexcode_controller_1_precision
    RearLeft.spin(REVERSE)
    FrontLeft.spin(REVERSE)

def IntakeForward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision,
vexcode_controller_1_precision, IntakeSpeed, IntakeState, ButtonHeld
    if (IntakeState == "Off" or "Backward") and ButtonHeld == "Off":
        Intake.spin(FORWARD)
        IntakeState = "Forward"
        ButtonHeld = "On"
    elif IntakeState == "Forward" and ButtonHeld == "CanReset":
        Intake.stop()
        IntakeState = "Off"
        ButtonHeld = "Off"

def IntakeBackward():
    global DriveSpeed, Stuff, vexcode_brain_precision, vexcode_console_precision,
vexcode_controller_1_precision, IntakeSpeed, IntakeState, ButtonHeld
    if (IntakeState == "Off" or "Forward") and ButtonHeld == "Off":
        Intake.spin(REVERSE)
        IntakeState = "Backward"
        ButtonHeld = "On"
    elif IntakeState == "Backward" and ButtonHeld == "CanReset":
        Intake.stop()
        IntakeState = "Off"
        ButtonHeld = "Off"

def IntakeStop():
    Intake.stop()

def when_started1():
    Main()

when_started1()
```

Fig. 1: Truncated drivetrain code showing implementation of intake functionality

There were no other changes to the robot's code other than the additions above (see 10/10/23 meeting entry).

By using a toggle functionality to change the direction of the intake or to stop the intake, this can make it easier for the driver to adjust the intake as needed. Changes may be made depending on the effectiveness of the code, including the remapping of buttons if needed.

Meeting Entry 10/31/23

Duration of session: 1 hr. (William came before George & Samuel, partially asynchronous)				
Task	Team Members	Status	Deadlines	Notes
Complete first catapult prototype	George	In Progress	11/7/23	Revisions to be made (e.g., gear sizing), some progress was lost
Test intake, practice driving	William	Completed	10/31/23	
Modify first intake prototype	Samuel, Minjae	Completed	11/7/23	Minor changes in advance of second prototype

Drivetrain: Develop & Plan

William first started working on testing and driving the intake mechanism of the robot. Through driving on the field, we noticed that the intake has trouble holding the triball when reversing but is able to possess it without difficulty if the robot is turning or moving forwards. We currently suspect that this is a problem with the motor coasting instead of holding. Although this could be made better with code making the motor hold its position, a physical solution would be that when the triball is taken in, it is not dragging against the competition flooring. This also minimizes the friction of the triball when dragged, affecting the movement of the robot.

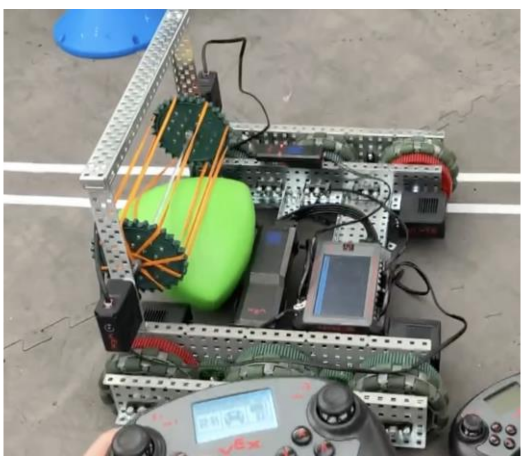


Fig. 1: (Video) Triball in process of intaking

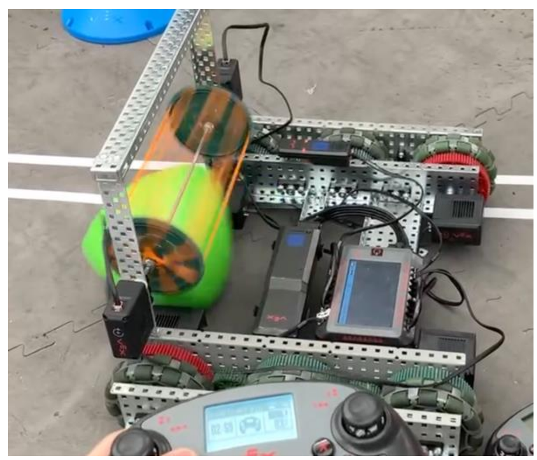
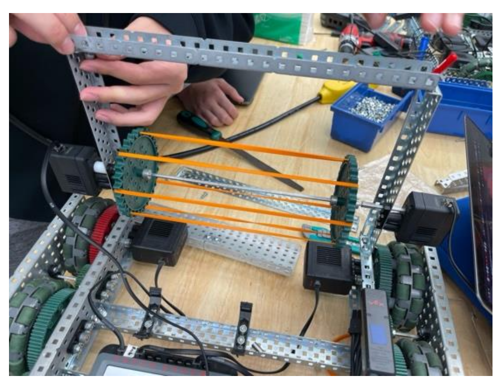


Fig. 2: (Video) Triball being shot out from intake

Fig. 3: Intake. Some minor changes were done, such as by correcting the length of the L-channel holding the intake on the top so that it is equal to the length below and trimming down the length of the shaft, making the shaft fit better between the two 5.5W motors.



Meeting Entry 11/07/23

Duration of session: 2 hrs. (Members came asynchronously, all but Samuel spent 1 hour total)				
Task	Team Members	Status	Deadlines	Notes
Consider improvements for prototype	Matthew, David	In progress	11/14/23	Will consider remaining mechanisms (climbing, catapult/output)
Move motor positioning	Samuel, William	Complete	11/7/23	Re-wiring necessary, to be done next session
Modify & optimize source code	Lucas	In Progress	11/9/23	

Create & Improve: First Prototype (Overall)

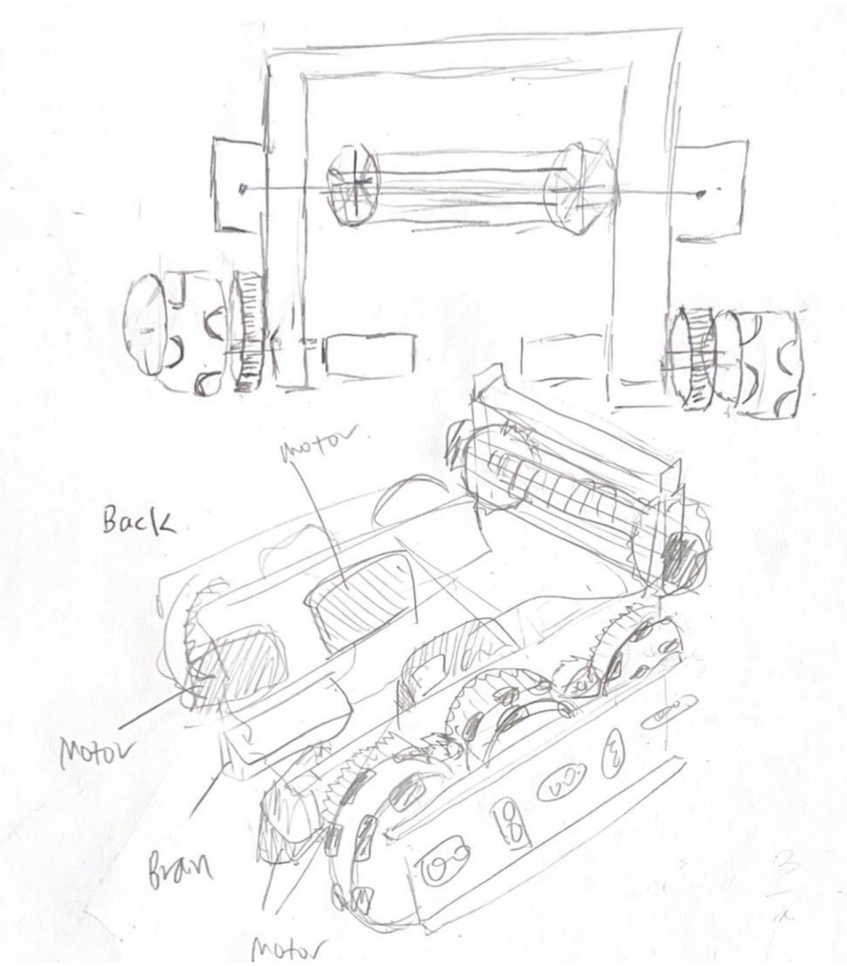


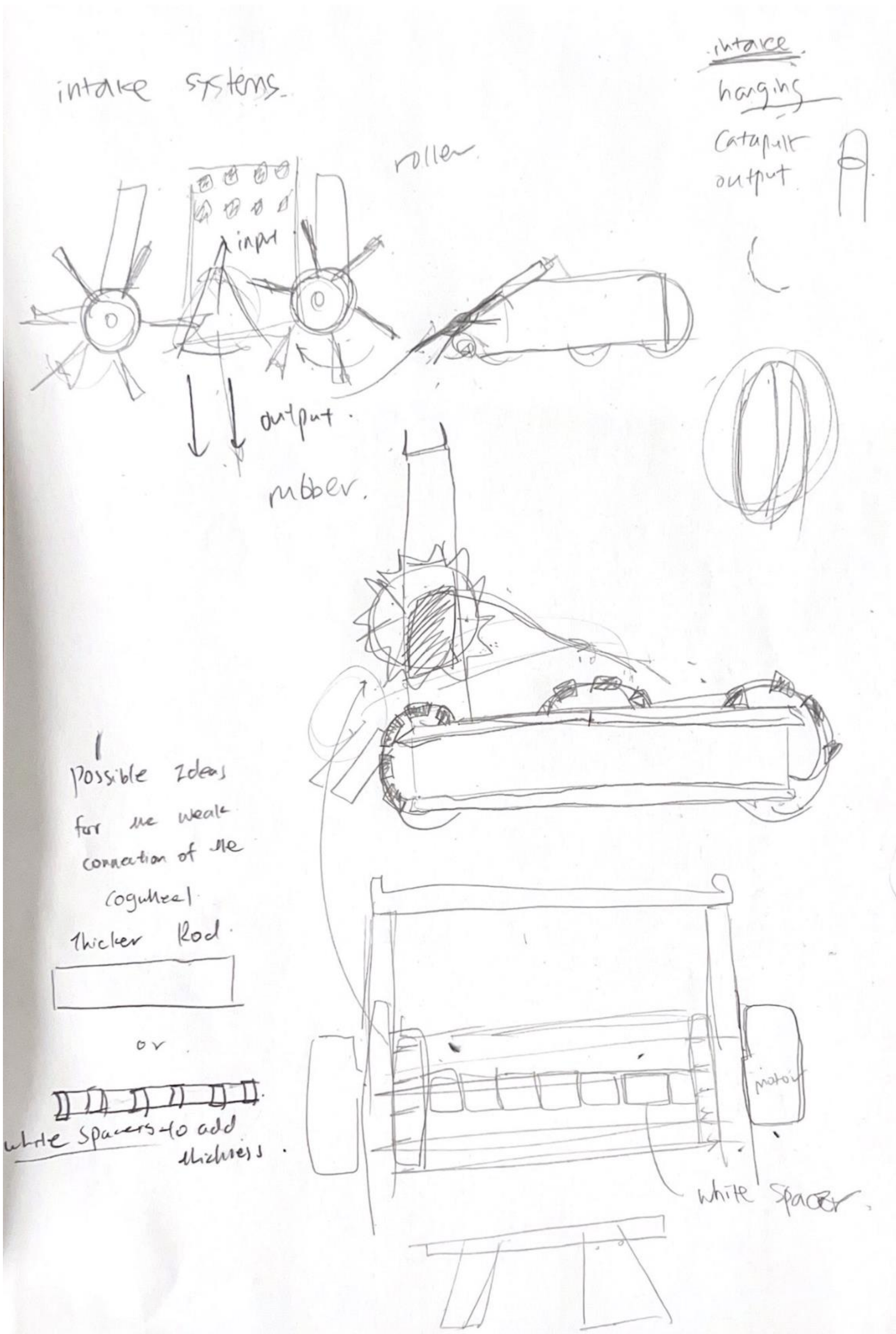
Fig. 1: Sketch of existing robot by David and Matthew

Reflecting on the current prototype, Matthew and David spent time sketching and prototyping potential ideas for improving the existing prototype of the robot as well as considering design ideas for a second prototype of the robot.

Time was spent primary on considering potential improvements to the intake mechanism, including changes that can be implemented in the short-term and long-term changes that could be done in future prototypes. (See Figure 2).

Some of the short-term ideas including using spacers or high strengths shafts to fix ongoing issues with the intake, such as the LS shaft bending and the roller generally being unstable. A long-term solution conceptualized include the use of two vertical rollers attached on the side, potentially using the conveyor belts with 'fins' attached or flex wheels. Inspiration for this was taken from existing designs online, such as 9499E in the Change Up season, 7282B during Tower Takeover and the default robot from the VEX V5 Competition Super Kit.

PHOTO BELOW



<u>Fig. 2:</u> Design conceptualization for intake by Matthew and David

Create & Improve: Drivetrain

Me and William spent the session changing the drivetrain so that the two motors are located near the back of the robot. The justification for this design improvement has been mentioned in previous entries. Although making this straightforward change should be relatively simple, it took a bit of time for me and William, highlighting that a possible design consideration could be improving the design of the drivetrain so that it can be more easily repairable.

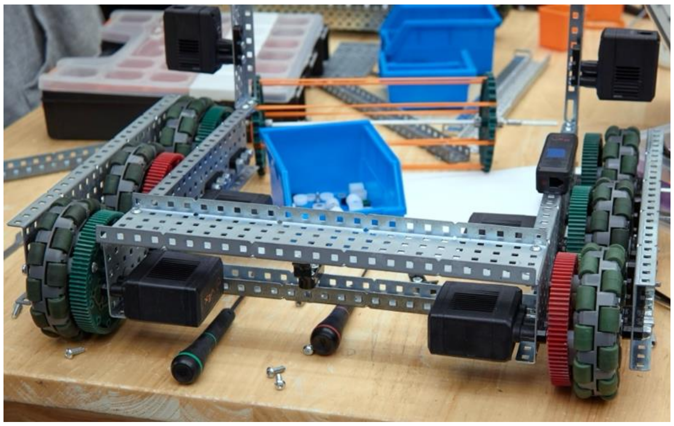


Fig. 3: Side view of changed drivetrain

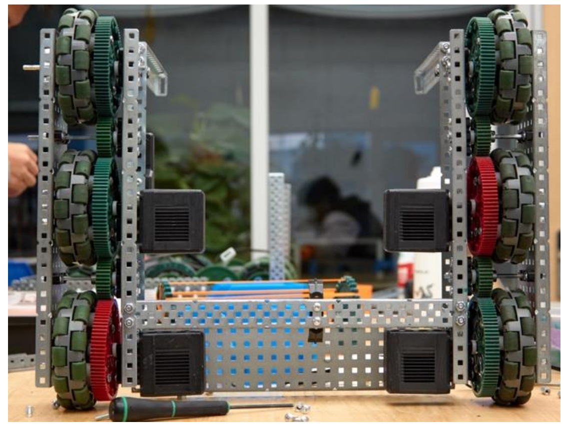


Fig. 4: Bottom view of drivetrain

The drivetrain will be tested in the next session before further changes can be undergone.

Create & Improve: Programming

Lucas spent time trying to reformat William's code by cleaning it up and also changing the functionalities from a modified tank drive (as known as 'William's drive') to some of the other decided drive control systems. The first one that Lucas will implement is an arcade drive and a tank drive (see meeting entry 10/25/23).

CODE SNIPPET (PYTHON) **CODE BELLOW**

Continued from last page

```
#Forward, off, reverse
class robot():
    def __init__(self, Dspeed=0, Ispeed=100)-> None:
        self.DriveSpeed = Dspeed
        self.IntakeSpeed = Ispeed
        self.IntakeState = False
        self.ButtonHeld = False

    #INERTIAL-----
    # def calibrate(self)-> None:
    #         # Calibrate the Drivetrain Inertial
    #         sleep(200, MSEC)
    #         brain.screen.print("Calibrating")
    #         brain.screen.next_row()
    #         brain.screen.print("Inertial")
```

```
#       drivetrain_inertial.calibrate()
    #       while drivetrain_inertial.is_calibrating():
    #           sleep(25, MSEC)
    #       brain.screen.clear_screen()
    #       brain.screen.set_cursor(1, 1)

    def Main(self):

        #INERTIAL------
        # drivetrain.turn_to_heading(90, DEGREES, wait=True)
        # drivetrain.drive(FORWARD)
        while True:
            intake.set_velocity(self.IntakeSpeed, PERCENT)
            whole.set_velocity(self.DriveSpeed, PERCENT)
```

Fig. 5: Truncated code by Lucas

Instead of using functions, Lucas is using classes. This will be elaborated on further in a future entry.

Meeting Entry 11/09/23

Duration of session: 1 hr.				
Task	Team Members	Status	Deadlines	Notes
Test & Redesign drivetrain prototype	Samuel, William, Minjae, Geoffrey	In progress	11/14/23	Will consider remaining mechanisms (climbing, catapult/output)
Modify & optimize source code, driver control	Lucas	Halfway Done	11/14/23	In process of debugging

Create & Improve: Drivetrain(Samuel) My team conducted tests on the robot by driving it. We noticed that the planned placement of the battery (at the bottom of the robot) would impair the robot from going over the center barrier. In addition, the current height of the intake would affect the robot's ability to go under the horizontal elevation bar which could affect the maneuverability of the robot. We first started making changes to the robot by adjusting the battery holders and reattaching the brain so that they are put at a more suitable location (back of the base).

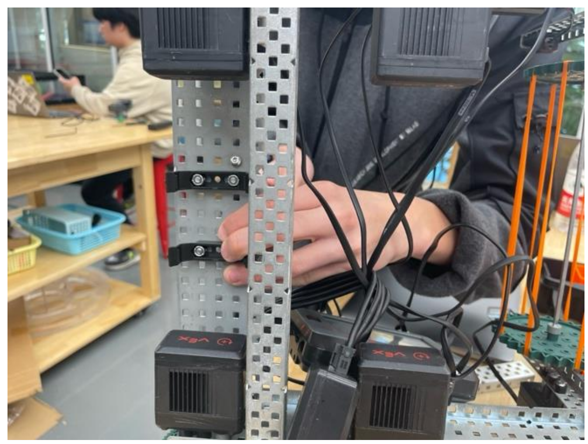


Fig. 1: Changed battery placement on robot

A consideration of doing this could be that the center of mass for the robot is shifted significantly back, which could cause the robot to tip over when climbing the barrier. The risk for this happening should be minimized since the robot tipping over could immobilize it for an entire round.

Because a lot of time have been spent optimizing/improving the drivetrain, more time will be spent to

Create & Improve: Programming

(Lucas) Considering I have only been introduced to the VEX5 API not long ago and with a variety of commitments required of me to attend it has been quite a struggle catching up with programming requirements and learning the different aspects of robotics. Nevertheless, I have tried optimizing as much of the code as possible, both the time and space complexity and the readability. Furthermore, I aim to learn as much robotics as possible before the end of the year, so during the last tournaments I will have implemented PIDs, and potentially AI.

When I first observed the initial code that was transformed to text from the block by Willliam, I noticed many unnecessary import global variables and contradictory conditions. The code was quite simple, first, define all running parts and add movement functions and another general main function that constantly runs with a while true loop detecting if there is any

Continued from last page

controller movement, and if there is, the conditional statement will then run its specific movement function. My first idea which I did in around 30minuets was just to simply clean up the code, in which the space and efficiency change was neglectable, but the code would be more readable; getting rid of any contradictory conditions and not needed variables or extra code. However, within the past few days I wondered what would happen if I used classes instead of simple functions.

The first felt curious of this when I learned about the autonomous phase of the competition, I thought that using a class and calling functions from it would make future addons simpler to modifier and add. Thus, I began to change William's code entirely, although not very good I believe I have made a start on the class and will update shortly.

Define & Inquire: Vision Sensor

In conjunction with an MS Robotics session on November 13th, I spent some time testing and exploring the vision sensors. This was done partially as a demonstration for the MS Robotics teams (which I am the helper/coach for one of them) but also as a self-exploration for the potential use of a vision sensor for 86832D's competition purpose.

The vision sensor was used rarely in all previous seasons, so I wanted to test it out for possible use within 86832D's robot this year. I spent the duration of the MS Robotics session testing the use of the vision sensor, including its internal sensing and vision utility as well as looking at the example code for the vision sensor within VEXcode.

The equipment I used to test the vision sensor included a standalone V5 brain with a V5 radio, battery, vision sensor, and controller to wirelessly connect the brain to the laptop. The vision sensor was simultaneously linked with the brain and the computer through a USB-A cable, where the vision utility program can be used to see the functionalities of the vision sensor and test out different color signatures.

Fig. 1: (messy) Layout for testing the vision sensor. The program running is the example code in VEXCode entitled "Detecting Objects (Vision)"

Fig. 2: The vision sensor in action (with vision utility on, detecting the color signatures of the game elements for VIQRC Full Volume)

I experimented with setting up color signatures for the three types of game elements in the 2023-2024 VIQRC game. I also looked at the VEXCode API ("vex:vision") for the vision sensor. In terms of its functionalities, it is able to return the count of detected objects that comply with a given criteria (e.g. signature) and it can also measure the XY coordinates, width and height of the objects. This can be helpful for the robot's autonomous period or autonomous skills to be able to detect triballs on the field without human input.

The underlying functionalities of the vision sensor, when paired with other given information of the robot, could find the distance between the robot and objects on the field, for example.

Works Cited

Meeting Entry 11/14/23

Duration of session: 2 hrs				
Task	Team Members	Status	Deadlines	Notes
Test drivetrain & make improvements	Samuel, William	Complete	11/14	
Finish catapult prototype	George, Charlie	In Progress	11/7	(Due to management/time issues, documentation for this is combined in meeting entry 11/21/23)
Research vision sensor	David	Complete	11/14	Work will be documented in a separate Define & Inquire entry, not this meeting entry

Create & Improve: Drivetrain

(Samuel) After further testing at the start of the session, I also noticed that the robot had difficulties possessing a triball while going over the long barrier. Although this depends on the placement of the triball (sometimes the robot COULD possess and go over the long barrier), in the majority of times the triball would wedge the robot, preventing the robot from going over the long barrier. After further trialing I noticed that the robot could go over the long barrier while in possession when the triball is oriented with its edge facing the long barrier.

While testing out scoring with the robot, I noticed that the robot was capable of plowing triballs using the back of the robot with its flat surface and with the back of the robot being lower than the height of the goal. Intaking and outputting, however, seemed to be difficult if the triball is oriented away from the intake near the back of the robot.

This issue is mainly caused by the triball contacting the field elements or field tiles while the robot is moving. To solve this issue, the intake can be modified so that when a triball is input then it is held with it not touching the field tiles.

(William) We've got some issues. The vertical bars that the Triball intake mech. were too high, and at such a height the vehicle cannot even pass the sides. I suppose that it's also not going to pass the size check because it's too high. In order to fix this, I and Samuel did some reposition. Little troubles occur during the progress of reposition and material change but in the end most of them are solved. We also changed the bar from steel into aluminum, and that is because steels are way too heavy, if in future when we finished the full set of mech. attached to our robot, while most of the bars are made of steel, it's not possible to make it cross the middle line, the motors not going to have that torque; also the drive speed can be very slow, and that may waste time. D: The left-over issues is that intake mechs are now quite unstable, due to the disassemble of the top stiffener bar - I mean, it's kind of wobbly, that's no good. Next meeting's plan is to make that top stiffener, and make sure nothing is unstable. I don't want to see something drop off the robot when I'm driving, that would be very embarrassing. :-(

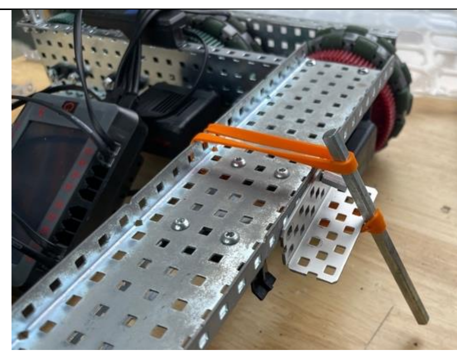
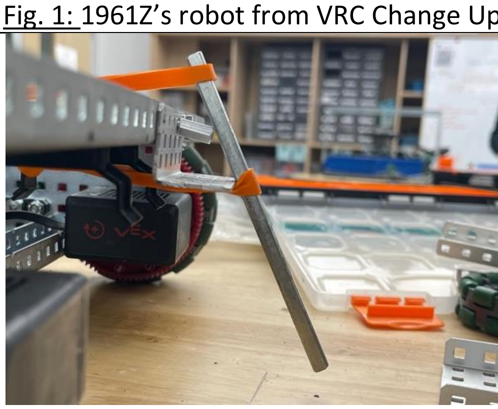
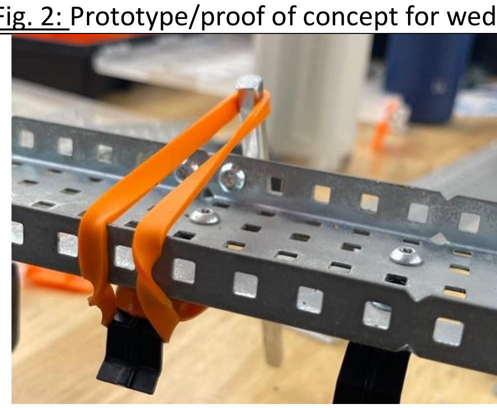
Meeting Entry 11/21/23

Duration of session: 1 hr.				
Task	Team Members	Status	Deadlines	Notes
Conceptualize drivetrain improvements (defense)	Samuel, Minjae, William	Complete	11/14	
Finish catapult prototype	George, Charlie	Halfway Done	11/7	Attachment to robot base/testing base not yet done
Conceptualize intake improvements	David, Matthew	Complete	11/14	

Define & Inquire: Drivetrain Defensive Mechanisms

During this session, we spent time thinking about ways that the drivetrain could be improved to have more functionality or be more effective. One of the main considerations we have currently is the use of defensive mechanisms to give the robot an advantage when being pushed.

Our current consideration involves the use of HS shafts spaced apart to form the wedge. The alternative method for a unibody wedge would be difficult for our team to implement due to the material restrictions of VRC preventing us from making a similar 3D-printed wedge, not including the potential issues it could have when crossing over the long barrier. This idea may still be explored later on, but its implementation would be more difficult than that with the HS shafts.

 Fig. 1: 1961Z's robot from VRC Change Up[i]	 Fig. 2: Prototype/proof of concept for wedge
 Fig. 3: Side view, prototype	 Fig. 4: Back view, prototype

Title: Meeting Entry 11/14/23	Author: Samuel Yao	Date: Nov 25 2023

The use of a wedge could give a competitive advantage specifically when the robot is driving head-on with another robot or driving into the side of another robot by going underneath the base/wheels of the robot and exploit vulnerabilities of weaker chassis designs by partially lifting up the robot from one side (e.g. between the wheels) or by lifting up the front/back portion of a robot. This would cause the offensive robot to potentially lose traction because their wheels are lifted up in the air, giving our robot more of an advantage.

This implementation would need to be improved for competition use. The HS shafts are currently at a steeper angle, making it more difficult for this wedge to be effective in going under other robots. The positioning of the shafts is impractical, since it is banded to battery holders and is vulnerable to the elastic band slipping. The top could be banded with more loops in the elastic band, since when force is applied to the bottom portion of the HS shaft (as would happen in competition use), the top portion would lose tensions, possibly resulting in the top elastic band to become loose and make the mechanism incapable of completing its purpose.

In addition, the considered implementation of HS shafts as a wedge could be problematic. Because the back of the robot was originally considered mainly for the plowing of triballs. The inclusion of these HS shafts could potentially result in the plowing of multiple triballs at once to be considered as possession because of the concave portions are formed. This could be alleviated by making the wedge retractable (potentially with the use of pneumatics) or by designing the entire back portion of the base so that it is convex with the addition of the wedge.

This idea of using wedges was inspired by the BLRS Wiki entry “Defensive Mechanisms.”[i]

Create & Improve: Catapult

Throughout the past few sessions, George and Charlie were focused almost exclusively on designing the catapult, our shooting mechanism for match loading and launching triballs.

Although 86832D initially intended on making a catapult with a slip gear, we decided to test out the use of the use of a connecting rod catapult[ii]. Using this sort of design would eliminate the need for the use of slip gear and allow for a greater rate at which triballs can be launched. This does come at the cost of potentially limiting the power the catapults are launched, since the use of these gears in this way could limit the elastic force of the elastic bands (that is the primary force in launching the triballs) through the friction of the gears not being free spinning as if a slip gear was used. Even if this does affect the power, it is necessary for the team to be able to launch multiple triballs in a short duration of time, minimizing potential time waste spent at the corner of the field match loading will allow for more time during a match in completing other tasks, mainly scoring triballs, defending my own team’s goal or endgame climbing.

The catapult we built was more specifically inspired from 7682’s ‘Choo-choo’ catapult designed for VRC Nothing but Net[iii]. The motion of the catapult and the connecting arms resemble the motion of traditional locomotive wheels, with the two-aluminum c-channels (held down by me) screwed into the body of the robot, and the arm (the c-channel in the center with gears attached) being the actual portion of the catapult that propels the triball forward. The one-hole-wide plates (that will be cut short) are to connect the arm to the body of the robot, that being the two c-channels.

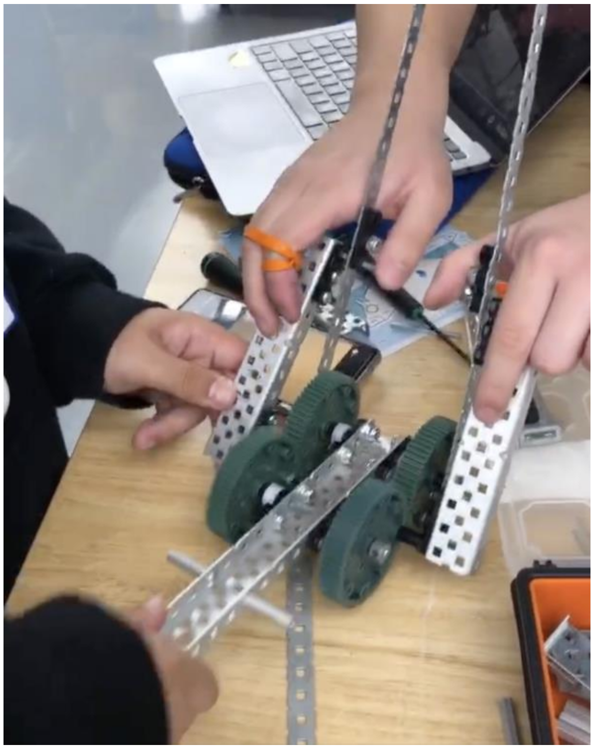


Fig. 5: (Video) Catapult in “down” position

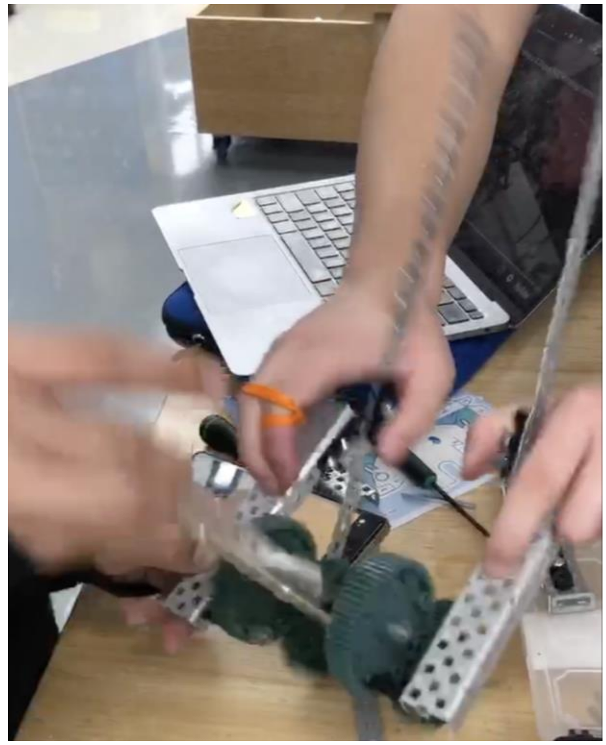


Fig. 6: (Video) Catapult moving to “up” position

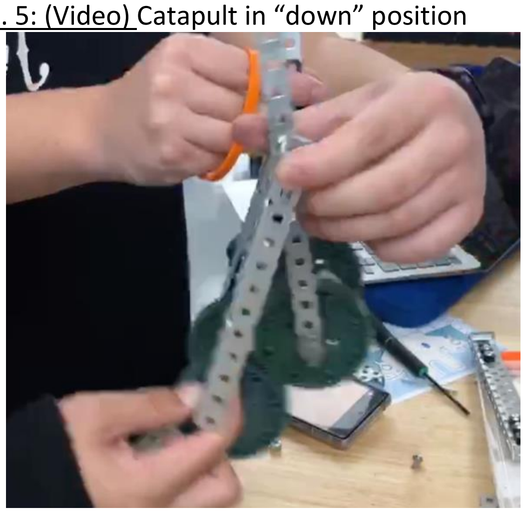


Fig. 7 (Video): Catapult in “down” position, side

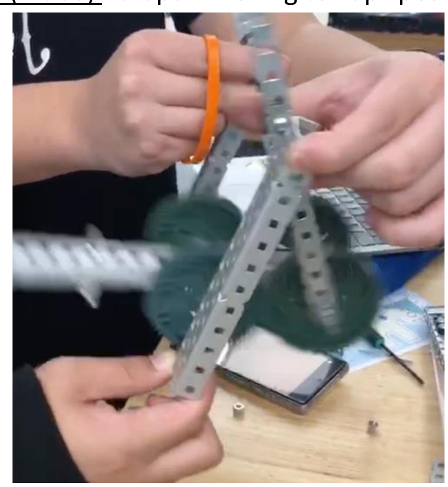


Fig. 8 (Video): Catapult in “up” position, side

The main design consideration is the attachment of the catapult to the robot as well as the power of the catapult. For it to be effective in launching triballs, the catapult needs to run at a high enough RPM so that the applied force is great enough for it to reach the other side of the field. This will be tested and looked into in future sessions.

Develop & Plan: Intake

Matthew and David spent time to further look into potential improvements to the

Meeting Entry 11/23/23

Duration of session: 1.5 hrs.				
Task	Team Members	Status	Deadlines	Notes
Improve Intake System Shafts	William	Complete	11/23	Axis was too long caused stretch on 2 bars.
Finish catapult prototype	George, Charlie	Halfway Done	11/30	
Finish brief code	Lucas	Halfway Done	11/30	Almost there...

850 [William] Today, Samuel's busy and He attended for only few minutes. George and Charlie tried to find more ideas and continued making the catapult. Since the code wasn't finished, I didn't have too much to do, thus I fixed the issue with the axis length, now they are much better. Lucas continues to code, and he has made some process. The test on the driver train was a success and he is halfway done with his program. For the next meeting, I'll get some alcohol and clean the gears and wheels since they are such dirty and have influenced my driving experience.

Title: Meeting Entry 11/23/23	Author: William Pan	Date: Nov 24 2023

Late Nov - Dec Team Organization

With the upcoming ISB Robotics Scrimmage organized on the 9th of December, the (confirmed) sessions remaining for 86832D to work on the robot until the end of December are on as follows on the calendar below, with associated deadlines and tasks for each day. There may, possibly, be more time to work on the robot if the club supervisors are available during the weekends.

Confirmed Sessions Calendar (Remaining hours: Approx. 10 / 5 sessions confirmed)

Date (and approx. hours)	Tasks due/Deadlines	Notes
Sun. November 26, 5 hrs.	- Catapult attachment to base - Intake improvements - Hardware troubleshooting	Impromptu weekend session, limited attendance
Tue. November 28, 2 hrs.	- Finalization of intake - Catapult optimization - Completion of robot hardware (base, catapult, intake)	Limited attendance, IB Music Recital
Thu. November 30, 1 hr.	- Completion of robot code - Code testing	Limited attendance, HS Strings Concert
Tue. December 5, 2 hrs.	- Robot test drive - Autonomous code completion	Regular attendance (Lucas may be absent, so coding may be problematic)
Thu. December 7, 1 hr.	- Robot test drive - Last-minute changes	Regular attendance

Due to scheduling issues, 86832D has been mostly working asynchronously with team members coming on different days at different times. As a result, cooperation within the team beyond our 'groups' was considerably difficult. There were, in my opinion, lapses of communication between team members which resulted in potential time wasted during sessions when there was not effective leadership from myself, the lead designer or lead programmer because of our absence.

The attendance for robotics sessions from 86832D (since the 19th of October, attendance was not recorded previously), in hours, are shown. Extra hours represent the approximate time spent on the robot outside of regular robotics sessions, such as during study leaves or study halls for classes.

	19-Oct	21-Oct	24-Oct	26-Oct	31-Oct	7-Nov	9-Nov	14-Nov	21-Nov	23-Nov	Total	Extra
Charlie	0	0	1	0	0	0	0	1	1	0	3	
David	0	0	2	0	1	1	0	1	1	0	6	
George	0	0	1	0	1	0	0	1	1	1	5	3
Lucas	0	0	1	0	0	1	1	0	0	1	4	3
Matthew	0	0	1	0	0	1	0	0	1	0	3	
Minjae	0	0	0	0	1	0	0	0	1	0	2	
Samuel	1	2	0	1	1	1.5	1	1	1	0.1	9.6	3
William	1	3	1	1	1	1	1	0.5	0.5	1	11	

Continued from last page

In addition to the varying skill and commitment levels from the various team members, I will try and schedule a full team meeting prior to the ISB Scrimmage to ensure that tasks and priorities are aligned properly, ensuring that team 86832D will be able to use our time the most effectively.

Some of our members have committed rather little to the team thus far due to schedule conflicts. I will try and talk to these members to see if they are able to commit more to the team if they have more time and availability.

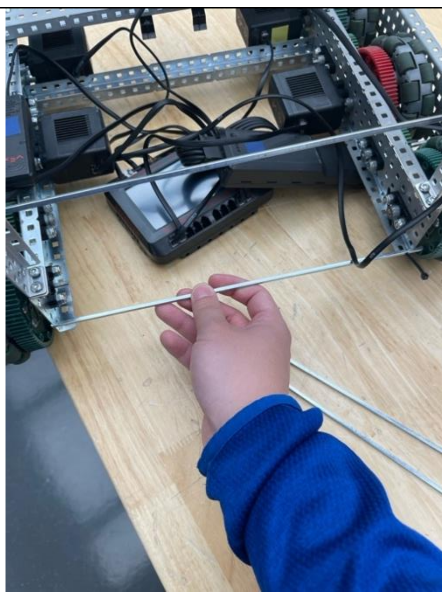
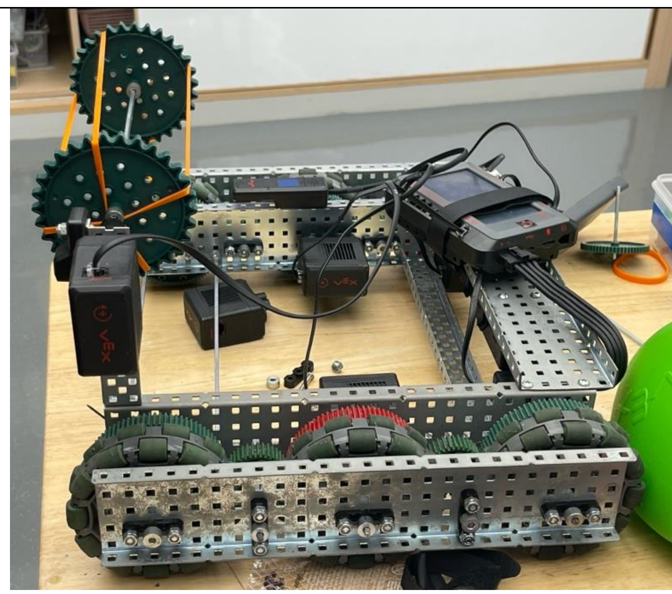
Meeting Entry 11/26/23

Duration of session: 6 hrs. William, 4 hrs. Samuel				
Task	Team Members	Status	Deadlines	Notes
Troubleshoot drivetrain drifting	William, Samuel	Complete	11/26	
Intake/base improvements	Samuel, William	Halfway Done	11/26	
Code arcade drive	William	Complete	11/26	Rework with the old code so it works with joystick.

In this session (with myself and William) we worked on some of the various objectives as outlined in the team organization plan for the remaining sessions up until the ISB Scrimmage.

Create & Improve: Intake Mechanism

Inspired by the use of HS Shafts in the base as structural support as shown in team 21417A's robot in Over Under[iv], I decided to try the use of normal shafts in the base as a short-term solution to the triball touching the ground when possessed within the robot.

 Fig. 1: Comparison of the size with 12" drive shaft	 Fig. 2: Attachment of two LS Shafts on base (sorry for bad image, forgot to take more!)

910 I intended on using HS Shafts in the same manner as 21417A, but because we did not any HS shafts that are sufficiently long for our purposes, we decided to use LS shafts since the focus of this design is to engineer a short-term solution to hold the triball. I will improve this design in a future prototype by designing a more permanent solution to this current issue.

The height of the intake was increased to accommodate for the LS shafts so that the Triball can fit through, but this resulted in the intake being too high to contact any of the triballs from the match floor.

The current intake will be lowered slightly to accommodate for the height difference, but in the next prototype we will redesign the intake to ensure that the intake can work sufficiently.

The mechanism was removed to consider redesigning to the intake mechanism and to make structural fixes to the base of the robot.

920 **Create & Improve: Drivetrain**

When testing out driving during this session, William noticed that there was drifting issues with the robot, resulting in the robot drifting to the right when the controller input should result in it going straight ahead. After further trialing by William, we figured that this problem was more apparent with the motor spinning at a significant velocity (at 100+ rpm). This problem could be because that one of the unpowered shafts in the drivetrain was not free spinning even when the gear is, but we figured out that the issue was also with a motor in the drivetrain (Front Left motor) disconnecting a lot, resulting in the drivetrain having issues travelling in a straight line.

I took some time in retightening some of the shaft collars on the drivetrain and replacing the inserts of the free-spinning gear which apparently had free-spinning inserts instead of regular inserts.

930 **Create & Improve: Programming**

[W.P.] I have been waiting for the code for like a few weeks, but right now there is no more time. So, with some nice block-based coding, I improved my old block-based code, and it fully functioned with the joystick. It can even change the speed as the joystick pushes more forward or backward. It’s like a CVT, such cool. I mean, the speed change was very accurate. However, due to driving train issues, we were not able to fully fix the drift bug, but it is stable when the speed was low.

Meeting Entry 11/28/23

Duration of session: 1 hr. (Asynchronous, members coming for varying durations)				
Task	Team Members	Status	Deadlines	Notes
Repair drivetrain	David, Matthew, Samuel	Complete	11/26	
Code testing & Drivetrain troubleshooting	William, Lucas	Halfway Done	11/26	
Catapult prototyping & testing	Samuel	Halfway Done	11/26	Motor needs to be repositioned, switch to red.

Create & Improve: Drivetrain

Create and Improve Code and DriveTrain

(Lucas) As the team has been remarkably busy with academics, we have not managed to spend enough time on the inner programming of the robot, implementing PIDs, and creating a fully sophisticated program. However, through weekly sessions where we continue to test William's code, we have concluded several issues that may be beneficial to the overall simplicity of the driver.

The most obvious drivetrain error that we first noticed is driving inconsistency. Even with logically correct programming, the drivetrain groups left, and right were not running at the same place, thus the robot randomly turns and moves in unpredictable motion. I suspect that this error is because of the wait function in William's block programming which may stack overtime and disrupt the normal while True loop that achieves polling of which buttons are pressed. I came to this conclusion because my text programming works, and the wait function set was much lower in milliseconds than Williams. This was later proven as my code when driving straight does not have that issue.

I have done much work with text programming throughout the time given to me, familiarizing myself with the Vex API in Visual Studio code as I wanted to explore the possibilities of coding in Python. Although normal button pressing and detection with motor groups running and intakes, there is still much error in my joystick programming which still most of the time do not obey my intended logic. Nevertheless, I will continue to update and work on my program with new additions of PIDs and potentially Machine Learning if I have time in preparation for TIS which is an upcoming competition.

Continued from last page

Create & Improve: Catapult Mechanism

Meeting Entry 11/30/23

Duration of session: 1 hr. (Asynchronous, members coming for varying durations)				
Task	Team Members	Status	Deadlines	Notes
Code testing & Drivetrain troubleshooting	William, Lucas	Halfway Done	11/26	
Catapult prototyping & testing	Samuel	Complete	11/26	Catapult basic functionality achieved, optimization and redesign needed.

Create & Improve: Catapult

After repositioning the motor by adding an additional gear on the catapult, making enough space for the motor and wire to be placed, I attached the existing catapult prototype to the robot to test its implementation on the robot. The attachment of the catapult to the robot isn't ideal, since I only used standoffs to keep the catapult attached to the body of the robot. Regardless, basic functionality was achieved, with the robot capable of launching the triball a short distance when tested outside of the

970 field. field. 1

Meeting Entry 12/02/23

Duration of session: 6 hrs.				
Task	Team Members	Status	Deadlines	Notes
Code testing & Drivetrain troubleshooting	William	Complete	12/3	Autonomous Code to be completed.
Catapult Implementation	Samuel, William	Complete	12/3	Wow!
Redesign & implementation of Intake	Samuel, William	Complete	12/3	Ready for scrimmage, but can still improve

Summary, Create & Improve: Programming

[William] Today was so productive that we actually finished all the parts we needed for a competition. The critical parts include the Drive Train, Catapult and the intake. Even though, in the end, one motor of the drive train had a minor error, they still function well during the testing part.

First, the drive train was nearly perfect with steel case and appropriate brain and battery position. We could have a race with ONLY the drive train BUT it does not perform the perfect effectiveness. With careful changes, the drive train now gets more stable, but we still need to clean the gears and wheels since they are very dirty and have contained a bunch of dirt.

Second, Catapult now functions well. Even though the strength was not perfect but both I and Samuel were satisfied with its performance – if the loader operates well, the catapult will be able to shoot over the middle barrier, and that is exactly what we want. Also, due to strong force, the whole robot now bounces off the ground shortly after the catapult releases. However, that will not be a problem, because I can make the drive train on the loading bar thus it will not leave the allocated area of loading. The improvements of the catapult are surely exciting, especially since we saw it function better than we expected.

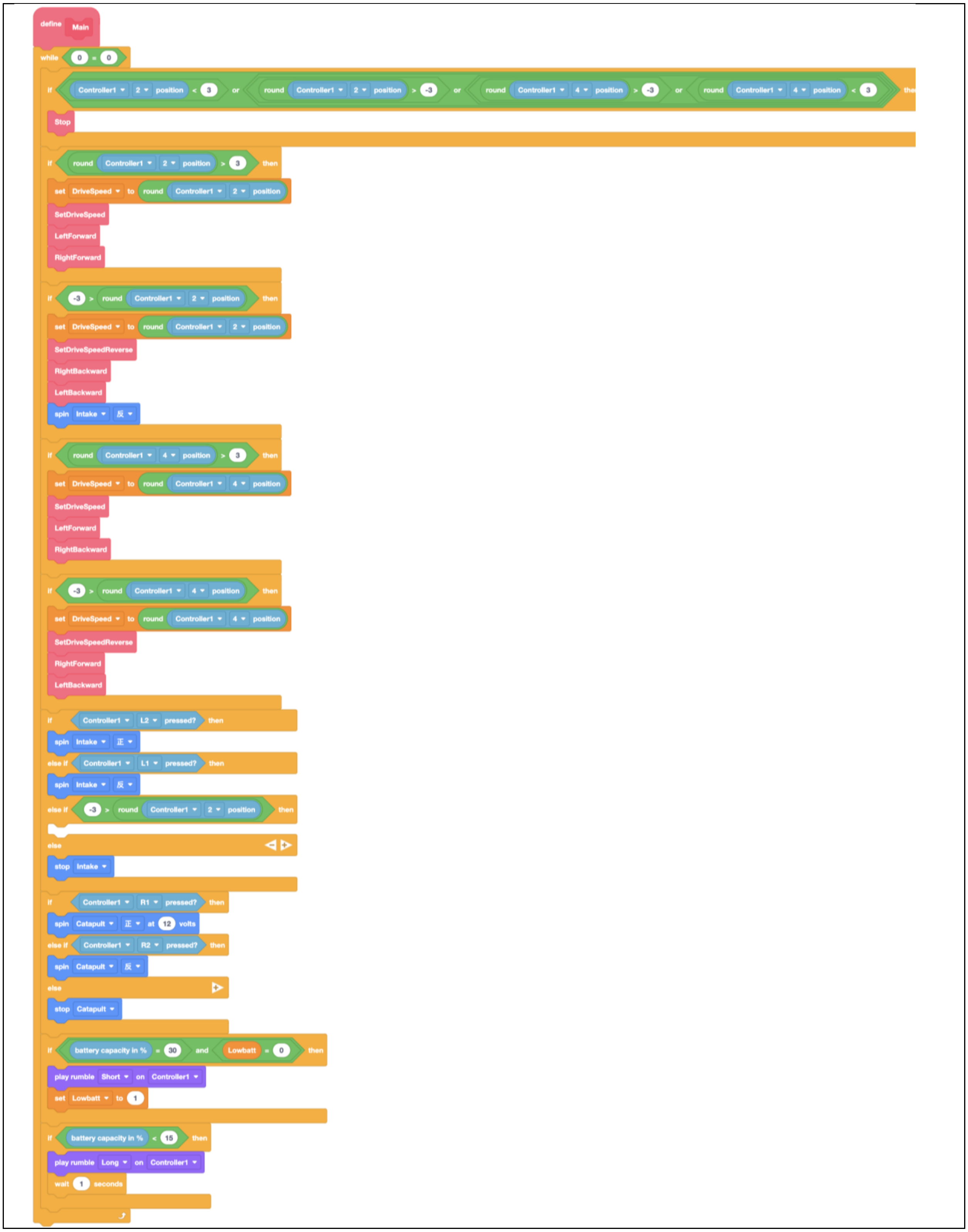
Third, the Intake was re-designed and now the height is about as perfect as we can make. We also made a bar so the triball won't get stuck in the insider area of the drive train. The ways of mounting were not the best, so we found that the mount sometimes collapses a little when we are heavily using the intake system, due to its thickness and thus affects the intake height, since that part also depends on the mount. We can try to solve this problem on Tues and Thurs sessions because they are not a though problem, and we just need to replace the two mounts with some stronger ones.

The code was also improved. I have abandoned text-coding because I do not want to force myself to read something I really don't know with. So, I started using regular block-based coding and I found it was much more easier than text-based. With the improvements of the code, there are some new features than the original plan: first, the intake now spins whenever I drive backwards so the triball stays inside of the intake; second, the left joysticks no longer control the throttle, but they have now been used as turning. This is because with some try-outs I found that the program can detect how much the joystick moved and re-writes the speed immediately, like I mentioned before.

With a lot of troubleshooting processes, we also agreed that Science and Biology and Physics are like "It just can't do", but Engineering is like "不行也得行" (It has to work even if it can't.) We found that we just needed to find the solution without making sure it made sense; The only thing that we make sure of is that it works. It doesn't matter even if the solution is weird.

CODE SNIPPET (BLOCK)

Continued from last page



Title: Meeting entry 12/02/2023	Author: Samuel Y	Date: Dec 5 2023

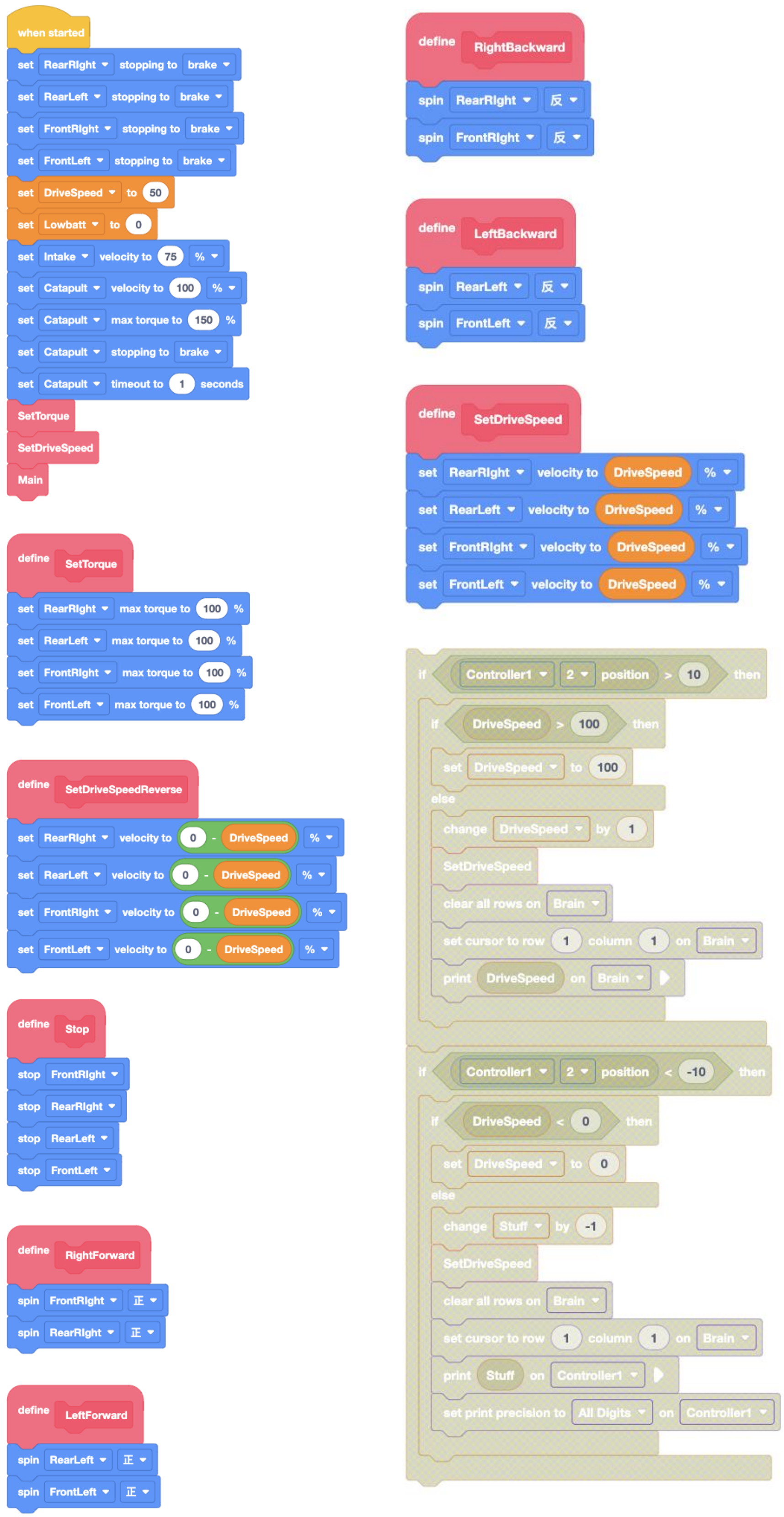


Fig. 1: William's source code from VEXCode V5 Blocks

Title: Meeting entry 12/02/2023	Author: Samuel Y	Date: Dec 5 2023

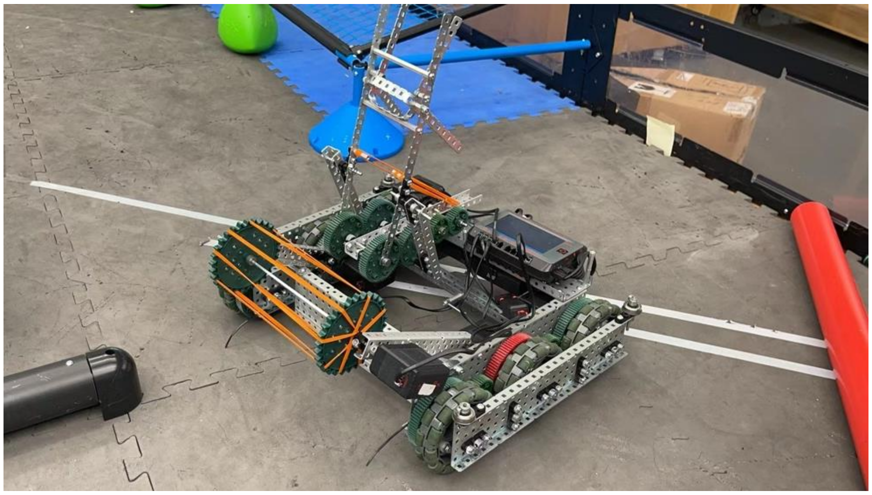
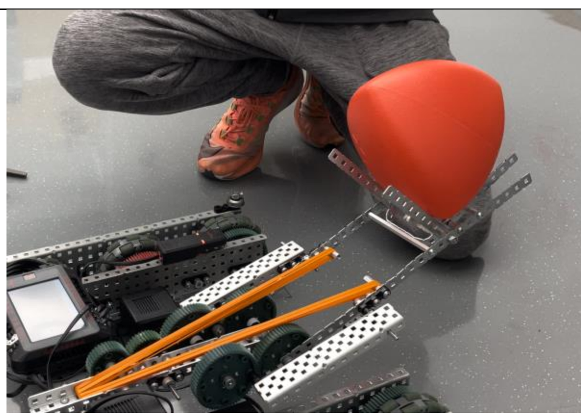
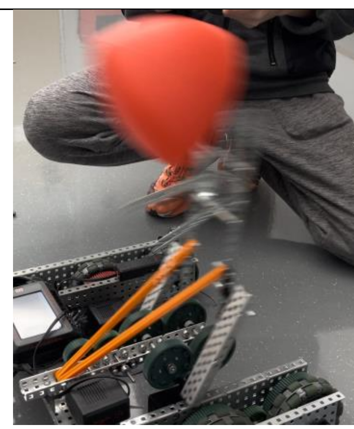


Fig. 2: The finished Robot

Create & Improve: Catapult

[Samuel] At the end of Thursday's session, the robot only had basic functionality at one point, with it not being sufficiently optimized for match loading and a broken motor. After reattaching a new motor with the same motor cartridge (the internals of the motor in the casing were broken), the catapult functioned rather well. After testing out different functionalities of the catapult with how many rubber bands were on the catapult, I visually noted that using four rubber bands, two one each side, the tension from the elastic bands was enough to propel the triball to the other side of the field. If the robot is oriented correctly, the triball could be directly hit into the goal.

 Fig. 3: (Video) The catapult, under tension, right before launching the triball	 Fig. 4: (Video) A triball launched from the catapult as tension is released

Although this catapult is effective, the arc that it travels along is rather low so that it can clear the long barrier but it might not be able to go over a robot that is blocking the match loads, rendering the catapult not useful in such a scenario. In the next two sessions we will think about a driver strategy that could deal with such a situation.

Through our testing of the catapult on the field, we also found out that the robot, instead of simply touching the match load bar as before, could drive onto the match load bar to change the trajectory of the triball so that, instead of clearing above the bar, the triball is aimed more downwards so that it hits or barely misses the long barrier, resulting in the triball coming to rest near the front of the goal instead of near the back portions of the field with a normal arc. In some situations, the triball may also hit the field tiles before the long barrier, then bounce back up and over the long barrier. The use of such a strategy, as an unintended design feature, will be explored as William practices driving more. The driving team members during the scrimmage would also need to practice more to ensure that they can accustomed to the fast match loading speeds of the catapult.

Fig. 5: The robot in the 'alternate' match load configuration

Some other safety considerations for the drive team members are that the catapult could possibly contact the hands of others. To prevent injury, I will have the driver team members don gloves while match loading and recommend them to wear long-sleeved clothing and to tie their hair to prevent the catapult cutting the skin or for the hair to entangle with the catapult.

Develop & Plan: Intake

A problem with our previous iteration of the intake mechanism is that the intake is fixed to the robot, at a fixed angle, without the ability to pivot as the robot goes up against the wall or goal. Because the intake was nonflexible with the previous design, it made it difficult for the triball to be scored into a goal.

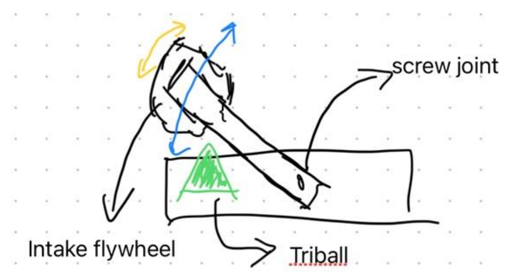


Fig. 6: Side view sketch of hinged intake

By using a screw joint from the intake to the base of the robot, the intake can then pivot from the screw joint to be lifted up partially as needed, such as if the robot is trying to score a triball in possession into a goal.

The implementation of this design idea was fairly basic, since the primary difference between the previous intake and this iteration is the screw joint that allows the intake to be flexible.

Create & Improve: Intake

With these plans for the intake, me and William decided to take the existing roller for our intake with the sprockets and elastic bands and modified it so that it is hinged on two screw joints one, on each side of the robot. Since the design of this flywheel is fairly simple, the mechanism was completed within the session.

Define & Inquire: Catapult

Due to extreme business in regard to studying and other personal matters, I (George) was unable to complete a full writeup of the design process of the catapult as well as the physics principles behind it. When considering what catapults to use, the primary aspects desired were small size, lightweight, and fast operating. As there are no plans to add a loading mechanism, instead using match loading, the catapult did not need to have a low resting (or tensioned) position and needs high launch speed. After considering these factors, it was decided to use a connecting-arm type launcher. This launcher is similar to a type of linear puncher following the same principle.

Meeting Entry 12/07/23

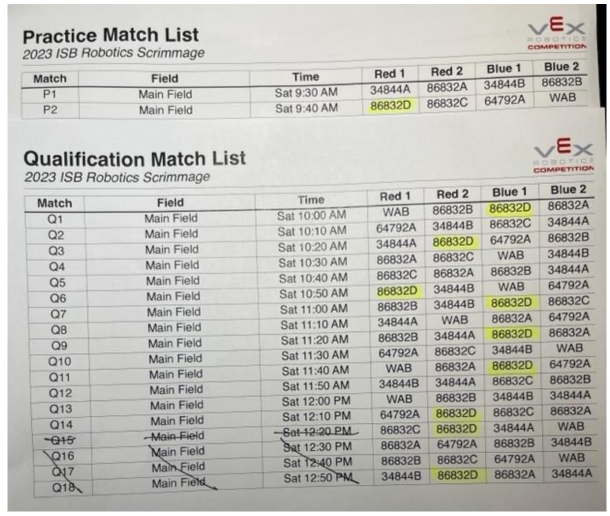
Duration of session:1 hr.				
Task	Team Members	Status	Deadlines	Notes
Driver & Autonomous Code	William	Halfway done	12/8	Hard
Test Drive	William, Samuel	Complete	12/3	
Minor Repairs	Samuel	Complete	12/3	Ready for scrimmage,

This is the last session before the ISB Scrimmage. Although the session in itself is 2 hours long today, we each spent around an hour working on the robot since preparations had to be made for the upcoming ISB Robotics Scrimmage on the 9th of December in regard to field and venue setup.

Title: Meeting Entry 12/07/23	Author: Samuel Yao	Date: Dec 5 2023

ISB Robotics Scrimmage (Dec 9)

During the scrimmage, team members William and Lucas competed on behalf of 86832D, with assistance from teams 86832A and 86832C with match loading during certain matches. Although I (Samuel) gave consistent verbal guidance during the event, I was not able to compete with my team because of my volunteer role as the TM Operator, referee and volunteer coordinator.



Practice Match List
2023 ISB Robotics Scrimmage

Match	Field	Time	Red 1	Red 2	Blue 1	Blue 2
P1	Main Field	Sat 9:30 AM	34844A	86832A	34844B	86832B
P2	Main Field	Sat 9:40 AM	86832D	86832C	64792A	WAB

Qualification Match List
2023 ISB Robotics Scrimmage

Match	Field	Time	Red 1	Red 2	Blue 1	Blue 2
Q1	Main Field	Sat 10:00 AM	WAB	86832B	86832D	86832A
Q2	Main Field	Sat 10:10 AM	64792A	34844B	86832C	34844A
Q3	Main Field	Sat 10:20 AM	34844A	86832D	64792A	86832B
Q4	Main Field	Sat 10:30 AM	86832A	86832C	WAB	34844B
Q5	Main Field	Sat 10:40 AM	86832C	86832A	86832B	34844A
Q6	Main Field	Sat 10:50 AM	86832D	34844B	WAB	64792A
Q7	Main Field	Sat 11:00 AM	86832B	34844B	86832D	86832C
Q8	Main Field	Sat 11:10 AM	34844A	WAB	86832A	64792A
Q9	Main Field	Sat 11:20 AM	86832B	34844A	86832D	86832A
Q10	Main Field	Sat 11:30 AM	64792A	86832C	34844B	WAB
Q11	Main Field	Sat 11:40 AM	WAB	86832A	86832D	64792A
Q12	Main Field	Sat 11:50 AM	34844B	34844A	86832C	86832B
Q13	Main Field	Sat 12:00 PM	WAB	86832B	34844B	34844A
Q14	Main Field	Sat 12:10 PM	64792A	86832D	86832C	86832A
~~Q15~~	~~Main Field~~	~~Sat 12:20 PM~~	86832C	86832D	34844A	WAB
Q16	Main Field	Sat 12:30 PM	86832A	64792A	86832B	34844B
Q17	Main Field	Sat 12:40 PM	86832B	86832C	64792A	WAB
Q18	Main Field	Sat 12:50 PM	34844B	86832D	86832A	34844A

Fig. 1: Practice and Qualification Schedule

During the scrimmage, 86832D played a total of 11 matches, with one practice match, 8 qualification matches and 2 elimination matches. Due to the tournament only having 8 teams, the match schedule was rather packed, with teams having minimal time to prepare between matches, often back-to-back. This even resulted in the tournament falling behind schedule, resulting in one qualification match for 86832D being cancelled.

In total, team 86832D got **fifth place in the qualification** rounds, winning 3 matches, losing 3 matches and tying one match. Even though our qualification seed isn't the most desirable, I strongly believe that 86832D very well considering that out of all teams we have scored **the highest autonomous bonus points (AP) in the entire tournament** with the maximum possible score of **56**, indicating that **for every qualification round we played, we have won the autonomous bonus**. 86832D also received the **highest Strength of Schedule Points (SP**) **in the entire tournament** with **478**, scored the **highest total points** in the entire tournament with 591 and won in the **highest-scoring match** in the tournament with 107 points, possibly indicating that 86832D participated in qualification matches with more high-scoring teams as with the generated schedule. Additional analysis of qualification performance will be included in a separate reflection entry.

Rank	TeamNum	TeamName	Wins	Losses	Ties	WP	AP	SP	NumPlayed	Win Percenta	Average Poin	Total Points	High Score
1	86832C	ISB Dragons C	6	1	0	12	24	373	7	0.857	72.57	508	106
2	86832A	ISB Dragons A	4	2	1	9	24	411	7	0.643	74.57	522	107
3	WAB	Speeding Tige	4	2	1	9	12	449	7	0.643	74.00	518	97
4	86832B	ISB Dragons B	4	3	0	8	4	370	7	0.571	65.86	461	97
5	86832D	ISB Dragons D	3	3	1	7	56	478	7	0.500	84.43	591	107
6	34844A	mission mupp	3	4	0	6	36	306	7	0.429	49.86	349	73
7	64792A	BIS Panthers	2	4	1	5	28	419	7	0.357	60.57	424	84
8	34844B	Still Don't Hav	1	6	0	2	36	349	7	0.143	50.29	352	80

Fig. 2: Exported Tournament Manager statistics for Qualification.

For the elimination rounds, **86832D decided to team up with 86832A**. William's basis for making this decision weighed on the fact that 86832A had a **primarily offensive** robot (with its weight, size and pushing capabilities) that is less mobile and that 86832D had a **more mobile** robot that was able to traverse the field more quickly. Additionally, both teams had different **match loading and launching mechanisms** (86832A had a vertical flywheel, 86832D had our 'choo-choo' catapult). By having two robots with differing specializations, we believe that our robots could focus on **defense and offense** one at a time, with 86832D focusing on scoring and 86832A focusing on defense.

Title: ISB Robotics Scrimmage	Author: Samuel Yao	Date: Dec 11 2023

Post-Scrimmage Feedback & Changes

2024-01-12 ~ 01-17

Situation report: changes

[William] **After the Scrimmage, I made more changes to get ready for our APAC competition. Samuel has already left this team and joined team C, so I will be updating this notebook; George, Lucas and other Juniors are busy with their IB Diploma, thus I am the only person left that is available for most of the sessions. However, I need to do both program and building work, the frequency of updating this notebook is not going to be very often. Lastly, please ignore some of mine Chinglish grammar, I know they are rather silly :(.**

The scrimmage proved the design of the drive train and its intake system. I'd say, they are kind of strong and I don't really need to fix anything, while some other teams are trying to use every second before their next game to fix their broken things. I am very glad that we've had some good quality. However, the choo-choo catapult is somewhat damaged and it has been bent a little, due to our abuse. The design is really weird but all the people who have seen it say, "this just works". It surely did very well in most of the rounds, so I will not change its design, but I need to think about the structure strength problem. It is in fact too wobbly, and I sometimes will panic because it seems like it's going to break, before the real competition.

It did not break yet, but I think it will eventually cause some issues. I also need to think about how I cross the sides with this kind of height. It is way too high, so I need to spin the catapult motor. But with little problematic arcade controls, I often struggle and waste time. For the middle bars, that catapult caused my robot's center of mass problematic – it is at the back, which when I was trying to make it cross, it seems that it may flip over. However, that was a mirage, but at the first attempt I still wasted some time trying to see if it was going to flip.

So, now I must think how to balance the center of mass and program issues about arcade control. Yes, that control is kind of OK, but too silly when I'm trying to do some rapid maneuvers. The turning speed was too fast – and it could not move during turns. So, I was thinking about vector control. When I am driving forward, I should be able to turn, right? And the idea of Vector can be helpful. Move forward will receive a +90 deg (0,100) signal; Right will be 0 deg (100,0); Left will be +180 deg (-100,0) and backward will be –90 deg (0,-100). This idea is good, and I need to adapt it to Vex.

I planned to use 1 and 2(the right joystick) to bind this thing. With my test I found out that moving 1 left will receive a positive value. Now, here's when the logic gets very complicated. 2 will be using to move straight, for that, 2 will report some value, and we set the value as the drive speed. For turn, it is difficult. Its value will be add on to the current drive speed, but only applies to one side of the motor group. For example, if the controller reports 1(+50) and 2(+50), we need to let the right motors go faster in order to make it turn. So: the drive speed will be 50, and the right motors will be 50+50. The basic rule is here:

If turn left – make right motor faster

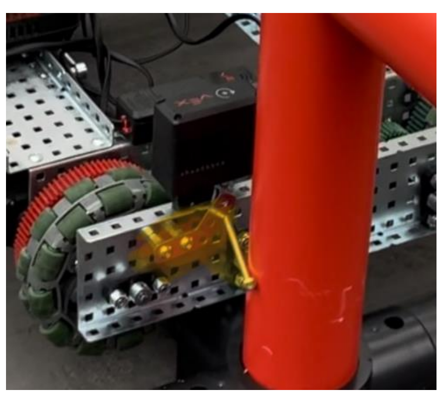
If turn right – make left motor faster

Turning right means 1 reports a negative value, but we want it to be positive. However, that is not a problem. Yes, VEXcode V5 has the absolute value feature, no need to do much. I just need to know what value causes what to operate.

Afterall, there is no really big deal, the only thing is the position of the Brain. Because the need of catapult, I need to reposition brain since it blocks the catapult position. It is not a must-done but, it would cause center of mass issue and make the robot flip when trying to cross the middle bar. If I don't do any balance, I am doomed. The catapult is too high so the sidebars will be such struggle to cross. I have to manually lower it when attempting to do so and will waste a few seconds. Not good! Every second is important. If I just think "Oh, there is 1:45 so few seconds wasted is not going to cause trouble", I will eventually waste more time and lose the game. As a driver, I need to make everything as fast as I can because it will be very critical to control the time and do the elevation in the last few seconds in order to get more points. If nobody does elevation, then I get 20 points, that means 4 triballs.

CI: Elevation

For only one person, pneumatics would require too much work, so I decided that I should use motors. The motor only needs to extract the elevation thing and hold it at the position. However, the torque will be crazy high, since most of the robot mass will be carry on that single motor. In order to make the motor and my life easier, I need to add some mechanism that locks the elevation tile after it extracts. No V5 parts are good for that, and I only have a little space. So, I need to homemade it, by cutting some steel tiles.



The fig. Clearly shows the motor is struggling because my lock piece is poorly made. It is bent a little backward, so I have to think and replace that lock thing. With this kind of situation, I am concerned if it

Continued from last page

1210 can hold 20 seconds but won’t cause permanent motor damage. In the actual test the motor makes big sounds because it is trying to hold it at the position (already set motor mode to LOCK) but its torque is not able to make that much. I do think it will eventually break the motor, so I cannot take that risk because nobody will help me replace it, and I need to spend time between qualifications to do other things like tactics reflection. I want to make sure it is good to go after every match.

*Update from APAC: This mechanism got upgraded. This pervious version was not stable.

Post APAC feedback and changes

We placed 4th in the APAC competition... **In total of 42 teams**. In qualification rounds we just got 8th which means we are guaranteed to pick our alliance. Other ISB teams are out of elimination rounds so I need to pick a good teammate that has a good intake system and has a relatively fast driving speed, so they will match up with my robot – it has good catapult but drives not fast. This ideology in Chinese is called 取长补短.

Even though we didn't get any trophies, there were a lot of robots, and we got to scout them; the qualification rounds also made me a little more experienced at driving and strategy making. There were few things that made our robot did bad:

1. The blocker from opponent. Even though only a few teams got that mech, it is very useful for blocking my catapult and mine is no longer useful in that match, since every time I try to load triball and shoot, that robot will instantly lift their blocker and drive close to me. It is very annoying.
2. The elevation isn't as stable as we think of. In my semifinal round, the elevation didn't work as intended, thus it didn't lift the robot from the ground. However, that round's score was so close that the win or loss depends on if my robot lifted. It did not so we lost the opportunity to chase for championship. I was very depressed and felt very bad about my driving skill issue. If I drive it a little forward, then we would eventually win it. However, after the match I realize I cannot rely on my driving skills since I will get very nervous, and it is unstable. What is guaranteed and will not change too much is those mechanisms we made. Robots are more reliable than humans; they don't feel nervous or under pressure. If there is nothing much I can do with nervousness, then I try to control what I can control, which is the elevation and the structure strength of the drive train.
3. Alliance selection was not done that perfectly. I should spend more time scouting and picking the teammate, and this time I picked a teammate who only has good drive speed, good intake but no elevation or catapult. That eventually made our strategy limited, it is not very good. Having some redundancies is always good, because if mine catapult or elevation didn't work, they still have theirs that can be alternative.

R/S: The strategy of Over Under game

2024/2/14

After scouting most teams in APAC, I think I need to think about the strategy, since I found that it highly impacted game performance. An important concept is that all robots made for Over Under game must or want to contain these features: Drive Train (DT), Intake (IN) and Catapult (CT). According to me and my teammate's scouting data, most of the APAC teams only get their DT and IN up and running. Some teams are trying to figure out how to make the CT work but, only a few teams have all three features. In fact, most teams who last until the elimination rounds mostly have all the features (and some juicy all-blue motor drive, they are good).

Another important “unwritten rule” is all teams with CT also have an IN. This probably means the teams are likely to start with their INs, and only after they finished making IN, they turn to make the CT. INs clearly has more priority than the CTs, but APAC showed that having a catapult could massively increase the efficiency of adding triballs to the alliance zone (Also add opponent’s SP). These quadrangles really look like Chinese rice dumplings (粽子) and are super scrollable. If the CT is poorly made, then it can shoot them to the corner of the field and eventually get trapped because of that bar (see fig.1). Even with a satisfied and controlled catapult, the triball may still move because of the shooting power, inertial makes it continue roll for a while.

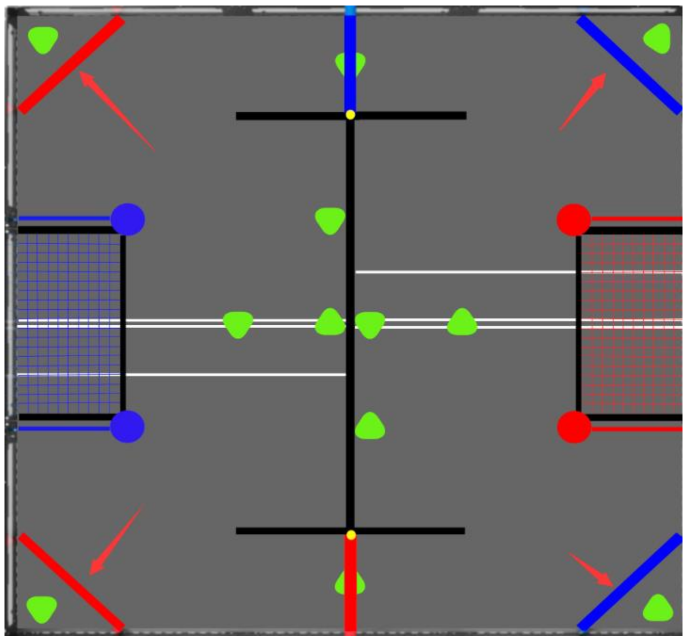


Figure 1: the corner of the field

We can see that you are not going to make them out once the balls get stuck in these areas. The intake can’t reach them over the bar; making a mechanism to collect these stuck ones probably have a poor quality-price ratio. Our catapult sometimes has this kind of issue, and it eventually made the corner full of balls. Nobody can reach them, and it is a pity that they cannot be collected in the goal.

Other catapults in other teams have some different bugs. The team’s CT may be so powerless that it even cannot make the ball cross over the center barrier to some extent. As far as I know, both Jun’s team (Team C) and Ryan’s team (Team A) have had some similar situations when they were developing a catapult.

Ryan was having a flywheel for all his robot versions, including the first “Tank” and the second “Speeder”. All of these flywheels are too weak, and it is such unstable when spinning – the robot will be heavily shaking. For his second robot, there was a severe torque problem – The motor gets overheated after running for 10 seconds because there is too much torque (thus it is not helpful). In my opinion the only reason is the flywheels are not able to collect kinetic energy from the rubber bands, they just rely on spinning speed. Well, it is probably not that good for launching this scale of projectiles, they are a little heavy and too big.

Jun’s team did use rubber bands (even more rubber bands than my team’s) but it didn’t cross the bar until recently. I think that is quite reasonable, because their catapult is too short.

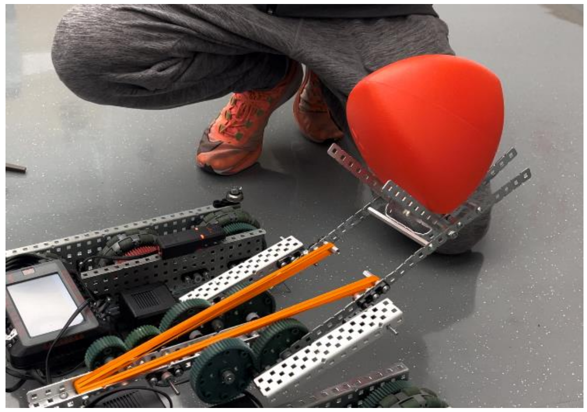


Figure 2: our catapult

This figure clearly showed that our catapult is quite long, I mean the distance from the center of rotation. There is an interesting fact that things will have a longer displacement distance if they are far away from the center of rotation, compared to things close to them, if they are attached to the same rotation energy source and share the center of rotation. This makes sense because the radius has been changed. If there is longer radius, then longer perimeter. Archimedes once said: *Give me a fulcrum and I can pry the earth* (with a very long pole, of course). The principle of the lever tells us: The farther you are to the fulcrum (also center of rotation) the faster projectile moves, vice versa. Of course, what we do is a laborious lever to create more speed, since the ball is relatively light.

In APAC games it really looks like the catapult is hard to make to some extent. I saw some teams from Shanghai didn’t make the catapult work because they did not have enough power. Also, from WAB’s elimination round I saw their flywheel worked badly and required a long time to shoot the ball out, thus making them lose the round. That design is an old version from the early Over Under games. I saw a similar design when I was in Philips Andover in Summer 2023 (Unfortunately, I did not take any pictures). I don’t think it was their skill issue, but the design is too outdated to chase a good place in **2024** Over Under games. But back in 2023 Summer, I saw the PA robot (with same design) working extremely well. I suppose more new designs went out and beat the old ones up, that is totally normal. If WAB can make some changes to its stability issue, then they will do better.

From our scouting, in one match, if neither of the teams have a catapult, then both the teams will receive a relatively low score, because match load takes “years” to finish (The teams must “load” the ball in front their intake and drive to their side). For those matches that have only one catapult, there will be some risks for the catapult robot - because the balls may be stolen from the opponent (because they got two intakes running) if their teammate is slow in speed and cannot collect the ball with good efficiency. That's why it's important to communicate with your teammates before the game - We need to know exactly how fast the robot can go - otherwise maybe don't use catapults.

 Continued from last page

One feature I did not talk about is the elevation. In some close games, the elevation can help the team. However, this is built on if opponents don’t get any of the elevations, that could increase alliance's score by more than 10. For some games, the alliance can still win the game even if they have less triball than their opponents. However, a successful elevation is equivalent to maximum of 4 triballs in the goal zone. With this equation, the team should want to improve their catapult first because they will be able to shoot all the 4 triballs in less than 8 seconds, their teammates can quickly push those in the goal. As long as they and their teammates co-op well, they should expect all the triballs will be pushed in before the last 15 seconds. If that is the case, then the elevation can maintain the score (in the last 30 seconds, the opponent cannot interrupt or touch the robot if they are trying to elevate or already did so).

Few good designs combine the catapult and blocker. So, if the blocker robot sees me match loading, they run in front of me and block those balls. In APAC only one team’s blocker worked well, and I had to give up the use of catapult because they quickly spotted my movements and runs in front of me. Very annoying. Some other teams also got a blocker, but they were not high enough and the team got embarrassed because they were unable to catch even one single match load from my catapult. For future games I am considering using my catapult as a blocker. However, it does need future validates and I am wondering if it’s strong enough to block the match loads. If it breaks when blocking, then the gains do not make up for the losses and I must give up this idea.

There will be a holiday session on Feb 18, and I must change the motors first (new blue motors have arrived), and then manage to verify all these hypotheses.

Work Session: Feb 17

Duration: 10:00 - 16:00			
Item	Status	Due date	Note
Switch the intake motor to blue ct.	Done	Today	Works well.
Rework Automation	Done	Tomorrow	Minor instability issue
Catapult fix	Done	Today	One of the shaft was damaged
Prove of catapult as blocker	Done	Today	-

It was a very productive day, I proved some of my theories and fixed the catapult.

Very first thing, the blue cartridges arrived. It is a good time to work with changing the cartridges. However, my original plan was to change the drive train to all blue motors. The problem with this idea is – Jun told me it won’t get enough power to cross the middle bar. With a little thinking I think he convinced me. The robot right now is a little struggle to climb the middle bar, and if I change the motors, it will be even worse. So, with this consideration I decided not to change that but only the intake. The intake surely needs an upgrade since it was too slow.

Also, after APAC games the catapult was broken because of our abuse. I tried to troubleshoot and found one shaft very canted. Fortunately, only that was broken, and the rest actually stayed strong. I still don’t know why it’s just well, and that is very different from its outlook. After I finished fixing things, I asked Jun to test out my catapult and use it as a blocker. Jun’s catapult is now fixed and normally crosses to the other side of the field. And the result is very satisfied – it did block the triball (See figure). For some of triballs it didn’t stop it but slowed it down, thus it also won’t cross. I accept both situations, since I did what is intended to do – make the opponent unable to cross the triball to their side, which slows their speed because they’ll need to collect them from my side and push them.

Considering the use of the catapult from 2023 December ISB scrimmage to the 2024 February APAC games, it did not break until recently. This catapult surely demonstrates George’s ability of innovation and adaptability. George has some great ideas and he really proved them with above-great quality. We fully organized the catapult ability and scored more points by more communication and trust to our teammates. This is one of the good ways to win matches.

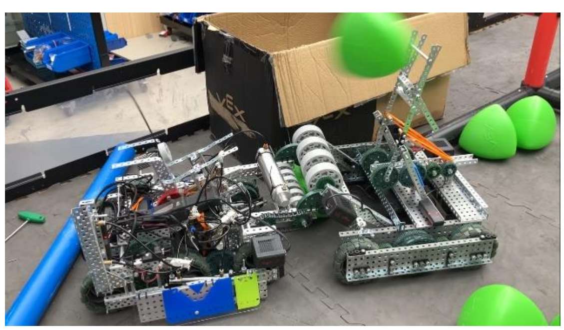


Figure 1: catapult blocking Jun’s shooting triball

With the remaining time I did some automation fixes. The one used in APAC was too rough and it did not have satisfied performance. So, this time I need to not just fix but also rewrite it. With some struggle and

Continued from last page

brain burning, I coded a better version of autonomous by classic block-based coding. It pushes alliance triball in and also collect the ones locate in the side bar and relocate it to the scoring gate.

However, I made a huge mistake: If I do so, then I am actually scoring for my opponent, since I will be in defensive zone – the opponent zone will have one green triball inside **their** goal, and it counts as theirs. Thus, I have to change the last parts of automation. By Jun's helping I managed to have a new plan: I push the side bar triball by intake to my zone, and make the catapult down - so it touches the top of side bar, which this movement score more autonomous points.

Tomorrow, I need to focus on increase the stability of autonomous. It's now not stable because I'm using block-based and spinning xx degrees doesn't mean the robot will actually spin that degree. With testing I found make the motor spin 0.925 rounds is equivalent to spin the robot 90 degrees.

I feel that working good is more important than looking good.

Work Session: Feb 18

Duration: 10:00 - 12:30			
Item	Status	Due date	Note
Revise Automation	Working	Feb 20	Fixed instability issue

Today was a quick little session. Fixed Automation issue and trained driving skill. Automation still needs improvement and it now only costs 10~ seconds, which means I can try to send the middle triballs to the other side, it will not be any hard, just requires time to try out and program it step-by-step. We still have one session on Feb 20 so it is okay.

Work Session: Feb 19

Duration: 15:45 - 16:25			
Item	Status	Due date	Note
Revise Automation	Finished	Feb 20	

Today was a quick mini-session. Added some blocks – Now the automation will go back to the shooting zone after it hits the corner triball. No longer try to catch the center triball since our robot will easily get DQ and I cannot guarantee that it will stay in the zone.

I also trained some driving skills like to control the endgame to make it stable. The estimate time of completing endgame costs around 10 seconds. In real games, to avoid panicking, I should round this value up to 20 seconds in order to have time relocate the robot and to attach on the middle bar. If the opponent knows our endgame, then the time should be around 30 seconds, since I faced issue when the opponent is trying to stop me from hanging the robot (In the 3rd runners-up game from APAC). The elevation on that game was very critical. If I could elevate, then the score would be enough to let us make the bronze trophy to some extent（The score was only less than 5 points behind the opponent.）. This is extremely regrettable, and I was feeling bad. Successful elevation may save the whole match if the triball scores are close; something is always better than nothing, even if that is the lowest tier.

RS: Robotics Data ANALYSIS

1390

SCHOOL	TEAM NAME	TEAM NUMB	TOTAL RANKING	W/L/T	OPR(AVG	DPR(AVG	CCWM(A	ISB RANKING	W/L/T2	OPR(ALL)	DPR(ALL)	CCWM(A	APAC RANKING	W/L/T3	OPR(ALL)	DPR(ALL)	CCWM(A	CONCORDIA RANKING	W/L/T4	WP/AP/S	OPR(ALL)	DPR(ALL)	CCWM(A
AISG	AISG2	AISG2	1	9-0-0	73.3	21.6	51.7						1	9-0-0	73.3114	21.6288	51.6826						
*SASPX	SAS Puxi - W (PXR	45888W	2	14-0-0	53.3	18.5	34.9						2	9-0-0	54.6582	8.45233	46.20587	1	5-0-0	10-36-149	52	28.5	23.5
YCIS Shanghai	YCIS 1	89644A	2	4-1-0	0.0	-34.5	34.5											2	4-1-0	8-32-265	0	-34.5	34.5
SASPD	SASPD4	1294X	4	11-3-0	40.4	18.9	21.6						4	7-2-0	78.8635	35.2231	43.6404	4	4-1-0	8-32-259	2	2.5	-0.5
CISS	CISS A	13212A	4	13-1-0	36.4	18.3	18.2						3	9-0-0	57.8575	10.0268	47.8307	5	4-1-0	8-32-249	15	26.5	-11.5
AISG	AISG1	AISG1	5	7-2-0	40.9	33.3	7.6						5	7-2-0	40.9449	33.3048	7.6401						
BCIS	mission muppete	34844A	6	3-4-0	16.5	24.0	-7.5	6	3-4-0	16.5	24.0	-7.5											
HKIS	HKIS1	936A	6	6-3-0	44.9	41.9	3.0						6	6-3-0	44.9023	41.8818	3.0205						
ISB	ISB Dragons D	86832D	7	9-6-1	39.7	21.3	18.4	5	3-3-1	19.5	14.0	5.5	8	6-3-0	59.9295	28.5948	31.3347						
YCIS Shanghai	YCIS 1	89644B	7	4-1-0	10.0	14.5	-4.5											7	4-1-0	8-24-224	10	14.5	-4.5
NIS	NIS	NIS	7	6-3-0	51.1	35.7	15.4						7	6-3-0	51.1441	35.7079	15.4362						
*WAB	Speeding Tigers	WAB	7	10-5-1	40.7	36.1	4.6	3	4-2-1	38.5	32.0	6.5	11	6-3-0	42.83	40.1001	2.7299						
BCIS	Still Don't Have O	34844B	8	1-6-0	13.5	23.0	-9.5	8	1-6-0	13.5	23.0	-9.5											
BIS	BIS	64792A	8	6-5-1	14.2	11.3	3.0	7	2-4-1	10.5	16.0	-5.5						9	4-1-0	8-16-238	18	6.5	11.5
ISB	ISB Dragons C	86832C	10	11-4-1	33.1	26.8	6.3	1	6-1-0	31.5	21.0	10.5	19	5-3-1	34.6283	32.5439	2.0844						
CISG	Get Better Dot Or	89944A	10	3-2-0	8.0	-8.5	16.5											10	3-2-0	6-24-161	8	-8.5	16.5
*SASPD	SASPD1	1294A	11	9-5-0	50.3	42.8	7.5						10	6-3-0	60.6727	42.0755	18.5972	11	3-2-0	6-16-307	40	43.5	-3.5
*CISS	CISS D	13212D	11	10-4-0	37.6	38.8	-1.2						15	6-3-0	37.2507	50.1337	-12.883	6	4-1-0	8-24-306	38	27.5	10.5
CISS	CISS C	13212C	13	9-5-0	29.9	21.3	8.6						14	6-3-0	55.8127	50.0717	5.741	12	3-2-0	6-16-218	4	-7.5	11.5
ISB	ISB Dragons B	86832B	15	8-8-0	29.3	31.0	-1.7	4	4-3-0	27.5	24.0	3.5	25	4-5-0	31.0024	37.9722	-6.9698						
SASPD	SASPD2	1294B	15	8-6-0	20.3	23.6	-3.3						9	6-3-0	46.6917	34.7502	11.9415	21	2-3-0	4-12-272	-6	12.5	-18.5
*CISS	CISS B	13212B	16	8-6-0	56.5	43.5	13.0						18	5-4-0	52.0718	35.5811	16.4907	13	3-2-0	6-12-239	61	51.5	9.5
SASPD	SASPD3	1294C	16	8-6-0	28.6	31.8	-3.1						16	6-3-0	50.2961	41.0494	9.2467	16	2-3-0	4-24-271	7	22.5	-15.5
SASPX	SAS Puxi - T (Racc	45888T	17	7-7-0	12.7	25.7	-13.0						20	5-4-0	34.3733	34.8949	-0.5216	14	2-3-0	5-12-210	-9	16.5	-25.5
HKIS	HKIS2	936B	17	6-3-0	39.3	48.3	-9.0						17	6-3-0	39.3476	48.2992	-8.9516						
CISS	CISS E	13212E	18	7-7-0	22.7	26.3	-3.6						33	3-6-0	20.3199	36.0519	-15.732	3	4-1-0	8-32-262	25	16.5	8.5
SSIS	SSIS Team 4	55852D	19	7-7-0	21.6	33.3	-11.7						22	5-4-0	33.1563	47.1167	-13.9604	15	2-3-0	4-28-237	10	19.5	-9.5
广东广雅中学	GYRobots	14285A	19	2-3-0	19.0	16.5	2.5											19	2-3-0	4-16-214	19	16.5	2.5
/SASPX	SAS Puxi - U (912	45888U	20	7-7-0	30.0	28.1	1.9						21	5-4-0	46.9864	39.6598	7.3266	18	2-3-0	4-16-230	13	16.5	-3.5
ISB	ISB Dragons A	86832A	20	5-10-1	29.1	44.0	-14.9	2	4-2-1	29.5	36.0	-6.5	38	1-8-0	28.6506	51.9433	-23.2927						
WCIS	RoboWells_001 ('	90317A	20	6-8-0	39.3	36.8	2.5						12	6-3-0	51.6332	35.1106	16.5226	28	0-5-0	1-0-297	27	38.5	-11.5
/SASPX	SAS Puxi - V (Ken	45888V	21	6-8-0	25.9	36.6	-10.7						13	6-3-0	39.7756	26.6757	13.0999	29	0-5-0	0-12-115	12	46.5	-34.5
HIS	HIS2	HIS2	23	5-3-1	28.8	27.2	1.7						23	5-3-1	28.8481	27.1538	1.6943						
WCIS	RoboWells_002 ('	90317C	24	5-9-0	18.9	25.2	-6.3						39	1-8-0	20.7116	38.9084	-18.1968	8	4-1-0	8-24-181	17	11.5	5.5
AISG	AISG3	AISG3	24	4-5-0	35.1	49.2	-14.1						24	4-5-0	35.1085	49.2125	-14.104						
SSIS	SSIS Team 2	55852B	25	6-8-0	11.7	26.4	-14.8						27	4-5-0	24.3386	37.3745	-13.0359	22	2-3-0	4-12-245	-1	15.5	-16.5
SSIS	SSIS Team 1	55852A	25	5-9-0	15.7	23.1	-7.4						30	3-6-0	8.37525	30.6907	-22.31545	20	2-3-0	4-16-211	23	15.5	7.5
*SSIS	SSIS Team 3	55852C	26	6-8-0	34.3	45.5	-11.2						28	4-5-0	32.5716	46.562	-13.9904	23	2-3-0	4-12-191	36	44.5	-8.5
??	A.L.F	53475A	26	1-4-0	16.0	37.5	-21.5											26	1-4-0	2-16-141	16	37.5	-21.5
DCSPD	DCSPD2	DCSPD2	26	4-5-0	27.6	47.9	-20.3						26	4-5-0	27.5638	47.8992	-20.3354						
??	Armando	53475B	27	1-4-0	5.0	21.5	-16.5											27	1-4-0	2-8-175	5	21.5	-16.5
SSIS	SSIS Team 5	55852E	29	3-11-0	24.1	33.4	-9.3						41	1-8-0	26.144	33.2584	-7.1144	17	2-3-0	4-16-297	22	33.5	-11.5
/SASPX	SAS Puxi - S (9132	45888S	30	4-10-0	17.7	27.4	-9.7						35	2-7-0	21.4235	27.2502	-5.8267	24	2-3-0	4-8-291	14	27.5	-13.5
WCIS	RoboWells_002 ('	90317D	30	4-10-0	21.5	24.2	-2.7						34	3-6-0	29.9909	38.9243	-8.9334	25	1-4-0	2-24-239	13	9.5	3.5
CISG	Trevor Phillips Inc	89944B	30	0-5-0	13.0	9.5	3.5											30	0-5-0	0-4-190	13	9.5	3.5
*WCIS	RoboWells_001 ('	90317B	30	3-11-0	34.8	22.9	11.9						29	3-6-0	37.5659	43.3228	-5.7569	31	0-5-0	0-0-0	32	2.5	29.5
DCSPD	DCSPD1	DCSPD1	31	3-6-0	22.2	49.9	-27.7						31	3-6-0	22.2227	49.9247	-27.702						
SASPX	SASPX6	SASPX6	32	3-6-0	39.5	41.3	-1.8						32	3-6-0	39.5126	41.3018	-1.7892						
HIS	HIS1	HIS1	36	2-7-0	31.1	35.9	-4.8						36	2-7-0	31.0722	35.9181	-4.8459						
BISS	BISS2	BISS2	37	2-7-0	24.0	47.1	-23.0						37	2-7-0	24.012	47.0614	-23.0494						
HKIS	HKIS3	936C	40	1-8-0	15.1	63.5	-48.4						40	1-8-0	15.0889	63.5306	-48.4417						
BISS	BISS1	BISS1	42	0-9-0	11.6	46.8	-35.2						42	0-9-0	11.5702	46.7526	-35.1824						

*This is only the part of the chart. To see more, please go to this link: Click

Samuel and Silas made an chart showing all the match results of APAC teams in the 2023-2024 Over Under games. Looks like for these games my team placed in a good position with relatively good data. The OPR means offensive power rating and the DPR means defensive power rating. With the OPR, the more the better; with DPR, the less the better. Even our win/lose is a 9/6, we still placed the 7th place in this chart. I think this is because we did well on our offensive by massively shooting triballs to our zone in almost every single game.

1400

[i] https://wiki.purduesigbots.com/hardware/defensive-mechanisms
[ii] https://wiki.purduesigbots.com/hardware/shooting-mechanisms/catapult
[iii] https://www.youtube.com/watch?v=aIDwkuRAjqs&ab_channel=WingusDingusRobotics
[iv] https://www.youtube.com/watch?v=jD-ZuIfMb6A&ab_channel=RobokauzRobotics