TYNIEBRAINS

ENGINEERING PORTFOLIO

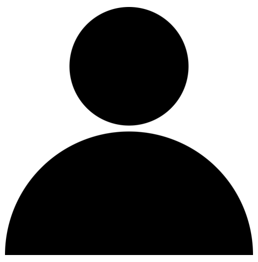
#28436

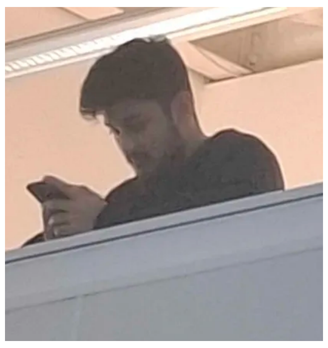
BIG DREAMS. BIG AIMS. TYNIE BRAINS.

THE TEAM

SOFTWARE


Akshith B


Ebaa A


Omar M

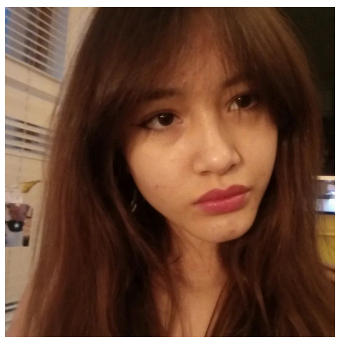

Rishon A

The software team consists of 4 people. Omar M, Rishon A, Ebaa A and Akshith B. They focus on programming the robot and building the website.

DOCUMENTATION

The documentation team consists of 4 people. Aashi M, Aqiilah M, Srisarvesh PK and Rishon A. They focus on the logs, portfolio and emailing sponsors.


Aashi M

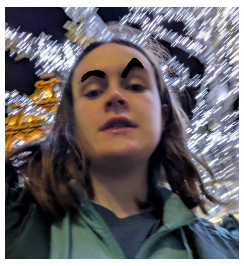

Aqiilah M


Rishon A

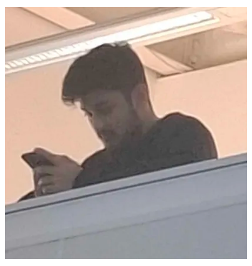

Srisarvesh PK

BUILD


Akshith B


Dylan


Omar A

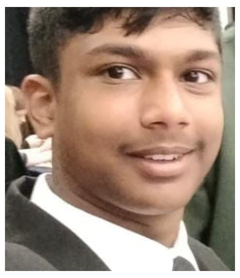

Omar M

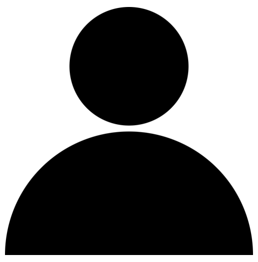

Srisarvesh PK

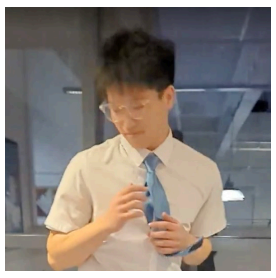
The build team consists of 5 people. Dylan D, Akshith B, Srisarvesh PK, Omar A and Omar M. They work on building the robot.

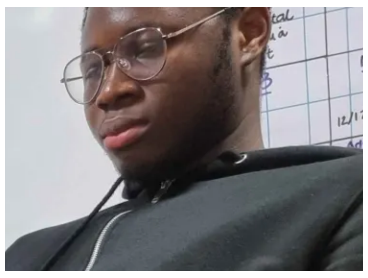
DESIGN

The design team consists of 4 people. Akshith B, Ebaa A, Malik A and Isaac C. The design team works on calculations and the robot's design


Akshith B

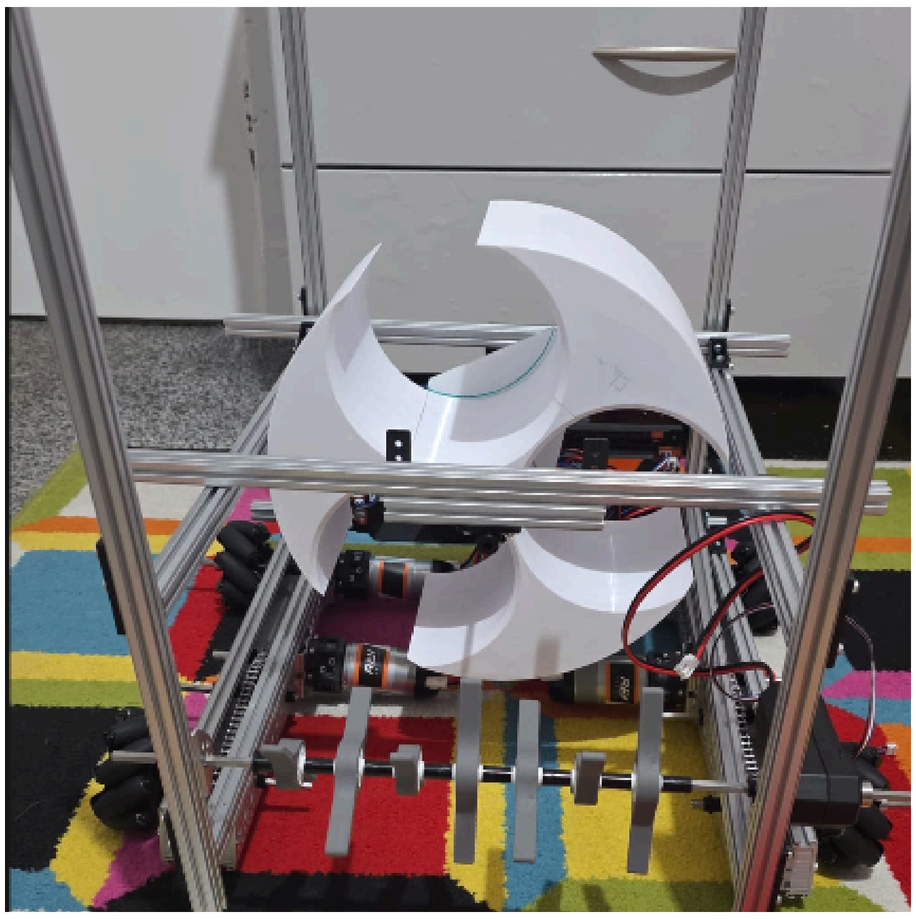

Ebaa A


Isaac


Malik

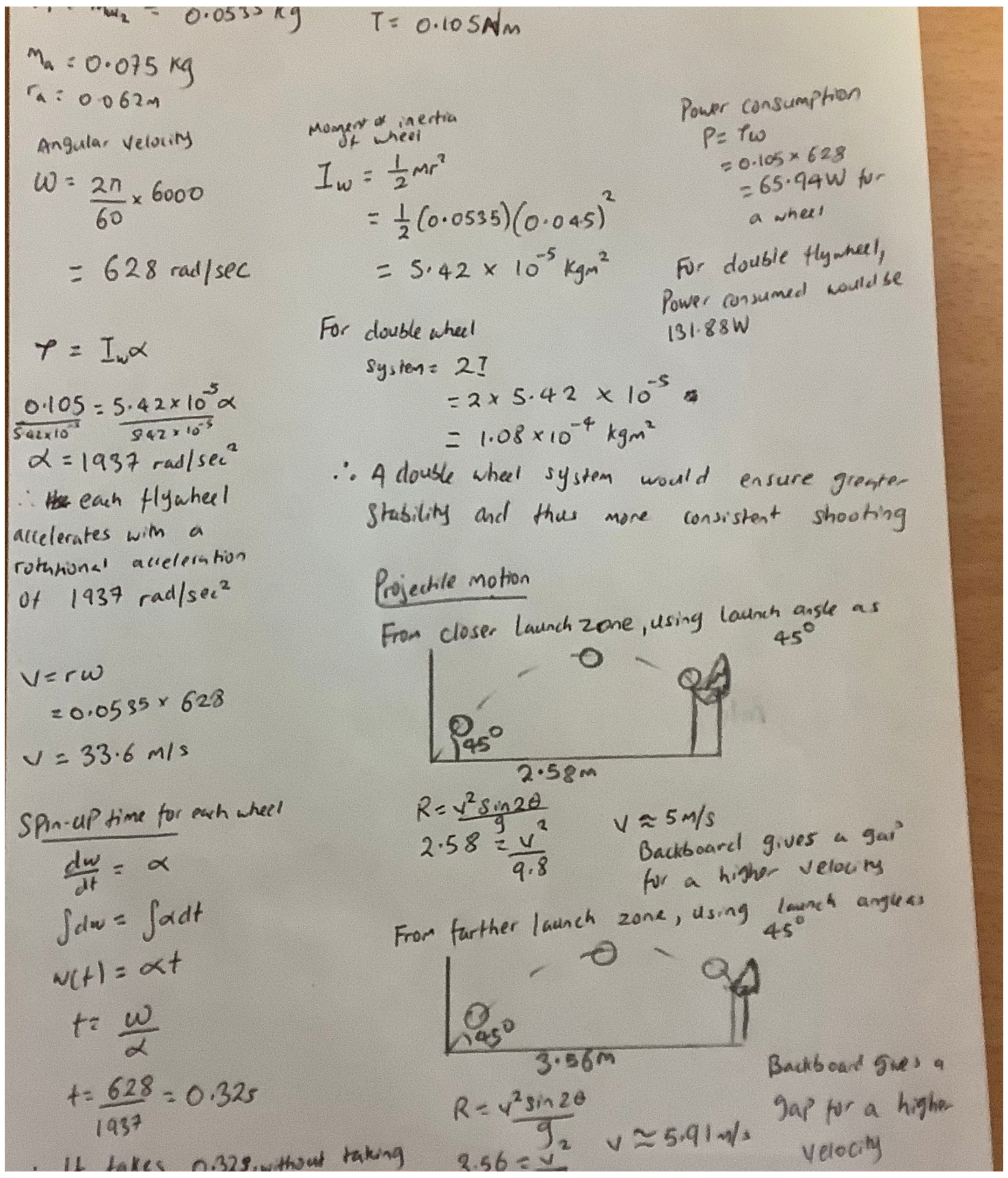
THE ROBOT

Our robot, BrainBot, was engineered for the DECODE challenge with a focus on reliability, fast cycle times, and consistent scoring. From the start of the season, we prioritized modular design in our code that allowed each subsystem to be developed, tested, and improved independently. This approach helped us iterate quickly and maintain the robot efficiently. Every design choice— like our scoring mechanism—was made to maximize precision, reduce failure points, and support a strong autonomous routine.

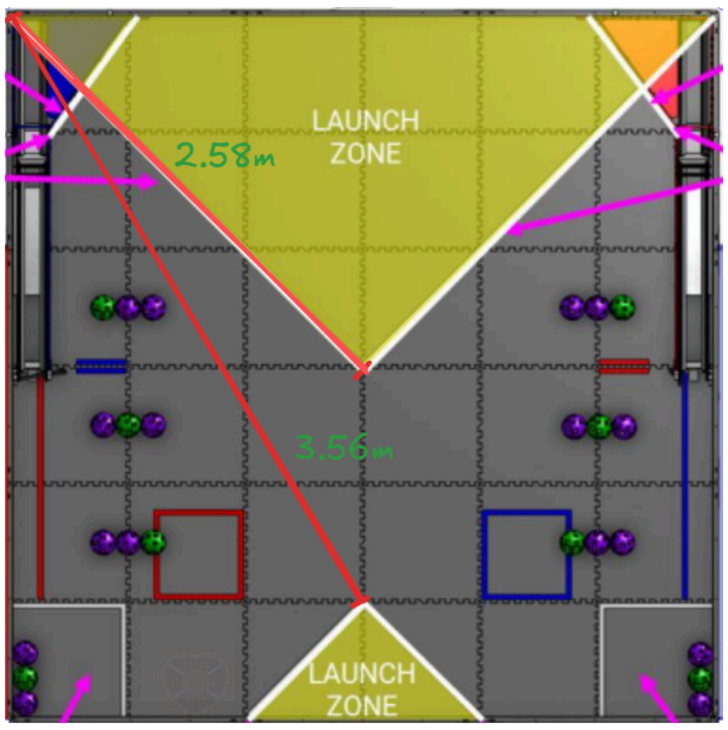


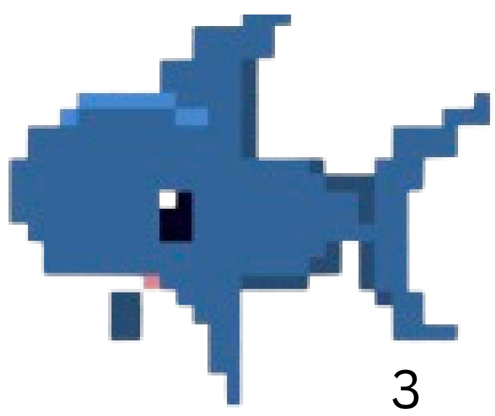
BEHIND THE ROBOT

CALCULATIONS

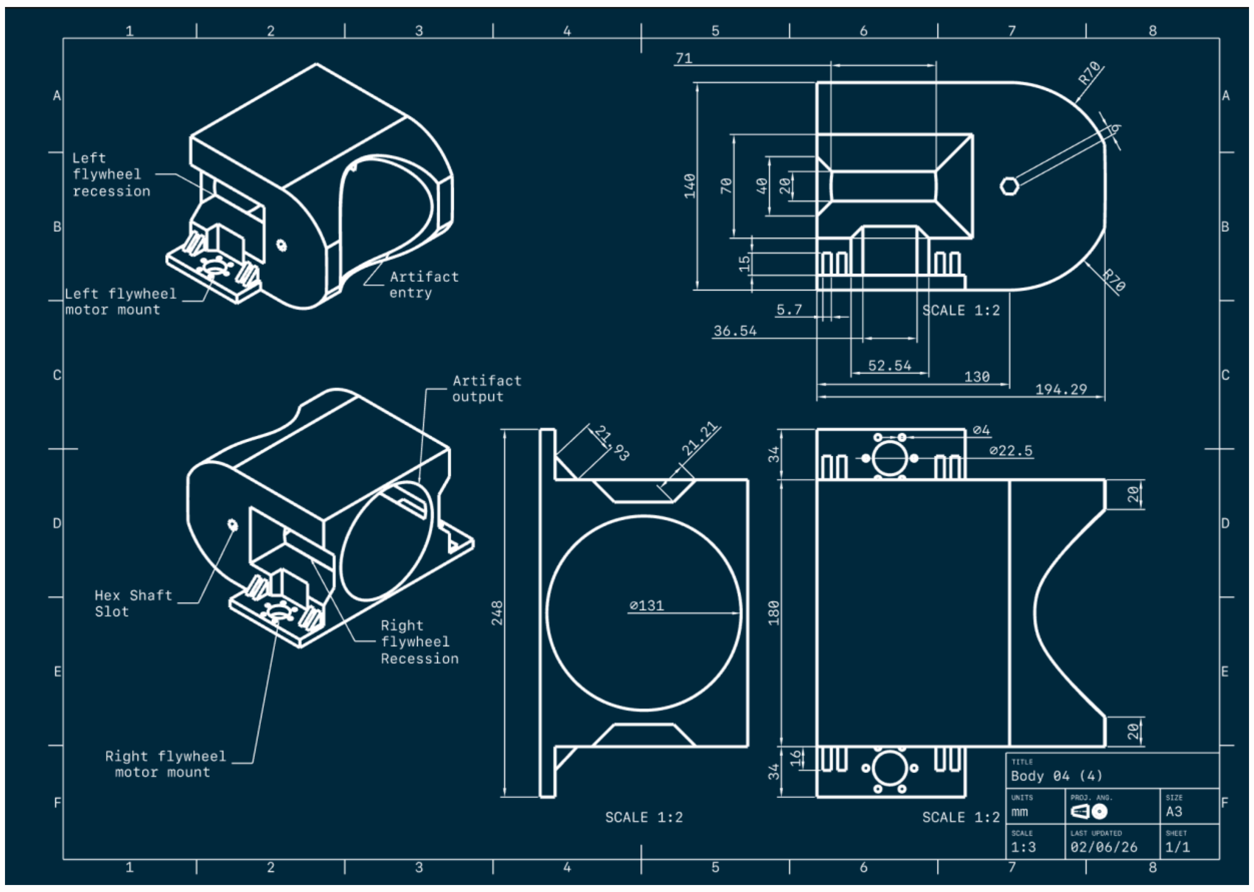


Our build team had people working on calculations to optimise the robot's performance, specifically within the flywheel to calculate the angles of projection relative to the distance.





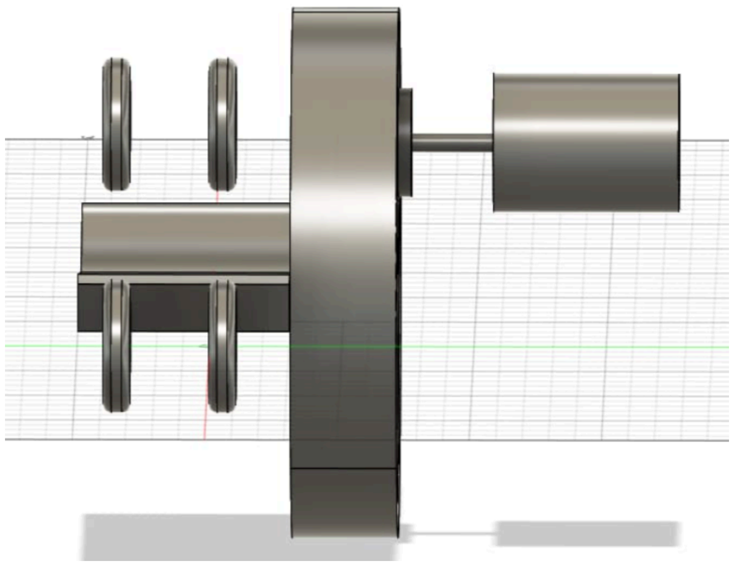
DESIGN



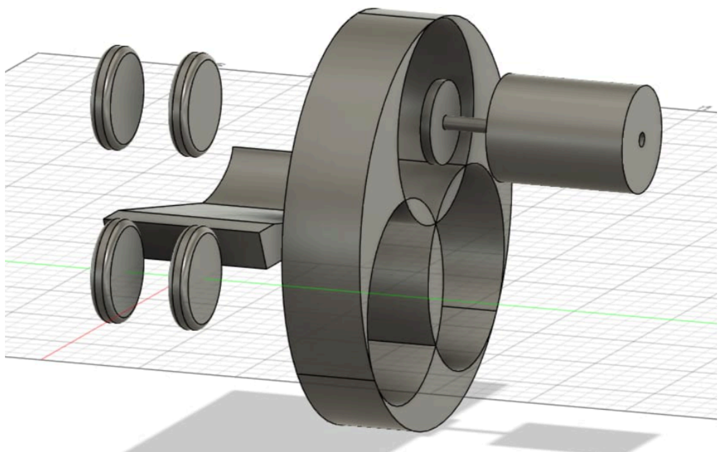
During early testing, the single-flywheel launcher struggled to achieve consistent shot velocity. We observed significant energy loss due to friction between the flywheel and the artifacts, which reduced both accuracy and range.

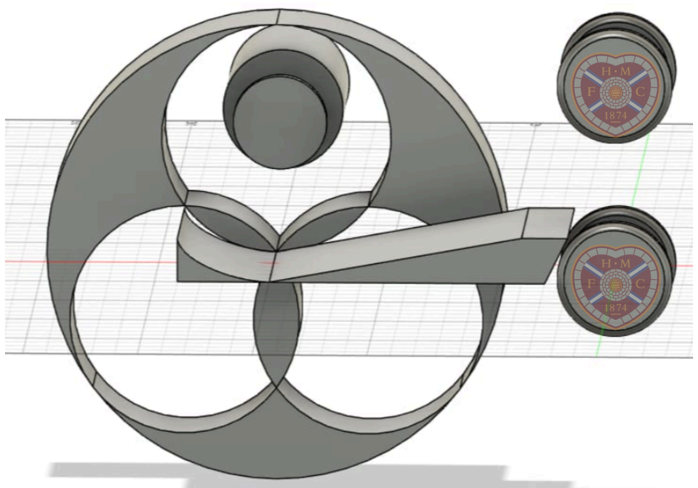
Using two wheels distributes rotational load across two axles, which reduces friction at each bushing and minimizes speed loss during operation. The dual-wheel setup also provides better rotational stability, resulting in more predictable compression and a smoother release of the artifacts. External testing showed that this configuration maintained higher and more consistent RPM, which directly improved accuracy and shot repeatability.

CAD MODELING

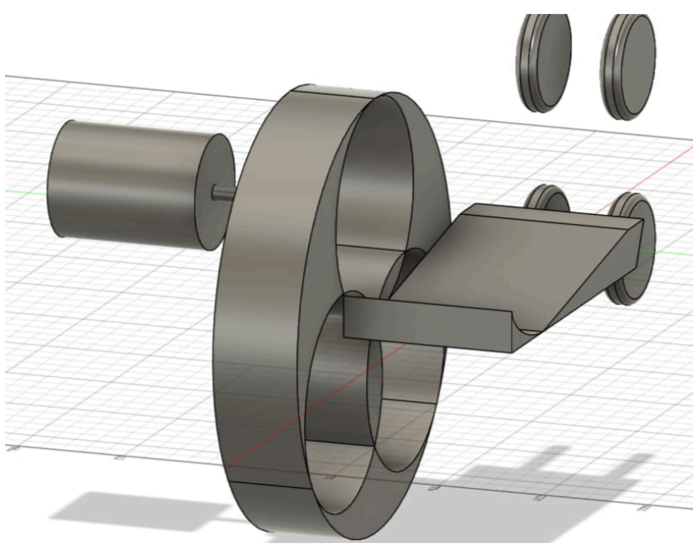


CAD modeling shaped a major part of our engineering process this season. The indexer for the artifacts is the best example: every dimension, angle, and mounting point was first created in CAD so we could test clearances and refine how the pieces interacted before ever 3D printing material. This let us identify issues early and ensure that the final physical build matched our design intent. Using CAD as the foundation of our workflow made the robot easier to assemble, and far more precise than relying on trial-and-error alone.



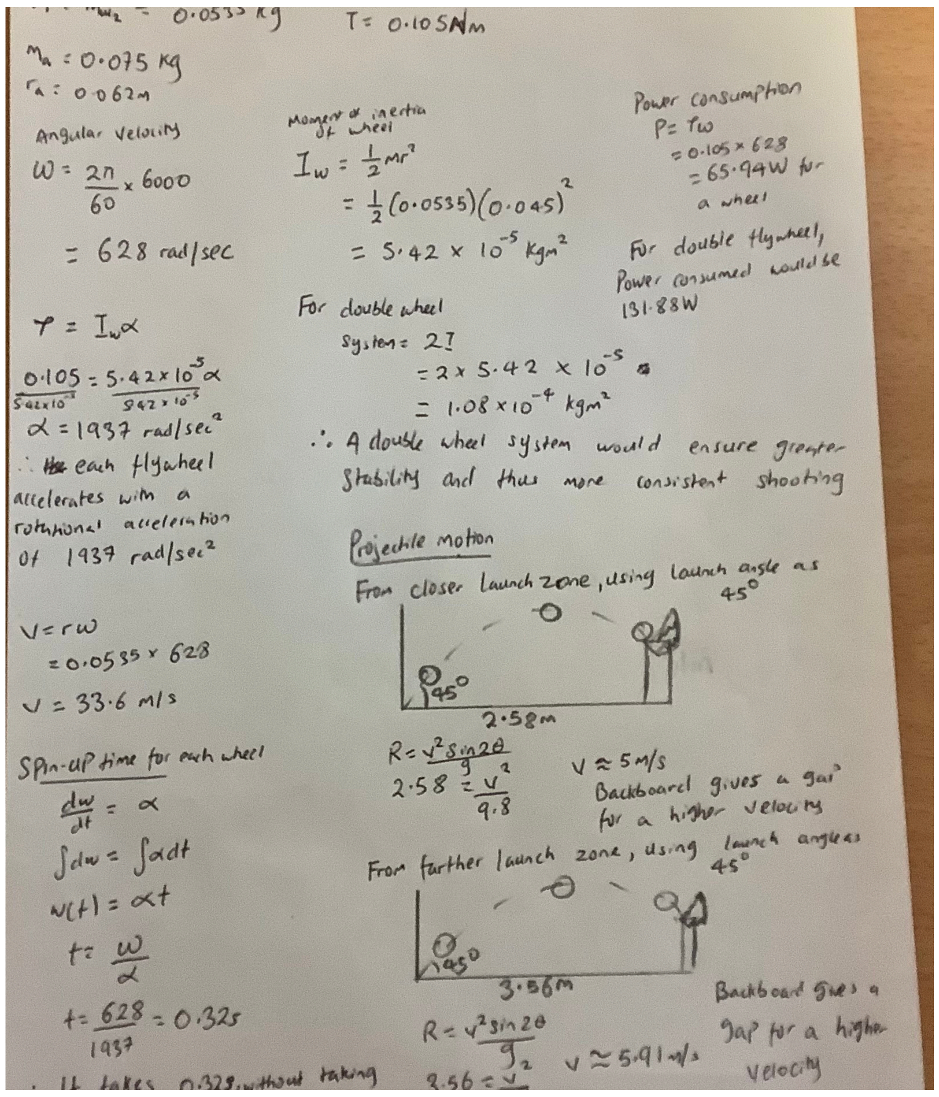


3D printing made it simple for us to turn ideas into real parts without a long build process. Once something was modeled, we could print it the same day and test how it fit on the robot, then tweak the design and reprint if needed but by the use of thermoplastics we could reuse filament. This helped us make custom pieces that would've been difficult to make by hand, and it let us experiment without worrying about wasting materials. Being able to go from a quick CAD sketch to a physical part easily, kept our design process fast and flexible.



CALCULATIONS

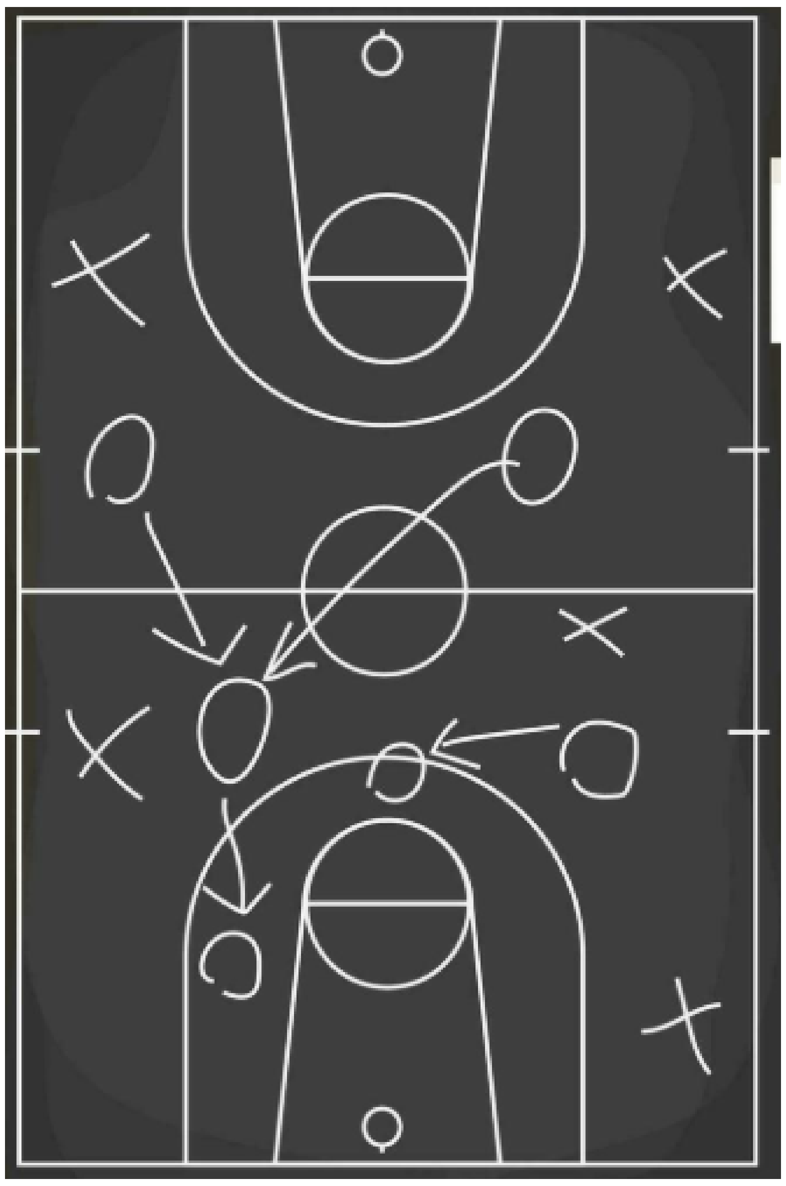
After figuring out certain parameters such as the angular velocity of the flywheel and the torque of its wheel, we were able to determine a series of quantities that would affect the flywheel shooter. Using Newton's rotational law, we were able to quickly figure out the angular acceleration of the flywheel as 1937 rad/sec^2. We then used this to find out the spin-up time of the wheel as 0.32s which we expect to be very similar to the actual spin-up time of the flywheel from rest to full speed, as air resistance is quite negligible in this scenario. Such a spin-up time is what we consider quite good for its purposes. After some discussion as a team, we decided to go for a dual-flywheel shooter with variable angle rather than the initially planned hooded single flywheel shooter.



This is because a dual flywheel shooter would have double of a single flywheel's moment of inertia. This means that it would be more stable and would make more consistent shooting. It can also be easily controlled and reconfigured while a hood can be very hard to adjust. A hood would also lead to more loss of energy through friction. A downside is that the dual-flywheel shooter would consume double the power from the battery, but such a trade-off was easily accepted. The projectile motion of the artifact was also modeled, from both launch zones to have a blueprint of what might actually happen

GAME PLANS

During competition, our team is prepared to respond quickly and effectively if the robot experiences mechanical or software issues. We designed both our code and hardware with modularity in mind, allowing us to isolate problems and replace or adjust components without needing to dismantle the entire robot. Our simplified mechanical layout makes it easier to identify faults, while our modular code structure lets us debug individual subsystems independently, reducing downtime between matches. By assigning clear troubleshooting roles and practicing rapid repairs throughout the season, we ensure that we can diagnose issues under pressure and return the robot to full functionality as efficiently as possible.



If our shooter system becomes unusable during a match, we immediately shift our focus to supporting our alliance in the most effective way possible. Our priority is always to help our alliance come out on top, so we adapt our role to maximize overall scoring potential. In this situation, we concentrate on assisting with artifact handling—collecting, transporting, and positioning artifacts to keep our partner's scoring cycle running smoothly. By staying flexible and communicating clearly with our alliance, we ensure that even without our primary scoring mechanism, we remain a strategic asset who can help maintain match momentum and contribute to a winning outcome.

CONNECTING

SCRIMMAGE

During the 4th of December, 2025, our team as well as several others from Scotland had all met up inside the HMFC football stadium for a tech scrimmage. This gave our team the opportunity to socialise and connect with other team and work on our robot. We have leant things from the scrimmage that has influenced our design deeply.

OPEN ROBOT SESSION

We hosted open robot sessions for the younger students at our school to come and learn about the robot and its functions. We were explaining to them what FTC was about, its core values and the aim of this year’s DECODE challenge. They really enjoyed it as they got the chance to drive our robot around and also turned on the feeder system to try pick up the artifacts. Overall, they had a lovely time and were all happy to have a shot at driving the robot!

GOALS

INSPIRE YOUNGER GENERATIONS WITHIN ROBOTICS

To achieve this we hosted open robot events for the younger students to ask questions and find out what FTC is about.

USE CAD MODELING TO WORK ON MECHANISMS AND 3D PRINT THEM

We used CAD modelling for most of our designs and used our 3D printers to make them real. We also used sustainable materials like thermoplastics

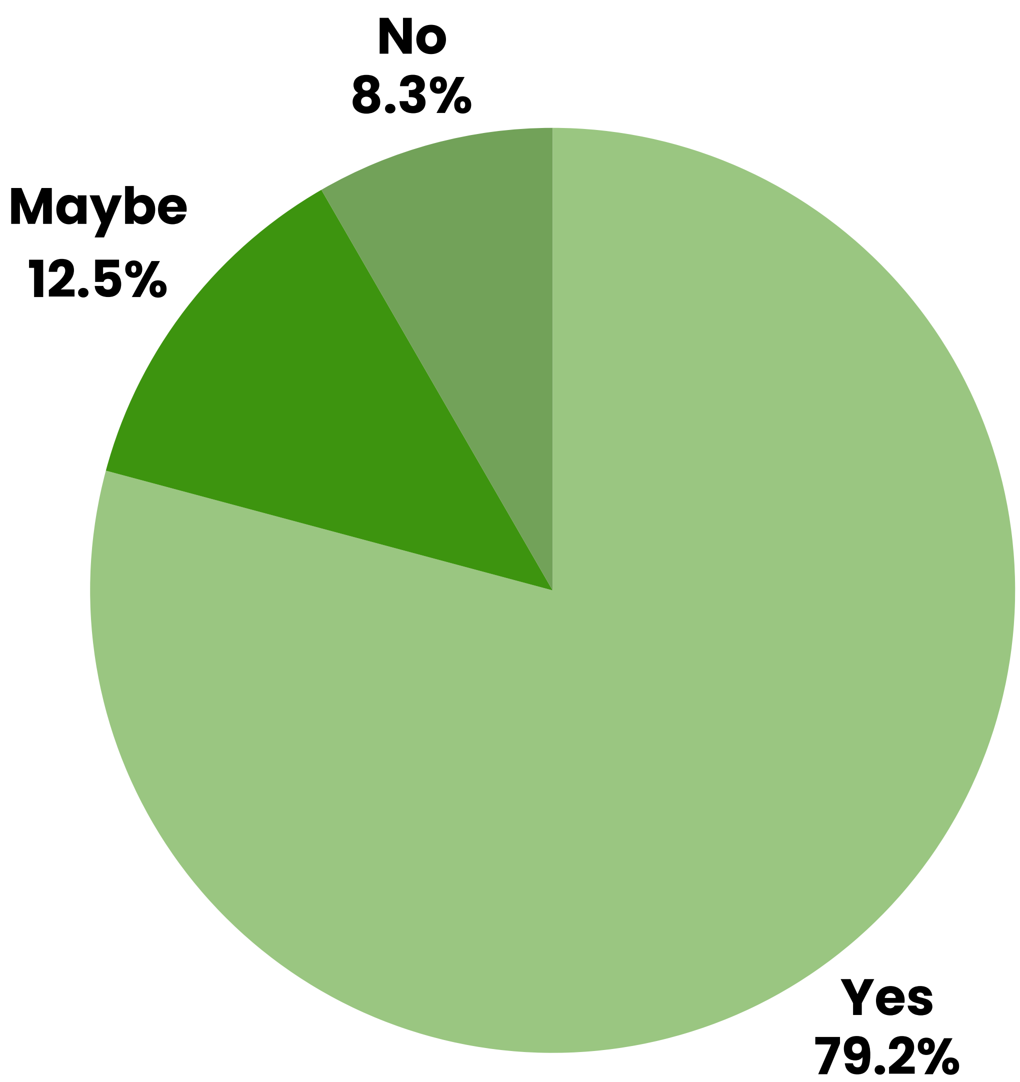
FIND SPONSORS

Hearts were kind enough to give us a jersey signed by the whole team. We were also looking for more sponsors, but time wasn't on our side and will hopefully have more for the nationals.

OUTREACH

We've been updating our engineering portfolio for the FTC Decode challenge, and we've added a bunch of new content to our website. Our latest robot design work is now uploaded, along with notes on what we've changed, what we're testing next, and what we've learned along the way. We also refreshed our team updates to highlight recent outreach, events we've taken part in, and some behind-the-scenes progress from our meetings. We'll keep posting updates as we go so everyone can follow along with our season.

We conducted a survey in the school with 48 pupils from S1 to S4. This is what they said:



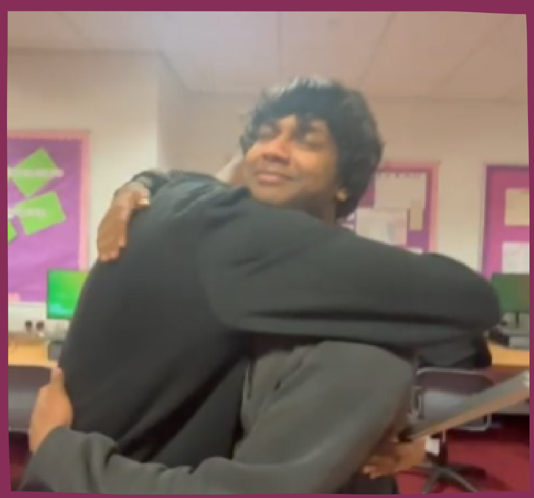
Would students join FTC next year after learning about our robot?

SOCIALS

← **tyniebrains**

2 posts | 28 followers | 8 following

"Big dreams. Big aims. Tynie brains."
Tynecastle High FTC team



so far, we have made 2 videos

with a total of 1.2k views

We have 38 likes

SUSTAINABILITY

RECRUITING

Our team places a strong emphasis on sustainable growth by actively recruiting younger students and introducing them to the world of robotics early. Rather than limiting our outreach to those already interested in STEM, we intentionally engage a wide range of students and create opportunities for them to explore robotics in an accessible, welcoming environment. Through demonstrations, open sessions, and hands-on activities, we aim to spark curiosity and build confidence, helping students discover interests they may not have realized they had. This approach not only strengthens our current team but also ensures a steady pipeline of enthusiastic, well-prepared members who can continue representing our school in FTC long after we graduate.

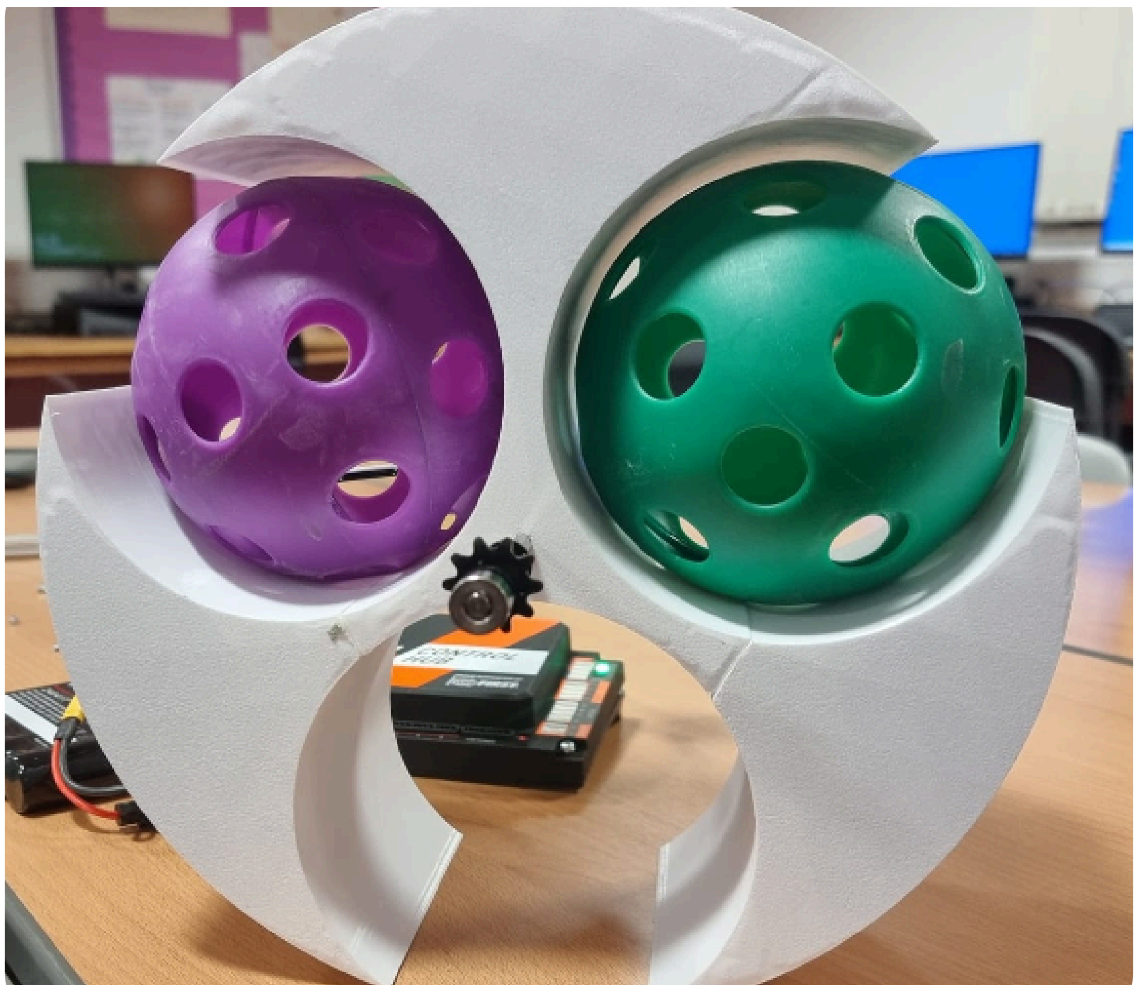
MATERIALS

To support long-term sustainability in our robot design, we prioritise the use of thermoplastics and other recyclable materials whenever possible. Thermoplastics allow us to prototype and manufacture components efficiently while also reducing waste, since many of these parts can be reheated, reshaped, or reused in future seasons. By intentionally selecting materials with lower environmental impact, we minimise our carbon footprint and model responsible engineering practices for younger team members. This approach not only benefits our current build cycle but also reinforces a culture of sustainability that we hope future FTC teams at our school will continue to develop.

Thermoplastic 3D Printing Filament

FINANCIALS

Our season began in December with a strong financial foundation thanks to the £1,000 we earned by winning a Raspberry Pi competition, which gave us the money to purchase essential components and begin prototyping. As the season progressed, we expanded our funding through community partnerships, most notably with Hearts FC, who generously sponsored us by donating signed jerseys for the team to sell. This support not only boosted our budget but also strengthened our connection to the local community. Together, these resources allowed us to operate sustainably, invest in higher-quality materials, and ensure that every team member had access to the tools they needed to contribute meaningfully.



Our team invested a significant portion of our budget this season, but much of that spending was intentional and strategic. A large share went toward thermoplastics and other reusable materials. These plastics can be reheated, reshaped, and repurposed in future FTC seasons, meaning the upfront expense directly reduces long-term material costs. Even though the initial outlay looked high, it strengthened our ability to experiment and build sustainably. This approach also helped us stretch our resources by turning this year's purchases into assets for next year's FTC robot rather than single-use components that will not be used after us.

FINANCIALS
BUDGET BREAKDOWN

AMAZON

	QTY	PRICE
Filament	3	£44.97
Cordless screwdriver	2	£8.98
Cable connectors	1	£39.90
Nut driver	2	£8.99

RS COMPS

	QTY	PRICE
Color sensor	1	5.93
Distance sensor	1	£23.66
VAT+Delivery		£14.26

REV

	QTY	PRICE
Smart Robot Servo	2	£42.88
75mm Mecanum Wheel Set	1	£126.13
Expansion Hub	1	£210.21
VAT+ Delivery		£92.01

LiMELiGHT

	QTY	PRICE
Limelight 3A	2	£288
Delivery		£27

TOTAL COST	£932.92

SOFTWARE

```java
package org.firstinspires.ftc.teamcode;
import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
import com.qualcomm.robotcore.hardware.Blinker;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.DcMotorSimple;
import com.qualcomm.robotcore.hardware.Servo;

@TeleOp(name = "MrRobot")
public class MrRobot extends LinearOpMode {

    private DcMotor FL, FR, RL, RR;
    private DcMotor launcher, fly;
    private Servo mainServo;
    private boolean isServoActive = false;

    @Override
    public void runOpMode() {
        // Hardware Map
        FL = hardwareMap.get(DcMotor.class, "FL");
        FR = hardwareMap.get(DcMotor.class, "FR");
        RL = hardwareMap.get(DcMotor.class, "RL");
        RR = hardwareMap.get(DcMotor.class, "RR");
        mainServo = hardwareMap.get(Servo.class, "mainServo");
        launcher = hardwareMap.get(DcMotor.class, "launcher");
        fly = hardwareMap.get(DcMotor.class, "fly");

        // --- MOTOR DIRECTIONS ---
        // 1. Lift robot. Push stick UP.
        // 2. If right wheels spin backwards, keep these as REVERSE.
        // 3. If right wheels spin forwards (but reversed is set), change to FORWARD.
        FL.setDirection(DcMotorSimple.Direction.FORWARD);
        RL.setDirection(DcMotorSimple.Direction.FORWARD);
        FR.setDirection(DcMotorSimple.Direction.REVERSE);
        RR.setDirection(DcMotorSimple.Direction.REVERSE);

        // Brakes help mechanum accuracy
        FL.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
        FR.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
        RL.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
        RR.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);

        telemetry.addData("Status", "Initialized");
        telemetry.update();

        waitForStart();
```

motors and servos are initialised and modules are imported from official ftc libraries

brakes to help with accuracy espeically with high speeds

```java
        while (opModeIsActive()) {

            // --- CONTROLS ---
            double y = -gamepad1.left_stick_y; // Forward/Back
            double x = gamepad1.left_stick_x * 1.1; // Strafe (Left/Right)
            double rx = gamepad1.right_stick_x; // Turn

            // --- MATH ---
            // Denominator ensures we don't exceed power limit of 1.0
            double denominator = Math.max(Math.abs(y) + Math.abs(x) + Math.abs(rx), b: 1);

            double flPower = (y + x + rx) / denominator;
            double blPower = (y - x + rx) / denominator; // Back Left
            double frPower = (y - x - rx) / denominator;
            double brPower = (y + x - rx) / denominator; // Back Right

            // --- OUTPUT ---
            FL.setPower(flPower);
            RL.setPower(blPower);
            FR.setPower(frPower);
            RR.setPower(brPower);
```

as non autonomous op mode turns on, controls of the 4 motors are assigned to the controller

normalisation of the throttle
so that it is not greater than 1

```java
        // --- OTHER MECHANISMS ---
        if (gamepad1.dpad_up) launcher.setPower(1);
        else if (gamepad1.dpad_down) launcher.setPower(0);

        if (gamepad1.dpad_left) fly.setPower(1);
        else if (gamepad1.dpad_right) fly.setPower(0);

        if (gamepad1.y) {
            mainServo.setPosition(0);
        } else if ((gamepad1.x || gamepad1.b) && !isServoActive) {
            isServoActive = true;
            new Thread(() -> {
                mainServo.setPosition(1);
                try { Thread.sleep(millis: 300); } catch (InterruptedException e) {}
                mainServo.setPosition(0);
                isServoActive = false;
            }).start();
        }
      }
    }
}
```

buttons are assigned to various functions like activating the flywheel and launcher motors

multi threading is used to allow for simultaneous use of servos and motors and to improve efficiency

END OF OUR ENGINEERING PORTFOLIO
TEAM TYNIE BRAINS