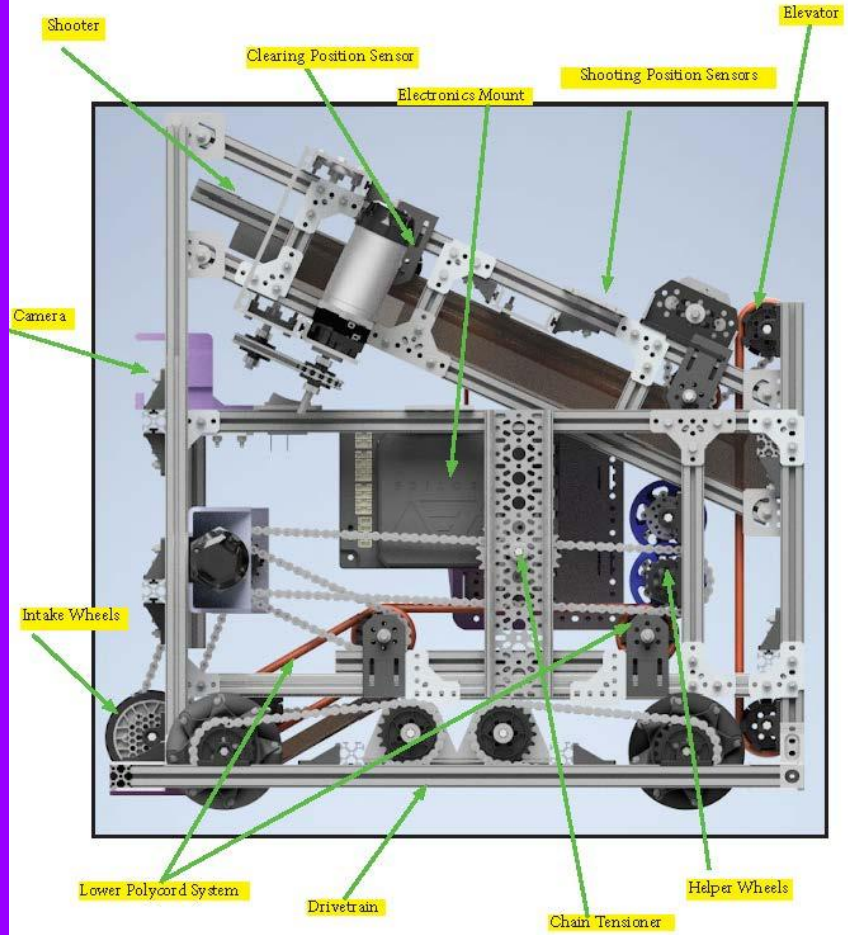

PirateTekerz
PT
11109

Parts of our Robot



Strategy	Analasis	Design	Construction	Testing	Reflection

Date: Nov 17, 2020

Author(s): Caleb

Contributors: Caleb, Finn, Braden, Lucas, Nathan, Mentor Dave, Mentor Chris, Mentor Jessica

Tasks Accomplished

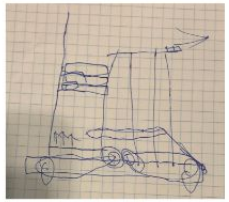
- We all showed our designs and discussed them
- We then made pro and con lists for each item

Next Tasks

- We will on friday at least eliminate one of our ideas and mabye make a final decision

Different designs

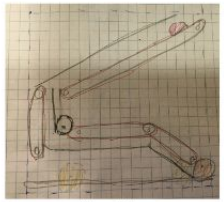
- Elevator (Lucas)
- Polycord (Caleb)
- Ramp (Finn)



Elevator Idea

This is lucas' idea, the rings would be brought to the back of the robot via. a polycord conveyor. When they get to the back of the robot they would then drop onto an elevator. Then the rings would be lifted up to the top of the robot and a servo would push them into the shooter.

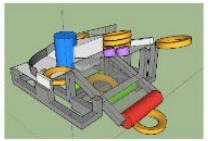
This was my idea, the ring would first be sucked up by the intake roller. Then the ring would be pulled along by polycord to the back of the robot. Then the ring would turn and by sucked up with more polycord. After the ring is 10 inches up it would make another turn and move to the shooter. The polycord on the way to the shooter would be under the ring. Once the ring is at the shooter it would be shot. The ring would be indexed in the lower part of the system and the upper part of the system would move the ring quickley to the shooter.

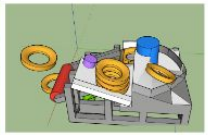


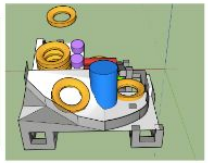
Polycord idea

Ramp idea

This was finns idea, It would have a flip down intake that would suck the rings up into the robot. The rings would be conveyed to a ramp with two rollers. When the rings get to the ramp they would be brought around a 180 degree turn with a spinning column and up to the top of the robot. Then the rings would drop into a stack. The rings would then be pushed one by one into the shooter.







Reflections

I think that our team is making good progress and that we will be able to accomplish our goal of eliminating one idea on friday.

Initials _____ **Approved** ____

Robot needs:	Time period	Easy	Medium	Hard
All 3 Shooter Intake Reliable robot	Auto	1. Mid goal 2. Park 18+5=23	1. High goals 2. *Wobble* 3. Park 36+15+5=56	1. *Wobble* 2. Power shot 3. Park 15+45+5=65
2 Wobble goal device	Tele-Op	1. Mid goal (6*3)*4=72	1. High Goals 2. Move wobble (6*3)*6=108	1. High Goals (6*3)*6=108
	End game	1. Mid goal (2*3)*4=24	1. Drop zone 2. High goals (6*3)+20	1. Power shots 2. Drop zone 45+20=65
	Estimated Score:	119	202	238

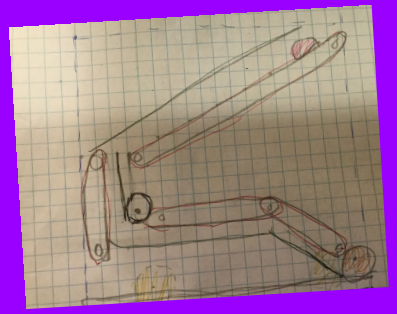
Drivetrain pros\cons

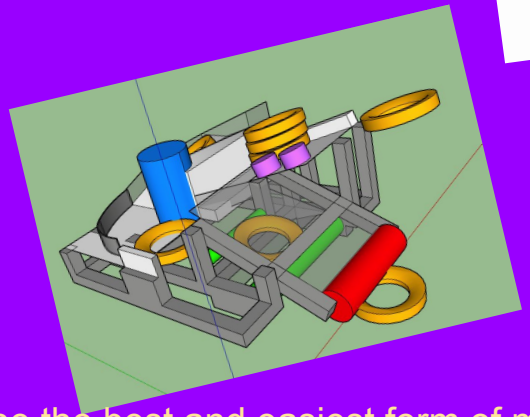
cons

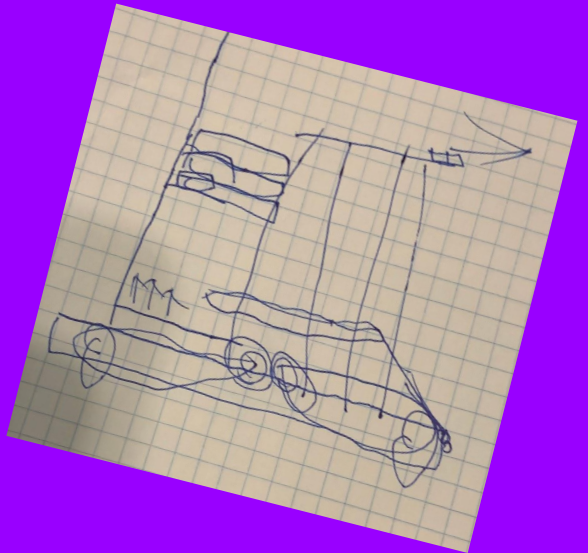
1. Swerve is eliminated
2. Tank
   a. Not omnidirectional
   b. Hard to get rings when they are near the wall which they often are
3. Mecanum
   a. We have never made one that worked before
   b. Weight distribution
   c. Single motor for each wheel
4. Holonomic
   a. Never done one before
   b. Weird frame
   c. One motor for each wheel
   d. Low traction
5. Slide
   a. Had issues with getting traction in middle (fixable)

Pros

1. Tank
   a. Simple
   b. Fast
   c. Diversity of wheel choice
   d. Motors chainable
2. Mecanum
   a. Omnidirectional
   b. Fast
3. Holonomic
   a. Omnidirectional
   b. Fast
   c. Simpler than mecanum
   d. Has the easiest strafing
4. Slide
   a. Done one before
   b. Omnidirectional
   c. Simple
   d. Can revert to tank if breaks down
   e. Motors chainable
   f. Fast

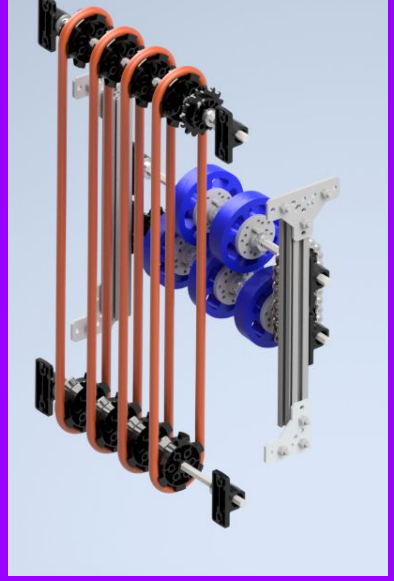


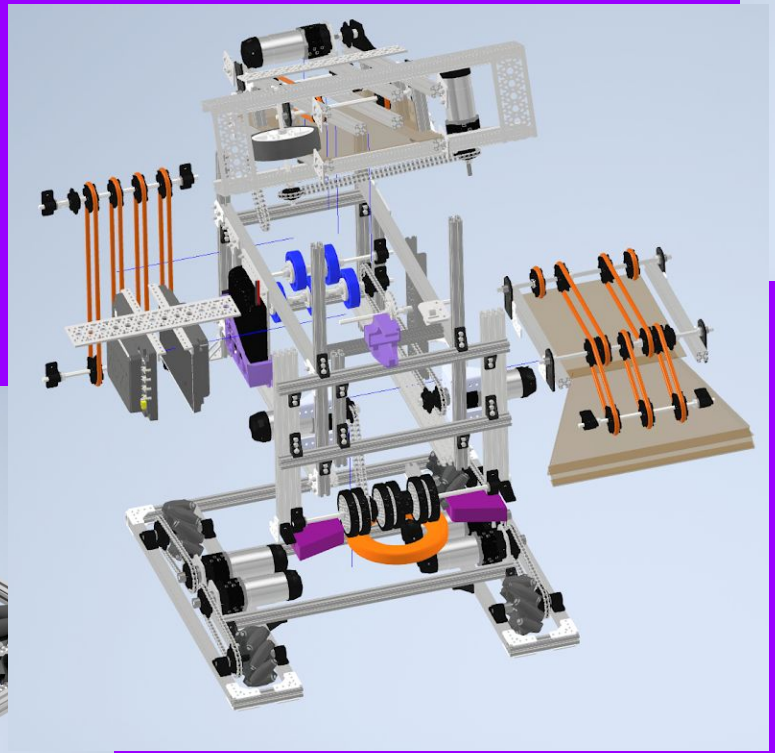


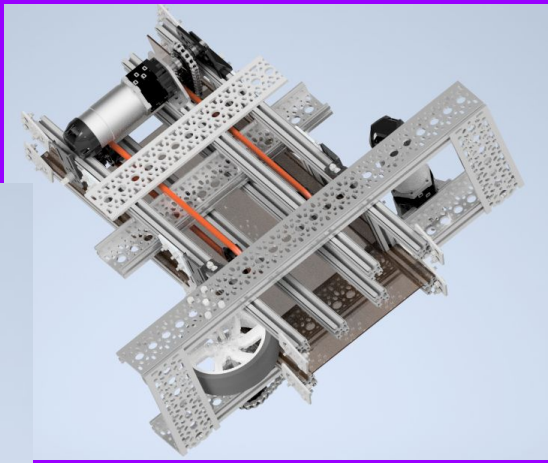


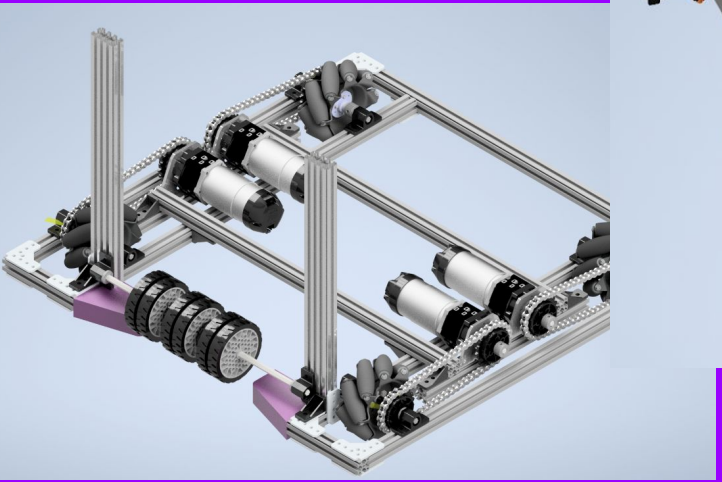
High goals would be the best and easiest form of making points throughout the entirety if the game. We focused here to ensure greatest potential with unknown future.

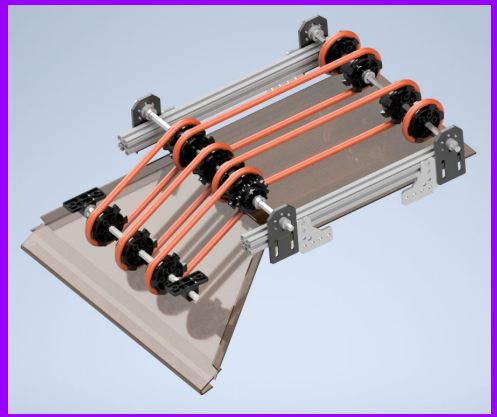
Exploded robot



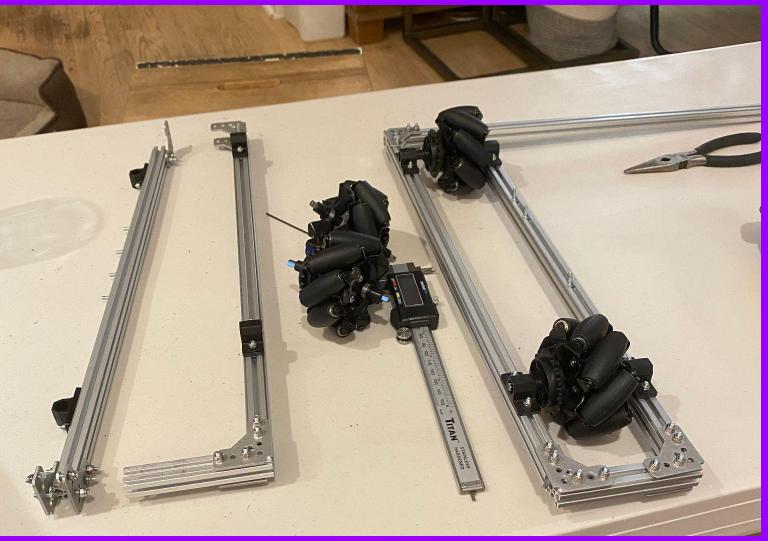


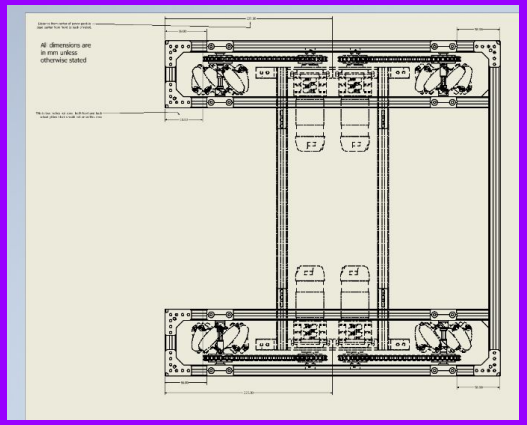


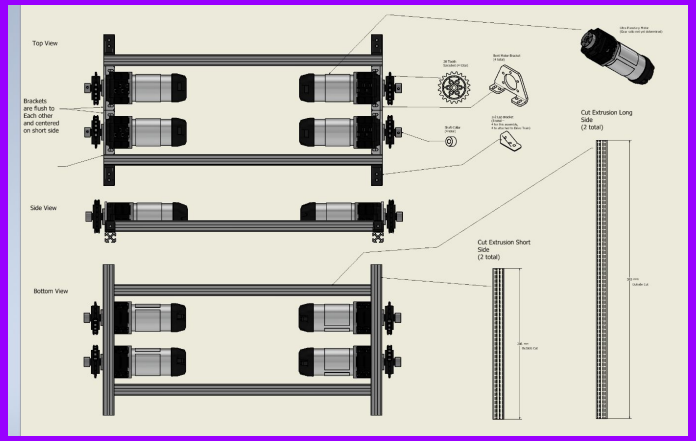




Finished Drivetrain

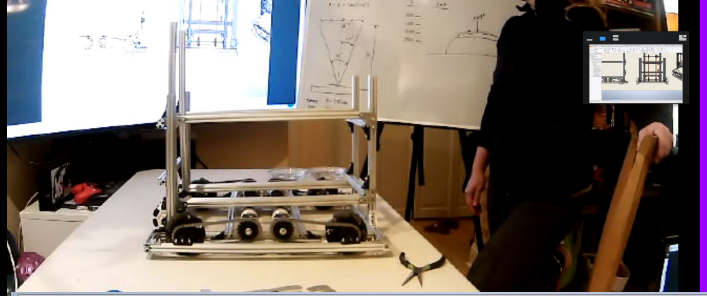


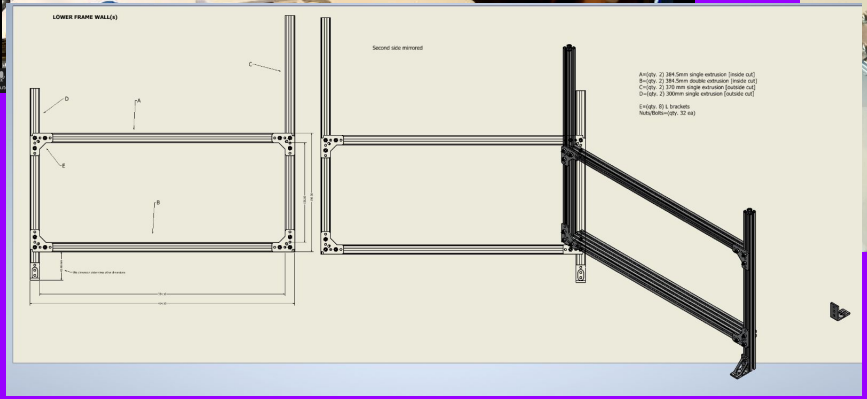


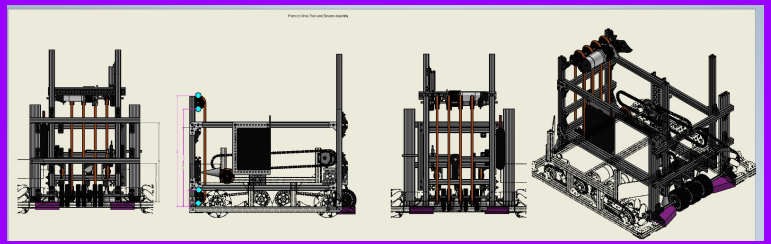


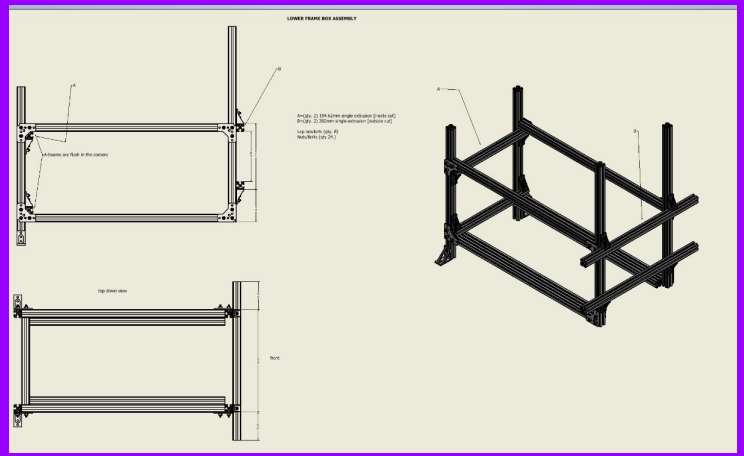
Cut list

Priority		Part Type	Quantity	Sizes	Cut Type
1	Intake	5mm Hex Shaft	1 ea	269.2 mm	Outside Cut
1	Intake	Double Extrusion	2 ea	265 mm	Inside Cut
1	Base Frame	Double Extrusion	2 ea	384.5 mm Could cut 3 from 1m and 1 from	Inside Cut
1	Base Frame	Single Extrusion	2 ea	384.5 mm	Inside Cut
1	Base Frame	Single Extrusion	2 ea	370mm	Outside Cut
1	Base Frame	Single Extrusion	4 ea	300mm	Outside Cut
1	Base Frame	Single Extrusion	2 ea	194.62 mm	Inside Cut
2	Shooter Frame	Single Extrusion	4 ea	420 mm	NO CUT
2	Shooter Frame	Single Extrusion	3 ea	394 mm	Outside Cut
2	Shooter Frame	Single Extrusion	4 ea	54.4 mm	Inside Cut
2	Shooter Frame	C-channel 15mm	1 ea	248 mm	NO CUT



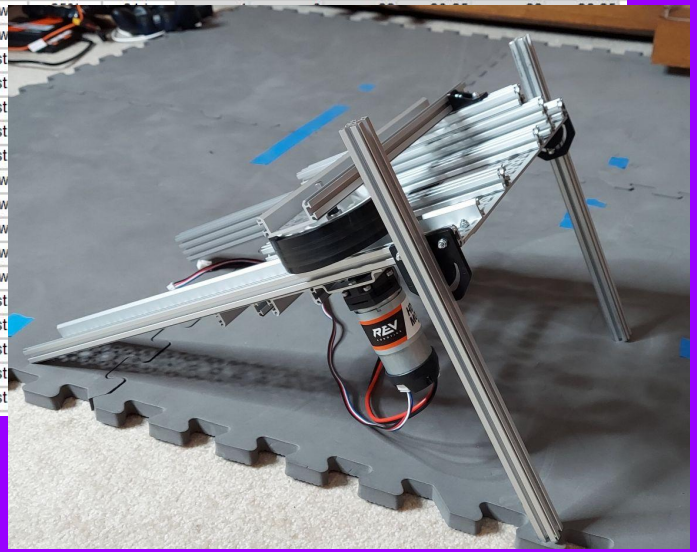


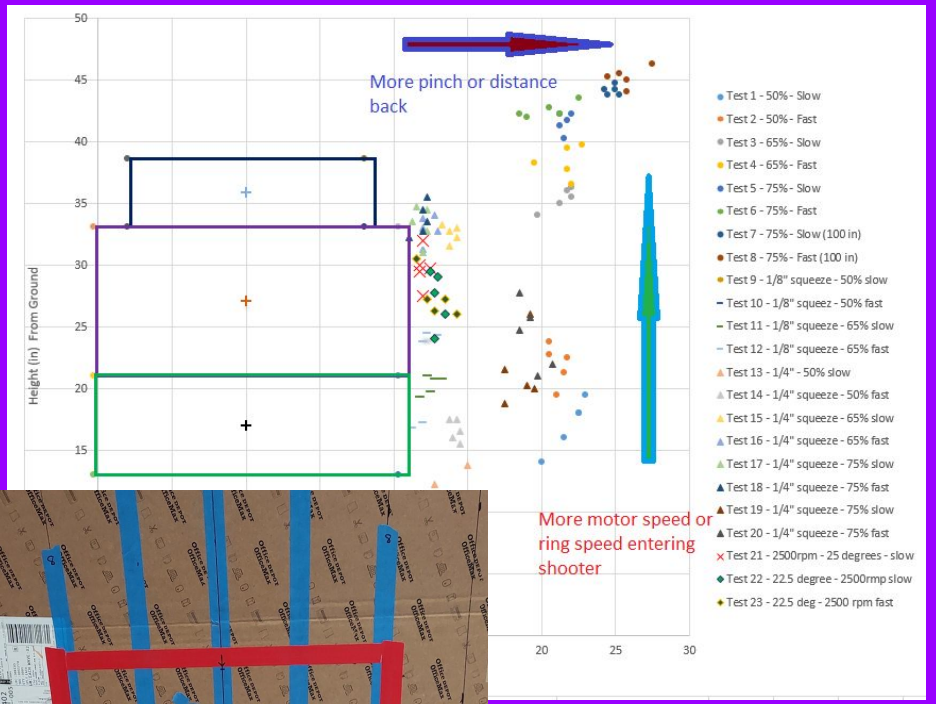




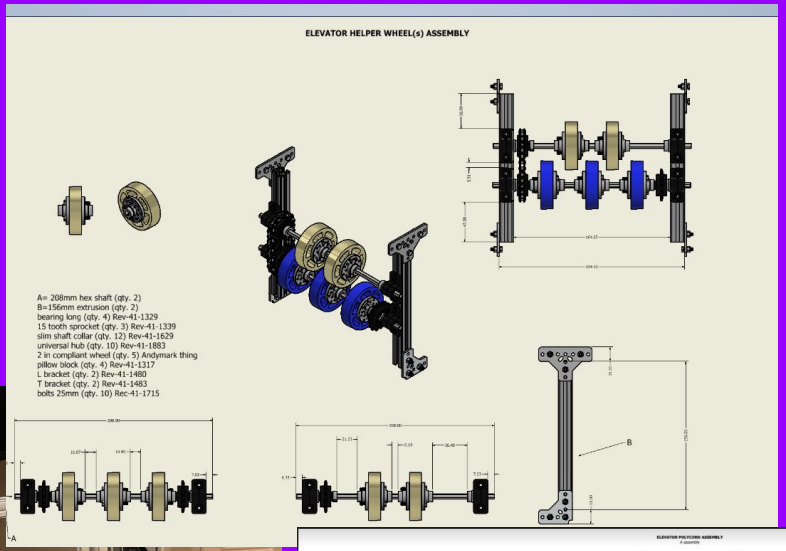
Building the frame

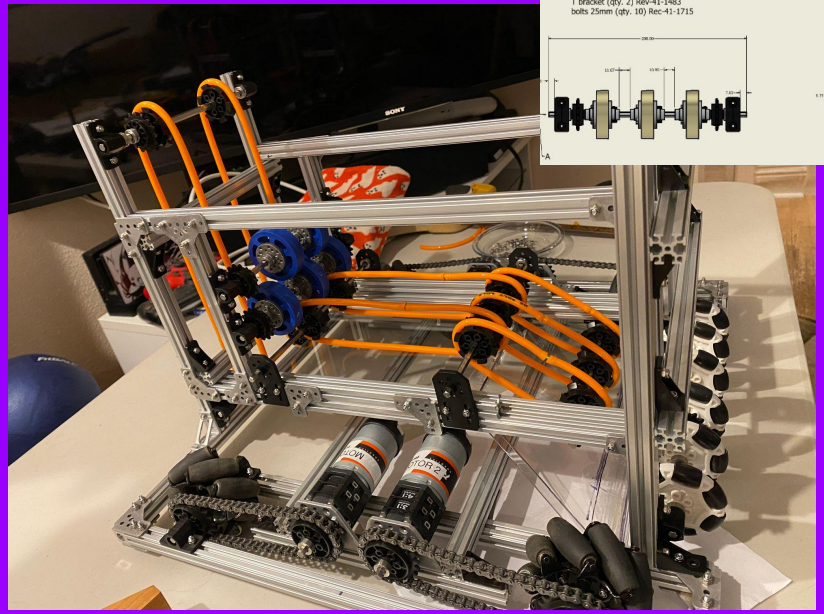
Test	Input Speed	Motor Speed	Distance from wal	Shot Iterration	landed distance from Wall	inches L/R from center	distance from ground	inches L/R from center true	distance from ground
base	---	---	---	0	0	0	0	0	0
1	Slow	50%	64 in	1	0	20	8	20	14
1	Slow	50%	64 in	2	0	22.5	12	22.5	18
1	Slow	50%	64 in	3	0	21.5	10	21.5	16
1	Slow	50%	64 in	4	0	22.5	12	22.5	18
1	Slow	50%	64 in	5	0	23	13.5	23	19.5
2	Fast	50%	64 in	1	0	21	13.5	21	19.5
2	Fast	50%	64 in	2	0	21.5	15.25	21.5	21.25
2	Fast	50%	64 in	3	0	20.5	16.75	20.5	22.75
2	Fast	50%	64 in	4	0	21.75	16.5	21.75	22.5
2	Fast	50%	64 in	5	0	20.5	17.75	20.5	23.75
3	Slow	65%	64 in	1	0	19.75	28	19.75	34
3	Slow	65%	64 in	2	0	21.25	29	21.25	35
3	Slow	65%	64 in	3	0	21.75	30	21.75	36
3	Slow								
3	Slow								
4	Fast								
4	Fast								
4	Fast								
4	Fast								
4	Fast								
5	Slow								
5	Slow								
5	Slow								
5	Slow								
5	Slow								
6	Fast								
6	Fast								
6	Fast								
6	Fast								
6	Fast								

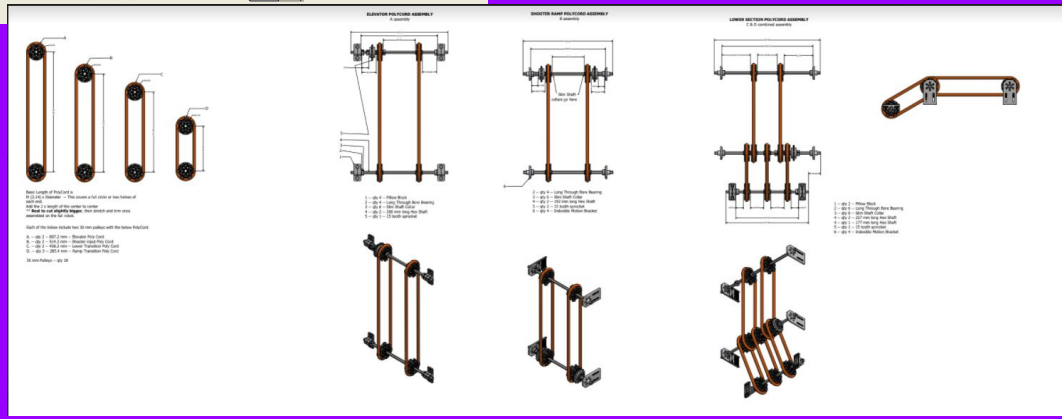




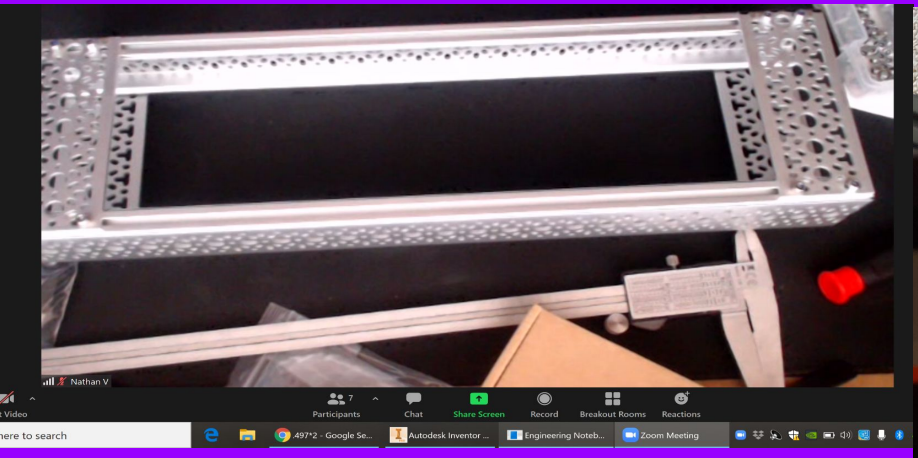
testing shooter prototype

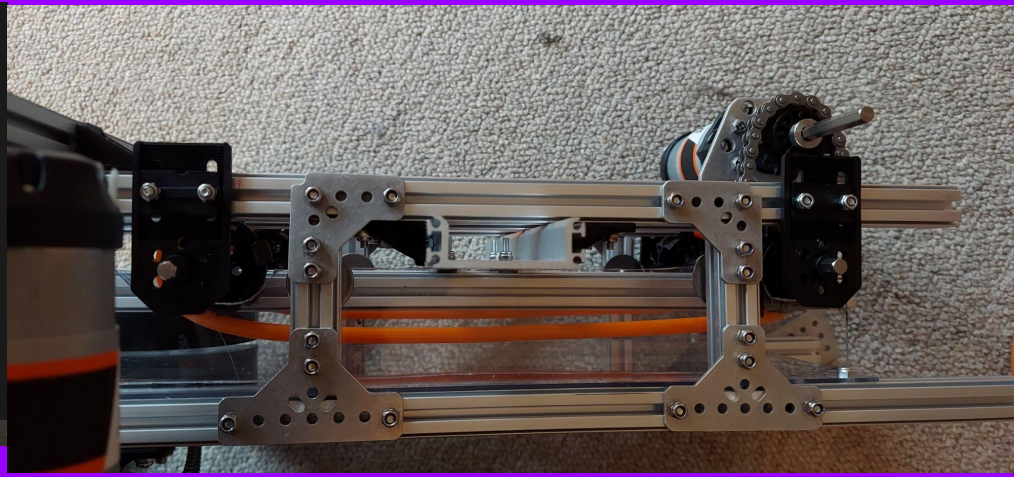


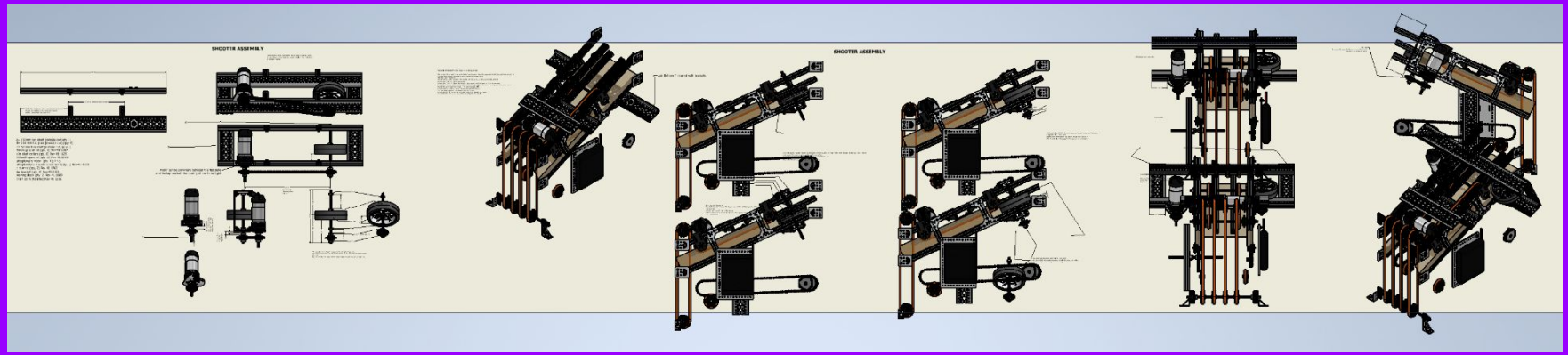




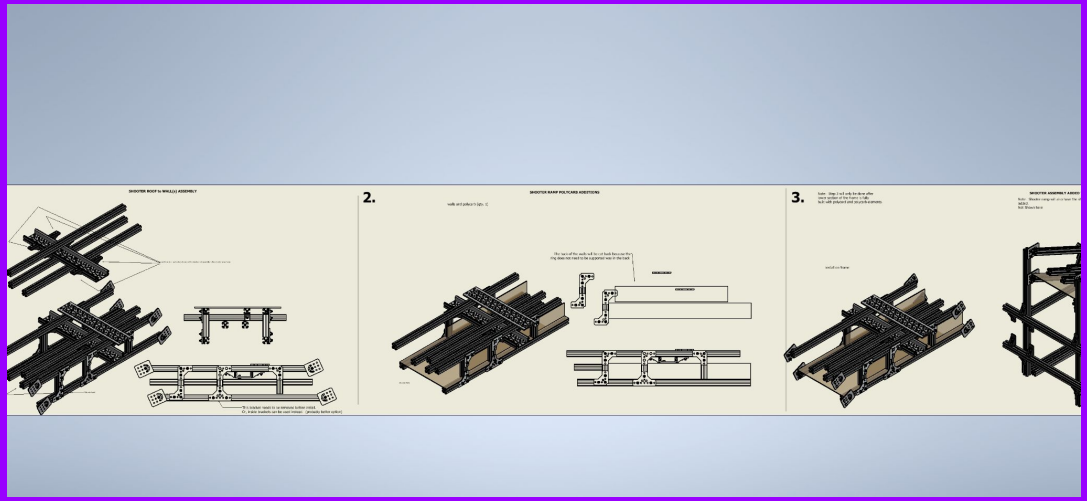
Got the polycord on.



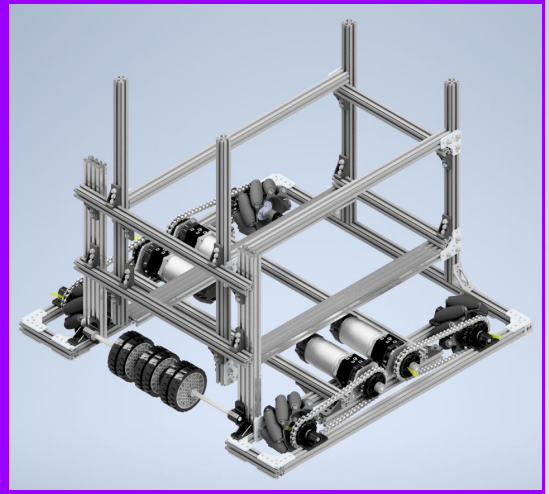


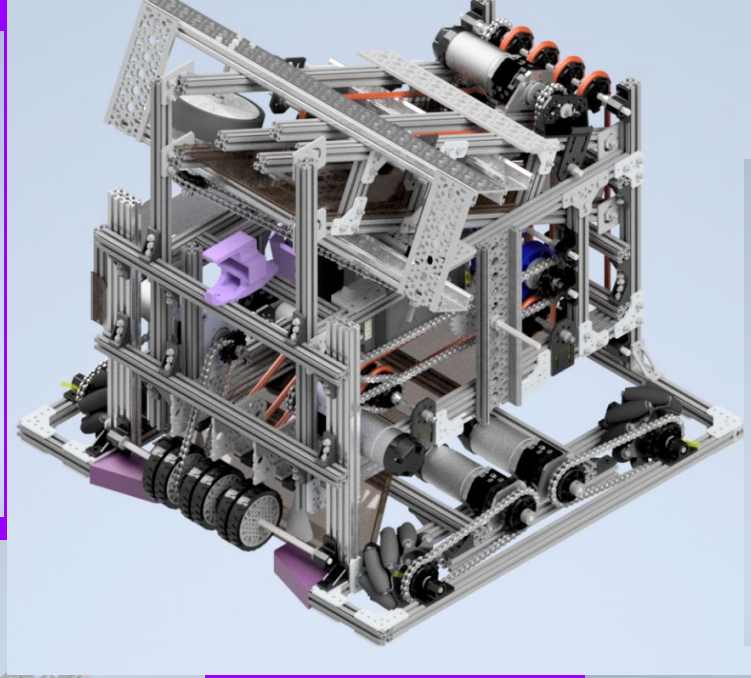


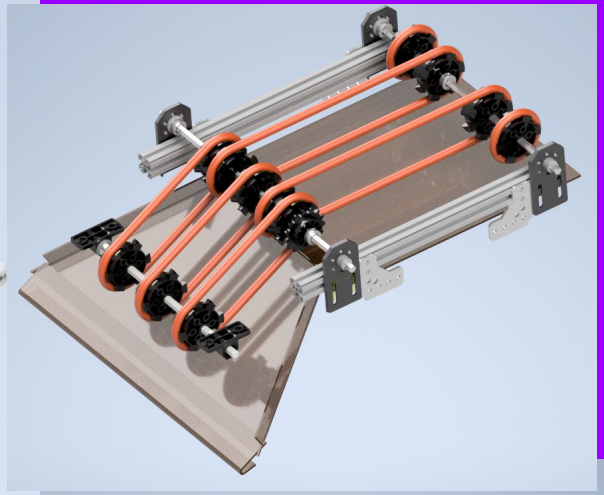
Started building the shooter

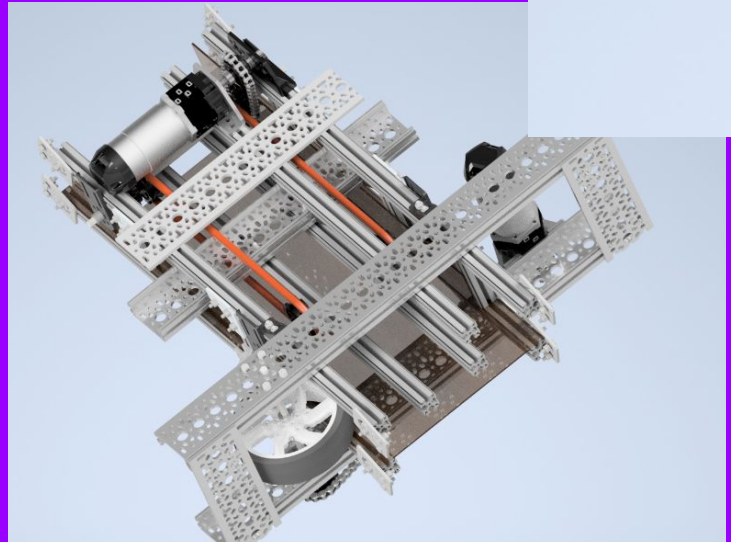


Put the shooter on.

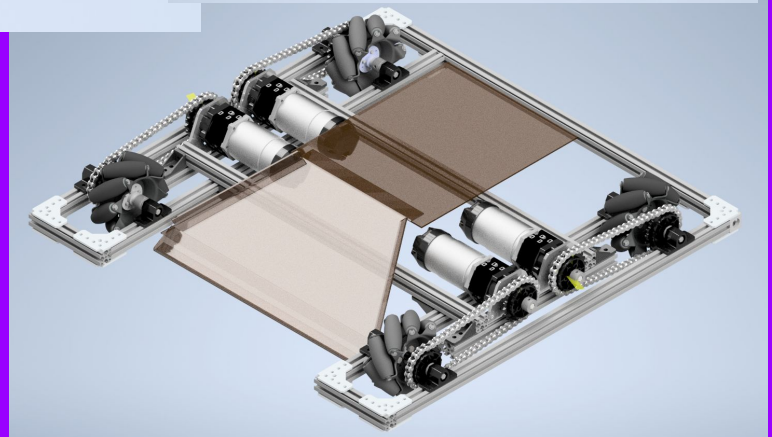


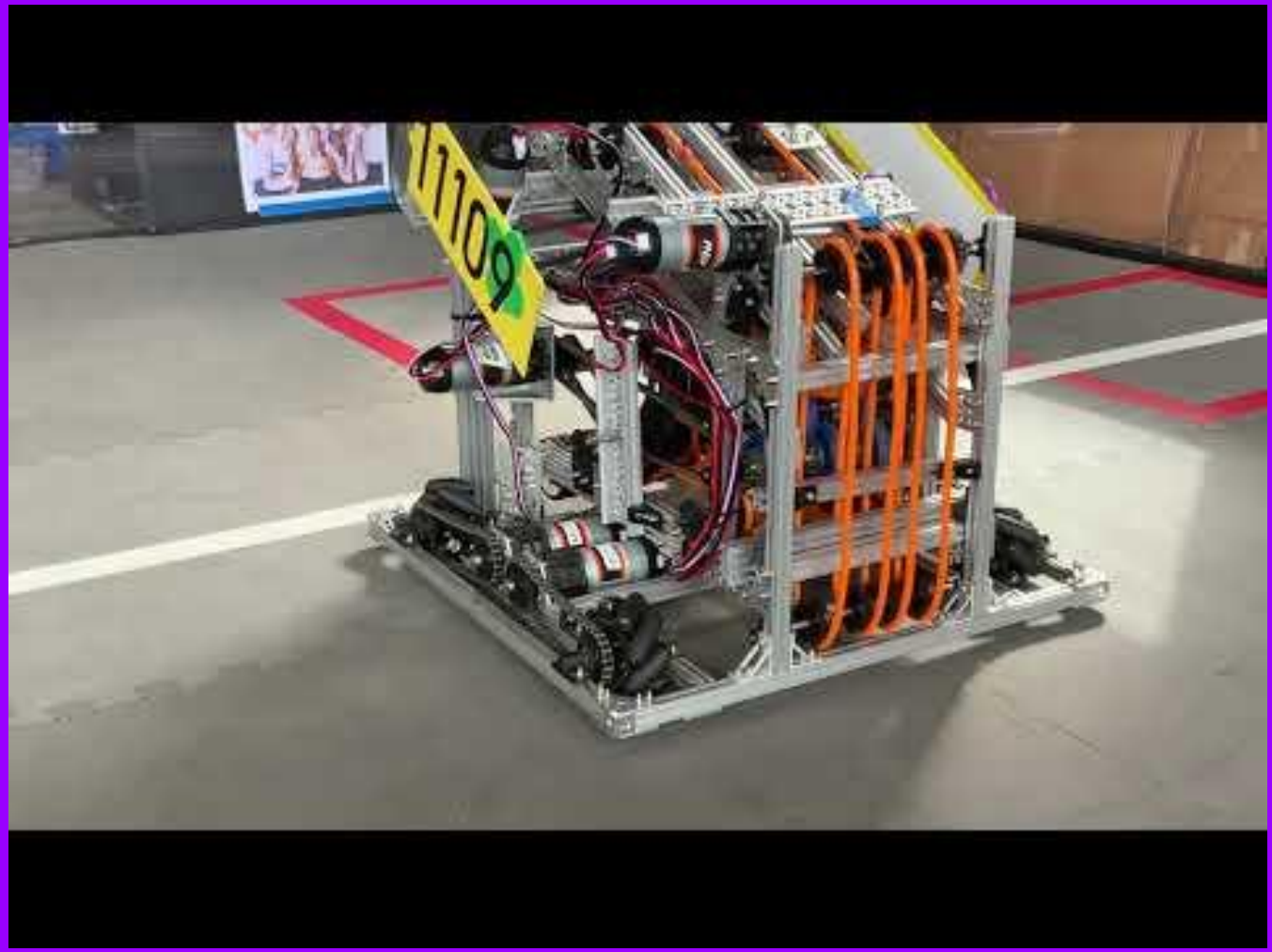






examples of CAD




1109

Strategy	Analasis	Design	Construction	Testing	Reflection

Date: Dec 15, 2020

Author(s): Caleb

Contributors: Caleb, Finn, Braden, Lucas, Nathan, Mentor Dave, Mentor Chris

Tasks Accomplished

- Talking about Nathan's prototype
- Building questions
- CAD
- Finn's sensors
- Lucas's Programming auto mapping

Next Tasks

- Nathan needs to change the wall height and add stuff on the bottom to make it stronger. He also needs to wire it up and shoot it tonig
- Braden will continue to work on the drivetrain
- I am going to incorperate mentor david's ideas on the CAD file
- Lucas will work on maooing out out Auto
- Finn-- refine sensor ideas, and work on CAD

Nathan's shooter prototype

The last one was a lot heavier. He is using plastic instead of cardboard on the floor now which seems to work better.



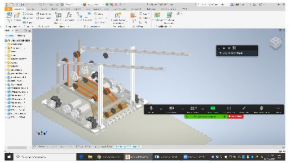
Building questions

Drivetrian

Braden and his parents are building the drivetrain at their house, and they had some questions about why their shafts had so much slop. We eventually found that they were using the wrong shaft collars.

CAD

I was able to get a considerable amount done. I think that the CAD file it getting pretty close to comletion. I moved the back towers in and changed my whole method of holding the polycord sprokets down. I also did some other minor changes to the power pack and lower pexiglass.



CAD

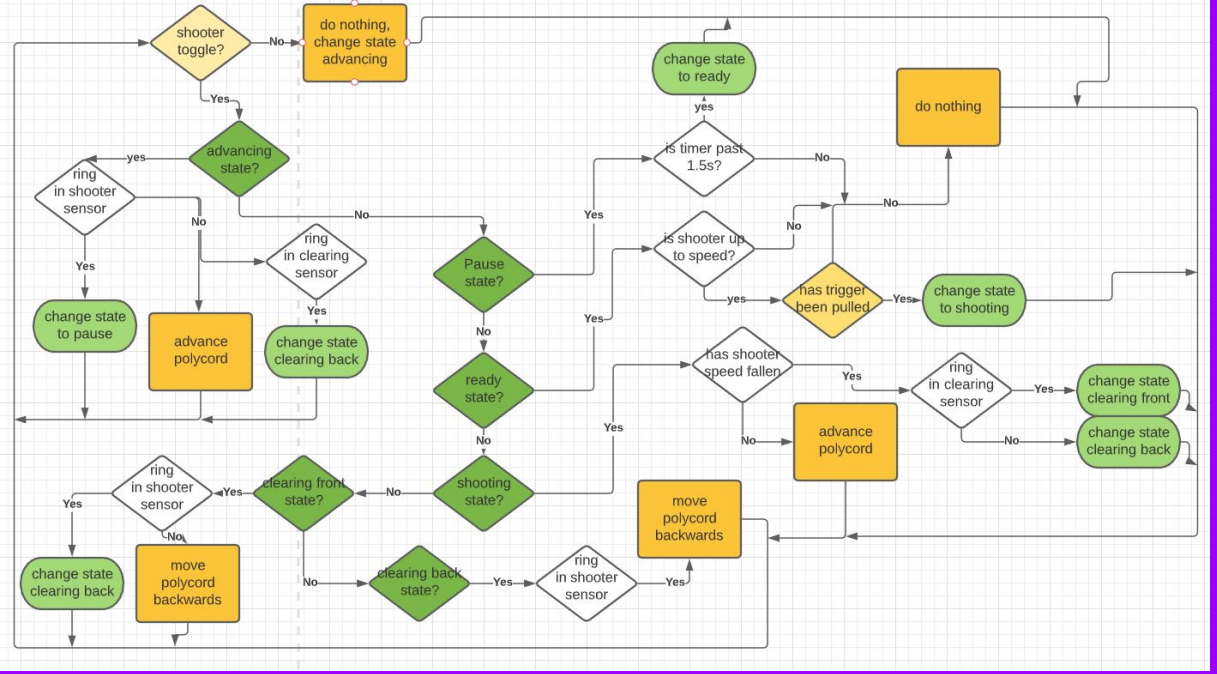
Finn's sensors

Sensor diagram

Finn made some progress on mapping out where the sensors will be. He will refine this and work on CAD.

Reflections

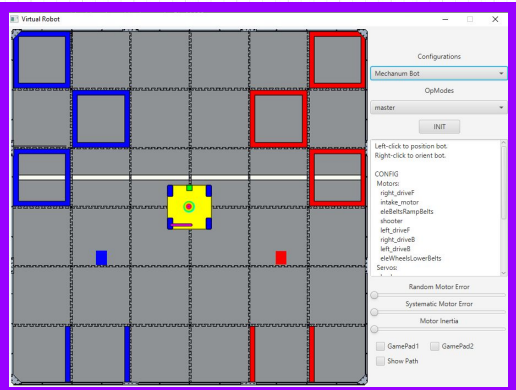
Initials _____ **Approved** ____



State machine flow chart

```
if (isShooterOn) {
    if (transitionState == transitionShooterMode.Advancing) {
        if (disSensors.isRingInEle()) {
            transitionState = transitionShooterMode.Pause;
            timeOut.reset();
        } else if (disSensors.isRingInForward()){
            transitionState = transitionShooterMode.ClearingFront;
            timeOut.reset();
        } else {
            transition.advancingTransitionMode();
        }
    }
    if (transitionState == transitionShooterMode.Pause) {
        if (pauseTransitionTime.milliseconds() > 500) {
            transitionState = transitionShooterMode.Ready;
            timeOut.reset();
        } else {
            transition.doNothingMode();
        }
    } else {
        pauseTransitionTime.reset();
    }
    if (transitionState == transitionShooterMode.Ready) {
        if (shooter.isUpToSpeed() && controls.shootToggle()) {
            transitionState = transitionShooterMode.Shooting;
            shooter.ResetMinMax();
            timeOut.reset();
        } else {
            transition.doNothingMode();
        }
    }
    if (transitionState == transitionShooterMode.Shooting) {
        if (!shooter.isUpToSpeed()) {
            timeOut.reset();
            if (disSensors.isRingInForward()){
                transitionState = transitionShooterMode.ClearingFront;
            } else {
                transitionState = transitionShooterMode.ClearingBack;
            }
        } else {
            transition.shootingTransitionMode();
        }
    }
    if (transitionState == transitionShooterMode.ClearingFront) {
        if (disSensors.isRingInEle()|| timeOut.milliseconds() > 2000) {
            transitionState = transitionShooterMode.ClearingBack;
            timeOut.reset();
        } else {
            transition.upperTransitionOuttake();
            transition.lowerDoNothing();
        }
    }
    if (transitionState == transitionShooterMode.ClearingBack){
        if (!disSensors.isRingInEle() || timeOut.milliseconds() > 2000){
            transitionState = transitionShooterMode.Advancing;
            timeOut.reset();
        } else {
            transition.upperTransitionOuttake();
            transition.lowerDoNothing();
        }
    }
} else {
    transitionState = transitionShooterMode.Advancing;
    timeOut.reset();
}
```

Auto shoot



Virtual test environment

```
public void drive(double x, double y, double rx, double gyroAngle) {


    double leftFP;
    double leftBP;
    double rightFP;
    double rightBP;

    double xIn = x;
    double yIn = y;

    // Compensate for gyro angle.
    double rotated[] = rotateVector(xIn, yIn, gyroAngle);
    xIn = rotated[0];

    //throttle is used to make up for the faster speeds when the robot moves laterally
    yIn = rotated[1];


    leftFP = yIn + xIn + rx; // FP meaning Front Power and BP meaning Back Power
    leftBP = yIn - xIn + rx;
    rightFP = yIn - xIn - rx;
    rightBP = yIn + xIn - rx;


    if (Math.abs(leftFP) > 1 || Math.abs(leftBP) > 1 ||
            Math.abs(rightFP) > 1 || Math.abs(rightBP) > 1) {
        double max = 0;
        max = Math.max(Math.abs(leftFP), Math.abs(leftBP));
        max = Math.max(Math.abs(rightFP), max);
        max = Math.max(Math.abs(rightBP), max);

        leftFP /= max;
        leftBP /= max;
        rightFP /= max;
        rightBP /= max;
    }
```

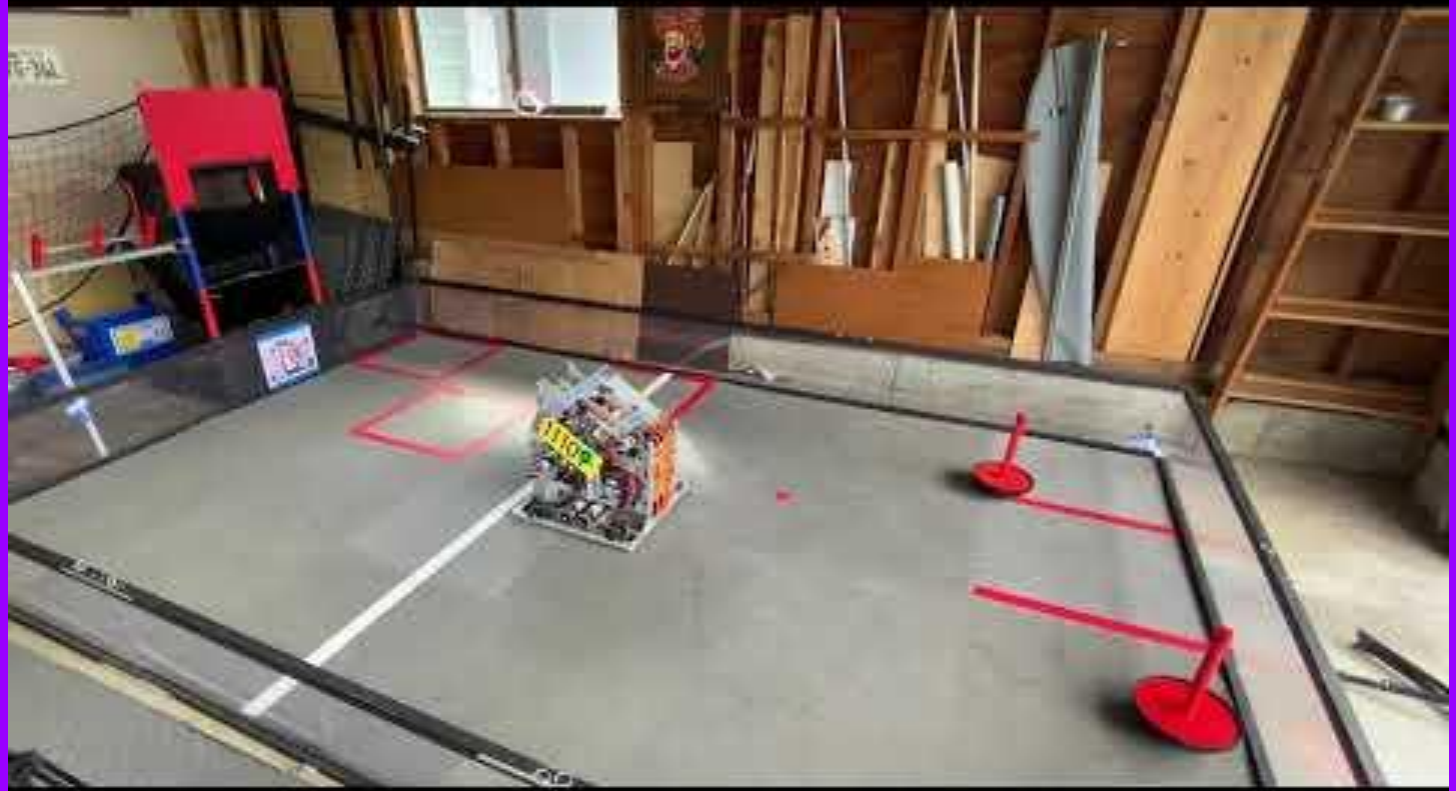
```
public boolean isUpToSpeed() {
    if (stablizationMode) {
        if (shooterMotor.getVelocity() > targetShootSpeed - 30 && shooterMotor.getVelocity() < targetShootSpeed + 30 ) {
            if (rangeTime.milliseconds() > 500){
                stablizationMode = false;
                return true;
            } else{
                return false;
            }
        } else {
            rangeTime.reset();
            return false;
        }
    } else {
        if (shooterMotor.getVelocity() < targetShootSpeed - 200) {
            stablizationMode = true;
        }
        return true;
    }

}
```

```
eleBeltsRampBelts = (DcMotorEx) hardwareMap.get(DcMotor.class, deviceName: "eleBeltsRampBelts");
eleWheelsLowerBelts = (DcMotorEx) hardwareMap.get(DcMotor.class, deviceName: "eleWheelsLowerBelts");
```

```
public void encoderDrive(double speed, double leftInches, double rightInches, double timeOutS) {
    double leftTargetTicks = leftInches * COUNTS_PER_INCH;
    double rightTargetTicks = rightInches * COUNTS_PER_INCH;
    double startOffset = drive.getLeftTicks();
```

Our drive train code

Our autonomous



Feel free to ask for links:

Portfolio:
https://qlip-vault-1.s3-accelerate.amazonaws.com/web/customer_files/3740688736268/portfoliov2.2.pdf?Expires=2075494478&AWSAccessKeyId=AKIAJROPQDFTIHBTLJJQ&Signature=mb55Yzdiba0xzGZ0dyccu4FiStc%3D

Autonomous video:
https://www.youtube.com/watch?v=vn0NuHrxk6c&t=1s