Robot Design

The 42 Watts

Rubrics : Validation slide

Rubric	Accomplished	How covered?
Identify	Clear mission strategy	Drive least distance.
	Consistent evidence of building and coding skills in all team members	● Clearly state that each of us built the robot. ● Each of us built an arm ● Each of us coded their run / arm.
Design	Clear evidence of an effective plan	See plan
	Clear explanation of robot and codes innovative features	ROBOT (one slide per each below) 1. REWA 2. Use of Rubber bands in the attachments. (Principle of fail gracefully) 3. Multiple missions simultaneously. (Two arms on one motor) + mechanical aligners CODE (one slide per each) 4. Obstacle Avoidance algorithm / Coordinate Plane Algorithm to get to the missions

Principles

- [Mission strategy] Minimize total distance Travelled
- [Design] Be Adaptable, right arms for the missions
- [Design] Build reusable components
- [Design] Multiple missions simultaneously
- [Design] Fail gracefully

Mission strategy : Dependency Graph

Points
Attachments
Zone
Difficulty

+

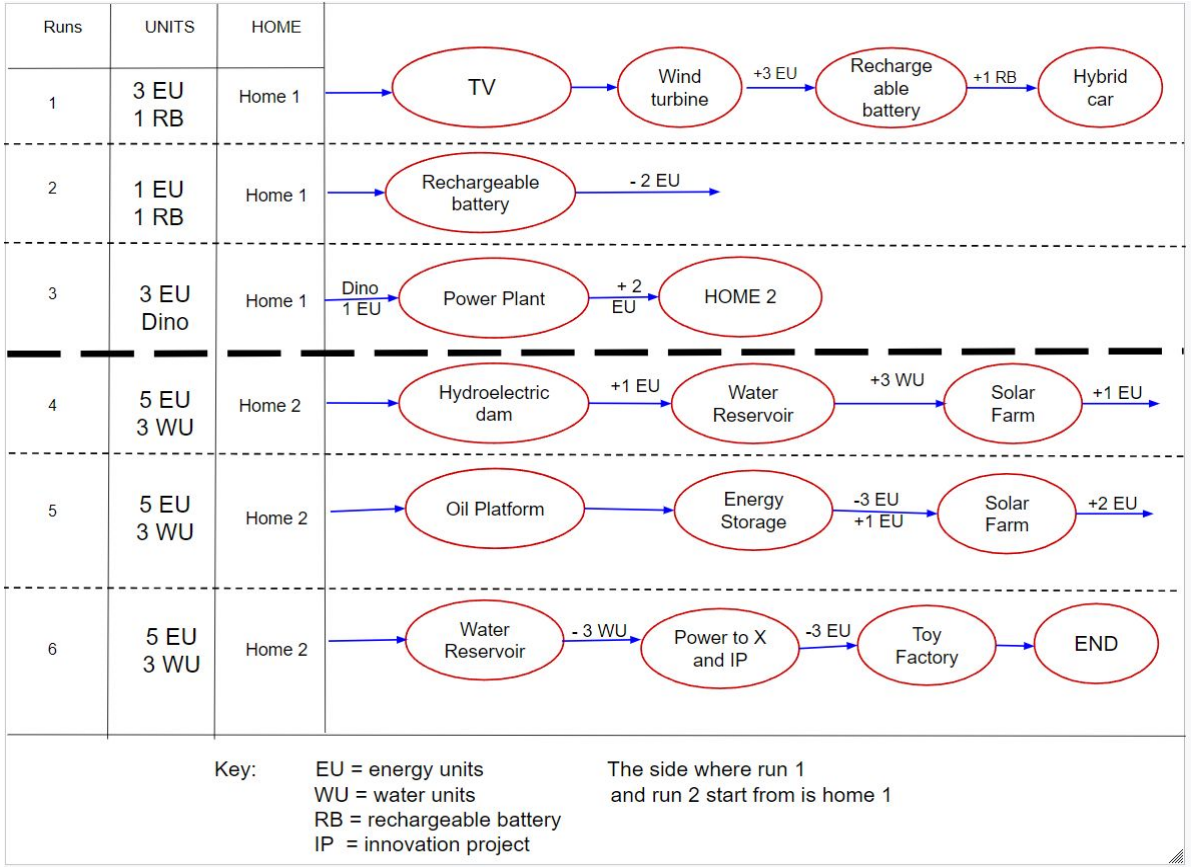


Figure: Dependency Graph

Mission strategy : Travel least distance

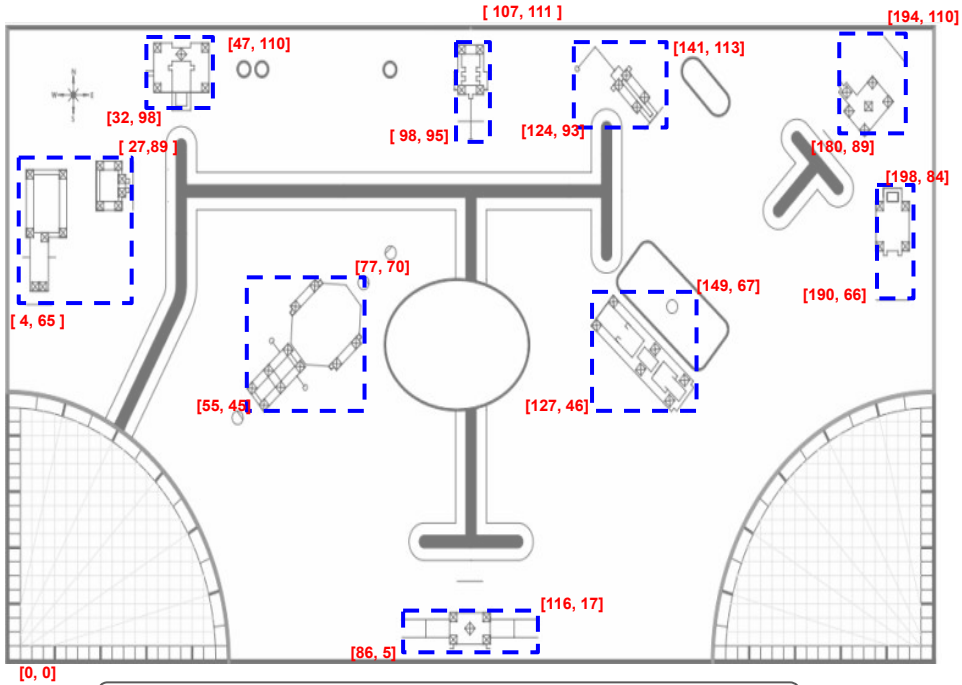


Figure: Coordinate plane

A	B	C	D	E
Mission	X(CM)	Y(CM)	Distance (CM)	Run number
Home 2	195	10	0.00	RUN 1
Watch Television	190	66	56.22	
Wind Turbine	180	90	26.00	
Hybrid Car	135	90	45.00	
Rechargable Battery	150	60	33.54	
Home 2	195	10	67.27	RUN 1.5
Rechargable Battery	150	60	67.27	
Home 2	195	10	67.27	RUN 2
Power Plant	100	20	95.52	
Home 1	20	10	80.62	RUN 3
Hydroelectric Dam	50	40	42.43	
Water Reservoir	80	70	42.43	
Smart Grid	98	95	30.81	
Solar Farm	80	100	18.68	
Home 1	20	10	108.17	RUN 4
Oil Platform drop the canisters	10	65	55.90	
Energy Storage	32	98	39.66	
Solar Farm	80	100	48.04	
Home 1	20	10	108.17	RUN 6
Innovation Project Model & Power to X	100	60	94.34	
Toy Factory	145	40	49.24	
Total Distance Travelled			**1176.58**	

Figure: Mission order, total distance

Mission strategy

Principle: Minimize total distance Travelled

15 missions

6 Runs

355 Average score

Total distance travelled 1176cm (Reduced from 1677cm)

ALL Team Members learned building and coding skills

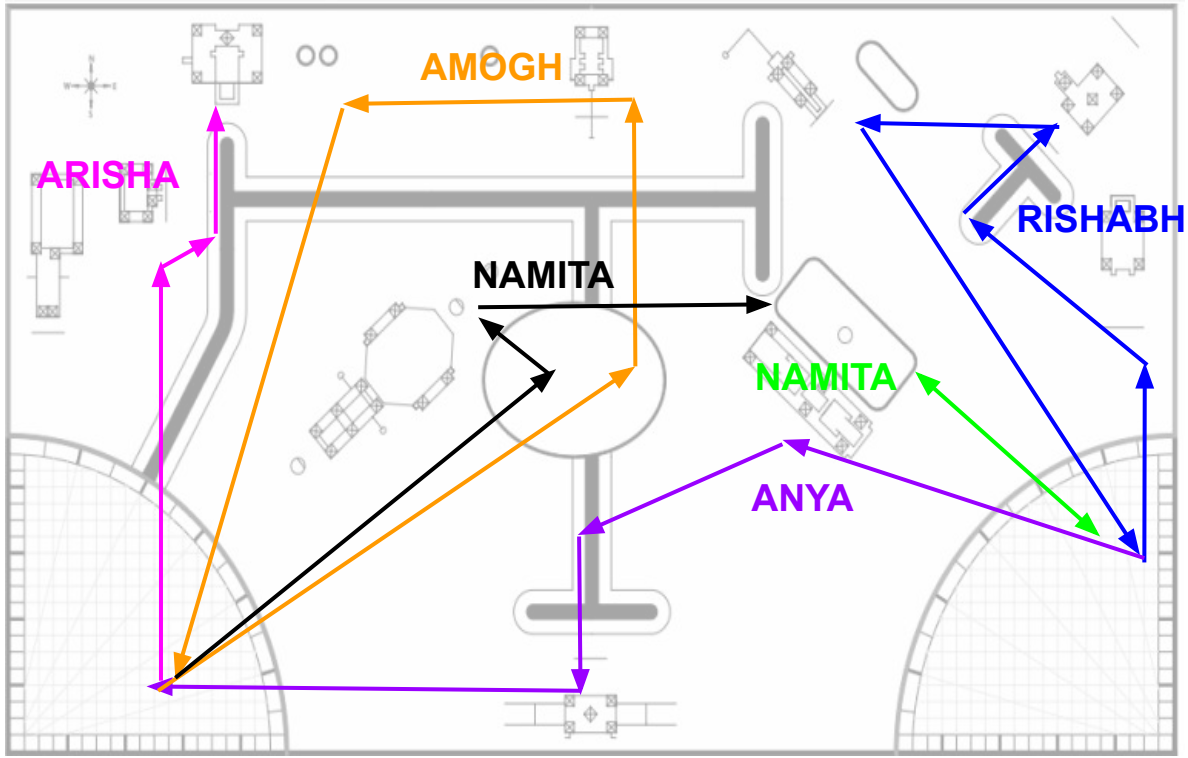


Figure : Runs

Coding Skills:

- Rishabh - TurntoAngle, Drive, Line Catching
- Amogh - Line Squaring
- Anya - GyroStraight
- Arisha/Nami - Programming in Python and GitHub Integration

Design: Effective plan

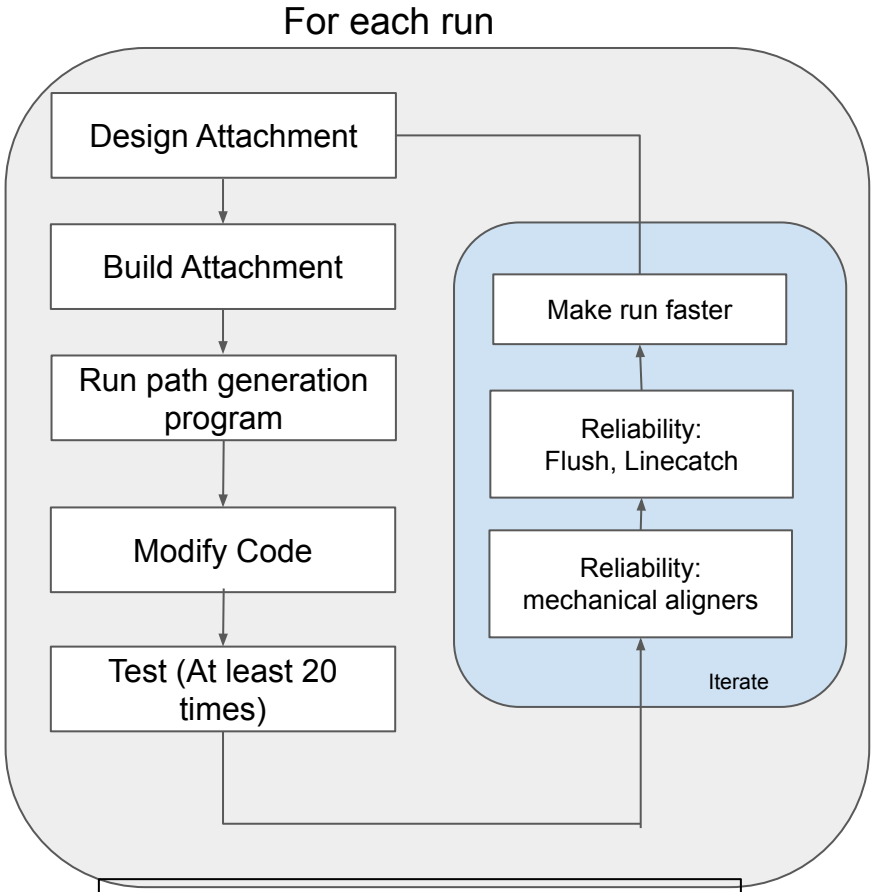


Figure: Plan

Tools used

- ❏ CAD: Studio 2.0
- ❏ Python
- ❏ Visual Studio
- ❏ GitHub

Innovation: ReWA: Reusable Worm gear Apparatus

Reusable hardware Component

Convert motion in one plane to another

Used in 7 attachments

Many mounts

Compact

No gear slip

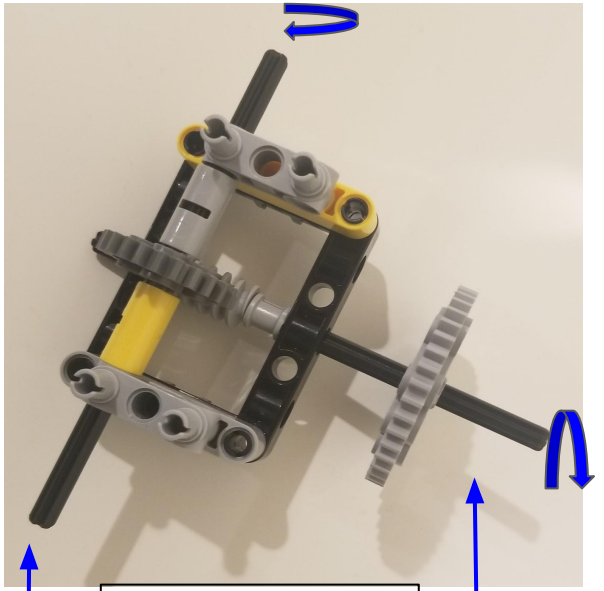


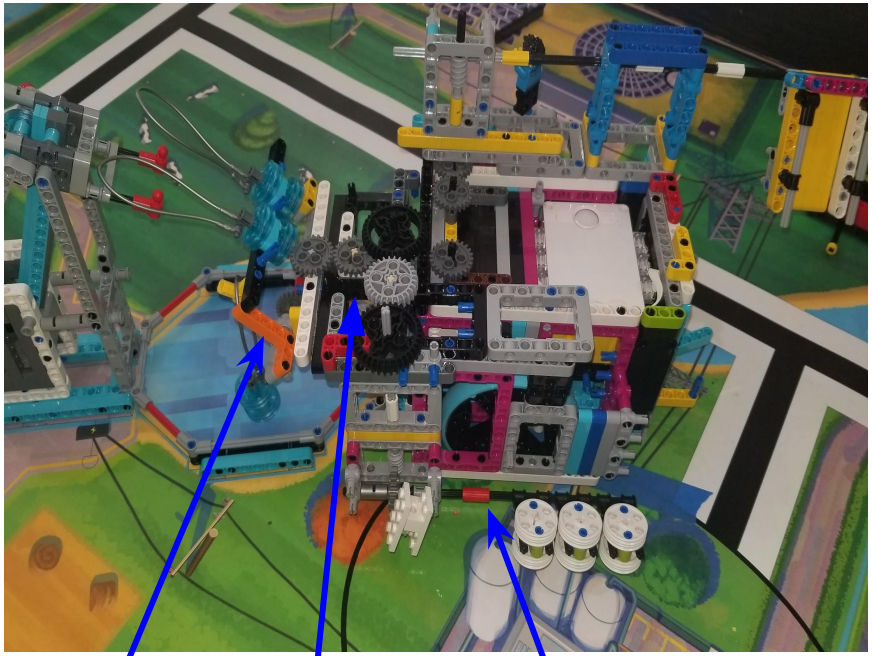
Figure: ReWA

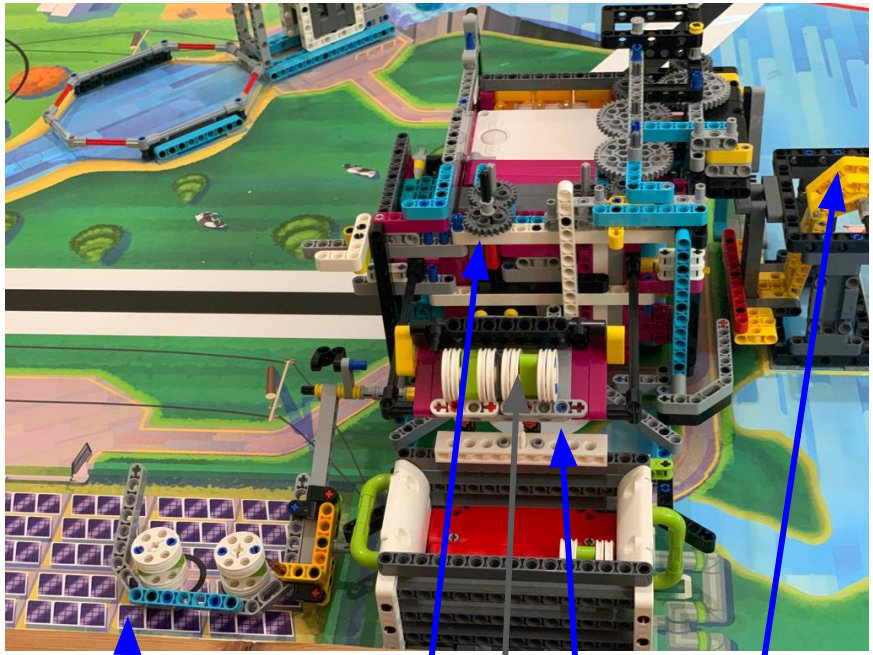
This axle turns the attachment. Can only be turned by the driving axle

Motor Axle

Principle: Reusable components

Innovation: Multiple mission simultaneously





Arm to drop off all water units

Both arms on same motor

Arm for Energy units and innovation project

Solar arm

Solar arm and energy tray on same motor

Energy Storage arm

Energy tray arm

Oil platform arm

Figure: Multiple missions simultaneously

Innovation (Code): Robot.goto(X, Y)

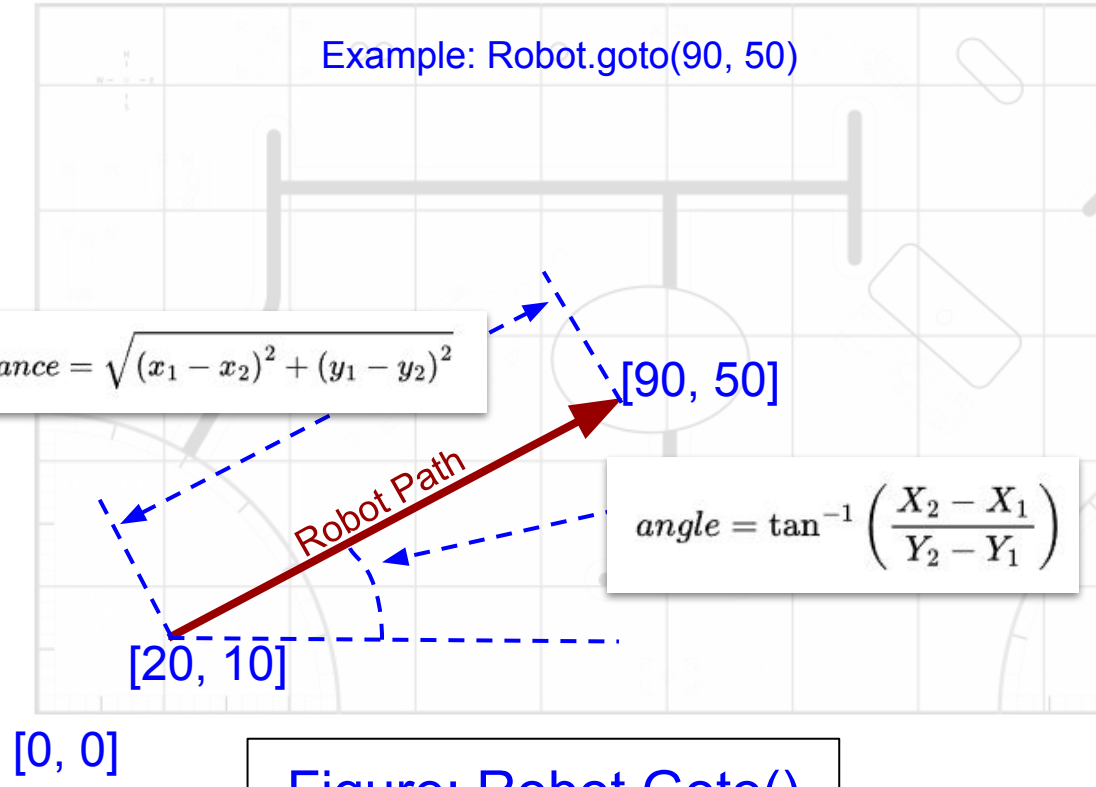


Figure: Robot.Goto()

```python
def goto(self,x2,y2,endAngle,speed):
    global angle, slope, quadrant2, distance
    angle = 0
    slope = 0
    quadrant2 = 0
    distance = 0
    x1 = self.currentLocationX
    y1 = self.currentLocationY
    a1 = self.currentRobotAngle
    def _calculateSlope(x1,y1,x2,y2):…
    def _findQuadrant(x1,y1,x2,y2):…
    def _findEndQuadrant(x1,y1,x2,y2,maxX,maxY,minX,minY):…

    def _calculateAngle(slope):…

    def _fixAngle(endQuadrant, rAngle):…
    def _findDistance(x1,y1,x2,y2):…
    def _move(speed):…

    _findQuadrant(x1,y1,x2,y2)
    _calculateSlope(x1,y1,x2,y2)
    _calculateAngle(slope)
    _fixAngle(quadrant2, angle)
    _findDistance(x1,y1,x2,y2)

    self.currentLocationX = x2
    self.currentLocationY = y2
    _turnToAngle(targetAngle = endAngle, speed = speed)
```

Figure: Code

Innovation (Code): Path finding program

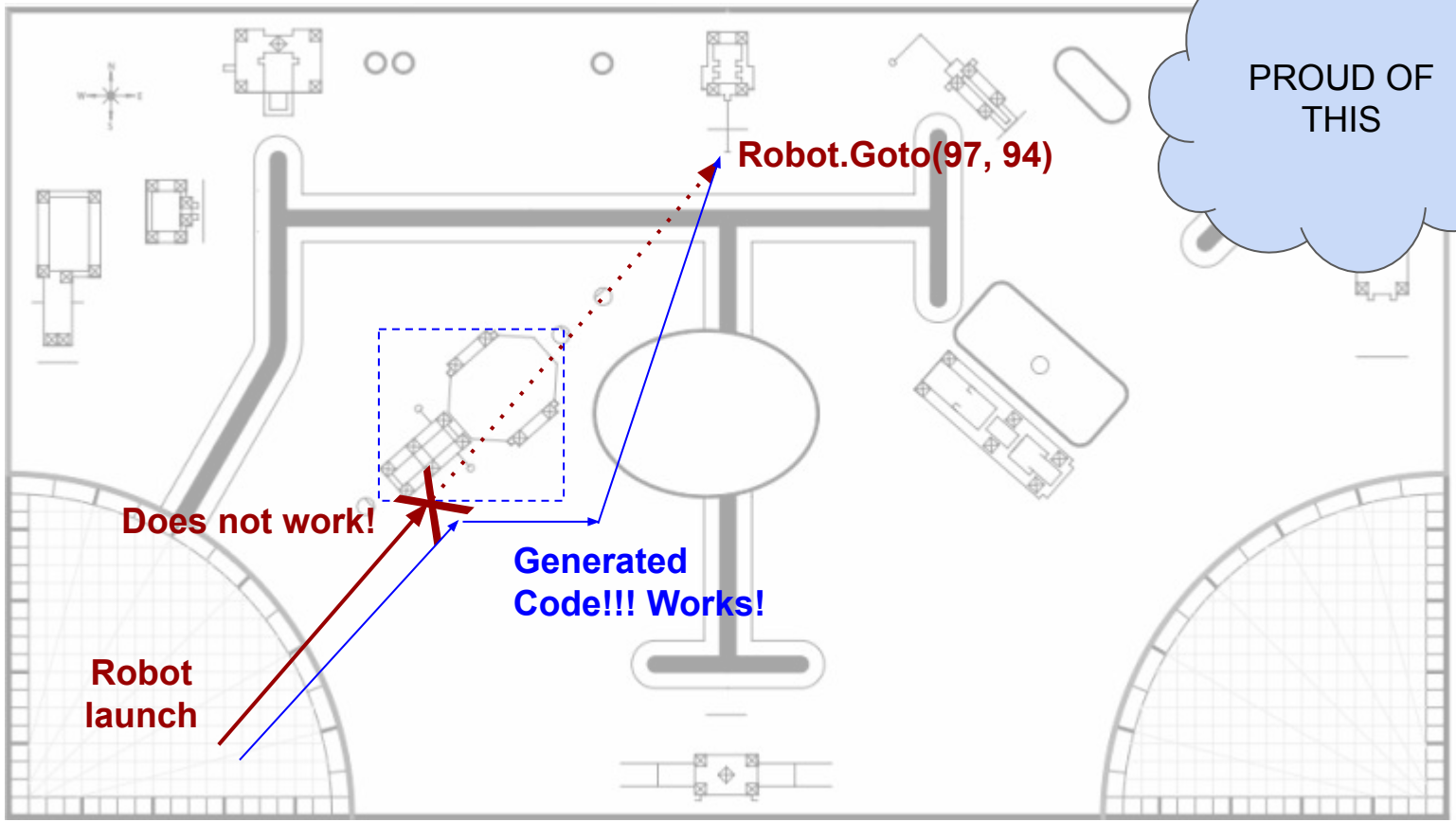


Figure : Path finding program

Innovation (Code): Path finding program

Program that writes code!!!

Obstacle Handling

line intersection, distance formula and recursion

```
1   # Driving from (30,10) HydroDam
2   _turnToAngle(targetAngle=-51, speed=50)
3   drive(speed=50, distanceInCM=45, target_angle=-51)
4
5   # Code to get past mission: HydroDam
6   _turnToAngle(targetAngle=0, speed=50)
7   drive(speed=50, distanceInCM=30, target_angle=0)
8
9   # Drive from (87,45) to (97,94)
10  _turnToAngle(targetAngle=-79, speed=50)
```

Figure: Sample output Code produced by pathfinder

Create: Robot and Sensor Functionality

- Box robot
- 6 attachments
- 2 color sensors
- Gyro sensor
- 2 vertical medium motors

- Cabling beams
- Charging ports
- Cages around wheels

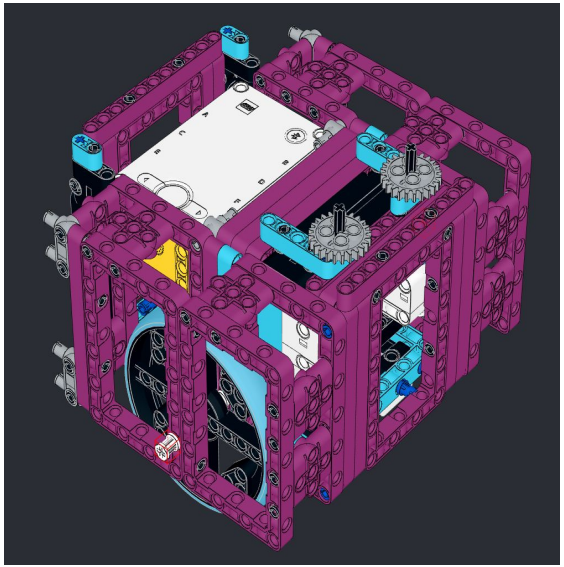


Figure: CAD Design

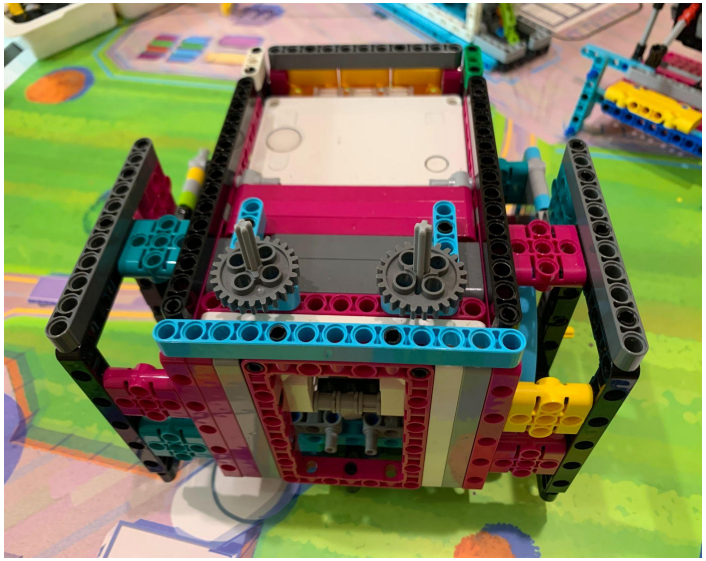
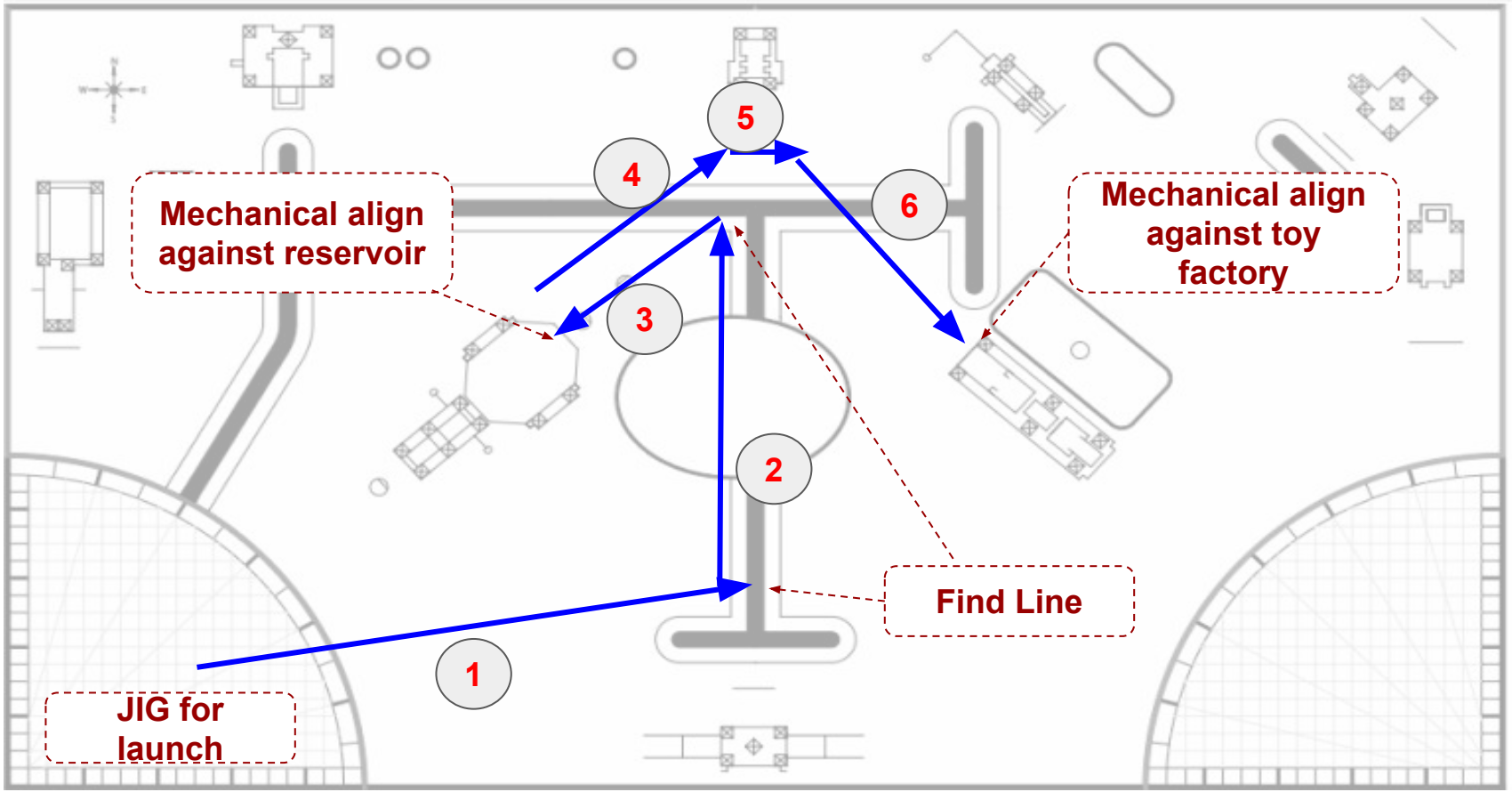


Figure: Final version

Create: How does the code make robot act?



Testing Robot and Code

un2 #	Toy Factory	EU front	EU left	EU right	Power plant raise	Power plant lower	Battery Level	Time Taken (seconds)	Notes	Improvements
1	Yes	Yes	Yes	Yes	Yes	Yes	7738	18.218		
2	Yes	Yes	No	No	No	Yes	7750	18.175	didn't raise completely EUs didn't get releas	
3	Yes	Yes	No	No	No	Yes	7712	18.056	didn't raise completely EUs didn't get releas	
4	Yes	Yes	Yes	Yes	Yes	Yes	7673	17.823	left EU got captured an swung off when the r turned. Still ended up launch area thoug	
5	Yes	Yes	Yes	Yes	Yes	Yes	7661	18.382		
6	No. didn't release dinosaur	Yes	No	No	No	Yes	7634	18.128	the energy units got stu didn't release the din	
7	Yes	Yes	No	No	No	Yes	7624	18.26	didn't raise completely EUs didn't get releas	
8	Yes	Yes	No	No	No	Yes	7605	18.126	didn't raise completely EUs didn't get releas	

Figure: Run 2 tests

Score	Max	Accuracy	Run 2	Run 3	Run 4	Run 5	Run6	Run7	Run8	Run9
Precision tokens	50	100	50	50	50	50	50	50	50	
Small inspection	20	100	20	20	20	20	20	20	20	
TV	20	95.83333333	20	20	10	20	20	20	20	
Windmill	30	94.44444444	30	30	30	20	20	30	30	
Hybrid car	20	75	20	20	0	10	0	10	20	
Rechargeable battery	10	100	10	10	10	10	10	10	10	
power plant	30	80.55555556	30	20	20	30	10	30	30	
solar energy unit pickup	5	100	5	5	5	5	5	5	5	
hydroelectric plant	20	85.41666667	20	20	20	20	0	20	5	
smart grid	30	61.11111111	20	20	20	0	20	20	20	
oil factory	15	97.22222222	15	10	15	15	15	15	15	
energy storage	30	97.22222222	30	30	30	30	20	30	30	
Energy storage tray	5	83.33333333	5	5	5	5	0	5	5	
Solar energy unit pickup - other 2 units	15	94.44444444	15	15	15	15	5	15	15	
powerToX	15	94.44444444	15	15	15	15	5	15	15	
Innovation model	10	100	10	10	10	10	10	10	10	
water reservoir	25	88.33333333	25	25	25	25	15	25	25	
toy factory	20	85.41666667	20	20	20	15	5	15	15	
Dinosaur	30	100	30	30	30	30	30	30	30	
	400	90.625	390	375	350	345	260	375	370	
Timing	145		138	132	140	145	150	143	139	

Figure: Full run accuracy test

Testing Robot and Code

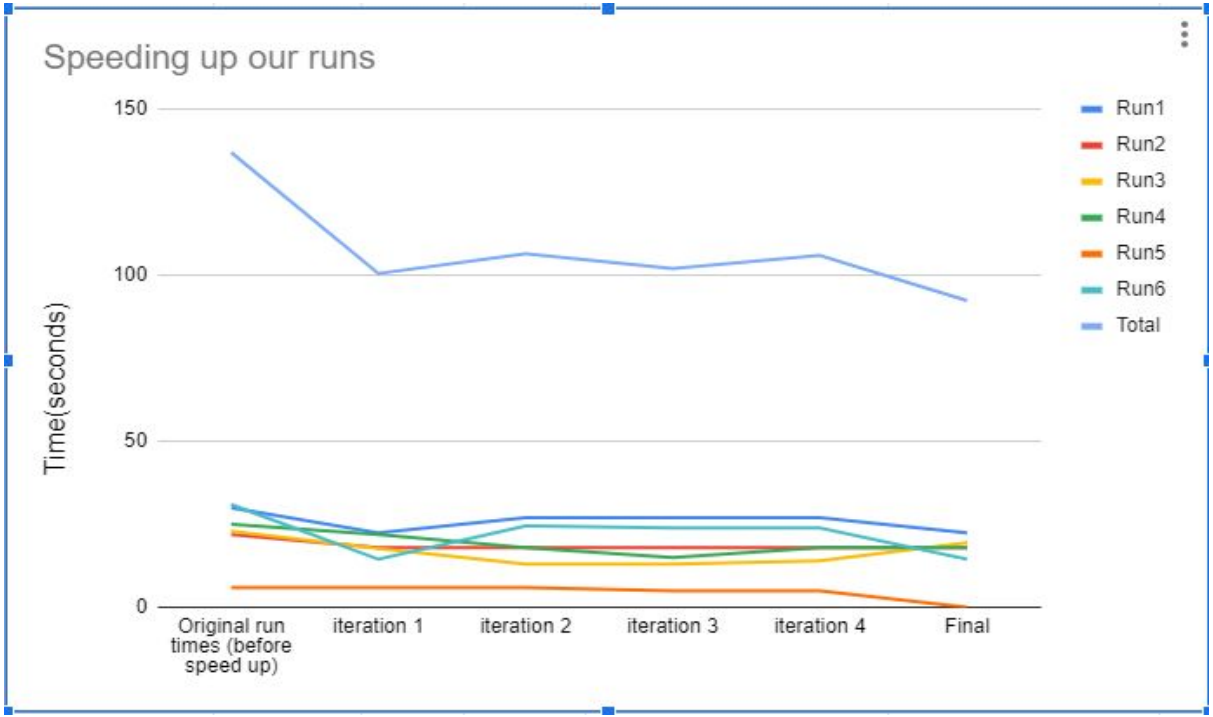
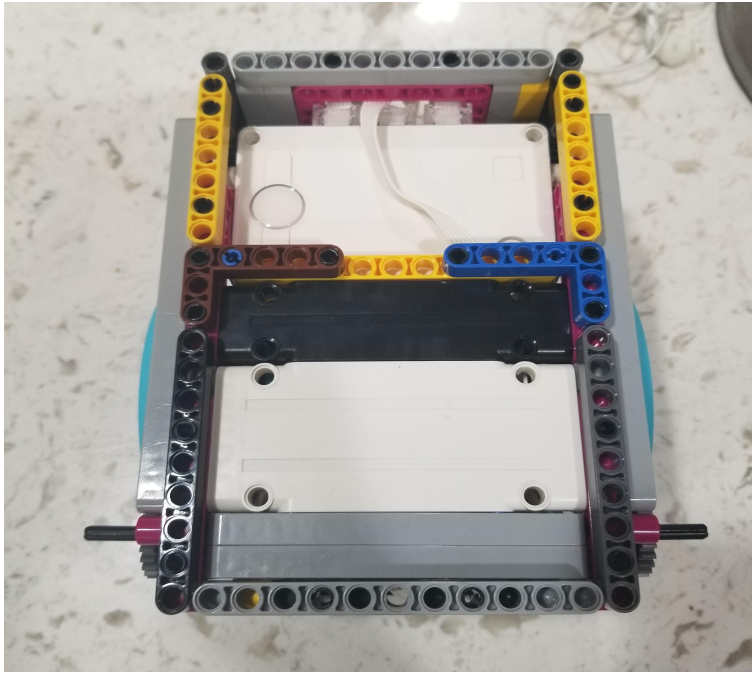


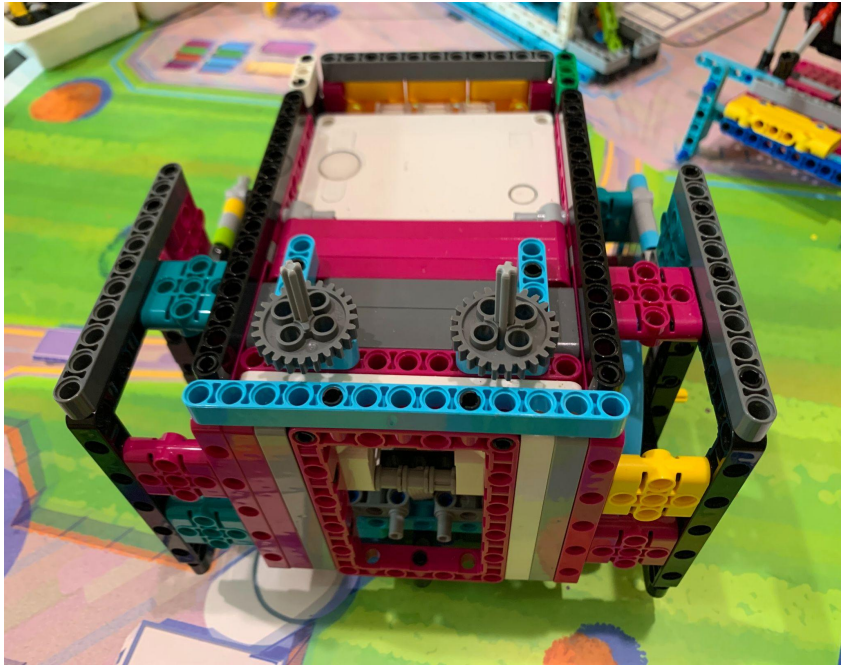
Figure: Speedup

Tested on
2 tables, 3 different robots

Iterate: Improvements to Marvin(Our Robot)



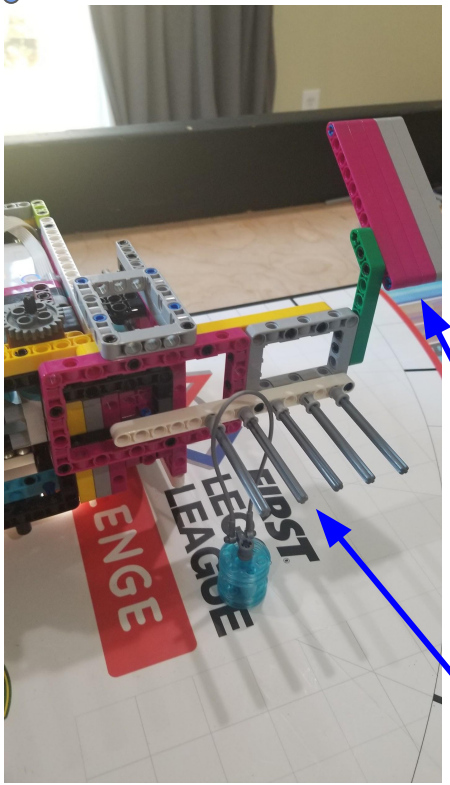
Marvin #10



Marvin #14 (Final)

Iterate: Rubber Bands

Principle: Fail gracefully



Rigid arm would snag

Lattice arm

Figure: Iteration 1

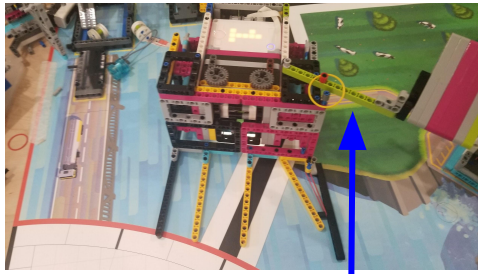
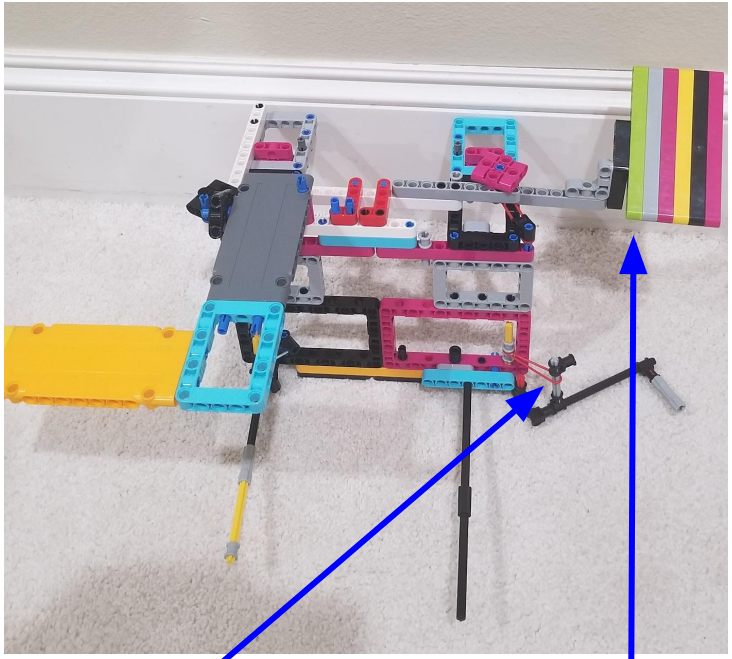


Figure: Iteration 2

Flexible arm with rubber band

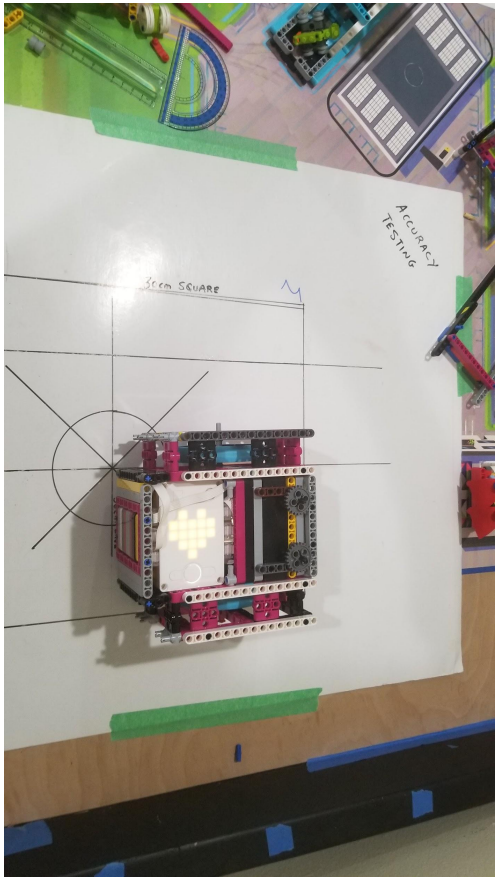


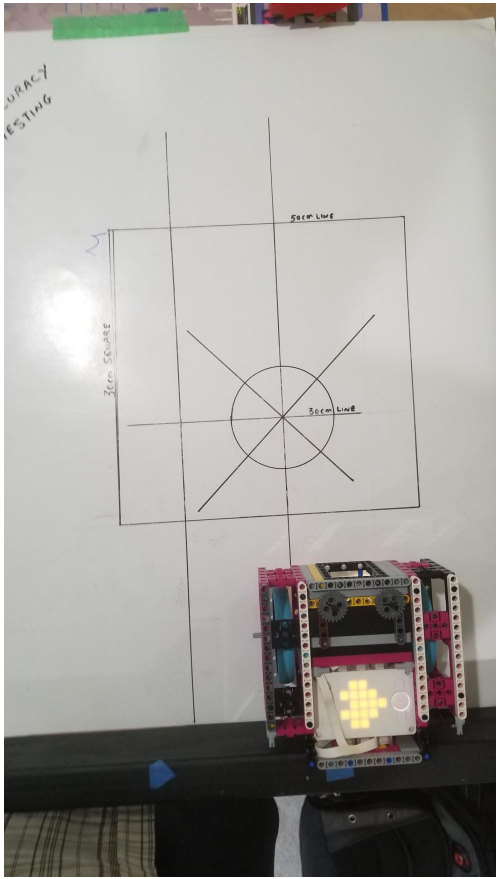
S shaped flexible arm for water unit

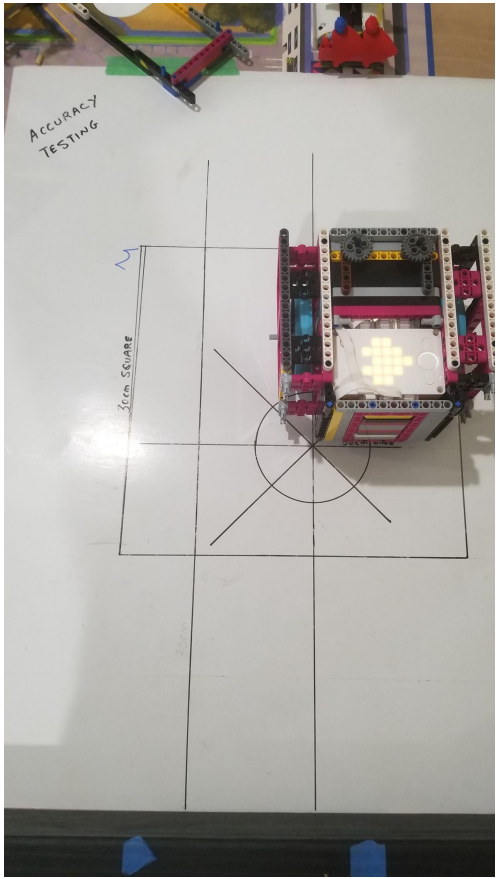
Flexible arm for Hydroelectric dam

Figure: Final Arm

Iterate: Code Improvement







Iterate: Code Improvement

Experiment : 720d Turn

Tested 5 Algorithms

Fast and Accurate turn

80% improvement in Turn accuracy

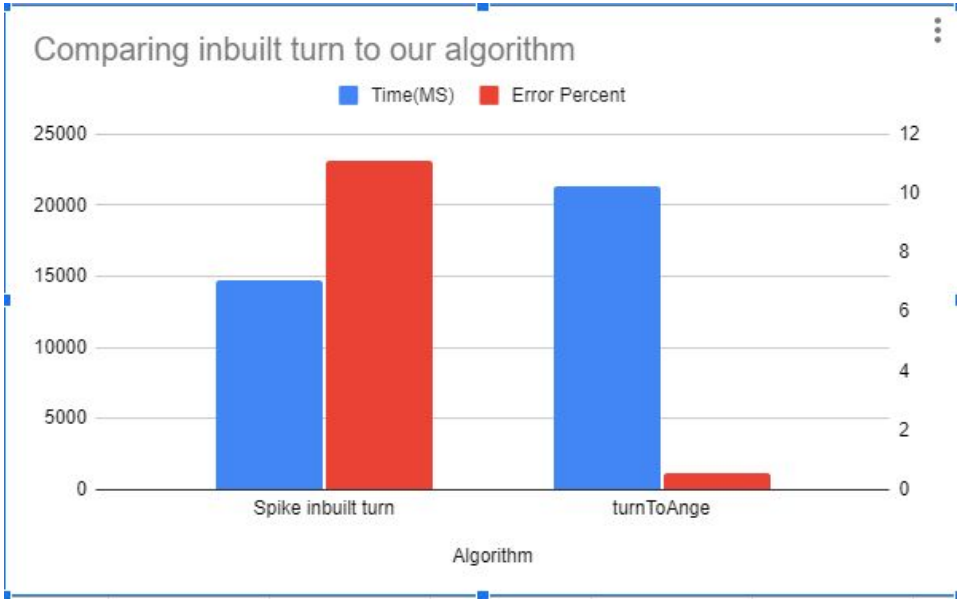


Figure: Turn Data

Thank you! Any Questions?

Year in Review!

153 github commits

~2300 lines of python code

~40 broken pegs
1 broken h-connector
1 broken axle

Our SuperPowered Learnings:

- Principles of design
- Python
- Coordinate plane driving
- Path finding algorithm
- Github for sharing our code and version control
- Fail gracefully
- ReVVA: Hardware reuse

Additional content

Innovation : Garage

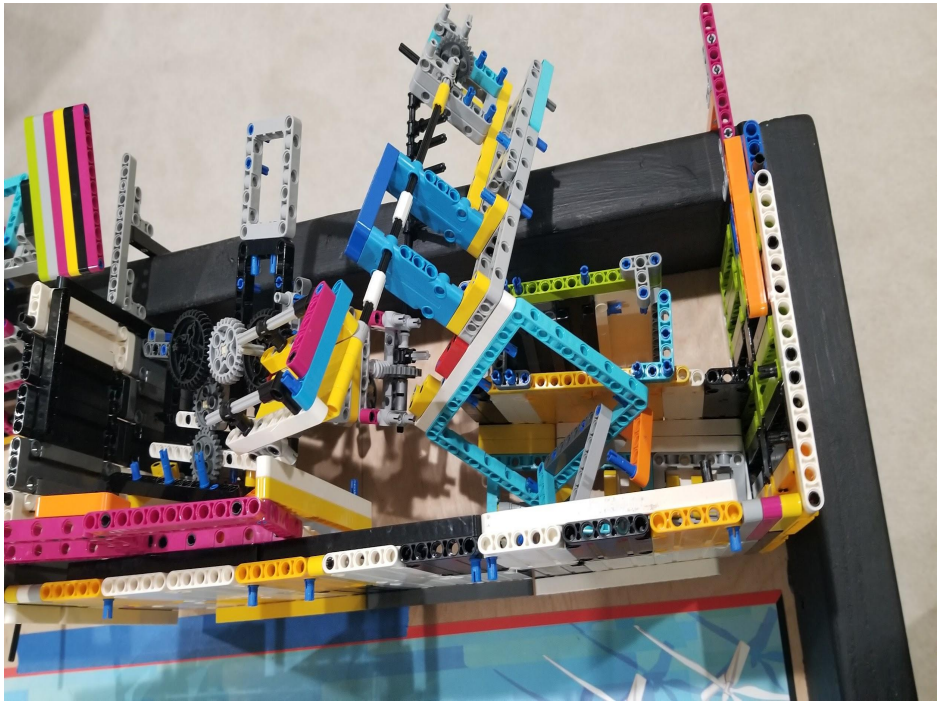


Figure : Garage with Arms in Left Home

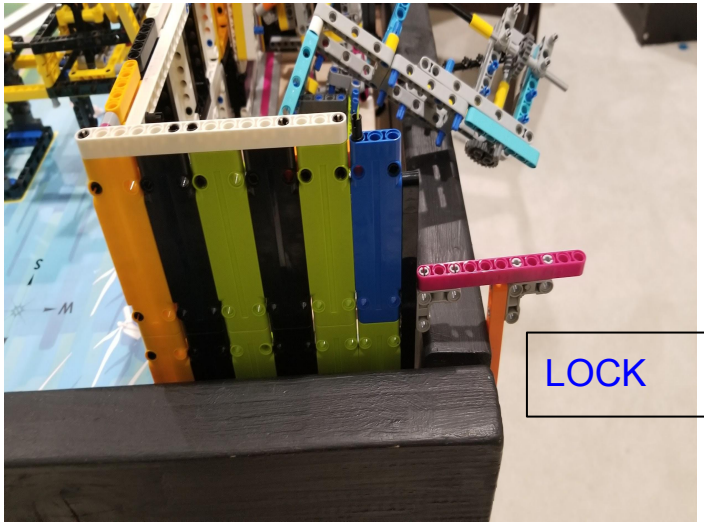


Figure : Adjustable lock

Create: Attachment Functionality

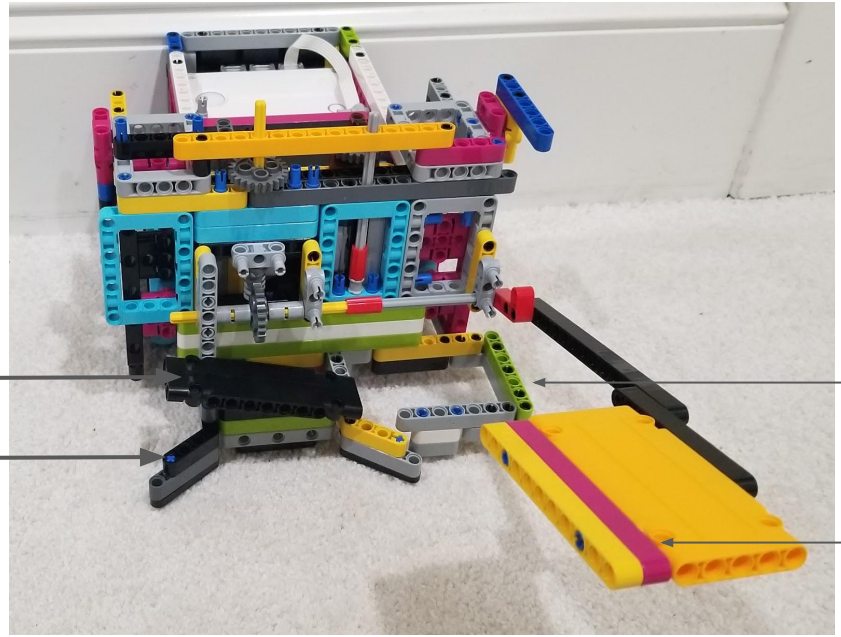
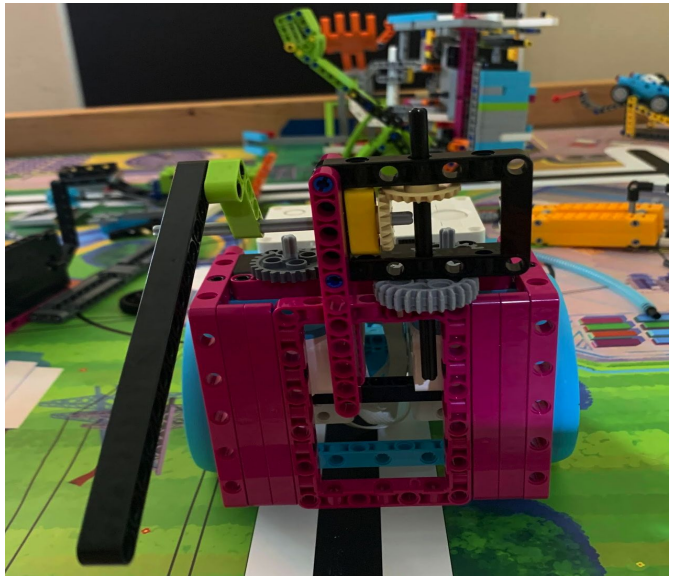
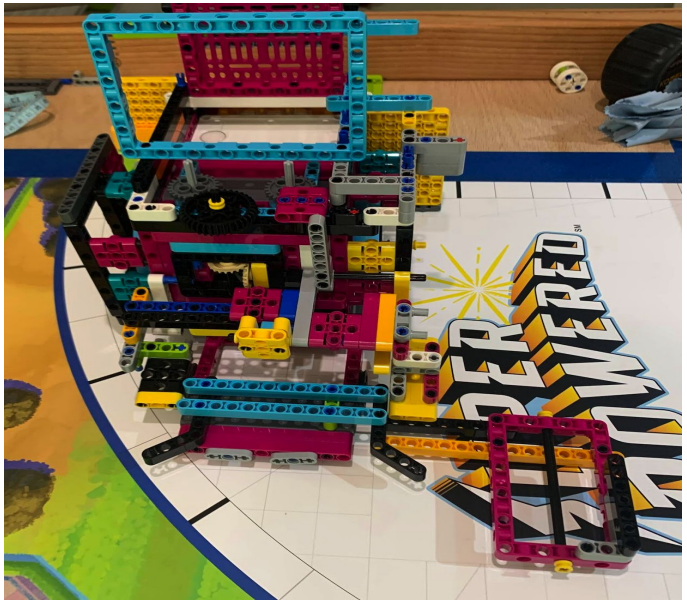


Figure: Spinny

Iterate: Attachment improvement



Problem: Energy unit not being captured



Final attachment

Innovation Code: Drive straight

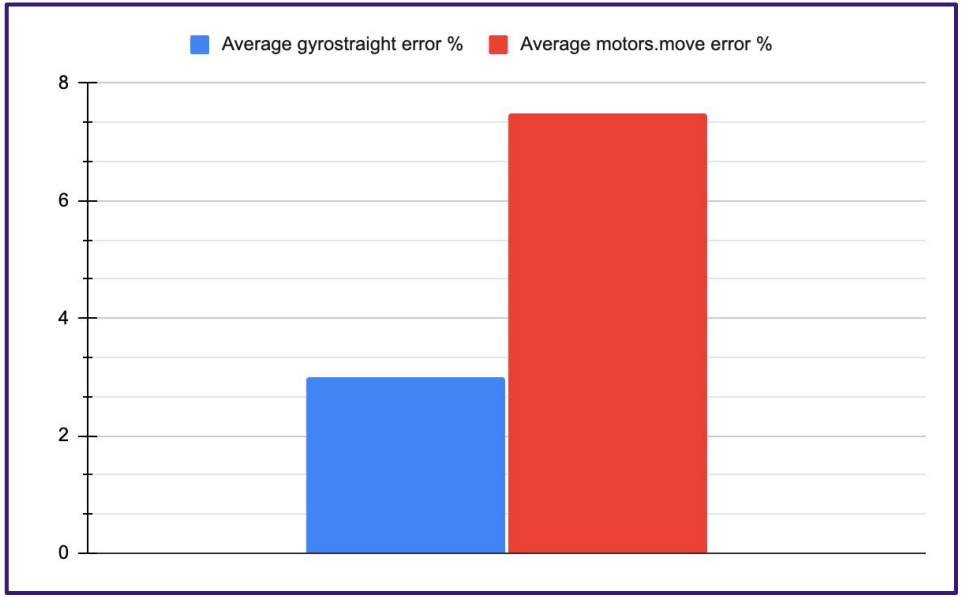


Figure 1: Gyrostraight vs. motors.move percentage error

Run #	Speed	Expected Distance	Actual Distance (in)	Abs. Error	Abs Error %
1	60	10	10.1	0.1	1
2	60	10	9.3	0.7	7
3	60	10	10.02	0.02	0.2
4	60	10	9.62	0.38	3.8
5	60	29	26	3	10.34
6	60	29	25	4	13.79
7	60	29	26.75	2.25	7.76
8	60	45	41.2	3.8	8.44
9	60	45	41.9	3.1	6.89
10	80	20	18	2	10
11	60	50	49	1	2
12	60	50	53.9	3.9	7.8
13	60	50	49	1	2
14	60	50	48.75	1.25	2.5
15	60	50	48	2	4
					5.834666667

```
if( (currentAngle <= 0 and targetAngle <=0) or
        (currentAngle>0 and targetAngle > 0) or
        (abs(currentAngle) < 90 and abs(targetAngle)<90)):
    correction = targetAngle - currentAngle
elif (currentAngle >= 90):
    correction = (360 - abs(currentAngle) - abs(targetAngle))
else:
    correction = -1*(360 - abs(currentAngle) - abs(targetAngle))
```

Slide	What?	Time	Notes
4	Mission strategy	22	
5	Mission strategy	15	
6	Mission strategy	4	
7	All team members worked on code	8	
8	Effective plan	20	
9	ReWA	16	
10	Multiple missions	28	
11	GOTO	20	
12	Obstacle avoidance	27	
13	Obstacle Avoidance	4	