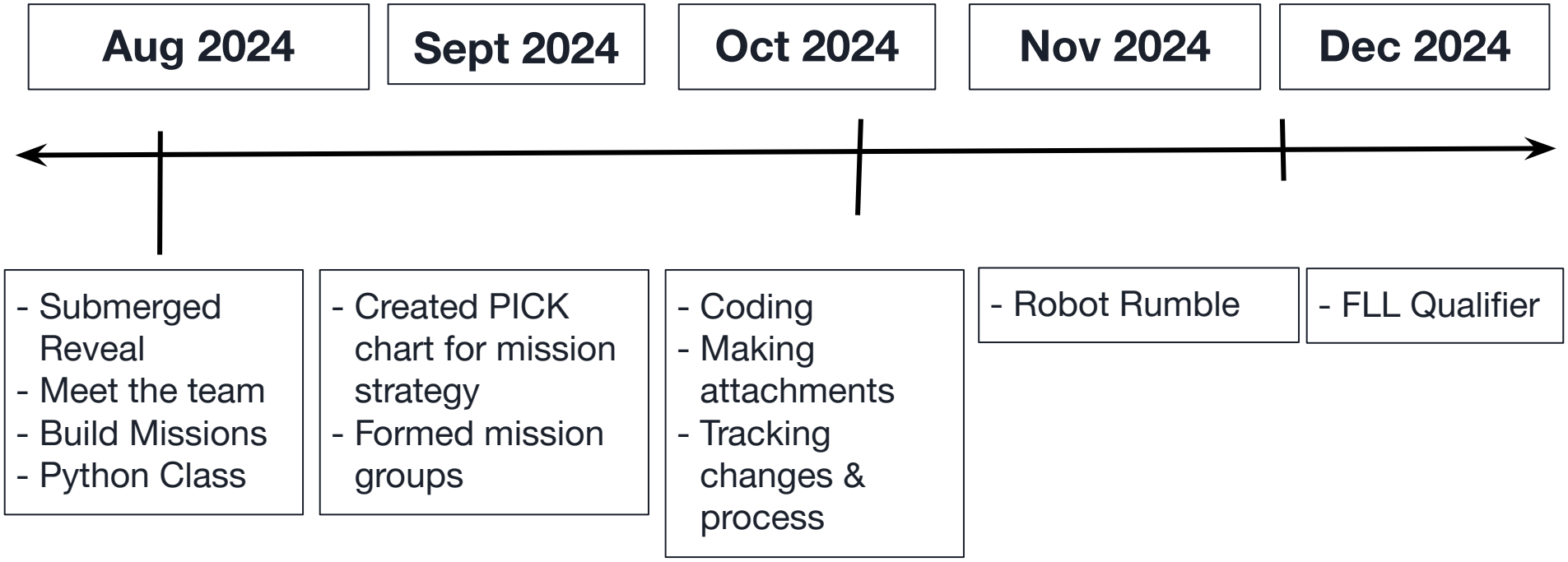

CAPTIBYTES
#3249
LIVE
FIRST
LEGO
LEAGUE

Robot Timeline



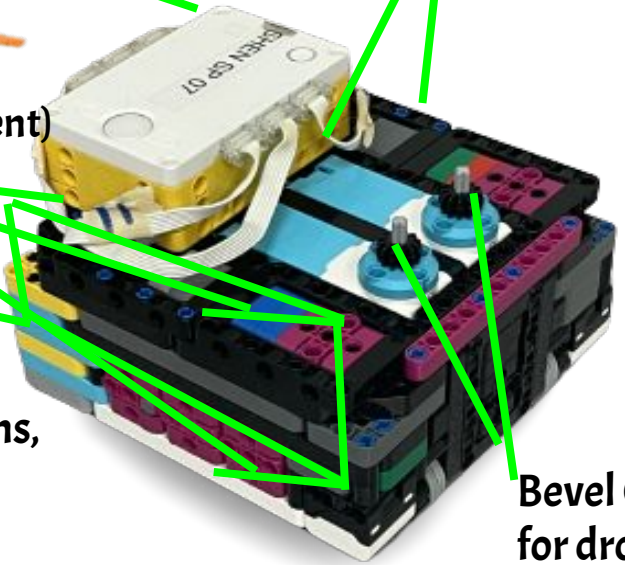
Meet the bots

OUR ROBOT: Meet Mary

Spike Prime Hub

Attachment motors (medium size)

Flat Frame (for easy alignment)



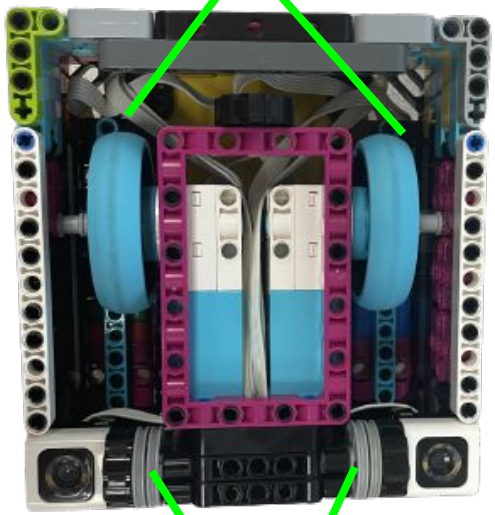
Used Python to code missions,

python™

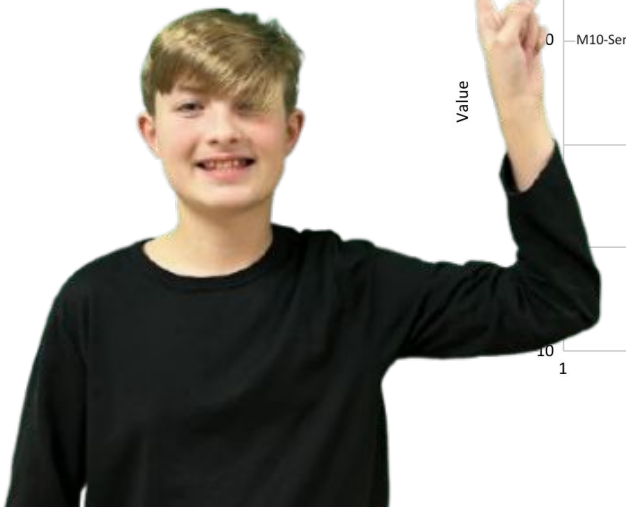
Bevel Gears for drop in attachments

Drive Train:

Wheels (large size)

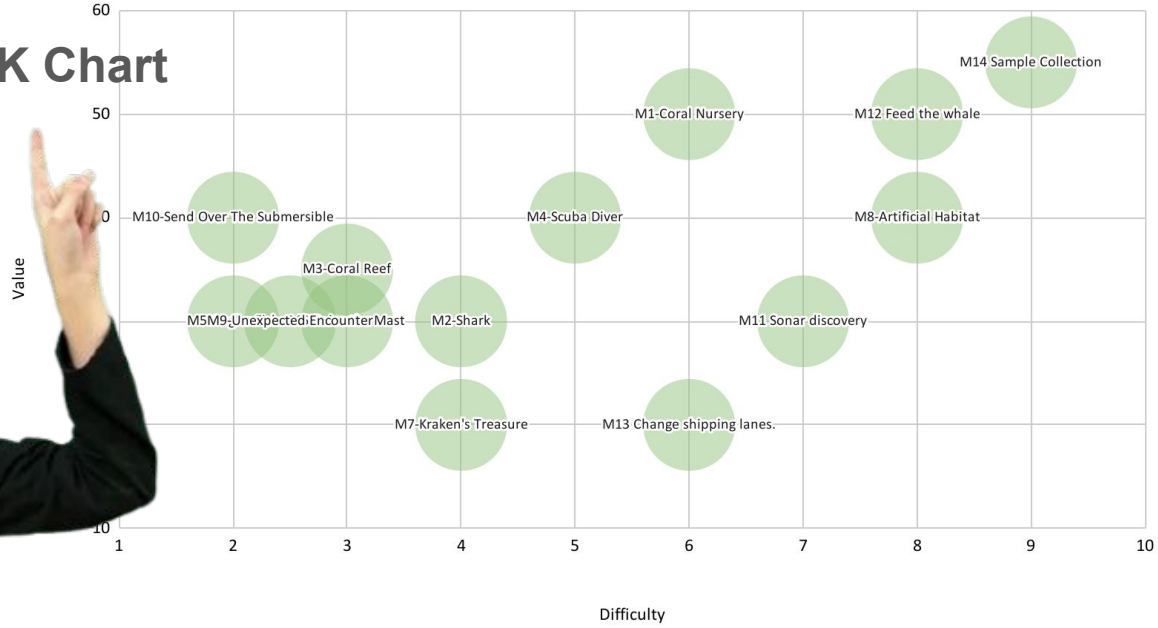


Black smooth wheel rims (no tread) for easy glide movement



Implement	Challenge
Possible	Kill

PICK CHART: Value vs. Difficulty

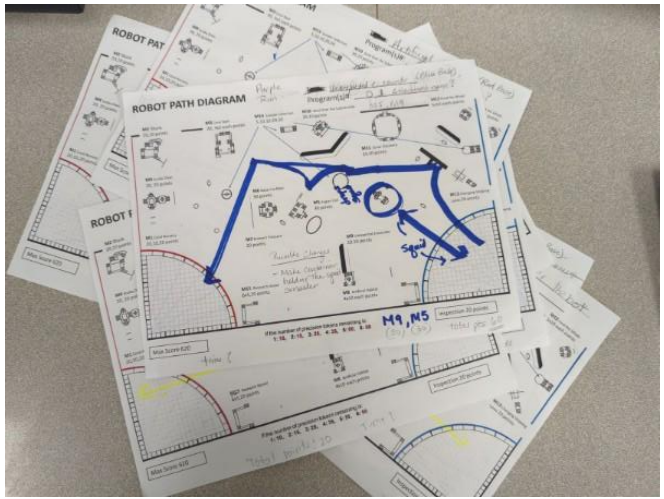


IDENTIFY: A PICK Chart was used to initially identify missions based on point value, and difficulty.

DESIGN : We used this data to group our missions for team members to work on.

Pseudo Code and Robot Path Diagram

We write pseudocode to help figure out how to code but doing it in a way of writing it out to say what it does.



Blue Base PseudoCode

1. Robot leaves base then goes straight and pushes into the "Unexpected Encounter" and picks up the Octopus.
2. The robot then quickly backs up into the blue base with the octopus and then the code is switched
3. Attach the new attachment for picking up all the samples and add the octopus to the green part of the attachment, then the robot will head out to drop the octopus off in its area
4. Then it will try and do "Angler Fish"
5. After all the krill and coral are collected it will head to the red base and then will be switched to the code for red base

Red Base PseudoCode

1. Robot leaves base then stops and turns, goes forward a bit then uses attachment and lifts up the yellow bars of "Raise the Mast" with a lot of power inorder to raise it.
2. We then back up into the "Coral Nursery" and make sure it pushes the thing back with an angled piece on the back of the robot.
3. We have the robot go forward a bit then turn and go straight then stop in front of the "Coral Reef" and then have a hammer-like attachment slam down into the yellow bar in the back to flip the coral up.
4. The robot then backs up and turns and goes forward with the same hammer-like attachment and slams down into the "Shark" or just have a small slope on the front/back to move into or push the yellow tab down to lift up the shark.
5. Then we have the robot back up and turn then go back to home base (red base)

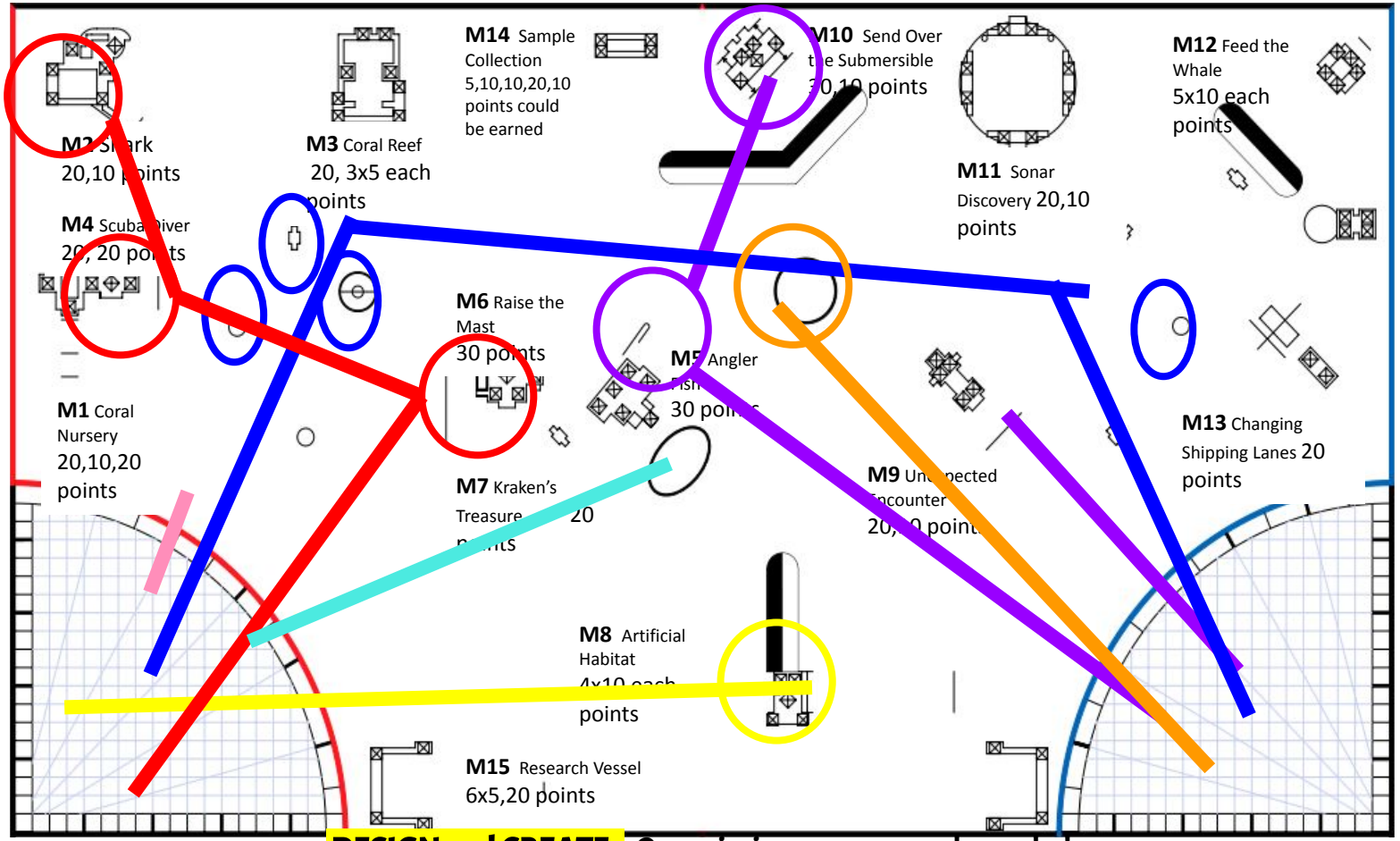
The Attachments

Pythagoras

Weightlifter

Hammer Time

The Kriller



DESIGN and CREATE Our missions runs are color coded.

Mary Hard at Work!

Blue Run - 5
Red Run - 90
Orange - 10
Purple - 80
Yellow - 30
Cyan-10
Pink-15
Penalties - 50
Inspection - 20

Rumble - 155
Qualifier - 310

```
async def main():
    await motor.run_for_degrees(port.F,degrees,360) #,acceleration=180,deceleration=180) #
#########################################################################################
elif (dist > 0):                                        #
    motor.stop(port.C,stop=1)                           #
    motor.stop(port.F,stop=1)                           #
    await motor.run_for_degrees(port.C,degrees,360) #,acceleration=180,deceleration=180) #
    print(degrees)                                      #                                 #
if (dist < 0):                                          #
degrees = int(dist*(360.0/(math.pi*d)))                 #
print(dist)                                             #
r = 8    #Radius of turning (using one wheel)           #
d = 5.5    #Diameter of one wheel                       #
dist = (2*math.pi*r)*(theta/360.0)                      #
                                                        #
await motor_pair.move_for_degrees(motor_pair.PAIR_1,degrees,0,velocity=v) #,acceleration=180,deceleration=180)
#########################################################################################################
#Takes angle of turn (theta)
# Parameters: theta = degrees to turn                   #
async def turn(theta):                                  #
    degrees = int(dist*(360.0/(math.pi*d)))             #
    print(degrees)                                      #
#Takes distance in centimeters (cm)
async def forward(dist,v):
    d = 5.5    #Diameter of one wheel
motor_pair.pair(motor_pair.PAIR_1,port.C,port.F)
motor_pair.pair(motor_pair.PAIR_2,port.B,port.D)
import math
import motor_pair
import motor
from hub import light_matrix, port
import runloop
```

Red Run Code

CREATE: Boilerplate

Our boilerplate is a collection of code created to share frequently used and tested commands. This allows for relatively easy and efficient coding.

```python
from hub import light_matrix, motion_sensor, port
import motor_pair
import motor
import runloop

import math
import sys

print(sys.version)

# Some constants for calculations
d = 5.5     # Diameter of one wheel
r = 8       # Radius of turning (using one wheel)

# Takes distance in centimeters (cm), and stop (whether or not to stop moving after the instruction)
async def forward(dist:float,stop:bool=False,**kwargs):
    '''Move bot forwards by a number of centimeters.

    - `dist: float:` The distance in centimeters
    - `stop: bool:` Controls whether or not motors will be issued a stop command at the end of the function.
    - `**kwargs: dict[str, Unknown]:` Allows the user to pass acceleration/deceleration arguments optionally.

    Examples:
    ```
        forward(46)   # Move forwards 46 centimeters
        forward(-23)   # Move backwards
        forward(11,True) # Move forwards and stop at the end
    ```
    '''
    degrees = dist*(360.0/(math.pi*d))

    velocity = kwargs.get("velocity",360)
    if (degrees < 0):
        velocity = -1*velocity

    time = round((degrees/velocity)*1000)
    time = abs(time)

    print('FORWARD: degrees:{degrees},time:{time}, velocity:{velocity}'.format(degrees=degrees,time=time,velocity=velocity)) #Debug

    motor_pair.move(motor_pair.PAIR_1,0,velocity=velocity,**kwargs)
    await runloop.sleep_ms(time)

    if (stop == True):
        motor_pair.stop(motor_pair.PAIR_1)

# Takes angle of turn (theta), wheel (which wheel to turn with), and stop (whether or not to stop moving after the instruction)
async def turn(pivot_on:str,theta:int,stop:bool=False,**kwargs):
    '''Turn bot by a number of degrees.

    - `pivot_on: str:` The wheel that the bot will pivot on.
    - `theta: int:` The angle that the bot will turn.
    - `stop: bool:` Controls whether or not motors will be issued a stop command at the end of the function.
    - `**kwargs: dict[str, Unknown]:` Allows the user to pass acceleration/deceleration arguments optionally.

    Examples:
    ```
        turn("left",90)   # Pivot by 90° on the left wheel
        turn("left",-90) # Pivot back by 90° on the left wheel
        turn("right",180) # Pivot by 180° on the right wheel
    ```
    '''

    motion_sensor.reset_yaw(0)
    dist = (2*math.pi*r)*(theta/360.0)
    degrees = dist*(360.0/(math.pi*d))

    velocity = kwargs.get("velocity",360)
    if (degrees < 0):
        velocity = -1*velocity

    print('TURN: wheel:{pivot_on}, degrees:{degrees}, dist:{dist}, velocity:{velocity}'.format(pivot_on=pivot_on,degrees=degrees,dist=dist,velocity=velocity)) #Debug

    if (pivot_on == "right"):
        motor.stop(port.F,stop=1)
        motor.run(port.C,velocity,**kwargs)
        if (theta > 0):
            while (motion_sensor.tilt_angles()[0] < theta*10):
                await runloop.sleep_ms(1)
        elif (theta < 0):
            while (motion_sensor.tilt_angles()[0] > theta*10):
                await runloop.sleep_ms(1)
    elif (pivot_on == "left"):
        motor.stop(port.C,stop=1)
        motor.run(port.F,velocity,**kwargs)
        if (theta > 0):
            while (motion_sensor.tilt_angles()[0] < theta*10):
                await runloop.sleep_ms(1)
        elif (theta < 0):
            while (motion_sensor.tilt_angles()[0] > theta*10):
                await runloop.sleep_ms(1)

    if (stop == True):
        motor.stop(port.C)
        motor.stop(port.F)

    print('TURN ERROR:', theta*10, ' ', motion_sensor.tilt_angles()[0], ' ', (motion_sensor.tilt_angles()[0]-theta*10)/10)
    motion_sensor.reset_yaw(0)
```

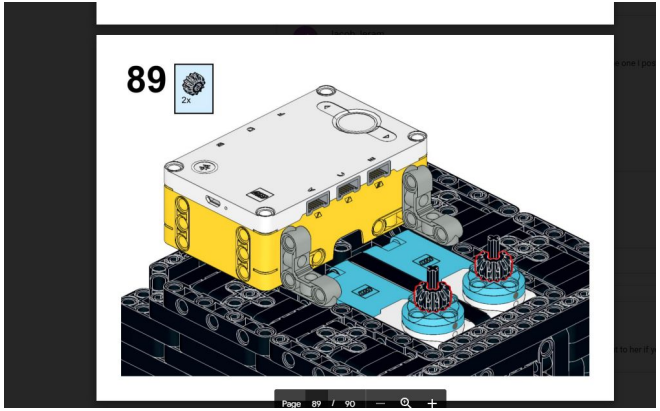
QR Code for our Code

GitHub

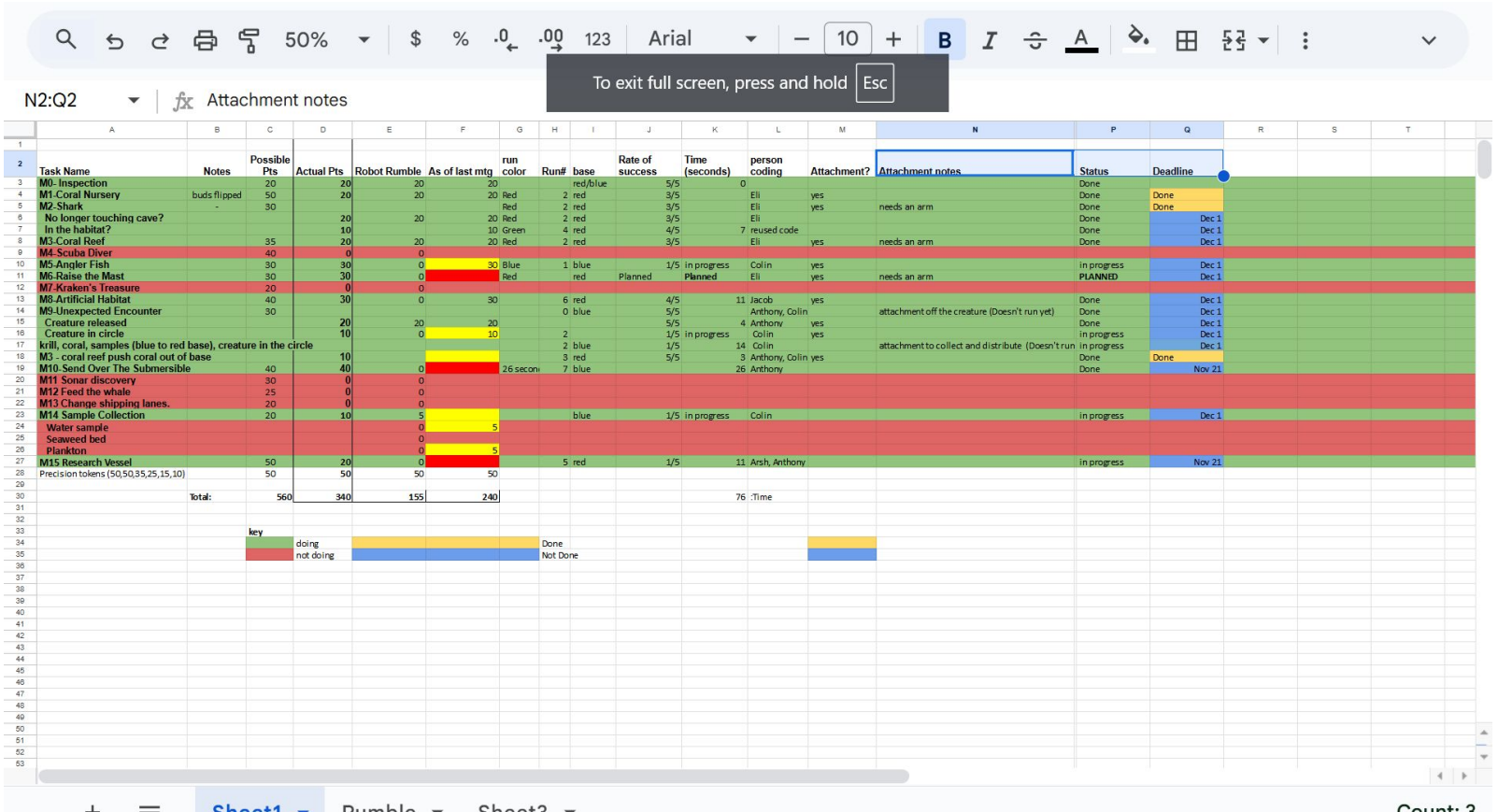
DESIGN & ITERATE: Robot Design Strategies

Bricklink

- New members built a backup using bricklink so that they were familiar with its design



Design & Iteration - Mission tracker



Task Name	Notes	Possible Pts	Actual Pts	Robot Rumble	As of last mtg	run color	Run#	base	Rate of success	Time (seconds)	person coding	Attachment?	Attachment notes	Status	Deadline
M0- Inspection		20	20	20	20			red/blue		5/5	0			Done	
M1-Coral Nursery	buds flipped	50	20	20	20	Red	2	red		3/5	Eli	yes		Done	Done
M2-Shark	-	30				Red	2	red		3/5	Eli	yes	needs an arm	Done	Done
No longer touching cave?			20	20	20	Red	2	red		3/5	Eli			Done	Dec 1
In the habitat?			10		10	Green	4	red		4/5	7 reused code			Done	Dec 1
M3-Coral Reef		35	20	20	20	Red	2	red		3/5	Eli	yes	needs an arm	Done	Dec 1
M4-Scuba Diver		40	0	0											
M5-Angler Fish		30	30	0	30	Blue	1	blue		1/5	in progress	Colin	yes		in progress	Dec 1
M6-Raise the Mast		30	30	0		Red		red	Planned	Planned	Eli	yes	needs an arm		PLANNED	Dec 1
M7-Kraken's Treasure		20	0	0											
M8-Artificial Habitat		40	30	0	30		6	red		4/5	11	Jacob	yes		Done	Dec 1
M9-Unexpected Encounter		30					0	blue		5/5		Anthony, Colin		attachment off the creature (Doesn't run yet)	Done	Dec 1
Creature released			20	20	20					5/5	4	Anthony	yes		Done	Dec 1
Creature in circle			10	0	10		2			1/5	in progress	Colin	yes		in progress	Dec 1
krill, coral, samples (blue to red base), creature in the circle							2	blue		1/5	14	Colin		attachment to collect and distribute (Doesn't run	in progress	Dec 1
M3 - coral reef push coral out of base			10				3	red		5/5	3	Anthony, Colin	yes		Done	Done
M10-Send Over The Submersible		40	40	0		26 secon	7	blue			26	Anthony			Done	Nov 21
M11 Sonar discovery		30	0	0												
M12 Feed the whale		25	0	0												
M13 Change shipping lanes.		20	0	0												
M14 Sample Collection		20	10	5				blue		1/5	in progress	Colin			in progress	Dec 1
Water sample				0	5											
Seaweed bed				0												
Plankton				0	5											
M15 Research Vessel		50	20	0			5	red		1/5	11	Arsh, Anthony			in progress	Nov 21
Precision tokens (50,50,35,25,15,10)		50	50	50	50											
	Total:	560	340	155	240						76	:Time				

key			
	doing		Done
	not doing		Not Done

DESIGN & ITERATE: Coding Strategies

- Attended summer class to learn python
- Veteran members taught newer members python
- Created a boilerplate of reusable code to make coding more efficient
- Commented our code so that others understand and can edit it

COMMUNICATE: IMPACT/Outreach

- Sharing code with Github and Bricklink on our website
- Shen FLL Robot Rumble - Nov 16
- Teaching club kids to code
- FLL Explore Expo - on January 25, 2025
- STEM night at Karigon Elementary
- FRC tournament at MVP arena
- Shen Science Health and Discovery Night

Thank you!

Questions?

Robot Rumble - November 16, 2024