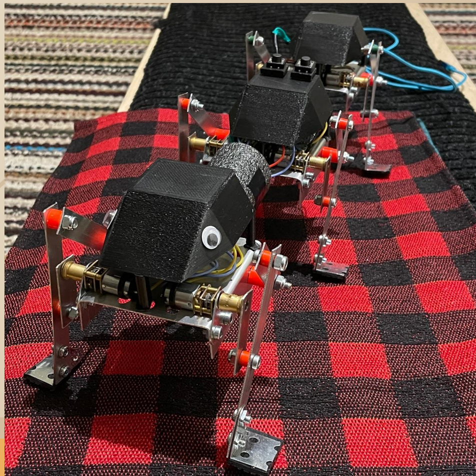
Myriapod Robot

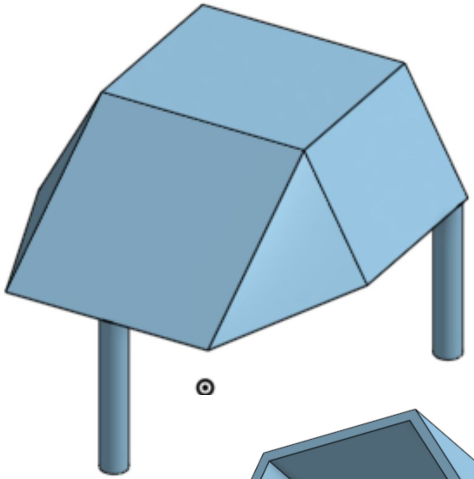
B.A.R.R.Y.

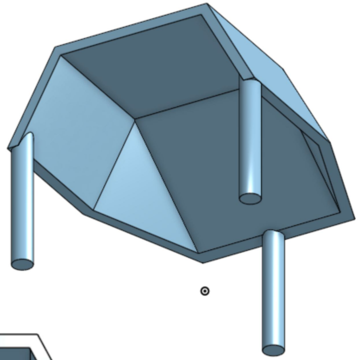
Barely Able Recon. Robot… Yeah

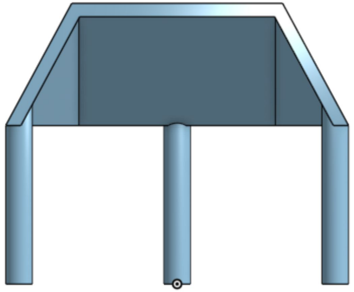
Best Powerpoint practices not followed due to this being supplemental information for the final presentation.

Katelyn Seto

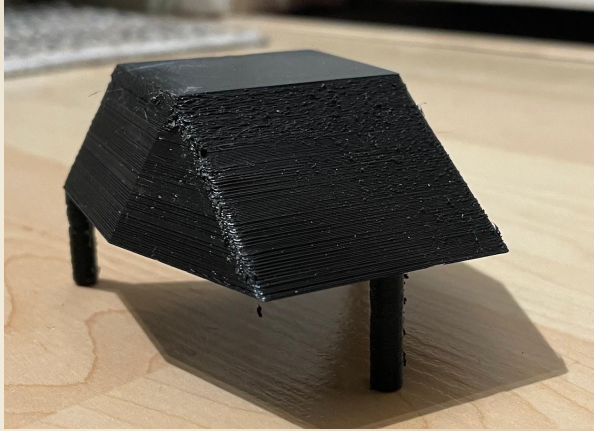




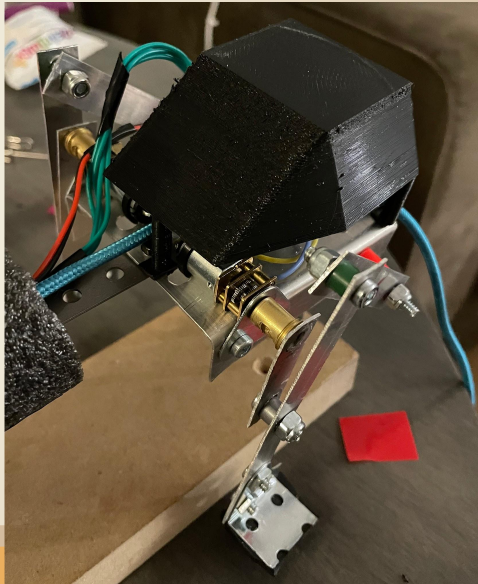




3D Printed Shell

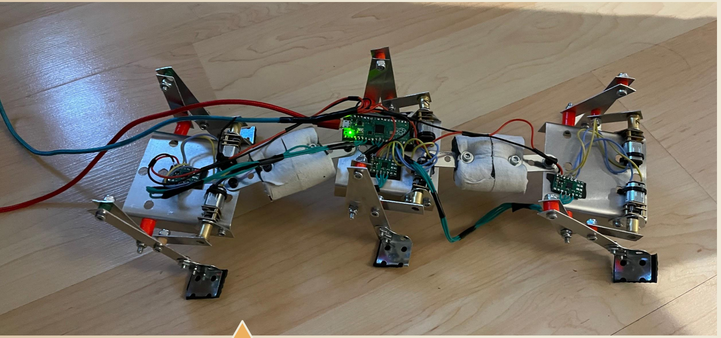


- Designed the shell to fit the dimensions of the aluminum body
- 3D modeled the shell structure with software called OnShape
- 3D printed 3 copies of the shell + cleaned up supports
- Installed onto the robot



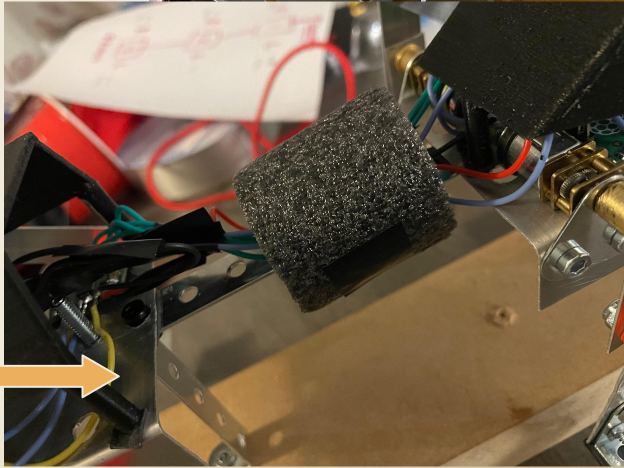
Connection Assembly

- Desired motion for the robot mirrors a myriapod and therefore requires both up-and-down motion but also left-and-right.
- Requires something semi-rigid. Copper pipe insulating foam noodle sourced and covered with hockey tape. Attached to metal links via screws resulted in desired flexibility while still holding the robot rigid enough to stand.



Robot mid-collapse due to twisting of the coupling.

- Robot was not able to handle the strain of its own weight and collapsed into a rowing motion. The decision was made to sacrifice the up-and-down motion and use a rigid coupling. This maintained the left-to-right motion enabling turning, but did not result in twisting that caused the robot to collapse once in motion.



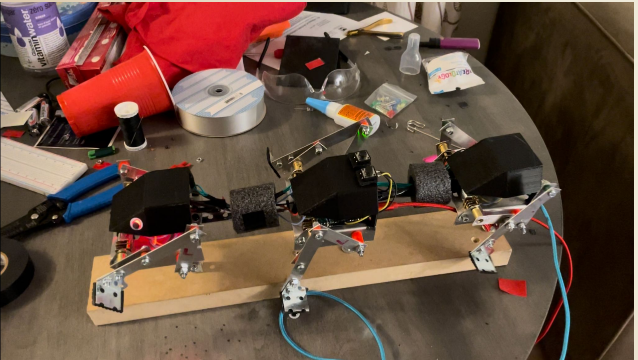
Testing: Step Motion

- Created code that had a left and right step function. Used that code in a step motion function and created a loop for the desired number of steps. When the function is called, the motion executes.
    - The “sleep” function determines how long the “step_left” motion is carried out. In the screenshot, it is carried out for 0.3 seconds.
    - Duration of each motion was determined by trial and error due to the motors reacting differently when on a tractive surface (carpet) and when on the stand.

```
#step code
def step_left():
    segment1left.backward(0.97) #backward due to wiring
    segment2right.forward(0.93)
    segment3left.backward(0.91) #backward due to

def step_right():
    segment1right.forward(0.93)
    segment2left.backward() #backward due to wiring
    segment3right.forward(0.94)

def step_motion():
    step_left()
    sleep(0.3)
    stop()
    sleep(0.3)
    step_right()
    sleep(0.3)
    stop()
    sleep(0.3)

x = 0
while x < 10:
     step_motion()
     x += 1
```

Percentages calculated in the “continuous” phase of testing to achieve as close to synchronous motion as possible.

- Results of testing were poor. The motion ended up lifting up both feet in the air simultaneously resulting in the robot collapsing.

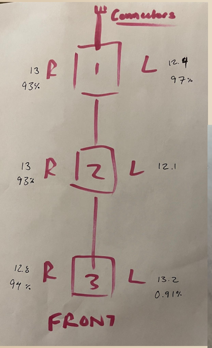


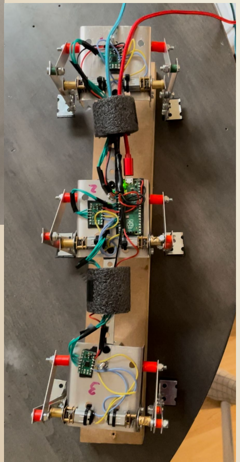
- Due to the code being time-based (DC motors provide no feedback for the code to use, therefore the only control available to me was duration), there were significant problems with determining when the full step was carried out as it varied highly.

- Motion was determined to not be viable with the remaining time and support.

Testing: Continuous Motion

- Used a video to gauge how many times each motor rotated within 10 seconds to calibrate the amount the other motors had to be reduced by in order to even out the speed.
  - Fluctuations caused by friction and differences in both the assembly and motors themselves.

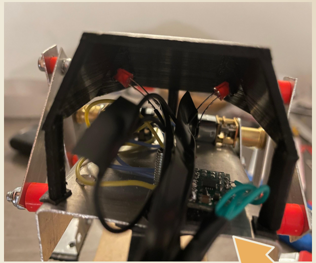


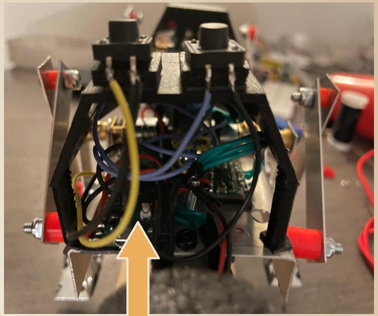


- Calibrating was not perfect and legs continued to synchronize periodically. Testing was taken to the ground and off of the stand.
- Robot proceeded to slam itself repeatedly into the ground and only move in a rowing motion. This was due to the traction of the ground affecting how much the motor is properly able to spin.
- Determined additional problem came from the weight. Resulted in attempting to provide support via a sling (pictured above).
  - Limitations of weight and ability of the motors unable to be overcome due to time and cost constraints.
- Resulting in the robot being able to walk with the appropriate motion style for 8 seconds before synchrony occurred once more. This is sufficient for a demo.

Testing: Incline

- Added 2 buttons: 1 for reset, 1 for initiation.
- Added 2 LEDs for the eyes. Also added 2 googly-eyes.
- Tested demo (continuous for 8 seconds with the support sling and without).
    - Robot was able to climb up and down a 10 degree incline covered in carpet.
    - Robot achieved superior motion with the sling on the second segment.
    - Robot was able to climb and descend the incline without the assist of the sling and with the rowing motion.





Interior of segment 1 & 2 assembled with the aid of Colin

Climbing and descending unassisted. Partial crawling and partial rowing motion.

```
#Notes
    #left side segment 2 has more resistance
    #'left' motors have to be backward
    #carpeted surface achieves best results
    #0.25 speed is too slow to move on carpeted surface
    #0.5 speed can run on carpet

#hardcoded demo
#moving forward
    #legs up in the air
segment1left.backward(0.97) #backward due to wiring
segment2right.forward(0.93)
segment3left.backward(0.91) #backward due to wiring
    #legs down on the ground
segment1right.forward(0.93)
segment2left.backward() #backward due to wiring
segment3right.forward(0.94)
sleep(8) #max run time w/ sling assisstance before it syncs up
stop() #halt all motors
```