Engineering Portfolio

Eric Broyles

Aerospace Engineering, B.S. — Purdue University

github.com/EricBroyles
linkedin.com/in/eric-broyles
eric.c.broyles@outlook.com

Professional Summary

I am a recent Aerospace Engineering graduate actively seeking challenging roles in unmanned systems, hypersonics, testing and validation, modeling and simulation, or systems engineering and design. I bring a strong foundation in aerospace fundamentals, programming experience in Python and MATLAB, growing proficiency in C++, and hands-on experience with CAD, CFD, technical writing, and technical presentations. I am highly motivated by complex problems and continuous learning. The best way to reach me is via email.

Portfolio Overview

Below is a collection of projects completed during my undergraduate studies. Many of these projects include codebases and additional results that can be found in more detail at:

github.com/EricBroyles/Portfolio

Selected Projects

PRIME Research Project

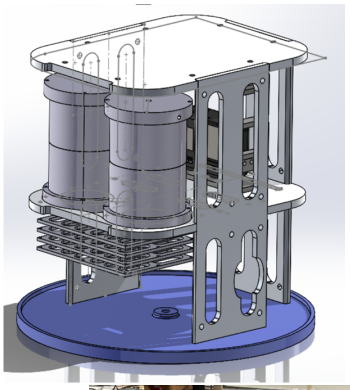
Purdue Rocket Instrumentation and Measurement Experiment

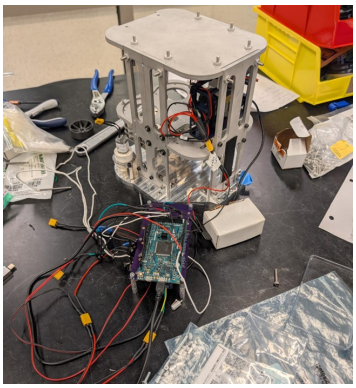
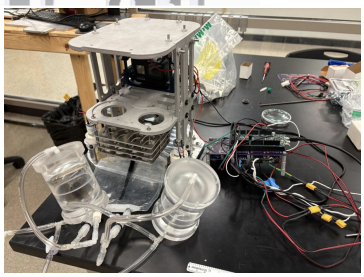
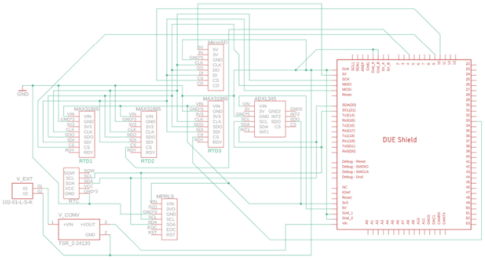
Project Overview
PRIME is a microgravity fluid-dynamics payload to be launched on a LEAP rocket. The project's goal is to study liquid transfer behavior between coupled tanks in zero gravity. Recent efforts focused on manufacturing key structural components, refining tank geometry, and enhancing electronics and imaging systems.

Technical Description
The PRIME payload consists of two transparent acrylic tanks connected through a sealed plumbing system with pumps that transfer fluid under microgravity conditions. Structurally, the payload employs 6061-T6 aluminum components to withstand launch loads and a mounting system for the GoPro Hero 8 camera. The payload integrates four LEDs for uniform lighting in the unlit rocket bay. Software control sequences are managed by C++ code that triggers sensor activation, lighting, camera recording, and pump operation in timed stages during flight.

My Role & Accomplishments
October 2025 — December 2025

- Wrote the C++ code to manage the launch sequence, automate sensor and camera activation, and control fluid transfer operations.
- Resolved hardware-software integration issues to ensure reliable data collection and payload performance.
- Authored launch documentation to support ongoing development and future team efforts.



RF Interference Classification Research Project

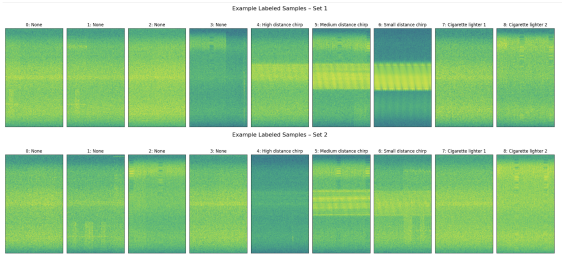
Purdue Data Mine

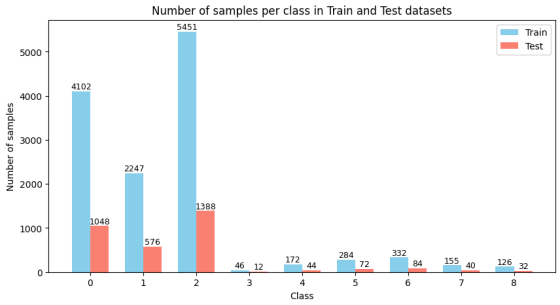
Project Overview & My Role

August 2025 — December 2025

I worked to develop machine learning models to classify nine types of radio-frequency interference observed along a German highway. The primary challenges were severe class imbalance and the similarities across the spectrogram data.

Model development was performed with Python and PyTorch using Purdue's Anvil supercomputer. I trained and evaluated multiple model architectures while experimenting with data normalization and augmentation techniques, including noise, dropout, and mix-up, to improve robustness and minority-class performance.

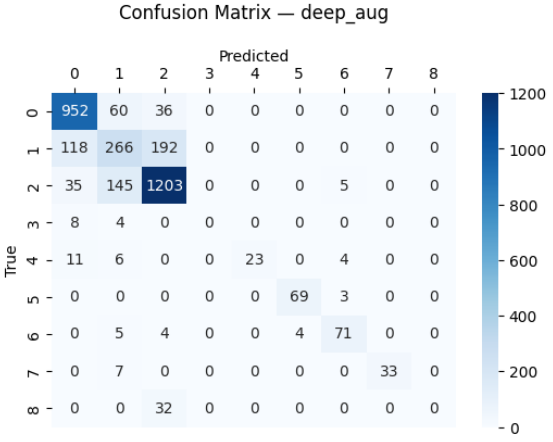




Best CNN (Baseline Model)

A convolutional neural network trained on normalized spectrogram data with targeted noise augmentation. This model serves as the baseline against which all subsequent approaches were evaluated. While achieving strong overall accuracy, it struggles with minority classes, particularly classes 3 and 8.

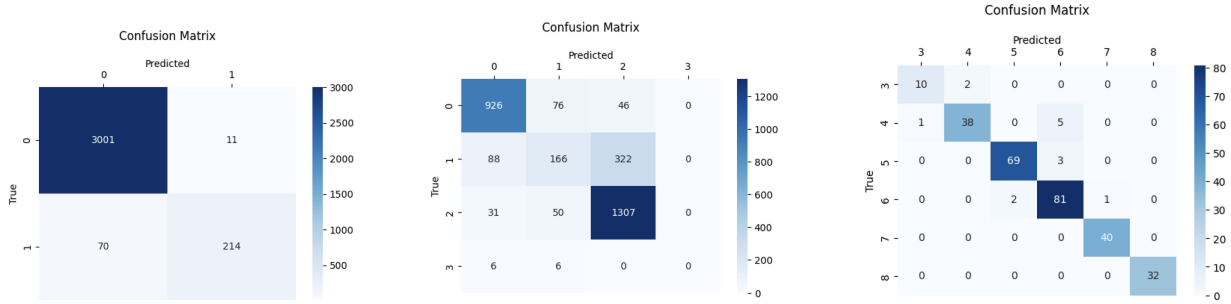
Accuracy: 79.4%
Loss: 0.5343
Parameters: 4.3M
Noise: Classes 1, 3



Best Hierarchical CNN

A hierarchical CNN architecture was developed using a binary classifier followed by sub-classifiers (0–3 and 3–8). While the 3–8 classifier performed well in isolation, the binary classifier's poor detection of classes 3 and 8 limited overall effectiveness. This method did not improve upon the standard CNN .

Model	Description	Acc	Loss	Notes
CNN Binary	no aug, 8 epochs, 4.3M params	97.5%	0.0879	0% class 3, 6% class 8
CNN 0to3	time & freq dropout, 8 epochs, 4.3M	79.3%	0.5242	0% class 3
CNN 3to8	time & freq dropout, 12 epochs, 4.3M	95.9%	0.1289	83% c3, 100% c8

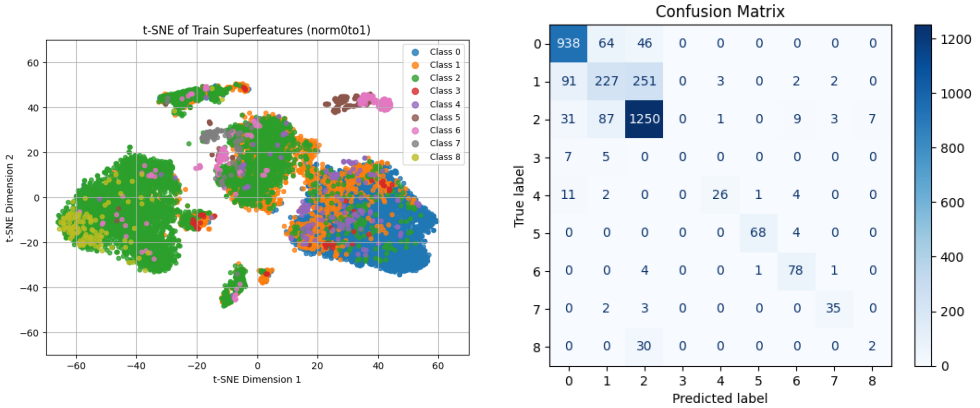


Best Neural Network

Spectrograms in the highway dataset are mainly noise variations with some signal patterns. Each spectrogram was described using "superfeatures" encoding statistical parameters (mean, median, standard deviation, etc.), reducing dataset size and helping models focus on descriptive data. This approach proved slightly more accurate than CNNs and trained 10x faster.

Superfeatures: Each sample is transformed into 1,344 features by resizing spectrograms from 512×243 to 128×64, then extracting statistical features (mean, std, median, min/max and their locations) across rows and columns, normalized using 0-to-1 scaling or z-score.

Model	Description	Acc	Loss
run2 cell1	superfeatures, 25 epochs, 822K params	79.6%	0.5428



Best Voting Models

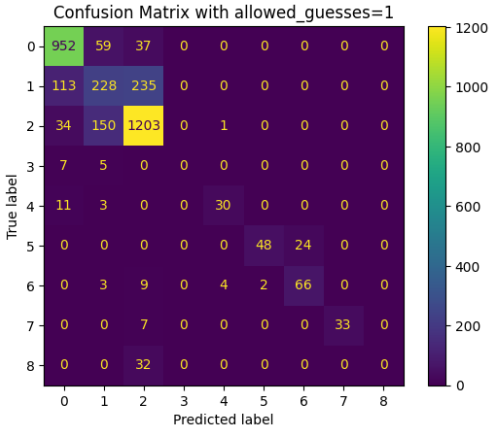
Voting models combine neural networks (which struggle with classes 1, 3, 8) with linear models (better at 1, 3, 8 but weaker on 0, 2). **Unfitted voting** sums probabilities across models. **Fitted voting** uses probabilities as features for a neural network. Neither approach improved over the standard NN.

Superfeatures: Each sample is transformed into 2,112 features using extended statistical measures including percentiles (75th, 25th, 90th, 10th) in addition to the previous features.

Model	Description	Accuracy G1	Accuracy G2
NN1	8 epochs, 1.2M params	78.2%	95.4%
NN2	8 epochs, 2.0M params	78.8%	95.7%
NN3	6 epochs, 5.5M params	78.0%	95.8%
Linear1	Superfeatures	63.6%	80.3%
Linear2	No mean, std	63.7%	79.8%
Linear3	No mean, std, percentiles	63.6%	80.0%

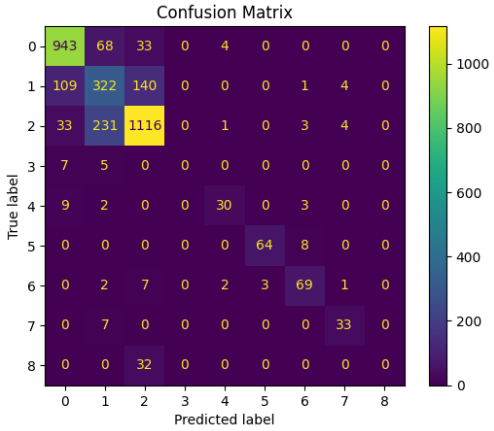
Unfitted Voter

Accuracy 77.7%



Fitted Voter

Accuracy 78.2%



PREVIEW Research Project

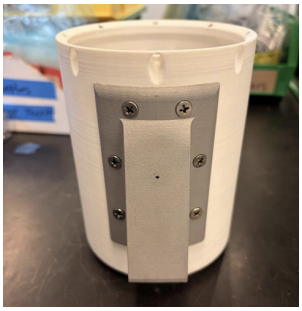
Purdue Rocket Experimental Video in Educational Work

Project Overview
PREVIEW is a student-led project developing a payload to collect sensor data and video footage of a rocket's exterior during hypersonic flight with PLUTO Aerospace. The payload withstands accelerations up to 150 g. During fall 2025, the team manufactured aluminum wedge structures, assembled the PCB through OSH Park, and developed launch and data acquisition software.

Technical Description
The payload uses a Raspberry Pi Zero with Pi Camera for video capture. Sensors include two pressure transducers (for Mach number estimation), an ADXL375 accelerometer (high-G measurements), and a K-type thermocouple (internal temperature monitoring). Components are housed in a circular frame with stacked disks separated by spacers to reduce mechanical stress. Two 9V NiMH batteries provide power, regulated to 5V for electronics.

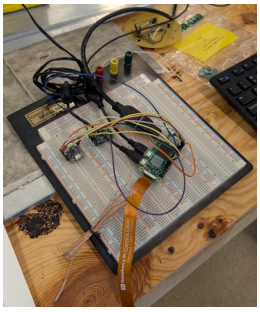
My Role & Accomplishments
January 2025 — October 2025

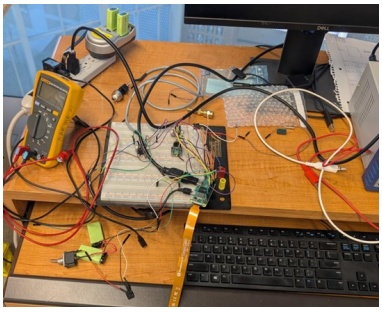
- Contributed to system design considerations for electronics integration.
- Improved SolidWorks CAD prototypes for structural and electronic components.
- Prototyped PCB designs using breadboards and aided layout development with Fusion 360.
- Improved Python scripts for launch sequence control and sensor data acquisition.
- Configured Raspberry Pi for WiFi power management and auto-run launch software on power-up using systemd service files.

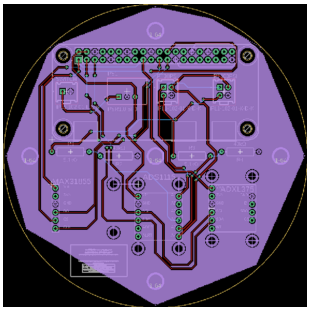

Camera Front Wedge


Camera Bottom Wedge


CAD Camera Front Wedge


Early Breadboard Prototype


Final Breadboard Prototype


PCB Design

AAE 451: Group 4

Design, Build, Fly

Eric Broyles, Achintya Gupta, Joey Massaro, Keegan Sharp and Belle Vasquez

PURDUE UNIVERSITY | School of Aeronautics and Astronautics

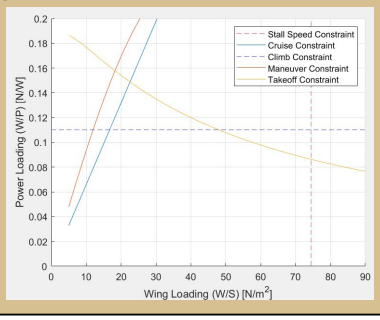
Design & Analysis

Requirements

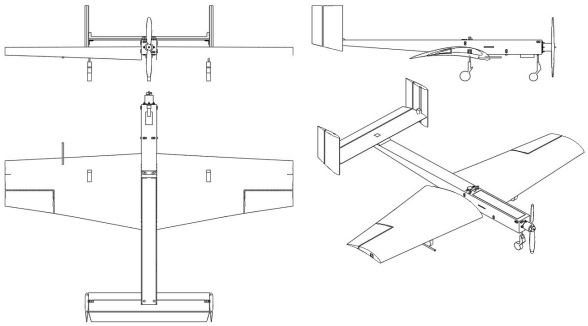
Size & Weight: Fits within a 75x75x150cm storage volume, max 150cm wingspan, and MTOW ≤ 6kg.
Performance & Handling: Takeoff within 25m, soft-field landing, stable and controllable with or without payload, and easy for an external pilot to fly.
Payload Requirements: Must carry at least one 518g steel bar with dimensions 2.75×0.25×6in.
Setup & Operations: Transition from storage to flight in under 5 min, payload and battery installed in under 2 min, and battery connects without removing components.
Power and Budget: Propulsion system limited to 1000W max electrical power, total budget is $400.

Design Parameters

Wing Airfoil: Eppler E423
Wing Area: 0.5m^2
Aspect Ratio: 4.5
MAC: 0.34m
Taper Ratio: 0.6
Root Chord: 0.42m
Tip Chord: 0.25m
Wing Sweep: 0°
C_D Trimmed: 0.03
C_D Untrimmed: 0.024
Tail Airfoil: NACA 0012
H-Tail Configuration



Aircraft & Manufacturing

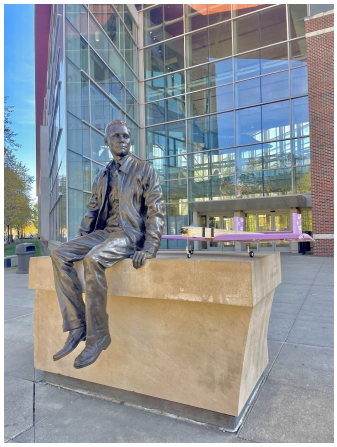


Propulsion System

Motor: Cobra C-3520/14 Brushless Motor
ESC: RC Electric Parts Brushless ESC Classic 4S 60A
Battery: Turnigy 4000mAh 4S 30C LiPo Pack
Propeller: APC 13x6.5E

Manufacturing Changes

- Smaller nuts & bolts
- Full box structure for tail boom
- Tail servos in the center back of tail boom
- ESC moved to the top of the fuselage



Weight Table

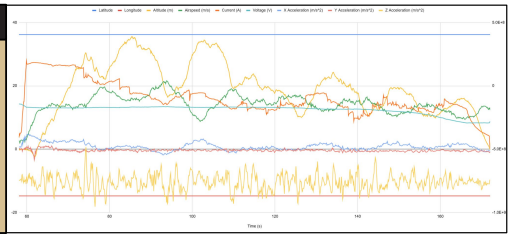
Parameter	Predicted (kg)	Actual (kg)
Empty Weight	2.875	2.483
Battery Weight	.452	.399
Total Weight	3.845	3.4

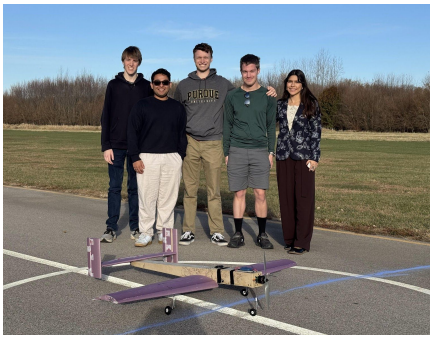
Flight Performance & Results

Flight Test Issues

Tail Flutter:
At high power settings the horizontal & vertical stabilizers would flutter due to insufficiently ridgid attachment to fuselage

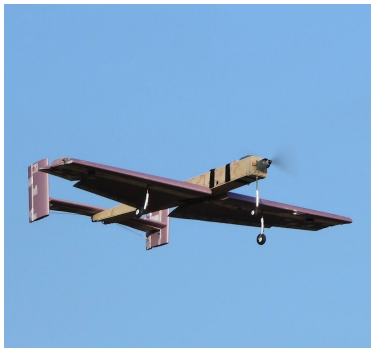
Crash Landing:
An insufficiently charged battery caused a loss of power, resulting in an uncontrolled descent and crash landing





Lessons Learned

Avoid Sharp Internal Corners in notched components, as they create major stress concentrations and weaken the structure.
Wing-spar was over-engineered; reducing its cross-section would significantly cut weight and improve its fit within the thin wing-tip profile.



RFP Requirements Verification

Parameter	PASS or FAIL
Can be stored within 1.5 x 0.75 x 0.75 m box	FAIL
Wingspan under 1.5 m	FAIL (1.51 m)
MTOW under 6 kg	PASS (3.4 kg)
Takeoff in under 25 m	PASS (15.95 m)
Climb to 30 m	PASS (35.6 m)
Successful Soft Field Landing	FAIL (Structural Failure)
Avoid Property Damage	PASS
Under 1000 W of Power Draw	PASS (368.6 W)
Connect Battery in Safe Manner	PASS
Assemble Without Battery / Payload in 5 Minutes	PASS
Install Battery / Payload in Under 2 Minutes	PASS
Stable / Controllable with & without Payload	PASS
Easy to Fly by External Pilot	PASS
Part Cost Under $400	PASS

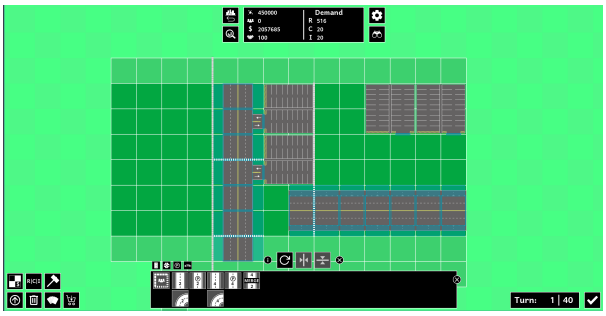
Transportation Simulation Game

Personal Project

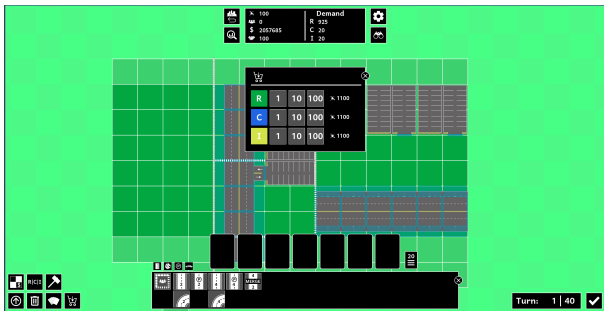
Project Overview & My Role
January 2025 — Ongoing

Development of a large-scale urban transportation simulation capable of modeling a 4096×4096 cell city (each cell representing 10 ft) with up to 65,536 autonomous agents navigating using A* pathfinding algorithms. The project aims to simulate realistic traffic patterns and pedestrian behavior across an urban environment.

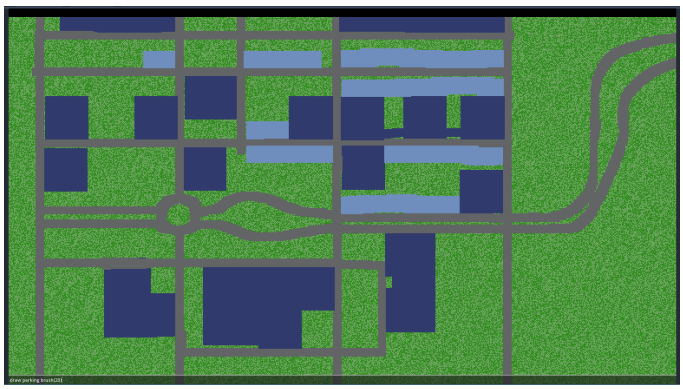
Technical Description
Early prototypes focused on learning the Godot Engine, UI development, and GDScript (a Python-like language). However, performance bottlenecks emerged when scaling to the target number of agents and city size. Recent development shifted to GPU-accelerated rendering using custom shaders for the cityscape visualization and C++ for high-performance agent simulation. The current implementation features a shader-based rendering system and a command-based interface for generating city layouts.



Early prototype: example city



Early prototype: some additional UI



Current prototype: shader-based rendering with command tool (bottom)

Autonomous Maze Navigation Robot

Course Project

Project Overview
Design and build an autonomous robot to navigate a 12 ft by 12 ft maze containing obstacles with IR signatures and magnetic signatures detectable via hall sensors. The robot must track its position and generate a visual representation of the completed maze.

Technical Description
The robot platform utilized a Raspberry Pi as the main controller, IR detectors, hall sensors, gyroscope, and ultrasonic sensors. Ultrasonic sensors detected maze walls for navigation, while IR detectors and hall sensors identified and classified different obstacle types. Position tracking was achieved through odometry and motor sensor feedback, with real-time mapping algorithms generating a representation of the maze layout as the robot progressed through the course.

My Role & Accomplishments
February 2022 — April 2022

- Programmed the robot in Python, implementing navigation algorithms and sensor integration.
- Debugged and resolved Raspberry Pi hardware-software integration issues.
- Successfully completed the maze navigation challenge, meeting all project requirements.