# Design Notes for the E-Vtol Simulation Project

Kodiak North

## 10/09/2024 – Initial list of notes/tasks/questions to guide the design

* First and foremost, I’ll need to configure VS Code to compile and run C++ executables.
* What are some unit tests I can incorporate into the code?
  + The first idea that comes to mind is a test to ensure all 5 vehicle types end up in the random list of 20 every time it is created.
  + I can also design a test to remove battery life from the equation and ensure each vehicle type will fly the correct distance given 1 hour of simulated time.
  + Similarly, I can test that the batteries drain in the right amount of time.
* I should obviously create a parent class to define an e-vtol vehicle with the provided characteristics, and then child classes for each specific vehicle type will handle the intricacies between different velocities, number of passengers, etc.
* How will I handle the possible generation of a fault?
  + A 25% chance of a fault per hour does not necessarily mean that a fault will occur every 4 hours!
* So the simulation does not actually take 3 hours, 1 loop step can be considered 1 second, or even 1 minute of elapsed time.
* How will the equation for energy consumption work?
  + Say cruise speed is 100 miles/hour, battery capacity is 50 kWh, and energy use at cruise is 2 kWh/mile…
  + How long the battery will last flying at cruise speed?
    - 50 kWh / (2 kWh/mile) = 25 miles… => 25 miles / (100 miles/hour) = 0.25 hours… => 0.25 hours \* (60 min/hour) = 15 min.
  + So the equation is:
    - ((Battery capacity [kWh] / energy use at cruise [kWh/mile]) / cruise speed [miles/hour]) \* 60 [min/hour] = duration in minutes
    - Will have to incorporate the loop step into this! Or perhaps this value could be stored for each vehicle type. As the system loops, it increments a counter. Once the counter equals the calculated duration, the battery is dead.
* Design a state machine for each vehicle. Upon initial thoughts, the states could be:
  + Idle, flying, charging, or waiting to charge
* Will need create a queue to hold vehicles that are waiting to charge, and also track chargers that are available or not.
* How will I assign a vehicle to a charger? It can probable be as simple as chargers 1, 2, and 3 are in the queue of available chargers. When a vehicle needs a charger, it is dequeued and its ID populates the ‘current charger’ member variable for that vehicle. Once charging is complete, the charger’s ID is enqueued back into the queue of available chargers.
  + If the queue of available chargers is empty, that means there are no available chargers and a vehicle will have to be placed in line.
* As for recording metrics:
  + If I track the flight time, I think I’ll know each vehicle’s distance travelled based on the cruise speed.
  + I can record vehicle charge time and store it in a member variable.
  + Store number of faults in a member variable.
  + Total passenger miles can likely also be calculated at the end of the simulation based on flight time.

Okay! So this is a lot of information spewed out. It’s time to organize it into a list of priorities, and concretely define the data structures that I’ll use.