

Traffic Drones - Full stack Exercise

Scenario:

There are two aerial drones that fly around London and report on traffic conditions. The drones are controlled by a central dispatcher that issues commands to specify their flight paths - a sequence of coordinates that should be visited at the specified times.

At each point on the flight path, each drone calculates the distance to the nearest London Underground tube station. If that station is within a certain range, the drone assesses the traffic conditions in the area, and generates a report.

Task:

Write a simulation that models this scenario.

- The two drones have IDs 6043 and 5937.
- The flightpath coordinates are provided in CSV files with the corresponding names.
- Another file (`tubes.csv`) contains the coordinates of each tube station to be taken into account.
- Each drone should operate independently on a separate thread. The dispatcher should read the coordinates from the supplied CSV files and deliver them to the drones.
- The drones should report on traffic at each point as described above. The reports should be delivered to a central report consumer as the simulation progresses.
- The report consumer should make the reports available to the UI as they arrive, either on request or via some push mechanism.

Notes:

- The drones have limited memory, so they can only consume ten points at a time.
- The format for `tubes.csv`:
 `station-name, latitude, longitude`
- The format for the flightpath CSV files:
 `drone-id, latitude, longitude, time`
- The 'time' of each coordinate represents the number of seconds from the start of the simulation that the drone should visit that point.
- Traffic reports should include the following information:
 - Drone ID
 - Time
 - Name of nearest tube station
 - Conditions of Traffic (HEAVY, LIGHT, MODERATE - can be chosen randomly)
- The threshold to determine whether a tube station is near enough to generate a report is 350 metres.

Deliverable:

- The assignment should be delivered as a webapp that allows the user to start the simulation and view the reports as they are generated.
- The communications between the dispatcher and the drones should be logged (to console, log file or via the web UI)
- Push your solution to a private git repository and grant access to us when you're happy with it.
- Include a README.md listing prerequisites (e.g. JDK version, any other runtimes required) and simple instructions on how to build and run the application.
- Feel free to include further documentation if you have time and feel it necessary (but this is not a requirement).

Tips:

- This is a fairly open assignment in terms of how you structure the solution. You will be judged on the overall quality of the code, particularly in terms of:
 - Clean, succinct, self-documenting code
 - Package and class structure
 - Adherence to OOP best practices and patterns
- Keep your solution simple.
- Feel free to use libraries for things like CSV parsing and calculating geographic distances.
- Use libraries and frameworks that you are familiar with - it is not recommended that you use this exercise as an opportunity to try out some new technology you've never used before.