Collaborative Surveillance of Large Geographical Area by Fleet of Drones

May 1, 2023

Group sd-May23-50

Rowan Collins, Joseph Edeker, Jaden Forde, Thomas Glass, Jacob Houts, Marcus Jakubowsky

Project Advisor: Dr. Goce Trajcevski
Graduate Researcher: Prabin Giri

Introduction

- The use of drone fleets for large area surveillance is a rapidly growing field, but testing is costly and difficult
- Our project aims to provide a simulation application for testing drone fleet algorithms to ease these challenges

Use Cases

Use Cases:
- City Planning (see figure)
- Map Surveying
- Agricultural Surveying
- Forest Fire Monitoring

Users:
- Municipal Planners
- Government Officials
- Farmers

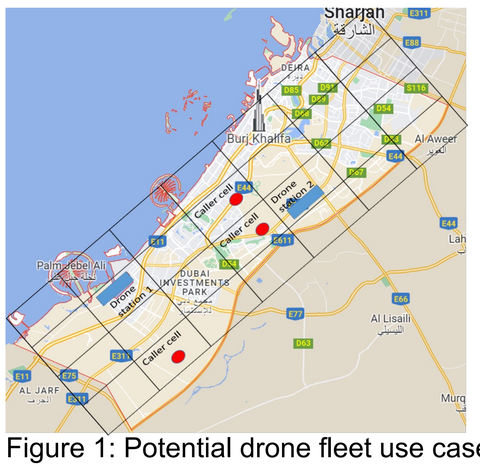

Figure 1: Potential drone fleet use case for surveying an urban area

Requirements

- Visualization of the drone flight using a 2d grid.
- Comparison of multiple drone test runs.
- Allows the user to login to a personal account profile.
- Simulation setup configuration through selected files.
- Software is easy to use and understand.

Architectural Design

Four Dockerized Applications:
- Java Backend Handling
- Python Algorithm Execution
- MySQL Database Storage
- Vite and React Frontend

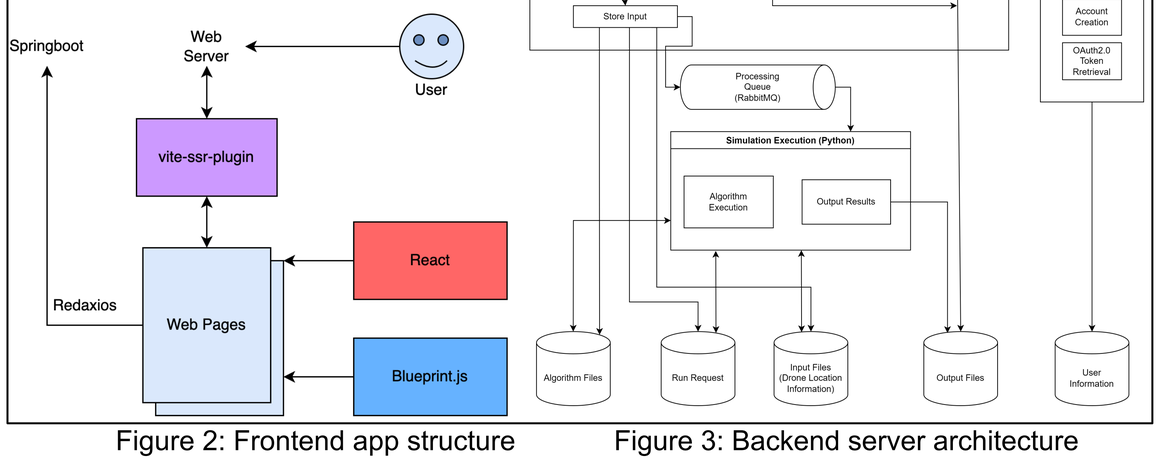

Figure 2: Frontend app structure

Figure 3: Backend server architecture

Technologies

User Interface:
- Vite
- React
- Redaxios

Server Backend:
- Spring Boot
- RabbitMQ
- Python
- MySQL

Testing

Methods:

JUnit 5 and Mockito
- Unit tests for backend
- CI/CD Pipeline runs tests during deployment of code

Postman
- Testing REST API calls

Manual Testing
- Frontend and algorithms

Application Views and Results

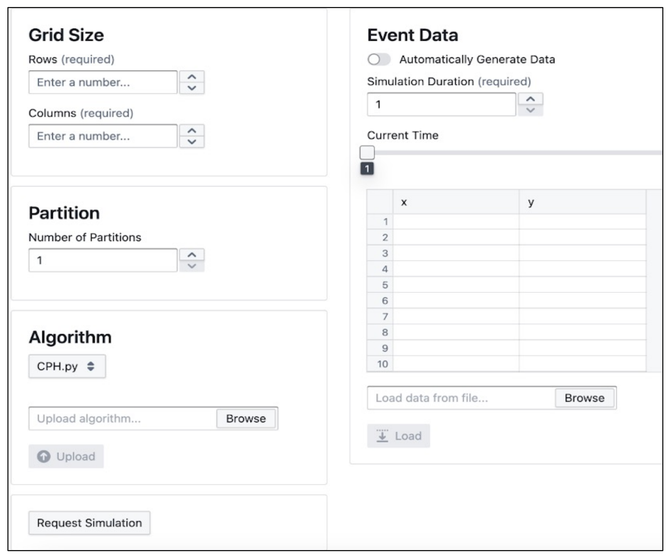

Figure 4: Web interface - simulation setup screen. Grid size, partitions (number of extra drones), algorithm selection, and event input from file upload or input entry

Figure 5: Web interface - Request dashboard. Completed and in-progress simulations can be viewed.

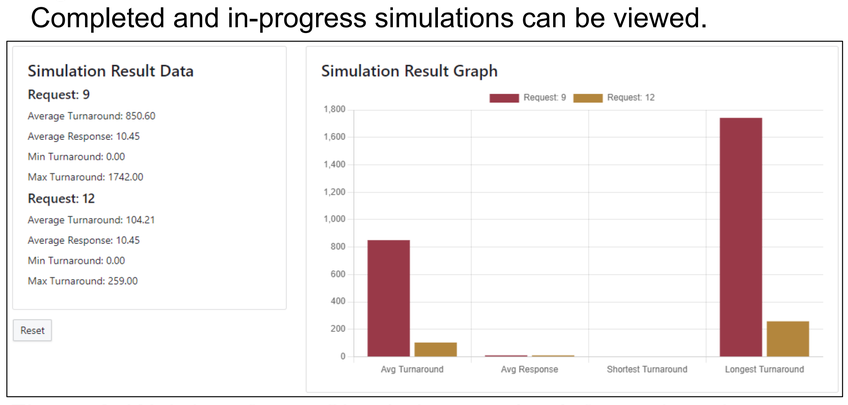

Figure 6: Web interface - Simulation results. Information about simulations may be viewed and compared with that of others.

Two Algorithms were provided for simulation:
CPH: Drones cascade hop along the path, replacing next in line. Event reached in 1 time.
Naive: A single drone flies around from event to event, grid drones are stationary (below)

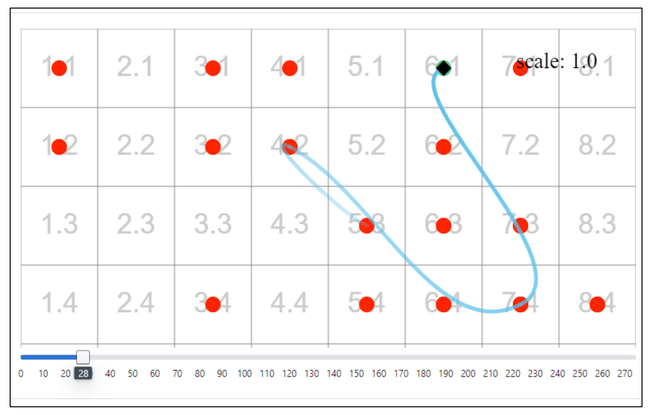

Figure 7: Web interface - drone path visualization. Red dots are events, which occur at a given time. The black dot is the drone, which paths around to events as they happen. Slider (bottom) allows the user to scroll through time and observe motion.

Security

Spring Security
- Username/Password
- Authentication Token
- Token needed for all requests

Results and Conclusions

- Functional application to run simulations with custom input and view results
- Useful in research to demonstrate complex algorithms
- Low coupling allowing for extendable functionality, allowing for use in future research endeavors

Future Work

- Custom drone algorithms
- Visualization for CPH algo
- Input hashing for security
- Email users upon completion