FlytBase Robotics Assignment 2025
UAV Strategic Deconfliction Demo Walkthrough
By Kartik singh

This document provides a walkthrough of the system behavior, output, and visualization in place of a recorded video.

The system checks whether a primary drone's mission conflicts with scheduled drones in space and time.

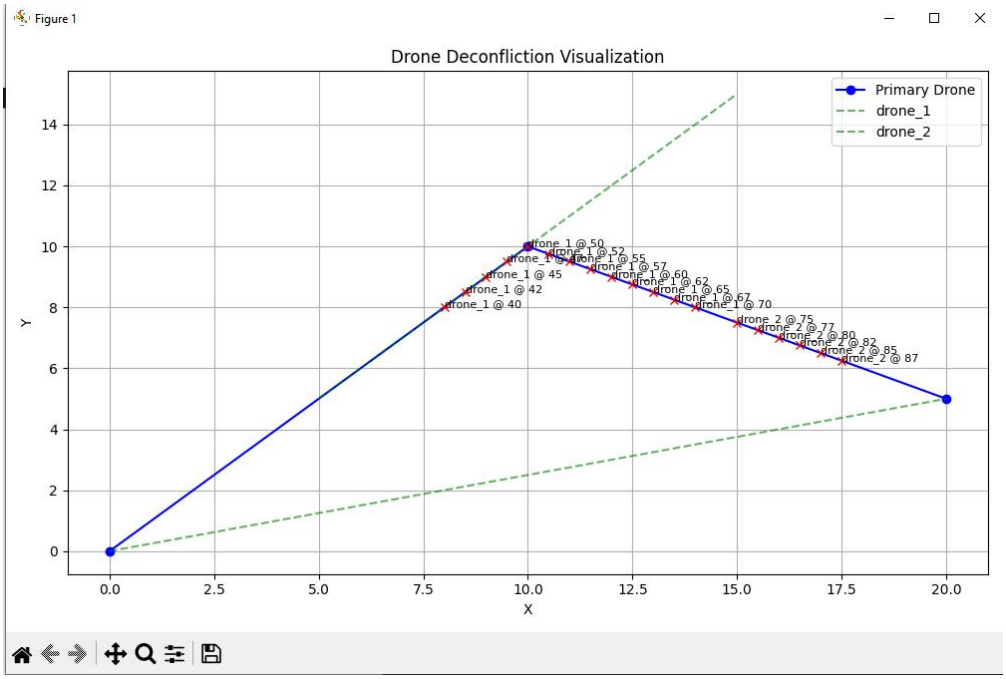
Key modules:
- conflict_checker.py: performs spatial + temporal checks
- visualizer.py: generates 2D + 4D graphs
- main.py: runs the process and outputs conflict reports

{'status': 'conflict_detected', 'conflicts': [
  {'location': (8.0, 8.0), 'time': 40.0, 'conflicting_drone': 'drone_1'},
  ...
]}

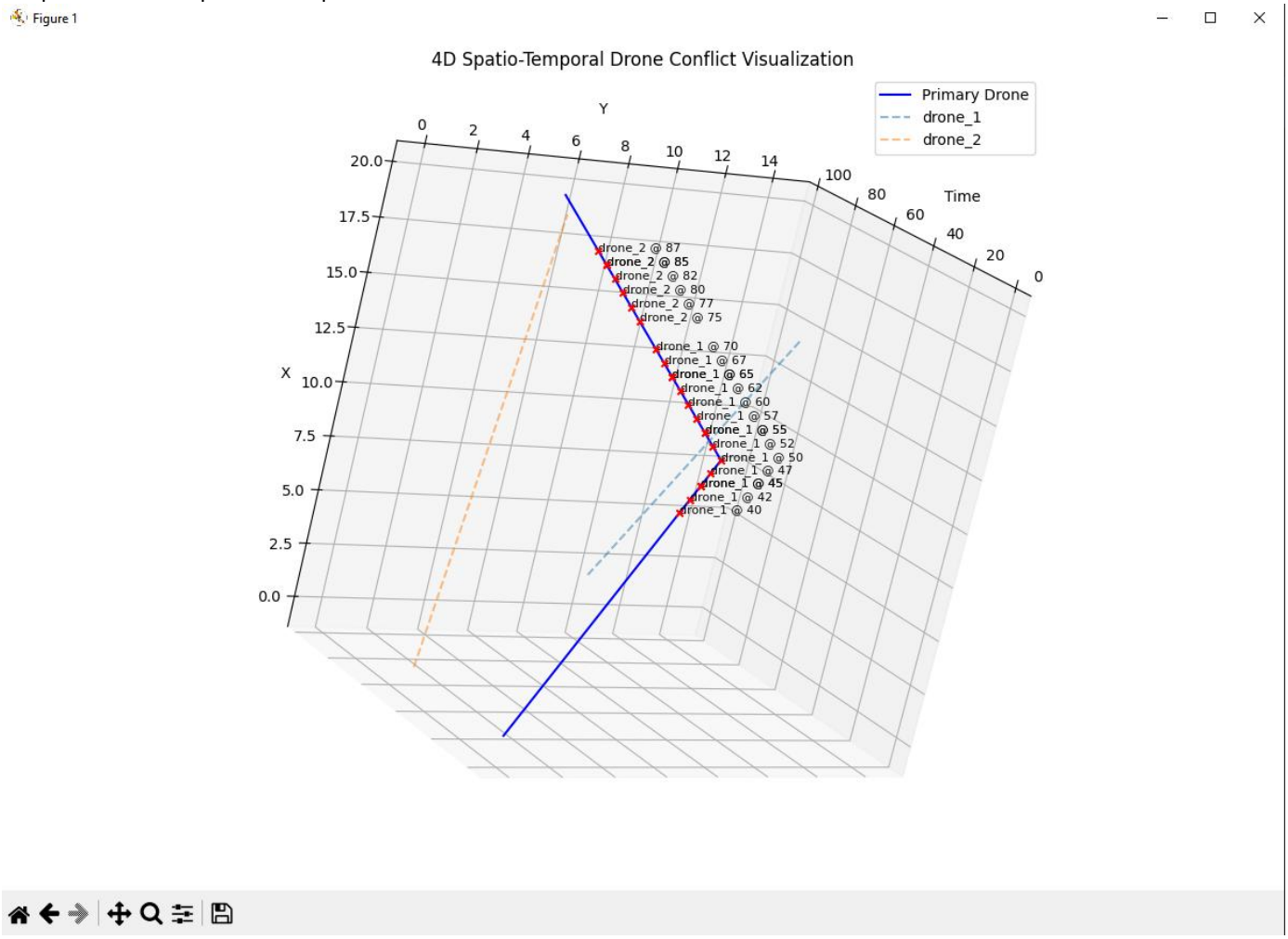
Terminal Output:
Shows printed conflict logs (drone ID, time, location)

```
PS D:\New folder (3)> python3 main.py
{'status': 'conflict_detected', 'conflicts': [{'location': (8.0, 8.0), 'time': 40.0, 'conflicting_drone': 'drone_1'}, {'location': (8.5, 8.5), 'time': 42
.5, 'conflicting_drone': 'drone_1'}, {'location': (9.0, 9.0), 'time': 45.0, 'conflicting_drone': 'drone_1'}, {'location': (9.0, 9.0), 'time': 45.0, 'conf
licting_drone': 'drone_1'}, {'location': (9.5, 9.5), 'time': 47.5, 'conflicting_drone': 'drone_1'}, {'location': (10.0, 10.0), 'time': 50.0, 'conflicting
_drone': 'drone_1'}, {'location': (10.5, 9.75), 'time': 52.5, 'conflicting_drone': 'drone_1'}, {'location': (11.0, 9.5), 'time': 55.0, 'conflicting_drone
': 'drone_1'}, {'location': (11.0, 9.5), 'time': 55.0, 'conflicting_drone': 'drone_1'}, {'location': (11.5, 9.25), 'time': 57.5, 'conflicting_drone': 'dr
one_1'}, {'location': (12.0, 9.0), 'time': 60.0, 'conflicting_drone': 'drone_1'}, {'location': (12.5, 8.75), 'time': 62.5, 'conflicting_drone': 'drone_1'
}, {'location': (13.0, 8.5), 'time': 65.0, 'conflicting_drone': 'drone_1'}, {'location': (13.0, 8.5), 'time': 65.0, 'conflicting_drone': 'drone_1'}, {'lo
cation': (13.5, 8.25), 'time': 67.5, 'conflicting_drone': 'drone_1'}, {'location': (14.0, 8.0), 'time': 70.0, 'conflicting_drone': 'drone_1'}, {'location
': (15.0, 7.5), 'time': 75.0, 'conflicting_drone': 'drone_2'}, {'location': (15.5, 7.25), 'time': 77.5, 'conflicting_drone': 'drone_2'}, {'location': (16
.0, 7.0), 'time': 80.0, 'conflicting_drone': 'drone_2'}, {'location': (16.5, 6.75), 'time': 82.5, 'conflicting_drone': 'drone_2'}, {'location': (17.0, 6.
5), 'time': 85.0, 'conflicting_drone': 'drone_2'}, {'location': (17.0, 6.5), 'time': 85.0, 'conflicting_drone': 'drone_2'}, {'location': (17.5, 6.25), 't
ime': 87.5, 'conflicting_drone': 'drone_2'}]}
Conflicts detected:
{'location': (8.0, 8.0), 'time': 40.0, 'conflicting_drone': 'drone_1'}
{'location': (8.5, 8.5), 'time': 42.5, 'conflicting_drone': 'drone_1'}
{'location': (9.0, 9.0), 'time': 45.0, 'conflicting_drone': 'drone_1'}
{'location': (9.0, 9.0), 'time': 45.0, 'conflicting_drone': 'drone_1'}
{'location': (9.5, 9.5), 'time': 47.5, 'conflicting_drone': 'drone_1'}
{'location': (10.0, 10.0), 'time': 50.0, 'conflicting_drone': 'drone_1'}
{'location': (10.5, 9.75), 'time': 52.5, 'conflicting_drone': 'drone_1'}
{'location': (11.0, 9.5), 'time': 55.0, 'conflicting_drone': 'drone_1'}
{'location': (11.0, 9.5), 'time': 55.0, 'conflicting_drone': 'drone_1'}
{'location': (11.5, 9.25), 'time': 57.5, 'conflicting_drone': 'drone_1'}
{'location': (12.0, 9.0), 'time': 60.0, 'conflicting_drone': 'drone_1'}
{'location': (12.5, 8.75), 'time': 62.5, 'conflicting_drone': 'drone_1'}
{'location': (13.0, 8.5), 'time': 65.0, 'conflicting_drone': 'drone_1'}
{'location': (13.0, 8.5), 'time': 65.0, 'conflicting_drone': 'drone_1'}
{'location': (13.5, 8.25), 'time': 67.5, 'conflicting_drone': 'drone_1'}
{'location': (14.0, 8.0), 'time': 70.0, 'conflicting_drone': 'drone_1'}
{'location': (15.0, 7.5), 'time': 75.0, 'conflicting_drone': 'drone_2'}
{'location': (15.5, 7.25), 'time': 77.5, 'conflicting_drone': 'drone_2'}
{'location': (16.0, 7.0), 'time': 80.0, 'conflicting_drone': 'drone_2'}
{'location': (16.5, 6.75), 'time': 82.5, 'conflicting_drone': 'drone_2'}
{'location': (17.0, 6.5), 'time': 85.0, 'conflicting_drone': 'drone_2'}
{'location': (17.0, 6.5), 'time': 85.0, 'conflicting_drone': 'drone_2'}
{'location': (17.5, 6.25), 'time': 87.5, 'conflicting_drone': 'drone_2'}
```

2D Conflict Visualization:
  Blue = Primary drone
  Green Dashed = Other drones
  Red X = Conflicts



Screenshot 3: 4D/3D Visualization
Z-axis = Time
Helps visualize temporal overlaps



The project runs successfully and detects spatial-temporal conflicts.
Visualizations clearly show the overlapping zones.
Code is modular and prepared for scaling or real-time extension.