Senior Design
ENG EC 463

Memo

To: Professor Pisano

From: Quentin Clark, Usman Jalil, Compton Bowman, Yafei Chen, Ahmet Caliskan

Team: 4

Date: 04/13/2024

Subject: Final Test Report

1.0 Prototype Test Results

We successfully demonstrated our project deliverables, including the completion of a physical drone platform and our full software suite. Because of safety concerns, we demonstrated our software stack, including web application, machine-learning enabled control algorithm, and delivery protocol, in simulation through AirSim.

2.0 Hardware

- Construction Completion Tests: We carefully examined all parts installed on the drone and tested if they were operating properly. The sensors including the camera and the GPS module could successfully send data to the computer. We also checked safety such as when we rotated propellers by hand to simulate the flying process, there were no components on the drone that would get in the way. The 3d printed harness for the Jeston and battery were secured and all wires fixed to the main body with zip ties, instead of hanging loose.
- Motor starts and output test: After connecting the drone to the battery, and turning on the safety switch and the FS-i6S remote control transmitter, when we adjusted the throttle up and down, the motors along with the propellers changed their speeds accordingly. The motors span smoothly and in the correct directions. For safety purposes, we clamped the drone to the table in the senior design lab to ensure the drone could not lift off during the test.

3.0 Software

To demonstrate our software stack, we set up two computers: a GPU workstation running AirSim, and a laptop connected to our website. Both connected to our DynaDB database we had configured before the test which was hosted on AWS. The workstation both A] runs the drone simulation, including physics simulation and rendering the virtual town environment and B] runs the script that would be run on the physical drone, using the AirSim APIs to get drone input/send drone output whereas on the physical drone we would be using the Mavlink protocol to talk to our drone controller.

We used the free Assetville Town Unreal engine environment [https://www.unrealengine.com/marketplace/en-US/product/assetsville-town] for our test. We set up our simulation starting location (in front of the police station) and ending destination to predefined locations (in front of the farm) [Fig1]. During the test, one of the testers asked us to set a new destination to ensure our destination was not unfairly favorable/pre-tuned, so we set a new impromptu destination per their request to in front of the warehouse [Fig1].

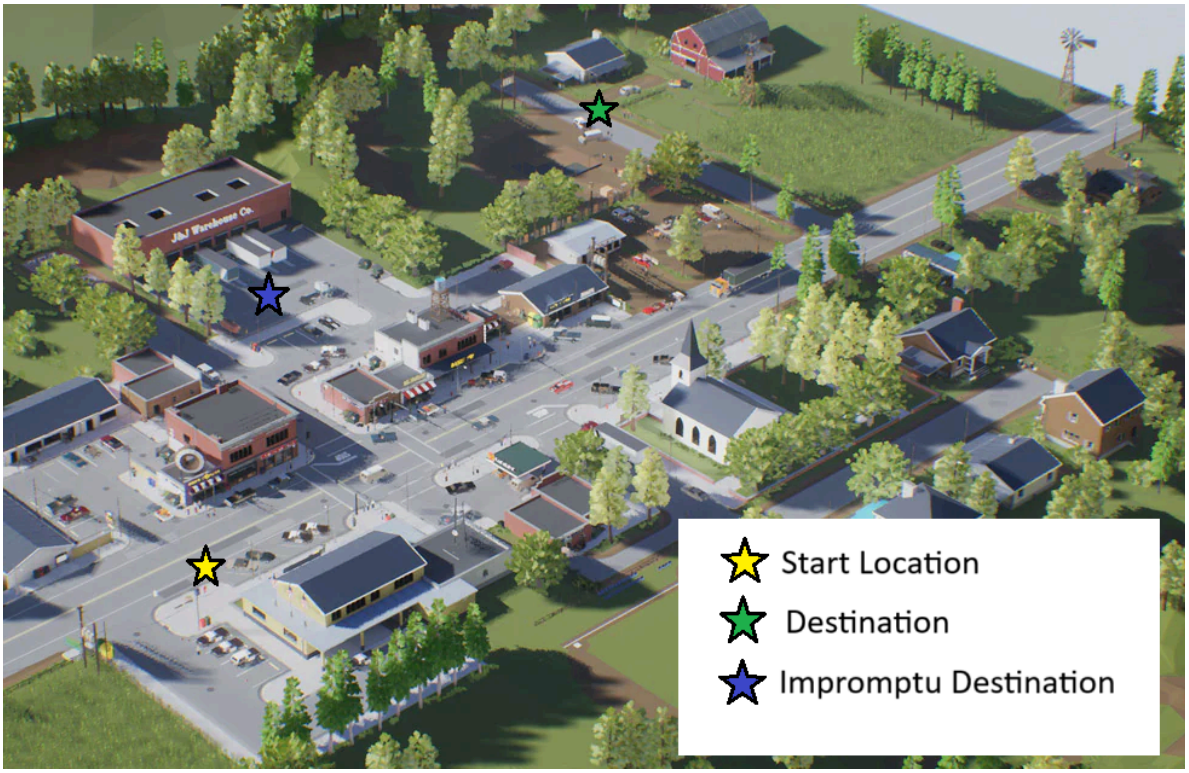

Figure 1 - Our simulation environment, marked with locations.

We initially experienced difficulty initializing flight from the website, but after fixing a variable that was incorrectly set for testing purposes, we were able to demonstrate drone flight take-off prompted by the website. It successfully reached its destination at the farm and flew back without crashing into obstacles, although it flew very close to a couple of trees. We were then prompted to attempt a new location, so we used a separate script to find the warehouse coordinates, re-initialized the flight, and it flew to the warehouse and back without difficulties. During the warehouse delivery, we also showed that the user could use the website to interrupt the drones' delivery by forcing it to land in place, and could then resume the delivery afterwords.

Lastly, we showed off our in-progress web application, by showing that the website could be refreshed to show up-to-date information being sent from the drone (currently GPS coordinates and control information, with video a non-functional work in progress)

4.0 Conclusion

In conclusion, we demonstrated that the core of our hardware and software deliverables were complete. We presented a completed prototype drone, with all the required hardware to be compatible with our software stack. We showed off our software stack by completing two successful drone deliveries in simulation without crashes or interruptions, demonstrating both the software for drone control (including our machine-learning based obstacle avoidance algorithm) and for the consumer experience, allowing the user to dispatch and monitor drone delivery progress.