sdmay19-35: Implementing a Web Portal System for Drone Simulation and Control

Week 6 Report

October 20 – October 26

Client: Ali Jannesari

Faculty Advisor: Ali Jannesari

Team Members

Bansho — *Test Engineer. Sensors Hardware Developer.*

Ian — *Scrum Master. Full Stack Developer.*

Li — *Test Engineer. Back-end Developer.*

Jawad — *Meeting Manager. Embedded Systems Developer.*

Mehul — *Project Lead. Computer Vision Developer.*

Sammy — *Report Manager. Lead Front-end Developer.*

Summary of Progress this Report

- Added support for additional commands in the terminal and mapped them to the appropriate drone movement method
    - The following commands, each of which take a single argument for distance to travel, are now supported:
        - moveForward
        - moveBackward
        - strafeLeft
        - strafeRight
        - turnLeft
        - turnRight
    - The issue in which the terminal commands would always use the default value was a small logical error that has been resolved. The command ascend 100 now goes 10 times as high as ascend 10, as expected.
- Made each propeller rotate independently.
    - In the animate function, iterate through all the propellers and apply a small rotation to the model about its Y axis.
    - Only apply the rotation to the Three.JS models, not the Cannon.JS bodies associated with them since that requires unnecessary computation.
    - However, even when taking steps to reduce the amount of computation power necessary, the independently rotating propellers cause lagging.
        - This is too early in the development phase to run into performance issues, perhaps reconsider alternatives like Gazebo.
        - Alternatively, don't move the propellers and instead replace them with a static propeller model that looks like it is rotating. However, other performance issues are likely to arise if we continue with our own simulator.

- Determined some potential choices for ROS compatible drones:
    - Gapter:
        - Costs approximately $1000
        - Gapter EDU provides comprehensive documentation and software packages which make development easy.
        - Has Linux operating system installed on the onboard computer
    - Crazyflie:
        - Costs approximately $250
        - Only 45 grams, too small.
    - Erle-Copter:
        - Costs approximately $580
        - Linux based operating system and uses ROS for various flight modes.
    - Decided on Gapter since it provides the functionality we need, extensive documentation is provided by Gapter EDU, and our client's can use it for a wider range of other purposes in the future.

Pending Issues

- Performance issues are occurring at an early stage in development, so an alternative must be researched and implemented.
- Still awaiting the camera for the Raspberry PI, cannot proceed with that component until it is received.
- All keyboard input is captured by the terminal when it should really only capture it when the terminal is focused.
- Minimizing/closing the terminal does not look good and should be improved.
- The joystick does not take into account the angle, which should be used in determining movement.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Bansho	Finished setting up the Raspberry PI and began experimenting with it.	7	38
Ian	Added additional terminal commands for drone movement.	6	42
Jawad	Researched drones and composed a list of pros and cons for each and sent to the client.	7	39
Li	Worked with Jawad to research drone options and create the list of pros and cons.	6	36

Mehul	Further researched computer vision and techniques for converting images to 3D models/environments.	6	37
Sammy	Added rotation to each propeller individually, began researching alternatives	7	43

Plans for Upcoming Reporting Period

- Frontend
  - Determine an alternative for our simulator and transition to that alternative.
  - Re-implement features such that they work with the new simulator.
  - Improve the movement and tweak the physics to be more believable.
  - Fix the terminal so it only accepts input when it is focused and allow it to minimize and close properly.
- Backend
  - Set up ROS and start interfacing with the front-end.
  - Start implementing computer vision techniques.
  - Establish communication between ROS and the Raspberry PI to mock the communication with the drone.
  - Coordinate with the client to place an order for the drone as well as any other necessary hardware.