Assignment #3:
Assigned: January 30th
Due: February 5th by midnight

Homework branch name: **Homework3**
5% of final grade

Objective:
To work with multiple drones and to develop a rudimentary collision avoidance system.

Overview:

Pull the new project 04_MultiDrones from our course repository. It contains the code you need for your homework assignment. We provide some initial functionality as a starting point that you can find in 04_multidrones under the multidrone2 directory. We also provide a five test cases in the form of ND1-5.json files. Each file includes the configuration information for several drones, their original home coordinates, and two waypoints representing flight origin and flight destination (in that order).

Your job is to build a **basic** collision avoidance system – which will likely be based on the use of altitude. For this assignment – normal flying altitude of all drones is 10 meters. They can *only* change altitude if they need to for purposes of collision avoidance. If they change altitude they should return to 10 meters once free of the potential for collision. You could select a different collision avoidance strategy (i.e., in place of altitude; however, this would likely take much more effort than is viable in a single homework assignment).

Format:

Required Functionality:

1. **Use the functionality in multidrone2 to:**
   a. MAKE SURE you know how to adjust the simulation speed as discussed in class (online lecture materials). You may increase the speed X2.
   b. Import the data from the .json file
   c. Instantiate a list of sitls, vehicles, and waypoints (routes). Please keep the names of these lists the same, as we will build upon this for our next assignment.

2. **Synchronize all the drones**
   a. All drones should be given instructions to takeoff to an altitude of 10 meters. Once all drones have reached that height they should be dispatched to their first waypoint as close to the same dispatch time as possible.
   b. As soon as an individual UAV reaches each waypoint, it should be assigned the next waypoint from its list. If no additional waypoints are available, the UAV should be issued a RTL command.
   c. Set the vehicle.groundspeed to 10 meters per second.
   Note: vehicle.groundspeed and vehicle.airspeed behave in a similar way if it isn't

windy – but in very windy weather, the UAV may not actually make much headway against the wind (or fly very fast with the wind) if you use vehicle.airspeed. Therefore groundspeed is preferable.

d. All drones should be assigned a simple_goto instruction to fly to its final waypoints. Note: You are not expected to use NED vectors in this assignment (unless you desperately want to!)

3. Collision Avoidance

Note: You need the ability to turn your collision avoidance algorithm ON or OFF.

a. You need to compute the distance between each pair of drones throughout their flight.

b. Select a minimum_separation_distance and frequently check for violations. You need to choose a sensible separation distance so that drones remain at the 10 meter altitude when possible, but without collisions.

c. If a violation is observed then take immediate collision avoidance steps.

d. Suggestion: Having a single minimum_separation_distance may be too simplistic. You could has a series of concentric rings (like an onion) and apply a set of layered strategies such as reducing speed if you come within certain distances of another drone. The final strategy is to change altitude. However, you need to synchronize this so that drones don't end up at the same altitude – especially if more than one drone is involved.

e. A collision is deemed to have occurred if the distance between the two drones is 4 meters or less (i.e., assuming a wingspan that requires this distance).

4. Tests

a. There are four test cases included in the 04_multidrones project entitled test1.json…..test5.json. Each test case will contain two or more UAVs and their routes. Each UAV will be assigned at least a home coordinate (state), and initial flight waypoint, and an ending waypoint. Some UAVs are assigned additional waypoints.

b. For each test case create a log showing the current coordinates of all drones at a single (or close to a single) timestamp.
UAV-0, UAV-1, UAV-2
1. (GPS Coords UAV-0), (GPS Coords UAV-1), (GPS Coords UAV-2)
2. (GPS Coords UAV-0), (GPS Coords UAV-1), (GPS Coords UAV-2)
Where each GPS Coords UAV-N are the (lat, lon, alt) of that UAV.

c. If the minimum_separation_distance is breached then insert a line in your log for each instance of a pairwise breach.
MINIMUM_SEPARATION_DISTANCE Breach: (GPS Coords UAV-1), (GPS Coords UAV-2)

d. If a collision occurs insert a similar row:
COLLISION: (GPS Coords UAV-1), (GPS Coords UAV-2)

e. Run all test cases with collision avoidance turned OFF. Generated log reports should be called: NoCA1.log, NoCA2.log,….NoCA5.log

f. Repeat with your collision avoidance activated. Generated log reports should be called: CA1.log…..CA5.log

5. Description: Describe your CA approach in 1 paragraph and discuss its success and failures. (YES!! It may not be entirely perfect.) Even if it works – would it still work if UAVs fly faster or if there are more UAVs. What do you think its limitations are.

Reading and Writing assignment:

Read the assigned reading (available under homework page and readings list on github.io
A Survey on Unmanned Aerial Vehicle Collision Avoidance Systems.

Also make sure to watch the second presentation which focuses on hazard analysis. In that presentation there are placeholders for you to review additional material on Automotive collision avoidance (CA), airplane (TCAS) CA, and human CA. After reviewing the material provided at those links and the 5 minute YouTube video please complete the following writing assignment that compares and contrasts these three collision avoidance systems with UAV CA. Use information from your reading on the "Survey of Unmanned Aerial Vehicle Collision Avoidance Systems" to inform your answers.

For each of the three CA systems (automotive, planes, people) describe the basic approach. Limit your description to about 3-4 sentences for each one.

Create a table that directly compares and contrasts these three approaches with CA for a UAV. Select 5 different factors to compare. Your table should be structured as follows – but make sure to name your factors!

Factors	Automotive Systems	Planes	People	UAVs
F1				
F2				
F3				
F4				
F5				

Grades:

This assignment is worth 5% of the final grade. Points will be assigned as follows:

5 points: **Complete functionality provided.** All tests run and catch collisions. (Note: We expect to see all logs, and will spot-check one or two test cases) Results are reported WITH and WITHOUT CA active. Commit all your code to Github under a Homework3 branch. Make sure your starting program is called main.py. Make sure all references are LOCAL!

1 point: **Description** (Step 5 above) What worked? What didn't work? What might you do differently if given more time.

4 points: **Reading and writing**: (See above).