CSE 321 Drone Collision Avoidance System (DCAS) Fall2015
Due date: 10/14/2017

Motivation:

Zero: Telecom Companies Turn To Drones For Help After Hurricanes, http://www.npr.org/2017/09/10/549922317/telecom-companies-turn-to-drones-for-help-after-hurricanes, last viewed Sept 14, 2017.

Trend1: First US drone delivery has taken place; See http://qz.com/458703/the-first-successful-drone-delivery-in-the-us-has-taken-place/, August 2015.

Trend 2*: **Project Octocopter**: "Amazon's secret R&D project aimed at delivering packages to your doorstep by "octocopter" mini-drones with a mere 30-minute delivery time", http://www.cbsnews.com/news/amazon-unveils-futuristic-plan-delivery-by-drone/, December 2013.*

Emerging trend 3: Project Wing*: Google has built and tested autonomous aerial vehicles, which it believes could be used for goods deliveries. The project is being developed at Google X, the company's clandestine tech research arm, which is also responsible for its self-driving car. Project Wing has been running for two years, but was a secret until now. Google said that its long-term goal was to develop drones that could be used for disaster relief by delivering aid to isolated areas.*

They could be used after earthquakes, floods, or extreme weather events, the company suggested, to take small items such as medicines or batteries to people in areas that conventional vehicles cannot reach. http://www.bbc.com/news/technology-28964260, August 2014.

Given this emerging scenario consider a concern in today's overcrowded skies with only the large commercial aircrafts. Federal guidelines require craft to be separated by at least 1,000 vertical feet and 3.5 lateral miles.

Incident 1: "Incident 2: "Drone slams into seating area at U.S. Open; teacher arrested"; http://www.cnn.com/2015/09/04/us/us-open-tennis-drone-arrest/, September 5, 2015.

Incident #2: "A commercial airliner narrowly missed colliding midair with a drone at 2,700 feet as it neared New York's LaGuardia Airport Friday morning, just hours after a string of incidents involving lasers pointed at planes, according to federal officials." http://www.foxnews.com/us/2015/05/29/ny-bound-pilot-swerves-to-avoid-collission-with-drone/, May 29, 2015.

Problem Statement: All aspiring "droners" (drone companies) are in discussion with Federal Aviation Authority (FAA) to set standards and policies to allow them to achieve their drone dream. But just imagine the issues and challenges in allowing drones in the sky that is already crowded. **Let us assume that** Federal Aviation Administration (FAA) wants to address these and such other problems in due to drones, planes, buildings and people, navigation of terrain, obstacle detection and avoidance, esp.in the context of many high tech companies entering the space with their own drone fleet. Understand most of today's aircraft's have explicit collision detection and alarm systems and there are strict rules about how close they can come to another flight.

One of the approaches we (CSE321ers) want to recommend is a combination of sensors on these drones, collision detection system and a **real-time system** that automatically controls the drone landing, take-off and cruising. **Your task is to design, implement and test this realtime software** we will call "**Drone Collision Avoidance System (DCAS)"**; assume that the sensors and collision detection systems are in place and we need to develop the software. Alternatively collision detection can be determined by some mutually agreed upon GPS/position based ***algorithm***.

Study of existing systems: Of course, for the larger crafts than the drones, there is a system called Traffic Collision Avoidance System (TCAS) that is based on sensors and instrumentation placed on each airplane in service. However we feel it is local solution centered on the aircraft and does not provide a global view of the whole picture for the drones since drones are operate on completely different "plane" and are remotely controlled; http://en.wikipedia.org/wiki/Traffic_collision_avoidance_system

Learning outcomes:

- Ability to apply your knowledge of problem solving, designing and implementing a collision avoidance system.
- Design using Finite State Machine (FSM) design tool.
- Learn to design using pthreads and C/C++ language.

Requirements:

1. DCAS is installed and operates at the drone control center (distributed all over the drone space) and monitors and controls the operation of the drones landing and taking of at these centers and if needed at their destinations.
2. Once the drone leaves the area /vicinity of a control center and reaches a cruising altitude the monitoring the handed-over to an intermediate DCAS station. So DCAS system is indeed a truly distributed realtime monitoring system. (This handover from one DCAS center to another is similar how you cell phone is handed over from one cell tower to another.)
3. You need to worry about only one DCAS center. It is possible that the model of operation for the DCAS is that it totally controls its assigned fleet. The various activities taken by a drone include:
    a. takeoff
    b. cruise
    c. deliver
    d. return
    e. avoid collision (navigate obstacles).
4. Regular flights have runways for takeoff and landing. We are assuming vertical takeoff and landing for the drones. However only one drone can take off from a center/station at any time. Same with landing. Here is the basic interpretation:
    a. A drone is assigned the task
    b. The drone picks up the payload
    c. The drone requests clearance to take off
    d. Once clearance is given the drone takes off.

5. DCAS software also keeps track of
    a. number of drones,
    b. whether they are active and
    c. their state.
6. When a drone requests landing, if a landing space is available it is allocated. If not, the drone keeps hovering around in a holding pattern. All the drones requesting landing are queued up and are served first come first serve basis.
7. Collision detection instrumentation (sensors, alarms etc.) will detect imminent mid-collision, run-way collision, and any collision within the DCAS station when the drones are moving around. Appropriate action to be taken when collision detection happens.
8. Life cycle of a drone:
    a. DCAS registers an active drone, it is at gate, gets clearance to take-off gets clearance for takeoff, takes-off, reaches cruising altitude, delivers payload and returns. Exceptional situations are collision detection and the corrective operations that may happen at various points.
9. Drones operate concurrently. So you need simulate then using concurrent control structures such as threads (pthreads, for example).
10. Assume that the drones move in a x-y coordinate system that has it (0,0) at the top left corner of your drone zone of a DCAS station.
11. Assume that path planned for a drone comprises of movements along the x-axis and y-axis, not diagonal. That is, it is a square grid with 4 directional movements. Use Manhattan distance heuristics.
12. Assume that drones may change speed (eg. Move in 2-cell distance per move instead of 1 cell) at any time.
13. Assume other conditions and make sure you document your assumptions.
14. Design this realtime system, implement it and test it with a system handling 10 drones and 1 station.

Design:

1. Start by designing the state diagram representing the life cycle of a drone. Also use a table-driven scheduling for the flights monitored by a DCAS. You will have to design these two before you can start implementing the code. You may have to revisit and revise your state diagram and the state transition table based on the needs.
2. Implement the DCAS system using C/C++ language. Define a function for each of the operations. Assume collision detection as an asynchronous event. (How will you simulate an interrupt and handle it?)
3. Drones operate concurrently. So you need simulate then using concurrent control structures such as threads (pthreads, for example).
4. Test your system for various conditions using sample scripts.
5. For implementation of the operation you may print out the message and allow some predetermined elapsed time for the operation taking place.
6. On completion your program should run DCAS application; should take sample scenarios as input and play out the scenarios and action taking place through a series of messages printed out.
7. You need a mechanism for realizing /implementing parallelism, mutual exclusion (collision avoidance and shared data), communication between the drone and the controlling station, and others.

8. Your output should be display as a map where you can visually see the drones moving around. A simple active ascii map printouts of where drones exist rather than positional changes in x,y coordinates may be intuitive.
9. Use an appropriate directory structure (src, compile etc). Use the make utility for building your executable.

What to submit? Please submit your design documents, well commented code online and clear simulation run outputs showing various conditions: submit cse-321 your files

You will have to demo your project to the TAs to get your grade.