Author: Shabaz Farooq

DECOY (Drone Control and Deployment) for Social Good

Description

The use of drones are becoming increasingly widespread in the 21st century. What was once a tool reserved for carrying out military missions is now tapping into every facet of our lives. Examples can be demonstrated with recent news about Amazon's drone delivery system. Essentially, it has become clear the technology used in the development of Drones is a critical tool for developers.

The DECOY project has been developed in C and its purpose is to analyze and assess critical scenarios in a life cycle of a drone using a 2D representation. The foundation of DECOY can be further developed to fit almost any scenario. Whether it be a mail delivery system or a military drone being used to analyze and report data. Specifically, DECOY demonstrates drone movements using advanced algorithms to detect collisions of other drones, or un-anticipated obstacles. Furthermore, DECOY's framework is set such that each drone is in constant communication with a base system. This is particularly useful to report and measure data.

Implementation and Drone life cycle

- DECOY begins by prompting a user for the number of drones to be used.
- A *grid system* is implemented by the use of a multidimensional array.
- The grid system is then initialized by storing a predefined 'empty spot character' within each cell of the array.
- *Home base*, *drop off points* and arbitrary *obstacles* are then stored using unique characters (that are recognized by the drone collision avoidance algorithm) in the array.
- The number of drones entered by the user are then initialized as individual drone threads.
- The life cycle of a drone begins by picking a *random* drop off point.
- It then generates a *safe move:*

    - A drone is allowed eight directional moves; the four diagonal moves and left, right, up, down.

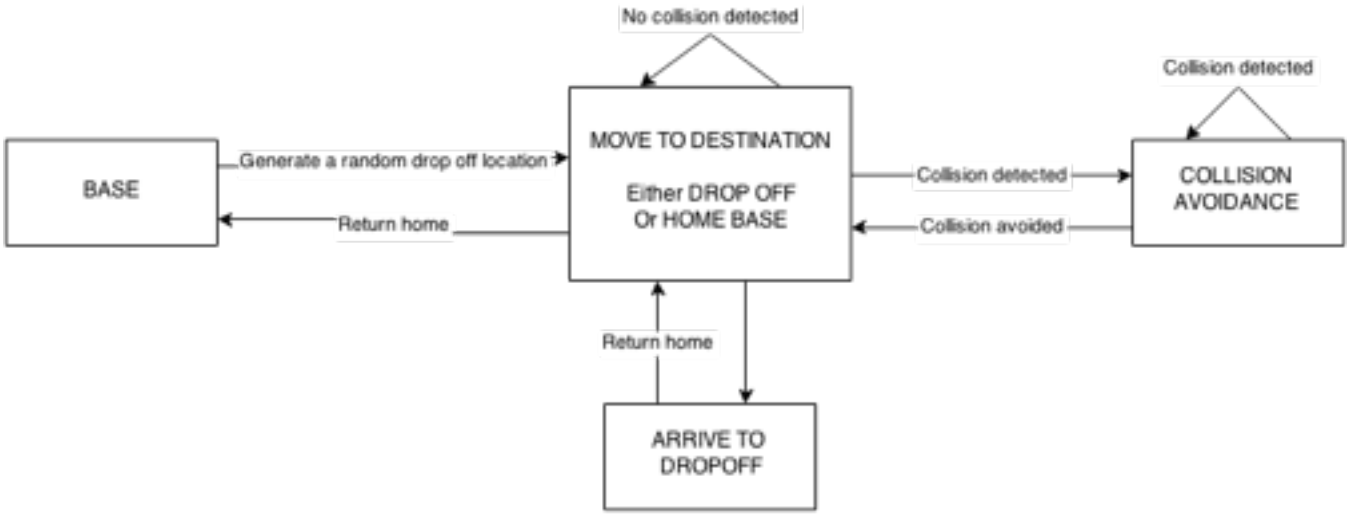
- It checks of the eight moves which is going to bring it closer to it's destination. More specifically it checks which move is not only going to bring it closer, but which move will bring it to its destination quicker.

- It then checks to see if the direction it wants to travel in is occupied by any *HAZARD;* such as another drone or an obstacle.

- If it is clear, it proceeds to the move. However, if it is not the collision detection algorithm attempts the next move which will bring it closer to it's destination. If for some reason a drone cannot move closer to it's destination it makes an arbitrary *safe move* to give way to another drone that may need to occupy the air space.

- Once a drone arrives to its destination it conveys a message to the DECOY system, registering it's arrival.
- It proceeds back to the home base.
- Before any and every move a drone checks with the DECOY *safe move* algorithm to ensure no collision can occur.
- Following every successful move a drone is put to SLEEP to allow other drones to communicate with the DECOY system.

State Diagram: Representing the lifecycle of a drone



Using the DECOY 2D system

```
dhcp205-072-074:lab1_src Shabaz$ ./Project1

Enter the number of drones you would like: 4
```

Users are promoted to enter the number of drones they would like

Drones travel to their destination. You can easily keep track of drones as they are represented with unique colors. Information such as collision avoidance re-route paths can be found in the upper left hand corner. And the drone count can easily be seen above the grid.