Aris Emery
CS5700
HW1 Report

My design:

The first design decisions that I had to make in my mind revolved around the messages. I knew that I had to separate them somehow, but I wasn't sure how. I decided to make a parent message class, and have all other messages be children of that class. The parent class would handle things like encoding that all messages must do, and the individual classes would handle differences between messages, such as degrees or centimeters for the second argument. For communication, I made the UDP communicator class to encapsulate all communications. The flier would access this class to send messages and receive messages from the drone. The messages that the flier sends came from pre-set missions, made with the missions class. The missions class lists string of commands, and the flier takes these commands, turns them into the right messages, and sends these messages to the UDP communicator to be sent to the drone. I have also included a main class, but its functionality is more trivial. This is where the flier is instantiated, and where the console based UI is located. For the strategy pattern, I decided to apply it to the message class. Every time the UDP communicator send a message, it first checks to see if the message is valid by calling message.isValid(). This will activate one of three is valid operations, depending on if there is just a string or a second argument, and if that argument is in degrees or centimeters. For the more complex commands, isValid() will check to make sure any command with centimeters is between 1 and 500, and any command with degrees is between 1 and 360.

Insights uncovered:

The main insight I uncovered was the usefulness of the strategy pattern. Initially I had no idea where I would include, but once I used it for the isValid() function, I couldn't imagine trying to do it any other way. Another key insight I had was the importance of designing your program to be testable from the start. This is something that I need to work on, as thinking of ways to test functions proved to be difficult at times.

UML diagram: