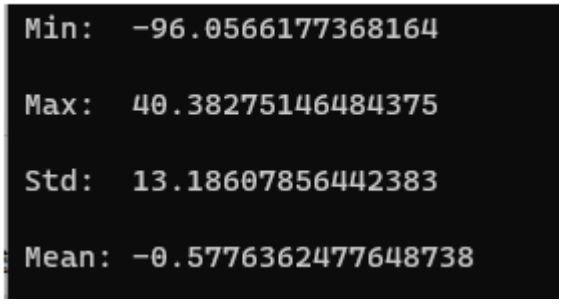
Deliverable 3

1. Unable to execute the test function. However, the highest success rate throughout the training is 51%.

   By executing the test and conversion script with the checkpoints from the training result file, the mean is -0.577, std: 13.18, min: -96.056, max: 40.382



   The code segment for this min/max is the following:

```python
## -- the master find min/max function, finding min/max ranges over all defined layers in num_it -- ##
def find_min_max(layers,num_it):
    vals = np.array([])
    mins = np.array([])
    maxs = np.array([])

    for i in range(num_it):
        print("it ",i,"\n")
        for layer in layers:
            vals = np.append(vals,run_pb(layer))
        mins = np.append(mins,np.min(vals))
        maxs = np.append(maxs,np.max(vals))

    print('Min: ',np.min(vals),'\n')
    print('Max: ',np.max(vals),'\n')
    print('Std: ',np.std(vals),'\n')
    print('Mean:',np.mean(vals),'\n')
    if (plot_min_max==True):
        ax = plt.gca()
        ax.grid()
        ax.plot(range(len(mins)),mins)
        ax.plot(range(len(maxs)),maxs)
        plt.title('Min/max over num it')
        plt.xlabel('Num it')
        plt.ylabel('Min/max')
        plt.legend(['Min','Max'])
        ax.set_axisbelow(True)
        plt.show()

    return np.min(vals),np.max(vals)
```

   The num_it variable is pre-defined to 1000 as the total amount of iterations it will perform. Unfortunately, there is no description nor comments on what matrices they are using but based on the highest success rate, I would say it has an Okay performance. If there are more time to train it such as a month or more, it may be better.

For the application, I can't find a way to make a new game using the weights it trained so I just leave it in the unreal engine and modified its environment which makes the map a little different than the one it was trained in by following the tutorial under its GitHub repository. I have modified all scripts that is needed for testing so that it won't restart training when you try to run the script.

The script they given isn't clear on where the drone is heading and how many collisions are allowed so I am going to briefly explain it here:

By my observations of its scripts and testing, the script would randomize the environment and destination coordinates each time it reset the gym so that the drone won't fly towards the same place all the time. If the drone stays in one position or stuck in a loop of actions such spinning around at a fixed height, it will reset the gym as well. Every time it resets the gym, you must click "no" on a popped window to choose multirotor mode.

However, sometime, the drone won't be able to reach the destination due to unknown reasons and it will start to do undefined behaviour which leads to a reset of the gym. My guess is it hasn't been well trained yet.

2. Here is the whole project. The way to run the test script is going to ".\autonomous_drone_rl\airlearning\airlearning-rl\run_time" and run the python file collect_data.py through command prompt. The version of python I used is python 3.7. You would also be asked to install all kinds of plugins such as tensorflow etc. I also recorded a video during testing so that you don't need to run everything yourself.
   Here is the google drive link to the zipped project file: https://drive.google.com/file/d/1gg2DvKkoCF6N3iJxkz9Jdp1aoFk62EZY/view?usp=sharing
   The recorded video is also in here.

   Meanwhile, if you want to try the model, I have uploaded the weights file into GitHub. It is named dqn_level_3_.hf5
   You can follow the tutorial here to get the unreal engine project and algorithms or code it

with other things. https://github.com/harvard-edge/AirLearning