Britny Mors

ME 493 – Autonomy

4 – 27 – 17

LR\_1: The program represents a ship using a class called “ship”. This is declared in line 48. Each ship is characterized by its start location, current location, omega value, theta value, dt, v, T, and a fitness value.

LR\_2: The goal for this program was classified using a class called “goal” it contains information of the goal location and the midpoint location of the goal.

LR\_3: This program does not exit the simulation loop if the boat is out of bounds. However, a very large positive fitness (fitness in this program is trying to be minimized) is continually added while the boat is out of range. This theoretically should cause the program to not select the policy that drives the boat out of range. This does however significantly slow down the simulation.

LR\_4: This program initializes a neural network at line 255.

LR\_5: The program can mutate a set of weights using the mutate function written for the policy class. The function takes in a mutation operator in this case a value of 0.2 and multiplies it by two random numbers between 0 and 1. These numbers are then subtracted from one another and this is added to a randomly selected weight to mutate the weight function. This is repeated a random number of times determined before the mutation loop.

LR\_6: The input and output boundaries are set in lines 258 through 262. The neural network is executed inside the evaluate function. The input limits for the x and y input were 0 to 100. The input limit for the theta value was between 0 and 6.28. This puts theta between 0 and 360 degrees. The output limit was set from -15 to 15.

LR\_7: The output of the neural network is saved into a variable called u and is used to update the ships position.

LR\_8: The program can calculate x(t+1) in line 92 inside of the updatepos function of the ship. The function takes in the new u value and saves the new x location at in the tx variable which is used in line 120 to overwrite the boatx position.

MR\_1:



