CS 440 MP1
Stephanie Lona

Part 2: Experimentation

1. In WorldWithoutThief
    a. Reward starts negative for the first episode then goes to 0.0. As episodes progress it stays at 0.0. When running the Policy Simulator it is clear that the robot goes up into the corner and stays there.
    b. When the epsilon is increased to .1, the first episodes are fluctuating between negative numbers and then it increasingly becomes more positive. There are occasional negative points and low points, but overall it increases, but not much over 200. This is because when randomness is thrown into the algorithm, the ability to choose different paths becomes greater. When there is a little bit of randomness, the agent takes more chances in early episodes to learn what is better and update the Q-Matrix for future episodes.
    c. When increasing the epsilon to something much larger, 0.5, the reward is consistently negative, rarely getting above 0. This is because the higher epsilon introduces too much randomness into the learning algorithm to the point where half the choices are completely random.
2. In WorldWithThief
    a. Using a low epsilon, and not knowing the thief makes the performance of my agent poor. The reward rarely goes above 0 and it is consistently lower than -10. This is because if the agent does not know where the thief is, it can’t adjust the Q-Matrix accordingly.
    b. Keeping the parameters the same and changing the knowledge of thief to yes the agent gets very good results. The early episodes start off negative, but by episode 5 it becomes positive. It is continually increasing until it hits around 285, then it fluctuates around there.
    c. To determine the best rate and epsilon, I first began by keeping the rate constant and varying epsilon. Below are some parameters I measured

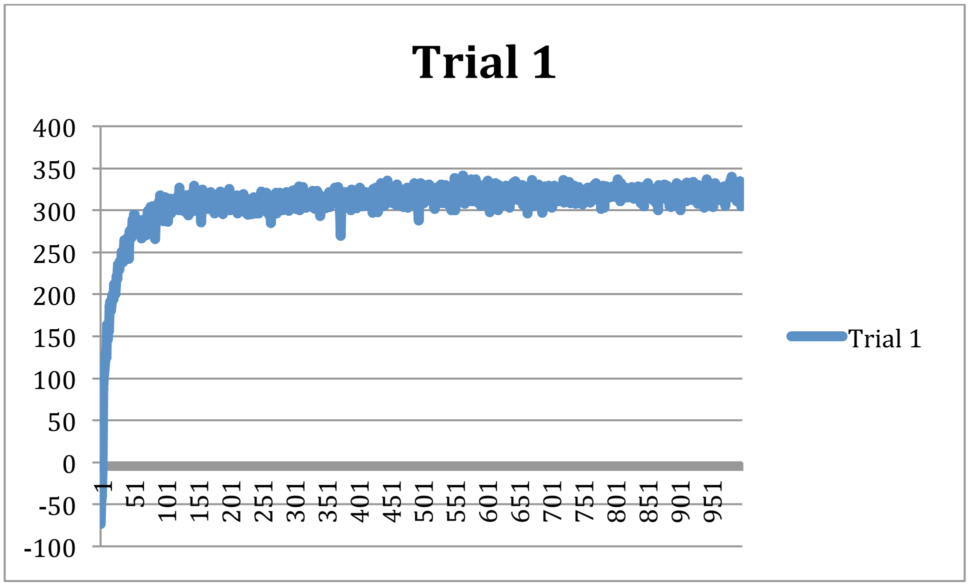
Epsilon	Approx. Highest Reward	Episode to get to Avg
0	175	25
.02	330	30
.03	315	50
.04	311	50
.045	300	50
.05	303	25
.06	280	40
.1	220	25

I then kept Epsilon at .02 and varied the learning rate. Below are my findings

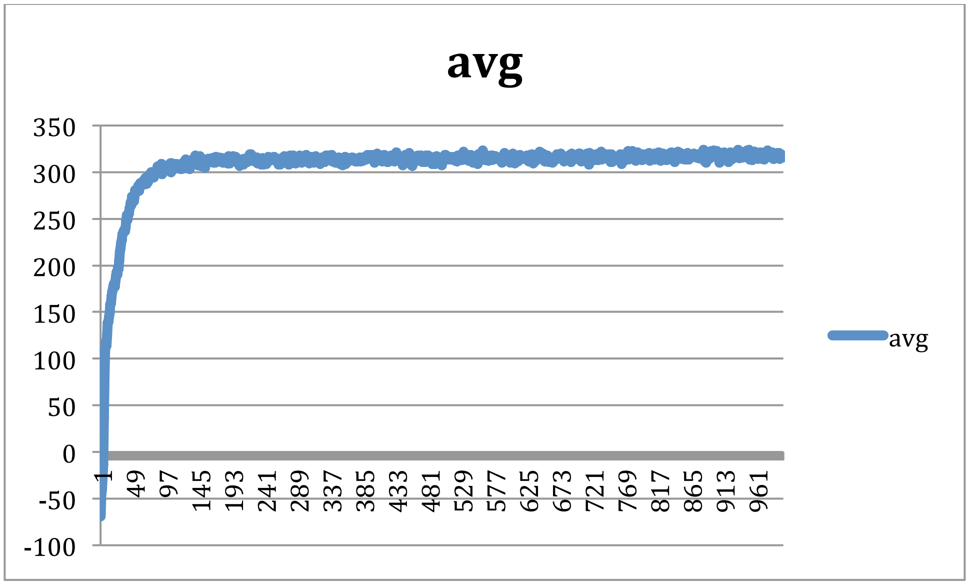
Epsilon	Approx. Highest Reward	Episode to get to Avg
.1	330	30

.15	340	25
.2	325	30
.3	315	40

3. For this I used epsilon = .02 and rate = .15. Below is the result for the first trial.



I then ran the program nine more times. Below is the graph of the average of all trials



In both graphs you notice that as episodes progress, the agent gets a better reward. Some significant differences between the average and the single trial are that the average is a much cleaner line. The average also hits its peak sooner. This is because when the data from 10 trials is put together there is less discrepancy from one point to the next.

The algorithm using the parameters I selected learns to get a bigger reward at a rate of around 50 and then the reward is consistently over 300!