

Reinforcement Learning

Project Report

Class and Section: CPE471.1001

Name: Lucas Pinto

Due Date: February 12, 2025

GitHub Repository: <https://github.com/PieFlavr/CPE471-Project-1>

World-Agent Design

The default environment is modeled by a 5x5 “Grid World”, with an “Agent” that can do ‘up’, ‘down’, ‘right’ and ‘left’ actions. Any invalid actions such as attempting to move off the grid can be attempted by the Agent, but they do not let the Agent out. The Agent and Grid World are implemented in the `agent.py` and `grid_world.py` file respectively.

Agent Rewards

The Agent is rewarded if it reaches the goal of the Grid World, which by default is set on the bottom right of the Grid World. The agent is heavily punished for attempting to make an invalid move, and is only lightly punished for every step it takes. This was done in an effort to make the Agent attempt to follow the least number of steps to reach the goal.

By default, the reward for reaching the goal is just the area of the grid, (`grid_length*grid_width`). The punishments are always (-1) for every step and (-5) for attempting to move out of bounds.

From testing higher grid sizes such as 100x100 Grid Worlds, this reward structure begins to break down as the Agent loses motivation to avoid invalid moves. Additionally, the Agent seems to create additional loops during its path even during the final action sequence. It is likely due to the relative scale of the goal reward to the punishments breaking down as the grid gets larger, but increasing the training episodes accordingly such as to 100k amend the issue.

Learning Algorithms

There are only two learning algorithms currently implemented in this project, the ordinary Q-Learning algorithm and the $Q(\lambda)$ algorithm. While it was possible to attach the Q-Tables to the Agent (to demonstrate policy), for the scope it was easier to leave it as a variable per training run. As for the eligibility traces of $Q(\lambda)$, they were implemented as a variable per-episode. While the algorithms were implemented in such a way that the action selection algorithm could be interchangeable, the only one implemented was Epsilon-Greedy Selection. For the most part, all of these were implemented in the `learning.py` file.

Algorithmically, they essentially worked as follows...

Q-LEARNING UPDATE

```
Initialize  $Q(s,a)$  arbitrarily
Repeat (for each episode):
..> Initialize  $s$ 
..> Repeat (for each step of episode):
..> ..> Choose  $a$  from  $s$  using policy derived from  $Q$ 
..> ..> Take action  $a$ , observe  $r, s'$ 
..> ..>  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$ 
..> ..>  $s \leftarrow s'$ 
..> until  $s$  is terminal
```

$Q(\lambda)$ UPDATE

```
Initialize  $Q(s,a)$  arbitrarily and  $e(s,a) = 0$ , for all  $s,a$ 
Repeat (for each episode):
..> Initialize  $s,a$ 
..> Repeat (for each step of episode):
..> ..> Take action  $a$ , observe  $r,s'$ 
..> ..> Choose  $a'$  from  $s'$  using policy derived from  $Q$ 
..> ..>  $a^* \leftarrow \arg \max_b Q(s',a')$  (if  $a'$  ties for max,  $a^* \leftarrow a'$ )
..> ..>  $\delta \leftarrow r + \gamma Q(s',a') - Q(s,a)$ 
..> ..>  $e(s,a) \leftarrow e(s,a) + \delta$ 
..> ..> For all  $s,a$ :
..> ..> ..>  $Q(s,a) \leftarrow Q(s,a) + \alpha \lambda e(s,a)$ 
..> ..> ..> If  $a' = a^*$ , then  $e(s,a) \leftarrow \gamma e(s,a)$ 
..> ..> ..> else  $e(s,a) \leftarrow 0$ 
..> ..>  $s \leftarrow s'$  ;  $a \leftarrow a'$ 
..> until  $s$  is terminal
```

Report Data Parameters

For this report, the training data for the 5x5 Grid World was produced with the following parameters:

	Q-LEARNING	Q(λ)
<i>Update Parameters</i>	$\alpha = 0.15, \gamma = 0.95,$ $\epsilon = 0.1$	$\alpha = 0.15, \gamma = 0.95,$ $\epsilon = 0.1, \lambda = 0.5$
<i>Episodes</i>	1000	1000
<i>Agent Start Position</i>	(0,0)	(0,0)
<i>Goal Position</i>	(4,4)	(4,4)
<i>Goal Reward</i>	25	25
<i>Invalid Move Reward</i>	-5	-5
<i>Step Reward</i>	-1	-1

Additionally, the training data for the 10x10 Grid World was produced with the same parameters except as follows:

	Q-LEARNING	Q(λ)
<i>Goal Position</i>	(9,9)	(9,9)
<i>Goal Reward</i>	25	25

With these parameters, it should take no less than a few seconds to generate the data required for the program to make its figures. Although, tests were performed with much higher grid sizes and episode counts. They worked although the time obviously grew with the increase of either. Anything above 10k episodes and a 100x100 grid size will begin to take more than a minute to generate everything, especially the animation of the action sequences. Additionally, although now mitigated, during programming possible issues with memory did occur so be warned of trying extremely large values.

RESULTS

The results of the above runs are shown below, but more extensively additional images and .csv data were recorded in the `project_data` folder. Furthermore, the following demo videos (5x5 and 10x10) contained example runs (not the same as analyzed data)...

- 5x5 DEMO: <https://youtu.be/a9IlX-IIqPI>
- 10x10 DEMO: <https://youtu.be/QAEhNdgtYeM>

While this report talks of the parameters of the 10x10, the figures and results are not discussed here. More detail, figures, and data are stored in the GitHub project repository: <https://github.com/PieFlavr/CPE471-Project-1>

5x5 Q-Learning + $Q(\lambda)$ Results

With regards to the Q-Tables, both seem to evolve as expected. The steps taken per episode seem to converge around 8 steps, which is the expected behavior. With regards to converging to that number, it seems $Q(\lambda)$ is faster with the onset of the behavior than Q-Learning, but appears to have a higher variation in efficiency. This is likely due to *Epsilon-Greedy Selection* as it either selects entirely randomly or exactly the best, which the *Eligibility Traces* in $Q(\lambda)$'s implementation might encourage otherwise bad behavior. You can see this in the action sequences as well, where Q-Learning has initially an almost completely random behavior, $Q(\lambda)$ seems to almost prefer really long “strides”. Considering the dimensionality of the Grid World being, well, a grid where diagonal movement practically doesn't exist, this is somewhat expected behavior.

5x5 Q-Learning Figures

Q-Tables

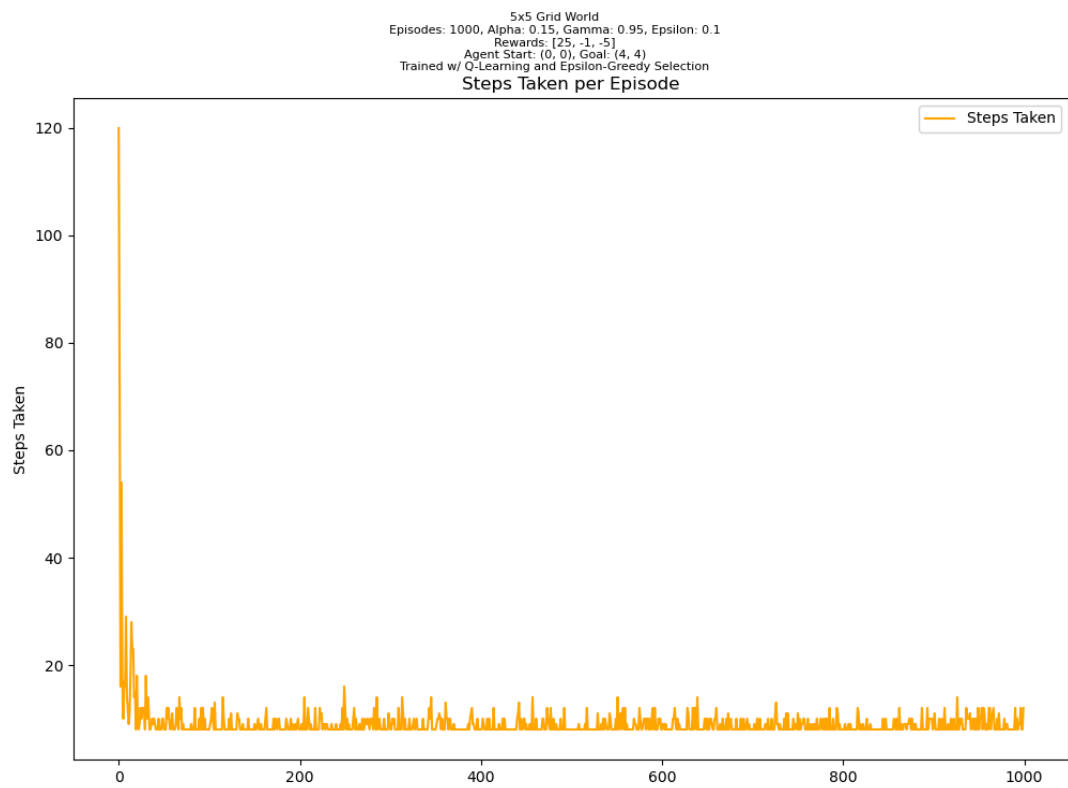
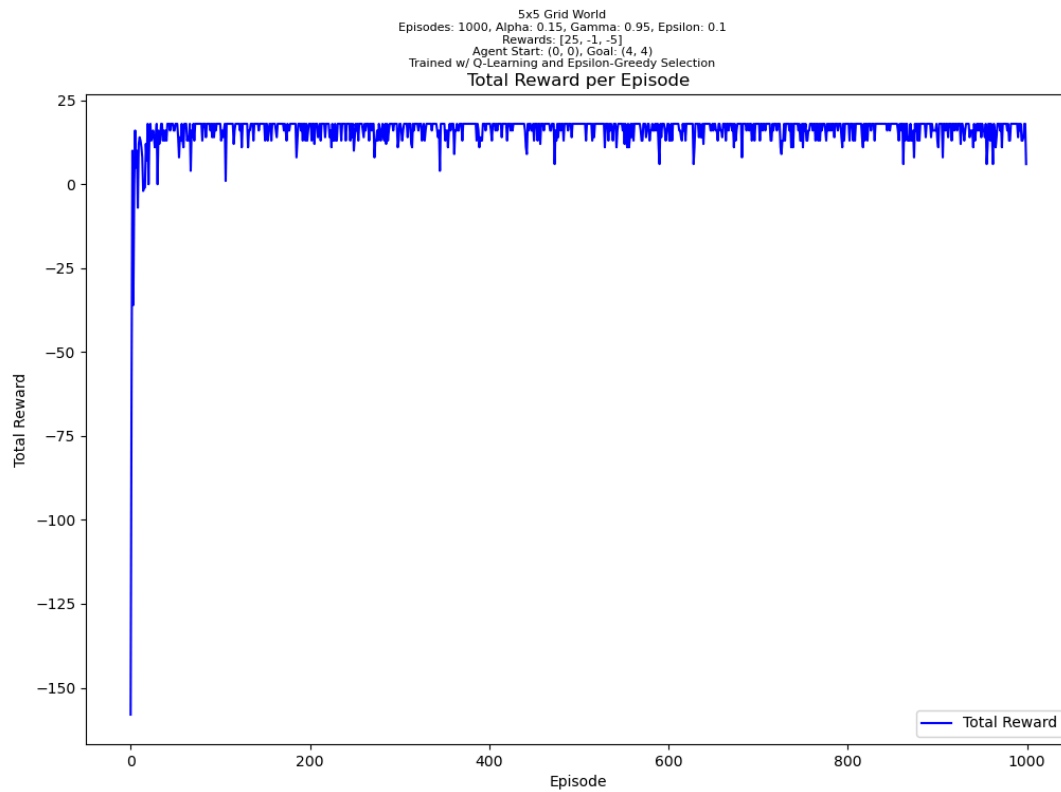
5x5 Grid World
Episodes: 1000, Alpha: 0.15, Gamma: 0.95, Epsilon: 0.1
Rewards: [25, -1, -5]
Agent Start: (0, 0), Goal: (4, 4)
Trained w/ Q-Learning and Epsilon-Greedy Selection
First Q-table

State(x,y)	up	down	left	right
(0,0)	-0.75	-0.2774999999999997	-1.408875	-0.2774999999999997
(0,1)	-0.298875	-0.298875	-0.75	-0.171375
(0,2)	-0.298875	-0.298875	-0.75	-0.15
(0,3)	-0.298875	-0.298875	-0.75	-0.298875
(0,4)	-0.298875	-0.75	-0.75	-0.15
(1,0)	-0.75	-0.2774999999999997	-0.171375	-0.2774999999999997
(1,1)	-0.298875	-0.15	-0.15	-0.15
(1,2)	-0.15	-0.2774999999999997	-0.15	-0.15
(1,3)	-0.2774999999999997	-0.2774999999999997	-0.298875	-0.2774999999999997
(1,4)	-0.298875	-0.75	-0.15	-0.15
(2,0)	-0.75	-0.40725	-0.171375	-0.15
(2,1)	-0.2774999999999997	-0.298875	-0.15	-0.15
(2,2)	-0.15	-0.15	-0.15	-0.298875
(2,3)	-0.15	-0.15	-0.3170437499999996	-0.15
(2,4)	-0.15	-0.75	-0.171375	0.0
(3,0)	-0.75	-0.2774999999999997	-0.171375	0.0
(3,1)	-0.2774999999999997	-0.298875	-0.171375	-0.15
(3,2)	-0.298875	-0.2774999999999997	-0.15	-0.15
(3,3)	-0.15	-0.15	-0.15	-0.15
(3,4)	-0.15	0.0	0.0	0.0
(4,0)	-0.75	-0.15	0.0	0.0
(4,1)	-0.15	-0.2774999999999997	-0.171375	-0.75
(4,2)	-0.2774999999999997	-0.15	-0.15	-0.75
(4,3)	-0.15	3.75	0.0	0.0
(4,4)	0.0	0.0	0.0	0.0

5x5 Grid World
Episodes: 1000, Alpha: 0.15, Gamma: 0.95, Epsilon: 0.1
Rewards: [25, -1, -5]
Agent Start: (0, 0), Goal: (4, 4)
Trained w/ Q-Learning and Epsilon-Greedy Selection
Last Q-table

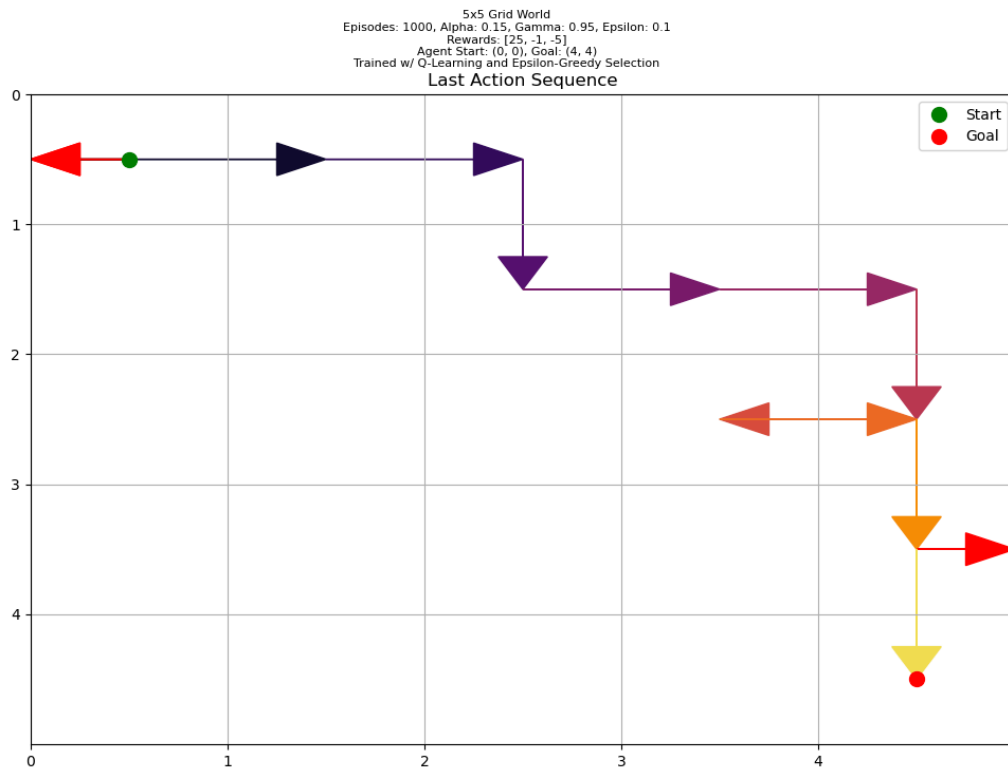
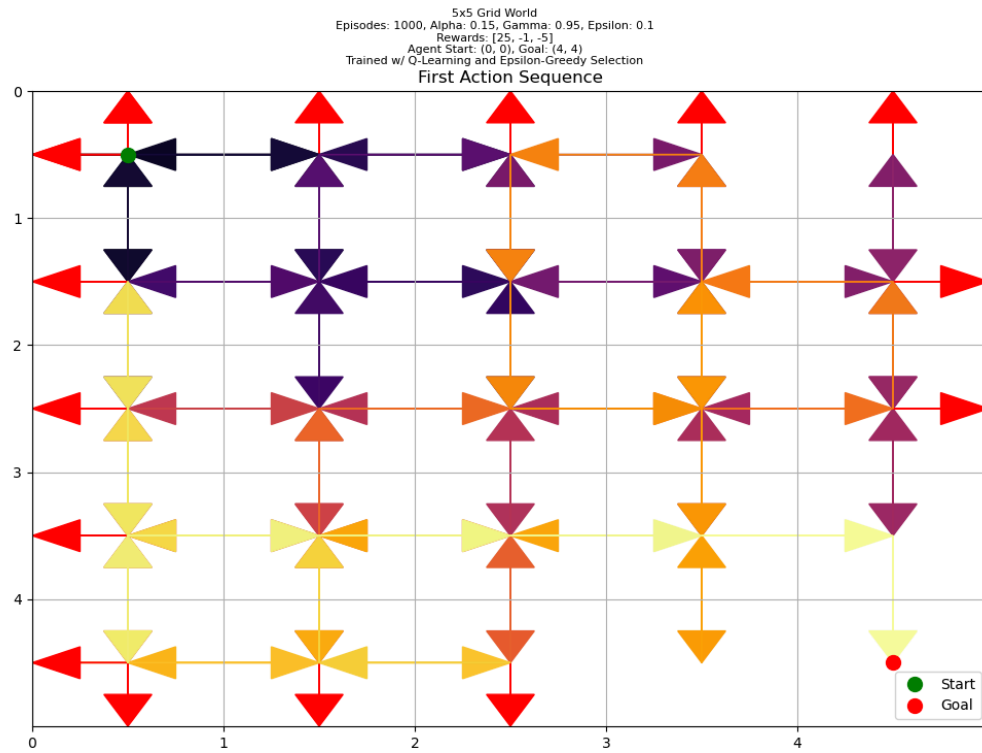
State(x,y)	up	down	left	right
(0,0)	5.167304735072013	9.468005948814008	5.688221145876282	11.425178324218688
(0,1)	-2.399793424993899	-1.8811116435819226	-2.205474329988281	12.433535549561146
(0,2)	-1.5998200890166316	1.7926378579027769	-2.190261675357422	-1.4062869266275018
(0,3)	-1.1916975690262437	-1.091412049339365	-1.494375	11.562836584754368
(0,4)	-0.9525807092221901	-1.494375	-1.494375	-0.8186239891406251
(1,0)	7.239655979825174	9.296596604281541	9.478980014983673	13.07913507812494
(1,1)	-1.6450623130660225	14.770650419074698	-1.7028651006366675	3.075761740099489
(1,2)	-0.8635097102697891	16.63769450520928	-1.2959680896296812	4.964919752296445
(1,3)	4.070431562730196	-0.7694018574829101	1.9314092883903098	18.578846089306083
(1,4)	-0.6086112984375	-1.4595890578125	-0.5932836867187499	1.147594058397029
(2,0)	8.781640959660356	14.820142187499943	11.227177433533525	13.93286235179352
(2,1)	10.686326186716729	14.471242468726384	12.021282168817494	16.652781249999947
(2,2)	-0.7865896524761635	-0.6660412237452393	2.995138140886659	18.300530678169473
(2,3)	4.673182410676839	2.07077334408211	1.0145942727980164	20.6121130736506
(2,4)	-0.298875	-0.75	-0.33825843749999995	15.619013841936445
(3,0)	0.346659698344969	16.616139510961215	-0.9906963424392091	2.145828228478925
(3,1)	12.5344328809058	18.339243332459343	14.67326629360675	18.581874999999958
(3,2)	4.178602638994793	5.285178466404595	1.7860905486251717	20.612471817253777
(3,3)	2.4563478981134734	22.74999755095827	2.7201801009648823	-0.15
(3,4)	0.6453001339830322	8.195687798381407	3.05455177641935	24.999999915236067
(4,0)	-0.75	18.185378649148586	-0.8226234120359734	0.05357319616497569
(4,1)	14.559927559549873	20.612499999999997	16.158496859942403	12.401042159049181
(4,2)	18.158842636572167	22.749999999999998	18.300974206059244	16.35643783343442
(4,3)	19.641131936820244	24.999999999999999	20.37318835767585	18.58160755386521
(4,4)	0.0	0.0	0.0	0.0

Reward and Step Plots



Action Sequences

(Darker = Earlier Actions ; Brighter = Later Actions; Red = Invalid Steps)



5x5 Q(λ) Figures

Q-Tables

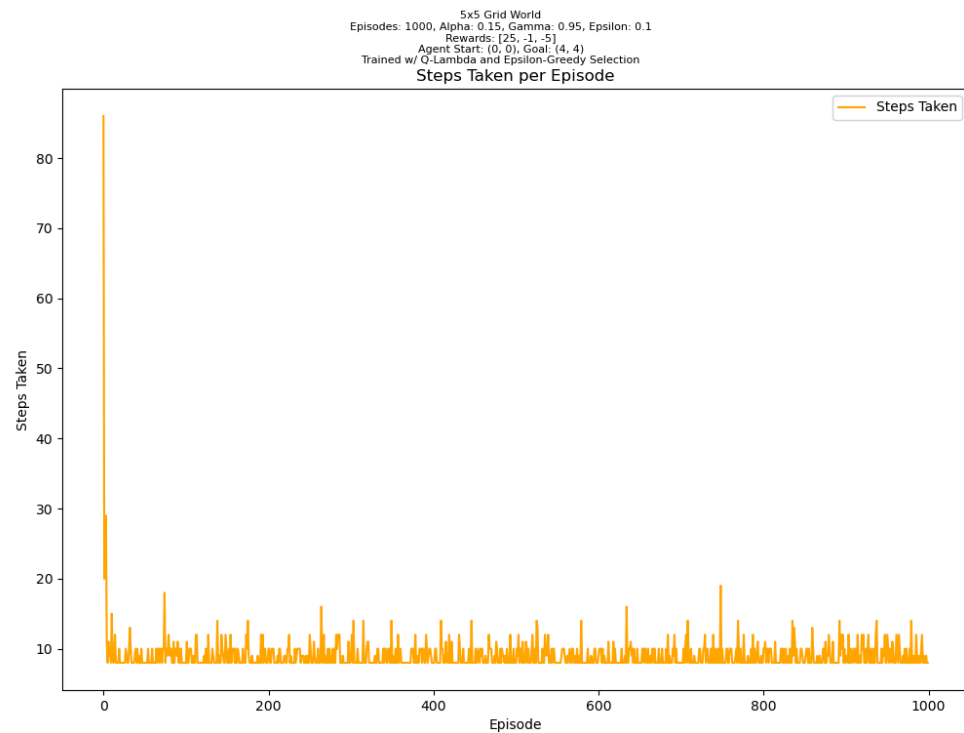
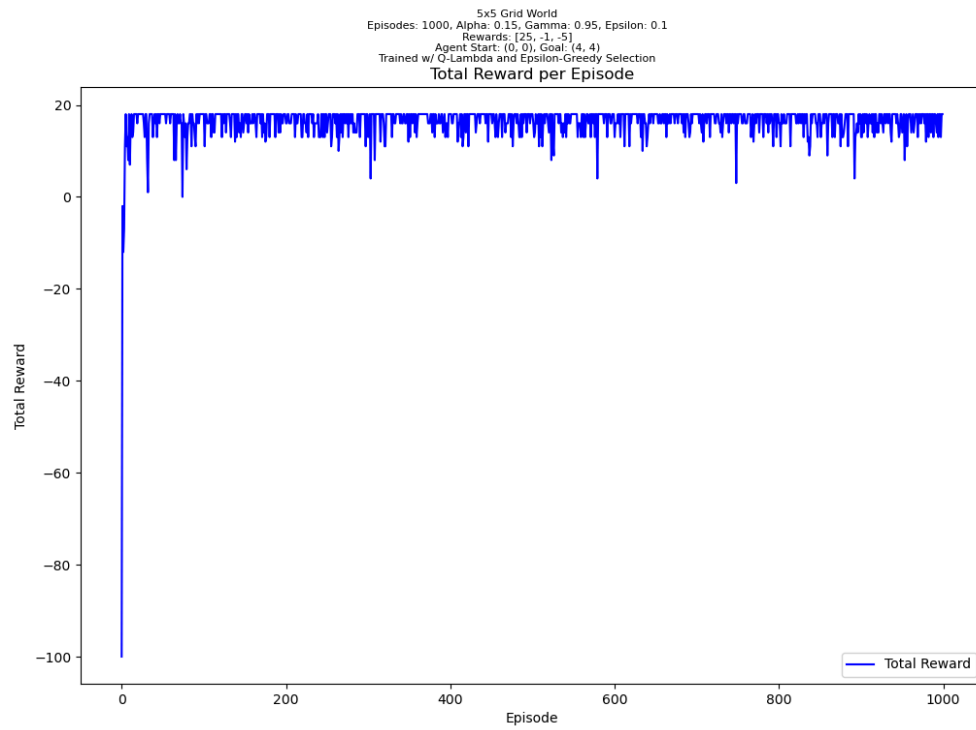
5x5 Grid World
Episodes: 1000, Alpha: 0.15, Gamma: 0.95, Epsilon: 0.1
Rewards: [25, -1, -5]
Agent Start: (0, 0), Goal: (4, 4)
Trained w/ Q-Lambda and Epsilon-Greedy Selection
First Q-table

State(x,y)	up	down	left	right
(0,0)	-0.9646128012754404	-0.6731835082506752	-1.0219010470372052	-0.5724232569204315
(0,1)	-0.6354029973426721	-0.6096160304768554	-0.8870268634689205	-0.5414393017206759
(0,2)	-0.8267747594787154	-0.4211775262491951	-0.8861053794978164	-0.5602948321246722
(0,3)	-0.5709000552614628	-0.6353682623490561	-0.8889880143373116	-0.2926063459732882
(0,4)	-0.5722693068102233	0.0	-1.021827920734856	0.0
(1,0)	-0.8893121198324877	-0.2932886733315532	-0.3192849787548059	-0.28640371459244635
(1,1)	-0.3016603649085329	-0.2893635060300737	-0.29153180484840063	-0.5488189938581806
(1,2)	-0.28744766536428495	-0.5529018719788424	-0.7145968522931592	-0.3500340815556509
(1,3)	-0.6351115798483544	-0.3231873579914403	-0.4517999246158021	-0.5813119008708084
(1,4)	-0.364604964192506	0.0	0.0	0.0
(2,0)	-0.8858698381049203	-0.28716571493146603	-0.2860417644314121	0.0
(2,1)	-0.570788173099837	-0.2887699261715074	-0.29439185891495007	-0.6176774713039129
(2,2)	-0.421124382224228	-0.29214721299264707	-0.314225874759654	0.14680078124999996
(2,3)	-0.2992572905108357	-0.29487083434195155	-0.6392442264192648	-0.31626081317800875
(2,4)	-0.30499123019358204	0.0	0.0	0.0
(3,0)	-0.8857642416471777	-0.28581945609932197	0.0	0.0
(3,1)	-0.5707380147824093	-0.43117341622834693	-0.32107846924577965	-0.42079087823614236
(3,2)	-0.4211005570216445	-0.28618041484463963	-0.08026962890625	0.6248437499999999
(3,3)	-0.35002276458528125	-0.2866956101992413	-0.2900636463124269	0.0
(3,4)	-0.28778023199840275	0.0	0.0	0.0
(4,0)	-0.8843917040937056	-0.2829299033551696	0.0	0.0
(4,1)	-0.5700860594445101	-0.2798524281161464	-0.25973375342336336	0.0
(4,2)	-0.2733735328760976	1.63125	0.0	0.0
(4,3)	0.0	3.75	0.0	0.0
(4,4)	0.0	0.0	0.0	0.0

5x5 Grid World
Episodes: 1000, Alpha: 0.15, Gamma: 0.95, Epsilon: 0.1
Rewards: [25, -1, -5]
Agent Start: (0, 0), Goal: (4, 4)
Trained w/ Q-Lambda and Epsilon-Greedy Selection
Last Q-table

State(x,y)	up	down	left	right
(0,0)	5.9922164091412276	11.511032361535669	5.990530696048046	10.610922116515948
(0,1)	9.996938685063029	13.16170376672017	7.556864273308373	13.154048407365641
(0,2)	10.082937752864119	4.174162581060396	9.131545421015472	14.875963592577687
(0,3)	-0.8369602834374436	-0.22846052744116085	-0.8889880143373116	9.010094054864071
(0,4)	0.3367255505671712	-0.8886403327703686	-1.3529558365608194	0.7407872356518062
(1,0)	-1.6471961732420441	-0.9523964734274196	1.9113982297828942	13.809841335080662
(1,1)	8.426627402358694	14.850332643521364	6.855706539644782	9.926060808426564
(1,2)	13.233907321161379	13.18053315988554	13.128218375265323	16.668963649930348
(1,3)	14.917883018990002	-0.08452571303347409	0.38087904580425624	1.2559630933375527
(1,4)	-0.6632878795182398	-1.0217494750813145	-0.5721041580659251	3.418725149385029
(2,0)	0.6120080460795149	15.84328243603548	0.9975016108349521	-0.584719803765449
(2,1)	3.7082574112911706	16.723430697739023	1.450761663281637	5.079488147100694
(2,2)	14.75850105275395	13.66840640674216	14.80936752502636	18.59381875233735
(2,3)	16.165588104123998	-0.8142876372769964	-0.6392442264192648	5.378230849428824
(2,4)	-0.30499123019358204	-0.8708366907080795	-0.25439303306964106	7.051543086300115
(3,0)	-0.8857642416471777	9.176869695006953	-0.30830548233643096	-0.9526609724068713
(3,1)	0.2623244031039349	-0.43117341622834693	-0.4994464925705716	19.233312479661578
(3,2)	16.497838068817078	19.35866194289796	16.607282161033364	20.62275055322101
(3,3)	-0.35002276458528125	1.2460606844556448	3.637970302872176	22.744196874760156
(3,4)	1.7736688381010406	2.4021093749999998	0.006538253906249958	13.9073671875
(4,0)	-0.8843917040937056	-0.8012330136055937	4.432049301584189	-0.9961939773689514
(4,1)	1.8259880474701864	20.633583121642413	7.133257597942389	2.9888498814896516
(4,2)	19.437159312227593	22.751832468930036	18.17688170398175	16.45427767240069
(4,3)	20.57111698892302	24.999999999999999	20.084409453371055	18.021342201426798
(4,4)	0.0	0.0	0.0	0.0

Reward and Step Plots



Action Sequences

(Darker = Earlier Actions ; Brighter = Later Actions; Red = Invalid Steps)

