My First Reinforcement Learning

- **Number of States:** 70 (7 x 10 grid)
- **Number of Actions:** 4 [Up (0), Right (1), Down (2), Left (3)]

The strategy used to balance the learning rate (α) and the exploration rate (ε) was to keep them constant. The exploration rate could be implemented with a variation in function of the number of episodes, like:

$$\varepsilon = e^{-Number\ of\ Episodes}$$

To design the Q-Learning algorithm we used the pseudocode provided in the Temporal-Difference Learning chapter of the book Reinforcement Learning by Richard S. Sutton and Andrew G. Barto. The former is shown below.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

To implement the policy algorithm, we used the explanation made in the same book but for ε-*greedy* algorithm:

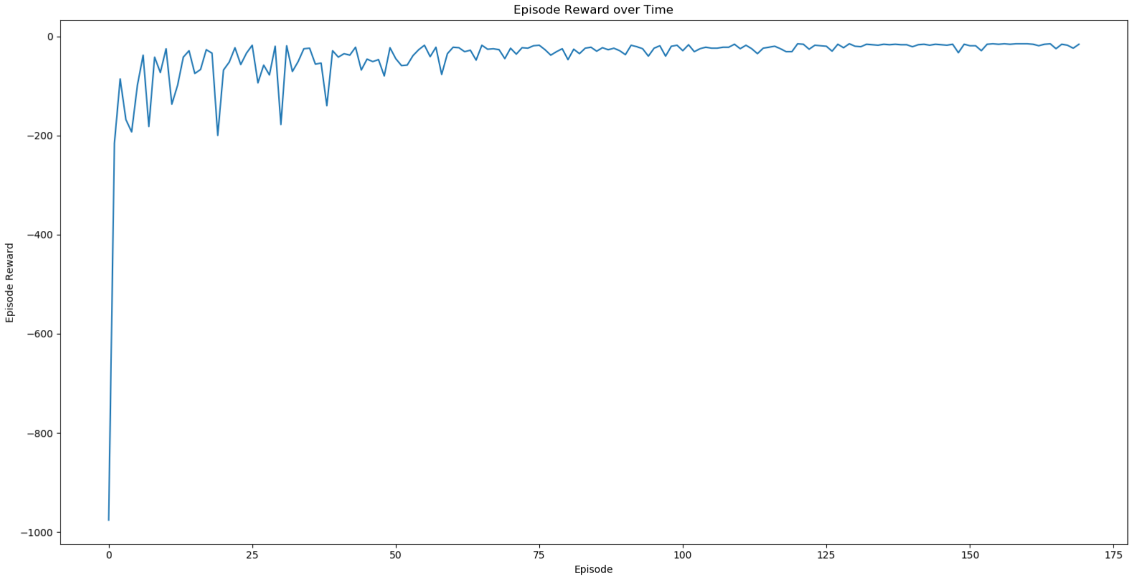
> The on-policy method we present in this section uses ε-*greedy* policies, meaning that most of the time they choose an action that has maximal estimated action value, but with probability ε they instead select an action at random. That is, all nongreedy actions are given the minimal probability of selection, $\frac{\varepsilon}{|\mathcal{A}(s)|}$, and the remaining bulk of the probability, $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$, is given to the greedy action.

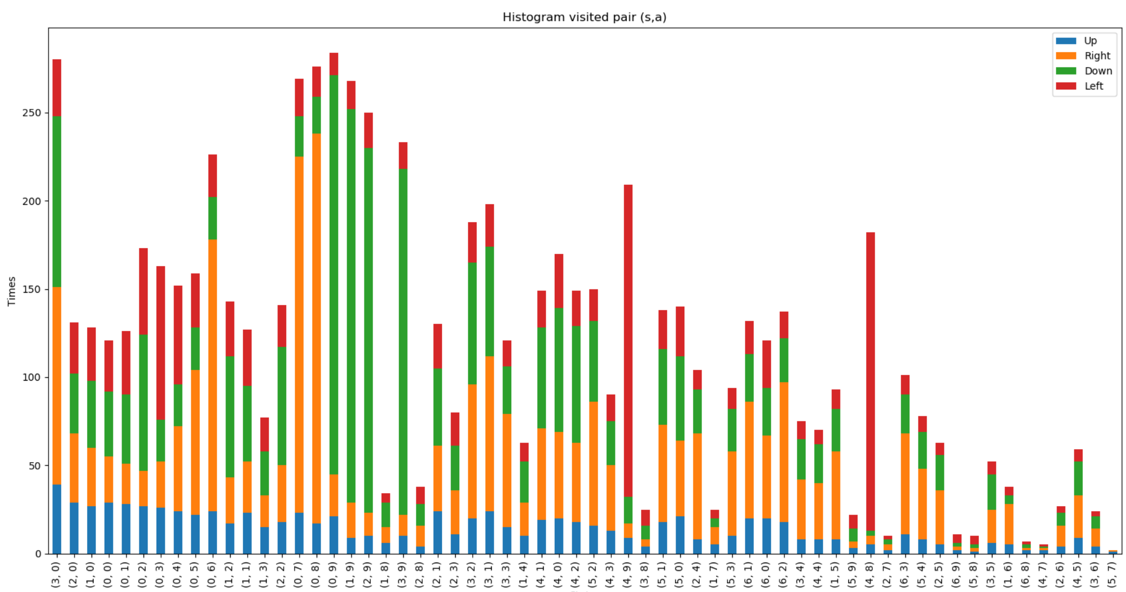
The states and actions of the environment are given by the Gym library as well as the reward in every state and a “done” flag when the agent steps in the goal state. The Q-Values are stored inside a dictionary in python and are initialized in 0, in other words, every state is initialized with the optimal value, given the current environment in which every state but final, have a negative reward.

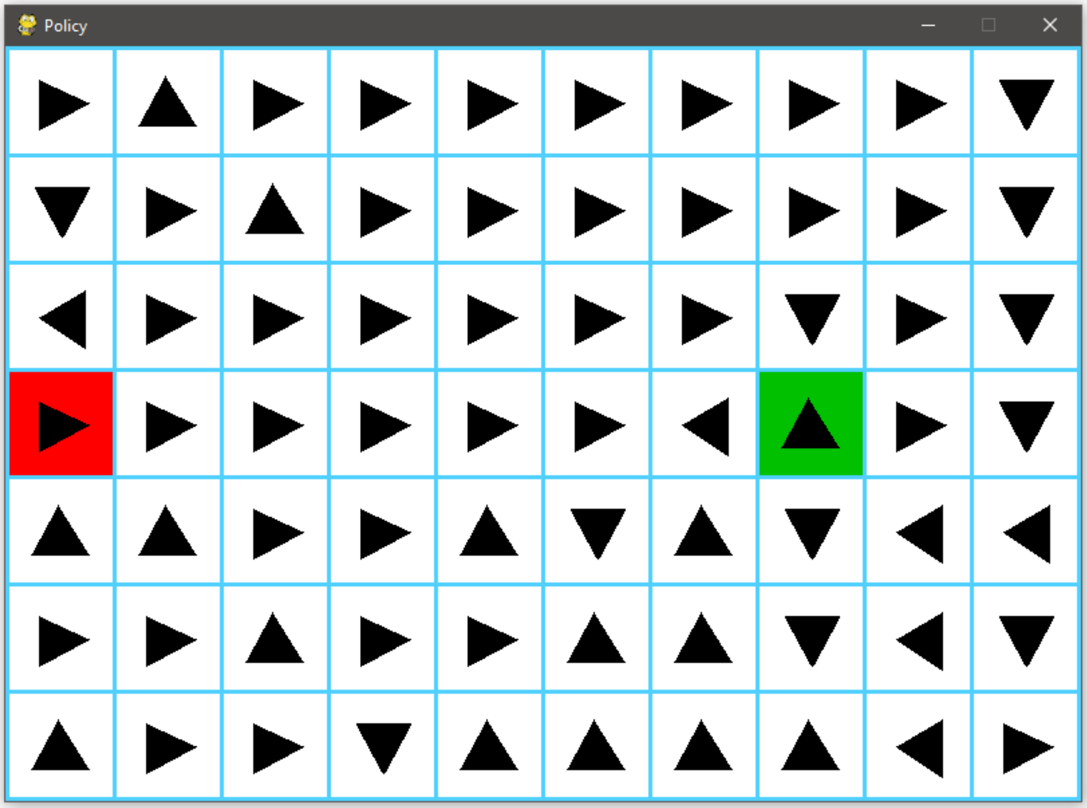
The firs try made with the implemented algorithm we used the same values for alpha, gamma, and epsilon provided by the solution given in the book:

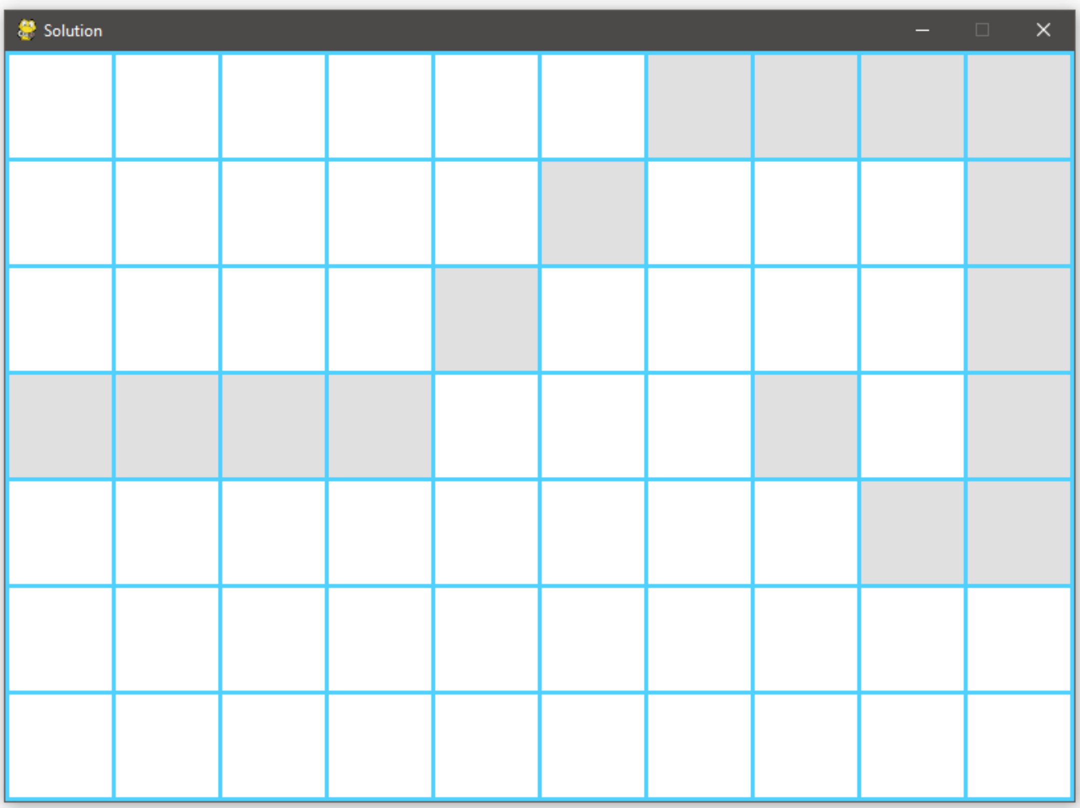
- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 170

The obtained results are shown below:

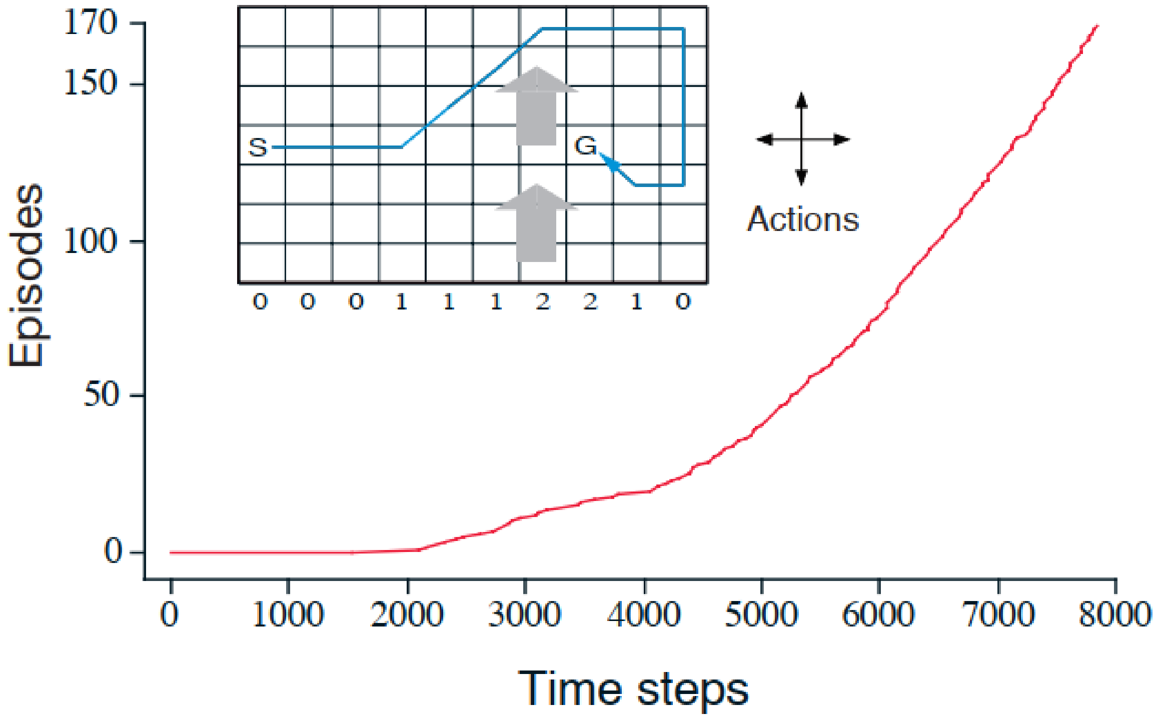





Policy


Solution

And as seen, the final result is completely satisfactory, fulfilling the solution given by the book.



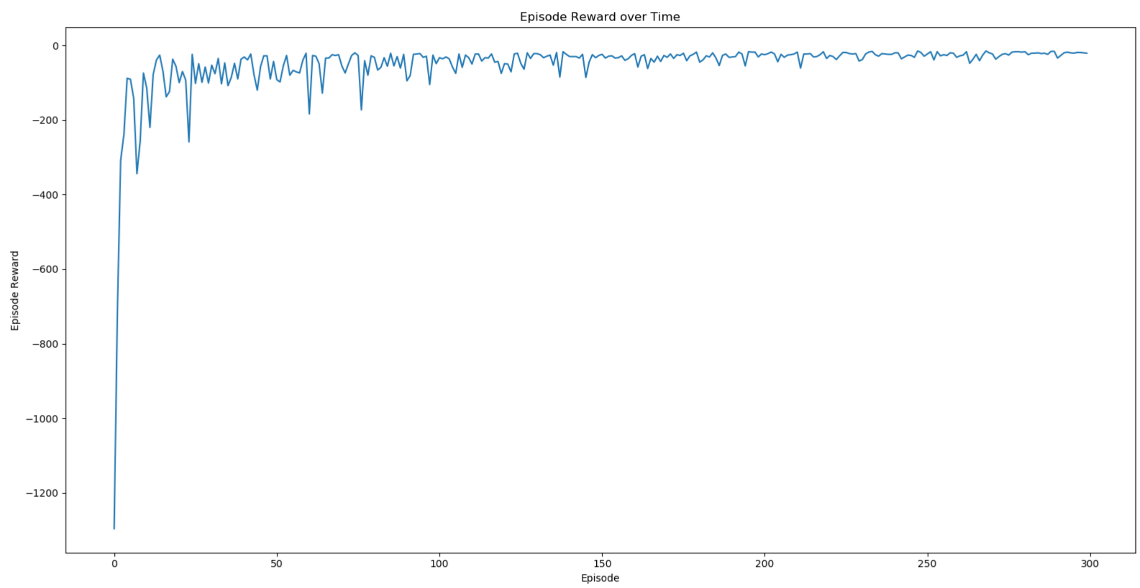
To realize the hyper-parameter sweeping, we decided to sweep 3 times each variable, and in the case of alpha, gamma and epsilon to implement a little piece of code in which the algorithm will use the minimal quantity of episodes to resolve the gridworld, this with the sake of comparison between variations.

First: Sweeping Alpha (Learning Rate)

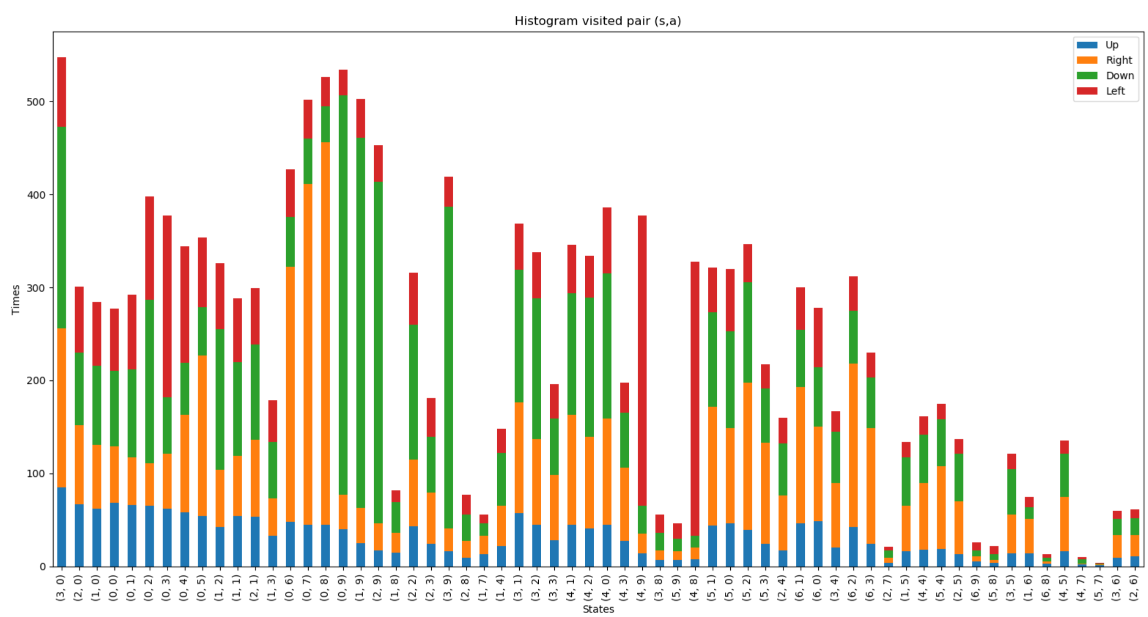
- **Alpha (α):** 0.2
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

Minimal number of Episodes: 151, 157, 154, 133, 179

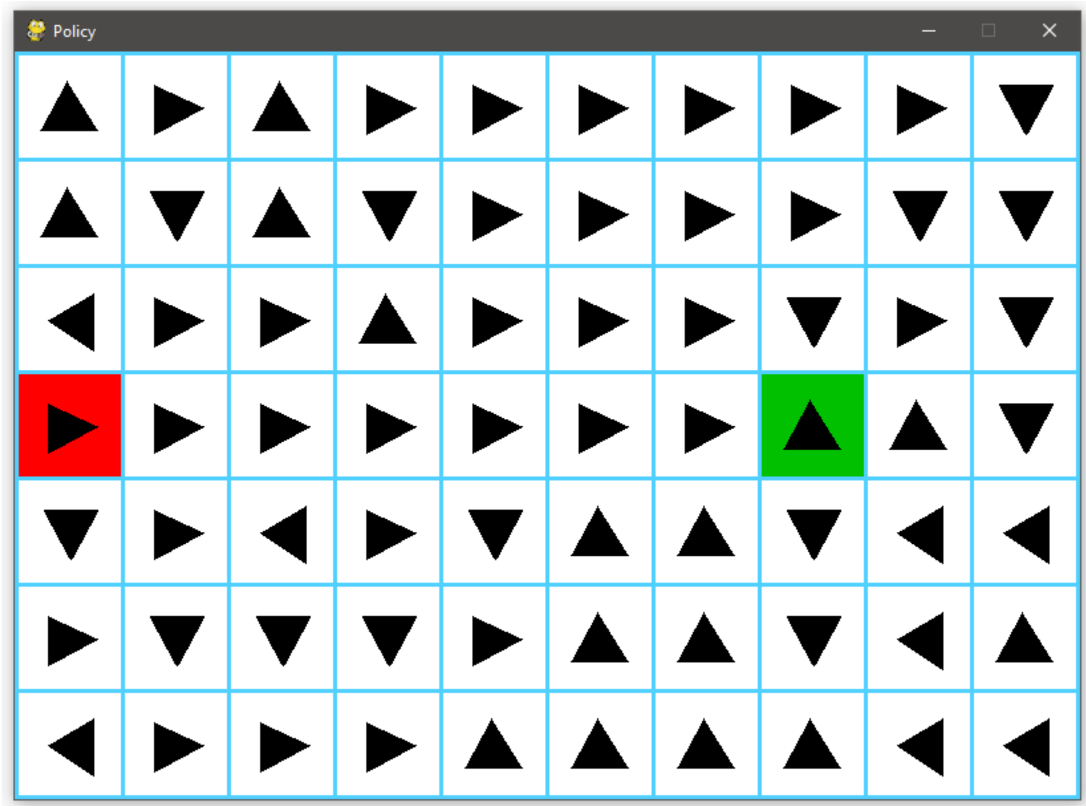
Average number of Episodes: 155

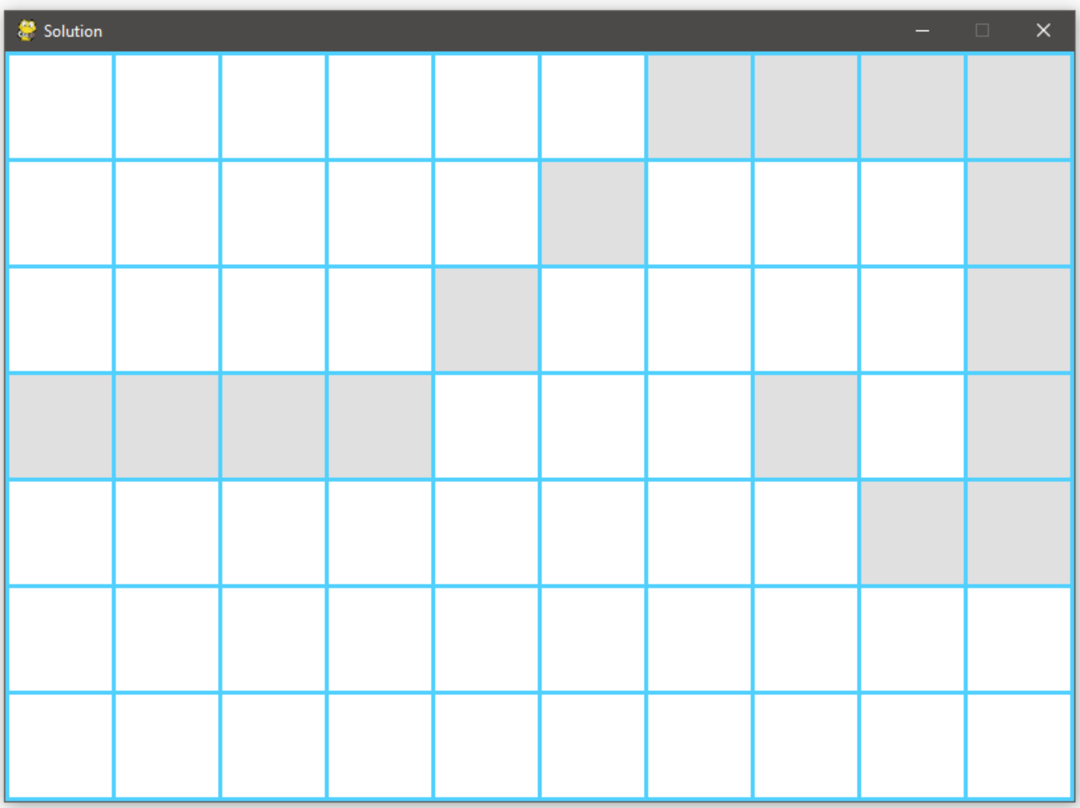


Episode Reward over Time
Episode Reward
Episode



Histogram visited pair (s,a)
Up
Right
Down
Left
Times
States

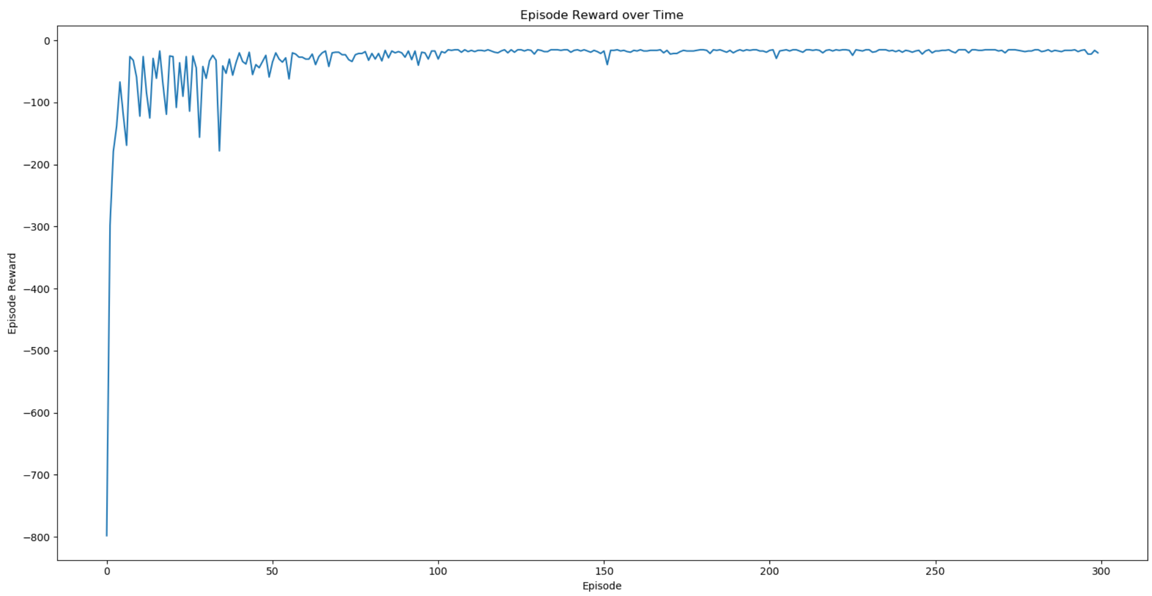

Policy

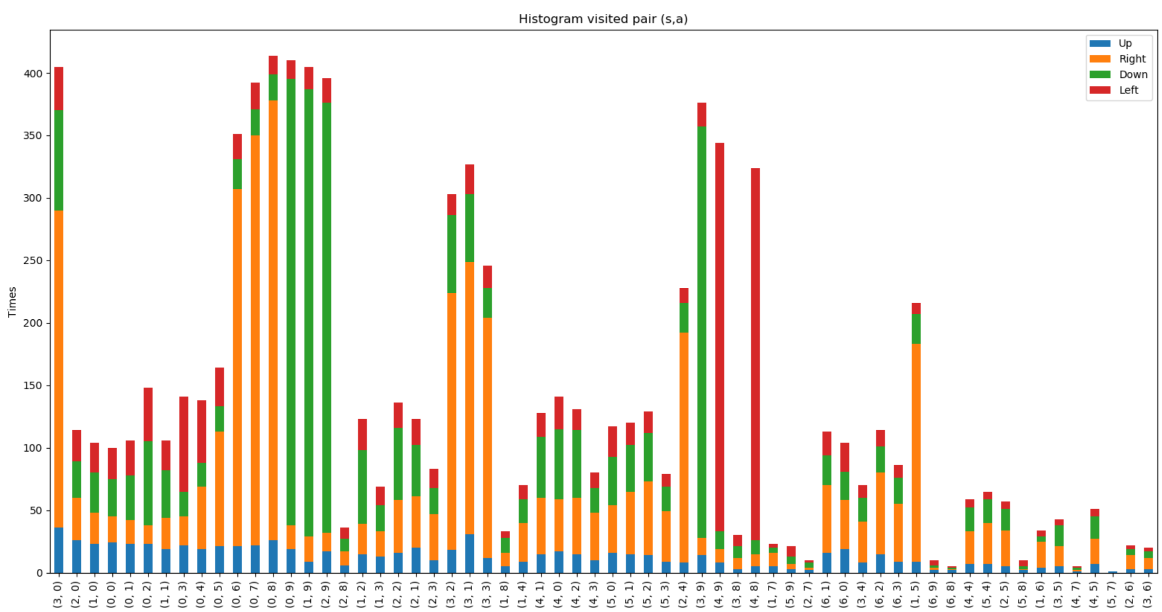

Solution

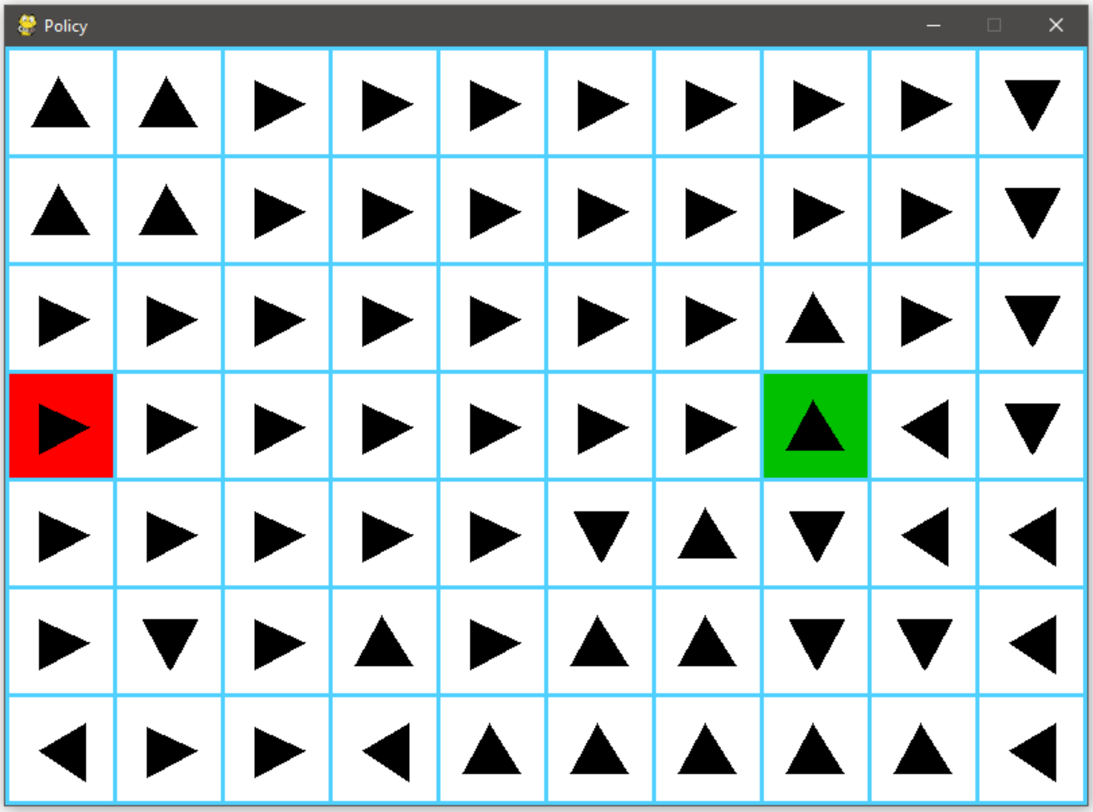
- **Alpha (α):** 0.6
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

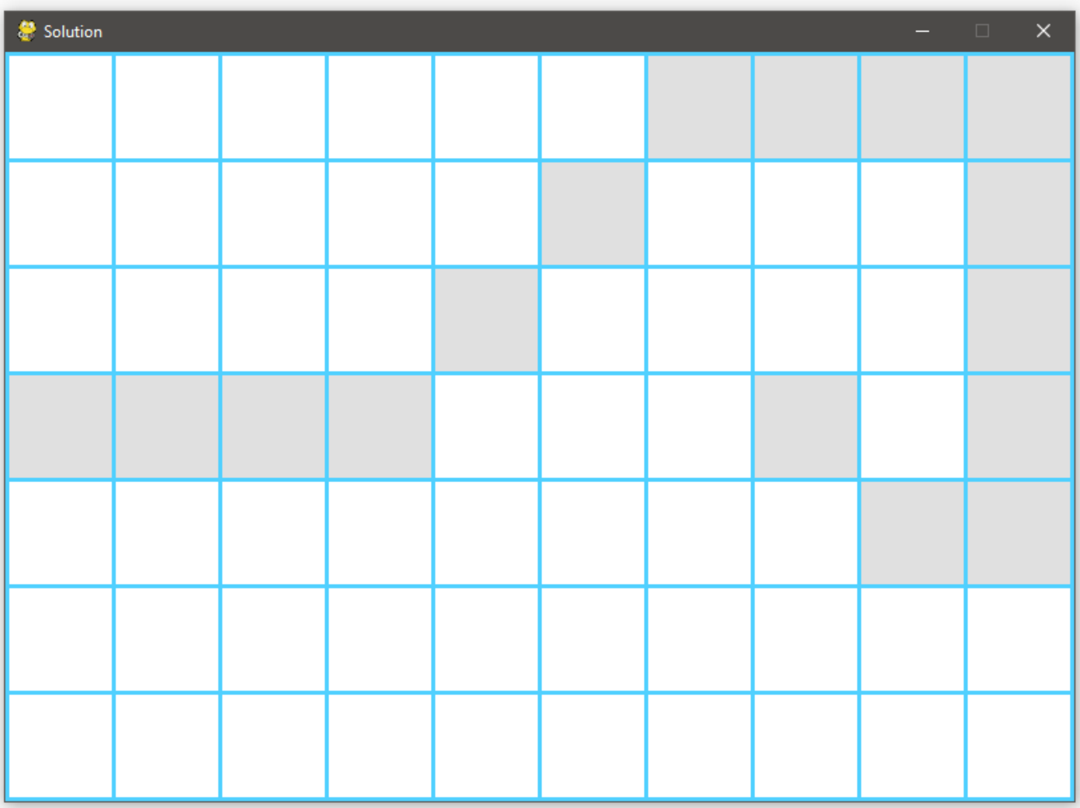
Minimal number of Episodes: 59, 42, 58, 52, 46

Average number of Episodes: 51





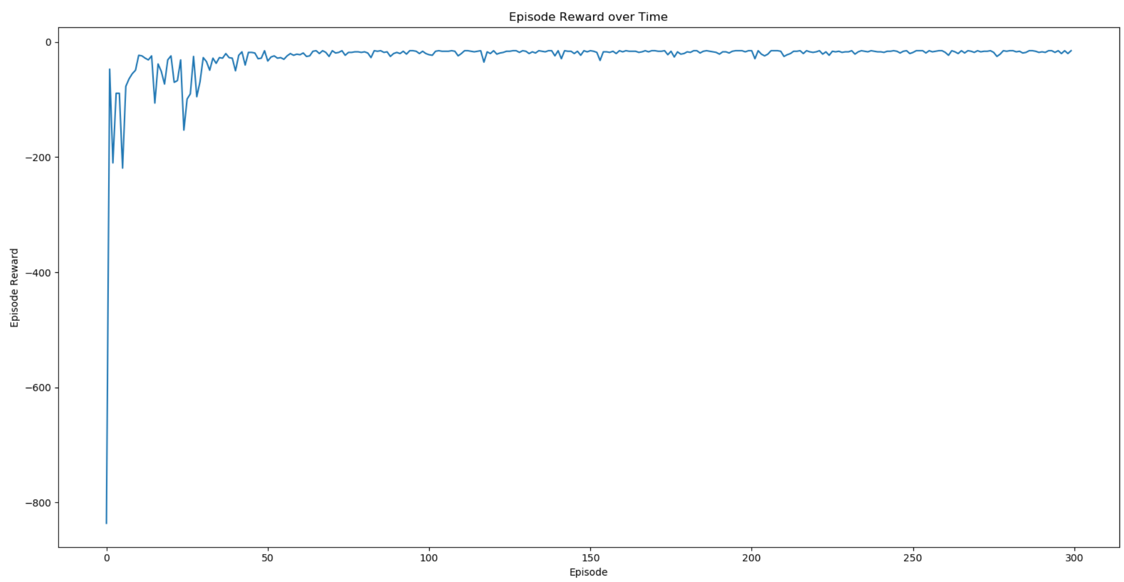

Policy

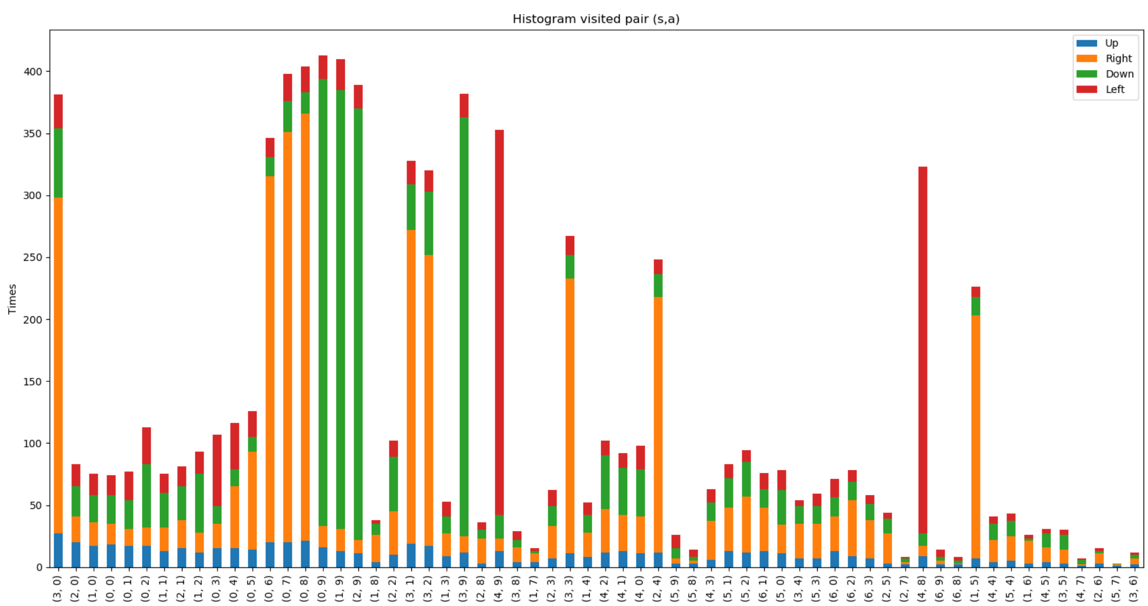

Solution

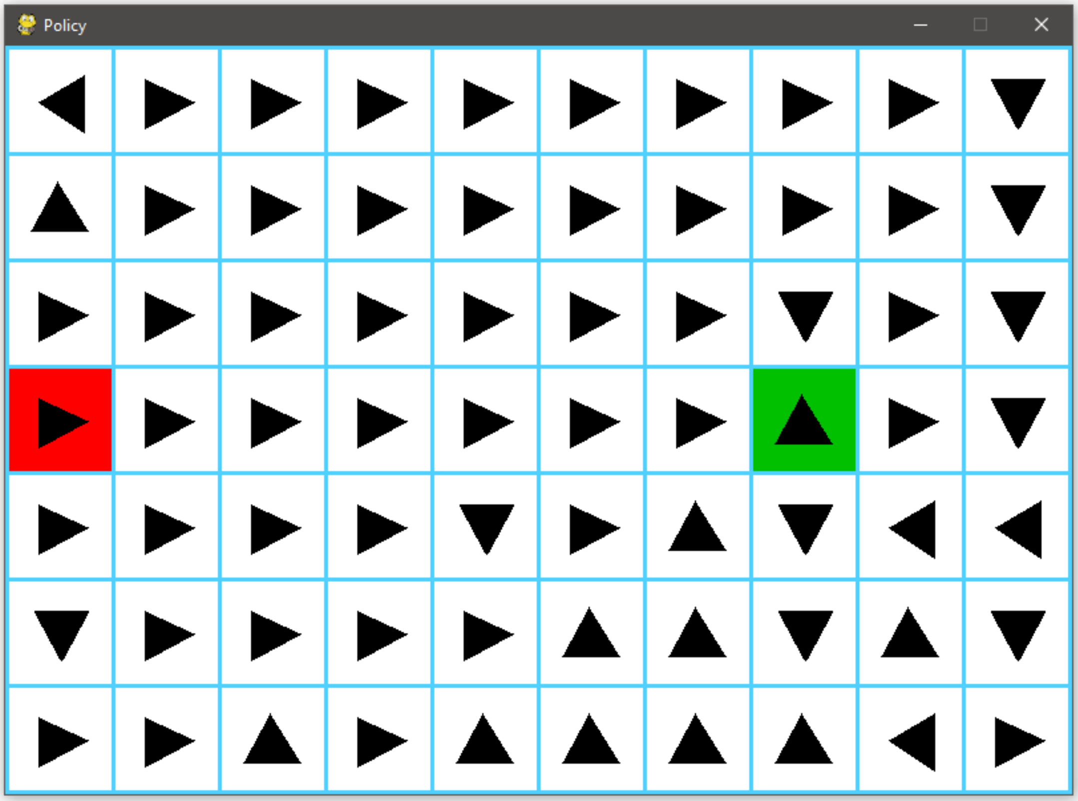
- **Alpha (α):** 0.9
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

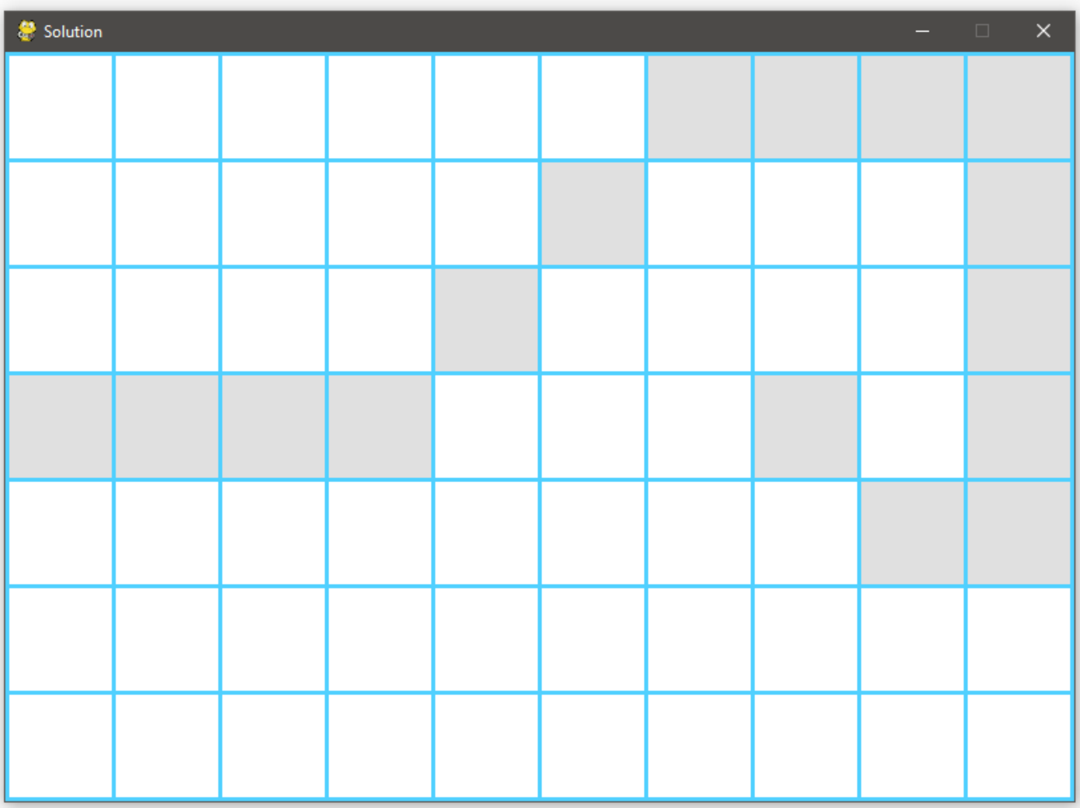
Minimal number of Episodes: 50, 58, 37, 62, 55

Average number of Episodes: 52






Policy

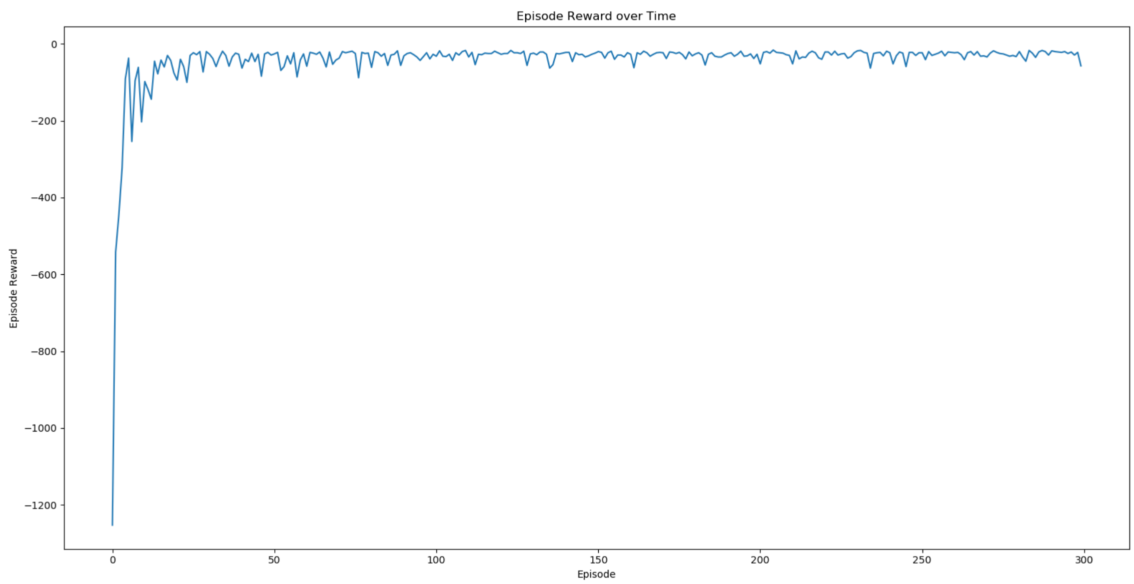

Solution

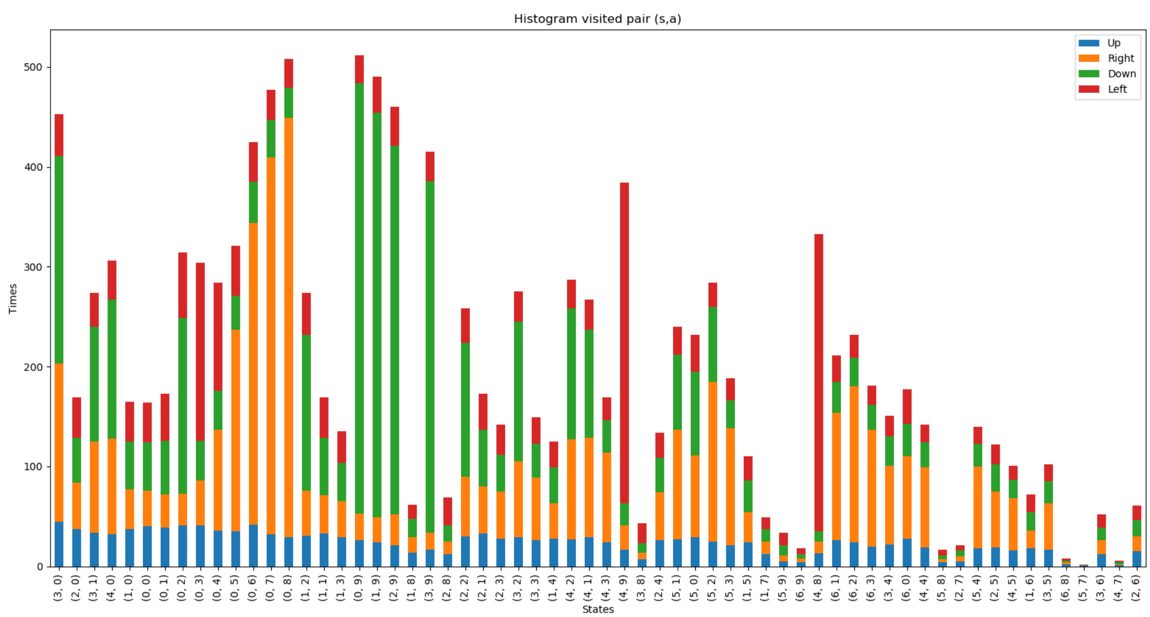
Second: Sweeping Gamma (Discounting Rate)

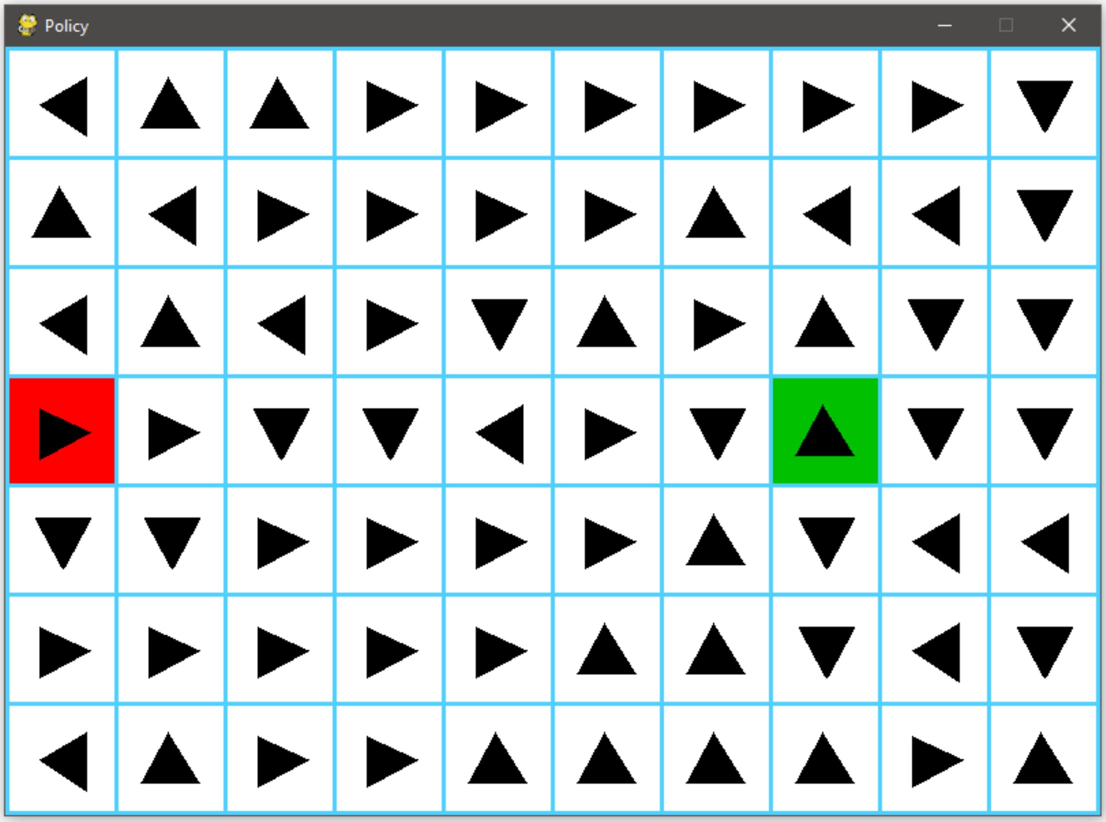
- **Alpha (α):** 0.5
- **Gamma (γ):** 0.2
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

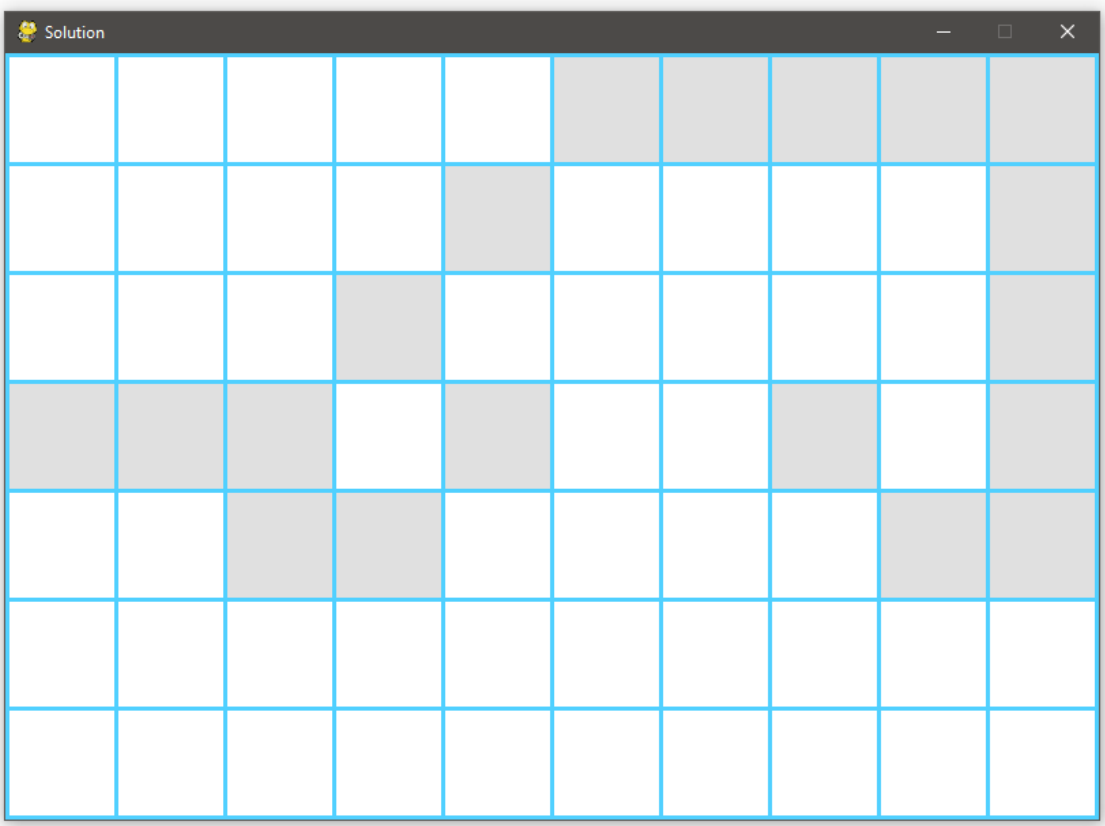
Minimal number of Episodes: 78, 55, 39, 51, 61

Average number of Episodes: 56





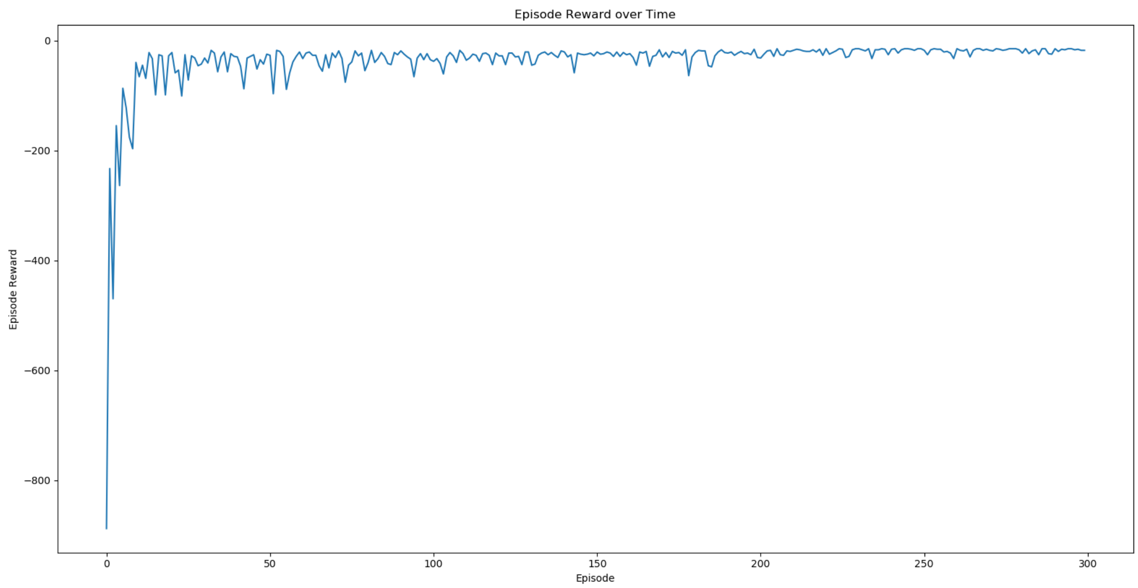

Policy

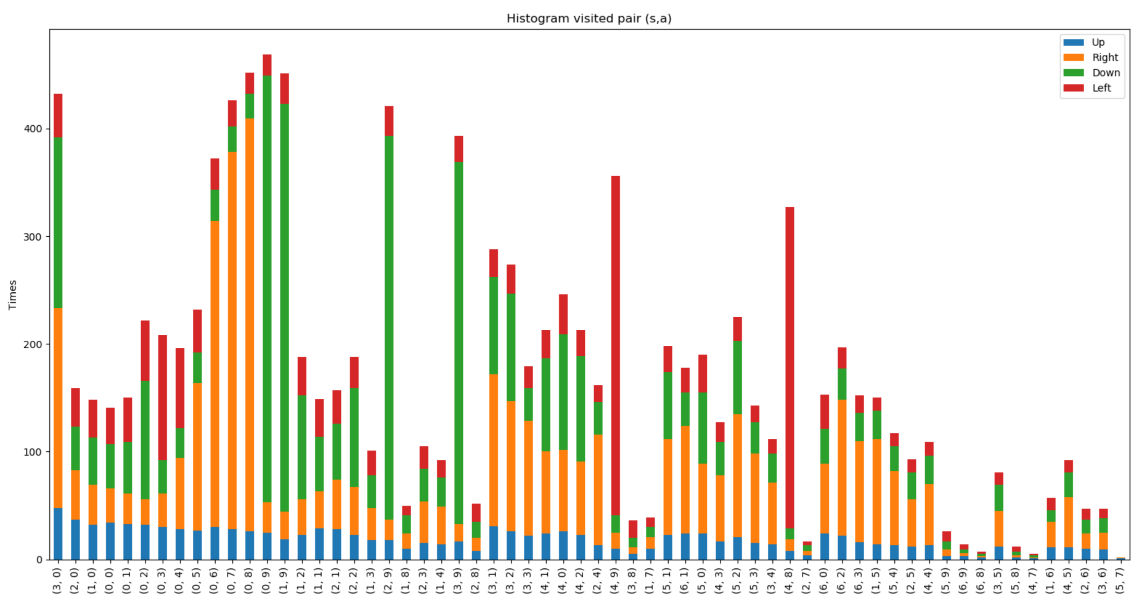

Solution

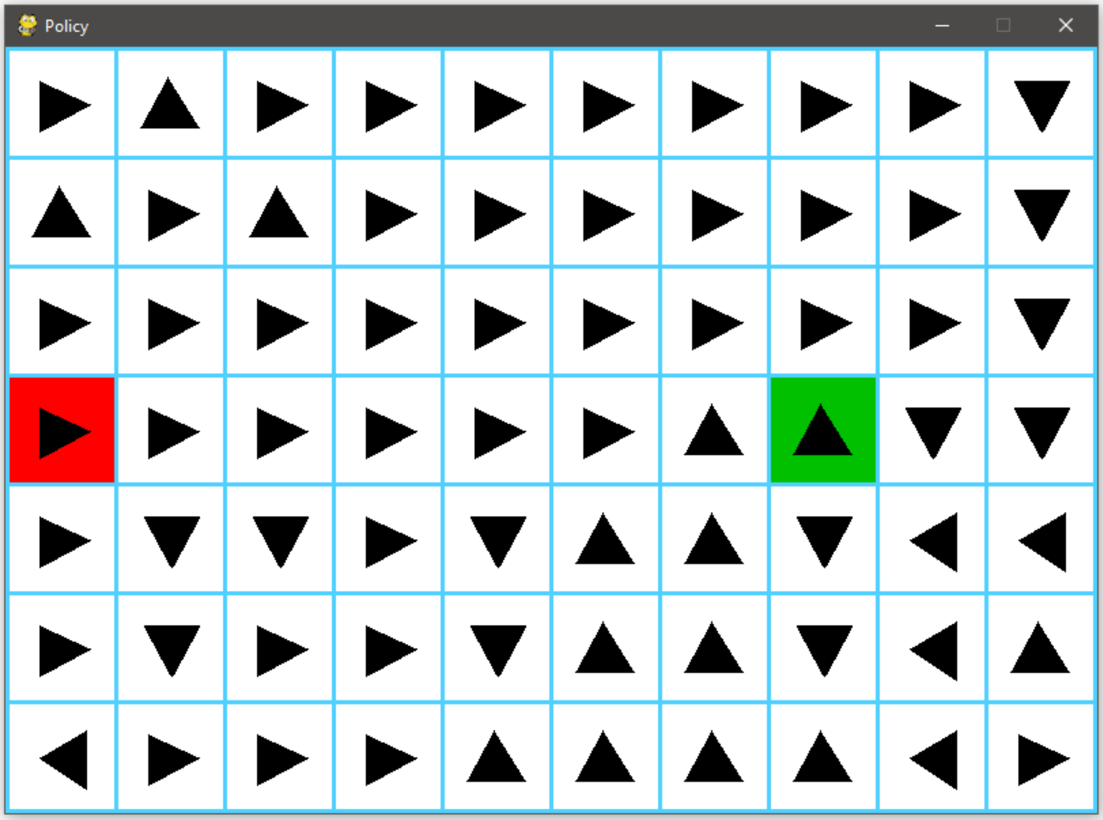
- **Alpha (α):** 0.5
- **Gamma (γ):** 0.5
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

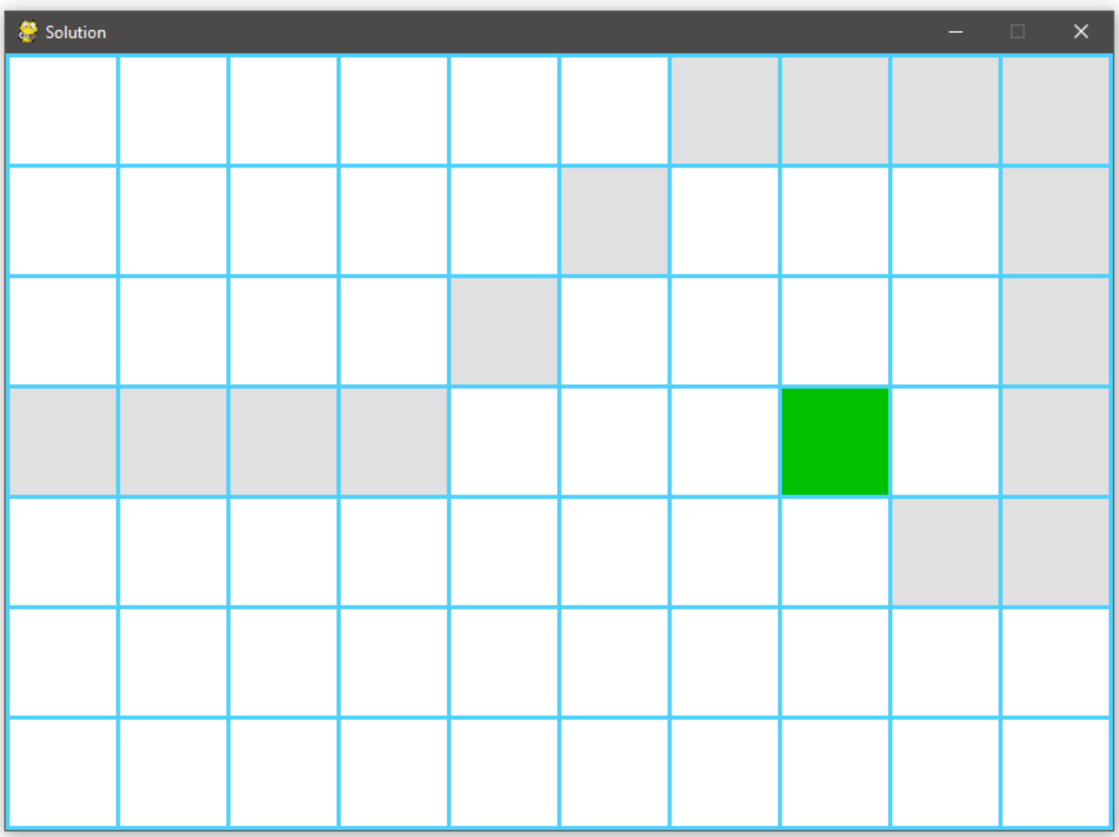
Minimal number of Episodes: 57, 43, 37, 45, 46

Average number of Episodes: 46





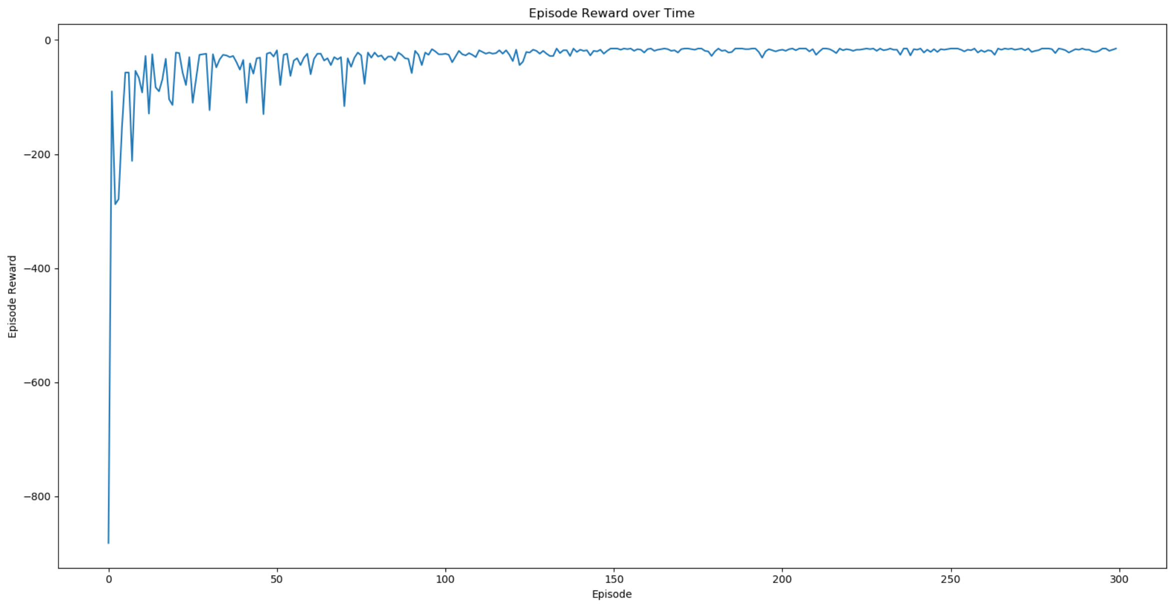

Policy

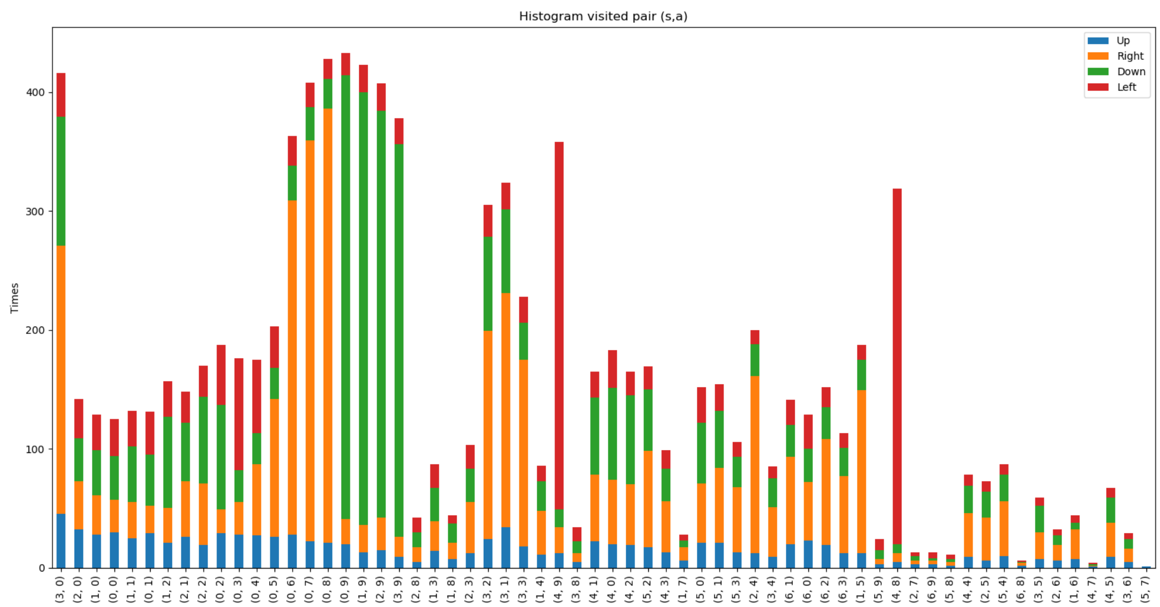

Solution

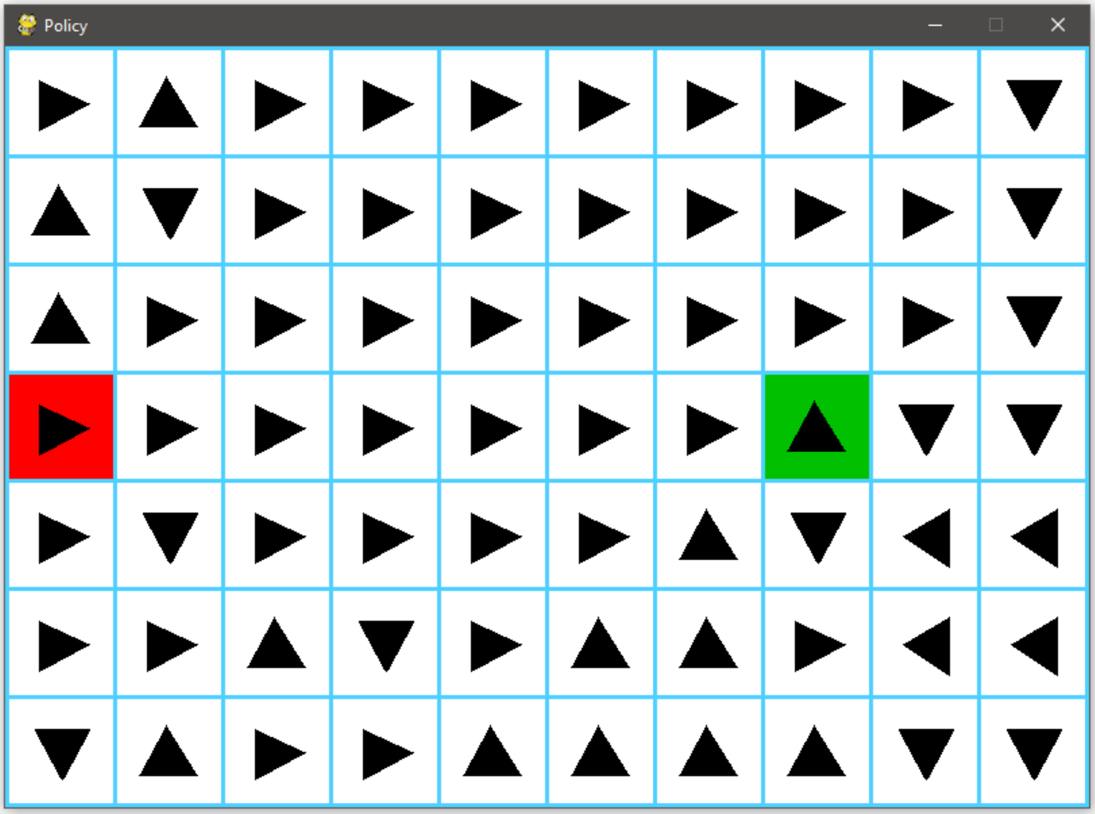
- **Alpha (α):** 0.5
- **Gamma (γ):** 0.8
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 300

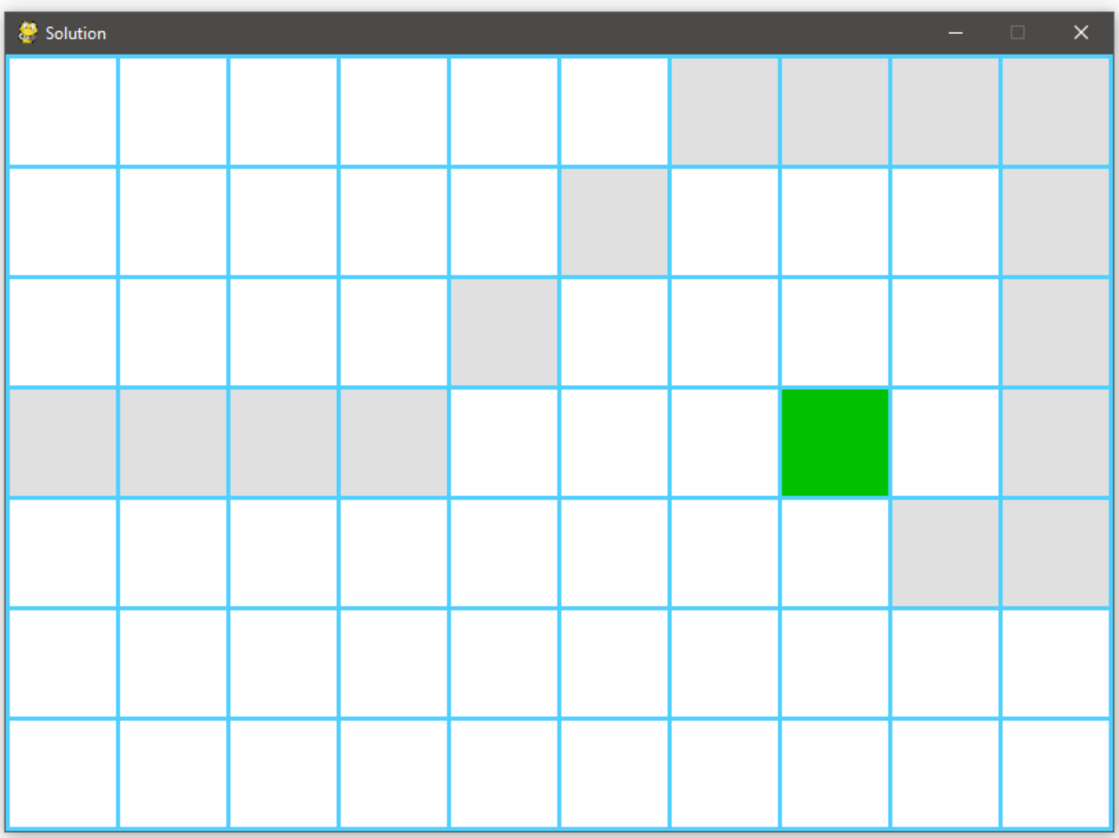
Minimal number of Episodes: 72, 70, 102, 79, 70

Average number of Episodes: 79






Policy

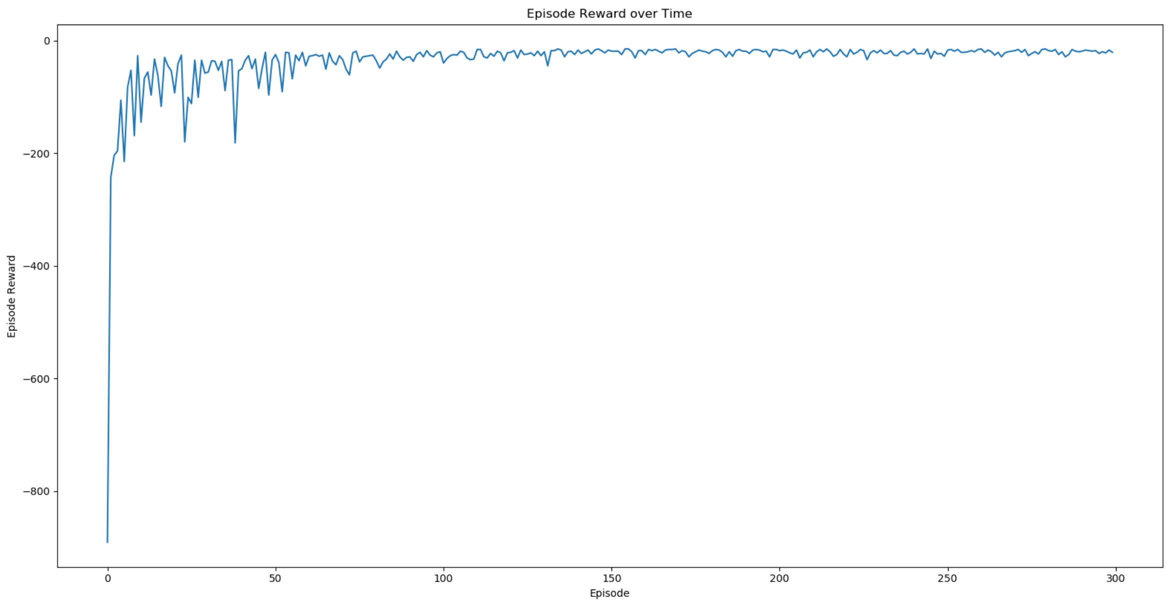

Solution

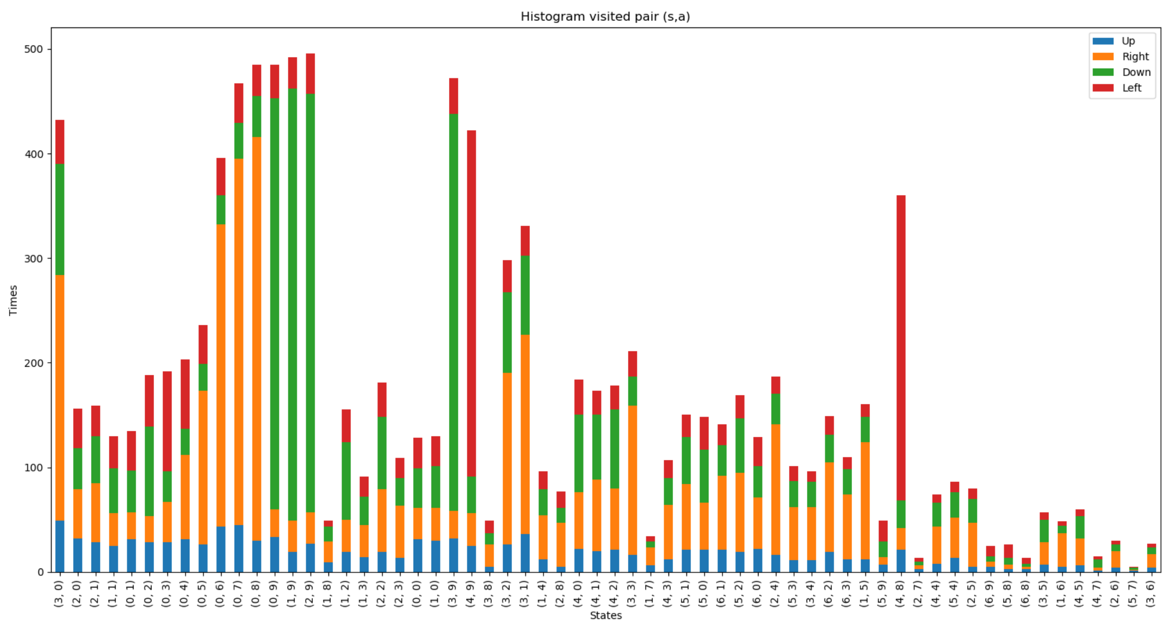
Third: Sweeping Epsilon (Exploring Rate)

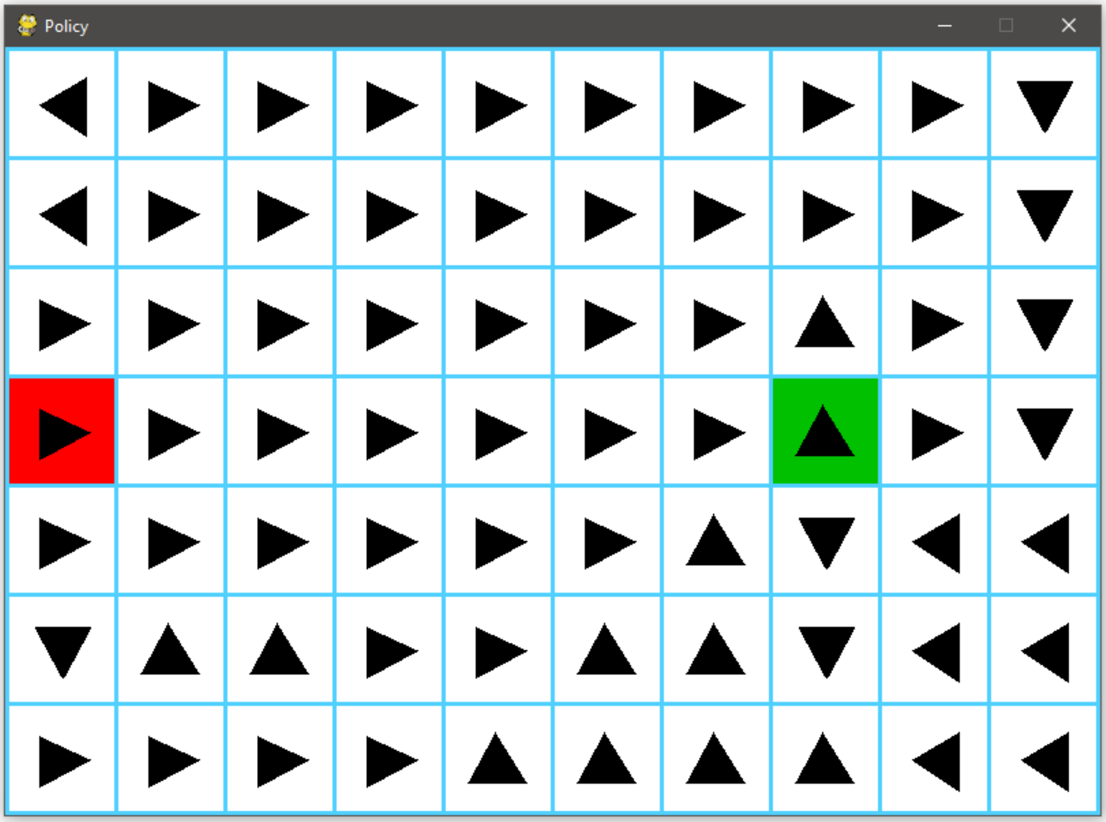
- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.2
- **Number of Episodes:** 300

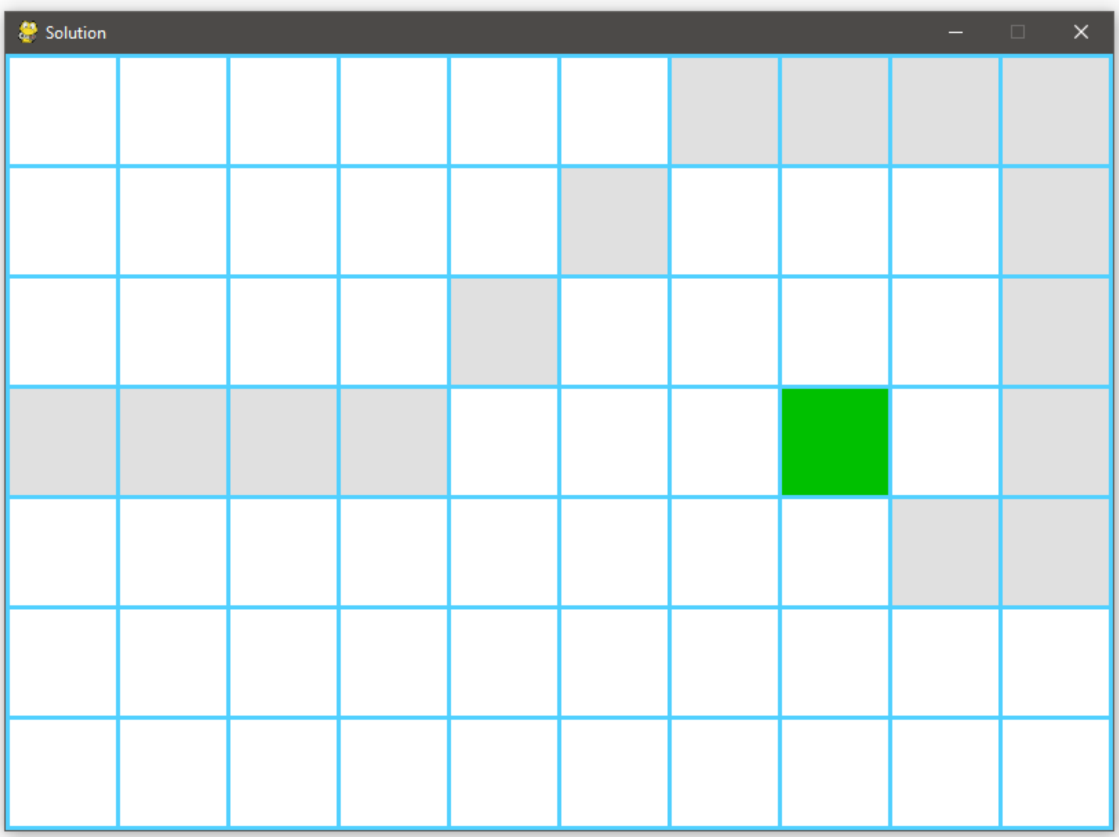
Minimal number of Episodes: 48, 76, 43, 66, 72

Average number of Episodes: 61





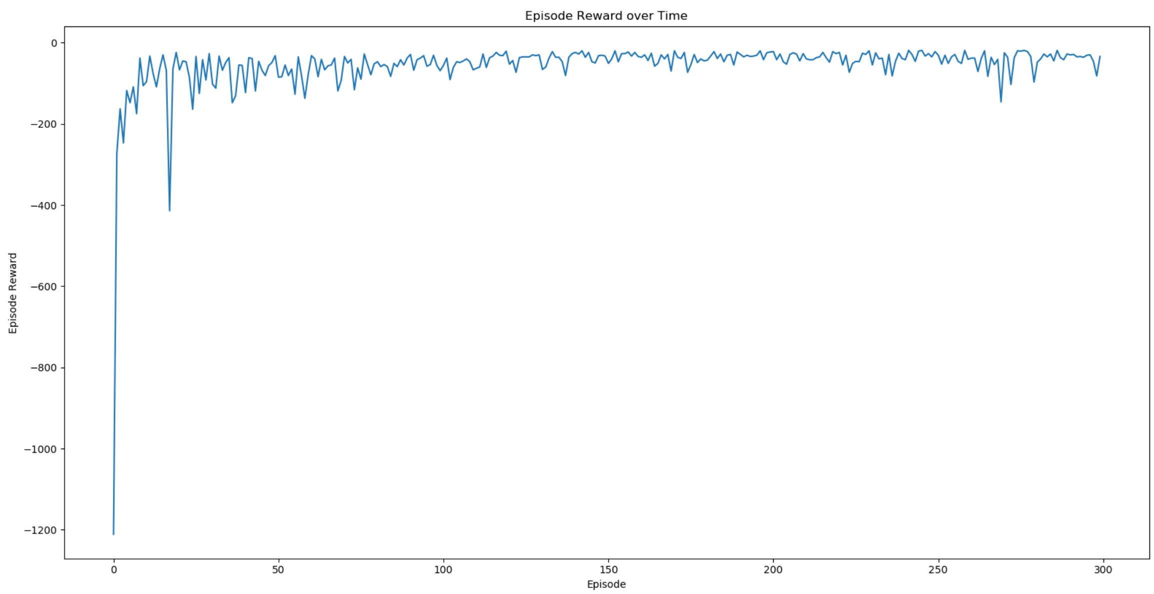

Policy

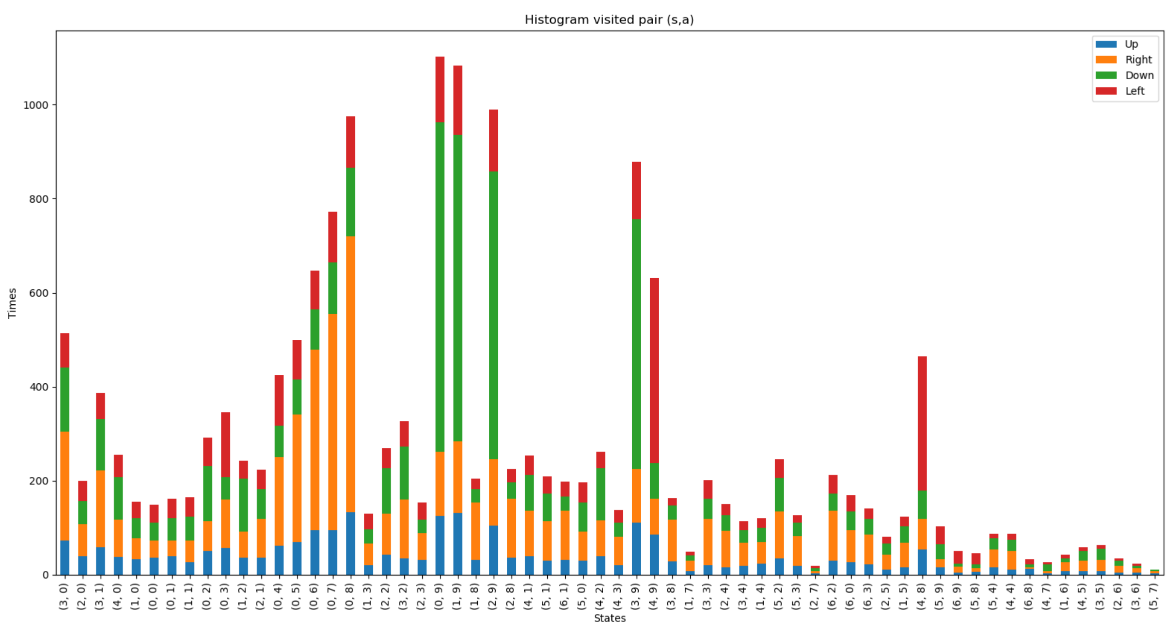

Solution

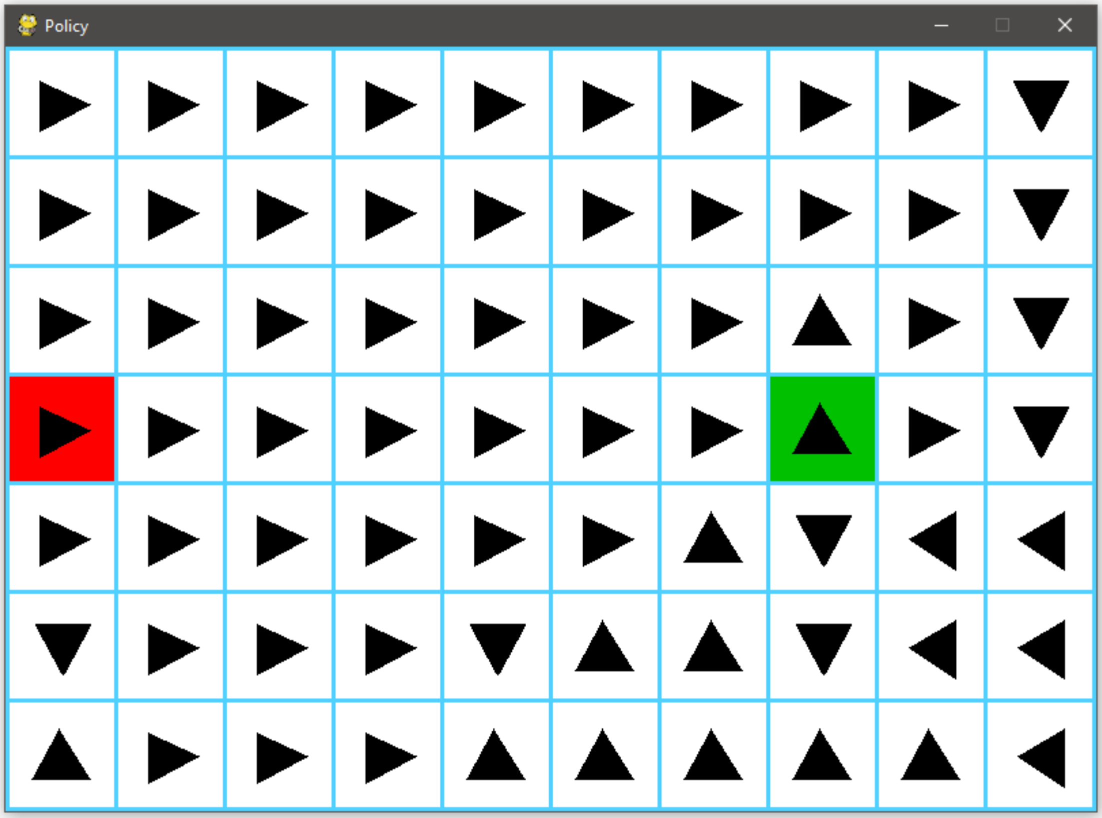
- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.5
- **Number of Episodes:** 300

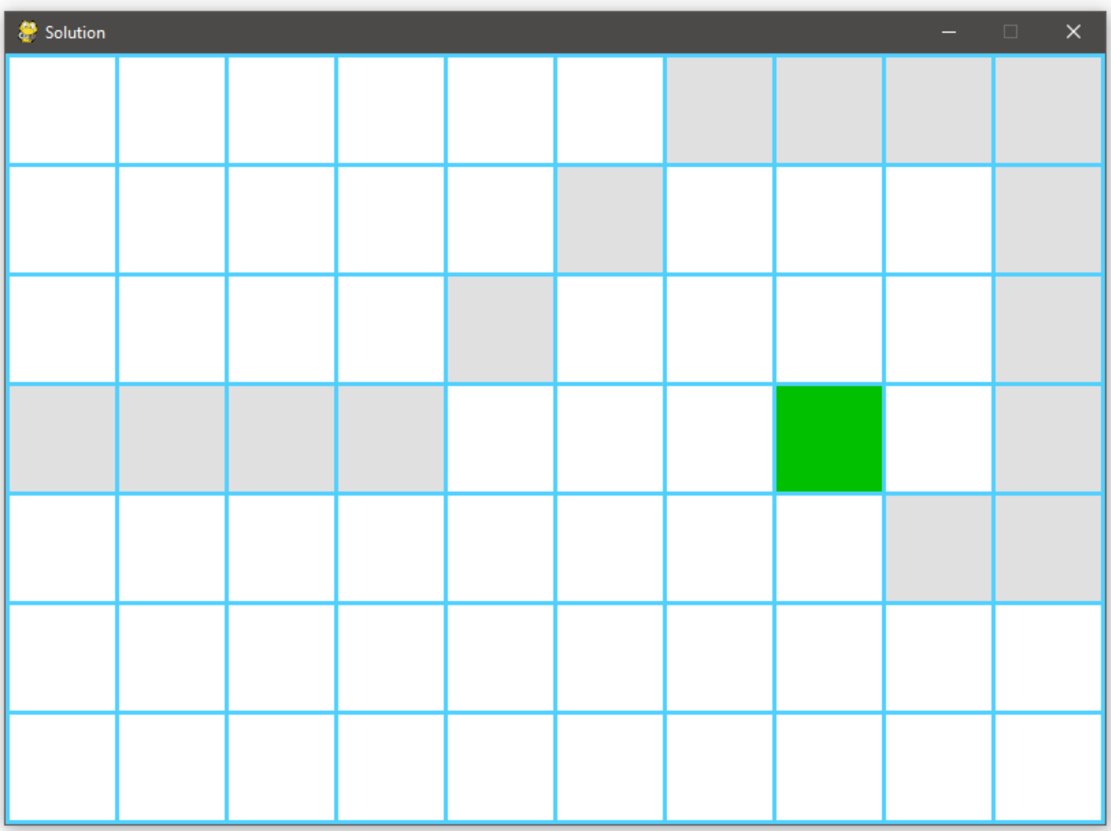
Minimal number of Episodes: 37, 31, 30, 32, 36

Average number of Episodes: 33





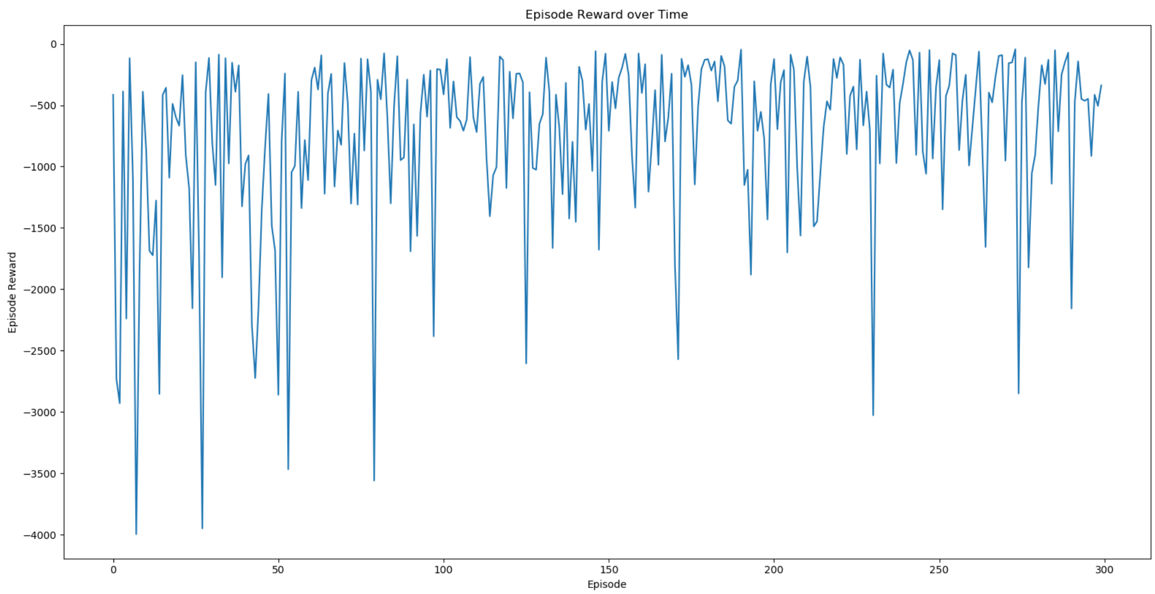

Policy

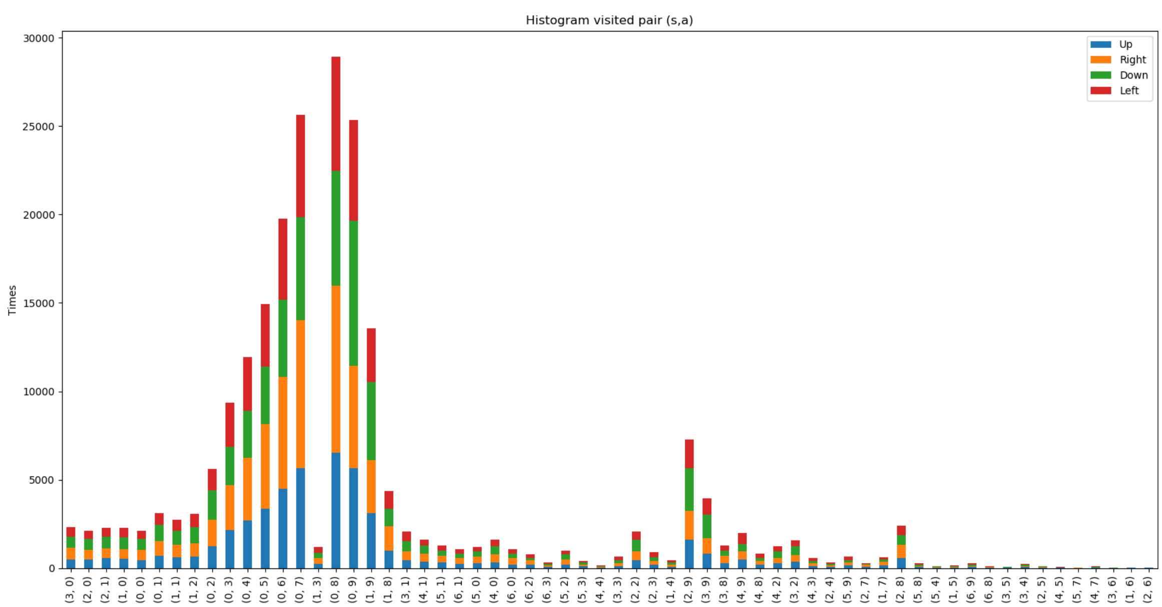

Solution

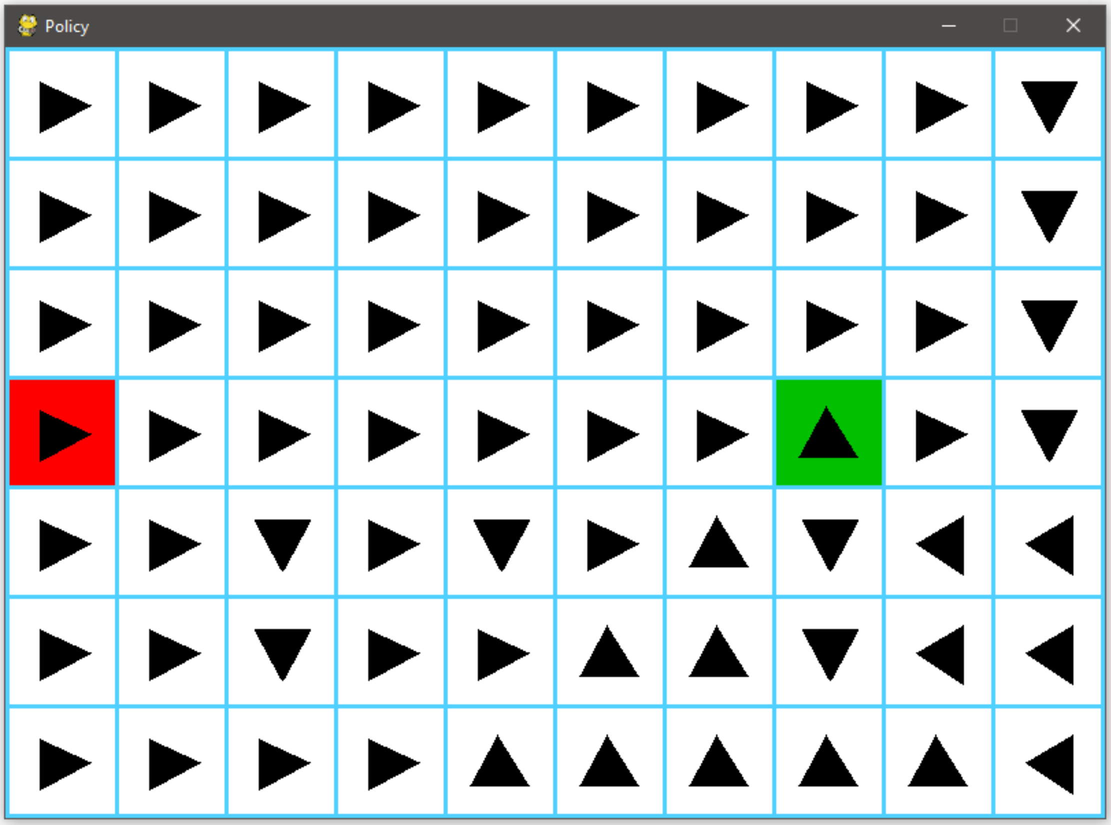
- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.9
- **Number of Episodes:** 300

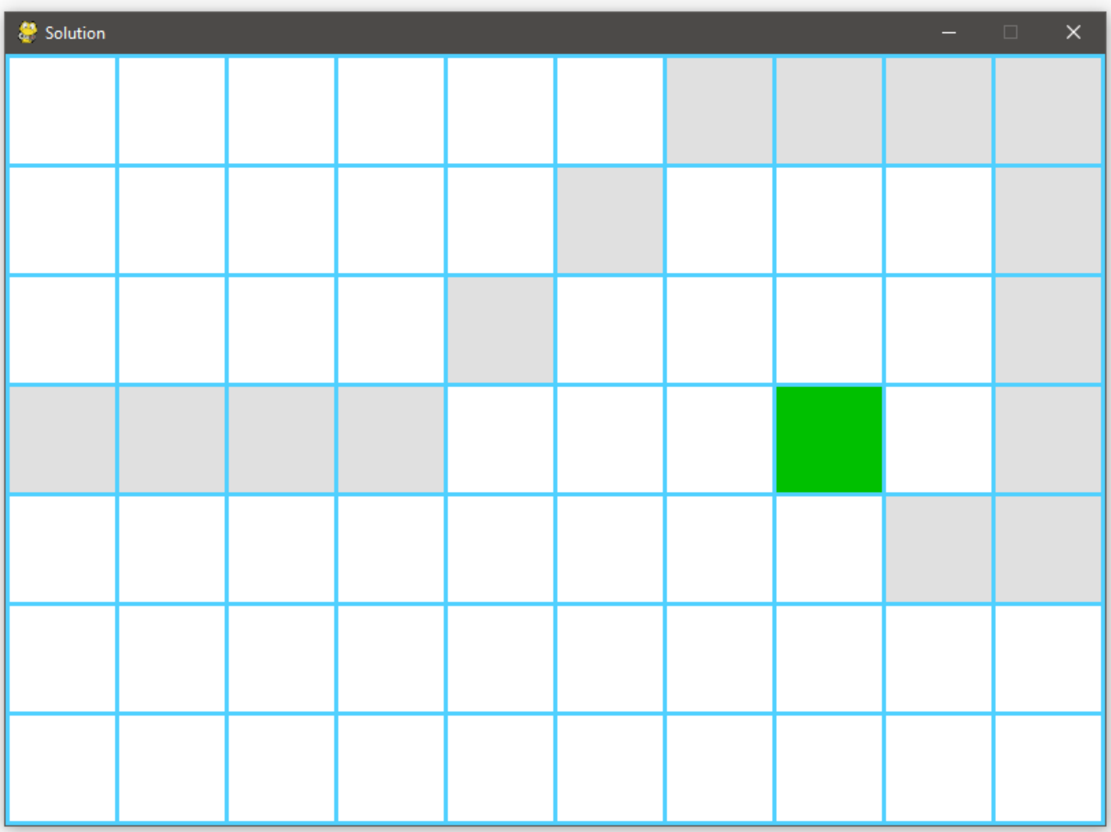
Minimal number of Episodes: 129, 416, 276, 432, 100

Average number of Episodes: 271



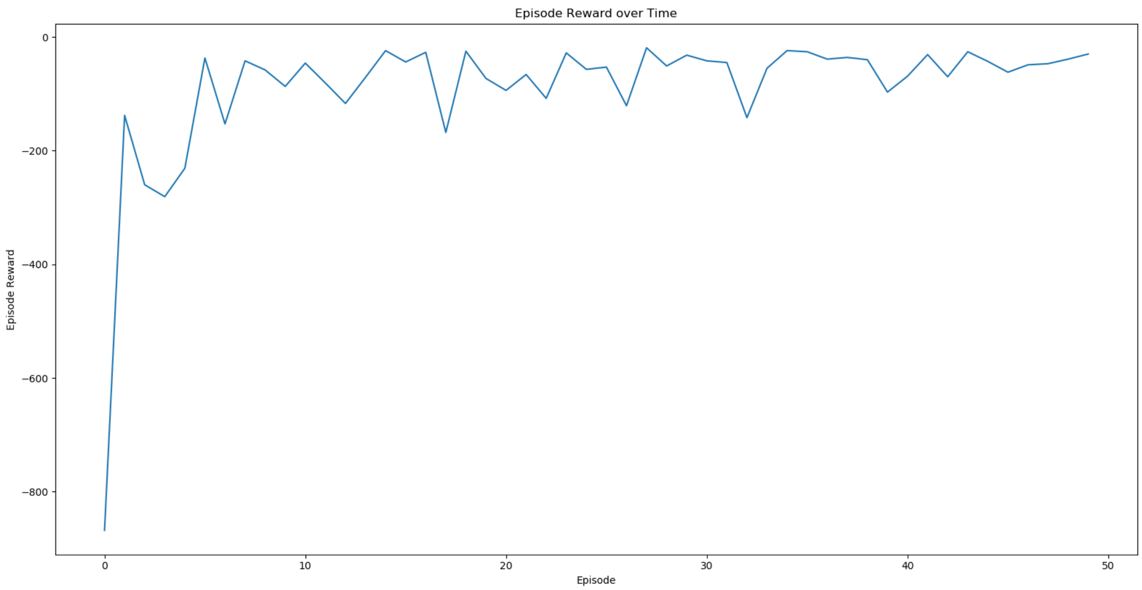


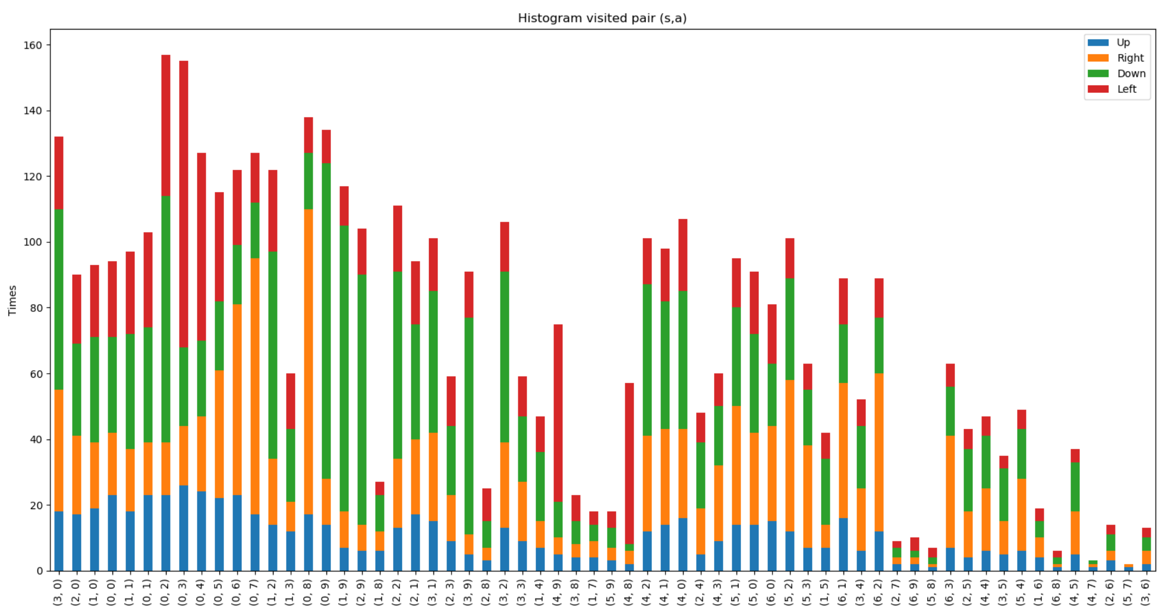

Policy

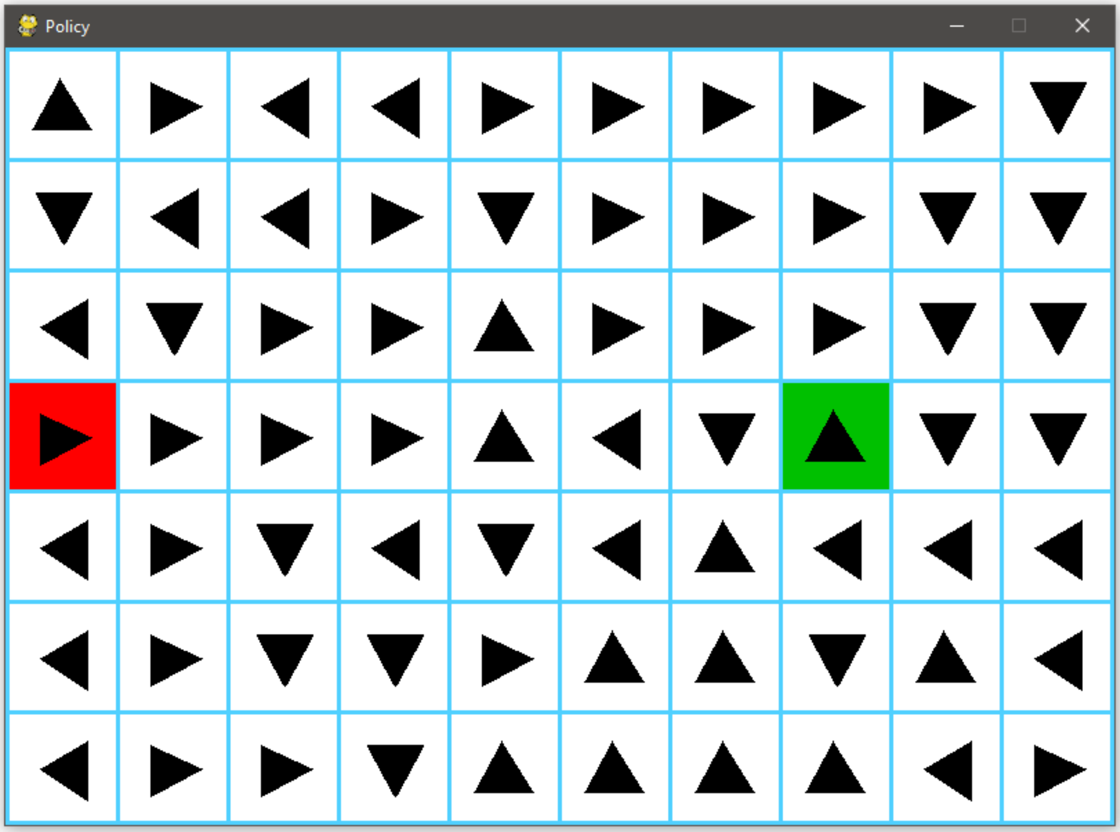

Solution

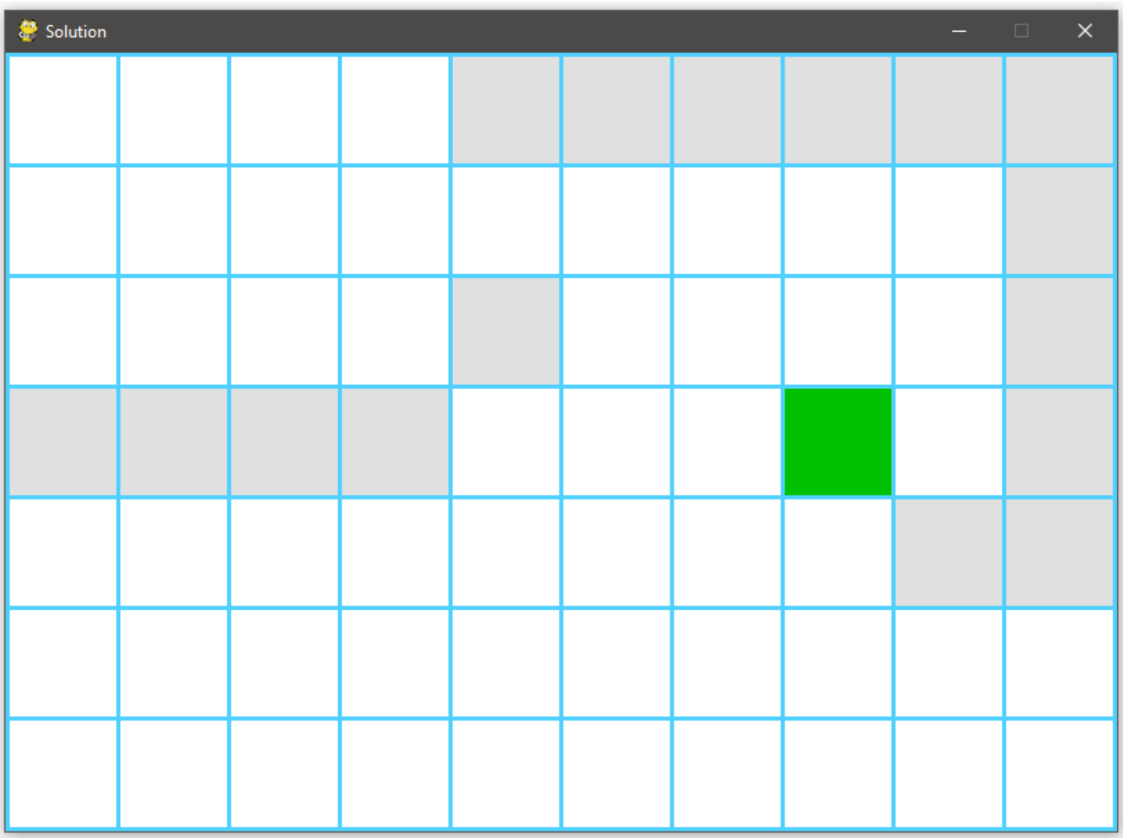
Fourth: Sweeping Episodes

- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 50



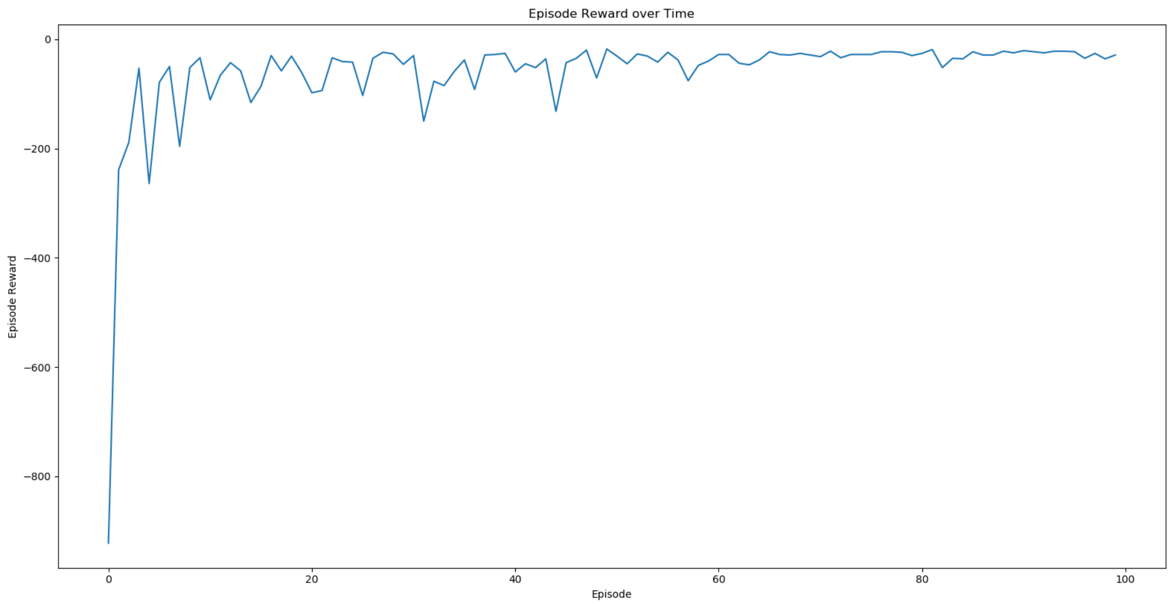


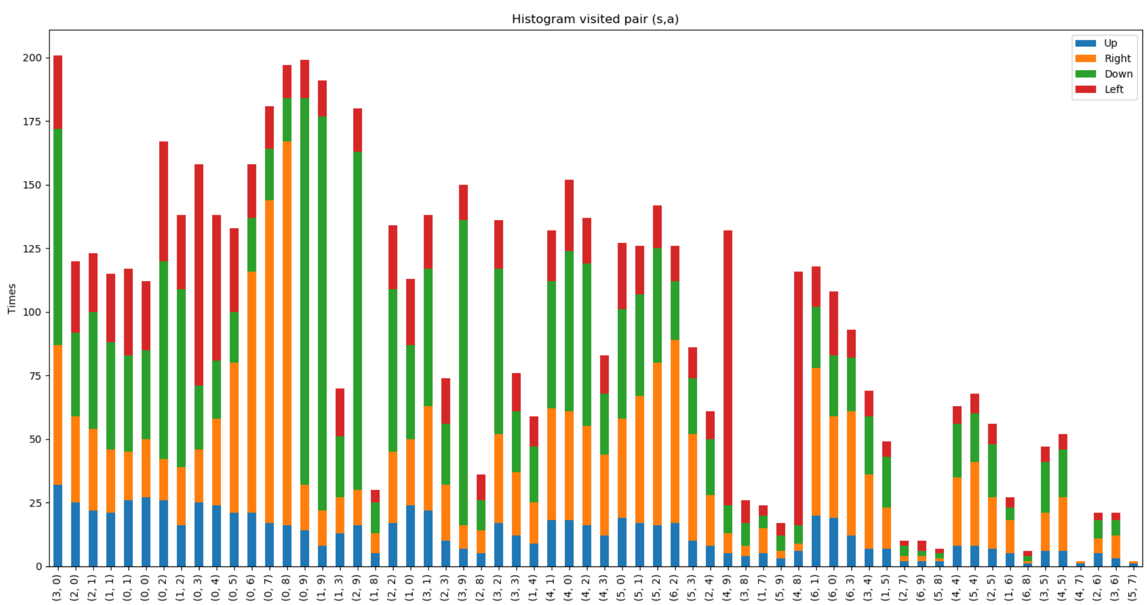

Policy

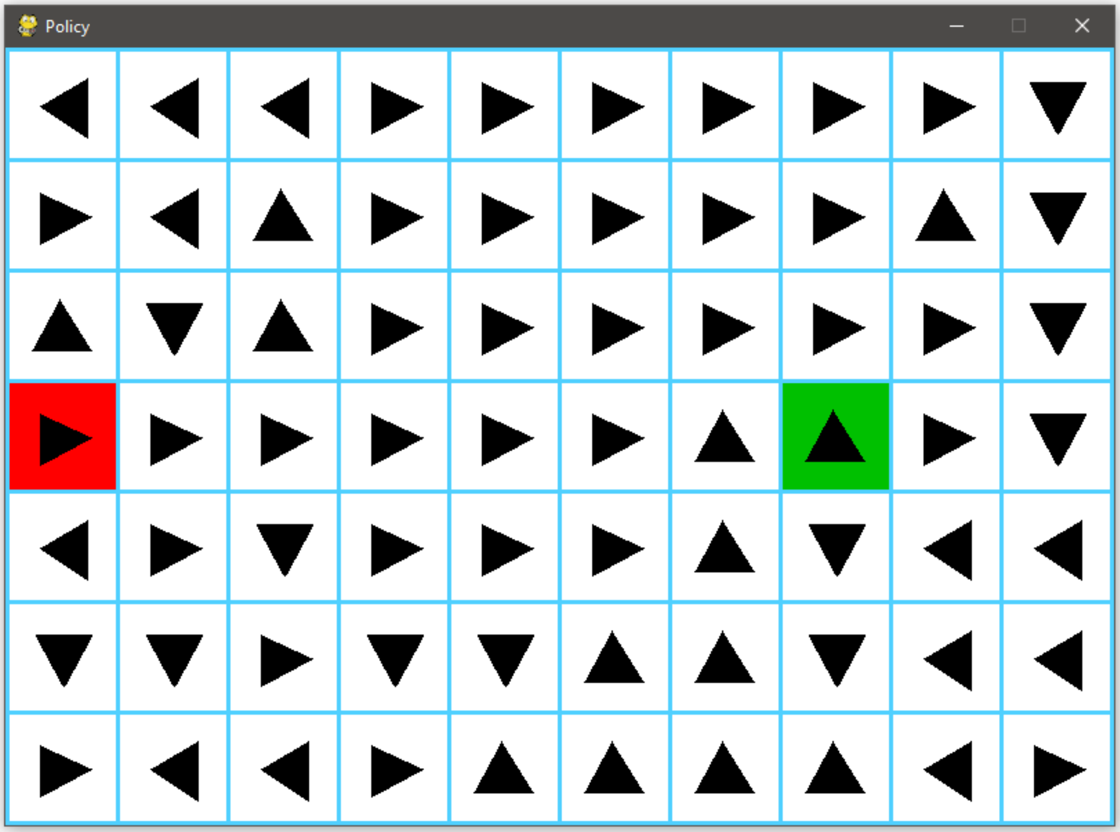

Solution

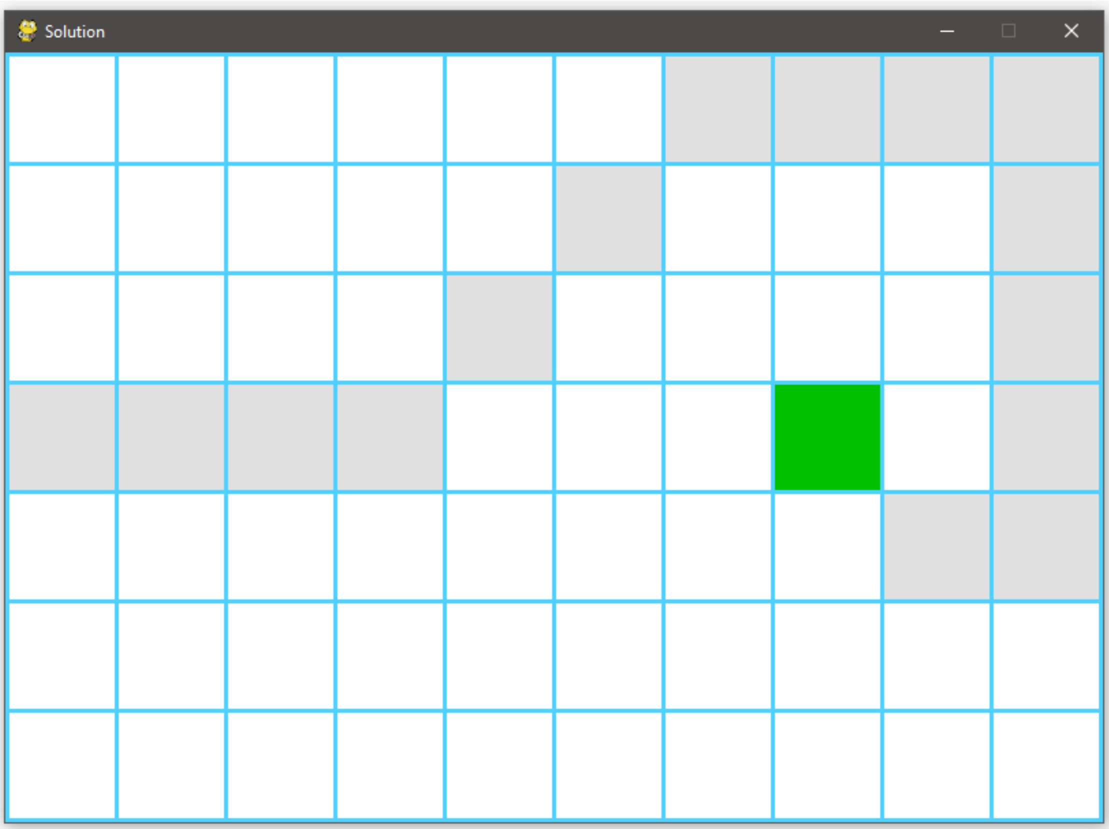
Alpha (α): 0.5

- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 100

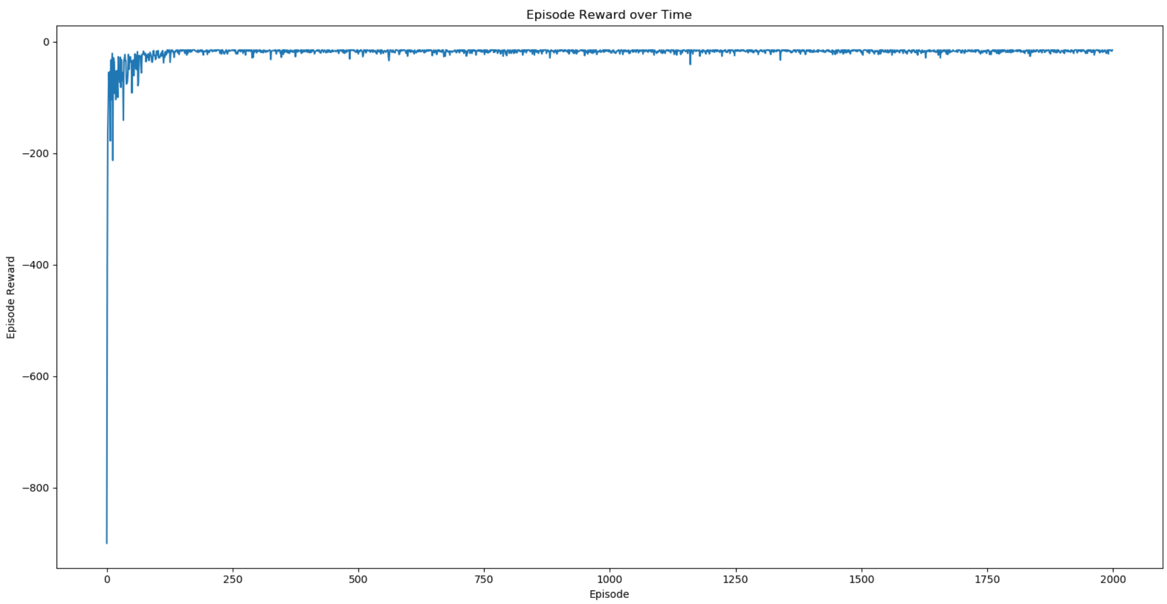


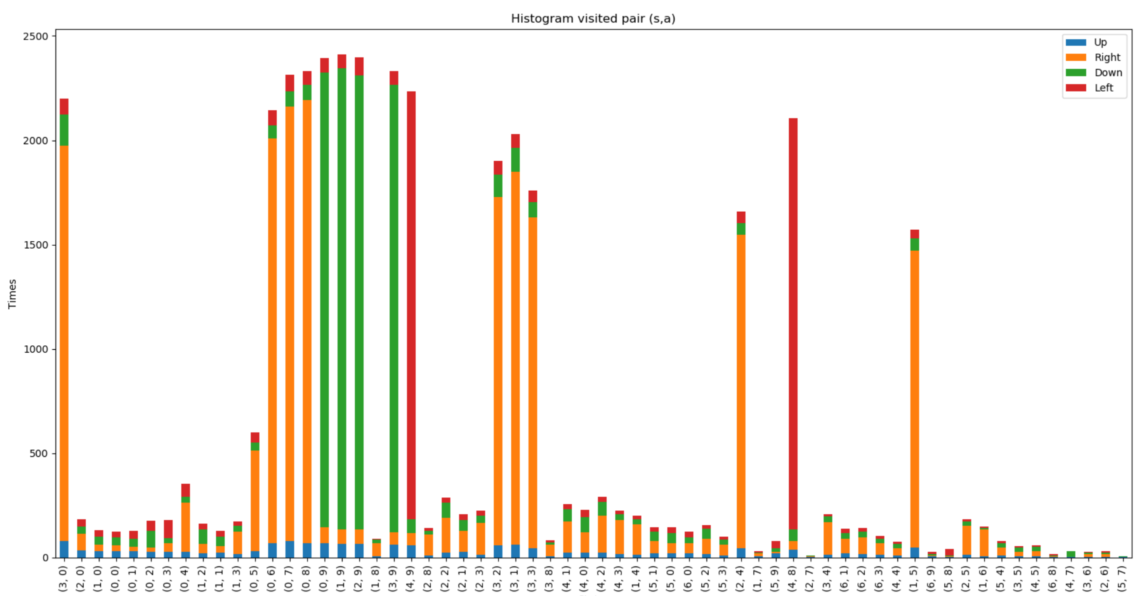


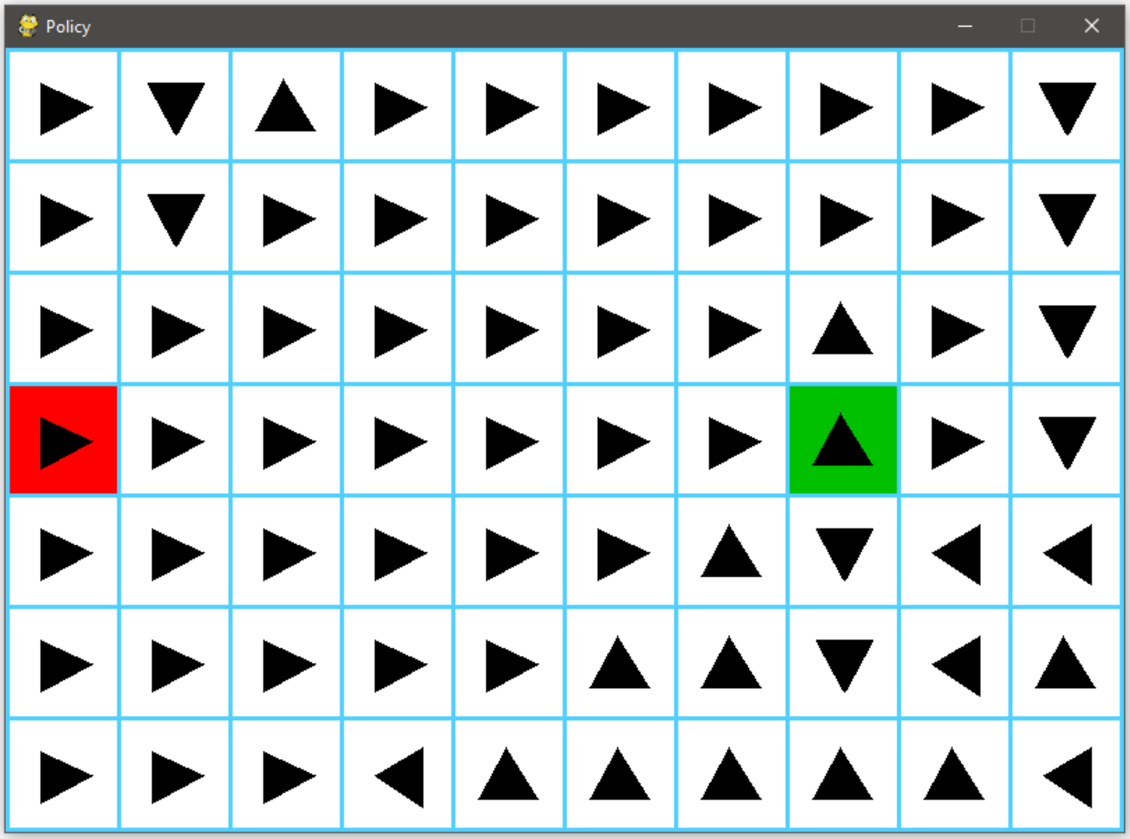

Policy

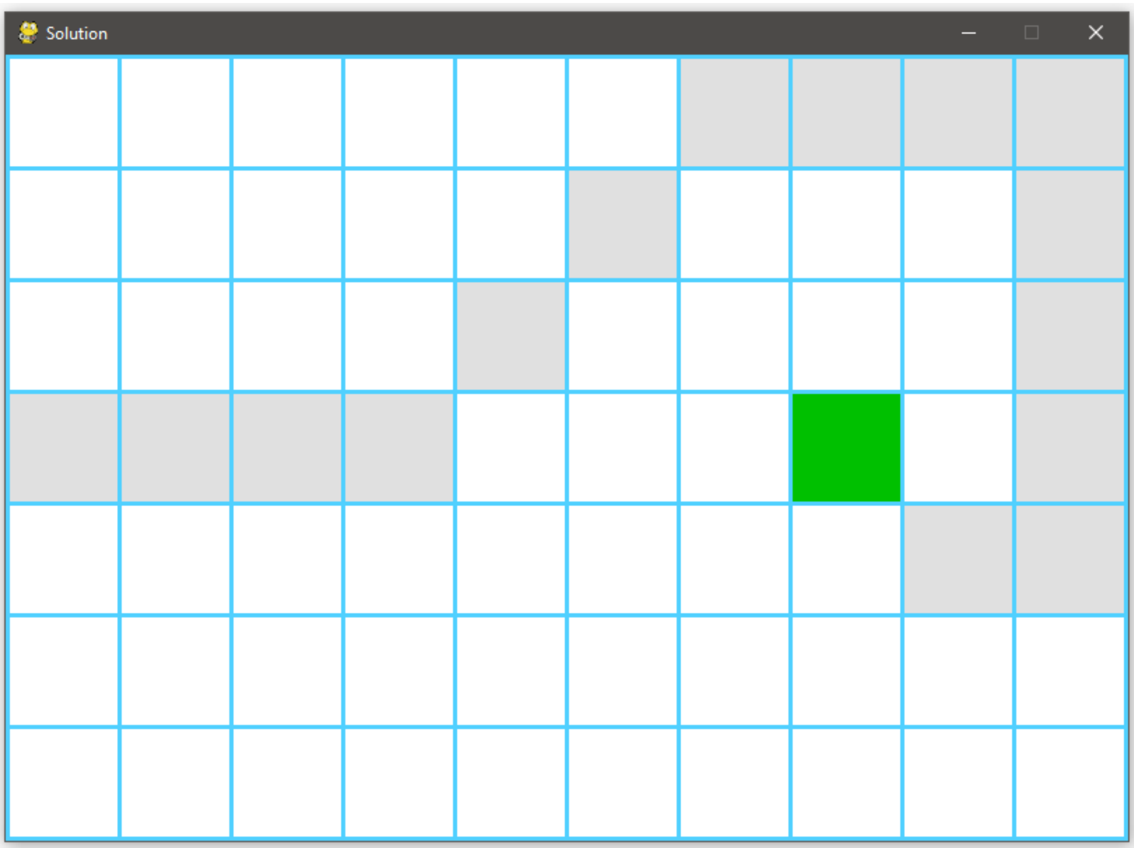

Solution

- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.1
- **Number of Episodes:** 2000






Policy


Solution

Just looking at the average episodes needed to get a path, we can say that the best configuration is:

- **Alpha (α):** 0.5
- **Gamma (γ):** 1
- **Epsilon (ε):** 0.5
- **Number of Episodes:** 300

And looking through all the graphics for every change made in the variables, we can say that:

- A learning rate higher than 0.6 doesn’t makes sense for the current environment.
- A higher gamma doesn’t mean better results, and values around 0.5 could work really good.
- The Epsilon for the greedy algorithm implemented don’t work with values higher than 0.6
- Is not needed more than 500 episodes to get the policy with a good configuration.